IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 May 2022, accepted 16 June 2022, date of publication 27 June 2022, date of current version 30 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3186319

Companion Mobility to Assist in Future Human Location Prediction

QUAN T. NGO[1], DOI THI LAN[1], SEOKHOON YOON[1], (Member, IEEE), WOO-SUNG JUNG[2], (Member, IEEE), TAEHYUN YOON[2], AND DAESEUNG YOO[2]

[1]Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea
[2]Artificial Intelligence Research Laboratory, Intelligent Robotics Research Division, Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea

Corresponding author: Seokhoon Yoon (seokhoonyoon@ulsan.ac.kr)

This work was supported by the Institute of Information and Communication Technology Planning and Evaluation (IITP) Grant by the Korean Government through MSIT (Development of 5G-Based Shipbuilding and Marine Smart Communication Platform and Convergence Service) under Grant 2020-0-00869.

ABSTRACT Location prediction plays an important role in modeling human mobility. Existing studies focused on developing a prediction model which is based solely on the past mobility of only the person of interest (POI), rather than including information on the mobility of her/his companions. In fact, people frequently move in a group, and thus, using mobility data of a person's companions can enhance accuracy when predicting that person's future locations. Motivated by this, we propose a two-phase framework for predicting an individual's future locations that fully benefits from spatio-temporal contexts embedded in that person's and his/her companions' mobility. The framework first determines the POI's companions, then predicts future locations based on mobility information for both the POI and selected companions. Two companion selection methods are proposed in this work. The first method uses spatial closeness (SC) to determine the companions of the POI by measuring the similarity of the individuals' geographic distributions. The second method builds person ID embedding (PIE) vectors, and cosine similarity is used to select the POI's companions. To mitigate the curse of dimensionality, the framework also uses a stacked autoencoder in which the encoder compresses a high-dimensional input feature (e.g., location, time, and person ID) into a low-dimensional latent vector. For the second phase of the framework, a bidirectional recurrent neural network (BRNN)-based multi-output model is proposed to predict a person's future locations in the next several time slots. To train the BRNN model, weighted loss is used, which takes into account the importance of each future time slot to predict the POI's locations accurately. Experiments are conducted on two large-scale Wi-Fi trace datasets, demonstrating that the proposed model can effectively predict human future locations.

INDEX TERMS Human mobility, location prediction, deep learning, similarity mining, companion detection, embedding.

I. INTRODUCTION

Human mobility prediction is key in a wide range of applications, including advertising, traffic management, urban planning, and contagious disease control [1]–[3]. Governments can conduct better transportation planning and can implement scheduling to ease traffic jams and handle crowd aggregations by predicting where people will be [4]. Ride services Uber and Grab, for example, rely largely on precise mobility prediction algorithms to better estimate their customers' travel demands, scheduling resources to satisfy them [5], [6]. In opportunistic mobile social networks, where a mobile user is considered a node, correctly predicting the next node positions helps to reduce the number of route discoveries as well as the hop counts for paths between source and destination [7], [8]. Especially in industrial zones,

The associate editor coordinating the review of this manuscript and approving it for publication was Yiqi Liu.

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

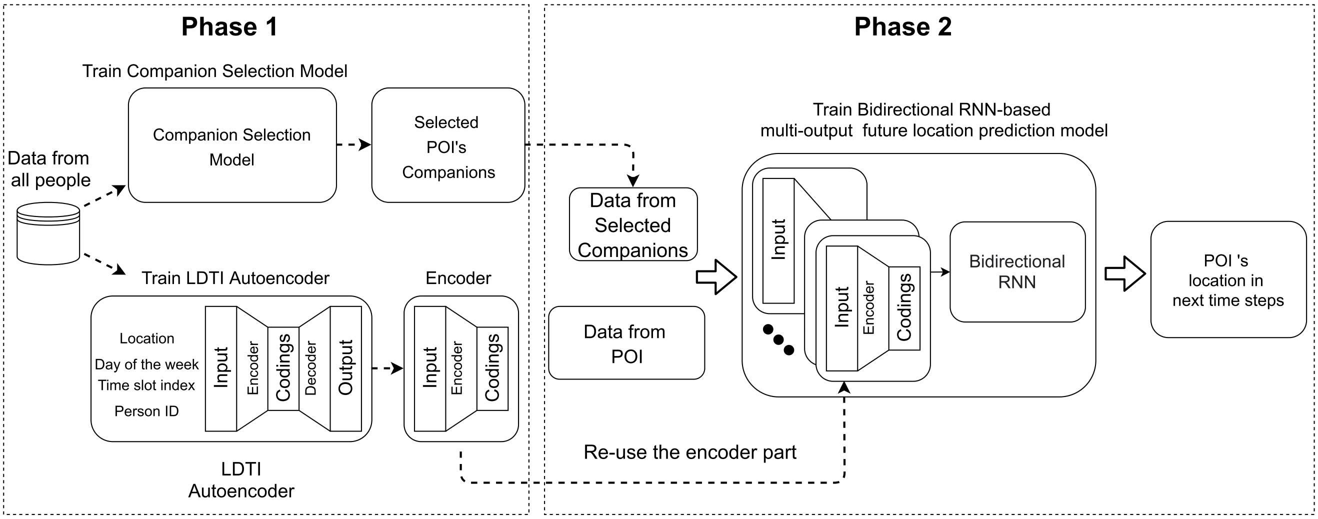


FIGURE 1. Two-phase framework for predicting an individual's future locations.

predicting the location of workers based on their recent locations is needed for improving safety and reducing the time needed for rescue operations if an incident occurs [9], [10].

