CARAN: A Context-Aware Recency-Based Attention Network for Point-of-Interest Recommendation

MD. BILLAL HOSSAIN (Member, IEEE), MOHAMMAD SHAMSUL AREFIN (Senior Member, IEEE), IQBAL H. SARKER (Member, IEEE), MD. KOWSHER (Member, IEEE), PRANAB KUMAR DHAR (Member, IEEE), AND TAKESHI KOSHIBA (Member, IEEE)

1Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram 4349, Bangladesh
2Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ 07030, USA
3Faculty of Education and Integrated Arts and Sciences, Waseda University, Shinjuku-ku, Tokyo 169-8050, Japan

Corresponding author: Mohammad Shamsul Arefin (sarefin@cuet.ac.bd)

ABSTRACT
Point-of-interest (POI) recommendation system that tries to anticipate user’s next visiting location has attracted a plentiful research interest due to its ability in generating personalized suggestions. Since user’s historical check-ins are sequential in nature, Recurrent neural network (RNN) based models with context embedding shows promising result for modeling user’s mobility. However, such models cannot provide a correlation between non-consecutive and non-adjacent visits for understanding user’s behavior. To mitigate data sparsity problem, many models use hierarchical gridding of the map which cannot represent spatial distance smoothly. Another important factor while providing POI recommendation is the impact of weather conditions which has rarely been considered in the literature. To address the above shortcomings, we propose a Context-Aware Recency based Attention Network (CARAN) that incorporates weather conditions with spatiotemporal context and gives focus on recently visited locations using the attention mechanism. It allows interaction between non-adjacent check-ins by using spatiotemporal matrices and uses linear interpolation for smooth representation of spatial distance. Moreover, we use positional encoding of the check-in sequence in order to maintain relative position of the visited locations. We evaluate our proposed model on three real world datasets and the result shows that CARAN surpasses the existing state-of-the-art models by 7-14%.

INDEX TERMS
Attention model, location based services, point-of-interest, recommendation system, spatio-temporal.

I. INTRODUCTION
The advancement of modern smart devices with location based services made it easy for people to share the locations they are visiting and their check-in information in the location based social networks (LBSNs). Such check-in data point in LBSNs yield an excellent possibility to understand the mobility of a user. Mobility prediction has a wide area of applications, ranging from recommendation systems and location based services to smart transportation and urban planning. Some recognized LBSNs are Foursquare, Yelp, Gowalla, and Facebook place where millions of check-in information are recorded. The huge volume of accumulated online footprints attracted researchers on recommending POIs (Point of Interests) which are of high interests to the users [1]–[3]. Such recommendation system can help LBSN services to improve their user experience by providing suggestions about convenient POIs [4]. It will also enable POI holders to predict the time period of next customer arrival. It may also be useful for location-aware online advertising services. With the rapid increase of such mobile applications, it has become crucial to recognize the mobility patterns of users from their past trajectory.

The task of POI recommendation differs from other recommendation systems (for example, movies, goods, news recommendation) in the sense that it has strong spatiotemporal...
dependencies of the visited POIs [12]–[14]. Zhu for modeling sequence data and capturing long term dependence on the visited locations [5]–[7]. In Fig. 1, a user’s check-in sequence is demonstrated including various contextual information like time, distance, and weather condition. From the figure, we can realize that the user prefer to go to restaurant or to the park if the weather is clear. The user also choose to go to bar during late of the night. The distance to travel also play a role when deciding where to go next. All these contextual information are crucial for providing a personalized prediction for the next location. In POI recommendation, the goal is to recommend the next possible POI that the user might be interested in based on contextual information and trajectory of the historical visit sequence. However, the task is particularly challenging due to high sparsity of the data and difficulties in incorporating various contextual information into a unified predictive model.

In the literature, various techniques have been proposed to make a personalized recommendation. Early years, methods using matrix factorization and Markov chain model were applied together for analyzing the sequential data [8]–[10]. He et al. proposed a latent factor model for capturing successive visit sequences for exploring user preferences [11]. Later on, Recurrent Neural Network (RNN) based techniques with variations in the gate mechanism have been employed for modeling sequence data and capturing long term dependencies of the visited POIs [12]–[14]. Zhu et al. [15] introduced a variation of Long Short Term Memory (LSTM) network to equip time intervals into the model for the recommendation task. However, the model did not consider the geographical distance between the two neighboring check-ins. Later, Zhao et al. [16] suggested Spatio-Temporal Gated Network (STGN) with two pairs of distance gate and time gate for controlling short term and long term interest of the user to enhance the memory network of the LSTM. One major drawback of RNN based methods (and its variants) is that, most of them rely on the output of the last hidden layer activation and when the sequence gets very large, they fail to focus on early visit sequences. As a result, the recommendation accuracy fails to improve further.

Recently, attention mechanism is becoming very popular and showed remarkable results in modeling sequential task [17]. Current state-of-the-art models in POI recommendation tries to adopt the attention mechanism on top of the RNN models [18]–[20]. However, these models fail to incorporate personalized item frequency (PIF) [21] when generating the recommendations. To resolve this issue, Luo et al. [22] used bi-attention architecture for all pairs of historical visit sequence including repetitive check-ins. Although the model learned the PIF information using matrix representation of historical check-ins, it did not make use of the order of the visited locations.

When generating recommendation, it is vital to make use of as much contextual information as possible. There is a scope to provide better personalized recommendation by incorporating weather information within the context of the visited POI. For example, a user might be interested in going to a theater during sunny weather but not during rain, whereas other users might have a different taste of choice. Or users may prefer travelling shorter distance in case of poor weather condition (e.g., storm, snow). Trattner et al. in [23] showed that it is possible to incorporate the weather data to enhance the quality of recommendations. However, weather information is rarely considered for POI recommendation. Recency (recently visited locations) of the visited POIs is another important factor for generating the recommendation [24], [25]. For example, consider a scenario where a user always visits nearby restaurant after returning from office. Thus, the model should give more focus on recommending nearby restaurants if the last visited POI was office. So, before generating the recommendation the model should learn which historical POIs should be given more focus on as well as the time of the check-in and the condition of the weather that reflects the user preference.

