Journal of Information Systems and Telecommunication
Vol.11, No.1, January-March 2023, 12-23
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

# Developing A Contextual Combinational Approach for Predictive Analysis of Users Mobile Phone Trajectory Data in LBSNs

Fatemeh Ghanaati[1], Gholamhossein Ekbatanifard[2*], Kamrad Khoshhal Roudposhti[2]

[1]. Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran
[2]. Department of Computer Engineering, Lahijan branch, Islamic Azad University, Lahijan, Iran

Received: 09 Dec 2021/ Revised: 12 Feb 2022/ Accepted: 22 March 2022

## Abstract

Today, smartphones, due to their ubiquity, have become indispensable in human daily life. Progress in the technology of mobile phones has recently resulted in the emergence of several popular services such as location-based social networks (LBSNs) and predicting the next Point of Interest (POI), which is an important task in these services. The gathered trajectory data in LBSNs include various contextual information such as geographical and temporal contextual information (GTCI) that play a crucial role in the next POI recommendations. Various methods, including collaborating filtering (CF) and recurrent neural networks, incorporated the contextual information of the user' trajectory data to predict the next POIs. CF methods do not consider the effect of sequential data on modeling, while the next POI prediction problem is inherently a time sequence problem. Although recurrent models have been proposed for sequential data modeling, they have limitations such as similarly considering the effect of contextual information. Nonetheless, they have a separate impact as well. In the current study, a geographical temporal contextual information-extended attention gated recurrent unit (GTCI-EAGRU) architecture was proposed to separately consider the influence of geographical and temporal contextual information on the next POI recommendations. In this research, the GRU model was developed using three separate attention gates to consider the contextual information of the user trajectory data in the recurrent layer GTCI-EAGRU architecture, including timestamp, geographical, and temporal contextual attention gates. Inspired by the assumption of the matrix factorization method in CF approaches, a ranked list of POI recommendations was provided for each user. Moreover, a comprehensive evaluation was conducted by utilizing large-scale real-world datasets based on three LBSNs, including Gowalla, Brightkite, and Foursquare. The results revealed that the performance of GTCI-EAGRU was higher than that of competitive baseline methods in terms of Acc@10, on average, by 42.11% in three datasets.

**Keywords:** LBSN; Trajectory Data; Contextual Information; GRU.

## 1- Introduction

Nowadays, people widely use location-based social networks (LBSNs) and enjoy location-based services (LBSs) using their mobile devices for sharing their locations with others by making check-ins at locations or points of interests (POIs) that they have visited, including shops, museums, and restaurants [1]. The massive record of users' check-in data provides a chance to conduct research on people's mobility behaviors, in particular, for POI recommendation systems [2,3]. In addition, governments can use predictions about people's future destinations and develop better transportation and scheduling strategies for alleviating traffic jams and handling crowd congestions [5,6,7,8]. Some geographical and temporal information exists in a user's historical check-in sequence [4,9], having different effects on recommending the next POI. In this study, it was attempted to separately consider this contextual information to better train the proposed model. Human mobility is extremely complex and diverse; therefore, many previous studies were unable to simply determine the offering of the next POI recommendation [4,6]. Matrix factorization (MF) and other collaborative filtering (CF) techniques have widespread use for recommending a list of personally ranked POIs to the users [2]. Typically, approaches to MF include contextual information about the user. This helps provide valuable recommendations to users who lack enough historical check-ins and is generally referred to as the cold-start problem. However, the employment of collaboration filtering (CF)-based methods complicates the processing of

✉ **Gholamhossein Ekbatanifard**
ekbatanifard@liau.ac.ir

sequence data and capturing of dynamic user's preferences [2,6,11]. As a result, the ongoing challenges lie in the manner of integrating the information of different features to accurately model users' complex behavioral preferences and then recommending reliable POIs [13].

Recurrent neural networks (RNNs) have recently been successfully applied to sequential recommender systems [1,4,8,15]. Thus, long-term dependencies can be captured by the hidden states of recurrent methods [4,16]. Many types of recurrent-based approaches have considered geographical and temporal factors to enhance the performance of POI recommendation algorithms [2,4,11,12,15]. Nonetheless, the present RNN-based POI recommendation methods face the alleviation of the cold-start problem [11]. In this regard, one of the excellent choices is to incorporate RNN-based POI recommendation methods with the MF method to enjoy the benefits of each one [2]. The user's historical check-in behaviors do not significantly pose any problems in predicting the next behavior; hence, it is necessary to take only the important information into serious consideration [1,11]. Therefore, the attention mechanism (AM) has been proposed to deal with this challenge. The AM can enhance the capability of the neural network in capturing long-term dependencies and boost the ability to interpret neural networks [18]. In this study, the idea of the AM was used to address the most important contextual information.

## 1-1- Motivations

This study focused on the next POI recommendation through modeling check-in sequences and considering geographical and temporal contextual influences separately and proposed a novel geographical temporal contextual information extended attention gated recurrent unit (GTCI-EAGRU) for the next POI recommendation. Among the recurrent models, the GRU model is highly simple and does not include many parameters in contrast to the long-short term memory (LSTM) model. In addition, this model can ignore the earlier unit hidden state, which is impossible with the traditional RNN [4,6]. Thus, a GRU network was developed to model check-in sequences while paying attention to geographical distances and time intervals between two successive check-ins [19]. It is noteworthy that any piece of contextual information needs individual consideration during modeling since the effects of contextual information on user behavior are different [2,3]. Further, the GRU network was upgraded by inspiration from the AM to consider more important contextual information.

Furthermore, factorization approaches were employed, and the preference score was computed by the dot product. Following the prediction scores, it is possible to recommend top-k POIs to a user, and there is a high chance that the user will go there if the score is higher. The Bayesian personalized ranking (BPR) framework [20] learned the parameters of GTCI-EAGRU. In the last stage, three general datasets were utilized to conduct extensive experiments. Five up-to-date POI recommendation methods were compared with Brightkite, Gowalla, and Foursquare to evaluate the model.

## 1-2- Main Contributions

1- The proposed architecture is presented by combining the development of the GRU model with the MF method, which aims to apply the strengths of the models and reduce the challenges of each of these methods. According to the MF method, in the CF approach, places visited on social networks by a user on social networks can affect the next POI of other users on those networks. However, CF-based approaches are weak in modeling sequential data and do not consider the effect of sequential data on modeling, while the next POI prediction problem is inherently a time sequence problem. Although recurrent models have been proposed for sequential data modeling, they have limitations. The traditional RNN model cannot integrate the corresponding check-in contextual information into the modeling. Newer recurrent models also consider the effect of temporal and spatial contextual information similarly, while they have a separate effect.

Therefore, there is a need to develop these models. In the recurrent layer of the proposed architecture, a development of the GRU model is presented using three attention gates that consider the contextual information separately and in terms of their importance.

2- Within the recurrent layer of the proposed architecture, the flexibility of the GRU model is employed, and the GRU model was expanded following the attention-based approach. Moreover, three additional attention gates were proposed, including timestamp contextual attention gate (Gts), geographical contextual attention gate (Gge), and temporal contextual attention gate (Gte). The Gts controls the influence of timestamp earlier visited locations, whereas Gge and Gte control the effect of the hidden state of the earlier recurrent unit based on geographical distances and time intervals between two successive check-ins, respectively. This innovation makes it possible to extend the model to another context.