Existing methods use people's recent and historical mobility information to predict their next locations [11]–[14]. Those authors focused on developing models that capture the sequential nature of human mobility preferences. Note that, these methods typically do not consider the spatio-temporal context underlying movement by a person's companions, and thus, cannot benefit from companion-mobility information. In fact, people frequently move with friends, colleagues, or coworkers; therefore, incorporating mobility information about companions can improve the performance of a location-prediction model. Motivated by this, we propose a two-phase framework that takes advantage of a person's companions' mobility information to predict the next location of the person-of-interest (POI). In a nutshell, we propose two new methods to choose companions of the POI. Then, utilizing mobility information of both the POI and his/her chosen companions, a bidirectional recurrent neural network (BRNN) predicts the POI's future locations.

For the first phase, two companion selection methods are proposed to determine the POI's companions so their mobility information can help predict the POI's future locations. The first method uses a spatial closeness (SC) [15] metric that measures the similarity in mobility between the POI and others, and then selects the most similar individuals as the POI's companions. For the second companion selection method, a person ID embedding (PIE) matrix is learned. Each embedding vector in the PIE matrix represents an individual's mobility characteristics. Similarity scores between the POI's PIE vector and vectors of other people are calculated using cosine similarity. Those who have PIE vectors with the highest similarity score to the POI's PIE vector are selected as her/his companions.

In machine learning, one-hot encoding vectors are usually used to represent the categorical input of the model. However, using one-hot vectors often leads to the curse of dimensionality when the length of the vectors becomes large. This degrades the performance of the model and increases the training time. Therefore, in the first phase of the framework, we train a stacked autoencoder [16], then reuse the encoder to mitigate the curse of dimensionality. To be more specific, instead of directly feeding the one-hot vectors of different input features (such as location, day of the week, time slot of the day, and ID) into the prediction model in the second phase, they first pass through the encoder layer of the autoencoder. The encoder layer converts input features into dense representations. This not only reduces the input dimension but also makes use of complicated correlations between input features, which improves the model's prediction.

For the second phase, the BRNN-based multi-output prediction model predicts future personal locations using recent mobility information from the POI and his/her selected companions. The RNN has a drawback in that it processes input in the exact temporal sequence. This indicates that the current input is contextualized by prior input, but not by future input. The RNN processing chain is duplicated in the BRNN, so input is processed in forward and reverse temporal order. This enables the BRNN to consider the future context. In addition, rather than simply predicting the individual's position in the very next time slot, the multi-output model is designed to predict his/her location in several time slots with different time gaps. For example, assume that the time slot length is 15 minutes and a person is now in time slot t, the model will predict positions in time slots $t+1$, $t+4$, $t+8$, $t+12$, and $t+16$ (15 minutes, one hour, two hours, three hours, and four hours later, respectively).

The proposed model was trained and tested using two large-scale Wi-Fi trace datasets, the Dartmouth dataset [17] and the UB dataset [18]. We compared our model's performance to that of several baselines: the Markov model [11], SERM [12], VANext [13], and DPBPT [14]. The results show that the proposed model outperformed the baseline models in predicting the person's next several locations.

In summary, our main contributions are as follows.

- A multi-output BRNN-based prediction model is proposed to predict an individual's future locations based on recent locations. In addition, to mitigate the curse of dimensionality, an autoencoder is proposed.
- Two companion selection methods are used to determine companions whose mobility information potentially helps predict the POI's future locations. In particular, a novel companion selection method based on a person ID embedding vector is proposed.
- The proposed model is evaluated with two large-scale Wi-Fi trace datasets, and the results demonstrate that our model can effectively predict someone's future locations.

The rest of this paper is organized as follows. Section II discusses the related works. Section III states the problem definition. Then, the proposed companion mobility-based human location prediction model is presented in detail in Section IV. The experimental datasets are described in section V. Section VI discusses the performance of the proposed models and compares them with their counterparts. Finally, the conclusions drawn from this work are discussed in Section VII.

II. RELATED WORKS

A. HUMAN MOBILITY ANALYSIS

The study of human movement using modern monitoring technologies like mobile phones [19], GPS [20], WiFi [21], and RFID devices [22] has received a lot of attention recently. A variety of research has been conducted with the goal of identifying features of human movement [23]–[26], for example, divided human mobility into three categories: geographic, temporal, and social connection. In terms of spatial features, they concentrated on geographic mobility, or how far and where a person moves. Temporal factors were taken into account, such as pauses, which represent the amount of time a person spends in a given area. Inter-contact time was defined as the elapsed time between two adjacent contacts for a pair of people, whereas the connectedness characteristic indicates contact or encounter between two people. It is worth noting that incorporating human mobility features aids in precisely predicting human movement. Thus, a mobility prediction model is developed in this paper, which takes into account features of human movement such as time, location, and social correlation.

