Modified node2vec and Attention based Fusion Framework for next POI Recommendation

Akshi Kumar[a], Deepak Kumar Jain [b,c], Abhishek Mallik[d], Sanjay Kumar [d,*]

[a] *Department of Computing, Goldsmiths, University of London, 8 Lewisham Way, London, United Kingdom*
[b]*Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, School of Artificial Intelligence, Dalian University of Technology, Dalian, 116024, China*
[c]*Symbiosis Institute of Technology, Symbiosis International University, Pune, India*
[d]*Department of Computer Science and Engineering, Delhi Technological University, Main Bawana Road, New Delhi-110042, India*

Abstract

The rise of location-based services has led to the widespread adoption of location-based social networks (LBSNs), which play a vital role in making recommendations for the next Point-of-Interest (POI). This paper introduces a modified node2Vec and attention-based fusion framework for the next POI recommendation. We start by pre-processing the raw data to gather the relevant information and present a modified node2vec algorithm to generate the feature vectors for users and locations. These feature vectors are then processed using the attention-based framework. The processed features are then used to create well-labeled and balanced datasets which are grouped by specific time intervals. These datasets are then used for training various ML classifiers which are ensembled in a weighted manner to make an improved fusion based recommendation system.The intensive experimental simulations demonstrate the effectiveness of the proposed framework over existing state-of-art methods.

Keywords: Attention-based framework, Ensemble learning, Information Fusion, Next Point-Of-Interest recommendation system, Location-based Social Networks (LSBNs), Modified node2vec embedding

1. Introduction

. Nowadays, various location-based online social network services like Foursquare, Gowalla, Facebook check-ins, Twitter check-ins, Yelp, etc, have become very popular. Such locations based services help users to share their movements, locations, their views about those locations, etc. Such services have also enhanced the study and analysis of location-based social network (LBSNs) applications [1]. The LBSNs application-based data has also increased the attention of both academics as well as industries toward the next Point-Of-Interest (POI) based recommendation systems. The next POI-

*Corresponding author

Email addresses: Akshi.Kumar@gold.ac.uk (Akshi Kumar), dkj@ieee.org (Deepak Kumar Jain), abhishekmallik265@gmail.com (Abhishek Mallik), sanjay.kumar@dtu.ac.in (Sanjay Kumar)

based recommendation system is one of the prominent research topics finding a lot of business value. Formally, the next POI recommendation system refers to suggesting several locations to users which they can visit next based on their and other users' previous preferences and choices [2, 3]. It helps in improving the user experience and convenience. Most of the traditional work done in the field of next POI recommendation uses conventional classification techniques like Markov chain-based stochastic models [4], Matrix Factorization based methods [5], etc. They use explicit feature engineering techniques to extract relevant information from the data. Various contemporary works have also focused on using techniques like Transitive dissimilarity [6], Collaborative filtering [7], Support vector machines [2], Gaussian Modeling [8], etc. Even though the above-mentioned machine learning-based classical approaches gave good performances, they still require explicit feature engineering. However, feature engineering isn't a trivial task, and it requires huge domain expertise and knowledge. Moreover, it is a tedious process to examine the relevancy and co-relationship among the features. This difficulty in feature engineering has led to the adoption of deep learning-based approaches as they do not require explicit feature engineering and obtain better results.

Over the years, various deep learning techniques have proved their classification prowess for various problem statements like image classification, text classification, etc. [9]. Moreover, deep learning techniques are also helpful in appropriately modeling the complex multi-dimensional relationships between structured and unstructured data. Recent years have seen an increase in the application of deep learning-based techniques for the next POI recommendation. Xu et al. [10] and Chen et al. [3] proposed CNN-based techniques for the next POI recommendations. Recent works have also included recurrent neural network-based techniques like gated recurrent unit (GRU) [11, 12], long short-term memory (LSTM) [13, 14], and vanilla recurrent neural network [15, 16].

The accuracy of the next POI recommendation depends on how accurately the model can represent the interaction between the user and the location. However, most of the existing works in this field have used classical machine learning and deep learning techniques without considering the user-location interaction properly. This can lead to suboptimal performance of the model. Therefore, it is important to explore ways to better model this interaction. Another issue with the existing literature on the next POI recommendation is the lack of attention to the use of node embeddings for generating feature vectors. Node embeddings are useful for inferring the topological and structural details in the user-location relationship, which can provide valuable information for improving the accuracy of the recommendation. By leveraging node embeddings, the model can capture the implicit relationships between the user and the location, which may not be apparent in the raw data. Moreover, it is important to note that machine learning and deep learning models are not always stable throughout and give varying performances based on the datasets and problem statements. This variability in the model's performance can be attributed to the complex nature of the data, which may have high dimensionality, sparsity, and noise. Therefore, it is important to experiment with different models and hyperparameters and validate the model's performance on different datasets to ensure robustness and generalizability.

In this work, we propose a Modified node2vec and Attention-based fusion framework for next Point-of-Interest recommendation. The initial step involves preprocessing the raw data, extracting only the relevant details like user id, location id, and check-in times and discarding any irrelevant information. This helps in increasing the relevancy of the data to the problem statement. Further, we group together the data into time intervals of weeks based on their check-in times. This helps maintain an achievable computational while giving necessary details about the check-ins. The user-location interaction is also modeled using a user-location bipartite graph so that their topological and structural features can also be extracted. Then we present a modified node2vec algorithm for generating the feature vectors for the users and locations in the graph. This helps in optimally exploring the local as well as the global substructure of the graph while also analyzing the user-location, user-user, and location-location associativity. The extracted features are then processed further using an attention component to incorporate the effect of the neighbors on the feature vectors by assigning a suitable degree of importance to each of the neighbors. Datasets are then generated with the processed feature vectors and the labels representing the existence or non-existence of the edges amongst the users and locations in the graph. The dataset is divided into training and testing datasets. The proposed framework is iteratively trained on the training datasets to incorporate the evolving preferences of users and the popularity of locations over time. We train several machine learning and deep learning-based classifiers on the training datasets. Each classifier captures different details from the data and performs differently based on the dataset under consideration. Hence, we ensemble the performance of each of the classifiers using a weighted manner to assign suitable importance to the classifier based on their performance for that particular dataset during the training period. This makes our framework dataset agnostic and gives improved performance. We carry out intensive experiments on several real-life next POI-based datasets and compute various standard performance metrics. We also compare the performance of our framework with several machine learning and deep learning-based baseline algorithms and various contemporary next POI-based recommendation systems. The obtained results highlight the performance efficacy of our framework as compared to other techniques for recommending the next points of interest to the users. The major contributions of this work are as follows.

(i) We propose an improved fusion-based framework for the next Point-of-Interest recommendation using a modified node2vec embedding and attention-based component.

(ii) We introduce a modified node2vec algorithm for feature generation, which helps in appropriately inculcating the topological and structural details of the user-location graph in the feature vectors.

(iii) An attention framework is used to process the generated features, which help in accumulating the effect of the neighbors on the features of the target nodes based on a suitable degree of importance.

(iv) We ensemble various deep learning and machine learning classifiers in a weighted manner to pool together their performance based on the details captured by each of them individually and obtain improved results

(v) The intensive experiments conducted on various real-life datasets reveal the exemplary performance of the proposed framework against various baseline and contemporary next POI recommendation techniques.

The rest of the paper is organized as follows. Some of the previously done work towards the next POI recommendation system is done in Section 2. The details of our proposed work, datasets and evaluation metrics is described in Section 3. The obtained experimental results are discussed in Section 4. Finally, the concluding remarks are mentioned in Section 5.

2. Related Work

In this section, we discuss some of the previously done work in the field of the next POI recommendation systems. Extensive work has been done by researchers on this topic, ranging from Markov Chain models, and collaborative filtering to machine learning and deep learning-based approaches. A heuristic-based approach to generate travel routes and recommend travel packages was proposed by Yu et al. [17]. They combined the historical preferences of the users along with the spatiotemporal details of the data and fed it to a collaborative filtering approach. The heuristic-based approach used by Yu et al. combines historical user preferences and spatiotemporal details to generate travel routes and recommend travel packages. However, the effectiveness of the approach heavily depends on the design and selection of heuristics. The reliance on heuristics may result in suboptimal recommendations in scenarios where the heuristics do not accurately capture user preferences or fail to consider important contextual factors. Zheng et al. [18] used collective matrix factorization to extract the next POIs and corresponding trajectories by utilizing the user-location activity and location attributes. While collective matrix factorization is used by Zheng et al. to extract the next POIs and corresponding trajectories based on user-location activity and location attributes, the performance of this approach can be influenced by the assumptions made in the factorization process. Inaccurate or incomplete user-location activity or location attribute data may lead to suboptimal factorization results and affect the accuracy of the recommended next POIs. Several works have also been done to explore Markov chain models. For example, Chen et al. [19] used topic modeling along with the classical Markov chain model to improve the performance and to generate an interest-aware next POI recommendation system while extracting users' interests. Chen et al. combine topic modeling with a classical Markov chain model to improve the performance of next POI recommendations and extract users' interests.

