Intelligent Multimedia Processing & Communication Systems Journal

IMPCSJ

J IMPCS (2025) 22: 51-69

DOI 10.71856/impcs.2025.1224513

Research Paper

Point of Interest Recommendation by Reality Mining Approach in Recommender Systems

Fatemeh Ghanaati*

Department of Computer Engineering, Ast.C., Islamic Azad University, Astaneh Ashrafiyeh, Iran.

* *Corresponding Author*, Fatemeh.Ghanaati@iau.ac.ir

Article Info

Article history:

Received: 13 Oct 2025

Accepted: 22 Nov 2025

Keywords:

Deep learning,
Location Based Social Networks,
Reality Mining, Point of Interest,
Recommender Systems.

ABSTRACT

Location-based social networks (LSBNs) are emerging services that have gained considerable popularity in recent years with the rapid advancement of mobile technology. LSBNs enable users to log their locations by recording entries, including various forms of contextual information (CI). Reality mining (RM) involves collecting and analyzing environmental and behavioral data from mobile devices to uncover predictable patterns, such as human mobility trends, which can enhance sequential next Point of Interest (POI) recommendations in recommender systems. Probabilistic models and sequence-based algorithms are among the most widely used approaches for learning mobility patterns in RM, although each presents its own challenges. In this study, for the first time, incorporate CI from LBSNs in a reality mining framework to predict users' next POI within recommender systems. To this end, we propose a Contextual Extended Gated Recurrent Unit (CEGRU) architecture designed to separately investigate the impact of CI on location prediction. The CEGRU model extends the traditional GRU by introducing two distinct attention gates to better capture the impact of contextual variables on user movement behavior. Furthermore, this research introduces a novel experimental setup that evaluates model performance under two different dataset density conditions. This innovation enables the determination of the optimal dataset density for effectively assessing the proposed model. Comprehensive experiments were conducted on three large-scale real-world LBSN datasets, including Brightkite, Gowalla, and Foursquare. The results demonstrate that CEGRU outperforms competitive baseline methods on the Brightkite and Gowalla datasets in terms of Acc@10.

Publisher: Islamic Azad University – Zanjan Branch

I. Introduction

Location-based social networks (LSBNs) services such as ride-hailing, targeted advertising, and food delivery have become an integral part of daily life with the rapid advancement of smartphone technology [1]. To deliver more effective services in such applications, it is essential to accurately predict users' future locations by recommending successive points of interest (POIs) [2]. LSBNs allow users to record their locations through check-ins that include various forms of contextual information (CI), such as geographical and temporal contextual information (GTCI). The GTCI associated with user check-ins plays a crucial role in analyzing movement patterns and predicting users' next POIs. Effective successive POI recommendation not only supports intelligent, location-based advertising and personalized user experiences but also helps service providers optimize user engagement and promote exploration of new places [3, 23, 25].

Reality mining (RM) is defined as the collection and analysis of environmental data from mobile devices associated with human social interaction, with the aim of identifying predictable behavioral patterns [4, 5, 6, 7]. One key area of RM research involves predicting human mobility patterns, which can be utilized to improve successive POI recommendations and monitor individuals' locations during pandemic situations [5, 8, 9]. RM investigates human behavior through wireless devices such as smartphones and Global Positioning System (GPS) sensorsto construct an accurate representation of individuals' activities, movements, and social interactions [4, 5]. With advances in machine learning (ML) and statistical analysis, RM now provides a broader understanding of both collective and individual human behavior [8]. To further enhance this capability, we propose employing deep learning (DL) techniques and predictive big data analytics on LBSN check-in data to generate more accurate mobility predictions. The primary objective of this study is to develop a Recurrent Neural Network (RNN)-based RM model capable of forecasting user movement patterns by incorporating CI derived from LBSN check-ins, with the potential for application to other types of CI in the future. Previous RM studies [5, 10] have largely overlooked the integration of contextual data such as temporal and geographical factors captured through user check-ins on LBSNs. However, CI plays a critical role in modeling user mobility behavior and has a distinct influence on predicting users' future locations.

This study focuses on RM by modeling sequences of user check-ins while separately considering the influence of GTCI. To achieve this, we propose a novel contextual extended gated recurrent unit (CEGRU) architecture for location prediction, consisting of four layers: input, output, embedding, and recurrent. Compared with other deep learning–based recurrent models, the Gated Recurrent Unit (GRU) architecture is relatively simple and requires fewer parameters. Furthermore, unlike conventional Recurrent Neural Networks (RNNs), the GRU architecture can selectively disregard the hidden state of the previous unit when it is not relevant [11, 12]. As a result, a GRU network was developed to model check-in sequences by incorporating the time intervals (Δt) and geographical distances (Δg) between two successive check-ins [13]. Fig.1 illustrates an example of a user's check-in sequence.

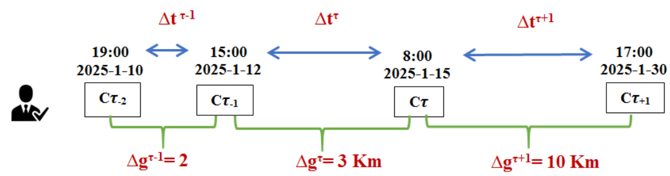

Fig.1. An example of a user's check-in sequence

Since the effects of CI on user behavior vary, any type of GTCI should be considered individually during modeling [14, 15]. The GRU network was extended by introducing two extra attention gates to independently capture the influence of important CI, inspired by the attention mechanism (AM). In the output layer of the CEGRU architecture, the preference score is computed using the dot product operation, and the top-*k* POIs are recommended to users based on these prediction scores. A higher score indicates a greater likelihood that the user will visit the corresponding location. The CEGRU parameters were optimized using the Bayesian Personalized Ranking (BPR) framework [16]. In the final stage, extensive experiments were conducted on three benchmark datasets , including Brighkite, Gowalla, and Foursquare to evaluate our proposed model. The performance of CEGRU was compared against five state-of-the-art location methods.

A. Problem Statement

Probabilistic models and sequence-based algorithms are the most widely used approaches for learning user movement patterns in RM. However, probabilistic approaches, which often use the Markov model, struggle to capture behavioral patterns in long movement sequences, while sequence-based methods perform poorly when predicting users' future locations in low-repetition or sparsely visited areas. Users can record check-ins on various LSBNs to share their locations. The check-ins collected in LBSNs contain GTCI, each exerting a distinct influence on predicting a user's next location [15, 19]. Previously, no GTCI was incorporated into sequence modeling in RM investigations (Challenge #1). In addition, several LBSN-based approaches for successive POI predictions, such as collaborative filtering (CF) and RNNs, used the CI of user trajectory data to forecast future POIs. However, CF methods ignore the sequential nature of the data, despite the fact that successive POI prediction is inherently a time-sequence problem (Challenge #2). Although recurrent models address sequential data modeling, they often overlook the role of CI, which can significantly influence predictions (Challenge #3). The CEGRU model was presented in this study to address these limitations.

B. Main Contributions

The following are significant contributions to this work:

1) Mobile phones equipped with sensors and GPS receivers are the primary data sources for RM applications.

Consequently, most research in this field has focused on analyzing location-based data obtained from GPS, which is often affected by noise and dispersion. In this study, the RM data source is users' registered check-ins on LBSNs, which include the GTCI of trajectory data. This represents the first time that contextual data has been incorporated into RM. Secondly, the construction of a GRU-based model for RM is proposed for the first time. The proposed architecture enhances the traditional GRU model by introducing two additional attention gates in the recurrent layer, leveraging an attention-based mechanism: the Geographical Contextual Attention Gate (GAG) and the Temporal Contextual Attention Gate (TAG). These gates regulate the influence of the previous recurrent unit's hidden state based on time intervals and geographical distances between successive check-ins. Notably, the time-stamp attention gate was excluded due to its minimal impact on prediction accuracy [26], reducing model parameters and computational overhead. Thirdly, this research investigates, for the first time, the effect of dataset density on model performance. This analysis aims to identify the optimal density for selecting datasets when evaluating the proposed model. Experiments were conducted under two distinct dataset density states, examining each dataset at multiple levels of change, following the methodology of [23]. This novel experimental design provides a framework for future dataset selection in RM research. Finally, extensive experiments were performed on three large-scale, real-world datasets, Gowalla [20], Brightkite [20], and Foursquare [21] widely used in related studies for predicting user POIs in LBSNs. The results demonstrate the effectiveness of the proposed CEGRU architecture.

Reality Mining aims to extract meaningful behavioral patterns from large scale real world data generated by human activities. With the rapid growth of location based social networks (LBSNs), massive volumes of user check-in data have become available, providing rich spatiotemporal traces that reflect users' real-world mobility behaviors. These check-in sequences can be regarded as observable manifestations of human activities, making them a valuable data source for Reality Mining studies. To this end, we propose the CEGRU model, which explicitly incorporates geographical and temporal contexts into the recurrent learning process to mine latent spatiotemporal behavioral patterns from real world user check-in data. The central assumption of this work is that explicitly modeling heterogeneous contextual factors, such as temporal intervals and geographical distances, within a recurrent architecture enables more effective Reality Mining of human mobility behaviors. Our experimental results demonstrate that this approach effectively captures mobility patterns, leading to superior performance in next location prediction compared to existing benchmarks.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents the study's background, while Section IV describes the details of the proposed model. The experimental results are reported in Section V, and Section VI concludes the paper.

II. Related Research

This section reviews related work on location prediction, which is organized into three categories: Reality Mining (RM) approaches, deep learning (DL) approaches, and hybrid methods.

RM approaches: Ferrari et al. [18] proposed a method for classifying and predicting users' whereabouts patterns using an RM dataset, which logs places visited as determined by GSM-based geolocation. Their approach involved automatically labeling routine locations from mobility data and developing a prediction mechanism to infer users' future whereabouts. Latent Dirichlet Allocation (LDA) was employed to extract high-level patterns, referred to as "themes," from the mobility dataset. Farrahi et al. [9] introduced an RM framework for large-scale, unsupervised learning of human routines through simultaneous modeling of user positions and proximity interactions. They proposed a multimodal behavior bag that combines semantic modeling of location changes across multiple time scales with interaction types derived from Bluetooth proximity data. LDA was applied to identify common multimodal human activities captured in RM data, representing routine behaviors. Jung et al. [10] focused on uncovering real-world social relationships, such as those between family and friends, using RM. They posited that an individual's context is intertwined with the contexts of socially connected peers, and that neighbors' contexts can significantly influence personal behavior. This assumption underlies the concept of social affinity, where stronger social ties result in greater contextual influence from others. Based on RM, Eagle and Pentland [6] explored the structure underlying daily human behavior. Their models aggregated multimodal data from individuals and communities within social networks. By calculating a weighted sum of an individual's principal eigen behaviors, their approach could simulate daily behavior and, if computed halfway through the day, predict the remaining behaviors. This method leverages the vast amounts of rich data collected continuously from mobile devices and nearby phones. Choujaa and Dulay [22] employed information-theoretic approaches to optimize the selection of time points for predicting mobile phone users' activities over the following three weeks. Their method analyzed RM cellular data to minimize uncertainty and to infer an individual's activity at one time from the activities of others at different times.