Another major challenge in POI recommendation is sparsity of the spatiotemporal information. It’s difficult to learn every possible continuous geographical distance and time interval without partitioning them into discrete bins. To minimize the sparsity of the temporal domain, early works divided every day into hours of discrete time slots [26]. Hierarchical gridding of the map was performed [27], [28] for reducing the spatial sparsity. However, dividing the map into discrete grids cannot properly reflect spatial distance between two POIs in the neighboring grids, as they will give the same distance if the POIs were close together or further apart within the grids. Therefore, keeping these drawbacks in mind, the research question addressing in this paper is - "RQ: How to minimize above-mentioned limitations and generate context-aware recommendation that can focus on user’s historical check-ins maintaining relative order of the visit sequence?".

To answer this research question, we present a context-aware recency based attention network (CARAN) for the...
recommendation of next point-of-interest. In CARAN, we incorporate weather information along with the spatiotemporal context for reflecting better user preference. We use attention network that can learn which locations to give more focus on depending on user’s historical check-ins. To maintain the relative order of the check-in sequence, we perform positional encoding. Instead of using hierarchical gridding mechanism, we use linear interpolation technique for spatial quantization which is more sensitive to geographical interval compared to gridding mechanism. In summary, the contribution of our work is listed below:

- We propose CARAN, that can effectively learn to give focus on user’s past visited locations by incorporating contextual information with recency based attention mechanism.
- Along with spatiotemporal data, CARAN incorporates weather information of the visited POIs which results in a richer contextual information and can provide better personalized recommendation. To the extent of our knowledge, CARAN is the first model that integrates weather condition and spatiotemporal information into a unified model.
- We perform positional encoding of the check-ins for preserving the relative order of the visit sequence within the historical trajectory of the visited POIs.
- To reflect smooth spatial distance between two POIs, we use linear interpolation instead of gridding mechanism.
- We perform extensive experiments on three real-world datasets for fine-tuning and evaluating CARAN. The results show that CARAN outperforms the state-of-the-art models for POI recommendation by 7-14%.

II. RELATED WORK
In this section, we discuss about various methods that are used in the field of POI recommendation system.

A. COLLABORATIVE FILTERING BASED METHODS
Collaborative Filtering (CF) methods examine the interactions between users and items to construct patterns between them when generating recommendations. Early years CF based methods were very popular for general recommendation system [29] (e.g., movie recommendation, item recommendation, music recommendation). It has been also used extensively for POI recommendation task where the model tried to make use of check-ins of related users or POIs [9], [30]. These techniques tried to represent every user and POI into latent vector space which was learned from observed user-item matrix. Then the recommendation is provided based on the similarity between users and POIs [31], [32]. For calculating similarity, various methods like Euclidean Distance, Cosine similarity, and Pearson similarity were used. The choice of similarity measure greatly influenced the performance of the recommendation.

Many models tried to incorporate the geographical information [33], and temporal information [34] with the CF based methods. Most of the models considered the spatial effect by considering the distance between POIs as penalty. Jiao et al. [26] fit a curve for reflecting the correlation between user’s travel distance and travel probability. Ye et al. [35] modeled geographical information using power law distribution into user-based CF framework. Major drawbacks of the CF based methods is that, most of the models work with user-item interaction matrix and fail to show the effect of spatiotemporal or sequential influence which is very important in predicting dynamic mobility of the users.

B. MARKOV CHAIN BASED METHODS
When generating POI recommendation, it is essential to contemplate different time relations and spatial distances among historical check-in sequences. In order to realize the impact of sequential information, many models utilized the properties of Markov Chain (MC) [36]-[38]. Rendle et al. [8] proposed the first MC model for the recommendation task using sequential data. It is also possible to combine MC based methods and CF based methods into a unified model for sequential recommendations as show in [39]. Cheng et al. [9], made use of Factorized Personalized Markov Chains (FPMC) with addition of physical restrictions among neighboring POIs after dividing the map into discrete grids. Liu et al. [40], proposed a multi-order Markov model that incorporates geographical influence and temporal popularity. Zhang et al. [41] used ensemble of Hidden Markov Models (HMMs) for characterizing movement regularity. MC based methods were popularly used for their simplicity since they try to find the probability of visiting next POI based on the immediate previously visited POI. However, they suffer from strong Markov assumptions and also cannot model long-term dependency.

C. NEURAL NETWORK BASED METHODS
In recent years, neural network based methods are showing promising results and successfully applied to the POI recommendation system. When modeling various features of users or items, neural networks perform really well for simulating nonlinear patterns and complex interactions [42], [43]. Zhao et al. [44], applied word2vec framework for modeling sequential context using temporal POI embedding. Yang et al. [45] proposed a semi-supervised learning framework for mitigating data sparsity and used a deep neural network framework for learning the embeddings of POIs and users. Due to the success of modeling sequential data, RNN based methods have become prevalent in the field of POI recommendation [46]-[48]. Li et al. [49], proposed a model that utilizes the time intervals between successive check-ins and explicitly modeled the timestamps for recommendation. Wang et al. [50], used similarity tree for organizing the locations and applied word2vec for embedding which is followed by RNN to model successive movement behavior. Yao et al. [51] proposed Semantics Enriched Recurrent Model (SERM) that combines embedding of diverse factors
(location, user, keyword, time) for capturing spatiotemporal transition regularities. In [52], Contextual Attention Recurrent Architecture (CARA) was proposed for leveraging both sequential and contextual information related to user’s dynamic preference. In order to incorporate spatiotemporal effect on long sequences, many variations of the RNN with attention network were proposed [18], Zhu et al. [15] adapted time gates with LSTM for capturing user’s short term interests. In [53], a multi attention network (MANC) was proposed for learning contextual information of neighborhood POIs. ATST-LSTM [19] uses LSTM network followed by an attention module for giving focus on input check-ins but only considers successive check-ins. LSTPM [54] used geo-dilated RNN for learning short-term preferences of the users.

Overall, most of the existing models are good at modeling short-term preferences but fails to capture long-term relationships between non-consecutive POIs. Also, the models overlooked the impact of weather condition on the recommendation results. In contrast, our proposed model uses a recency based attention model by incorporating weather information and preserving sequential information of the visited locations which can model non-consecutive visits and results in a more personalized POI recommendation system.