However, the limitations of this approach lie in the assumptions and limitations of both topic modeling and Markov chain modeling. The accuracy and effectiveness of the approach heavily depend on the quality of the topic modeling results and the assumptions made in the Markov chain model, which may not fully capture the complexity and dynamics of users' interests and their transitions between POIs. Some notable deep learning-based models for the next POI recommendation systems have been introduced. Spatial-Temporal Recurrent Neural Networks (ST-RNN) were proposed by Liu et al. [15]. It captures the spatiotemporal details of the data with the help of a Recurrent Neural Network. It captures the geographical influence and the temporal cyclic effect using a distance and time-based transition matrix, respectively. They train the transition matrix using linear interpolation. While ST-RNN captures spatiotemporal details using a Recurrent Neural Network (RNN), its performance may be affected by the assumptions made in constructing the distance and time-based transition matrix. The linear interpolation used for training the transition matrix may not fully capture the complex dynamics of geographical influence and temporal cyclic effects, potentially limiting the accuracy of next POI recommendations. The "Flashback" model was proposed by Yang et al. [20], which uses a vanilla RNN. They use the spatiotemporal contexts along with the hidden RNN states to utilize the sparse user mobility data. They also consider user embeddings for modeling the user choices and capturing the spatiotemporal effects by taking the weighted average of historical states. The Flashback model utilizes a vanilla RNN and spatiotemporal contexts to handle sparse user mobility data. However, the model's effectiveness heavily relies on the availability and quality of the spatiotemporal context. In scenarios with limited or noisy context information, the model may face challenges in accurately capturing user preferences and spatiotemporal effects. A Hierarchical Spatial-Temporal Long-Short Term Memory (HST-LSTM) was proposed by Kong et al. [21], which encodes the periodicity of user's trajectories using hierarchical modeling. They improve the next POI recommendation capability of their approach by using an encoder and decoder-type architecture to capture a user's past trajectories and preferences. While HST-LSTM incorporates hierarchical modeling to encode the periodicity of user trajectories, its performance may be affected by the assumptions made in the encoder-decoder architecture. The model's capability to capture a user's past trajectories and preferences may be limited by the choice of encoding and decoding mechanisms, potentially resulting in suboptimal next POI recommendations. The temporal and Multi-level Context Attention (TMCA) model was proposed by Li et al. [14]. They use contextual attention to capture the contextual factors and temporal attention to model temporal factors. They also use an LSTM-based encoder-decoder architecture and unify the heterogeneous factors using contextual embeddings.

The TMCA model uses contextual attention and an LSTM-based encoder-decoder architecture to capture contextual and temporal factors. However, the model's performance may be influenced by the complexity and diversity of the contextual factors present in the dataset. The effectiveness of the contextual embeddings and their integration with the LSTM-based architecture may vary depending on the specific dataset and recommendation scenario. Geographical-Temporal Awareness Hierarchical Attention Network (GT-HAN) model was proposed by Liu et al. [22, 23]. They initially proposed GT-HAN [22], which they later improved as GT-HAN [23]. GT-HAN uses the influence of locations, susceptibility of locations, and the spacing amongst the locations to analyze the variation in the geographical co-influence. It uses a spatiotemporal and contextual attention layer. It helps in capturing the geographical co-influence details along with the semantic factors. It explores the spatiotemporal details using the

BiLSTM model and using the contextual attention network for modeling the changing user choices. GT-HAN explores geographical co-influence, semantic factors, and spatiotemporal details using a hierarchical attention network. However, the model's performance may be impacted by the availability and quality of data related to the influence, susceptibility, and spacing among locations. Inaccuracies or limitations in these factors may affect the model's ability to capture the complex dynamics of geographical co-influence and make accurate next POI recommendations. Liu et al. [24] presented a t-LocPred, a time-aware Location Prediction model. It extracts the correlation between the POIs and the time periods using the ConvAoI layer, which is a combination of the CNN layer and the ConvLSTM layer. The CNN layer is used for short-term modeling existing within a day while the ConvLSTM layer is used for the long-term modelling existing within the week. The memattLSTM captures the long-term spatiotemporal correlations using an augmented LSTM model and attention mechanism which are space and time aware, respectively. The memattLSTM filters all the POIs that a user is likely to visit. While t-LocPred considers time-aware modeling using the ConvAoI layer and memattLSTM, the model's performance may be influenced by the assumptions made in the CNN and ConvLSTM layers. The accuracy of the short-term and long-term modeling may depend on the underlying patterns and dynamics within the dataset. In scenarios with atypical or non-repetitive temporal patterns, the model's performance may be limited.

STGN (Spatio-Temporal Gated Network) model was presented by Zhao et al. [25]. It implements a modification to the classical LSTM model by adding two gates to capture short-term choices and two gates for modelling the long-term choices. It also modifies the cell state and there are two types of cell states for short term and long term choices. The authors also present an improved model of the STGN called the Spatio-Temporal Coupled Gated Network (STCGN) model. It has lesser number of parameters which helps in reducing the training time while improving the performance. The STGN model introduces modifications to the LSTM model to capture short-term and long-term choices. However, it may face challenges in effectively capturing the complexities of spatiotemporal dependencies and user preferences in next POI recommendation scenarios. The model's performance may be influenced by the quality and granularity of the input data and the chosen gating mechanisms. Feng et al. [11] proposed the DeepMove model. Instead of next POI recommendation it predicts a similar use case called human mobility. It extracts the spatiotemporal and user features using a multi-modal dense representation framework. The extracted features are passed to a GRU unit to capture the spatiotemporal and user dependencies. The periodicity in user trajectories are captured using an attention module. While the DeepMove model focuses on human mobility prediction, its application to next POI recommendation may have limitations. The model's performance heavily relies on the availability of spatiotemporal and user features, which may not be easily obtainable or accurately captured in all scenarios or datasets. The attention module's effectiveness in capturing periodicity may vary depending on the underlying patterns in user trajectories. The Contextual Attention Recurrent Architecture (CARA) model was proposed by Manotumruksa et al. [26]. It uses a Gated Recurrent Unit along with two different types of gates, namely, the Contextual Attention Gate (CAG) to learn a user's contextual transitions on his choices and Time and Spatial-based Gate incorporate the details about the temporal and spatial distancing amongst successive check-ins. The CARA model utilizes a Gated Recurrent Unit (GRU) and different types of gates to capture contextual transitions, temporal distancing, and spatial distancing. However, the model's performance may be influenced by the granularity and resolution of the spatial and temporal information available. The model may face challenges in effectively capturing fine-grained contextual details and incorporating the diverse factors that can impact user choices in next POI recommendations.

Recently, several hybrid models have been proposed for next POI recommendations. The Geo-Temporal sequential embedding rank (Geo-Teaser) model was proposed by Zhao et al. [27]. This model utilizes the target POI to generate the representations for the context POIs. An objective function is maximized to learn the temporal POI embeddings by the model. The user preference on POIs is learned through the Bayesian Personalized Ranking (BPR) which is used by the geographically hierarchical pairwise preference ranking model. The temporal embedding architecture and the pairwise ranking model are combined in a unified way by the core Geo-Teaser model. The Geo-Teaser model focuses primarily on temporal POI embeddings and pairwise ranking based on geographical hierarchy. It may not effectively capture the diverse contextual factors and user preferences that can influence POI recommendations. Chang et al. [28] proposed a Content-Aware hierarchical POI Embedding (CAPE) model based on Instagram data. The dataset contained check-in details for the POIs as well as a textual description for them. The authors used a context layer and a content layer for capturing the geographical co-influence and the textual descriptions, respectively by passing them through an embedding model. The CAPE model relies on Instagram data, which may introduce biases in the recommendation process due to the specific characteristics and user behavior of the Instagram platform. The model's performance and generalizability to other datasets or domains may be limited. Xi et al. [29] proposed a Bi-directional Spatio-Temporal Dependence and users' Dynamic Preferences (Bi-STDDP) model. Bi-STDDP captures the local temporal dependencies in relationships, intricate global geographical co-influence, and the dynamic preferences of users. They also incorporate the embeddings for locations, embeddings for users and also the temporal patterns and periodicities in their framework. Although Bi-STDDP considers local temporal dependencies, global geographical co-influence, and dynamic user preferences, it may face challenges in effectively capturing complex and evolving patterns in user behavior and preferences. The model's scalability and efficiency in handling large-scale datasets could be a potential limitation. Gao et al. [30] proposed a spatiotemporal graph representation learning method named as GraphTrip. They create a spatiotemporal graph (ST-Graph) by performing a location-aware information fusion. They address the sparsity of periodic regularity using a dual-grained human mobility learning module. They improve the trip inference by fusing the explicit information regarding the trip data as prior knowledge. GraphTrip focuses on spatiotemporal graph representation learning and trip inference. However, the model's performance heavily relies on the availability and quality of the trip data. Limited or noisy trip data can potentially impact the accuracy and reliability of the recommendations. Additionally, the incorporation of explicit trip information as prior knowledge may introduce biases or assumptions that could limit the model's applicability to diverse scenarios. Sáenz et al. [31] proposed a tourist inflow forecasting model at a nationwide scale for Spain. They infer it as an edge prediction task. They model tourist mobility as a graph by fusing heterogeneous tourism data obtained from various sources and infrastructure features. Then they use an ensemble of graph neural networks (GNNs) to make final predictions. While the proposed approach shows promise, its effectiveness may be influenced by the availability and quality of the heterogeneous tourism data obtained from various sources. Inaccurate or incomplete data can affect the model's predictive capabilities. The scalability and computational complexity of the ensemble of Graph Neural Networks (GNNs) used in the model may also pose challenges for real-time or large-scale deployment.

3. Proposed Work

This section illustrates our proposed framework for the next POI recommendation system. We employ a modified node2vec embedding to generate the initial features, which are then further processed by the attention-based embeddings to capture better the neighborhood information provided by the users and POIs. The generated embeddings are then passed through various machine learning and deep learning classifiers, which are then ensembled together to improve performance. The proposed framework is divided into six phases, namely, (i) Preprocessing Phase, (ii) Modified node2vec based feature generation, (iii) Attention-based feature processing, (iv) Dataset Creation, (v) Training the machine learning and deep learning classifiers, (vi) Ensembling the classifiers and making recommendations. Fig. 1, shows the flowchart of our proposed next POI recommendation system. The various phases of our proposed framework are described below.