B. LOCATION PREDICTION

Zhang *et al.* [30] found significant stability in the predictability of human mobility by calculating the entropy of an individual's trajectory. Many research attempts have been made thus far to transform this predictability into practical location prediction models [31]–[33]. The majority of early location prediction approaches are based on patterns [24]–[26]. The authors of [25] suggested Where Next, a method for predicting the next position of a moving item with a high degree of precision. The prediction is based on previously extracted movement patterns known as trajectory patterns, which are a simple representation of moving object behaviors as sequences of regularly visited places with a standard journey duration. In [26], the authors proposed geographic-temporal-semantic-based location prediction (GTS-LP), a novel mining-based location prediction approach that considers a user's geography-triggered, temporally triggered, and semantically triggered intentions in order to estimate the likelihood of the user visiting a location. The above methods extract pre-defined mobility patterns (e.g., sequential patterns, periodic patterns) from trajectory traces, and use these patterns to predict future positions. However, these techniques suffer from the one-sided nature of pre-defined patterns.

Several studies attempted to predict where a person will visit based on past knowledge about that individual's historic locations [11], [27]–[29]. The Markov model is used in the majority of these investigations. For example, the authors in [11] expanded the Mobility Markov Chain (MMC) model to include n prior visited sites, and they built a unique future location prediction technique based on this mobility model, named n-MMC. According to the authors, the suggested Markov model has higher prediction accuracy than the original Markov model. In [27], a novel division method for pre-processing trajectory data was proposed in which the original trajectory's feature points are extracted based on structural changes in the trajectory, and then important locations are extracted by clustering the feature points with an improved density, peak-clustering algorithm. Finally, a multi-order fusion Markov model based on the AdaBoost algorithm predicts the next important location of mobile users. These methods, on the other hand, are unable to detect the long-term impact and periodicity of a person's historical movements.

Deep learning approaches, particularly recurrent neural networks (RNNs) like LSTM [34] and GRU [35], have recently become popular for capturing long-term sequential impacts and movement patterns. The authors of [12] proposed an RNN-based architecture that learns the embedding of many factors (e.g., user, location, time, keyword) as well as the transition parameters of a recurrent neural network. As a result, it successfully captures semantics-aware spatio-temporal transition regularities to increase accuracy in location prediction. The authors in [13] proposed an attention recurrent network for mobility prediction. With historical mobility attention, the authors suggested a latent variable model that infers the user's next location. However, training attention networks is challenging in location prediction because they require a large amount of data to achieve optimal parameters. More recently, the authors in [14] predict a vehicle's likely destinations and routes based on the most recent partial trajectory and contextual data. Nonetheless,

TABLE 1. Comparison of some location prediction approaches.

Categories	Strengths	Weaknesses	Paper
Pattern base-models	Models are simple	Models suffer from the one-sided nature of pre-defined patterns	[24]–[26]
Markov based-models	Models are simple	Models are unable to detect the long-term impact and periodicity of a person's historical movements.	[11], [31]–[33]
Deep learning-based models	Models are improved in terms of prediction accuracy	Models are more complicated and require large amount of data	[12]–[14]

their model did not consider the vehicle ID, which aids the model in distinguishing the pattern of vehicles and improves prediction accuracy. A comparison of approaches presented in this subsection can be found in Table 1.

The above methods, however, usually do not take mobility information from the POI's companions into consideration. In fact, incorporating companions' mobility information might improve model performance when predicting a POI's next location. In our proposed framework, we first define and select companions of the POI. Then, we train a BRNN model that uses mobility information from the POI and her/his companions to predict the POI's future locations.

C. SOCIAL CORRELATION DETECTION

Human decision-making is heavily impacted by social contacts [36]–[38]. Studies on the similarity of human mobility [39] show that social ties are a substantial influencing factor in people's mobility patterns. This has given rise to the possibility of improving human mobility models by including companion movement. Graph clustering has been used in a lot of research to find social groups in which a network is separated into disjoint communities using clustering techniques [40]. Several studies on community detection have been published based on network members' contact histories, such as encounter frequency and length, and on a person's total number of previous contacts [41], [42]. Authors in [41] proposed group discovery using co-location (GDC) and decentralized GDC (DGDC) methods, which use the frequency and length of meetings to reliably discover groups. In [42], the authors used models that can reliably assess, predict, and cluster multi-modal data from people and groups within a population's social network to reveal the structure inherent in everyday activities. By evaluating the Bluetooth-based encounter history from a real-life mobility dataset, the work in [43] found social groupings within a network of mobile users. The community discovery approach focuses on creating similarity measurements that can depict the degree of social ties between users by taking into account spatio-temporal components of human interactions, and then using clustering algorithms.