### III. PROPOSED CARAN MODEL

In this section, we first formulate the problem and then explain different layers used in the CARAN architecture. CARAN mainly consists of four layers: 1) **Input layer**, consisting of contextual information of the check-ins and spatiotemporal matrices, 2) **Embedding layer**, that converts contextual information and spatiotemporal matrices into their latent vector representation through embedding, 3) **Attention layer**, applies recency attention and predicts probability of a POI being selected as the next recommended POI, and finally, 4) **Output layer**, consisting of two phases, during training the model performs negative sampling to compute loss and during testing the model recommends top-k POIs. The overall architecture of CARAN is shown in Fig. 2.

### A. PROBLEM FORMULATION

Let us consider a location based social network, where $U = \{u_1, u_2, \ldots, u_{|U|}\}$ is the set of users and $L = \{l_1, l_2, \ldots, l_{|L|}\}$ is the set of POI locations. Each $l_i \in L$ is geocoded using a pair $(lat_i, lon_i)$ indicating the latitude and longitude of $l_i$ respectively. In addition to latitude and longitude, each POI contains categorical information (e.g., park, museum, restaurant, etc.) which is represented using the set $V = \{v_1, v_2, \ldots, v_{|V|}\}$. When a user $u_i$ checks-in at location $l_i$ at time $t_i$, the tuple $(l_i, t_i)$ gets added with the historical trajectory of $u_i$. Historical check-in sequence of user $u_i$ can be represented by the set, $S^{u_i} = \{s_1^{u_i}, s_2^{u_i}, \ldots, s_{n}^{u_i}\}$ where $s_j^{u_i} \in (l_i, t_i)$ indicating $j$'th check-in of $u_i$. For each user $u_i$, we get a variable length historical check-ins of size $m^{u_i}$. For simplicity, we convert total check-ins of each user into a fixed length of size $n$. If total check-ins of the user $u_i$ is less than $n$, i.e., $m^{u_i} < n$, then we perform zero padding on the right of the sequence which is later masked off in the model. Otherwise, we take the last $n$ check-ins of user $u_i$.

Given the historical check-in sequence $S^{u_i}$ of the user $u_i$, the goal of POI recommendation is to suggest top-$k$ relevant POIs that the user $u_i$ might be interested in. For the ease of understanding, all the notations used in our paper are described in Table 1.

| Symbol | Description |
|--------|-------------|
| $u, l, t, v, w, c$ | user, poi, time, poi-category, weather |
| lat, lon | latitude, longitude |
| $U, L, V, W$ | set of users, set of POIs, set of categories, set of all possible weather conditions |
| $S^{u_i}$ | Historical check-in sequence of user $u_i$ |
| $s_j^{u_i}$ | $j$'th check-in of user $u_i$ |
| $m^{u_i}$ | Total check-ins of user $u_i$ |
| $n$ | Fixed sequence length considered in the paper |
| $E(.)$ | Embedding of ( ) |
| $C$ | Contextual information |
| $H(x, y)$ | Haversine distance between $x$ and $y$ |
| $R(u_i)$ | Recency attention of user $u_i$ |
| $M^S, M^T$ | Trajectory spatial, temporal matrix |
| $M^{S^u}, M^{T^u}$ | Candidate spatial, temporal matrix |
| $P$ | Recommendation probability |

### B. INPUT LAYER

In the input layer, contextual information of the users are collected and two spatiotemporal matrices are formulated.

1) **CONTEXTUAL INFORMATION**

Given the user id $u_i$ and checked-in location $l_i$ at time $t_i$, we first retrieve the category $v_j$ of $l_i$. Then we retrieve the weather information $w_i$ using the OpenWeatherMap API (https://openweathermap.org/) from $(lat_i, lon_i)$ of $l_i$ and time $t_i$. The API responses with the current weather condition of that location which is one of the ten following categories: Clear, Rain, Clouds, Haze, Mist, Fog, Thunderstorm, Snow, Drizzle, and Smoke. The contextual information for the $i$'th check-in can be represented as, $c_i = (u_i, l_i, t_i, v_j, w_i)$. For all the users and their check-in information, we accumulate the contextual information for passing it into the next layer.

2) **SPATIOTEMPORAL MATRICES**

For modeling spatial and temporal interval between two locations, we define two spatiotemporal matrices: 1) trajectory spatiotemporal matrices, and 2) candidate spatiotemporal matrices. Each entry of these matrices represent the spatial distance or temporal interval between two check-ins. Trajectory spatiotemporal matrices attempt to find the correlation between non-consecutive check-ins, while the candidate spatiotemporal matrices focus on the distances and temporal intervals between all check-ins and next possible POI.

For trajectory spatiotemporal matrices, the temporal interval between $i$'th and $j$'th check-in is calculated using $|t_j - t_i|$
and the distance between two locations \( l_i \) and \( l_j \) are calculated using \( H(l_i, l_j) \) indicating the Haversine distance \([55]\) for great-circle distance of Earth. Given the check-in sequence of length \( n \), the trajectory spatial matrix \( M^S \in \mathbb{R}^{n \times n} \) and trajectory temporal matrix \( M^T \in \mathbb{R}^{n \times n} \) is calculated as,

\[
M^S_{i,j} = H(l_i, l_j), \quad 1 \leq i, j \leq n \tag{1}
\]

\[
M^T_{i,j} = |t_i - t_j|, \quad 1 \leq i, j \leq n \tag{2}
\]

In order to assist in computing probability of final recommended POI and for incorporating PIF information, we form candidate spatiotemporal matrices. For spatial candidate matrix, we compute the spatial distance between all candidate POIs \( i \in [1, |L|] \) and all checked-in locations \( j \in [1, n] \) using \( H(l_i, l_j) \). For temporal candidate matrix, we compute the time interval between \( i \)th check-in and \((n + 1)\)th check-in using \( |t_i - t_{(n+1)}| \) which is later broadcast along the row \(|L|\) times for converting into two-dimensional matrix and incorporating with the spatial candidate matrix. So, the spatial candidate matrix \( M^S \in \mathbb{R}^{|L| \times n} \) and temporal candidate matrix \( M^T \in \mathbb{R}^{|L| \times n} \) is calculated as,

\[
M^S_{i,j} = H(l_i, l_j), \quad 1 \leq i \leq |L|, \quad 1 \leq j \leq n \tag{3}
\]

\[
M^T_{i,j} = |t_i - t_{(n+1)}|, \quad 1 \leq i \leq |L|, \quad 1 \leq j \leq n \tag{4}
\]

C. EMBEDDING LAYER

In this layer, we perform embedding of contextual information and spatiotemporal matrices for converting them into their latent vector representation.