3.1. Data Preprocessing

Most of the next POI based datasets exist in raw form and contain various information like the GPS coordinates of the locations, attributes of the locations, attributes of the users, etc. However, some of the datasets include only a subset of these information and not all of them. This creates an inconsistency amongst the datasets. To overcome this and achieve uniformity across the datasets, we preprocess the raw data. We only keep the user id, location id, and check-in times while removing all the other attributes. After preprocessing, we obtain datasets having information about users checking in to locations over various timestamps. However, processing the dataset over each timestamp is computationally expensive and doesn't provide information about the behavior of the users or the popularity of the locations. Hence, we group the timestamps into various time intervals over weeks. This helps our framework capture enough information about the users and locations while maintaining computational feasibility. Thereby, we obtain uniform datasets separated in T time intervals. We denote the preprocessed dataset as $\mathcal{D} = (u_i, l_i, t_i)_{i=1}^{N}$, where u_i represents the user ID, l_i represents the location ID, and t_i represents the timestamp at which the user checked in to the location. In addition, we denote the set of unique users and locations in

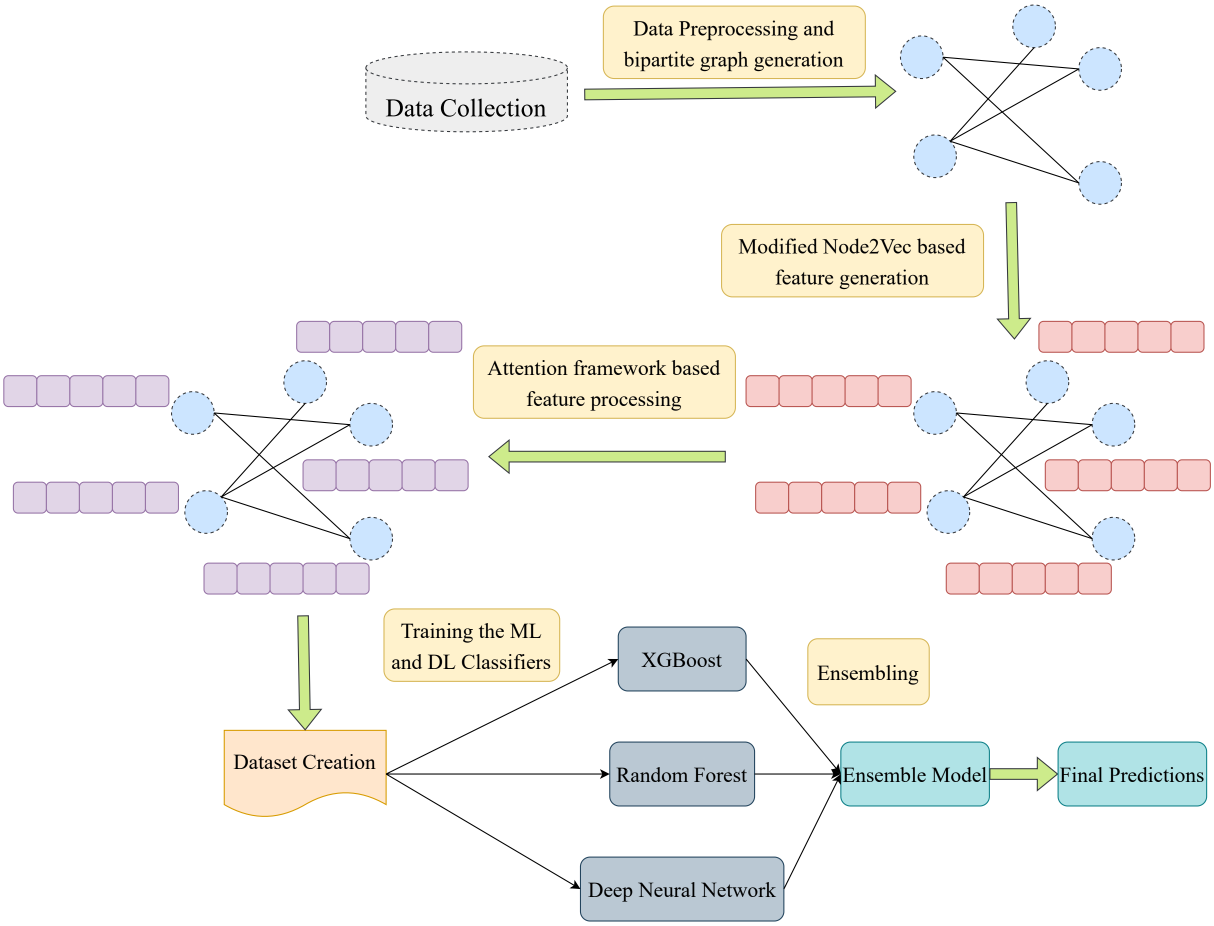

Figure 1: The flowchart of proposed framework for next POI recommendation system.

the dataset as $\mathcal{U}$ and $\mathcal{L}$, respectively. Finally, we denote the set of time intervals as $\mathcal{T} = T_1, T_2, \ldots, T_T$, where T_t represents the t-th time interval over weeks.

3.2. Modified node2vec based Feature Generation

The preprocessed datasets obtained in the previous phase contain no features defining the users or the locations. To obtain the feature set, we generate graph for each interval of the datasets and then exploit the topological and structural details of the formed graphs to generate the features for the users and locations. For this purpose, we proposed a modification to the existing node2vec graph embedding [32]. This generates the feature vectors for the nodes in the user-location graph by representing them in a lower dimensional feature space of dimension d, where d is the length of the feature vector. This helps in capturing the details related to the places that a user has visited along with the popularity of the various locations.

As part of our proposed modified node2vec feature generation process, we start by preprocessing the graph and generating the transition probabilities. The transition probabilities are then used to stimulate the random walk which are then used to generate the feature vectors for the nodes in the user-location graph. The generated feature

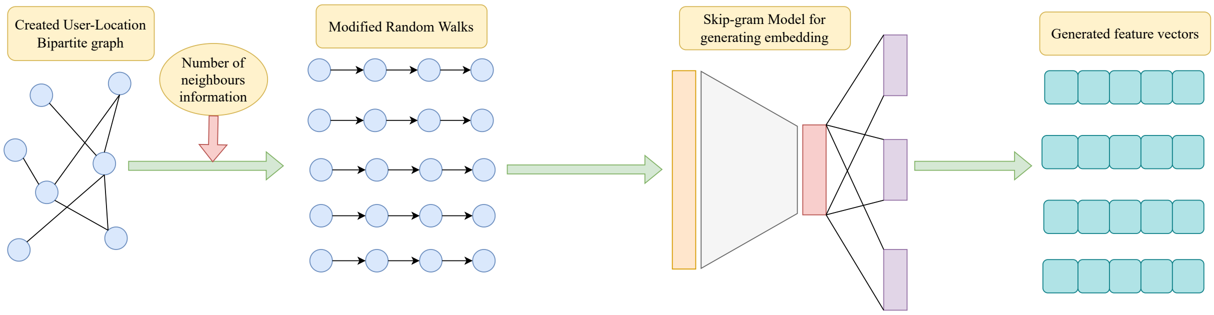

Figure 2: Flow diagram of our proposed modified node2vec feature vector generation algorithm.

vectors are further optimized using the Stochastic Gradient Descent (SGD). The random walks are guided by two control parameters, namely, the return parameter (p) and the in-out parameter (q). The probability of a node being immediately revisited in the random walk is controlled by the return parameter (p). While the random walk is controlled to be more localised or globalised using the in-out parameter (q). Let there be a random walk which covers the trajectory of going from node u to node v. Let the random walk be going to the node w next. The transition probability for such a random walk can be calculated using the search bias, $\alpha_{pq}(u,s)$ which is represented below by Eq. 1.

$$\alpha_{pq}(u,w) = \begin{cases} \frac{1}{p}, \, d_{uw} = 0 \\ 1, \, d_{uw} = 1 \\ \frac{1}{q}, \, d_{uw} = 2 \end{cases} \tag{1}$$

Here, the shortest path between the nodes u and w is represented by d_{uw}. The shortest distance d_{uw} can take value from $\{0,1,2\}$. The probability of choosing a node w as the next step in the trajectory in the random walk is represented by the unnormalised transition probability. The search bias is then used to calculate the unnormalised transition probability using Eq. 2. In Eq. 2, the weight of the edge (v,w) is represented by wt_{vw}. The value of wt_{vw} is 1 for the unweighted graphs.

$$\pi_{vw} = \alpha_{pq}(u,w) \cdot wt_{vw} \tag{2}$$

The processed next POI based user-location graphs are unweighted and the weight of the edges are 1. Hence, the factor wt_{vs} isn't relevant to this study. Thus, in the proposed modification to the classic node2vec, to calculate the unnormalised transition probability, we use the total number of neighbors of the node w instead of using the weight of the edge (u,w). This is denoted by $N(w)$ and Eq. 2 modifies to 3.

$$\pi_{vw} = \alpha_{pq}(u,w) \cdot N(w) \tag{3}$$

Then we use the Eq. 4 to select the next node in the random walk. Here, the i^{th} node in the random walk is denoted by c_i, η represents the total number of transition probabilities and is the normalisation constant. Whereas the set of all edges is represented by E. The random walks generated thereby are used to train the Skip-gram model of the word2vec and SGD (Stochastic Gradient Descent) technique is used to improve the performance and the feature vectors for each node in the user-location graph is generated. The length of the feature vectors is d, representing the d dimensional plane in which the nodes are represented. Hence, with the help of our proposed modified node2vec technique, we obtain a feature matrix, $M^{n \times d}$ having dimensions $n \times d$, where n is the total number of nodes in the user-location graph. Fig. 2, shows the flow diagram of our proposed modified node2Vec algorithm based feature generation technique. The generated feature vectors are further processed and used in the next steps.

$$P(c_i = w \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vw}}{\eta}, (v,w) \in E \\ 0, otherwise \end{cases} \tag{4}$$