These social correlation detection approaches, however, have drawbacks. They are unable to evaluate many aspects of human movement at the same time, such as location, time, and personal preference. Furthermore, if the data size increases (e.g., a large number of people, a long experimental period), these methods become more time-consuming. Our proposed social correlation detection (i.e., the companion selection method) is based on a deep learning technique that can incorporate many mobility features and that handles big datasets.

TABLE 2. Table of term.

Term	Meaning
L_t^i	One-hot vector that indicates the location of person i in time slot t
ID_i	One-hot vector that indicates the ID of person i
C_i^j	One-hot vector that indicates the ID of the j^{th} companion of person i
$L_t^{C_i^j}$	One-hot vector for location of the j^{th} companion of person i in time slot t
D_t	One-hot vector that indicates the day of the week
T_t	One-hot vector that indicates time slot t
E_l	Location embedding matrix
E_{id}	Person ID embedding matrix
k	Number of time slots to look back
POI	Person of interest
PIE	Person ID embedding

III. PROBLEM DEFINITION

Assume a day is divided into equal time slots (e.g., 15 minutes), and POI i is currently in time slot t. This work considers the problem of predicting the location of POI i in future time slot $t+m$ ($m \geq 1$) given that the recent mobility information of POI i and his/her companions are known. Instead of only predicting the POI's location in the very next time slot (i.e., $m = 1$), the proposed model is designed to predict the POI's locations in several time slots based on different time gaps (e.g., example, $m \in \{1, 2, 3\}$). Terms used in this paper are listed and defined in 2.

Let ${L_t}^i$ be the location of person i in time slot t, and let X be the recent spatio-temporal information (e.g., location, day of the week, time slot of the day, ID) of POI i and his/her companions. Note that $X = (X_{t-k+1}^i, X_{t-k+2}^i, \ldots, X_t^i)$, where k ($k \geq 1$) is the number of recent time slots. The next location

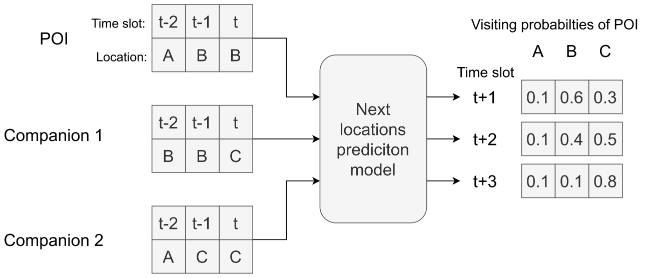


FIGURE 2. A toy example of how the prediction model work with $k = 3, L = \{A, B, C\}, m \in \{1, 2, 3\}$.

prediction is now considered as a classification problem, and the objective of this work is to learn $\hat{P}(L^i_{t+m}|X)$, which is the probability that user i will visit each location during time slot $t + m$.

A toy example of how the prediction model works is presented in Fig. 2. We consider a scenario in which the number of locations is 3, $L = \{A, B, C\}$, the number of recent time slots $k = 3$, and number of POI's companions is 2. Assume that the POI is in time slot t, the next locations prediction model predicts the POI's locations in time slot $t+1$, $t+2$, $t+3$ (e.g., $m \in \{1, 2, 3\}$). In Fig. 2, the output for time slot $t + 1$ is $(0.1, 0.6, 0.3)$, from which the predicted location of the POI at time slot $t + 1$ is B.

IV. METHODOLOGY

In this section, we first provide an overview of the proposed two-phase framework, and we then discuss in detail how the proposed framework selects the POI's companions and predicts her/his future locations.

Figure 1 depicts an overview of the proposed method, which consists of two phases. In Phase 1, two models are trained. The first is a selection model to determine the POI's companions. The second model in Phase 1 is the location, day, time, ID autoencoder (LDTI-AE), which is trained to reconstruct a concatenated vector of those features. The encoder in the LDTI-AE is reused as one layer of the prediction model in the second phase.

In Phase 2, the BRNN-based multi-output model is trained. This model takes data from the POI and his/her companions as input and predicts the POI's future location for the next several time slots. In this model, the encoder from the LDTI-AE (e.g., the model trained in Phase 1) is used to compress input features (e.g., location, time slot index, the day of the week, and ID) into dense latent representations that help to mitigate the curse of dimensionality. In the remainder of this section, the details of each part are presented.

A. COMPANION SELECTION METHODS

In this subsection, two companion selection methods are presented for identifying the POI's companions. Each uses different metrics to estimate the similarity between mobility patterns. The most similar companions are then chosen. The key advantage of these methods is that they just use the correlation in movement behavior of people, rather than requiring the information on actual social relationships between them. Note that, even if a person moves alone, the companion selection methods still can identify her/his companions.

TABLE 3. Samples of location-person vectors.

Person	Location 1	Location 2	...	Location N
u	0.064	0.453	...	0.088
v	0.082	0.621	...	0.001
...	...	...	...	...