3- In this research, user contextual information is classified into two categories of absolute and transitional content information. The first category includes check-in timestamp and geographical coordinates and the second one consists of the time interval and geographical distance between two successive consecutive check-ins. Our proposed architecture considers two types of absolute and transitional contextual information separately. This category focuses on developing a model to consider more contextual information in the future.

4. Some comprehensive experiments were conducted on three large-scale real-world datasets, namely, Brightkite, Gowalla [21], and Foursquare [14] that are widely used in related studies to predict the user POI in LBSNs. The aim was to show the effectiveness of the proposed GTCI-EAGRU architecture for the next POI recommendation.

## 1-3- Problem Statement

Human mobility prediction is important for a wide spectrum of LBSN applications, and the next POI recommendation is one of the usages of predicting people's mobility [1]. In some LSBNs, users share their location by registering check-ins. The check-ins gathered in LBSNs contain geographical and temporal contextual information (TCI), and each piece of information has a separate effect on predicting the user's next location [3, 8]. In previous studies, some restrictions were applied for dividing sequence into different check-in trajectories such as using the time interval of less than six hours [1]. Nonetheless, applying restrictions for the time interval and geographical distance, when considering registered check-ins in data preprocessing, is not a proper approach for the mentioned purpose. The AM can address the mentioned issue. Instead of using multiple assumptions to consider the time interval or geographical distance constraints between two check-ins, it can be addressed by automatic weighting given to the model inputs inspired by the AM.
According to evidence [2], CF-based approaches have weaknesses in sequential data modeling and fail to consider the effect of sequential data, while the problem of the next location prediction is inherently a matter of time sequence (Challenge 1). Traditional recurrent models are unable to consider contextual information, but this information is highly important in determining the next POI (Challenge 2). Meanwhile, some earlier studies, based on recurrent models, consider the effect of temporal and geographical contextual information (GCI) to be the same, while they have a different effect (Challenge 3). Furthermore, according to [11], some proposed architectures, which are a combination of recurrent models and AM, are highly complex (Challenge 4). In this work, the GTCI-EAGRU model was proposed to address the above-mentioned challenges.

## 1-4- Organizations

The remaining parts of this research are as follows: The related methods are briefly reviewed in Section 2. Sections 3 and 4 describe some preliminaries to the study and the details of the GTCI-EAGRU network, respectively. In Section 5, an illustration of the experiments is presented, followed by providing the results of the proposed method. Finally, Section 6 summarizes conclusions and an outline for future works.

# 2- Related Works

This section classifies related studies under three approaches generally used for the next POI recommendations, including CF, RNN, and AM. Table 1 provides a summary of related works with their challenges considered in our research.

Table 1. summarize of related works

| Model Name | Model Approach | Method summery | challenges |
|---|---|---|---|
| [28] Unified method | CF based | Believing that time plays an important role in POI recommendations and defining a new problem, namely, the time-aware POI recommendation to recommend POIs for a given user at a specified time in a day | Focusing on temporal contextual information and paying less attention to geographic contextual information |
| [30] LORI | CF based | Applying a confidence coefficient for each user in the integration process and designing a learning-to-rank based algorithm to train confidence coefficients | Not taking into consideration time interval and geographical distance |
| [33] ST-RNN | RNN based | Extending RNN and using a transition matrix for capturing the temporal cyclic effect and geographical influence | Vanishing gradient problem in long sequence due to the use of the traditional RNN |
| [22] STGN | RNN based | Modifying the basic LSTM model slightly by introducing gates and cells to capture short- and long-term preferences | Considering the same effect for temporal and geographical contextual information |
| [8] SERM | RNN based | Jointly learning the embedding of multiple factors (user, location, time, and keywords) and the transition parameters of an RNN in a unified framework | Not taking into account the geographical distance in the training of this model |
| [35] CA-RNN | RNN based | Employing adaptive context-specific input matrices and adaptive context-specific transition matrices | Using a traditional RNN model and restrictions on paying attention to the contextual information, low performance |
| [1] ATST-LSTM | AM and RNN based | Developing an attention-based spatiotemporal LSTM network to focus on the relevant historical check-in records in a check-in sequence selectively using the spatiotemporal contextual information | Encountering with high complexity of implementation and a lack of attention to the scarcity |
| [6] Deep Move | AM and RNN based | Capturing complex dependencies and multi-level periodicity nature of humans using embedding, GRU, and AM | Not taking into account the time interval between two checks to model the behavioral pattern of user check-ins |
| [11] DAN-SNR | AM based | Makes use of the self-AM. By leveraging multi-head self-attention, the DAN-SNR can model long-range dependencies between any two historical check-ins efficiently and weigh their contributions to the next destination adaptively | Using only the attention mechanism and had low performance rather than applying recurrent neural networks for modeling the sequential influence and social influence |

*Note*. CF: Collaborating filtering; RNN: Recurrent neural network; AM: Attention mechanism; LORI: Learning-to-rank-based integration; ST-RNN: Spatiotemporal-Recurrent neural network; STGN: Spatio-temporal gated network; SERM: Semantics-enriched recurrent model; CA-RNN: Context-Aware Recurrent Neural Networks; ATST-LSTM: Attention-based Spatiotemporal-Long short term memory; DAN-SNR: Deep attentive network for social-aware recommendation.

## 3- Preliminaries

The research problem is formulated, and the applied preliminaries in this study are presented in the following section.

### 3-1- Notations and Definitions

Table 2 presents some primary notations used in this study.
Definition 1 (Check-in): A check-in is an action that a user takes under a geographical and temporal context. In addition, it is a registration of a location in the LBSN that contains geographical and temporal information. When a user u checks in a location l (including latitude and longitude) with venue-Id v at the timestamp t, the check-in record can be modeled as a quadruple: cu, v, t < u, l, v, t >.
Definition 2 (Check-in sequence): A user's check-in sequence or $S^u$ is a set of all user check-ins.
Definition 3 (Trajectory): Given a user u, a trajectory t is a sequence of chronologically ordered check-in associated with u. For example $tr_u$: $< u, l_1, v_1, t_1 >,…,< u, l_i, v_i, t_i >,…,< u, l_k, v_k, t_k >$, where $tr_u$ is the trajectory of a user u before time tk. Here, a trajectory set $Tr^{(u)}$ is used to denote all the trajectories of user u.
Definition 4 (POI): In LBSNs, a POI is a spatial item related to a geographical location and known as a venue, including a hotel or an office. In this research, POI is represented by *v*, and the set of POIs is demonstrated as $V=\{v_1, v_2,...\}$. Each POI v has a unique identifier and geographical coordinate, consisting of geographical latitude and geographical longitude.
Definition 5 (the next POI recommendation): Given all users' trajectories, the aim of the next POI recommendation is to predict the most likely location $vk$ that a user $u$ will visit at a certain time point $t_{N+1}$.
Definition 6 (POI recommendations): Given a set of users' check-in sequences $S^u$ and a set of POIs $V$, the POI recommendation task is to recommend top-k POIs that are preferable for user $u$.