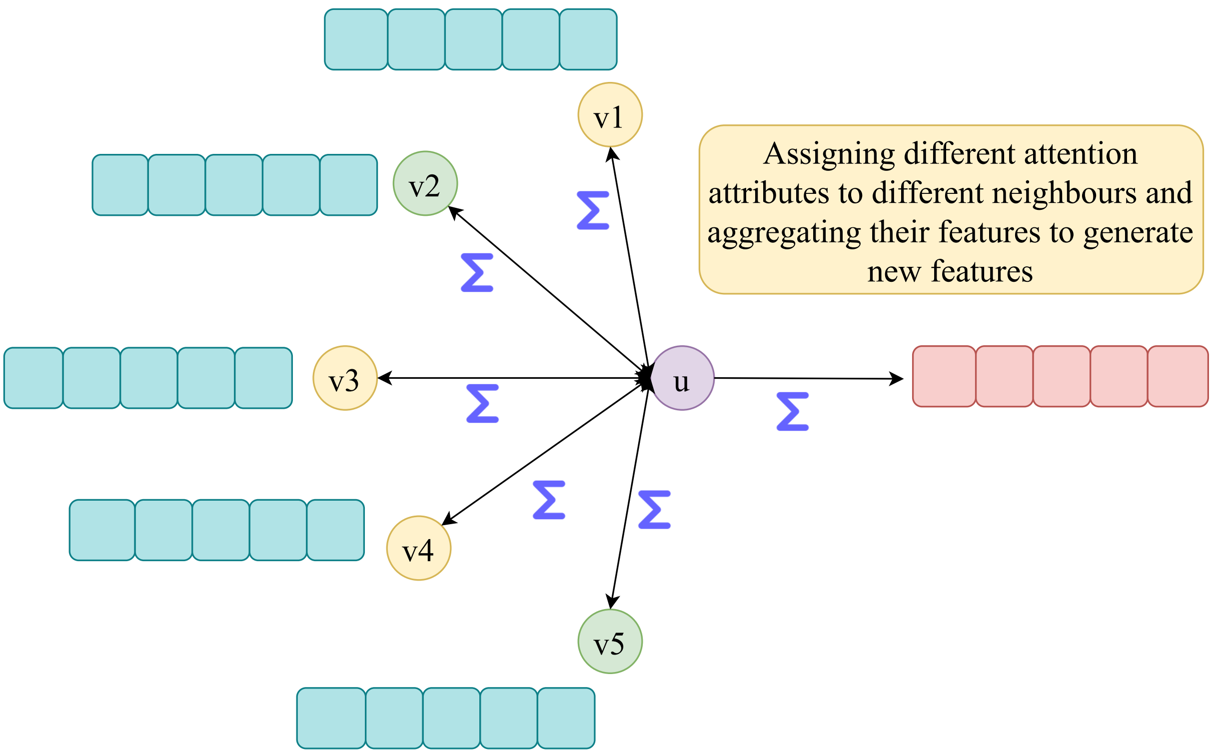

Figure 3: The proposed attention based feature processing technique

3.3. *Attention based Feature Processing*

The feature vectors generated in the previous step are further processed in this step to better capture the neighborhood details. It also helps placing the similar nodes closer while the dissimilar nodes farther away in the d dimensional feature space. This highlights the associativity amongst the users and locations. For this purpose, we modify the works presented by Bahdanau et al. [33] and Veličković et al. [34], to employ an attention based component to efficiently incorporate the neighborhood details in the feature vectors. As part of our proposed modification, we use the modified node2vec based feature vectors which had been generated in the previous step along with the attention component to obtain an edge based feature vector which efficiently represents the linkage amongst the connecting nodes. The used attention component helps our proposed technique to suitably assign varying degrees of importance to the individual contributions made by the neighboring nodes of the target node. We start by generating a randomly initialised parametrised feature matrix $P^{d \times d}$. This is used to assign random weights to the various elements of the feature vector. Thereby, we obtain a weighted feature matrix S as per Eq. 5.

$$S^{n \times d} = M \cdot P^T \tag{5}$$

The generated weighted feature matrix (S) is then used to obtain a node connectivity matrix S'. The feature connectivity matrix is a 3 dimensional matrix containing the feature connectivity vectors for each node pair u and v. More formally, if an edge exists between the nodes u and v then their feature connectivity vector (S'_{uv}) is formed by concatenating the weighted feature vectors S_u and S_v. However, if there is no edge between the nodes u and v then their feature connectivity vector (S'_{uv}) is represented by Z, a zero populated 1 dimensional vector of length $2d$. The dimension of the feature connectivity matrix is $n \times n \times d$. The above process is represented in Eq. 6.

$$S'_{uv} = \begin{cases} S_u \oplus S_v, \text{ edge exists between } u \text{ and } v \\ Z, \text{edge doesn't exist between } u \text{ and } v \end{cases} \tag{6}$$

We then perform the dot product of the feature connectivity matrix S' with an attention matrix A, which has a dimensionality of $1 \times 2d$ and is randomly initialised. This helps in assigning the degree of importance to the neighbors of the target edge. The above process generates an attention feature matrix λ which is shown in Eq. 7.

$$\lambda = S' \cdot A^T \tag{7}$$

The obtained attention feature matrix (λ) is used to generate an feature coefficient matrix, Λ as per Eq. 8. We then use softmax function to normalise this matrix and obtain a normalised feature coefficient matrix, α. This is shown in Eq. 9 Here, $\Gamma(i)$ represents the set of the neighbors of the node i.

$$\Lambda = LeakyReLU(\lambda) \tag{8}$$

$$\alpha_{uv} = \begin{cases} \frac{e^{\Lambda_{uv}}}{\sum_{k \in \Gamma(u)} e^{\Lambda_{uv}}}, \Lambda_{uv} \neq 0 \\ 0, \Lambda_{uv} = 0 \end{cases} \tag{9}$$

Finally, we use neighborhood aggregation to generate the processed feature vectors F for every user and location in the user-location graph. This helps in neighborhood structure and connectivity in the processed feature vector of the target node by assigning a suitable degree of importance to the nodes in its neighborhood. This is shown in Eq. 10.

$$F_u = \sigma\left(\sum_{k\in\Gamma(u)} \alpha_{uv} S_v\right) \tag{10}$$

The attention based processed feature vectors vividly characterise a user or a location based on the locations that a user has visited or the users that have visited a location, respectively. This helps in recommending places to users based on the similarities amongst the users as well as the similarities amongst the locations. In simpler terms, it helps our model in extracting the user-user and location-location similarity more efficiently. Fig. 3, shows the mechanism of our proposed attention based feature processing technique.

3.4. Formulation of Dataset

This phase illustrates the process of dataset creation for our proposed framework. We split our processed dataset into training and testing datasets to train our framework and then evaluate its recommendation capabilities on the testing dataset. Following the notion that a recommendation system uses the historical information about the users and the location to make the future recommendations, we reserve the initial T' time intervals for training while the later intervals are used for testing. More formally, the training dataset comprises of $0-T'$ time intervals while the testing dataset consists of $T'-T$ time intervals. Let D_t be the dataset for the t^{th} time interval. The attributes for the dataset D_t are the processed feature vectors of the users and locations as calculated in the previous phase. While the existence or non-existence of the edges amongst the users and locations for the $t+1^{th}$ time interval act as the labels for the dataset. If an edge exists then the value of the label is 1 else it is 0 for non-existent edge as per Eq. 11. Here, E_t is the set of edges for the user-location graph in the t^{th} time interval. To understand the evolving associativity and dissociativity amongst the users and locations, we iteratively train our model over the time intervals belonging to the training dataset. A testing dataset D'_t is also generated for the t^{th} time interval as per the above discussion. Hence, we obtain training datasets $D_t, t \in [0, T')$ and testing datasets $D'_t, t \in [T', T)$. The mathematical definitions of the training and testing datasets is given in Eq. 12 and Eq. 13, respectively. Here, $\oplus$ refers to the concatenation of the vectors.

$$label(u,v) = \begin{cases} 1,\ (u,v) \in E_t \\ 0,\ (u,v) \notin E_t \end{cases} \tag{11}$$

$$D_t = \{F_u \oplus F_v \oplus label(u,v)\},\ (u,v) \in E_t\ and\ t \in [0, T') \tag{12}$$

$$D'_t = \{F_u \oplus F_v \oplus label(u,v)\},\ (u,v) \in E_t\ and\ t \in [T', T) \tag{13}$$

3.5. Training the Machine Learning & Deep learning Classifiers

After successful creation of well-labelled and well-balanced datasets in the previous step, now we move onto training several machine learning and deep learning based classifiers. For this study, we use XGBoost, Random Forest, and deep neural network. The choices for these classifiers have been made so that we can utilize the qualities of an ensemble based classifier, a boosting algorithm and an artificial neural network. Once these models are trained on the training datasets (D_t) then we use the trained models to make recommendations on the testing datasets (D_t'). The results obtained by each of these classifiers are then ensembled together in a weighted manner to achieve improved performance. The parametric details adopted while training the classifiers are given below.

1. XGBoost classifier: XGBoost is an algorithm which implements the gradient boosted decision trees. XGBoost adopts a sequential method to generate decision trees by laying emphasis on the weights assigned to the variables. All independent variables are assigned random weights and fed to decision trees for predicting the target values. The weights assigned to the under-performing variables is increased and they are again fed into decision. This helps in optimizing the weak learners to give a more efficient and effective performance. For training the XGBoost classifier, we use 200 estimators, a maximum depth per tree as one. Decision tree based classifier is used as the weak classifier. The trained XGBoost classifier can be represented by Eq. 14.

$$XGB = XGBoostClassifier(D_t) \tag{14}$$

2. Random Forest classifier: Random Forest is a technique of training multiple decision tree classifiers in parallel to reduce the individual variance of each decision tree and then use majority voting to obtain the optimal performance. Since, XGBoost classifier and Random Forest classifier use different methodology for training the decision trees, so they complement each other and generate improved and generalised results. We use 100 estimators and six as the maximum depth per tree for the Random Forest classifier. $Gini$ is used as the splitting criterion. The trained Random Forest classifier can be represented by Eq. 15.

$$RF = RandomForestClassifier(D_t) \tag{15}$$

3. Deep Neural Network: Most of the real-life datasets and interactions exhibit complex relationships and a deep neural network performs very well in learning such complex relationships and dependencies. It contains an input layer, several hidden layers, and an output layer. Each layer consists of a group of several neurons. The output of a neuron H_i can be represented by Eq. 16. Here, the activation funstion is represented by σ, the number of neurons in the previous layer are represented by N, the weights are represented by W_{ij} while the input to the current layer is represented by x_j. We train the deep neural network for 100 epochs having 64 as the batch size. We use $Softmax$ as the activation function. $Binarycross-entropy$ is used as the loss function

while the optimizer is chosen to be *Adam*. The trained Deep Neural Network classifier can be represented by Eq. 17.