DL Approaches: DL-based recurrent models have recently shown significant advances in representing sequential user behavior and improving location prediction. For instance, Liu et al. [29] proposed a Spatial-Temporal Recurrent Neural Network (ST-RNN) to address continuously valued spatial-temporal contextual input challenges in location prediction. Zhao et al. [30] introduced STLSTM, an extended version of Long Short-Term Memory (LSTM) that incorporates distance and time gates to capture spatiotemporal relationships between successive check-ins. Specifically, separate distance and time gates are designed to regulate short-term interest updates, while additional gates capture latent location transition patterns. A task-specific decoder is also employed to enhance long-term interest modeling. To reduce parameter complexity and improve efficiency, our proposed model similarly integrates linked input and forgets gates. Kumar and Nezhurina [31] analyzed Twitter data and developed an ML model to predict users' future locations. ML methods enable systems to learn from past data and apply this knowledge to forecasting and decision-making for unseen instances. Yao et al. [19] presented a Semantics-Enriched Recurrent Model (SERM) for jointly learning embeddings of multiple factors (e.g., location, user, keyword, and time) and the transition parameters of an RNN within a unified framework. To better capture the interaction between user activities and site preferences, Zung et al. [32] proposed an Interactive Multi-Task Learning (iMTL) framework. This model incorporates a temporal-aware activity encoder with fuzzy representations for uncertain check-ins to reveal latent activity transition patterns, and a spatial-aware location preference encoder that uses the learned patterns to enhance both activity and location prediction tasks interactively. Liu et al. [33] proposed the Context-Aware RNN (CA-RNN), which replaces the constant input and transition matrices of traditional RNNs with adaptive, context-specific counterparts. These adaptive matrices capture the external contextual settings of user behaviors such as location, time, and weather and model how global sequential transitions are influenced by varying time intervals between consecutive actions. Moreover, attention mechanisms in DL have recently proven highly effective for improving interpretability and modeling long-term dependencies [12, 17, 24, 34–36]. Vaswani et al. [36] introduced the Transformer architecture, which eliminates recurrence and relies entirely on attention mechanisms to capture global input–output dependencies. Building on this concept, Huang et al. [17] developed ATST-LSTM, an attention-based spatiotemporal LSTM for next POI recommendation. Feng et al. [12] proposed the DeepMove model, which employs an attentional RNN to predict user mobility from long and sparse trajectories. Similarly, Huang et al. [37] introduced a Deep Attentive Network (DAN-SNR) for social-aware next POI recommendation. Despite these advancements, traditional RNN-based techniques still struggle to capture long-term dependencies effectively and may suffer from vanishing or exploding gradient problems. Table I summarizes the related studies and the corresponding challenges addressed in this research.

TABLE I: Summary of related works

Model Name	**Model Approach**	**Method summery**	**challenges**
[18] Topic prediction mechanism, [9] Multifaceted Behavior Package	RM Based	Latent Dirichlet Allocation Algorithm	The presence of noise and data dispersion and weakness in identifying temporal order and sequential dependencies in data
[10] Semi-automatic method	RM Based	Reality mining using social and spatial data contextual information and statistical analysis	Lack of predictability of a person's next location if they are in low-density areas
[6] Special Behavior Identification Model	RM Based	Analysis, prediction, and clustering of multimodal data of individuals and groups within a social network	Weakness in identifying temporal order and sequential dependencies in data
[22] Quantitative Predictability Method	RM Based	Presenting a method based on probability theory	Failure to properly extract behavioral patterns from long sequences of people's movements
[5] Detecting the dynamic structure of a real contact network	RM Based	Providing a method based on a Statistical Risk Model	Weakness in extract behavioral patterns present in long movement sequences of persons
[8] Real-time pattern prediction system	RM Based	K-means clustering algorithm	Limited energy of mobile devices
[29] ST-RNN	DL 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
[30] STGN	DL 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
[19] SERM	DL 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
[33] CA-RNN	DL 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
[17] ATST-LSTM	DL Based	Developing an attention-based spatiotemporal LSTM network to focus	Encountering with high complexity of implementation and a

Model Name	Model Approach	Method summery	challenges
		on the relevant historical check-in records in a check-in sequence selectively using the spatiotemporal contextual information	lack of attention to the scarcity
[37] DAN-SNR	DLAM 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

III. Preliminaries

This section presents notations and definitions, as well as the preliminary information that we used in our study.

A. Notations and Definitions

The primary notations utilized herein are listed in TABLE II.

TABLE II. Notations and description

Notation	Description
u, l, v, t	user, location (longitude and latitude), venue or POI, timestamp
lat v, lng v	POI *v*'s latitude and longitude (geographical coordinates)
$c_{u,v,t}$	user *u*-recorded check-in in POI *v* and timestamp *t*
$\Delta g, \Delta t$	geographical distance and time interval between two successive check-ins
S^u	a set of all user *u*-generated check-ins
Us, V, T	set of users, POIs, and timestamps
v_τ^u	POI visited by user *u* at timestep τ
t_τ^u, g_τ^u	vector representations of time interval and geographical distance
tr_u	a sequence of chronologically-ordered check-ins linked to *u*
$\phi u, \phi v, \phi t$	the latent factors of user *u*, POI *v*, and timestamp t
$\hat{h}, h$	the hidden and candidate states of CEGRU
z_r, r_r	update and reset gates of GRU
σ	sigmoid function

Check-in data: A check-in refers to an action performed by a user at a certain location and time. A check-in is an LBSN registration of a location that includes geographical and temporal data. The check-in record can be described as a quadruple: $c_{u,v,t}$ $<u, l, v, t>$ when a user u checks in at a location l (longitude and latitude) with venue-Id v at the timestamp t. S^u (user's check-in sequence) refers to a set of all users' check-ins.

Trajectory: A trajectory t is a series of chronologically ordered check-ins associated with a user u. For instance, tr_u: $<u, l_1, v_1, t_1>,\ldots,<u, l_i, v_i, t_i>,\ldots,<u, l_k, v_k, t_k>$, where tr_u denotes a user u's trajectory prior to time t_k. Here, a trajectory set Tr (u) is employed to represent all user u's trajectories.

POI in LBSNs: A POI in an LBSN is a spatial item linked to a geographical place and referred to as a venue, e.g., an office or a hotel. Here, *v* represents POI, and $V=\{v_1, v_2,...\}$ denotes a set of POIs. Each POI v has its own specific identifier and geographic coordinates, comprised of geographical latitude/longitude.

POI recommendations task: The POI recommendation task is to recommend the top-*k* POIs favored by user *u*, considering a collection of user check-in sequences S^u and a set of POIs V. The goal of the successive POI recommendation is to forecast which location v_k will be visited by a user *u* at a certain point in time t_{N+1}.

B. Deep Learning-based Recurrent Models

The primary challenge in successive POI recommendation lies in the joint and efficient learning of user POI preferences alongside the sequential correlations between check-ins [17]. This challenge is typically addressed by employing hidden states that capture the sequential patterns embedded in the input sequence [14, 15, 19]. Hidden states encode either the CI or the temporal dynamics of user trajectories. However, conventional RNNs struggle to capture long-term dependencies due to the exploding and vanishing gradient problems [14, 37]. To mitigate these limitations, the Long LSTM model was introduced, leveraging a gating mechanism to retain and regulate long-term information [17, 38, 41]. Each LSTM unit maintains a cell state (c_t) and a hidden state (h_t) similar to RNNs. The information flow between LSTM cells is governed by three gates: the input gate**,** forget gate**,** and output gate. However, the presence of three gates makes LSTM models computationally intensive and slower to train, especially on large datasets. To overcome these drawbacks, the GRU was proposed as a simplified yet effective alternative [11, 12, 38]. The GRU model replaces the three LSTM gates with only two: the reset gate and the update gate, which control how much of the previous hidden state should be forgotten or retained in the current computation [15]. These mechanisms enable GRU to learn temporal dependencies more efficiently with fewer parameters. The three fundamental recurrent models including RNN, LSTM, and GRU are illustrated in Fig. 1 as block diagrams [14, 30].

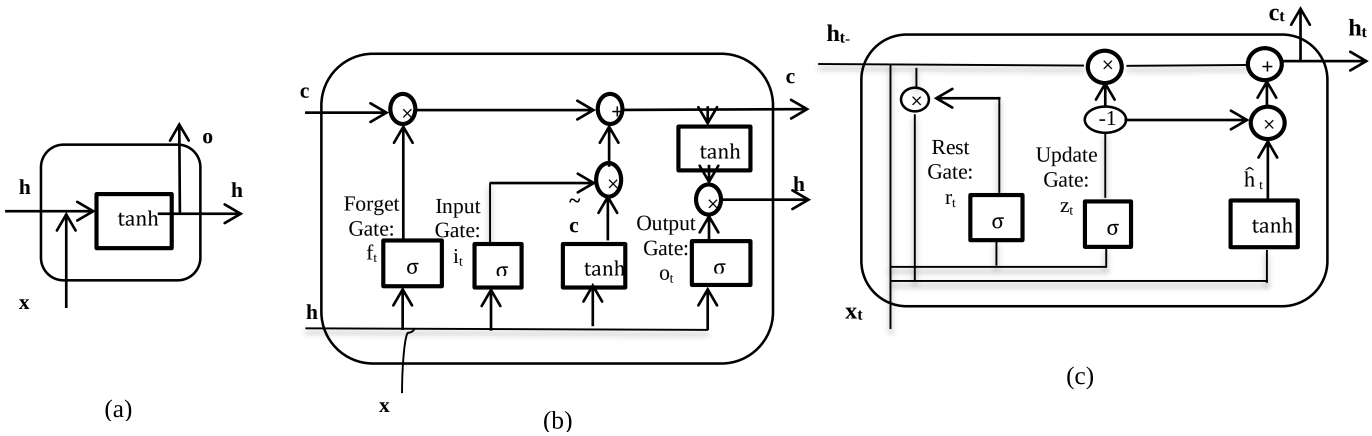

Fig.2. Block diagram of the (a) RNN, (b) LSTM, and the (c) GRU cell.