Table 2: Notations and descriptions used in this study

| Notations | Descriptions |
|---|---|
| *u, l, v, & t* | User, location (including latitude and longitude), venue or POI, and timestamp |
| $c_{u, v, \& t}$ | A check-in recorded by user *u* in POI *v* and timestamp *t* |
| *lat v & lng v* | Latitude and longitude of POI *v* (i.e., geographical coordinates of POI *v*) |
| $\Delta t$ & $\Delta g$ | Time interval and geographical distance between two successive check-ins |
| $S^u$ | A set of all check-ins generated by user *u* |
| *Us, V, & T* | Sets of users, POIs, and timestamp |
| $v_\tau^u$ | POI visited by user *u* at time step $\tau$ |
| $g_\tau^u$ & $t_\tau^u$ | Vector representations of geographical and temporal intervals |
| $tr_u$ | A sequence of chronologically ordered check-ins related to *u* |
| $Tr^{(u)}$ | All trajectories from user *u* |
| $\phi u$ | The latent factor of user *u* |
| $\phi v$ | The latent factor of POI *v* |
| $\phi t$ | The latent factor of timestamp t |
| *d* | The number of latent dimensions |
| $v^+$ | A set of positive POIs (visited venues) for each user $u \in Us$ |
| $v^-$ | A set of positive negative POIs (unvisited venues) for each user $u \in Us$ |
| $\sigma$ | Sigmoid function |

### 3-2- MF in CF Based Approach

CF-based methods aim to discover similarities in the user's previous behavior and make predictions to the user based on a similar preference with other users [25]. There are various model-based CF algorithms, but MF is the most commonly applied in recommender systems [2]. MF seems to be the most accurate approach for lowering the problem from high levels of scarcity in the recommender systems database. Generally, MF models map both users and items to a joint latent factor space of dimensionality d in such a way that user-item interactions are modeled as inner products in that space. In the next POI recommendation, the item is the same POI or venue that a user has selected at the time of the check-in.
Accordingly, each venue v is related to a vector $q_v \in R^d$, and each user u is associated with a vector $p_u \in R^d$. For a given venue v, the elements of qv measure the extent to which the venue possesses those factors, positive or negative. For a given user u, the elements of $p_u$ measure the extent of interest the user has in venues that are high on the corresponding factors, positive or negative. The resulting dot product ($q_v^T p_u$) captures the interaction between user u and venue v- the user u's overall interest in the venue's characteristics. This approximates the user's rating of venue v, which is denoted by $r_{uv}$, leading to the following estimate [36]:

$$\hat{r}_{uv} = p_u q_v^T \quad (1)$$

The objective is to minimize the prediction error or the loss function in Eq. (2) where K is the set of (u,v) pairs of known ratings [25, 36].

$$min \sum_{(u,v)\in k} (r_{uv} - p_u\ q_v^T) \quad (2)$$

Different approaches exist [13,36] for the extension of MF using RNN models to capture the user's dynamic preferences from the sequence of user's check-ins. Specifically, with respect to the sequence of a user's check-ins, the output of an RNN model can be effective in representing a user's dynamic preferences and modifying MF-based approaches.

## 3-3- GRU in RNN Based Approach

The next POI recommendation is immediately faced with the challenge of learning personalized user preferences for POIs and the sequential correlations jointly and efficiently between the check-ins [1]. To solve this problem, the RNN takes a sequence of inputs and learns the sequential pattern of the input sequence using hidden states [2,3,8]. The problem that the RNN faces is the exploding and vanishing gradients; therefore, it cannot capture long-term preferences [2,11]. The problems can be solved by long-short term memory (LSTM), which employs a gate mechanism and can capture long-term preferences [1,23].

The information flow among consecutive LSTM cells is controlled through input, forget, and output gates. LSTM resolves the problems of the RNN, but it has three gates thus the training of an LSTM-based model is slower and requires a large amount of training data. GRU [4,6,23,40] has updated and reset gates in the network, dealing with the update degree of each hidden state. In fact, it determines which information should pass to the next state [2,3]. Fig. 1 displays the block diagram of basic GRU. As shown, GRU uses only two gates (i.e., reset and update gates). The GRU-based model can be trained faster and perform better compared to LSTM when there are less training data. GRU calculates hidden state $h_\tau$ at time $\tau$ from the output of update gate $z_\tau$, reset gate $r_\tau$, current input $x_\tau$, and previous hidden state $h_{\tau-1}$. $\hat{h}_\tau$ and $h_\tau$ are computed from the reset gate as follows:

$$z_\tau = \sigma\,(W_z\, x_\tau + U_z\, h_{\tau-1} + b_z) \tag{3}$$

$$r_\tau = \sigma\,(W_r\, x_\tau + U_r\, h_{\tau-1} + b_r) \tag{4}$$

$$\hat{h}_\tau = tanh\,(W\, x_\tau + U\,(r_\tau \odot h_{\tau-1}) + b_h) \tag{5}$$

$$h_\tau = (1 - z_\tau) \odot h_{\tau-1} + z_\tau \odot \hat{h}_\tau \tag{6}$$

where $\odot$ is a basic multiplication operation, and W and U represent weight matrices for training the network.

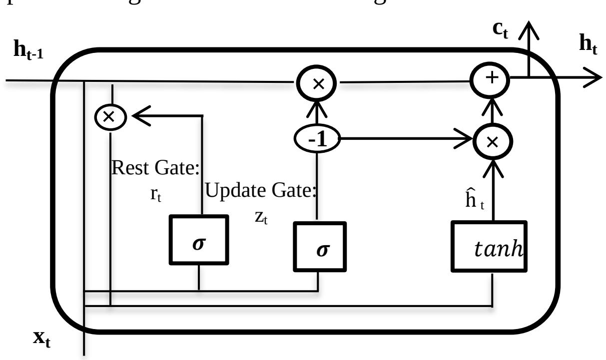


Fig. 1. An illustration of the GRU cell including two gates.[3]

## 3-4- Attention Mechanism

The AM was proposed based on the selective AM in the human visual system [1,18]. It should be noted that humans are prone to giving higher attention to key parts of the input, helping in breaking down a complex input into simpler parts that can easily be processed accordingly. Rather than paying attention to all available information, selective attention mainly focuses on the most relevant information in a system. Accordingly, learning to pay attention to the specific components of the input data resulted in different attention models in deep learning [4,11].

The present study proposed a novel model that is applied to this mechanism for the next location prediction. The key idea in the AM is that inputs are mapped to query, key, and value vectors. The outputs are calculated by taking the weighted sum of the value vectors where weights are determined by a function of query and key values [11]. Specifically, the attention function presents a query and a group of key-value pairs to a context vector, which is a weighted sum of all values. The queries, keys, and values are merged as $Q$, $K$, and $Vval$ matrices, respectively [1,18]. For the output of the attention function, an alignment function or the compatibility function, which measures the quality of the match between the input query matches and the corresponding key, calculates the weight assigned to each value. Eq. (7) is used for the computation of the matrix of outputs where $(Q, K)$ refers to the attention function [18]:

$$Attention\ (Q, K, V_{val}) = Softmax\ (f(Q,K))V_{val} \tag{7}$$

Additive attention and dot-product (multiplicative) attention are two of the most commonly used attention functions and are defined as follows [1]:

$$f_{add}(Q,k) = tanh(w_Q + W_K\, k) \tag{8}$$

$$f_{mul}(Q,k) = Q\, k^T \tag{9}$$

In theory, these two functions are similar in computation complexity. Additive attention and dot-product attention use a feed-forward neural network with a single hidden layer for the calculation and optimized matrix multiplication operation, respectively [1]. The present project, which is inspired by previous studies [2,3], employed a feed-forward neural network to calculate the alignment function to develop the GRU model.