1) Spatial closeness: The first method uses spatial closeness [15] metric. This metric compares the geographic distributions of individuals to measure the closeness between them in terms of mobility.
   a) Location-person vector construction: First, a location-person vector with a length equal to the number of locations is constructed for each person. Each element in the vector presents the probability of an individual being seen at a specific location. Table 3 shows samples of location-person vectors.
   b) Spatial closeness calculation: The SC score between the two people with vectors u and v is calculated as shown below:

$$\text{SC score} = 1 - \frac{u.v}{\|u\|_2\|v\|_2} \tag{1}$$

   The SC score indicates the distance between vectors u and v in the location-person vector space. Those who usually visit the same location (i.e., companions) will get a low SC score, otherwise, the SC score between vectors will be high.
2) Person ID embedding model: The embedding approach is used in the second companion selection model. In neural networks, embedding represents a low-dimensional, learned, continuous vector of discrete variables by mapping a discrete (categorical) variable to a vector of continuous numbers. In addition, they capture the semantics of the input features by grouping semantically similar input features in the embedding space.
   a) Person-ID-embedding matrix:
   An embedding-based companion selection model is developed to find a group of companions whose mobility information can help to predict the POI's future locations. The model learns a person ID embedding matrix that replaces a discrete person ID with a continuous PIE vector that represents the mobility characteristics of that person. Cosine similarity is then used to measure the similarity between the PIE vector of the POI and vectors of other people.
   The architecture to train the PIE matrix is shown in Fig. 3. The model consists of 2-input branches. The first branch takes ID_i – a one-hot vector

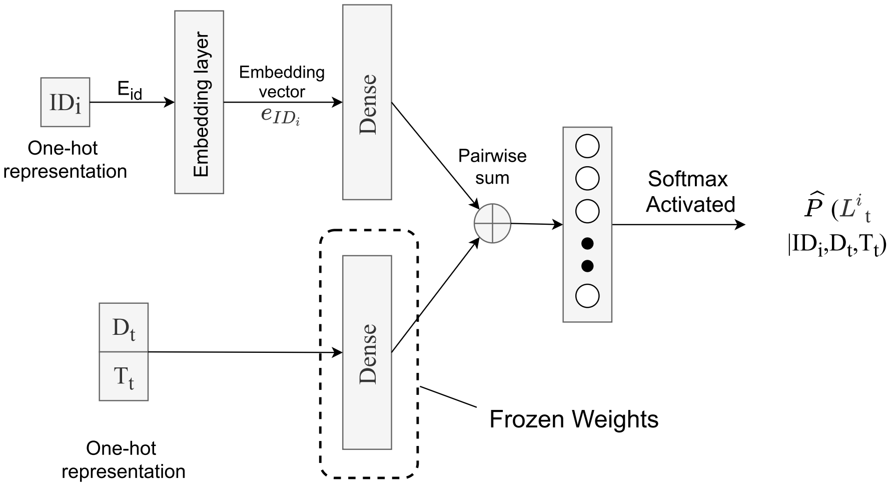


FIGURE 3. Model architecture for learning a PIE matrix in phase 1.

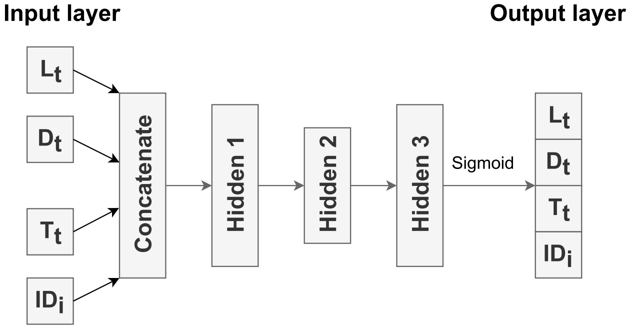


FIGURE 4. LDTI - Autoencoder architecture in phase 1.

representing the ID of person i as input. It then passes through PIE matrix E_{id} such that every person's ID can be transformed into a PIE vector, e_{ID}. The second branch takes as input one-hot vectors for the day of the week and the time slot index (D_t and T_t). Then, each of the two branches is fed into a dense layer, as shown in Fig. 3, and those two dense layers have the same number of neurons (e.g., the number of neurons equals the number of locations). The weights of the dense layer in the second branch are frozen. Lastly, outputs from the two dense layers are pairwise summed and softmax activated to estimate the individual's current location ($L_t{}^i$).

b) Similarity measurement: Given the PIE matrix, a cosine similarity metric is used to calculate the similarity score between PIE vectors u and v of two people, as shown below:

$$\text{Cosine similarity score} = \frac{u.v}{\|u\|_2\|v\|_2} \qquad (2)$$

As described above, temporal information (e.g., the day of the week, the time slot index) and spatial information (e.g., locations) are used to train a model that can estimate a person's current location. Note that weights of the dense layer in the second branch are frozen (as shown in Fig. 3), and therefore, the model only updates parameters of the dense layer and the embedding layer in the first branch. This requires the PIE matrix to be updated in a way that effectively reflects the correlation between a person and his/her visited locations. So, if two people usually move in a similar way (i.e., visit the same place at a specific time), their PIE vectors are forced to be close to each other in the embedding space. Thus, the similarity score between these two PIE vectors will be high. On the other hand, if the two people's movements differ, their PIE vectors in the embedding space are far apart, and their PIE vectors' similarity score is low.