The GRU model contains only two gates the reset gate and the update gate as illustrated in Fig. 2. Compared to the LSTM, the GRU architecture is computationally more efficient and faster to train, particularly when the amount of training data is limited. The gates in the GRU regulate the update degree of each hidden state, thereby determining which information should be retained and which should be discarded in the transition to the next state [14, 15]. At a given time step τ, the GRU computes the hidden state $h_\tau$based on the outputs of the update gate z_τ, the reset gate r_τ, the current input x_τ, and the previous hidden state $h_{\tau-1}$. The reset gate determines how to combine the new input with the previous memory to calculate the candidate hidden state $\hat{h}_\tau$, and subsequently, the final hidden state $h_\tau$is updated as follows [39]:

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

$$r_\tau= \sigma(W_r x_\tau + U_r h_{\tau-1} + b_r) \quad (2)$$

$$\hat{h}_\tau = \tanh(W x_\tau+U (r_\tau \odot h_{\tau-1}) + b_h) \quad (3)$$

$$h_\tau=(1- z_\tau) \odot h_{\tau-1}+ z_\tau \odot \hat{h}_\tau \quad (4)$$

where $\sigma(\cdot)$denotes the sigmoid activation function, $\tanh(\cdot)$is the hyperbolic tangent activation, and $\odot$represents the element-wise multiplication operator. Moreover, $W$and Urepresent the weight matrices used during network training. In this study, a feed-forward neural network was employed to compute the alignment function, enabling the development of an enhanced GRU model inspired by [14, 15]. Furthermore, two attention gates were introduced to account for temporal intervals and geographical distances between successive check-ins, thereby improving the model's ability to capture contextual dependencies in user mobility patterns.

IV. Description of the Proposed Model

The CEGRU architecture, illustrated in Fig. 4, consists of four main layers: input, embedding, recurrent, and output. The subsequent sections describe each of these layers in detail, along with the parameter learning process employed in the model.

A. CEGRU Layers

The input layer stores the model inputs, which include contextual information (CI) derived from user check-ins. It should be noted that the input to the proposed model consists of user check-in sequences. Each check-in record is represented by a quadruple that can be described:
$c_{u,v,t}$ <u, l, v, t>, including the user identifier, the spatial information of the visited location (i.e., latitude and longitude), the venue or POI identifier, and the associated timestamp. Based on these inputs, the Contextual Information (CI) vector is constructed to embedding spatiotemporal user's check-ins information at each time step. Fig. 3 presents an illustrative example of the feature construction pipeline using sample check-in data from the Gowalla dataset.

[user]	[check-in time]	[latitude]	[longitude]	[location id]
196514	2010-07-24T13:45:06Z	53.3648119	-2.2723465833	145064
196514	2010-07-24T13:44:58Z	53.360511233	-2.276369017	1275991
196514	2010-07-24T13:44:46Z	53.3653895945	-2.2754087046	376497
196514	2010-07-24T13:44:38Z	53.3663709833	-2.2700764333	98503
196514	2010-07-24T13:44:26Z	53.3674087524	-2.2783813477	1043431
196514	2010-07-24T13:44:08Z	53.3675663377	-2.278631763	881734
196514	2010-07-24T13:43:18Z	53.3679640626	-2.2792943689	207763
196514	2010-07-24T13:41:10Z	53.364905	-2.270824	1042822

Fig.3. Feature Construction Pipeline from Sample Gowalla Check-ins

In particular, the CI vector is formed by concatenating four components derived directly from the user's check-ins: (1) a location embedding associated with the venue or POI identifier, (2) a temporal embedding obtained from discretized timestamp features such as time of day and day of week, and (3),(4) a user embedding representing individual mobility preferences including geographical distance (Δg) and time interval (Δt) between two successive check-ins.

Within the recurrent layer of the CEGRU model, this transition CI is emphasized through two specially designed attention gates. As a result, the time interval (Δt) and geographical distance (Δg) between successive check-ins are computed within this layer. Given a user u, a venue v_1, and a timestamp t_τ at timestep τ, the time interval and geographical distance between v_1 and the previously visited venue v_2 at timestep $\tau - 1$ are computed as: $\Delta t_\tau = t_\tau - t_{\tau-1}$ and Δg_τ = dist(lat v_1, lng v_1, lat v_2, lng v_2), respectively (Eq.10). The geographical distance Δg_τ is calculated using the Haversine formula, which computes the angular distance between two points on the surface of a sphere. In this calculation, the latitude serves as the first coordinate and the longitude as the second, both measured in radians. Two data dimensions are required for this computation: latitude and longitude.

$$D(x,y)=2 \arcsin[\sqrt{\sin^2(\frac{x_1-y_1}{2}) + \cos(x_1)\cos(y_1)\sin^2\left(\frac{x_2-y_2}{2}\right)}] \quad (5)$$

Before entering the recurrent layer, the **embedding layer** is used to embed inputs from a sequence of check-ins. In this layer, latent factors for the user**,** POI (venue), and time are generated, denoted as, $\emptyset u_i \in U$, $\emptyset v^{\tau}_i \in V$ and time $\emptyset t^{\tau} \in T$, respectively. The set of parameters for the embedding layer is defined as $\theta_e = \{U, V, T\}$. The venue latent factor $\phi^j_{v_\tau}$, the time latent factor ϕ_{t_τ}, and the contextual transition features (Δg_τ and Δt_τ) are subsequently passed to the recurrent layer for CEGRU training. **In the recurrent layer**, the GRU model is enhanced with two attention gates: the GAG and the TAG, as proposed by Manotumruksa et al. [14] and Kala et al. [15]. These gates control the influence of the previous recurrent unit's hidden state based on the geographical distance and time interval between successive check-ins, reflecting the observation that contextual information impacts users' dynamic preferences differently. The output of this layer is the hidden state of the recurrent unit at timestep τ, denoted as h_τ, and is formally defined as follows:

$$h_\tau = f(\emptyset v^{\tau}_j, \emptyset t^{\tau}, \Delta t_\tau, \Delta g_\tau, \theta_r) \quad (6)$$

The following describes how the traditional GRU is extended to incorporate absolute and CI. To estimate the hidden state h_τ, the update and reset gates of the GRU are employed, taking into account the user's check-in sequence S_u and the user's dynamic preferences at timestep τ:

$$z_\tau = \sigma(W_z \emptyset v^{\tau}_j + U_z h_{\tau-1} + b_z) \quad (7)$$

$$r_\tau = \sigma(W_r \emptyset v^{\tau}_j + U_r h_{\tau-1} + b_r) \quad (8)$$

$$\hat{h}_\tau = \tanh(W \emptyset v^{\tau}_j + U(r_\tau \odot h_{\tau-1}) + b_h) \quad (9)$$

$$h_\tau = (1 - z_\tau) \odot h_{\tau-1} + z_\tau \odot \hat{h}_\tau \quad (10)$$

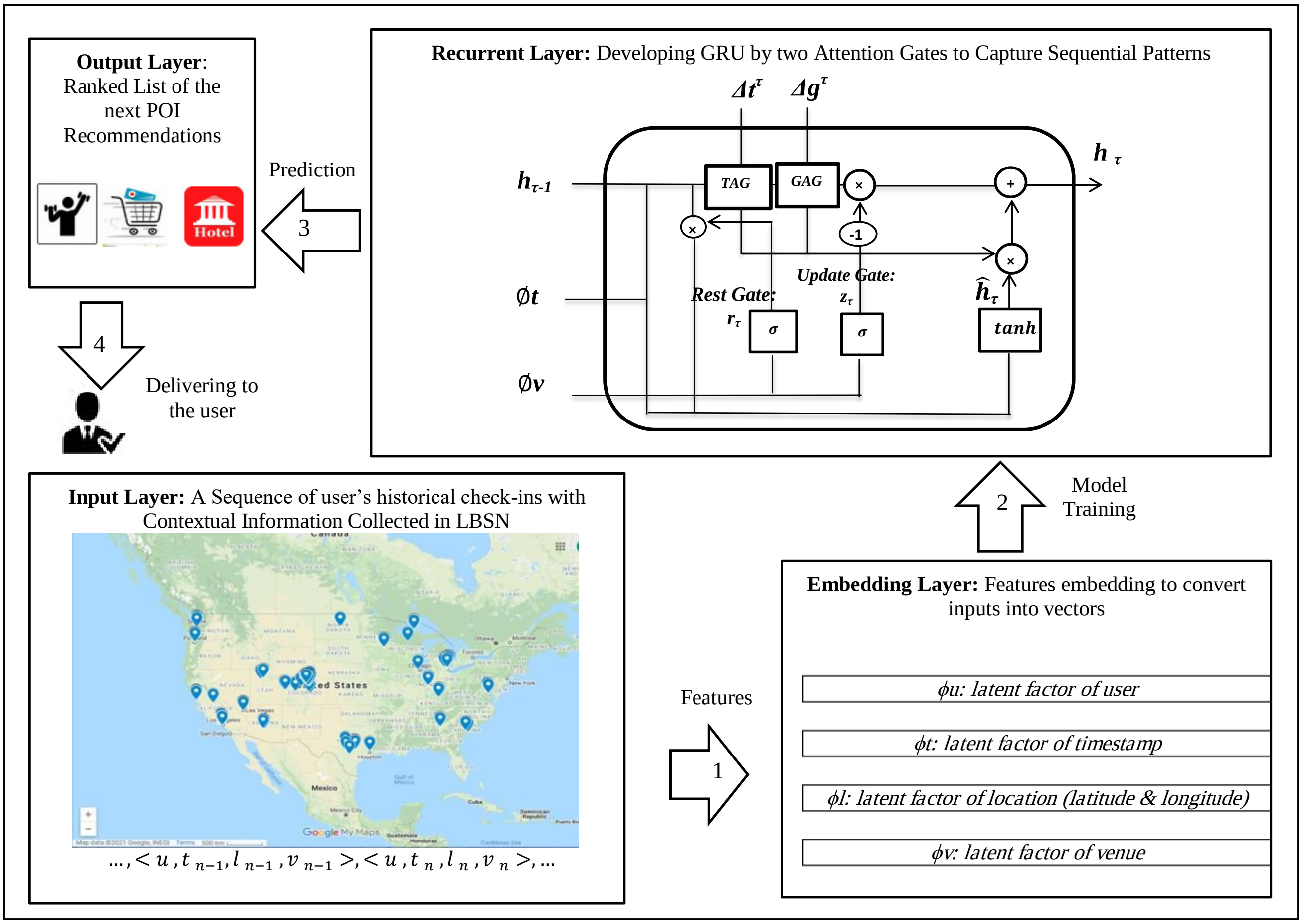

Fig.4. General CEGRU Architecture; historical check-ins are collected in LBSNs and used as input in the proposed model; the proposed CEGRU's output is a ranked POI list that might be interesting to a user in the future based on historical sequences of check-ins at time *t*

In the equations above, $\phi_{v_\tau}^{j}$ denotes the latent factor of the venue v visited by the user at timestep τ. Besides, *tanh() and* $\sigma()$ denote the hyperbolic tangent and sigmoid functions, respectively. Moreover, U implies a recurrent connection weight matrix used to capture sequential signals between every two neighboring hidden states, namely h_τ and $h_{\tau-1}$, by employing $\odot$, indicating the element-wise product. In addition, *W and b* are indicate the transition matrix between the venues' latent factors and corresponding biases, respectively. Furthermore, $\theta_r = \{W, U, b\}$ represents the set of recurrent layer's parameters.