# 4- Proposed GTCI-EAGRU Model Description

The GTCI-EAGRU architecture consists of input, embedding, recurrent, and output layers. Fig. 2 presents a schematic of our purpose architecture. The details of these layers and the learning procedure for the parameters are provided as follows:

## 4-1- Input Layer

The input layer contains model inputs that include absolute context and relative or transition context. In the proposed model, the absolute context is user id, timestamp, geographical coordinates (including latitude and longitude), and venue id. Further, the relative context (also called the transition context) is the time interval (Δt) and geographical distance (Δg) between two successive check-ins.

The geographical distance and time intervals are calculated in the input layer. For a given user u, venue $v_n$, and time $t^\tau$, the geographical distance ($\Delta g^\tau$) and time interval ($\Delta t^\tau$) between the POIs at current time $t^\tau$ and previous time $t^{\tau-1}$, as well as the given venue $v_n$ and venue $v_{n-1}$ previously visited at time are computed as: $\Delta t^\tau = t^\tau - t^{\tau-1}$ and $\Delta g^\tau$ = dist (lat $v_1$, lng $v_1$, lat $v_2$, lng $v_2$), respectively, where dist () is the Haversine and its function is as Eq.(10)[1]. It should be noted that the Haversine distance is the angular distance between two points on the surface of a sphere. The former coordinate of each point is taken as the latitude and the latter one is the longitude given in radians. The data dimension must be two[2].

$$D(x,y)=2\arcsin\left[\sqrt{\sin^2\left(\frac{x_1-y_1}{2}\right)+\cos(x_1)\cos(y_1)\sin^2\left(\frac{x_2-y_2}{2}\right)}\right] \quad (10)$$

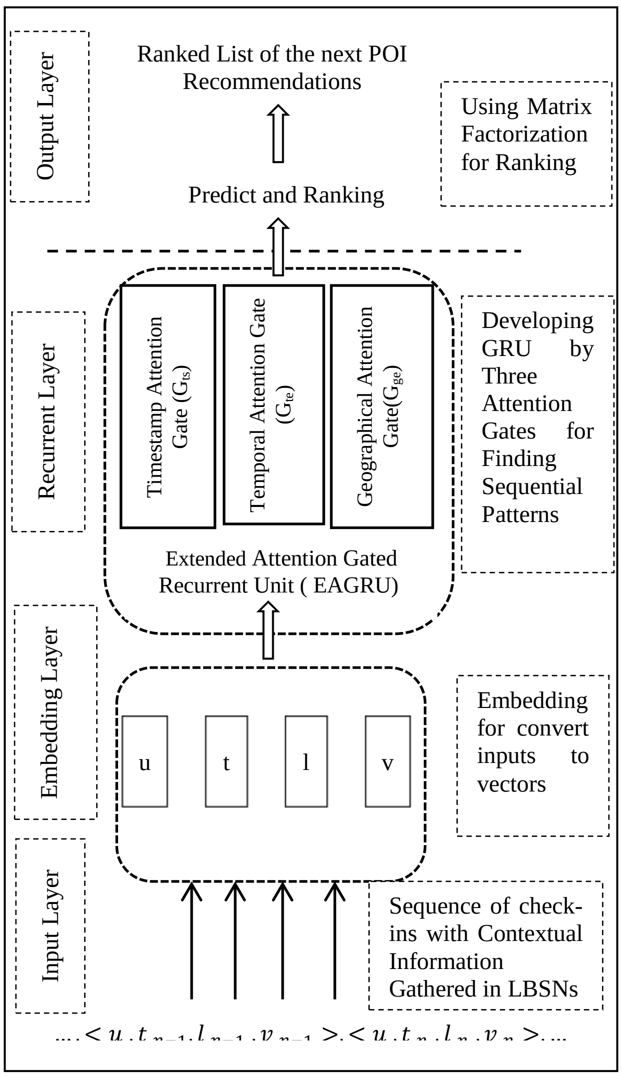


Fig. 2. GTCI- EAGRU architecture with input, embedding, recurrent, and output layers

*Note*. GTCI- EAGRU: Geographical temporal contextual information Extended attention gated recurrent unit; LBSN: Location-based social network. In the recurrent layer, the GRU model is extended with three additional attention gates, and a ranked list of the next POI recommendations is provided in the output layer.

[1] https://scikit-learn.org.

[2] In general, the Eq. (10) is used to calculate the Haversine distance between samples in X and Y ($x_1$ and $x_2$ are latitude and longitude of X and $y_1$ and $y_2$ are latitude and longitude of Y, respectively).

## 4-2- Embedding Layer

This layer is for embedding inputs from the check-in sequence before it goes to the recurrent layer. In this layer, embedding or latent factors are generated from the inputs. In addition, the latent factors of the user, namely, POI (or venue) and time are generated as $\emptyset u_i \in U$, $\emptyset v^\tau_i \in V$ and time $\emptyset t^\tau \in$ T, respectively. Note that θe = {U, V, T} denotes the set of the parameters of the embedding layer. Next, the latent factors of venue $\emptyset v^\tau_j$, the latent factors of the given time $\emptyset^\tau_t$, and the contextual transition features ($\Delta g_\tau$ and $\Delta t_\tau$) are passed to the recurrent layer for training using GTCI-EAGRU.

## 4-3- Recurrent Layer

In this layer, the GRU model was developed with three attention gates. Following Manotumruksa et al. [2] and Kala et al. [3], this study presented timestamp attention gate ($G_{ts}$), geographical attention gate ($G_{ge}$), and temporal attention gate ($G_{te}$). The input of $G_{ts}$ is the check-in time (i.e., the time that the check-in is registered by a user in LBSN and includes the year, month, day, hour, minute, and second). This gate is used to specify a more important timestamp in the sequence of historical check-ins of a user. However, the input of $G_{te}$ is the time interval (Δt) between two successive check-ins used to specify more important time intervals in the sequence of historical check-ins of a user. The input of $G_{ge}$ is the geographical distance (Δg) between two successive check-ins applied to specify a more important geographical distance in the sequence of historical check-ins of a user. The output of this layer is the hidden state of the recurrent unit at time step $\tau$, $h_\tau$, and is defined as Eq. (11):

$$h_\tau = f(\emptyset v^\tau_j, \emptyset t^\tau, \Delta t^\tau, \Delta g^\tau; \theta_r) \quad (11)$$

As mentioned earlier, the purposed model treats the absolute and relevant (or) transition contextual information separately. It is noteworthy that this contextual information has a different effect on the user's dynamic preference and requires independent consideration. The following part describes the extension of the traditional GRU for the integration of absolute and relevant contextual information.

Generally, in the GRU model, given the user's sequence of check-ins $S^u$ and dynamic preference at time step τ, the hidden state ($h^\tau$) is estimated by the update and reset gates, which are defined as:

$$z_\tau = \sigma(W_z\, \emptyset v_j^\tau + U_z h_{\tau-1} + b_z) \quad (12)$$

$$r_\tau = \sigma(W_r\, \emptyset v_j^\tau + U_r h_{\tau-1} + b_r) \quad (13)$$

$$\hat{h}_\tau = tanh(W_h\, \emptyset v_j^\tau + U_h\, (r_\tau \odot h_{\tau-1}) + b_h) \quad (14)$$

$$h_\tau = (1 - z_\tau) \odot h_{\tau-1} + z_\tau \odot \hat{h}_\tau \quad (15)$$

where $\emptyset v_j^\tau$ represents the latent factor of venue j that user i visited at time step $\tau$.