B. LOCATION-DAY-TIME SLOT-ID AUTOENCODER (LDTI-AE)

An autoencoder is an artificial neural network that can learn dense representations of input data, which are referred to as latent representations or codings. The autoencoder is useful for dimensionality reduction since these codings have a much lower dimensionality than the input data. In this paper, we propose using a stacked autoencoder that takes one-hot vectors for the location, day of the week, time slot, and the person's ID as input. This LDTI-AE is forced to reconstruct the input vectors, and the LDTI-AE compresses the spatio-temporal and personal features into dense latent representations that can reflect complicated user mobility characteristics. The architecture of the LDTI-AE is shown in Fig. 4.

The proposed LDTI-AE consists of three hidden layers along with an input layer and an output layer. L_t, D_t, T_t, ID_i are one-hot vectors that represent the location, day of the week, the index of time slot t, and the ID of person i, respectively. The hidden layer is much smaller than the total

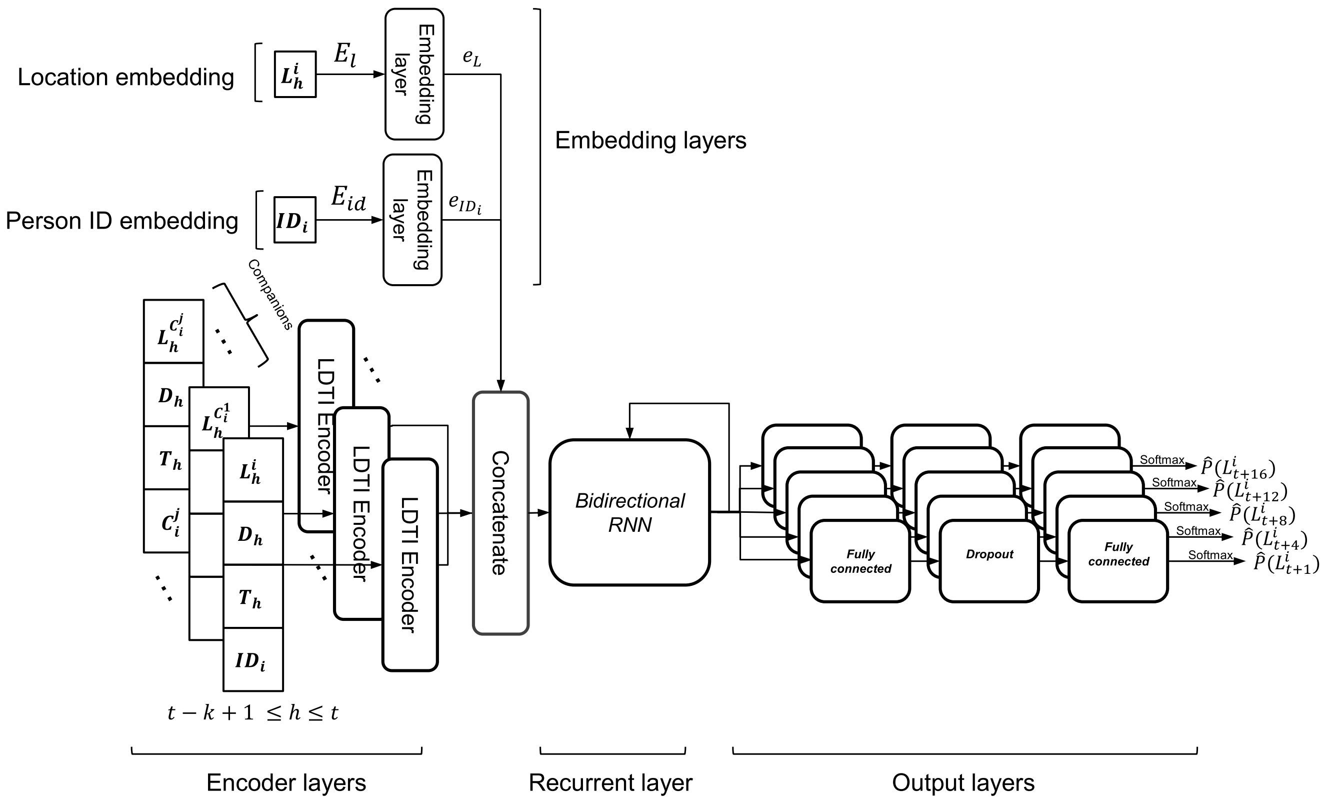


FIGURE 5. The BRNN-based future locations prediction model.

size of all input vectors, and the ReLU activation function is used for hidden layers 1 and 2. Since the output is the concatenated vector of L_t, D_t, T_t, and ID_i, the sigmoid activation function is used in the output layer, and cross-entropy loss is used to train the autoencoder.