To effectively model users' check-ins in sequential order, the relevant CI must be examined independently. To address this, the proposed GAG and TAG were introduced. The GAG is designed to capture spatial transition dynamics and takes as input the geographical distance between two consecutive check-ins, together with the previous hidden state $h_{\tau-1}$. This enables the model to adaptively regulate the influence of spatial proximity on the hidden state update. Formally, GAG is defined as Eq.(11), where Δg denotes the geographical distance between two consecutive check-ins. The TAG focuses on temporal transition patterns and takes as input the time interval between two consecutive check-ins, together with the previous hidden state $h_{\tau-1}$. By explicitly modeling temporal gaps, this gate allows the network to dynamically adjust the impact of irregular temporal behaviors in user mobility sequences. Formally, the TAG is formulated as Eq.(12) in our manuscript, where Δt represents the time interval between two consecutive check-ins.

$$GAG = \sigma\left(W_{GAG,h}\, h_{\tau-1} + W_{GAG,t}\, \Delta g^{\tau} + b_{GAG}\right) \qquad (11)$$

$$TAG = \sigma\left(W_{TAG,h}\, h_{\tau-1} + W_{TAG,g}\, \Delta t^{\tau} + b_{TAG}\right) \qquad (12)$$

where the attention gates, GAG and TAG, capture the influence of both time intervals and geographical distances between successive check-ins. For instance, if the distance between two check-ins is short, the impact of the previous hidden state $h_{\tau-1}$ is likely to remain significant even when the time interval between them is long. Together, the GAG, TAG, and the reset gate r_τ regulate the contribution of the previous hidden state $h_{\tau-1}$ to the current hidden state h_τ.

The proposed attention gates (GAG and TAG), together with the traditional GRU update and reset gates, are integrated to compute the next hidden state. Accordingly, the conventional GRU equations are modified to incorporate these attention gates, resulting in the CEGRU architecture as follows:

$$z_\tau = \sigma(W_z \emptyset v^\tau_j + U_z h_{\tau-1} + W_z((\phi t^\tau + (GAG \odot \Delta g^\tau) + (TAG \odot \Delta t^\tau)) + b_z) \quad (13)$$

$$r_\tau = \sigma(W_r \emptyset v^\tau_j + U_r h_{\tau-1} + W_r((\phi t^\tau + (GAG \odot \Delta g^\tau) + (TAG \odot \Delta t^\tau)) + b_r) \quad (14)$$

$$\hat{h}_\tau = \tanh(W \emptyset v^\tau_j + U(r_\tau \odot h_{\tau-1}) + W_r((\phi t^\tau + (GAG \odot \Delta g^\tau) + (TAG \odot \Delta t^\tau)) + b_h) \quad (15)$$

The hidden state $h_\tau$is updated according to the steps described above and serves as the output of the recurrent unit at timestep τ. At the same timestamp, the output layer estimates user u's preference for venue $v$as follows:

$$h_\tau = (1 - z_\tau) \odot h_{\tau-1} + z_\tau \odot \tanh(W \emptyset v^\tau_j + U(r_\tau \odot h_{\tau-1}) + W_r((\phi t^\tau + (GAG \odot \Delta g^\tau) + (TAG \odot \Delta t^\tau)) + b_h) \quad (17)$$

At timestamp τ, the output layer estimates user u's preference for venue v as follows:

$$\hat{c}_{u,v,t} = \emptyset u_u h^{\tau T} \quad (18)$$

Previous studies have shown that pairwise loss functions outperform classification-based loss functions in capturing patterns from sequential data and offer more efficient training for recurrent-based recommender systems [14, 15, 41, 16]. Specifically, as noted by Manotumruksa et al. [14, 41], the BPR framework [16] can be used to estimate the parameters of both the recurrent and embedding layers, as well as the probability distribution over all venues, taking into account the hidden state h_τ.

B. Network training

In this study, the datasets consist of sampled triplets, each containing one user and two POIs: one positive (visited) and one negative (unvisited). As previously discussed, the pairwise BPR framework is employed to learn the parameters of both the embedding and recurrent layers ($\Theta = \{\theta_e, \theta_r\}$). BPR leverages the relative ranking of POI pairs, operating under the assumption that a user prefers observed POIs over all unobserved POIs [11, 17]. Within this framework, CEGRU aims to maximize the probability at each sequential position $k$as follows [11, 16, 17]:

$$P(u, t, v > v') = \frac{1}{1+e^{-x}} (o_{u,t,v} - o_{u,t,v'}) \quad (19)$$

where, $v$and v'denote the positive (visited) and negative (unvisited) POIs, respectively. To optimize the network's objective function for next POI recommendation, a regularization term is incorporated into the loss function, as expressed below [16]:

$$J = -\sum_{(v,v')} \ln P(u, t, v > v') + \lambda/2 \, \| \Theta \|^2 \quad (20)$$

where denotes the regularization strength, and Θrepresents the set of model parameters. Following Manotumruksa et al. [14] and Kala et al. [15], the dimensions of the hidden layers $h_\tau$and latent factors d(set to $d = 10$) in the CEGRU architecture were fixed consistently across both datasets. All parameters of the recurrent and embedding layers were initialized randomly using a Gaussian distribution. At the start of training, the batch size and learning rate were set to 256 and 0.001, respectively. Model parameters were optimized using the Adam optimizer. The recurrent layer of CEGRU includes parameter sets U, W, and $b$for the reset gate as well as for the GAG and TAG update gates. The model outputs a score for each POI, reflecting the probability of being the next POI in a user's sequence. A summary of the CEGRU learning algorithm is provided below [23,26]:

Algorithm 1: Training of CEGRU
Input: Set of users Us and set of historical check-in sequences S^u
//construct training instances
Initialize $D = Us_u D^u = \emptyset$ D^u is a set of check-in trajectory samples combined with negative POIs of u
For each user $u \in Us$ **do**
  For each check-in sequence $S^u = \{s_{t1}{}^u, s_{t2}{}^u, \ldots, s_{tn}{}^u\}$ **do**
    Get the set of negative samples v'
    For each check-in activity in S^u **do**
      Compute the embedded vector $v_\tau{}^u$
      Compute the geographical contexts vector $g_\tau{}^u$
      Compute the temporal contexts vector $t_\tau{}^u$
    End for
    Add a training instance $(\{v_\tau{}^u, g_\tau{}^u, t_\tau{}^u\}, \{v'\})$ into D^u
  End for
End for
//train the model
Initialize the parameter set Θ
While (exceed(maximum number of iterations)==FALSE) **do**
  For each user u in U **do**
    Randomly select a batch of instances $D_b{}^u$ from D^u
    Find Θ minimizing the objective (23) with $D_b{}^u$
  End for
End While
Return the set of parameter Θ

V. Experimental Results

To evaluate the effectiveness of the proposed method, empirical experiments were conducted on three publicly available LBSN datasets. These experiments were designed to address the following research questions corresponding to the challenges outlined in Section 1.1: **RQ1**: How can users' check-in CI be effectively utilized in RM? **RQ2**: RQ2: How can the basic GRU architecture be extended to separately consider transition CI associated with check-in sequences for RM? **RQ3**: Does CEGRU, which incorporates multiple types of contextual data via two additional attention gates, improve POI prediction and outperform existing methods?

The experiments were conducted using three publicly available LBSN datasets: Brightkite, Gowalla, and Foursquare. To mitigate data sparsity and the cold-start problem, users with fewer than ten check-ins and POIs with fewer than ten visits were removed from all three datasets, following the approach of Manotumruksa et al. [14]. For data preprocessing, intelligent sampling [27] which considers the intrinsic characteristics of the data rather than relying on simple random sampling was employed. This approach ensured the inclusion of all POIs while maintaining their original frequency distribution and prevented bias toward any particular geographical region. To preserve both geographical and temporal diversity and to reflect the true data distribution, a combination of controlled oversampling and undersampling was applied. Oversampling was used to increase the number of check-ins for users with fewer records, while undersampling was used to reduce the number of check-ins for users with excessively frequent activity. In this study, each check-in record is represented as a quadruple containing a user identifier, check-in timestamp, geographical coordinates, and POI (location ID). Sequential user trajectories were constructed based on each user's check-in history across the three datasets. The data density for each dataset was computed using Equation (21) [42]:

$$Density = (|check - ins|)/(|users| \times |POIs|) \quad (21)$$

In this research, the effect of dataset density on model performance is investigated for the first time. This innovation enables the identification of an optimal density range for selecting appropriate datasets when evaluating the proposed model. To achieve this, a series of experiments was conducted under two experimental states. In each state, the dataset described in [23] was considered the main level, and each dataset was examined at multiple levels of variation. This experimental design provides a novel framework that can guide dataset selection in future studies. In the first experimental state, the number of users was kept constant while the number of check-ins in each dataset was adjusted to 50% and 150% of the main level. Table 1 presents the datasets selected for this state. As shown in the table, dataset density increases nonlinearly as the number of check-ins grows. A summary of the statistics for the three datasets in the first experimental state is provided in Table III..

TABLE III. Statistics of the three datasets in the First State of experiments

Dataset	Level%	#users	#check-ins	#POI	Density
	50%		338361	15054	0.0245
Brightkite	100%	915	676721	7527	0.0928
	150%		921510	3341	0.2635
	50%		307170	10022	0.0293
Gowalla	100%	1047	614340	5011	0.1170
	150%		921510	3341	0.2163
	50%		54098	38490	0.0023
Foursquare	100%	615	108195	19245	0.0091
	150%		162293	12830	0.0206

In the second experimental state, and again considering the dataset described in [23] as the main level, the number of selected users was modified to 50% and 150% of the main level, while keeping the number of check-ins in each dataset constant. Table 2 presents the datasets used in this experimental state. As illustrated in the table, dataset density decreases nonlinearly as the number of users increases. The statistical characteristics of the three datasets in the second experimental state are summarized in Table IV.

TABLE IV. Statistics of the three datasets in the Second State of experiments

Dataset	Level%	#users	#check-ins	#POI	Density
	50%	458		3763	0.3928
Brightkite	100%	915	676721	7527	0.0982
	150%	1373		11290	0.0522
	50%	524		2506	0.4680
Gowalla	100%	1047	614340	5011	0.1170
	150%	1571		7516	0.0521
	50%	308		9623	0.0364
Foursquare	100%	615	108195	19245	0.0091
	150%	923		28868	0.0041

Figure 5 illustrates the impact of varying parameters on dataset density across all experimental datasets. As shown in Fig. 5(a), when the number of check-ins increases while the number of users remains constant, the density increases exponentially. Among the datasets, Gowalla exhibits the highest sensitivity to the increase in check-ins, whereas Foursquare maintains a relatively low density, even with higher numbers of check-ins ndicating a broader spatial distribution of POIs. In contrast, Fig. 5(b) demonstrates that when the number of users increases while the number of check-ins remains constant, the density decreases nonlinearly. The density variations in Gowalla and Brightkite are nearly similar, while Foursquare consistently shows a lower density across all user levels. Following prior research [14, 15], the leave-one-out cross-validation approach was employed to evaluate the performance of the proposed CEGRU architecture. In this method, each user's most recent check-in was used as the test instance, while 100 randomly selected POIs that the user had not previously visited were included as negative samples. These, together, formed the testing set, whereas the remaining check-ins were used as the training set. The CEGRU model's objective was to rank the 100 candidate venues according to their likelihood of being the next visited POI, considering temporal, geographical, and contextual transition information with the actual (ground-truth) check-in expected to rank highest. In line with Manotumruksa et al. [14] and Kala et al. [15], the hidden layer dimension $h_\tau$and latent factor dimension $d = 10$were fixed. As previously noted, the parameters of the recurrent layer were randomly initialized using a Gaussian distribution [28]. The Adam optimizer [43] was utilized for parameter learning, owing to its faster convergence compared to traditional Stochastic Gradient Descent (SGD), which relies on a fixed learning rate across iterations. To further prevent overfitting, the batch size was set to 256, and the dropout rate was adjusted to 0.2.