1) CONTEXT EMBEDDING

Given the contextual information \( c_i = (u_i, l_i, t_i, v_i, w_i) \), here we perform multi-modal embedding for encoding contextual information. We use embedding technique instead of one-hot encoding because total category of each contextual information (i.e., users, locations) can be very large and will take huge computation and memory power. Besides, one-hot encoding will only increase the sparsity of the data. Hence, we choose to perform embedding of the contextual information considering embedding dimension \( d_{model} = d \). In order to reduce sparsity, we map the week of the day by dividing the continuous time into slots of \( 7 \times 24 = 168 \) hours. This discretization across the time domain helps the model to learn user mobility throughout the week. Then, individual context is embedded and added together to form the embedded context \( E(c_i) \in \mathbb{R}^d \) as shown in (5).

\[
E(c_i) = \sum_{x \in (u_i, l_i, t_i, v_i, w_i)} E(x) \tag{5}
\]

Context embedding is performed on the \( n \) historical check-in of \( S^n \). In order to maintain the order of the check-in sequence, we perform positional encoding \([17]\) to the embedded context. For check-in position \( i \in [1, n] \), and embedding dimension \( j \in [1, \frac{d}{2}] \), the positional encoding \( PE \in \mathbb{R}^{n \times d} \) is computed as in (6).

\[
PE_{i,j-1} = sin\left(\frac{i - 1}{10000^j}\right), \quad 1 \leq i \leq n, \quad 1 \leq j \leq \frac{d}{2}
\]
\[ PE_{i,j} = \cos\left(\frac{i - 1}{10000^{(j-1)/d}}\right), \quad 1 \leq i \leq n, \quad 1 \leq j \leq \frac{d}{2} \]  

(6)

Then for each check-in position \( i \in [1, n] \), the context embedding \( E(c_i) \in \mathbb{R}^d \), and positional encoding \( PE(i) \in \mathbb{R}^d \), the final embedded context of user’s historical check-ins \( E(C) \in \mathbb{R}^{n \times d} \) is calculated as in (7).

\[ E(C_i) = E(c_i) + PE(i), \quad 1 \leq i \leq n \]  

(7)

2) MATRIX EMBEDDING

For the spatiotemporal matrices, if we try to learn continuous geographical distances and time intervals, then it will easily lead to sparse representation. We partition spatial distances and temporal intervals into discrete bins of hundred meters and one hour as the basic unit respectively. To reduce sparsity, it is possible to perform discrete bin embedding of the matrices. However, the latest study suggests performing linear interpolation for improved performance [56]. Hence, a linear interpolation embedding is performed on each element of the spatiotemporal matrices for smooth representation of the intervals with dimension \( d \). Finally, the spatial matrix and temporal matrix are added together by taking summation of the last embedded dimension to get the embedded matrix representation. In (8), we show how trajectory matrix embedding \( E(M) \in \mathbb{R}^{n \times n} \) is obtained from their spatiotemporal matrices. Similar calculation is performed on candidate spatiotemporal matrices to obtain candidate matrix embedding \( E(M') \in \mathbb{R}^{|L| \times n} \).

\[
\begin{align*}
E(M^S_{i,j}) &= \frac{E(\Gamma^S)(\Gamma^S - M^S_{i,j}) + E(\gamma^S)(M^S_{i,j} - \gamma^S)}{\Gamma^S - \gamma^S} \\
E(M^T_{i,j}) &= \frac{E(\Gamma^T)(\Gamma^T - M^T_{i,j}) + E(\gamma^T)(M^T_{i,j} - \gamma^T)}{\Gamma^T - \gamma^T} \\
E(M) &= E(M^S) + E(M^T)
\end{align*}
\]  

(8)

where, \( \Gamma^S \) and \( \Gamma^T \) indicate the upper bound of spatial and temporal intervals respectively, and \( \gamma^S \) and \( \gamma^T \) represent the lower bound of spatial and temporal intervals respectively.

D. ATTENTION LAYER

In this layer, we perform recency attention and compute the final candidate probability for recommendation.

1) RECENCY ATTENTION

We consider latest \( n \) visited locations of the user and using self-attention mechanism find out which visits should be given more focus on when generating the recommendation. This module combines user’s trajectory matrix embedding \( E(M) \) with the sequential context embedding \( E(C) \) and gives updated representation of each visit which can apprehend both the long-term and short-term dependencies. Furthermore, we perform masking on user’s check-in sequence if \( m^u \) is less than \( n \). As we mentioned in the problem formulation, we perform zero padding on the right, the Boolean mask \( \Psi \in (0, 1)^{n \times n} \) which is constructed using (9) will only contain ones on the upper-left portion of the mask. Later, we multiply this mask with the attention output so that the padding values do not impact final prediction.

\[ \Psi_{i,j} = \begin{cases} 
1, & \text{if } 1 \leq i, j \leq m^u \\
0, & \text{otherwise} 
\end{cases} \]  

(9)

Now, given the user’s context embedding matrix \( E(C) \in \mathbb{R}^{n \times d} \), along with the embedded trajectory matrix \( E(M) \in \mathbb{R}^{n \times n} \); recency attention \( R(u_i) \in \mathbb{R}^{n \times d} \) of the user \( u_i \) is computed using the self-attention mechanism as shown in (10).

\[
R(u_i) = \text{Attention}(Q, K, V, E(M), \Psi) = \left( \Psi \ast \text{softmax}\left(\frac{QK^T + E(M)}{\sqrt{d}}\right)\right)V
\]  

(10)

where, \( Q, K, \) and \( V \) are query, key and value of self-attention which is obtained by multiplying \( E(C) \) with learnable weights \( W_Q \in \mathbb{R}^{d \times d}, W_K \in \mathbb{R}^{d \times d}, \) and \( W_V \in \mathbb{R}^{d \times d} \) respectively. Note that, in (10) every multiplication is a matrix multiplication except for the multiplication between \( \Psi \) and the output of \( \text{softmax} \) which is an element wise multiplication.