C. BRNN-BASED FUTURE LOCATION PREDICTION MODEL

To predict future locations, a BRNN-based multi-output model using an LSTM cell is proposed. Data from the POI and his/her two selected companions is used as input for the BRNN model. The model will then predict the locations the POI may visit in the next time slots, $t + m, m \in \{1, 4, 8, 12, 16\}$. Figure 5 shows the architecture of the proposed model.

As shown in Fig. 5, the proposed model consists of four layers: the encoder layer, the embedding layer, the recurrent layer, and the output layer. Details of each layer are as follows.

1) The encoder layers: These employ the encoder (i.e., up to hidden layer 2 of the LDTI-AE) to compress the spatio-temporal and personal features of the POI and his/her companions into dense latent representations. Note that all encoder weights are frozen in this phase.
2) The embedding layers:
   a) Location embedding: This layer learns embedding matrix, E_l, such that every location L can be transformed into embedding vector e_L.
   b) Person ID embedding: The ID embedding layer learns embedding matrix, E_{id}, such that the ID of person i, ID_i, can be transformed into embedding vector e_{ID}. Please note that embedding matrix E_{id} in this layer is different from the PIE matrix that is described in the Companion Selection Model subsection in Section IV.
3) The recurrent layer: Output from the encoder and embedding layers is then concatenated before being fed into the recurrent layer. In this work, the recurrent layer is a BRNN that uses LSTM cells.
4) The output layers: The last hidden state of the LSTM cell then passes through five identical branches, each of which includes a fully connected layer with ReLU activation, a dropout layer, and a fully connected layer with a softmax activation function. As illustrated in Fig. 5, each branch corresponds to predicting the POI's location in a certain time slot. In particular, the output of the branch that predicts the location of POI i in time slot $t + m, m \in \{1, 4, 8, 12, 16\}$ is $\hat{P}(L_{t+m}^i)$, which is the probability the POI will visit each location in time slot $t + m$.

TABLE 4. Statistics on the two experimental datasets after pre-processing.

	Dartmouth	UB
Number of people (mobile users)	50	50
Number of locations (APs)	623 + 1 dummy	1243 + 1 dummy
Number of time slots per day	41	41
Experiment period	118 days	90 days

D. PARAMETER LEARNING

To train and update model parameters, the cross-entropy loss function is utilized. The total loss of the five outputs is computed as follows:

$$\text{Total loss} = \sum^{batchsize} \sum_{m} -\log(\hat{P}(L_{t+m})) \tag{3}$$

where $m \in \{1, 4, 8, 12, 16\}$ corresponds to output of the prediction model; $\hat{P}(L^{i}_{t+m})$ is the model's predicted probability for the POI's location in time slot $t + m$.

In practice, the predictions for near-future time slots are usually more useful than the predictions for distant time slots. Thus, weighted loss is applied to the model's five output loss terms. The concept of weighted loss applies a higher penalty to the loss term for output that corresponds to a predicted location in the near future but applies a lower penalty to the loss term for output that corresponds to the location in a distant-future prediction. The weighted total loss of the five output losses is computed as follows:

$$\text{Weighted total loss} = \sum^{batchsize} \sum_{m} -W_m \log(\hat{P}(L^{i}_{t+m})) \tag{4}$$

where

$$W_m = \frac{1}{m} \times h;\ h = \frac{1}{\sum_{n\in\{1,4,8,12,16\}} \frac{1}{n}} = \frac{48}{73} \tag{5}$$

is the weight for the loss term of the output that predicts the POI's location in time slot $t + m$; h is the constant that keep $\sum_{m\in\{1,4,8,12,16\}} W_m = 1$

V. DATASETS

Two large-scale datasets of Wi-Fi traces were used to train and test the proposed framework. Due to the relatively short communication range of Wi-Fi technology, the location of the connected access point (AP) can be considered the location of the mobile user at that time [44], [45]. Thus, human mobility can be represented as a sequence of connected APs. The first dataset is the Dartmouth dataset [17], which provides Wi-Fi logs of 13,888 mobile device carriers on the Dartmouth College campus. A log including a timestamp, the device ID, and the basic service set identifier (BSSID) was recorded when each mobile device associates or disassociates itself with a Wi-Fi AP. The second dataset is the Buffalo/phonelab-wifi dataset (UB dataset) [18], which contains five months of data from smartphones carried by a group of 284 University at Buffalo (UB) faculty members, staff, and students.

Dartmouth data from the 50 most active mobile users over 118 days (from January 3 to April 30, 2004) were selected as the dataset for training and testing the models. In this experiment, 623 APs (locations) were chosen. Similarly, from the UB dataset, the 50 most active mobile users for 90 days were selected, along with 1243 APs.

Since data samples in both datasets were collected at different time intervals, the data were pre-processed. We assumed a working day to be from 8:00 to 18:00, and is divided into time slots, then timestamp of each record was mapped to pre-defined time slots. In the two datasets, mobile users typically stay at a location for a period of time (e.g., during a class). We fixed the time slot duration to 15 minutes in this work. Thus, there are total of 41 time slots per day, including the last time slot. It should be noted that the user's most recent location within a specific time slot is considered the location of the user in that entire time slot. Furthermore, time slots with no data records were labeled with a dummy location. Table 4 shows statistics of the two datasets after pre-processing.