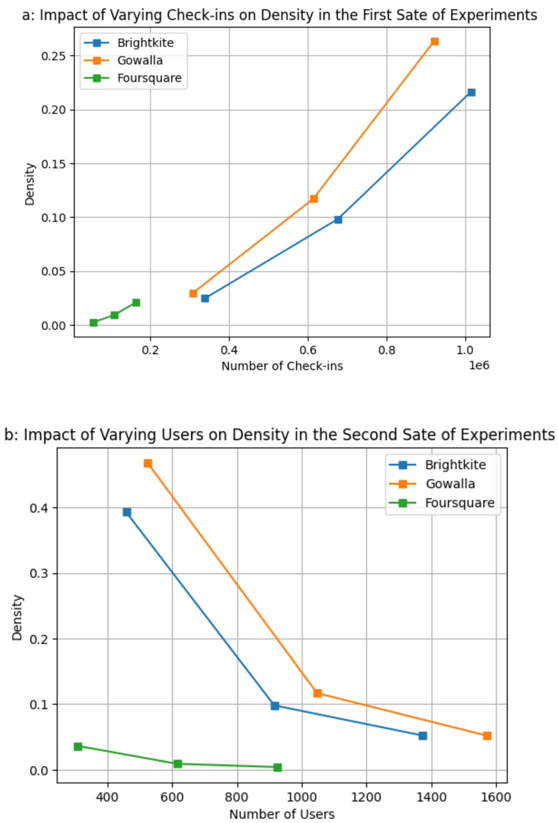

Fig.5. Impact of varying parameters on density Corporation of the Two Experiment States

The experimental procedure was designed and conducted in two distinct states to analyze the impact of dataset density on model performancep: **First state:** the number of check-ins was increased while keeping the number of users constant. **Second state:** the number of users was increased while keeping the number of check-ins constant. **First state**: While keeping the number of users constant, the number of check-ins in each dataset was varied to 50% and 150% of the main level. This manipulation aimed to examine how changes in check-in volume influence model performance. The training and testing accuracy@10 and loss@10 for the Brightkite, Gowalla, and Foursquare datasets across different epochs in the CEGRU model are illustrated in Fig. 6. **Second state**: In this experiment, the number of selected users was varied to 50% and 150% of the main level, while keeping the number of check-ins in each dataset constant. This setup aimed to analyze how user population size affects the model's performance. The training and testing accuracy@10 and loss@10 for the Brightkite, Gowalla, and Foursquare datasets across epochs in the CEGRU model for the second experimental state are presented in Fig. 7. The results comparing the two experimental states are presented in TABLE V. It should be noted that this table is provided to evaluate the effect of dataset density at different levels and to identify the optimal density for model performance. The optimal density of a dataset refers to the range of density values at which the model achieves its highest prediction accuracy, indicating the most effective balance between data sparsity and redundancy.

In recommender systems, low data density indicates that limited information is available for learning user preferences, which consequently reduces prediction accuracy. Conversely, excessively high data density introduces redundant information, resulting in only marginal improvements in accuracy. Therefore, identifying the optimal density, a concept first introduced in the pioneering experiments of this study, represents finding a "golden point" of data density. Beyond this range, prediction accuracy either declines due to data sparsity or plateaus because of redundant information. As described in Section V of the original manuscript, in the first experimental state, and using the dataset from [23] as the main reference, the number of check-ins in each dataset was varied to 50% and 150% of the main level, while keeping the number of users constant. In the second experimental state, again considering the [23] dataset as the main reference, the number of selected users was adjusted to 50% and 150% of the main level, while maintaining a constant number of check-ins in each dataset.

TABLE V. The results of the comparison of two states of experiments

Sate	**Dataset**	**Level**	**Density**	**Acc @10**	**Loss @10**
First	**Brightkite**	50%	0.0245	0.758	0.259
		100%	0.0928	0.798	0.196
		150%	0.2635	**0.834**	0.134
	Gowalla	50%	0.0293	0.727	0.238
		100%	0.1170	0.741	0.199
		150%	0.2163	**0.796**	0.120
	Foursquare	50%	0.0023	0.674	0.273
		100%	0.0091	0.724	0.177
		150%	0.0206	**0.760**	0.128
Second	**Brightkite**	50%	0.3928	**0.824**	0.127
		100%	0.0982	0.800	0.180
		150%	0.0522	0.765	0.254
	Gowalla	50%	0.4680	**0.791**	0.139
		100%	0.1170	0.775	0.177
		150%	0.0521	0.709	0.251
	Foursquare	50%	0.0364	**0.786**	0.142
		100%	0.0091	0.693	0.183
		150%	0.0041	0.689	0.251

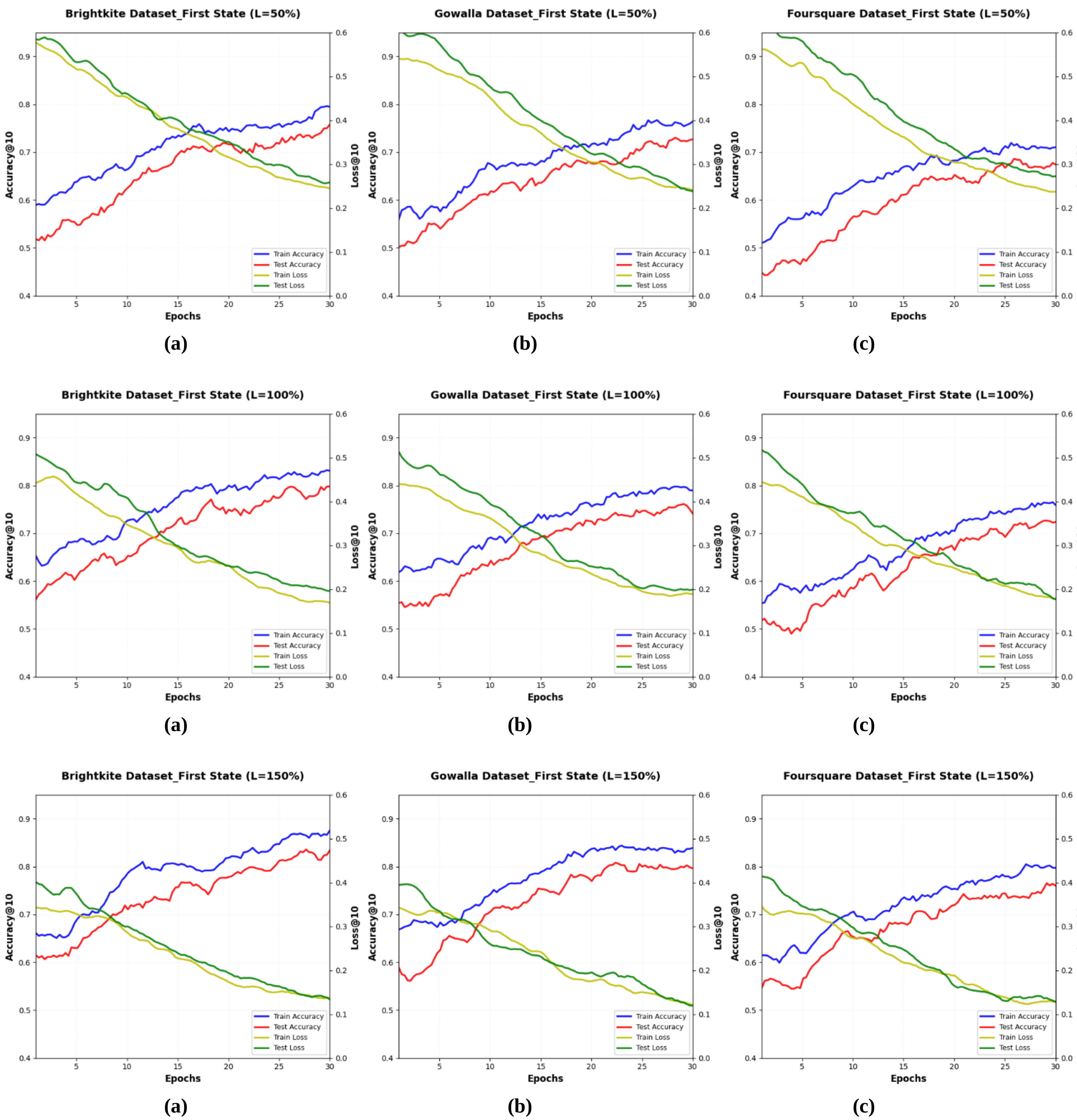

Fig.6. Training and testing accuracy @10 and loss @10 for three datasets vs. epochs in the first sate of experiments and in the three levels (%L). (a) Brightkite Dataset; (b) Gowalla Dataset; (c) foursquare dataset.