$$H_i = \sigma(\sum_{j=1}^{N} W_{ij} x_j) \tag{16}$$

$$DNN = DeepNeuralNetworkClassifier(D_t) \tag{17}$$

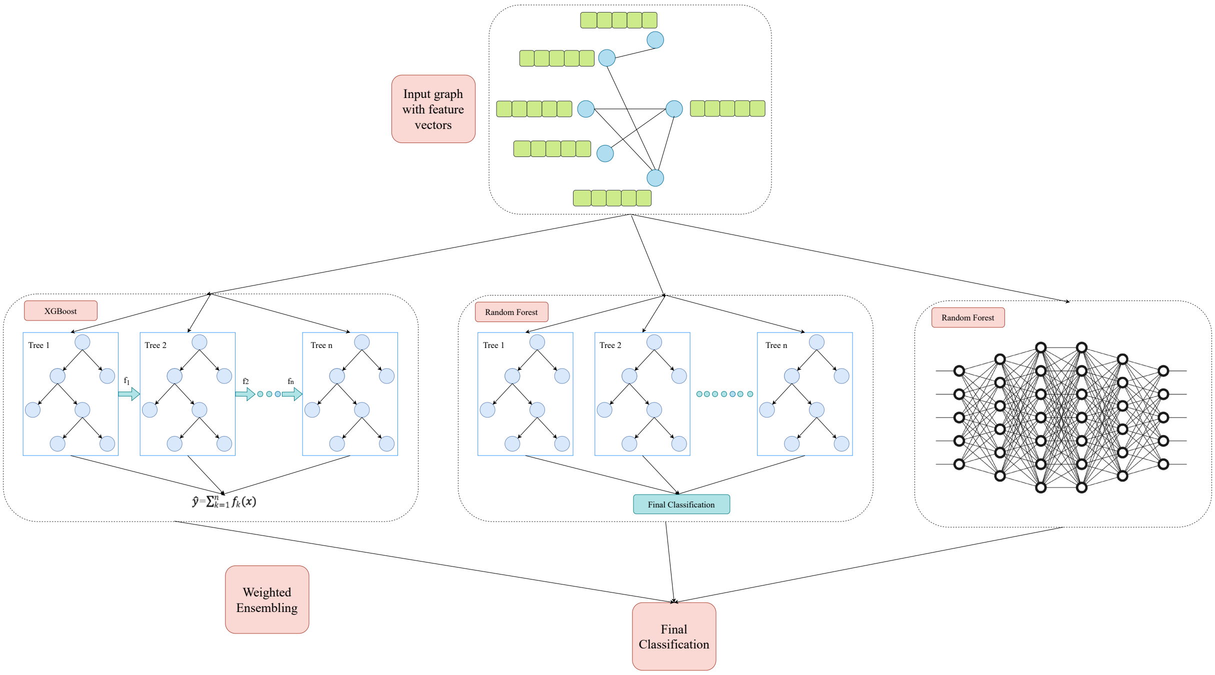

Figure 4: The weighted ensembling framework for combining the various machine learning and deep learning classifiers.

3.6. *Fusion of Classifiers and Making Recommendations*

Once all the machine learning and deep learning classifiers are trained, we use the trained classifiers to make recommendations and then ensemble the results obtained by them to generate a fusion framework based recommendation. The results are ensembled in a weighted manner by assigning weights to each classifier based on their performance on the last time interval of the training phase, i.e., $T' - 1$. Let CL be a generic classifier and ω_CL represent the weight assigned to the classifier. Then Eq. 18 represents the equation to evaluate the weight. Fig. 4, shows the weighted ensembling framework for combining the various machine learning and deep learning classifiers.

$$\omega\_CL = \frac{\Sigma_{k=1}^{k=len(D_{T'-1})} \left\lfloor \frac{CL(D_{T'-1}[k] \backslash L[k]) + L[k]}{2} \right\rfloor}{\Sigma_{k=1}^{k=len(D_{T'-1})} C(D_{T'-1}[k] - L[k])} \tag{18}$$

The above described weight function $\omega_C L$ assigns suitable importance to each classifier based on their recommendation accuracy during the training phase. The classifiers having better accuracy are more profoundly represented in making the final recommendations to obtain an improved performance. We don't average out the results obtained by all the classifiers as some classifiers perform very poorly over some problem statements or datasets, so taking their generalised average might diminish the performance instead of improving it. However, based on the Eq. 18, assigning a higher weight to the better performing classifiers might increase the performance. While taking into consideration the results obtained by the poor performers, we reduce bias and obtain a generalised result. Here, length of the dataset $D_{T'-1}$ is represented by $len(D_{T'-1})$, while the label for the k^{th} entry in the dataset is represented by $L[k]$. The recommendations made by the classifier CL without considering the labels attribute is represented by $CL(D_{T'-1}[k] \setminus L[k])$. Let τ represent the dataset without the label attribute. So the recommendations made by all the classifiers can be ensembled as per Eq. 19. While the final recommendations for the k^{th} data point ($\tau[k]$) are made as per Eq. 20. The recommendations made thereby are used to evaluate and compare the performance of the proposed framework.

$$\Upsilon[k] = \frac{XGB(\tau[k]) * \omega\_XGB + RF(\tau[k]) * \omega\_RF + DNN(\tau[k]) * \omega\_DNN}{\omega\_XGB + \omega\_RF + \omega\_DNN} \tag{19}$$

$$Recommendation[k] = \begin{cases} 1, \ \Upsilon[k] \geq 0.5 \\ 0, \ \Upsilon[k] < 0.5 \end{cases} \tag{20}$$

We introduce the diversity and serendipity properties to our model by incorporating the exploration and exploitation techniques in our framework in the form of the in-out parameters of our modified node2vec algorithm. It helps our framework in focusing on both local exploitation as well as global exploration. Moreover, a good mix of datasets from across the globe like NYC, Tokyo, etc. helps our model in being more generalized and achieving better performance across the datasets. The appropriate use of the attention framework also helps our model in assigning suitable importance to the neighboring users and locations to model the users' preferences.

3.7. Datasets and Evaluation Metrics

In this section, we illustrate the details of the various datasets and evaluation metrics used by us for running simulations and estimating the performance of our proposed Modified node2vec and Attention based Ensemble Framework for Next POI Recommendation. We use various real-life datasets along with several standard performance metrics which have been used extensively in the recent literature.

3.7.1. Datasets

In this section, we describe the various datasets used by us for our study. We use six real-life next POI recommendation based datasets. The datasets belong to different domains like restaurant check-ins, travel check-ins, etc. The datasets also have varying topologies, dimensions, sizes, etc. The description of the datasets are as follows.

1. Gowalla [35]: Gowalla (https://go.gowalla.com/) is a Point-Of-Interest based social networking website wherein users interact by sharing their check-in locations. It is collected over a period ranging from Feb. 2009 - Oct. 2010. It represents an undirected interaction network consisting of 53008 users, 121944 locations, and 3302414 check-ins.

2. NYCRestaurant [36]: The NYCRestaurant dataset is collected from Foursquare (https://foursquare.com/) website. It represents the check-ins made by the users across restaurants in New York City. The dataset is composed over a period ranging from 24 October 2011 to 20 February 2012. It includes 3112 users, 3298 restaurants, and 27149 check-ins.

3. NYC [37]: It is also a dataset collected from the Foursquare (https://foursquare.com/) website. It contains information about the places checked-in by users across the New York City. It is collected over a period ranging from from 12 April 2012 to 16 February 2013. It contains 1064 users, 5136 locations, and 147939 check-ins.

4. TKY [37]: This is also a dataset collected from the Foursquare (https://foursquare.com/) website. It contains information about the places checked-in by users across the Tokyo City. It has 2245 users, 7872 locations, and 447571 check-ins.

5. Brightkite [38]: Brighkite (http://www.brightkite.com/) is also a next POI based social network wherein users interact by sharing their check-in locations over the platform. The data is collected over a period ranging from Apr. 2008 - Oct. 2010. The dataset has 11142 users, 4369 locations, and 100069 check-ins.

6. Yelp [39]: Yelp (https://www.yelp.com/) is a social networking site wherein users share their reviews, opinions and experiences over various locations that they have checked in. It has 30887 users, 18995 locations and 860888 check-ins.

3.7.2. Evaluation Metrics

In this section, we describe the various evaluation metrics used by us for estimating the performance of our proposed Modified node2vec and Attention based Ensemble Framework for Next POI Recommendation. We use several standard performance metrics which have been extensively used by researchers in the recent literature. We use Area Under the Receiver Operating Characteristic Curve (AUC), Precision, Recall, Mean Reciprocal Rank (MRR), and F1 Score. The description of the performance metrics are as follows.

1. AUC:
   It is a very popular performance metrics for recommender system and is as important as accuracy is for classification tasks. It stands for the Area Under the Receiver Operating Characteristic Curve (AUC). For a binary classifier, it is used to understand the diagnostic capability. Its range varies from 0.0 to 1.0. The higher the AUC value, the higher the algorithmic accuracy is achieved. Here, N is the number of independent comparisons, N' is the number of missing links having a higher score while N'' is the number missing and nonexistent links having the same score.

$$AUC = \frac{N' + 0.5N''}{N} \tag{21}$$

2. Precision:
   Precision is anther important metrics that is used for recommendation tasks. The precision values for a next POI recommendation system refers to the measure of the POIs recommended to the users which they actually visited. More formally, it refers to the ratio of the number of positive classifications made by the classifier which were actually true to the total number of positive classifications made by the classifier. Mathematically, it can be represented as Eq. 22

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

3. Recall:
   Recall is also a very important metrics that is used for recommendation tasks. The recall values for a next POI recommendation system refers to the measure of the number of POIs recommended to the user from the total number of POIs visited by them. More formally, it refers to the ratio of the number of positive classifications made by the classifier that actually came up to be true. Mathematically, it can be represented as Eq. 23.