2) CANDIDATE PROBABILITY

In this module, we compute the probability of a POI being recommended from all \( |L| \) locations using the output of the recency attention \( R(u_i) \). From the embedding layer, we retrieve location embedding \( E(l_i) \) where \( i \in [1, |L|] \). Now, given the output of recency attention \( R(u_i) \in \mathbb{R}^{n \times d} \), candidate matrix embedding \( E(M') \in \mathbb{R}^{|L| \times n} \), and location embedding \( E(L) \in \mathbb{R}^{|L| \times d} \), the probability of all POIs \( P(L) \in \mathbb{R}^{|L|} \) is computed using the formula shown in (11).

\[
P(L) = \text{softmax}\left(\frac{E(L)R(u_i)^T + E(M')}{\sqrt{d}}\right)W_p
\]  

(11)

where, \( W_p \in \mathbb{R}^{d \times 1} \) is a learnable weight matrix that is multiplied with the output of \( \text{softmax} \). \( W_p \) learns to give focus on locations which are more suitable of being selected as the next recommended POI.

E. OUTPUT LAYER

The output layer works in two separate phases, one for training the model and another for testing the accuracy of recommended POIs.

1) TRAINING PHASE

Before training the model, we need to define a loss function that the model will try to optimize. In POI recommendation system, total number of locations are very large compared to the number of locations to be recommended by the model. So, due to the imbalance distribution of positive and negative class, we cannot compute the loss function for every predicted classes. Because in this way, the model will only focus on the negative classes for reducing total loss and recall rate will drop as well. In order to resolve this, we perform negative
sampling of the predicted candidate probability \( P(L) \). We randomly sample \( Q = \{ q_1, q_2, \ldots, q_\mu \} \) POIs at each training step for computing the loss. Here, \( \mu \) indicates the number of negative samples and can be considered as a hyperparameter of our model. After every iteration of the training, we also update the random seed of negative sampler.

Given the candidate probability, \( P(L) \in \mathbb{R}^{|L|} \), target location \( l_i \), and negative samples \( Q \in \mathbb{N}^\mu \), the loss is computed as shown in (12).

\[
    loss = - \left[ \log(P(l_i)) + \sum_{i \in [1, \mu]} \log(1 - P(q_i)) \right]
\]  

(12)

The full procedure of computing loss from candidate probability is presented in Algorithm 1.

**Algorithm 1** Negative Sampling for Loss Computation

**Input:**
- \( P \): Candidate probability
- \( l_i \): Target POI location
- \( \mu \): Number of negative samples

**Output:**
- \( Q \): Total loss

1: \( \text{seed} = \) global seed for random number generation
2: \( Q = \emptyset \)
3: for \( i = 1 \) to \( \mu \) do
4: \( q = \text{Random}(\{1, |P|\}, \text{seed}) \)
5: if \((q = l_i) \) or \((q \in Q) \) then
6: \( i = i - 1 \)
7: continue
8: end if
9: \( Q = Q \cup q \)
10: end for
11: \( loss = \log(P(l_i)) \)
12: for \( i = 1 \) to \( \mu \) do
13: \( loss = loss + \log(1 - P[Q|i]) \)
14: end for
15: Update \( \text{seed} \)
16: return \(-loss\)

2) TESTING PHASE

In the testing phase, we select top-\( k \) probable POIs that is recommended by the model. Then we evaluate and compare our model with other POI recommendation frameworks.

**IV. EXPERIMENTS**

In this section, we carry out experiments on three real world datasets to assess the proposed CARAN model. First, we explain the datasets used in our experiment. Then we show the trainable parameters of our model which is followed by evaluation and comparison of our model with other baseline models. Then we present the impact of various contextual information in recommendation performance. Finally, we discuss the stability of our model followed by visualization of positional encoding and recency attention layer.

| Dataset   | #Users | #POIs | #Categories | #Check-ins |
|-----------|--------|-------|-------------|------------|
| NYC       | 1083   | 5135  | 400         | 147938     |
| TKY       | 2293   | 7873  | 386         | 447570     |
| Gowalla   | 59008  | 121944| -           | 3302414    |

### A. DATASETS

For evaluating the proposed model, we use three public LBSNs datasets: NYC, TKY, and Gowalla. Here, NYC and TKY datasets are collected from Foursquare locating in New York and Tokyo city respectively. Gowalla is another widely used global scale LBSN dataset which is popularly used for evaluating POI recommendation models. Similar to other models, we remove POIs that are visited by fewer than 10 users. The statistics of the dataset after preprocessing are presented in Table 2. For Gowalla dataset, there is no categorical information of POI, hence we consider the category embedding as zero for Gowalla dataset.

### B. BASELINE METHODS

We compare our proposed CARAN model with the following baseline methods.

- **DeepMove** [47]: It combines recurrent network with attention mechanism for capturing multilevel periodicity.
- **STGN** [16]: An enhanced LSTM model for capturing user’s preferences. It uses two pairs of time gate and distance gate for incorporating sequential information.
- **LSTPM** [54]: RNN based method that uses non-local network for long-term preference and geo-dilated RNN for the short-term preference.
- **TiSASRec** [49]: Self-attention based network that models time intervals and timestamp of interactions. However, it does not consider any spatial information.
- **GeoSAN** [28]: A geography aware self-attention based model that performs hierarchical gridding of the map without explicitly considering spatiotemporal intervals.
- **STAN** [22]: Uses a bi-layer attention model for capturing user’s preference from historical trajectory.