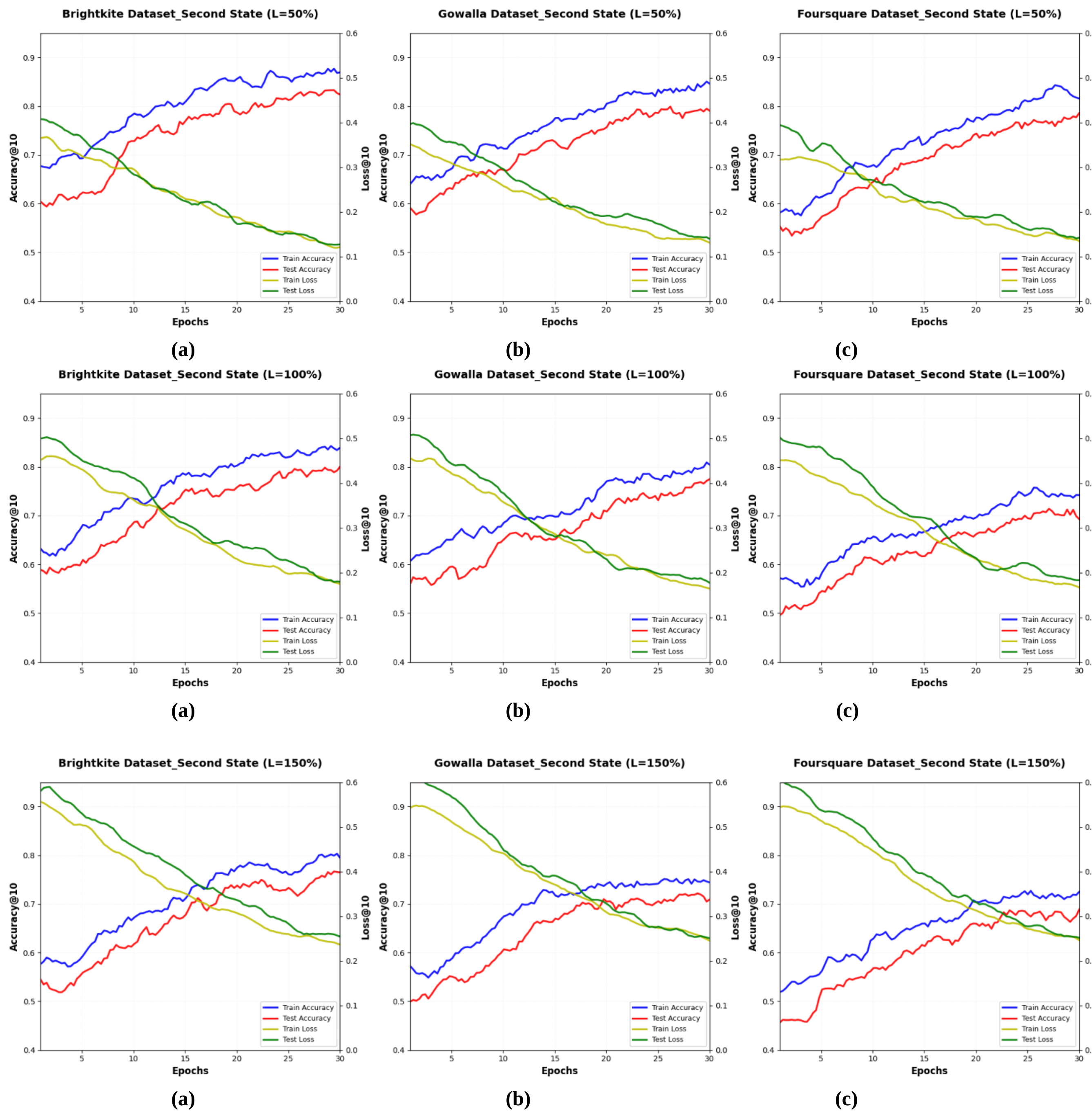

Fig.7. Training and testing accuracy @10 and loss @10 for three datasets vs. epochs in the second sate of experiments and in the three levels (%L). (a) Brightkite Dataset; (b) Gowalla Dataset; (c) foursquare dataset.

The comparison of the two experimental states is illustrated in Fig. 8. From the results, it is observed that in the first experimental state, across all three datasets, the highest accuracy was achieved at the 150% check-in level. In other words, increasing dataset density by raising the number of user check-ins while keeping the number of users constant, enhances the performance of the proposed CEGRU model.

By analyzing the results from the second experimental state, it is observed that across all three datasets, the highest accuracy was achieved at the 50% user level. In other words, increasing dataset density by reducing the number of users while keeping the number of check-ins constant, improves the performance of the proposed CEGRU model. Based on the findings from both experimental states, the optimal density range for each dataset in the context of the proposed architecture is summarized in TABLE VI:

TABLE VI. The optimal density range in each of the datasets for the CEGRU architecture

Dataset	Frist State	Second State	Optimized Density
Brightkite	0.2635	0.3928	(0.2635,0.3928)
Gowalla	0.2163	0.4680	(0.2163,0.4680)
Foursqure	0.0206	0.0364	(0.0206,0.0364)

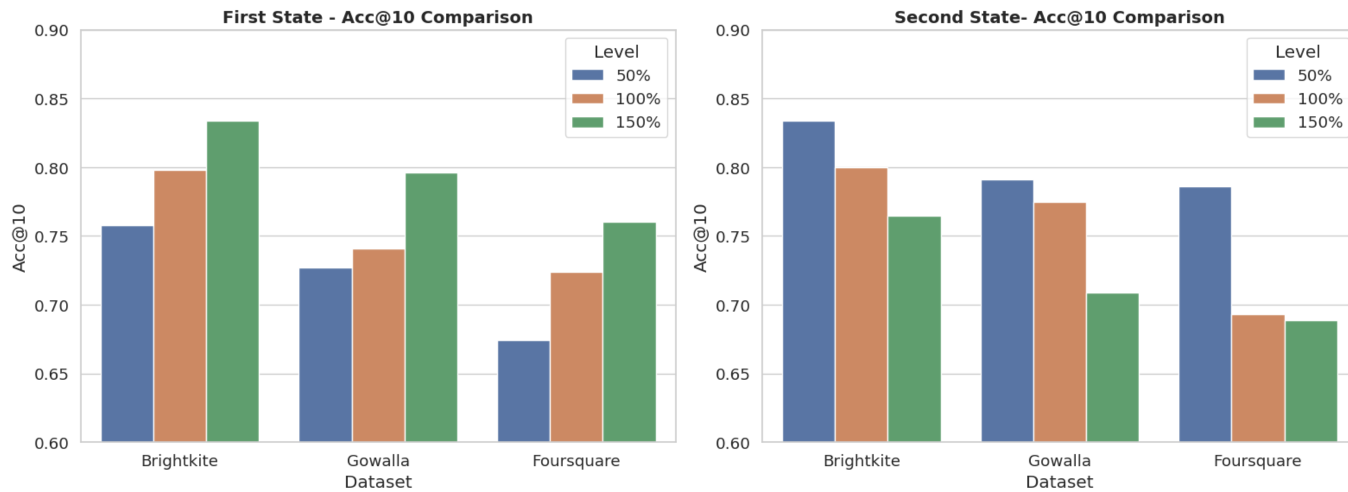

Fig 8. Comparison of two states of experiments

A. Comparison Methods

In this section, the performance of the proposed CEGRU model in the next POI recommendation task is evaluated by comparing it with six state of the art methods. Table VII summarizes these methods into different aspects. Based on data, they are categorized into RNN-, and AM-based approaches. The compared models are also classified according to the use of GCI and TCI.

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

Methods	**Approaches and Contextual Information**			
	RNN	**AM**	**GCI**	**TCI**
STGN [30]	√	×	√	√
LLRec [38]	√	√	√	√
Flashback [40]	√	×	√	√
GeoSAN [43]	×	√	√	√
DRCF [44]	√	×	×	×
CARA [14]	√	√	√	√
CEGRU	√	√	√	√

Note. RNN: Recurrent neural network; AM: Attention mechanism; GCI: Geographical contextual information; TCI: Temporal contextual information; STGN: Spatiotemporal gated network; LLRec: Light Location Recommender System; Flashback; DRCF: Deep Recurrent Collaborative Filtering; CARA: Contextual attention recurrent architecture; CEGRU: Contextual extended gated recurrent unit.

A brief description of these models is given below:

STGN: The Spatiotemporal Gated Network (STGN), proposed by Zhao et al. [30], enhances the traditional LSTM model by incorporating spatiotemporal gates (STGs) to capture the spatiotemporal relationships between successive check-ins. By introducing additional gates and memory cells to model both short- and long-term user preferences, STGN effectively extends the basic LSTM architecture for improved next POI prediction.

LLRec: Wang et al. [38] proposed LLRec, a model designed to capture long-term and short-term user preferences, as well as the textual features of POIs and the complex dependencies among user preferences. The model leverages embedding layers, recurrent components, and an attention mechanism to effectively integrate these factors for improved next POI recommendation.

Flashback: Yang et al. [40] proposed Flashback, a model designed to handle sparse user mobility data by performing flashbacks on the hidden states of RNNs. The model computes a weighted average of historical hidden states to more effectively capture spatiotemporal dependencies and improve the accuracy of next POI prediction.

GeoSAN: Lian et al. [43] proposed GeoSAN, which addresses the data sparsity problem by introducing a novel loss function. The model represents the hierarchical gridding of each GPS point and employs a self-attention-based geography encoder to effectively capture and utilize geographical information for improved next POI prediction.

DRCF: Manotumruksa et al. [44] proposed DRCF, which extends NeuMF to leverage traditional RNNs for modeling the sequential order of users' check-ins. The model consists of two components, each containing its own recurrent layer, to capture both user preferences and sequential dependencies for next POI recommendation.

CARA: CARA: Manotumruksa et al. [14] proposed the Contextual Attention Recurrent Architecture (CARA), a model designed to capture users' dynamic preferences by integrating feedback sequences with the CI associated with those sequences. This approach enables a more accurate modeling of user behavior in next POI recommendation tasks.

To evaluate the performance of the aforementioned methods, we used the recall metric (Acc@k, k = 10), which measures whether the ground-truth POI appears in the top-k recommended list. The general definition of

Acc@k is provided in Eq. (22) [3].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (22)$$

In the above equations, the terms TP, FP, TN, and FN are defined as follows:

- TP or true positive detection (in reality it is correct and correctly predicted)
- FP or false positive detection (in reality it is incorrect but incorrectly predicted correctly)
- TN or true negative detection (in reality it is incorrect and correctly predicted incorrectly)
- FN or false negative detection (in reality it is correct but incorrectly predicted incorrectly)

Generalizing the above definitions for evaluating user location prediction models in recommender systems, we obtain:

- TP, the number of predicted next POI that is actually present in the user's location recording sequence.
- FP, the number of predicted next POI that is actually not present in the user's location recording sequence.
- TN. the number of unpredicted next POI that is actually not present in the user's location recording sequence.
- FN. the number of unpredicted next POI that is actually present in the user's location recording sequence.

Since the output of the proposed CEGRU architecture is a ranking model, the evaluation criteria from Eq. (22) can be generalized to assess the model's performance in top-K prediction recommendations as follows:

$$Accuracy@k = \frac{number\ of\ samples\ correctly\ predicted}{total\ number\ of\ samples} \quad (22)$$

B. Results and Discussion

TABLE VIII presents a comparison of the recommendation performance of six methods across the three datasets. Bolded numbers in each column indicate the best-performing results. The results of the two experimental states are denoted as CEGRU-1 and CEGRU-2, corresponding to increasing the number of check-ins while keeping the number of users constant and increasing the number of users while keeping the number of check-ins constant, respectively.

TABLE VIII. A comparison between different methods for recommendation performance

Methods	Acc@10		
	Brightkite	**Gowalla**	**Foursquare**
STGN	0.2020	0.5231	0.3017
LLRec	-	0.3874	0. 3542
Flashback	-	0.3472	0.6236
GeoSAN	0.6425	0.6028	0.4867
DRCF	0.7363	-	0.8805
CARA	0.7385	-	0.8851
CEGRU-1	**0.8340**	**0.7960**	0.7600
CEGRU-2	0.8240	0.7910	0.7860

The experimental results indicate that methods that do not employ recurrent models and do not account for the temporal and geographical contextual information of users' movement trajectories separately exhibit reduced prediction accuracy in recommender systems.