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

   Here, TP represents True Positive or the existent links that were predicted as correctly. FP is False Positive, representing the non-existent links that were predicted as existent. While FN is False Negative, showing the existent links which are predicted as non-existent.

4. Mean Reciprocal Rank (MRR):
   Mean Reciprocal Rank is another suitable metrics to estimate the recommendation capabilities of a recommender system. It has been extensively popular amongst the tasks which have been used to generate a list of appropriate choices called recommendations. The recommendations made are ranked as per their suitability to the problem statement. Mathematically, MRR represents the average of the inverse of the ranks of the answers. More formally, it can be defined by Eq. 24.

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{24}$$

5. F1 Score:

   F1 score is a metric that combines both precision and recall using their harmonic mean to give more balanced information regarding the model's performance. More formally, it can be defined by the Eq. 25

$$\text{F1 Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{25}$$

4. Experimental Results

In this section, we discuss and analyse the experimentations performed by us to infer the performance of the proposed Modified node2vec and Attention based Ensemble Framework for Next POI Recommendation. We have conducted extensive experiments on several real-life datasets mentioned in Section 3.7.1. We have also evaluated various popular performance metrics mentioned in Section 3.7.2 to estimate the capabilities of our proposed framework. As part of our study, we compare the performance of our model with several baseline machine learning and deep learning algorithms like Logistic Regression, Multinomial Naive Bayes, Decision Trees, Random Forest, XGBoost, and Deep Neural Network. We also compare the performance of our model with several contemporary next POI recommendation systems, namely, STRNN [40], DeepMove [41], STGN [42], ARNN [43], LSTPM [44], TiSASRec [45], GeoSAN [46]. We have chosen the training dataset to be 80% of the total dataset. This gives us sufficient data for training while leaving enough testing dataset to obtain a good evaluation of the model performance.

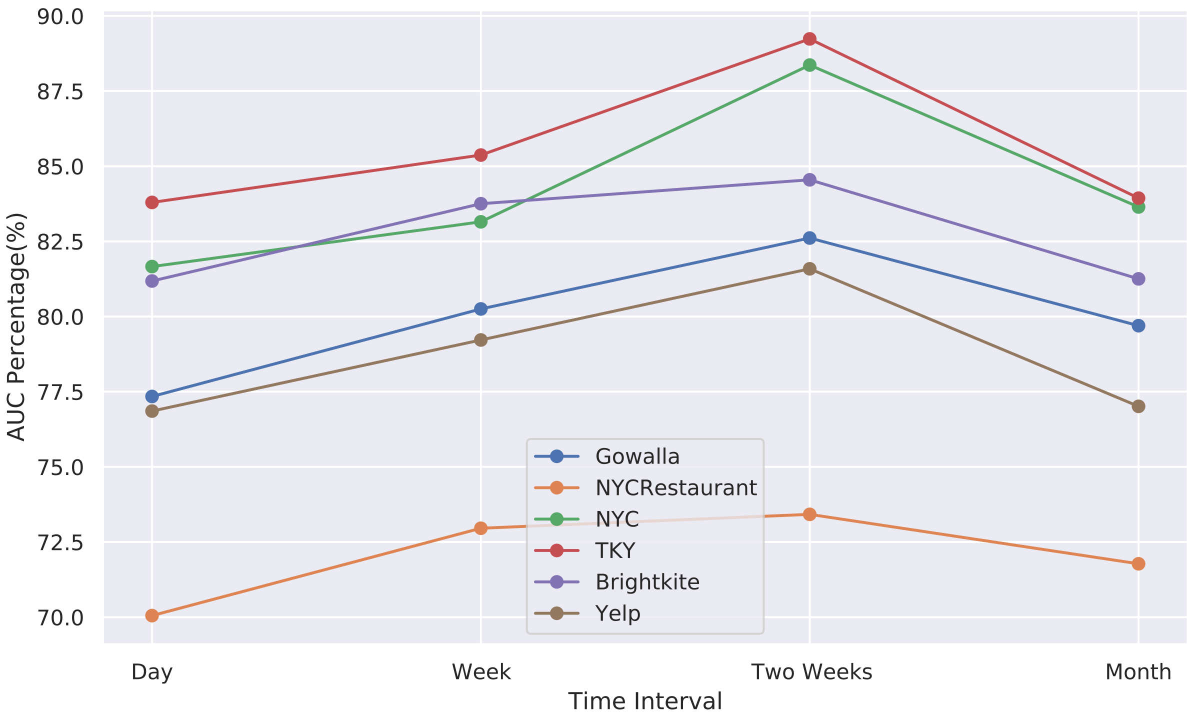

Figure 5: The effect of time interval on the AUC values obtained by our proposed framework on various datasets.

4.1. The Sensitivity to Time Interval

As discussed in Section 3.1, we group our dataset into sets of fixed time intervals. This is done to enhance comprehensibility of the data by our model while maintaining appropriate efficiency. In this section, we investigate the sensitivity of our proposed model to varying time intervals and aim to determine the most appropriate time interval. We run experiments for four different time intervals, namely, Day, data is grouped for each day, $Week$, data is grouped for each week, $Two\ Weeks$, data is grouped for two weeks, and finally $Month$, data is grouped for every month. The obtained results for various time intervals across all the datasets for our proposed framework are shown in Fig. 5. From Fig. 5, we see that as the time interval increases the performance of our framework also increases. This can be attributed to the amount of data that our framework is able to capture to model the user-preferences increases as the time interval increases. However, for the time interval of one $Month$, the performance drops. This could be due to the users evolving preferences over time or trying different things which they might not have liked and shouldn't be recommended. Moreover, with increasing time interval the model isn't able to appropriately model the user choices as the time advances. Hence, the performance deteriorates as the time interval increases. Based on the above discussion, we have chosen the time interval for our model to be $Two\ Weeks$, this gives us sufficient data to model the evolving user-preferences and location-popularity.

Table 1: The experimental results obtained by various baseline and contemporary techniques on the Gowalla dataset as evaluated for different evaluation metrics.

Methods	**AUC**	**Precision**	**Recall**	**MRR**	**F1 Score**
Logistic Regression	50.43	58.01	51.19	52.34	54.38
Multinomial Naive Bayes	51.34	60.32	50.89	56.45	55.20
Decision Trees	60.58	55.12	88.61	68.67	67.96
Random Forest	72.37	70.43	75.47	74.35	72.86
XGBoost	72.02	67.19	81.01	77.86	73.45
Deep Neural Network	77.77	79.6	73.91	75.56	76.64
STRNN	80.11	80.36	80.84	81.7	80.59
DeepMove	71.08	67.65	78.19	73.64	72.53
STGN	77.18	81.76	72.54	77.98	76.87
ARNN	77.0	79.52	72.14	76.77	75.65
LSTPM	70.12	69.3	74.1	72.72	71.61
TiSASRec	70.56	70.96	67.4	70.45	69.13
GeoSAN	76.76	76.31	76.78	77.91	76.54
Proposed Work	82.61	80.84	84.38	83.73	82.57

4.2. Gowalla Dataset

In this section, we discuss the results obtained for the Gowalla dataset. We evaluate the results over all the performance metrics and compare the performance with various baseline algorithms and contemporary techniques. The obtained results are given in Tab. 1, from which we can see that our proposed framework outperforms the other techniques in terms of AUC and MRR. In terms of Precision it is second to only STGN while for Recall it lags behind just Decision Trees. However, the difference between them is very minute and the performance of our framework is still good for precision as well as recall. This shows that our framework gives a good and stable performance across the evaluation metrics. The second best performer is STRNN which is also stable to the evaluation metrics and generates acceptable results as compared to others. Logistic regression and multinomial naive bayes are the worst performers with relatively poorer results to other techniques.

Table 2: The experimental results obtained by various baseline and contemporary techniques on the NYCRestaurant dataset as evaluated for different evaluation metrics.

Methods	AUC	Precision	Recall	MRR	F1 Score
Logistic Regression	51.8	69.7	66.78	62.37	68.20
Multinomial Naive Bayes	52.85	51.32	66.67	58.81	57.99
Decision Trees	59.97	70.28	36.39	48.86	47.95
Random Forest	64.59	67.5	57.27	62.47	61.97
XGBoost	72.54	70.76	80.42	76.39	75.28
Deep Neural Network	68.04	59.25	88.78	72.91	71.07
STRNN	70.49	67.05	87.02	76.89	75.74
DeepMove	67.82	72.63	59.23	66.37	65.25
STGN	61.8	62.37	70.55	67.83	66.21
ARNN	66.7	68.04	64.76	68.12	66.36
LSTPM	69.03	71.78	67.83	71.34	69.75
TiSASRec	71.38	71.6	72.67	69.21	72.13
GeoSAN	66.91	62.23	86.89	75.39	72.52
Proposed Work	73.42	81.27	61.45	73.38	69.98

4.3. NYCRestaurant Dataset

Tab. 2 present the results obtained by various techniques for the NYCRestaurants over all the evaluation metrics. For the NYCRestaurant also, the Logistic regression and the multinomial naive bayes models are the worst performers. However, XGBoost comes out to be the second best performer. Amongst the contemporary algorithms, TiSASRec is the best performer followed by STRNN. Our proposed framework comes out to be the best performer in terms of AUC and precision. It also achieves better results than Random Forest, XGBoost, and Deep Neural Network. This shows that our approach to ensemble them in a weighted manner based on their performance during the training phase is optimal. The improved results obtained by our framework can be attributed to the varying features captured by each of the candidate models individually. The appropriate ensembling technique also helps in achieving better results.

Table 3: The experimental results obtained by various baseline and contemporary techniques on the NYC dataset as evaluated for different evaluation metrics.