$\sigma()$ and $tanh()$ are the sigmoid and hyperbolic tangent functions, respectively. Furthermore, U is a recurrent connection weight matrix that captures sequential signals between every two adjacent hidden states $h_\tau$ and $h_{\tau-1}$ using $\odot$, which shows the element-wise product. Moreover, W and b are the transition matrix between the latent factors of venues and the corresponding bias, respectively. It should be noted that $\theta_r$ = {W, U, b} denotes the set of the parameters of the recurrent layer. Overall, W is the transition matrix between the latent factors of venues and b indicates the corresponding bias. Additionally, U is a recurrent connection weight matrix that captures sequential signals between every two adjacent hidden states. All the recurrent layer parameters (i.e., $W_z$, $U_z$, and $b_z$) are the set of the parameters of the update gate. $W_r$, $U_r$, and $b_r$, as well as $W_h$, $U_h$, and $b_h$ are the set of parameters of the reset gate and candidate hidden state, respectively. Similarly, $W_{Gts}$, $U_{Gts}$, and $b_{Gts}$ are the set of the parameters of our proposed $G_{ts}$. Finally, $W_{Gge}$, $U_{Gge}$, and $b_{Gge}$, as well as $W_{Gte}$, $U_{Gte}$, and $b_{Gte}$ are the set of the parameters of our proposed $G_{ge}$ and $G_{te}$, respectively.

At current step $\tau$, the correlation between the latent factor of absolute contexts $\phi t^\tau$ and the hidden state from the earlier step $h_{\tau-1}$ is calculated by Eq. (16):

$$G_{ts} = \sigma\, (W_{Gts}\, h_{\tau-1} + W_{Gts}\, \phi t^\tau + b_{Gts}) \quad (16)$$

To effectively model the users' sequential order of check-ins, the relevant contextual information needs to be examined separately. To address this issue, the current study proposed $G_{ge}$ and $G_{te}$ to individually incorporate the geographical distance ($\Delta g_\tau$) and time interval ($\Delta t_\tau$) between two check-ins as Eqs. (17) and (18):

$$G_{ge} = \sigma\, (W_{Gge}\, h_{\tau-1} + W_{Gge}\, \Delta g^\tau + b_{ge} \quad (17)$$

$$G_{te} = \sigma\, (W_{Gte}\, h_{\tau-1} + W_{Gte}\, \Delta t^\tau + b_{te}) \quad (18)$$

With the proposed gates for GTCI-EAGRU architecture, the equations of the traditional GRU are updated as Eqs. (19), (20), and (21):

$$z_\tau = \sigma(W_z\, \emptyset v_j^\tau + U_z h_{\tau-1} + \boldsymbol{W_z}((\boldsymbol{G_{ts}} \odot \boldsymbol{\phi t^\tau}) + (\boldsymbol{G_{ge}} \odot \boldsymbol{\Delta g^\tau}) + (\boldsymbol{G_{te}} \odot \boldsymbol{\Delta t^\tau})) + b_z) \quad (19)$$

$$r_\tau = \sigma(W_r\, \emptyset v_j^\tau + U_r h_{\tau-1} + \boldsymbol{W_r}((\boldsymbol{G_{ts}} \odot \boldsymbol{\phi t^\tau}) + (\boldsymbol{G_{ge}} \odot \boldsymbol{\Delta g^\tau}) + (\boldsymbol{G_{te}} \odot \boldsymbol{\Delta t^\tau})) + b_r) \quad (20)$$

$$\hat{h}_\tau = tanh(W\, \emptyset v_j^\tau + U\, (r_\tau \odot h_{\tau-1}) + \boldsymbol{W_r}((\boldsymbol{G_{ts}} \odot \boldsymbol{\phi t^\tau}) + (\boldsymbol{G_{ge}} \odot \boldsymbol{\Delta g^\tau}) + (\boldsymbol{G_{te}} \odot \boldsymbol{\Delta t^\tau})) + b_h) \quad (21)$$

In the following section, the hidden sate $h_\tau$ will be updated and as previously mentioned, it will be the output of the recurrent unit at time step τ.

## 4-4- Output Layer

In the next POI recommendations based on the MF approach, recommendations are mainly derived from a dot product of the latent factors of users $U \in R^{|U|\times d}$ and venues $V \in R^{|V|\times d}$ where d is the number of latent dimensions (i.e. $\hat{c}_{i,j} = \emptyset u_i\, \emptyset v_j^T$ ) and $\emptyset u_i$ and $\emptyset v_j$ denote the latent factors of user i and venue j, respectively [2,36]. In the output layer, the preference of user u on venue v at timestamp t is estimated using Eq. (22):

$$\hat{c}_{u,v,t} = \emptyset u_u\, h^{\tau\ T} \quad (22)$$

According to previous works, the pairwise loss function outperformed the classification loss function in learning patterns from sequential data and was more efficient for the network training of the recurrent-based recommendation [2,3,13,20]. Therefore, following Manotumruksa et al. [2,13], the pairwise BPR [20] can be applied to estimate the embedding and recurrent layer parameters and the probability distribution over all venues given the hidden state $h^\tau$.

## 4-5- Network training

This study employed datasets consisting of a set of sampled triplets each containing one user and a pair of POIs in which one POI is positive (known as visited) while the other one is negative (known as unvisited). As mentioned earlier, this study applied the pairwise BPR to learn the embedding and recurrent layer parameters ($\Theta$ = $\{\theta_e, \theta_r\}$). Based on an underlying assumption, stating that a user prefers the observed POI to all unobserved ones, BPR considers the relative order of the predictions for the pairs of POIs [1,4]. At each sequential position k in the BPR framework, the goal of GTCI-EAGRU is to maximize the following probability [1,4,20]:

$$P(u, t, v > v') = g(\, o_{u,t,v} - o_{u,t,v'}) \quad (23)$$

$v$ and $v'$ stand for a positive (visited) POI and a negative (unvisited) POI, respectively, and $g(\cdot)$ represents a nonlinear function defined by Eq. (24) as [1, 20]:

$$g(x) = \frac{1}{1+e^{-x}} \quad (24)$$

The objective function of the network for the next POI recommendation can be solved by integrating the loss function and a regularization term as follows [20]:

$$J = -\sum_{(v,v')} \ln P(u,t,v > v') + \lambda/2\, \|\, \Theta\, \|^2 \quad (25)$$

where $\lambda$ is used to specify the power of regularization and $\Theta$ is the parameter set. The dimension of the latent factors d and hidden layers $h_\tau$ of GTCI-EAGRU architecture d = 10 across three datasets can be set based on methods by Manotumruksa et al. [2] and Kala et al. [3], and all embedding and recurrent layers' parameters can be randomly be initiated with a Gaussian distribution. Initially, the learning rate and the batch size are set to 0.001 and 256, respectively. An Adam optimizer was employed to optimize the model parameters. The output of the GTCI-EAGRU model is a set of scores for POIs, similar to their likelihood of being the next POI in each sequence. A summary of the learning algorithm of GTCI-EAGRU is provided as follows:

| **Algorithm 1**: Training of GTCI-EAGRU | |
|---|---|
| **Input**: Set of users $U$s and set of historical check-in sequences $S^u$ | |
| **Output**: GTCI-EAGRU model $\{\ \Theta\ \}$ | |
| //construct training instances | |
| 1. | Initialize $D=Us_u\ D^u = \emptyset\ D^u$ is a set of check-in trajectory samples combined with negative POIs of $u$ |
| 2. | **For** each user $u \in U$s **do** |
| 3. | **For** each check-in sequence $S^u = \{s_{t1}{}^u, s_{t2}{}^u, \ldots, s_{tn}{}^u\}$ **do** |
| 4. | Get the set of negative samples $v'$ |
| 5. | **For** each check-in activity in $S^u$ **do** |
| 6. | Compute the embedded vector $v_\tau{}^u$ |
| 7. | Compute the geographical contexts vector $g_\tau{}^u$ |
| 8. | Compute the temporal contexts vector $t_\tau{}^u$ |
| 9. | **End for** |
| 10. | Add a training instance $(\{v_\tau{}^u, g_\tau{}^u, t_\tau{}^u\},\{v'\})$ into $D^u$ |
| 11. | **End for** |
| 12. | **End for** |
| //train the model | |
| 13. | Initialize the parameter set $\Theta$ |
| 14. | **While** (exceed(maximum number of iterations)==FALSE) **do** |
| 15. | **For** each user $u$ in $U$ **do** |
| 16. | Randomly select a batch of instances $D_b{}^u$ from $D^u$ |
| 17. | Find $\Theta$ minimizing the objective (23) with $D_b{}^u$ |
| 18. | **End for** |
| 19. | **End While** |
| 20. | Return the set of parameter $\Theta$ |