Although the STGN model separately incorporates geographical contextual information (GCI) and temporal contextual information (TCI), it does not employ an attention mechanism to adaptively weight these contextual factors. Moreover, its reliance on an LSTM-based architecture limits its ability to capture fine-grained sequential dynamics, resulting in lower prediction accuracy compared to GRU-based models such as CARA and the proposed CEGRU. The LLRec model accounts for both long-term and short-term user preferences through embedding representations, a recurrent component based on an RNN architecture, and an attention mechanism. However, it does not explicitly model temporal intervals and geographical distances as independent contextual signals within users' movement trajectories, which constrains its effectiveness in handling irregular spatiotemporal patterns. In addition, models built upon traditional RNN architectures often suffer from degraded performance when modeling long user check-in sequences due to vanishing gradient issues, limiting their ability to capture long-range dependencies. Similarly, the Flashback model leverages an RNN architecture and aggregates historical hidden states using a weighted mechanism to model spatiotemporal effects. Nevertheless, like LLRec, it lacks explicit modeling of spatial distances and temporal gaps between consecutive check-ins, leading to inferior performance compared to GRU-based approaches. GeoSAN employs a self-attention mechanism for POI recommendation and incorporates geographical–temporal contextual information (GTCI). Despite this, its performance remains lower than that of DRCF and CARA, as it primarily focuses on modeling spatial relationships among locations and does not sufficiently capture sequential transition dynamics in user trajectories. The DRCF model improves upon STGN by employing a recurrent architecture to model sequences of previously visited venues, thereby achieving higher prediction accuracy. However, it does not explicitly incorporate contextual information associated with individual check-ins, such as temporal intervals and geographical distances. This observation highlights that a strong sequential modeling architecture alone is insufficient; explicit integration of rich spatiotemporal contextual information is essential for accurate next POI recommendation. The CARA model further improves prediction performance by separately incorporating TCI and GCI and by jointly leveraging recurrent and attention mechanisms. Although this model demonstrates superior performance compared to other models, it incurs additional computational overhead due to the incorporation of a time-stamp attention gate. Building upon this idea, the proposed CEGRU model introduces a novel architecture that embeds two attention gates within the GRU framework to more effectively capture spatiotemporal transition dynamics. Specifically, the Temporal Attention Gate (TAG) and the Geographical Attention Gate (GAG) explicitly model time intervals and geographical distances between successive check-ins. The outputs of these gates independently influence the reset and update gates of the GRU, enabling CEGRU to adaptively handle irregular user mobility patterns and thereby achieve superior predictive performance. Overall, due to the adoption of the GRU architecture and the explicit modeling of heterogeneous spatiotemporal contextual factors through dual attention gates, the proposed CEGRU model consistently outperforms models based on traditional RNNs (e.g., LLRec, Flashback, and DRCF), as well as approaches relying primarily on recurrent architectures without adaptive contextual weighting (e.g., STGN) or attention mechanisms alone (e.g., GeoSAN). As shown in Fig. 9, experimental results demonstrate that this design improves prediction accuracy, confirming the effectiveness of the CEGRU architecture.

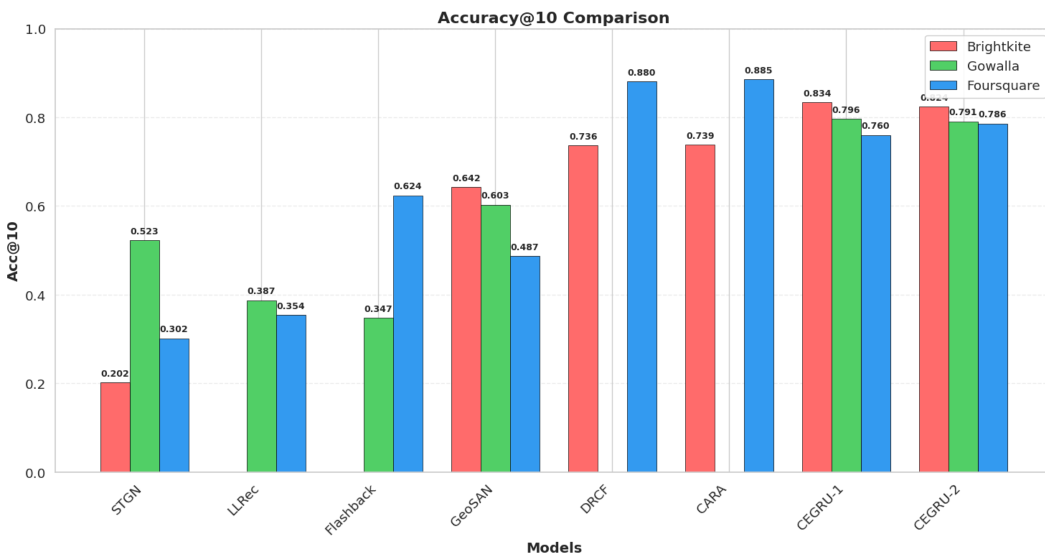

Fig. 9. A comparison between CEGRU and the baseline methods for Accuracy@10 on Brightkite, Gowalla and Foursquare datasets.

In response to RQ1, it can be stated that location prediction in LBSNs is a key application of RM. The CI of users' movement trajectories in LBSNs, including temporal and geographical data, provides valuable input for predicting movement patterns. By collecting and feeding this information into deep recurrent neural network models, it can be effectively utilized for RM tasks. To address RQ2, two attention gates were implemented as a feed-forward network to extend the GRU model. The outputs of these gates influence the GRU's reset and update gates, controlling the impact of users' GTCI in modeling their trajectory data. Regarding RQ3, the effectiveness of the proposed CEGRU model was evaluated by comparing its Accuracy@10 in the first and second experimental states against existing architectures, as shown in Fig. 10. The results indicate that CEGRU achieves higher accuracy than competing models on the Brightkite and Gowalla datasets. In the Foursquare dataset, however, due to its low density and high POI diversity, CEGRU's accuracy is 11.2% lower than CARA and 10.7% lower than DRCF, exhibiting a different behavior. Overall, the CEGRU model demonstrates an average improvement of 64.6% compared to the baseline methods.

From a computational perspective, the proposed CEGRU model maintains a complexity comparable to standard GRU and LSTM architectures. Similar to GRU, the dominant computational cost per time step arises from matrix multiplications associated with hidden state updates, resulting in a time complexity of $O(d_h \cdot d_i + d_h^2)$, where d_h and d_i denote the hidden and input dimensions, respectively[45]. CEGRU introduces two lightweight attention gates that operate on scalar spatiotemporal transition features, namely the geographical distance and time interval between consecutive check-ins. The additional computations introduced by these gates are linear with respect to the hidden dimension and therefore add only a modest constant factor overhead without changing the asymptotic complexity. As a result, CEGRU achieves improved predictive performance while preserving computational efficiency comparable to GRU/LSTM based models .

From a Reality Mining (RM) perspective, the proposed CEGRU model serves as a computational framework for extracting latent spatiotemporal behavioral patterns from real world human mobility data. By modeling temporal intervals and geographical distances through dual attention gates embedded within a GRU architecture, CEGRU effectively captures irregular and personalized mobility behaviors reflected in user check-in sequences.

It is important to note that prior RM studies typically focus on sensor based or communication data and adopt different experimental settings and evaluation metrics. In contrast, this work represents one of the first attempts to apply recurrent neural architectures to Reality Mining using large scale LBSN datasets. Despite these differences, the experimental results validate that the proposed model successfully mines meaningful behavioral regularities, thereby reinforcing the central role of Reality Mining in this study.

VI.Conclusion

RM seeks to uncover predictable behavioral patterns by collecting and analyzing machine-sensed ambient data related to human social interactions. With the increasing availability of large scale LBSN data, user check-ins has emerged as an important source of real world behavioral traces that reflect human mobility and activity patterns. In recent years, next location prediction has become increasingly important for a wide range of LBSN applications. GTCI plays a critical role in assessing individual activities for personalized POI recommendation.

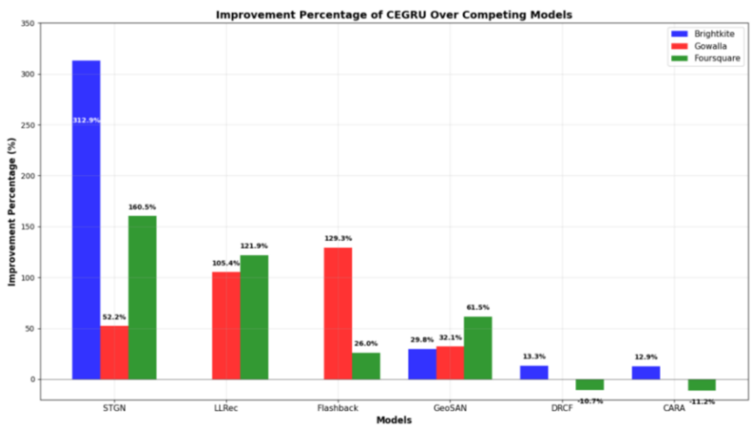

Fig. 10. Percentage of Improvement of CEGRU

Despite the relevance of contextual information obtained from users' trajectory data in LBSNs, such information has not been fully explored within the Reality Mining framework. To address the limitations of previous studies, this research proposed a novel CEGRU model for location prediction in Reality Mining using check-in data from LBSNs. The proposed architecture extends the standard GRU by separately incorporating geographical and temporal contextual information extracted from user check-ins. Inspired by attention mechanisms, two additional contextual attention gates are introduced to explicitly emphasize the impact of temporal intervals and geographical distances when modeling sequential user behavior. POIs are ranked for recommendation based on users' prior check-ins. In particular, the explicit modeling of heterogeneous spatiotemporal contextual factors for Reality Mining applications constitutes a key novelty of the proposed CEGRU architecture. The value of independently evaluating contextual information is evident from the comparison with baseline techniques. The proposed CEGRU architecture, enhanced with two contextual attention gates, demonstrated superior performance in next-location prediction and POI recommendation tasks. Extensive experiments on three large-scale LBSN datasets, including Gowalla, Brightkite, and Foursquare, showed that CEGRU consistently outperformed contemporary recurrent and attention-based models. Moreover, this study is among the first to systematically investigate the impact of dataset density on model performance in a RM setting by conducting experiments under two different density states. These experiments identify an optimal density range for each dataset, providing practical guidance for future Reality Mining research when selecting datasets for model evaluation. From a broader perspective, this work demonstrates that recurrent neural architectures equipped with explicit spatiotemporal contextual attention mechanisms provide an effective framework for mining latent human mobility patterns from real world social sensing data. In future work, the CEGRU architecture could be extended to incorporate social relationships among users in LBSNs, enabling more comprehensive RM analyses. Additionally, richer contextual information, such as textual and visual data from check-ins or environmental factors like weather, could be integrated to further enhance prediction accuracy and broaden the applicability of RM research.

REFERENCES

[1] Fan X, Guo L, Han N, Wang Y, Shi J, Yuan Y (2018) A Deep Learning Approach for Next Location Prediction. In: Proceedings of 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD).