### C. MODEL TRAINING

In Table 3, we have shown the total number of trainable parameters of our model. For each user having \( m^{u_i} \) check-ins, we consider first \([1, m^{u_i} - 3]\) check-ins as the training samples and \([2, m^{u_i} - 2]\) check-ins as the training labels. For evaluation, we take first \((m^{u_i} - 2)\) check-ins as the input and consider \((m^{u_i} - 1)\)’th check-in as the label. Finally, for testing, we take first \((m^{u_i} - 1)\) check-ins as the input and try to predict \((m^{u_i} - 1)\)’th check-in location. We set the embedding dimension \( d_{\text{model}} = d = 60 \), and maximum trajectory length \( n = 100 \). For training, we choose learning rate of 0.003 with 0.1 drop rate. For the optimization, we use Adam optimizer. All other parameters are same as the default values. During loss computation, we fixed the number of negative samples...
We use Recall@k as the evaluation metric which is popularly used in evaluating POI recommendation models. Our model recommends top-k POIs as the output and will contribute to the recall rate if the target POI is within the recommended top-k POIs. For a user u with m\(^k\) check-ins, and CARAN’s top-k prediction by considering first (m\(^k\) - 1) check-ins of \(u\), the Recall@k is computed as:

\[
\text{Recall@k} = \frac{1}{|U|} \sum_{i \in [1,|U|]} F(l_{m^k}, \text{top-k}(u))
\]

where \(F(x, y) = 1\), if \(x \in \{y\}\);
\(0\), otherwise (13)

### E. MODEL PERFORMANCE

We test CARAN on the datasets and compare recommendation performance with the baseline methods using Recall@5 and Recall@10. The comparison result is presented in Table 4. We can observe that CARAN considerably outperforms all the baseline methods with 7-14% improvement in the recall rate. It is clear from the literature that, collaborative filtering and Markov chain based methods cannot fully utilize long term sequential nature of the visit sequence which results in low recommendation performance. Hence, we do not show such methods in the comparison.

Compared to STGN, DeepMove and LSTPM performs well among recurrent network based models due to their capability of capturing periodicity. TiSASRec only considers relative timestamps of the check-ins and ignores the impact of spatial distance. GeoSAN performs hierarchical gridding for partitioning the map which cannot reflect spatial distance smoothly. STAN follows a bi-attention layer mechanism but fails to consider the sequential property. Besides, none of them considered the impact of weather condition on user’s preference. On the other hand, CARAN considers rich contextual information of user’s check-in sequence by incorporating weather information. It applies a recency based attention mechanism that can give focus on historical trajectory based on user’s preferences. Through positional encoding, the model is also able to capture relative order of the check-ins. As a result, the proposed CARAN model exceeded the current state-of-the-art methods with significant improvement.

### F. ABLATION STUDY

To measure the impact of various contextual information on the recommendation result, we use the following variations of CARAN:

- **CARAN-C**: Remove the impact of categorical information by setting the category embedding output to zero (i.e., \(E(v_i) = 0\)).
- **CARAN-PE**: We remove the positional encoding part and pass the output of context embedding directly to the recency attention module.
- **CARAN-W**: To ignore the impact of weather condition, we set the output of weather embedding as zero (i.e., \(E(w_i) = 0\)).
- **CARAN-W-PE**: Here, we remove both the positional encoding and the weather embedding.
- **CARAN-MM**: To observe the impact of trajectory and candidate spatiotemporal matrices, we do not perform addition of these matrices in the attention mechanism.

The recall rate of these variations are shown in Fig. 3. We can see that, removing the impact of categorical information (CARAN-C) of each location, performance slightly decreases. Ignoring positional encoding (CARAN-PE) and weather information (CARAN-W) result in low recommendation performance. Compared to these variations, CARAN-W-PE gives lower performance by ignoring both weather condition and positional embedding. If we do not incorporate the spatiotemporal information (CARAN-MM), the model fails to learn the distance and temporal preferences of the user and gives the lowest performance in this case. Hence, we can say that, all the contextual information are contributing together to provide relevant recommendation to the user.

### G. MODEL STABILITY

Variations in dimension size \(d_{\text{model}}\) and number of negative samples \(\mu\) can impact the recommendation performance. We analyze the effect of these two parameters on recall rate, Recall@5.

In Fig. 4, the impact of number of embedding dimension is shown. The figure demonstrates that the suitable value for \(d_{\text{model}}\) is 60. For NYC and TKY dataset, the model seems to become stable earlier at 50. However, for Gowalla dataset the model is stable after 60 due to large number of users and locations. So, we can say the model will generalize for embedding dimension greater than 60.

The impact of number of negative samples on recommendation result is shown in Fig. 5. From the figure we...
can see that, for number of negative samples less 15 the model gives good performance. After that, the recall rate decreases for NYC and TKY as the model gives more focus on negative samples compared to positive ones. However, due to the large number of POIs in Gowalla dataset, the rate of decrease in performance is not as high as the other two datasets which is an indication that for larger dataset we can consider larger value of \( \mu \). For our model, negative sampling is crucial as the recommendation performance may decrease drastically if the value of \( \mu \) is larger than specific threshold.

**TABLE 4.** Performance comparison of CARAN with the baseline methods.

| Method      | NYC Recall@5 | NYC Recall@10 | TKY Recall@5 | TKY Recall@10 | Gowalla Recall@5 | Gowalla Recall@10 |
|-------------|--------------|---------------|--------------|---------------|------------------|-------------------|
| DeepMove    | 0.3268       | 0.4014        | 0.2684       | 0.3509        | 0.1959           | 0.2699            |
| STGN        | 0.2439       | 0.3015        | 0.1940       | 0.2710        | 0.1528           | 0.2422            |
| LSTPM       | 0.2791       | 0.3564        | 0.2568       | 0.3310        | 0.2015           | 0.2701            |
| TiSASRec    | 0.3664       | 0.5020        | 0.3031       | 0.3693        | 0.2411           | 0.3546            |
| GeoSAN      | 0.4066       | 0.5267        | 0.2957       | 0.3740        | 0.2764           | 0.3645            |
| STAN        | 0.4669       | 0.5962        | 0.3461       | 0.4264        | 0.3016           | 0.3998            |
| CARAN       | **0.5337**   | **0.6491**    | **0.3916**   | **0.4614**    | **0.3405**       | **0.4307**        |
| Improvement | 14.30%       | 8.87%         | 13.14%       | 8.20%         | 12.89%           | 7.72%             |