Methods	**AUC**	**Precision**	**Recall**	**MRR**	**F1 Score**
Logistic Regression	59.62	67.0	39.18	51.74	49.45
Multinomial Naive Bayes	63.3	67.06	52.9	62.02	59.14
Decision Trees	72.58	74.75	68.77	72.18	71.64
Random Forest	72.49	74.23	69.19	74.18	72.49
XGBoost	76.74	76.02	78.46	79.35	77.22
Deep Neural Network	77.91	76.24	81.59	80.64	78.82
STRNN	82.72	80.02	85.38	83.45	82.61
DeepMove	66.45	63.05	76.78	71.87	69.24
STGN	72.53	68.51	83.28	77.74	75.18
ARNN	71.91	71.18	73.47	74.85	72.3
LSTPM	75.33	71.83	80.79	82.05	76.05
TiSASRec	80.07	85.04	73.68	80.96	78.95
GeoSAN	84.36	85.06	82.76	84.47	83.89
Proposed Work	88.36	92.88	83.01	88.49	87.67

4.4. NYC Dataset

This section discusses the results obtained for the NYC dataset over all the evaluation metrics by various techniques. The obtained results are presented in Tab. 3. The results show that our method outperforms all the other baselines and contemporary techniques by a considerable margin except for recall. For recall, our framework is right behind STRNN and STGN, in that order. For the baslines the best performance is achieved by Deep Neural Network while the Logistic regression is the worst performer. For the contemporary next POI recommendation techniques, DeepMove performs the worst while GeoSAN performs the best. However, our proposed framework gives the best and the most stable results across all the evaluation metrics. The better results obtained by our framework can be attributed to the modified random walk based node2vec algorithm employed by us to generate the feature vectors. It helps our model to optimally capture the local and global structure of the user-location graph to generate the feature vectors. It appropriately leverages the network structure and topology to give improved results.

4.5. TKY Dataset

In this section, we discuss the results obtained for the TKY dataset by various techniques for the task of next POI recommendation. Tab. 4, tabulates the results obtained by all the algorithms for all the evaluation metrics. The results show that our proposed framework outperforms all the other techniques by a huge gap, except for recall. For recall, LSTPM is the best performer closely followed by XGBoost. The results obtained by the various baselines and the contemporary techniques are in close agreement with each other for the TKY dataset. The Logistic Regression, Multinomial Naive Bayes, and XGBoost are the worst performers. The Random Forest and the GeoSAN are the best

Table 4: The experimental results obtained by various baseline and contemporary techniques on the TKY dataset as evaluated for different evaluation metrics.

Methods	**AUC**	**Precision**	**Recall**	**MRR**	**F1 Score**
Logistic Regression	60.45	75.14	31.9	45.34	44.78
Multinomial Naive Bayes	66.25	71.08	54.09	55.76	61.43
Decision Trees	74.13	81.96	61.76	71.63	70.44
Random Forest	76.11	82.95	66.36	74.71	73.73
XGBoost	66.69	62.24	86.23	73.89	72.3
Deep Neural Network	70.09	66.21	81.05	73.69	72.88
STRNN	74.79	72.37	82.03	78.19	76.9
DeepMove	75.26	74.53	78.32	77.47	76.38
STGN	76.42	75.45	78.5	79.76	76.95
ARNN	77.62	76.12	80.81	77.64	78.4
LSTPM	82.69	80.82	86.31	84.76	83.48
TiSASRec	79.71	85.69	71.33	78.26	77.85
GeoSAN	83.7	87.99	77.57	83.97	82.45
Proposed Work	89.23	93.23	84.85	89.92	88.84

performers in their respective categories. However, they fall short to the performance of our model. The improved results obtained by our framework can be credited to the attention based feature processing which helps in assigning a suitable degree of importance to the contributions of the neighbors in the feature vectors of the target node. It also helps in aggregating the impact made by the neighbors of a node towards the features of the nodes.

4.6. Brightkite Dataset

In this section, we discuss the results obtained by the various techniques for the task of next POI recommendation for the Brightkite dataset. The results obtained are present in Tab. 5. For the Brightkite dataset, Decision trees is the worst performer followed by Logistic regression and Multinomial naive bayes. The Deep Neural Network is the best performer amongst the baseline algorithms. For the contemporary algorithms, ARNN is the worst performer while TiSASRec is the best performer. From Tab. 5, we see that our proposed framework outperforms all the other techniques in terms of AUC and MRR. For precision, it loses to Deep Neural Network and TiSASRec while for recall, it loses to DeepMove and STRNN, however the loss margin is very minute. The improved performance of our proposed framework can be attributed to the iterative learning employed by us during the training phase which helps our framework to model the evolving preferences more efficiently and effectively. It also helps our model to capture the changing choices of the users and the changing popularity of the locations. It also helps in inferring the user-location, user-user and location-location relationship.

Table 5: The experimental results obtained by various baseline and contemporary techniques on the Brightkite dataset as evaluated for different evaluation metrics.

Methods	AUC	Precision	Recall	MRR	F1 Score
Logistic Regression	59.69	68.33	33.88	47.24	45.3
Multinomial Naive Bayes	60.05	62.86	51.56	57.56	56.65
Decision Trees	58.33	51.08	64.55	58.65	57.03
Random Forest	64.48	72.22	49.62	60.83	58.82
XGBoost	64.4	71.6	46.77	57.44	56.58
Deep Neural Network	68.09	85.53	46.1	61.16	59.91
STRNN	75.45	72.26	86.15	77.89	78.6
DeepMove	81.41	78.52	85.48	82.54	81.85
STGN	80.15	80.47	80.47	79.14	80.47
ARNN	71.51	73.98	69.47	72.07	71.65
LSTPM	74.59	73.98	73.98	74.15	73.98
TiSASRec	83.32	82.79	82.79	83.40	82.79
GeoSAN	77.3	78.45	73.98	78.32	76.15
Proposed Work	84.5	82.05	84.21	85.73	83.12

4.7. Yelp Dataset

Tab. 6 presents the results obtained by the various techniques over all the evaluation metrics for the Yelp Dataset. The results show that our proposed framework is the best performer as compared to other techniques in terms of AUC and MRR. For precision, it loses to Decision trees and XGBoost while for recall it loses to STRNN and DeepMove. It also shows that the Logistic regression and Multinomial Naive Bayes are the worst performers amongst the baselines while LSTPM and ARNN are the worst performers amongst the contemporary techniques. The best results are obtained by our framework, followed by STRNN and DeepMove. The optimal results achieved by our framework can be attributed to the proper feature vector generation using our modified node2vec algorithm which helps it in accurately capturing the topological and structural details of the user-location graph. The attention based feature processing also helps our framework in aggregating the neighborhood information and estimate the contributions of the neighbors towards the features of the target node. The iterative training of our framework also helps it in modelling the changing opinions of the users regarding various locations. It also helps in uncovering the evolving associations and dissociations amongst the users and locations. Lastly, the appropriate weighted ensembling of the various machine learning and deep learning classifiers also helps our framework in achieving optimal results. The above discussion demonstrates the efficacy and efficiency of our proposed framework as an optimal next Point-Of-Interest recommendation system.

4.8. Confidence Interval Analysis

In Tab. 7, we present the statistical significance of the model performance based on the confidence interval on AUC values for all the datasets. The AUC value represents

Table 6: The experimental results obtained by various baseline and contemporary techniques on the Yelp dataset as evaluated for different evaluation metrics.

Methods	AUC	Precision	Recall	MRR	F1 Score
Logistic Regression	56.29	64.15	27.42	40.58	38.42
Multinomial Naive Bayes	60.9	62.13	52.03	58.62	56.64
Decision Trees	65.36	84.75	38.17	53.69	52.63
Random Forest	68.96	74.23	57.6	65.53	64.86
XGBoost	71.83	83.13	54.76	67.89	66.03
Deep Neural Network	69.19	61.88	83.9	72.18	71.22
STRNN	77.28	72.15	89.76	82.91	80.0
DeepMove	76.7	66.88	90.35	77.72	76.87
STGN	74.69	69.93	85.6	78.67	76.98
ARNN	70.54	70.07	74.42	74.46	72.18
LSTPM	64.51	68.14	58.78	65.05	63.11
TiSASRec	76.48	77.39	72.95	76.45	75.11
GeoSAN	72.17	72.27	69.92	72.64	71.07
Proposed Work	81.58	79.58	86.92	84.21	83.09

the Area Under the Curve, which is a common evaluation metric for recommendation systems. The confidence interval provides a range of values within which we can be confident that the true AUC value lies. For instance, the 90% confidence interval for the Gowalla dataset is [82.34, 82.88] while the 99% confidence interval for the same is [82.19, 83.03]. This shows that if we were to run our experiments again then we can say with 90% confidence that our AUC values will lie in [82.34, 82.88] while with 99% that our values will lie in [82.19, 83.03]. It is evident that the confidence interval is wider for 99% as compared to 90%, thereby highlighting that as the confidence increases the confidence interval widens. Looking at Tab. 7, it is observed that the obtained confi-

Table 7: Statistical significance of the model performance based on the confidence interval on AUC values for all the datasets.

Confidence	**Gowalla**	**NYCRestaurant**	**NYC**
90%	[82.34, 82.88]	[72.12, 74.72]	[86.75, 89.97]
95%	[82.29, 82.93]	[71.87, 74.97]	[86.43, 90.29]
98%	[82.23, 82.99]	[71.57, 75.27]	[86.07, 90.65]
99%	[82.19, 83.03]	[71.38, 75.46]	[85.82, 90.9]
Confidence	**TKY**	**Brightkite**	**Yelp**
90%	[88.16, 90.3]	[83.94, 85.06]	[81.22, 81.94]
95%	[87.95, 90.51]	[83.83, 85.17]	[81.15, 82.01]
98%	[87.71, 90.75]	[83.7, 85.3]	[81.07, 82.09]
99%	[87.54, 90.92]	[83.62, 85.38]	[81.01, 82.15]

dence intervals are very close to the AUC values achieved by the proposed modified node2vec and Attention-based fusion framework. This indicates that the performance of our model is mathematically justified and demonstrates its superiority over other algorithms.