10.1109/CSCWD.2018.8465289
[2] Baral R, Li T, Zhu X (2018) CAPS: Context Aware Personalized POI Sequence Recommender System. https://doi.org/10.48550/arXiv.1803.01245
[3] Liu C, Liu J, Wang J, Xu S, Han H, Chen Y (2019) An Attention-Based Spatiotemporal Gated Recurrent Unit Network for Point-of-Interest Recommendation. ISPRS International Journal of Geo-Information 8(8): 355. https://doi.org/10.3390/ijgi8080355
[4] P (2015) Reality Mining: Data Mining or Something else. In: Proceedings of 10 th International Workshop on Knowledge Management (IWKM)
[5] Chen Y, Crespi N, Ortiz AM, Shu L (2017) Reality Mining: A Prediction Algorithm for Disease Dynamics Based on Mobile Big Data. Information Sciences 379: 82-93. https://doi.org/10.1016/j.ins.2016.07.075
[6] Eagle N, Pentland AS (2007) Eigenbehaviors: Identifying Structure in Routine. Behavioral ecology and sociobiology 63(7): 1057-1066. https://doi.org/10.1007/s00265-009-0739-0
[7] Eagle N, Pentland AS (2006) Reality mining: Sensing Complex Social Systems. Personal and Ubiquitous Computing 10(4): 255-268. https://doi.org/10.1007/s00779-005-0046-3
[8] Asri H, Mousannif H, Moatassime HA (2019) Reality Mining and Predictive Analytics for Building Smart Applications. Journal of Big Data 6(1): 1-25. https://doi.org/10.1186/s40537-019-0227-y
[9] Farrahi K, Gatica-Perez D (2010) Probabilistic Mining of Socio-Geographic Routines. IEEE Journal of Selected Topics in Signal Processing 4: 746-755. 10.1109/JSTSP.2010.2049513
[10] Jung J, Lee H, Chohi KS, (2009) Contextualized Recommendation Based on Reality Mining from Mobile Subscribers. Cybernetics and Systems: An International Journal 40(2): 160-175. https://doi.org/10.1080/01969720802634089
[11] Asraf A, Islam M, Haque M (2020) Deep Learning Applications to Combat Novel Coronavirus (COVID-19) Pandemic. SN Computer Science 1(6): 1-7. https://doi.org/10.1007/s42979-020-00383-w
[12] Feng J, Li Y, Zhang C, Sun F, Meng F, Guo A, Jin D (2018) DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In: Proceedings of International World Wide Web Conference Committee-IW3C2. https://doi.org/10.1145/3178876.3186058
[13] Chen Y, Long C, Cong G, Li C (2020) Context-aware Deep Model for Joint Mobility and Time Prediction. In: Proceedings of the 13th International Conference on Web Search and Data Mining. https://doi.org/10.1145/3336191.3371837
[14] Manotumruksa J, Macdonald C, Ounis I (2018) A Contextual Attention Recurrent Architecture for Context-Aware Venue Recommendation. In: Proceedings of the 41st international ACM SIGIR conference on research & development in information retrieval. https://doi.org/10.1145/3209978.3210042
[15] Kala KU, Nandhini M (2019) Context Category Specific sequence aware Point Of Interest Recommender System with Multi Gated Recurrent Unit. Journal of Ambient Intelligence and Humanized Computing. https://doi.org/10.1007/s12652-019-01583-w
[16] Rendle S, Freudenthaler C, Gantner Z, Schmidt Thieme L (2009) BPR: Bayesian Personalized Ranking from Implicit Feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. https://doi.org/10.48550/arXiv.1205.2618
[17] Huang L, Ma Y, Wang S, Liu Y (2019) An Attention-based Spatiotemporal LSTM Network for Next POI Recommendation. IEEE Transactions on Services Computing 14(6): 1585-1597. 10.1109/TSC.2019.2918310
[18] Ferrari L, Mamei M (2013) Classification and Prediction of Whereabouts Patterns from the Reality Mining Dataset. Pervasive and Mobile Computing 9: 516–527. https://doi.org/10.1016/j.pmcj.2012.04.002
[19] Yao D, Zhang C, Huang J, Bi J (2017) SERM: A Recurrent Model for Next Location Prediction in Semantic Trajectories. In: Proceedings of the ACM on Conference on Information and Knowledge Management. https://doi.org/10.1145/3132847.3133056
[20] Cho E, Myers SA, Leskovec J (2011) Friendship and Mobility: User Movement in Location-based Social Networks. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data. https://doi.org/10.1145/2020408.2020579
[21] Yang D, Zhang D, Zheng VW, Yu Z (2014) Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. IEEE Transactions on Systems, Man, and Cybernetics: Systems 45(1): 129-142. 10.1109/TSMC.2014.2327053
[22] Choujaa D, Dulay N (2010) Predicting Human Behaviour from Selected Mobile Phone Data Points. In: Proceedings of 12th ACM international conference on Ubiquitous computing. https://doi.org/10.1145/1864349.1864368
[23] Ghanaati, F., Ekbatanifard, G., & Khoshhal Roudposhti, K. (2023). Developing A Contextual Combinational Approach for Predictive Analysis of Users Mobile Phone Trajectory Data in LBSNs. Journal of Information Systems and Telecommunication (JIST), 1(41), 12. 10.52547/jist.32593.11.41.12.
[24] Xia H, Luo Y, Liu Y (2021) Attention neural collaboration filtering based on GRU for recommender systems. Complex & Intelligent Systems 7(3):1367-79. https://doi.org/10.1007/s40747-021-00274-4
[25] Salehi soleymanabadi, A., Khamforoosh., k, & Maihami., V. (2025). An Enhanced Framework for Efficient Point-of-Interest Recommendations in Recommender Systems. Journal of Intelligent Multimedia Processing and Communication Systems (IMPCS), 2(6), 10.71856/IMPCS.2025.1208219.
[26] Ghanaati, F., Ekbatanifard, G., & Khoshhal Roudposhti, K. (2023). Using a Flexible Model to Compare the Efficacy of Geographical and Temporal Contextual Information of Location-Based Social Network Data for Location Prediction. ISPRS International Journal of Geo-Information, 12(4), 137. https://doi.org/10.3390/ijgi12040137.
[27] Kamath, C. (2022). Intelligent sampling for surrogate modeling, hyperparameter optimization, and data analysis. Machine Learning with Applications, 9, 100373. https://doi.org/10.1016/j.mlwa.2022.100373
[28] He X, Liao L, Zhang H, Nie L, Hu X, Chua T (2017) Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web. https://doi.org/10.1145/3038912.3052569
[29] Liu Q, Wu S, Wang L, Tan T (2016) Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. In: Proceedings of AAAI Conference on Artificial Intelligence
[30] Zhao P, Zhu H, Liu Y, Xu J, Li Z, F. Zhuang F, Sheng V.S, Zhou X (2020) Where to Go Next: A Spatio-temporal Gated Network for Next POI Recommendation. In: Proceedings of Thirty-Third AAAI Conference on Artificial Intelligence. 10.1109/TKDE.2020.3007194
[31] Kumar S, Nezhurina M (2018) An ensemble classification approach for prediction of user's next location based on Twitter data. Journal of Ambient Intelligence and

Humanized Computing 10(11): 4503-4513. https://doi.org/10.1007/s12652-018-1134-3
[32] Zhang L, Sun Z, Zhang J, Lei Y, Li C, Wu Z, Kloeden H, Klanner F (2020) An Interactive Multi-Task Learning Framework for Next POI Recommendation with Uncertain Check-ins. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)
[33] Liu Q, Wu S, Wang D, Li Z, Wang L (2016) Context-aware Sequential Recommendation. In: Proceedings of IEEE 16th International Conference on Data Mining (ICDM). 10.1109/ICDM.2016.0135
[34] Gao Q, Zhou F, Trajcevski G, Zhang K, Zhong T, Zhang F (2019) Predicting Human Mobility via Variational Attention. In: proceedings of The World Wide Web Conference. https://doi.org/10.1145/3308558.3313610
[35] Zheng C, Tao D (2020) Attention-Based Dynamic Preference Model for Next Point-of-Interest Recommendation. In: Proceedings od International Conference on Wireless Algorithms, Systems, and Applications. https://doi.org/10.1007/978-3-030-59016-1_63
[36] Vaswani A, Parmar N, Uszkoreit J, Jones L, Gomez A.N, Kaiser L, Polosukhin I (2017) Attention Is All You Need. In: Proceedings of 31st Conference on Neural Information Processing Systems (NIPS 2017)
[37] Huang L, Ma Y, Liu Y, He K (2020) DAN-SNR: A Deep Attentive Network for Social-Aware Next Point-of-Interest Recommendation. ACM Transactions on Internet Technology 21(1): 1-27
[38] Wang, Q., Yin, H., Chen, T., Huang, Z., Wang, H., Zhao, Y., & Viet Hung, N. Q. (2020, April). Next Point-of-Interest Recommendation on Resource-Constrained Mobile Devices. In Proceedings of the Web Conference 2020 (pp. 906-916). https://doi.org/10.1145/3366423.338017.
[39] Semwa VB, Gupta A, Lalwani P (2021) An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition. The Journal of Supercomputing 77(11)12256-12279. https://doi.org/10.1007/s11227-021-03768-7
[40] Yang, D., Fankhauser, B., Rosso, P., & Cudre-Mauroux, P. (2020). Location prediction over sparse user mobility traces using rnns. In Proceedings of the Twenty-ninth International Joint Conference on Artificial Intelligence (pp. 2184-2190). 10.24963/ijcai.2020/302
[41] Wang X, Zhao Y, Pourpanah F (2020) Recent Advances in Deep Learning. International Journal of Machine Learning and Cybernetics 11:747–750. https://doi.org/10.1007/s13042-020-01096-5
[42] Yang K, Zhu J (2019) Next POI Recommendation via Graph Embedding Representation From H-Deepwalk on Hybrid Network. IEEE Access 7: 171105-171113. 10.1109/ACCESS.2019.2956138
[43] 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. https://doi.org/10.1145/3394486.3403252
[44] Manotumruksa, J., Macdonald, C., & Ounis, I. (2017, November). A deep recurrent collaborative filtering framework for venue recommendation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (pp. 1429-1438). https://doi.org/10.1145/3132847.3133036.
[45] Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., Muneer, A., Sumiea, E. H., Alqushaibi, A., & Ragab, M. G. (2024). RNN-LSTM: From applications to modeling techniques and beyond-Systematic review. Journal of King Saud University-Computer and Information Sciences, 36(5), 102068. https://doi.org/10.1016/j.jksuci.2024.102068

Appendix A

1-Realituy Mining (RM)
2-Location Based Social Network (LBSN)
3- Contextual Information (CI)
4- Point of Interest (POI)
5- Contextual Extended Gated Recurrent Unit (CEGRU)
6- Attention Mechanism (AM)
7- Collaboration Filtering (CF)
8- Geographical Contextual Attention Gate (GAG)
9- Temporal Contextual Attention Gate (TAG)
10- Machine Learning (ML)
11-Deep Learning (DL)
12- Recurrent Neural Network (RNN)
13- Global Positioning System (GPS)