**FIGURE 3.** Effect of various contextual information.

**FIGURE 4.** Impact of number of embedding dimensions.

**FIGURE 5.** Impact of number of negative samples.

**H. MODEL VISUALIZATION**

We use attention mechanism in our model which does not take order of the check-in sequence into account. In order to incorporate sequential information with the model, we use positional encoding of the input check-ins. In Fig. 6, embedding output of the positional encoding is shown. Cell \((i, j)\) of the matrix indicates the \(i\)th embedding output for the \(j\)th check-in position. For example, if we take the values of positional encoding at column 0 from each row, we will get the encoding of check-in position 0. We see an inherent pattern in the positional encoding which gets added with
V. CONCLUSION

In this paper, we presented an effective and novel POI recommendation system, named CARAN, that considers rich contextual information and performs recency based attention for modeling user’s preference. We use matrix representation for finding the relevance between non-consecutive locations within the user trajectory. The recency attention mechanism helps CARAN to learn which visits to give more focus on and can capture long term dependencies. To maintain the relative order of the check-in sequence, we incorporate positional encoding with the embedded check-in context of the user. We also use linear interpolation that replaces the hierarchical gridding mechanism for smooth representation of the spatiotemporal intervals. The negative sampler used for computing cross-entropy loss outperforms the traditional binary cross-entropy loss computation technique. Experimental analysis shows that CARAN provide better recommendation by considering weather condition and incorporating positional encoding with the system. We also showed that, the model is stable and robust under hyperparameter variation. Specifically, in this paper we developed a context-aware POI recommendation system that improves the recall rate of state-of-the art models by 7-14% proving the superiority of CARAN.

REFERENCES

[1] H. Yin, B. Cui, X. Zhou, W. Wang, Z. Huang, and S. Sadiq, “Joint modeling of user check-in behaviors for real-time point-of-interest recommenda-
tion,” ACM Trans. Inf. Syst., vol. 35, no. 2, pp. 1–44, Apr. 2017.
[2] D. Massimo and F. Ricci, “Popularity, novelty and relevance in point of interest recommendation: An experimental analysis,” Inf. Technol. Tourism, vol. 23, no. 4, pp. 473–508, Dec. 2021.
[3] G. Christoforidis, P. Kefalas, A. N. Papadopoulos, and Y. Manolopoulos, “RELIE: Point-of-interest recommendations using multiple network embeddings,” Knowl. Inf. Syst., vol. 63, no. 4, pp. 791–817, Apr. 2021.
[4] P. Han, Z. Li, Y. Liu, P. Zhao, J. Li, H. Wang, and S. Shang, “Contextualized point-of-interest recommendation,” in Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI), C. Bessiere, Ed., 2020, pp. 2484–2490, doi: 10.24963/ijcai.2020/344.
[5] Y.-C. Chen, T. Thaipisutikul, and T. K. Shih, “A learning-based POI recommendation with spatiotemporal context awareness,” IEEE Trans. Cybern., early access, Jul. 15, 2020, doi: 10.1109/TCYB.2020.3000733.
[6] Q. Zhu, S. Wang, B. Cheng, Q. Sun, F. Yang, and R. N. Chang, “Context-aware group recommendation for point-of-interests,” IEEE Access, vol. 6, pp. 12129–12144, 2018.
[7] M. Davtalab and A. A. Alesheikh, “A POI recommendation approach integrating social spatio-temporal information into probabilistic matrix factorization,” Knowl. Inf. Syst., vol. 63, no. 1, pp. 65–85, Jan. 2021.
[8] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized Markov chains for next-basket recommendation,” in Proc. 19th Int. Conf. World Wide Web (WWW), 2010, pp. 811–820.
[9] C. Cheng, H. Yang, M. R. Lyu, and J. King, “Where you like to go next: Successive point-of-interest recommendation,” in Proc. 23rd Int. Joint Conf. Artif. Intell., 2013, pp. 2605–2611.
[10] B. Chang, Y. Park, D. Park, S. Kim, and J. Kang, “Content-aware hierarchical point-of-interest embedding model for successive poi recommenda-
tion,” in Proc. UCIAC, 2018, pp. 3301–3307.
[11] J. He, X. Li, L. Liao, D. Song, and W. Cheung, “Inferring a personalized next point-of-interest recommendation model with latent behav-
ior patterns,” in Proc. AAAI Conf. Artif. Intell., vol. 30, no. 1, 2016, pp. 137–143.
[12] D. Kong and F. Wu, “Hst-LSTM: A hierarchical spatial-temporal long-
short term memory network for location prediction,” in Proc. IJCAI, vol. 18, no. 7, 2018, pp. 2341–2347.
[35] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, “Exploiting geographical influence for collaborative point-of-interest recommendation,” in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. (SIGIR)*, 2011, pp. 325–334.

[36] W. Mathew, R. Raposo, and B. Martins, “Predicting future locations with hidden Markov models,” in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, 2012, pp. 911–918.

[37] J. Chen, C. Wang, and J. Wang, “A personalized interest-forgetting Markov model for recommendations,” in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 16–22.

[38] Y. Du, C. Wang, Y. Qiao, D. Zhao, and W. Guo, “A geographical location prediction method based on continuous time series Markov model,” *PLoS ONE* vol. 13, no. 11, Nov. 3318, Art. no. e0207063.

[39] R. He and J. McAuley, “Fusing similarity models with Markov chains for sparse sequential recommendation,” in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 191–200.

[40] S. Liu and L. Wang, “A self-adaptive point-of-interest recommendation algorithm based on a multi-order Markov model,” *Future Gener. Comput. Syst.*, vol. 89, pp. 506–514, Dec. 2018.

[41] C. Zhang, K. Zhang, Q. Yuan, L. Tian, Z. Hantray, and J. Han, “Gnome: Group-level mobility modeling using geo-tagged social media,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1305–1314.

[42] Z. Zhang, C. Li, Z. Wu, A. Sun, D. Ye, and X. Luo, “NEXT: A neural network framework for next POI recommendation,” *Frontiers Comput. Sci.*, vol. 14, no. 2, pp. 314–318, Apr. 2020.

[43] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural collaborative filtering,” in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[44] S. Zhao, T. Zhao, I. King, and M. R. Lyu, “Geo-tesseract: Geo-temporal sequential embedding rank for Point-of-interest recommendation,” in *Proc. 26th Int. Conf. World Wide Web Companion (WWW Companion)*, 2017, pp. 133–132.