## 5- Experimental Result and Analysis

This section presents the experimental setup and empirical results of this study. Empirical experiments are conducted on three public datasets in LBSNs for validating the efficiency of the proposed method. To address the challenges made in Section 2-4, the experiments are designed for the following research questions:

**RQ1:** How can the basic GRU architecture be extended to separately consider the absolute and relative (or transition) contextual information associated with the sequence of check-ins?

**RQ2:** Is it important to model absolute and relative (or transition) contextual information separately?

**RQ3:** Does GTCI-EAGRU that leverages multiple types of contextual information improve prediction accuracy by applying additional attention gates? Or, does it outperform the previous methods?

### 5-1- Datasets and Experimental Settings

The experiments were conducted for evaluating three publicly LBSN datasets (i.e., BrightKite[1], Gowalla[2], and Foursquare[3] datasets). Following Manotumruksa et al. [2] some deletions were made to lessen data sparsity and cold start problems. Users with less than 10 check-ins and POIs with less than 10 were eliminated from the three datasets. Table 3 presents an overview of the statistics of the three datasets. In this study, a check-in record is a quadruple composed of a user, the corresponding check-in timestamp, the geographical coordinates of the check-in, and a location Id or POI. The check-in records in these three datasets were regarded as user sequences. The density calculation formula for three datasets is as follows [38]:

$$Density = \frac{|check-ins|}{|users|\times|POIs|} \tag{26}$$

Table 3. Statistics of the three datasets

| *Dataset* | *#Users* | *#Check-ins* | *#POIs* | *Density* |
|---|---|---|---|---|
| ***Brightkite*** | 915 | 676721 | 7527 | 0.0982 |
| ***Gowalla*** | 1047 | 614340 | 5011 | 0.1170 |
| ***Foursqure*** | 615 | 108195 | 19245 | 0.0091 |

A leave-one-out evaluation method was adopted to evaluate the efficiency of the proposed GTCI-EAGRU architecture based on earlier works [2], [3]. Each user's most recent check-in was taken as the base, and 100 POIs, which had not been visited before, were randomly selected for this purpose. They were the testing set, and the other remaining check-ins were considered as the training set. The task of the GTCI-EAGRU was to rank those 100 venues for each user as their preferred contexts (i.e., timestamp, time interval, and geographical distance), aiming at ranking highest the recent, ground truth check-in. Following Manotumruksa et al. [2] and Kala et al. [3], the researchers set the dimension of the latent factors d and hidden layers hτ of the proposed GTCI-EAGRU architecture: d = 10. As mentioned before, Gaussian distribution [32] was employed for the random initialization of the recurrent layer's parameters, and Adam Optimizer [39] was utilized for optimizing the parameters because it had a faster convergence compared to the stochastic gradient descent optimization, which automatically adjusts the learning rate for each iteration. In addition, the batch size and the dropout rate were set to 256 and 0.2, respectively, to prevent overfitting.

### 5-2- Comparison

The following five up-to-date methods were compared to validate the efficiency of the GTCI-EAGRU in the next POI recommendation task. Table 4 summarizes these methods into different aspects. Based on data, they are categorized into MF-

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

, RNN-, and AM-based approaches. The compared models are also classified according to the use of GCI and TCI.

A brief description of these models is given below:

**STGN:** Spatio-temporal gated network was proposed by Zhao et al. [22], and improved the LSTM network, in which STGs are introduced for capturing the Spatio-temporal relationships between successive check-ins. By introducing new gates and cells to capture short- and long-term preferences, STGN modified the basic LSTM model.

**ARNN:** An attentional RNN was proposed by Guo et al. [7] to jointly model the transition regularities and sequential regularity of similar locations (neighbors). Using embedding, knowledge graph, LSTM, and AM, the ARNN captured sequential, spatial, temporal, and semantic influences.

**GeoSAN:** By introducing a new loss function, Lian et al. [37] resolved the sparsity issue. **GeoSAN** represents the hierarchical gridding of each GPS point with a self-attention based geography encoder for better use of geographical information.

**DRCF:** To benefit from the traditional RNN to model the sequential order of users' check-ins, Manotumruksa et al. [13] extended NeuMF. DRCF has two components each having its recurrent layer.

**CARA:** By employing embedding, GRU, and two gating mechanisms, Manotumruksa et al. [2] captured various types of the impact of different contextual information.

Following earlier works [6-8,22], the current study used prediction accuracy (*Acc@k*, $k$ = 10) for evaluating the performance of the above-mentioned methods and checking if the ground-truth location can be found in the top-$k$ recommendation list. Generally, the *Accuracy@* is defined by Eq. (27) as follows [29]:

$$Accuracy@k = \frac{number\ of\ samples\ correctly\ predicted}{total\ number\ of\ samples} \quad (27)$$

Table 4. Summary of all the baseline methods used in this study

| *Methods* | *Approaches and Contextual Information* | | | | |
|---|---|---|---|---|---|
| | ***MF*** | ***RNN*** | ***AM*** | ***GCI*** | ***TCI*** |
| ***STGN*** | × | √ | × | √ | √ |
| ***ARNN*** | × | √ | √ | √ | √ |
| ***GeoSAN*** | × | × | √ | √ | √ |
| ***DRCF*** | √ | √ | × | × | × |
| ***CARA*** | √ | √ | √ | √ | √ |
| ***GTCI-EAGRU*** | √ | √ | √ | √ | √ |

*Note*. MF; Matrix factorization; RNN: Recurrent neural network; AM: Attention mechanism; GCI: Geographical contextual information; TCI: Temporal contextual information; STGN: Spatio-temporal gated network; ARNN: Attentional Recurrent Neural Network; GeoSAN: Geography-aware sequential recommender based on the Self-Attention Network; DRCF: Deep Recurrent Collaborative Filtering; CARA: Contextual attention recurrent architecture ; GTCI-EAGRU: Geographical temporal contextual information-extended attention gated recurrent unit.

## 5-3- Results and Discussion

Table 5 compares the recommendation results of six methods on the three datasets. The numbers in bold in each column represent the best performance.