5. Conclusion

Over the years, next Point-Of-Interest recommendation system has attracted huge attention from both the academics and industry, thereby becoming an important problem in the recommendation domain. It refers to the task of suggesting locations to users to visit based on their and other user's past preferences and visits. In this paper, we proposed a Modified node2vec and Attention based Ensemble Framework for Next POI Recommendation. We start by preprocessing the raw data to extract all the relevant information and group the dataset into fixed sized time intervals and generate a user-location graph for each time interval. Then we go on to generating the feature vectors for the nodes in the graph using the modified node2vec algorithm to suitably capture its structure and topology. The generated features are then processed using an attention component to improve the features to appropriately represent the impact of the neighbors of the target node on its features. Then we go onto creating well labelled and well balanced datasets for training and testing. The training dataset is used to iteratively train various machine learning and deep learning classifiers to capture the evolving preferences and popularity of the users and locations, respectively. The trained models are then ensembled in a weighted manner to generate improved results as compared to the individual models. We performed extensive experiments on the several real-life datasets and evaluated various popular performance metrics. We compared the performance of our proposed framework with several baseline algorithms and contemporary next POI recommendations techniques. The obtained results demonstrate that our proposed framework performs appropriately for the task of next POI recommendation. In future studies, there is scope for integrating the social connections and influence of the user's network into the proposed work to improve the recommendation process. Furthermore, the system could utilize real-time data from diverse sources like social media, news feeds, and sensor networks to offer timely and current recommendations.

References

[1] C. Yu, B. Xiao, D. Yao, X. Ding, H. Jin, Using check-in features to partition locations for individual users in location based social network, Information Fusion 37 (2017) 86–97. 2017.

[2] R. Gao, J. Li, X. Li, C. Song, Y. Zhou, A personalized point-of-interest recommendation model via fusion of geo-social information, Neurocomputing 273 (2018) 159–170. 2018.

[3] M. Chen, Y. Zuo, X. Jia, Y. Liu, X. Yu, K. Zheng, Cem: A convolutional embedding model for predicting next locations, IEEE Transactions on Intelligent Transportation Systems 22 (2020) 3349–3358. 2020.

[4] M. Chen, Y. Liu, X. Yu, Nlpmm: A next location predictor with markov modeling, in: Pacific-Asia conference on knowledge discovery and data mining, Springer, pp. 186–197. 2014.

[5] H. Wang, H. Shen, W. Ouyang, X. Cheng, Exploiting poi-specific geographical influence for point-of-interest recommendation., in: IJCAI, pp. 3877–3883. 2018.

[6] R. Baral, S. S. Iyengar, X. Zhu, T. Li, P. Sniatala, Hirecs: A hierarchical contextual location recommendation system, IEEE Transactions on Computational Social Systems 6 (2019) 1020–1037. 2019.

[7] C. Yang, L. Bai, C. Zhang, Q. Yuan, J. Han, Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1245–1254. 2017.

[8] W. Wang, J. Chen, J. Wang, J. Chen, Z. Gong, Geography-aware inductive matrix completion for personalized point-of-interest recommendation in smart cities, IEEE Internet of Things Journal 7 (2019) 4361–4370. 2019.

[9] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, S. S. Iyengar, A survey on deep learning: Algorithms, techniques, and applications, ACM Computing Surveys (CSUR) 51 (2018) 1–36. 2018.

[10] Y. Xu, X. Li, J. Li, C. Wang, R. Gao, Y. Yu, Ssser: Spatiotemporal sequential and social embedding rank for successive point-of-interest recommendation, IEEE Access 7 (2019) 156804–156823. 2019.

[11] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, D. Jin, Deepmove: Predicting human mobility with attentional recurrent networks, in: Proceedings of the 2018 world wide web conference, pp. 1459–1468. 2018.

[12] K. Kala, M. Nandhini, Context-category specific sequence aware point-of-interest recommender system with multi-gated recurrent unit, Journal of Ambient Intelligence and Humanized Computing (2019) 1–11. 2019.

[13] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen, H. Yin, Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pp. 214–221. 2020.

[14] R. Li, Y. Shen, Y. Zhu, Next point-of-interest recommendation with temporal and multi-level context attention, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, pp. 1110–1115. 2018.

[15] Q. Liu, S. Wu, L. Wang, T. Tan, Predicting the next location: A recurrent model with spatial and temporal contexts, in: Thirtieth AAAI conference on artificial intelligence. 2016.

[16] Q. Wang, H. Yin, T. Chen, Z. Huang, H. Wang, Y. Zhao, N. Q. Viet Hung, Next point-of-interest recommendation on resource-constrained mobile devices, in: Proceedings of the Web conference 2020, pp. 906–916. 2020.

[17] Z. Yu, H. Xu, Z. Yang, B. Guo, Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints, IEEE Transactions on Human-Machine Systems 46 (2015) 151–158. 2015.

[18] V. W. Zheng, Y. Zheng, X. Xie, Q. Yang, Collaborative location and activity recommendations with gps history data, in: Proceedings of the 19th international conference on World wide web, pp. 1029–1038. 2010.

[19] M. Chen, W. Li, L. Qian, S. Lu, D. Chen, Interest-aware next poi recommendation for mobile social networks, in: International Conference on Wireless Algorithms, Systems, and Applications, Springer, pp. 27–39. 2018.

[20] D. Yang, B. Fankhauser, P. Rosso, P. Cudre-Mauroux, Location prediction over sparse user mobility traces using rnns: Flashback in hidden states!, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pp. 2184–2190. 2020.

[21] D. Kong, F. Wu, Hst-lstm: A hierarchical spatial-temporal long-short term memory network for location prediction., in: IJCAI, volume 18, pp. 2341–2347. 2018.

[22] T. Liu, J. Liao, Z. Wu, Y. Wang, J. Wang, A geographical-temporal awareness hierarchical attention network for next point-of-interest recommendation, in: Proceedings of the 2019 on international conference on multimedia retrieval, pp. 7–15. 2019.

[23] T. Liu, J. Liao, Z. Wu, Y. Wang, J. Wang, Exploiting geographical-temporal awareness attention for next point-of-interest recommendation, Neurocomputing 400 (2020) 227–237. 2020.

[24] C. H. Liu, Y. Wang, C. Piao, Z. Dai, Y. Yuan, G. Wang, D. Wu, Time-aware location prediction by convolutional area-of-interest modeling and memory-augmented attentive lstm, IEEE Transactions on Knowledge and Data Engineering (2020). 2020.

[25] P. Zhao, A. Luo, Y. Liu, F. Zhuang, J. Xu, Z. Li, V. S. Sheng, X. Zhou, Where to go next: A spatio-temporal gated network for next poi recommendation, IEEE Transactions on Knowledge and Data Engineering (2020). 2020.

[26] J. Manotumruksa, C. Macdonald, I. Ounis, A contextual attention recurrent architecture for context-aware venue recommendation, in: The 41st international ACM SIGIR conference on research & development in information retrieval, pp. 555–564. 2018.

[27] S. Zhao, T. Zhao, I. King, M. R. Lyu, Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation, in: Proceedings of the 26th international conference on world wide web companion, pp. 153–162. 2017.

[28] B. Chang, Y. Park, D. Park, S. Kim, J. Kang, Content-aware hierarchical point-of-interest embedding model for successive poi recommendation., in: IJCAI, pp. 3301–3307. 2018.

[29] D. Xi, F. Zhuang, Y. Liu, J. Gu, H. Xiong, Q. He, Modelling of bi-directional spatio-temporal dependence and users' dynamic preferences for missing poi check-in identification, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pp. 5458–5465. 2019.

[30] Q. Gao, W. Wang, L. Huang, X. Yang, T. Li, H. Fujita, Dual-grained human mobility learning for location-aware trip recommendation with spatial–temporal graph knowledge fusion, Information Fusion (2022). 2022.

[31] F. T. Sáenz, F. Arcas-Tunez, A. Muñoz, Nation-wide touristic flow prediction with graph neural networks and heterogeneous open data, Information Fusion (2022). 2022.

[32] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855–864. 2016.

[33] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014). 2014.

[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint arXiv:1710.10903 (2017). 2017.

[35] E. Cho, S. A. Myers, 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, pp. 1082–1090. 2011.

[36] D. Yang, D. Zhang, Z. Yu, Z. Yu, Fine-grained preference-aware location search leveraging crowdsourced digital footprints from lbsns, in: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, pp. 479–488. 2013.

[37] D. Yang, D. Zhang, V. W. Zheng, Z. Yu, Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns, IEEE Transactions on Systems, Man, and Cybernetics: Systems 45 (2014) 129–142. 2014.

[38] E. Cho, S. A. Myers, 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, pp. 1082–1090. 2011.

[39] Y. Liu, T.-A. N. Pham, G. Cong, Q. Yuan, An experimental evaluation of point-of-interest recommendation in location-based social networks, Proceedings of the VLDB Endowment 10 (2017) 1010–1021. 2017.

[40] Q. Liu, S. Wu, L. Wang, T. Tan, Predicting the next location: A recurrent model with spatial and temporal contexts, in: Thirtieth AAAI conference on artificial intelligence. 2016.

[41] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, D. Jin, Deepmove: Predicting human mobility with attentional recurrent networks, in: Proceedings of the 2018 world wide web conference, pp. 1459–1468. 2018.

[42] P. Zhao, A. Luo, Y. Liu, F. Zhuang, J. Xu, Z. Li, V. S. Sheng, X. Zhou, Where to go next: A spatio-temporal gated network for next poi recommendation, IEEE Transactions on Knowledge and Data Engineering (2020). 2020.

[43] Q. Guo, Z. Sun, J. Zhang, Y.-L. Theng, An attentional recurrent neural network for personalized next location recommendation, in: Proceedings of the AAAI Conference on artificial intelligence, volume 34, pp. 83–90. 2020.

[44] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen, H. Yin, Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pp. 214–221. 2020.

[45] J. Li, Y. Wang, J. McAuley, Time interval aware self-attention for sequential recommendation, in: Proceedings of the 13th international conference on web search and data mining, pp. 322–330. 2020.

[46] D. Lian, Y. Wu, Y. Ge, X. Xie, E. Chen, Geography-aware sequential location recommendation, in: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 2009–2019. 2020.