[45] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, “Bringing collaborative filtering and semi-supervised learning: A neural approach for poi recommendation,” in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1245–1254.

[46] Q. Liu, S. Wu, L. Wang, and T. Tan, “Predicting the next location: A recurrent model with spatial and temporal contexts,” in *Proc. 50th AAAI Conf. Artif. Intell.*, 2016, pp. 194–200.

[47] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, “Deepmove: Predicting human mobility with attentional recurrent networks,” in *Proc. World Wide Web Conf.*, 2018, pp. 1459–1468.

[48] Y. Wu, K. Li, G. Zhao, and X. Qian, “Personalized long- and short-term preference learning for next POI recommendation,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1944–1957, Apr. 2020.

[49] J. Li, Y. Wang, and J. McAuley, “Time interval aware self-attention for sequential recommendation,” in *Proc. 13th Int. Conf. Web Search Data Mining*, 2020, pp. 322–330.

[50] M.-F. Wang, Y.-S. Lu, and J.-L. Huang, “SPENT: A successive preference learning for next POI recommendation,” in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 153–162.

[51] D. Yao, C. Zhang, J. Huang, and J. Bi, “SERM: A recurrent model using similarity-based POI embedding and recurrent neural network with temporal influence,” in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Apr. 2019, pp. 1–8.

[52] D. Yao, C. Zhang, J. Huang, and J. Bi, “SERM: A recurrent model for next location prediction in semantic trajectories,” in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 2411–2414.

[53] D. Yao, Q. Tian, T. Chen, Y. Liang, Q. V.-H. Nguyen, and H. Yin, “Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, Apr. 2020, pp. 214–221.

[54] N. R. Chopde and M. K. Nichat, “Landmark based shortest path detection by using A* and haversine formula,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 1, no. 2, pp. 298–302, Apr. 2013.

[55] Q. Liu, Z. Liu, and H. Zhang, “An empirical study on feature discretization,” 2020, arXiv:2004.12602.
MD. BILLAL HOSSAIN (Member, IEEE) received the B.Sc. degree in computer science and engineering from the Chittagong University of Engineering and Technology (CUET), Bangladesh, where he is currently pursuing the M.Sc. degree in computer science and engineering. He was a Research Assistant at CUET, from December 2018 to July 2019, where he is currently working as a Lecturer with the Department of Computer Science and Engineering (CSE). He is very enthusiastic about competitive programming. During the academic year, he achieved noteworthy rank in many national and international level programming contests. His research interests include algorithms, machine learning (ML), and recommender systems.

MOHAMMAD SHAMSUL AREFIN (Senior Member, IEEE) received the Doctor of Engineering degree in information engineering from Hiroshima University, Japan, with support of the scholarship of MEXT, Japan. As a part of his doctoral research, he was with the IBM Yamato Software Laboratory, Japan. He is affiliated with the Department of Computer Science and Engineering (CSE), Chittagong University of Engineering and Technology, Bangladesh. Earlier, he was the Head of the Department. He has more than 110 refereed publications in international journals, book series, and conference proceedings. His research interests include privacy preserving data publishing and mining, distributed and cloud computing, big data management, multilingual data management, semantic web, object-oriented systems development, and IT for agriculture and environment. He is a member of ACM and a fellow of IEB and BCS. He is the Organizing Chair of BIM 2021, the TPC Chair of ECCE 2017, the Organizing Co-Chair of ECCE 2019, and the Organizing Chair of BDML 2020. He visited Japan, Indonesia, Malaysia, Bhutan, Singapore, South Korea, Egypt, India, Saudi Arabia, and China for different professional and social activities.

IQBAL H. SARKER (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, Australia, in 2018. Currently, he is working as a Faculty Member of the Department of Computer Science and Engineering, Chittagong University of Engineering and Technology. He has published over 100 research papers including top-ranked journal articles and conferences with reputed scientific publishers like Elsevier, Springer Nature, IEEE, and ACM. He is an author of the book titled Context-Aware Machine Learning and Mobile Data Analytics (Springer Nature, Switzerland). His professional and research interests include data science, machine learning and AI, data-driven cybersecurity and threat intelligence, context-aware smart/intelligence computing, smart cities, systems, and security. He is one of the founders of the International AIQT Foundation, Switzerland; and a member of ACM. Recently, he has been listed in the world’s top 2% Scientist/Researcher list, prepared by Elsevier and Stanford University, USA, in 2021.

MD. KOWSHER is currently pursuing the Ph.D. with the Stevens Institute of Technology, Hoboken, NJ, USA. He worked as an Artificial Intelligence Scientist with Hishab Ltd. and an AI Engineer at NKSoft, USA. He reviewed many papers in ICCIDM; ICSECS; ICOCSIM; and Visual Computing for Industry, Biomedicine, and Art. He received the Best Paper Awards from various international conferences, such as ICONCS; IC4ME2; ICCCM; NISS; and ICIET. He was the champion of Robi r-ventures 2.0 and received the National Basis ICT Award, in 2019. In 2021, he received the Scientist of the Year Award for his excellent research in the field of AI from IEM, India. He also received the Provost Fellowship Award.

PRANAB KUMAR DHAR received the B.Sc. degree from the Chittagong University of Engineering and Technology (CUET), Chittagong, Bangladesh, in 2004, the M.Sc. degree from the University of Ulsan, Republic of Korea, in 2010, and the Ph.D. degree from Saitama University, Japan, in 2014. In 2005, he joined as a Lecturer at the Department of Computer Science and Engineering, CUET, where he is currently serving as a Professor. He has published over 30 refereed journal articles and 40 conference papers. He is an author of two books, one book chapter, and one patent. His research interests include multimedia security, digital watermarking, steganography, multimedia data compression, sound synthesis, digital image processing, and digital signal processing. He is a member of the technical committees of several international conferences. He serves as a reviewer for various reputed journals including IEEE, IEICE, Elsevier, Springer.

TAKESHI KOSHIBA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from the Tokyo Institute of Technology, in 1990, 1992, and 2001, respectively. He is a Full Professor with the Department of Mathematics, Faculty of Education and Integrated Arts and Sciences, Waseda University, Japan. His research interests include theoretical and applied cryptography, the randomness in algorithms, and quantum computing and cryptography.