Table 5. Comparison of different methods in recommendation performance

| Methods | Acc@10 | | |
|---|---|---|---|
| | **Brightkite** | **Gowalla** | **Foursquare** |
| **STGN** | 0.2020 | 0.5231 | 0.3017 |
| **ARNN** | - | 0.2336 | 0.4285 |
| **GeoSAN** | 0.6425 | 0.6028 | 0.4867 |
| **DRCF** | 0.7363 | - | 0.8805 |
| **CARA** | 0.7385 | - | 0.8851 |
| **GTCI-EAGRU** | **0.9751** | **0.9606** | **0.8901** |

*Note*.: STGN: Spatio-temporal gated network; ARNN: Attentional Recurrent Neural Network; GeoSAN: Geography-aware sequential recommender based on the Self-Attention Network; DRCF: Deep Recurrent Collaborative Filtering; CARA: Contextual attention recurrent architecture; GTCI-EAGRU: Geographical temporal contextual information-extended attention gated recurrent unit.

The comparison of the experimental results of the models demonstrated that the use of AM alone (i.e., the GeoSAN model) has not increased prediction accuracy. Moreover, the experimental results of other previous studies (e.g., DAN-SNR) revealed lower evaluation metrics values. Although the STGN model separately considered the GCI and TCI, it did not use the attenuation mechanism approach. It applied the LSTM model and was less prediction accurate compared to models that employed the GRU such as CARA. Although the ARNN model applied the LSTM model, it had a higher accuracy prediction in comparison with the STGN model due to the use of the attenuation mechanism. Similar to the STGN, it had less prediction accuracy compared to models that considered the GRU model.

The GeoSAN model only uses the AM for location recommendation, and despite considering geographical and TCI, it is less prediction accurate than DRCF and CARA models. The DRCF model pays attention to the sequence of previously visited venues while not taking into consideration the contextual information related to the check-ins. Thus, its prediction accuracy is lower than that of the hybrid models. However, it should be stated that the performance of these hybrid approaches was not worse than that of RNN and LSTM. Thus, it is worth modeling geographical and spatial contextual information for the task of the next POI recommendations. It means that it is insufficient to have a good network architecture, but more geographical and spatial contextual information of human check-in behaviors should be taken into account to obtain excellent results [1]. This is the reason for the outperformance of CARA over DRCF.

The accuracy prediction in the CARA model is higher compared to other models due to the separate use of TCI and GCI and a combination of the RNN, attention, and factoring approaches. Inspired by the idea behind this model, the researchers introduced a new initiative to employ three gates in the GRU model to address GCI and TCI to better predict the accuracy of the next POI recommendation. As mentioned in previous sections, the proposed model uses three separate attention gates, namely, $G_{ts}$, $G_{ge}$, and $G_{te}$, which consider the timestamp, geographical distance, and time interval between successive check-ins, respectively, and the output of each of them separately affects the values of the reset and update gates of the GRU model. As depicted in Fig. 3, the experiment results of the proposed models indicate that it has achieved this goal, and the accuracy prediction has been improved in the proposed GTCI-EAGRU architecture.

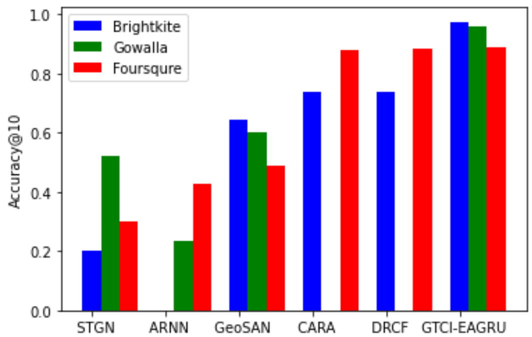


Fig. 3. Comparison of GTCI-EAGRU with baseline methods in terms of Accuracy@10 on three datasets

To answer RQ1 for the development of the GRU model, it should be mentioned that three gates were introduced and implemented as a feed-forward network. The output of these gates affects the values of the GRU reset and update gates, and they are responsible for controlling the geographical and temporal information of the user's trajectory data. To answer RQ2 and RQ3, these results were obtained (Table 6) by comparing the accuracy prediction of the GTCI-EAGRU model with up-to-date architectures.

Table 6. Percentage of Improvement of GTCI-EAGRU

| *Methods* | *Percentage of Improvement* | | |
|---|---|---|---|
| | ***Brightkite (%)*** | ***Gowalla (%)*** | ***Foursquare (%)*** |
| ***STGN*** | 79.28 | 45.54 | 66.10 |
| ***ARNN*** | - | 75.68 | 51.86 |
| ***GeoSAN*** | 34.11 | 37.25 | 45.32 |
| ***DRCF*** | 24.49 | - | 01.08 |
| ***CARA*** | 24.26 | - | 00.56 |
| ***Improvement*** | 40.54 | 52.82 | 32.98 |
| ***On Average in three Dataset*** | 42.11 | | |

## 6- Conclusions

In recent years, the next POI recommendation is of great importance for a wide spectrum of LBSN applications. The influences of contextual information (e.g., spatial and temporal context information) are crucial for analyzing individual behaviors for personalized POI recommendations. Hence, many studies have considered this contextual information to improve the performance of POI recommendation algorithms such as the CF and RNN. There are still many challenges regarding how to integrate contextual information to accurately model users' complex behavioral preferences and recommend reliable POIs to users.

The current study proposed a novel GTCI-EAGRU for the next POI recommendation by addressing the challenges concerning previous studies. Our proposed architecture was presented with the development of the GRU model, in which the contextual information of the user trajectory data is considered separately. Moreover, the development of the model inspired by the AM makes contextual information more important in modeling sequential user data. POIs were scored to provide recommendations to a user from her/his historical check-ins. The simple development of this model for considering more contextual information is one of the other features of the proposed model.

By comparing the experimental results of baseline methods, an increase in the accuracy of prediction indicates the importance of considering contextual information separately. The proposed GTCI-EAGRU architecture with three additional contextual attention gates worked well for the next POI recommendation.

In this study, the comprehensive experiments conducted on three large- scale datasets from the Brightkite, Gowalla, and Foursquare demonstrated a significant improvement in the GTCI-EAGRU architecture for the next POI recommendations compared with various up-to-date recurrent architectures and many different recent factorization approaches.

To enhance the quality of recommendations for the next POI, the GTCI-EAGRU architecture could be enriched by adding the impact of each user's social relationships with other users on LBSNs. Furthermore, it can be possible to include more contextual information (e.g., visual and text information) related to users' check-ins or the weather condition of the check-in registration location as well.

**Acknowledgment**

This manuscript is prepared based on PhD. thesis of the first author at Rasht Branch, Islamic Azad University, Rasht, Iran.

## References

[1] L. Huang, Y. Ma, Sh. Wang, Y. Liu, "An Attention-based Spatiotemporal LSTM Network for Next POI Recommendation", Journal of IEEE Transactions on Services Computing, vol. 12, 2019, pp. 1-13.

[2] J. Manotumruksa, C. Macdonald, I. Ounis, "A Contextual Attention Recurrent Architecture for Context- Aware Venue Recommendation", in 18th ACM SIGIR Conference on Research and Development in Information Retrieval, 2018, pp. 555-564.

[3] 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, https://doi.org/10.1007/s12652-019-01583-w.

[4] C. Liu, J. Liu, J. Wang, S. Xu, H. Han, Y. Chen, "An Attention-Based Spatiotemporal Gated Recurrent Unit Network for Point-of-Interest Recommendation", International Journal of Geo-Information, vol. 8, No. 8, 2019, pp.355-373.

[5] S. Wang, Z. Bao, J. Culpepper, G. Cong, "A Survey on Trajectory Data Management, Analytics, and Learning", ACM Computing Surveys, vol. 54, No.3, 2020, pp. 1-33.

[6] J. Feng Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, D. Jin, "DeepMove: Predicting Human Mobility with Attentional Recurrent Networks", in 18th ACM IW3C2 Conference on International World Wide Web, 2018, pp. 1459-1468.

[7] Q. Guo, Z .Sun, J. Zhang, Y. Theng, "An Attentional Recurrent Neural Network for Personalized Next Location Recommendation", in 34th ACM AAAI Conference on Artificial Intelligence, 2020, pp. 83-90.

[8] D.Yao,C. Zhang, J. Huang, J. Bi, "SERM: A Recurrent Model for Next Location Prediction in Semantic Trajectories", in 17th ACM CIKM Conference on Information and Knowledge Management, 2017 ,pp. 2411-2414.

[9] L. Zhang, Z. Sun, J. Zhang, H. Kloeden, F. Klanner, "Modeling hierarchical category transition for next POI recommendation with uncertain check-ins", Journal of Information Sciences, Elsevier, vol.515, 2019, pp. 169-190.

[10] L. Chang, W. Chen, J. Huang, Ch. Bin, W. Wang, "Exploiting multi-attention network with contextual influence for point-of-interest recommendation", Journal of Applied Intelligence, vol. 51, 2021, pp. 1904–1917.

[11] L. Huang, Y. Ma, Y. Liu, K. He, "DAN-SNR: A Deep Attentive Network for Social-Aware Next Point-of-Interest Recommendation", Journal of ACM Transactions on Internet Technology, Vol.21, No.2, 2020, pp. 1–27.

[12] G. Christoforidis, P. Kefalas, A. Papadopoulos, Y. Manolopoulos, "RELINE: Point-of-Interest Recommendations using Multiple Network Embeddings", Journal of Knowledge and Information Systems, Vol. 63, No.4, 2019, pp. 791-817.

[13] J. Manotumruksa, C. Macdonald, I. Ounis, "A Deep Recurrent Collaborative Filtering Framework for Venue Recommendation", in 17th ACM CIKM Conference on Information and Knowledge Management, 2017, pp. 1429-1438.

[14] D. Yang, D Zhang, V. Zheng, Z. Yu, "Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs", Journal of IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol.45, No.1, 2014, pp. 129 – 142.

[15] M. Quadrana, P. Cremonesi, D. Jannach, "Sequence-Aware Recommender Systems", Journal of ACM Computing Surveys, Vol.51, No.4, 201, pp. 1–36.

[16] Q. Cui, Y. Tang, S. Wu, L. Wang, "Distance2Pre: Personalized Spatial Preference for Next Point-of-Interest Prediction", in PAKDD Conference on Knowledge Discovery and Data Mining, 2019, pp. 289-301.

[17] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong, F. Zhang, "Predicting Human Mobility via Variational Attention", in IW3C2 Conference on International World Wide Web Conference Committee, 2019, pp. 2750–2756.

[18] A.Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need", in 31th NIPS Conference on Neural Information Processing System, 2017, pp. 5998-6008.

[19] Y. Chen, C. Long, G.Cong, C. Li, "Context-aware Deep Model for Joint Mobility and Time Prediction",in 13th ACM WSDM Conference on Web Search and Data Mining, 2020, pp. 106-114.

[20] S. Rendle, C. Freudenthaler, Z. Gantner, L. Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback", in 25th ACM UAI Conference on Uncertainty in Artificial Intelligence, 2009, pp. 452–461.

[21] E.Cho, S. Myers, J. Leskovec, "Friendship and Mobility: User Movement in Location-Based Social Networks", in 17th ACM KDD Conference on Knowledge Discovery and Data Mining, 2011, pp. 1082–1090.

[22] P. Zhao, H. Zhu, Y. Liu, J. Xu, F. Zhuang, V. Sheng, X. Zhou, "Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation",. in 33th AAAI Conference on Artificial Intelligence, 2019, pp. 5877-5884.

[23] A M. Islam, M. M. Mohammad, S. S. Das, M. E. Ali, "A Survey on Deep Learning Based Point-Of-Interest (POI) Recommendations", 2020, arXiv:2011.10187v1.

[24] C. Zheng, D. Tao, "Attention-Based Dynamic Preference Model for Next Point-of-Interest Recommendation". in 15th Springer WASA Conference on Wireless Algorithms, Systems, and Applications, 2020, pp. 768–780.

[25] D. K. Bokde, Sh. Girase, D. Mukhopadhyay, "Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey", International Journal of Advance Foundation and Research in Computer, vol.1, 2014, pp. 111-118.

[26] M. Gan, L. Gao, "Discovering Memory-Based Preferences for POI Recommendation in Location-Based Social Networks", International Journal of Geo-Information (IJGI), Vol.8, No.6, 2019, pp. 279-294.

[27] X. Meng, J. Fang, "A Diverse and Personalized POI Recommendation Approach by Integrating Geo-Social Embedding Relations", Journal of IEEE Access, Vol.8, 2020, pp. 226309- 226323.

[28] Q. Yuan, G. Cong, Z. Ma, A. Sun, N. Thalmann, "Time-aware Point-of-interest Recommendation", in 36th ACM SIGIR Conference on Research and development in Information Retrieval, 2013, pp. 363–372.

[29] P. Wang, H. Wang, H. Zhang, F. Lu, S. Wu, "A Hybrid Markov and LSTM Model for Indoor Location Prediction", Journal of IEEE Access, Vol.7, 2019, pp. 185928 – 185940.

[30] J. Li, G. Liu, C. Yan, C. Jiang, "LORI: A Learning-to-Rank-Based Integration Method of Location Recommendation", IEEE Transactions on Computational Social Systems, Vol.6, No.3, 2019, pp. 430 – 440.

[31] L.Yao, Q. Z. Sheng, Y. Qin, X. Wang, A. Shemshadi, Q. He, "Context-aware Point-of-Interest Recommendation Using Tensor Factorization with Social Regularization", in 38th.ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 1007–1010.

[32] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, "Neural collaborative filtering", in 26th ACM IW3C2Conference on World Wide Web Conference Committee, 2017, pp. 173-182.

[33] Q. Liu, S.Wu, L. Wang, T. Tan, "Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts", in 30th ACM AAAI Conference, 2016 , pp. 194–200.

[34] S. Kumar, M.I. Nezhurina, "An ensemble classification approach for prediction of user's next location based on Twitter data", Journal of Ambient Intelligence and Humanized Computing, Vol.10, No. 11, 2018, pp. 4503-4513.

[35] Q. Liu, S. Wu, D. Wang, Z. Li, L. Wang, "Context-Aware Sequential Recommendation", in ICDM Conference on Data Mining, IEEE, 2016, pp. 1053-1058.

[36] D. Bokde, S. Girase, D. Mukhopadhya, "Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey", Procedia Computer Science, Vol.49, 2015, pp. 136-146.

[37] D. Lian, Y. Wu, Y. Ge, X. Xie, E. Chen, "Geography-Aware Sequential Location Recommendation", in 26th ACM

ICGKDD Conference on Knowledge Discovery and Data Mining, 2020, pp. 2009–2019.
[38] K. Yang, J. Zhu, "Next POI Recommendation via Graph Embedding Representation from H-Deepwalk on Hybrid Network", Journal of IEEE Access, Vol 7, 2019, pp. 171105 – 171113.
[39] D. P. Kingma, J.L. Ba, "A Method for Stochastic Optimization", in International Conference for Learning Representations, 2015, arXiv:1412.6980v.