Because the two datasets were sparse, a large portion of time slots was labeled with the dummy location. Owing to this imbalance in the data, the model kept predicting the dummy location, which was of no value. To avoid this problem, all training and validation samples labeled with the dummy location were eliminated (i.e., all five labels for five outputs $t + 1, t + 4, t + 8, t + 12$, and $t + 16$ had a real location rather than the dummy location). It is worth noting that we only trimmed dummy locations from the labels, so dummy locations could still exist in the input features. Furthermore, to ensure that the model could predict all the time slots between 9:00 and 18:00, extra time slot data from 8:00 to 22:00 were used in the training phase.

Please note that for a fair assessment, in the test phase, the model was assessed independently with each output. During this phase, only the label for the considering output is required to have a location (e.g., the labels of other outputs can be dummy locations). For instance, to evaluate performance when predicting the location in time slot $t+1$, the model only needed to ensure the label for output $t+1$ was not the dummy location. Therefore, five distinct test sets were created during the testing phase to evaluate the model with each output for $t + 1, t + 4, t + 8, t + 12$, and $t + 16$. Figure 6 shows the input-label timelines of the five test sets.

Recall that in the UB and Dartmouth traces, the experiment periods were 90 and 118 days, respectively. For the day dimension, each dataset was randomly divided into training,

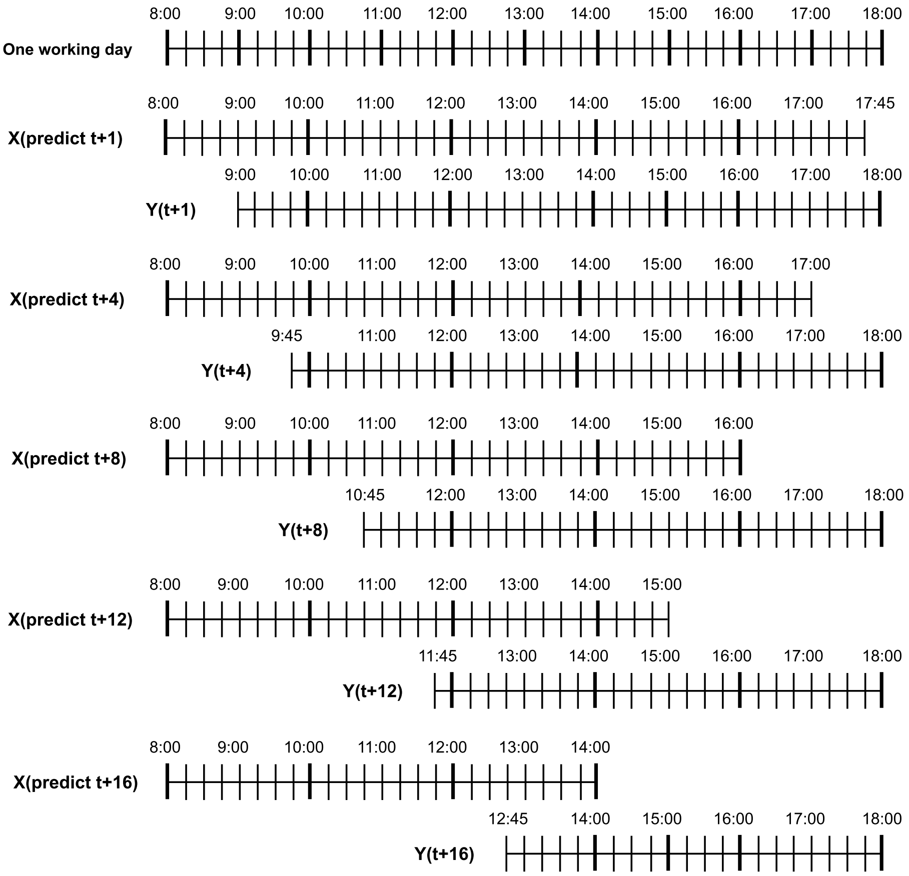


FIGURE 6. Input-Label timelines of five test sets.

validation, and test sets at a 6:2:2 ratio. The first two sets were used to fit the models and determine the best hyper-parameters, while the last set was used to evaluate model performance.

VI. EXPERIMENT RESULTS

In this section, we provide the results from a series of experiments carried out to evaluate the efficacy of our proposed framework. We begin with the baseline models and then move on to the implementation details. Finally, we present the experiment results followed by validation of the proposed approaches.

A. BASELINE MODELS

The proposed model was compared with the following baseline methods. The Markov model [11] learns to define movement regularities, then is used to select the place with the highest probability for a future location. SERM [12]captures diverse contexts underlying human movement, and enables semantics-aware next-location prediction by learning the embedding vectors of several variables (ID, location, time, keyword) and the transition parameters of a recurrent neural network. VANext [13] is a variational attention mechanism that predicts the next position of a person by combining past mobility pattern periodicity with recent check-in preferences. In [14], the next location is predicted based on the most recent locations and we use 'DPBPT' to present this baseline model.

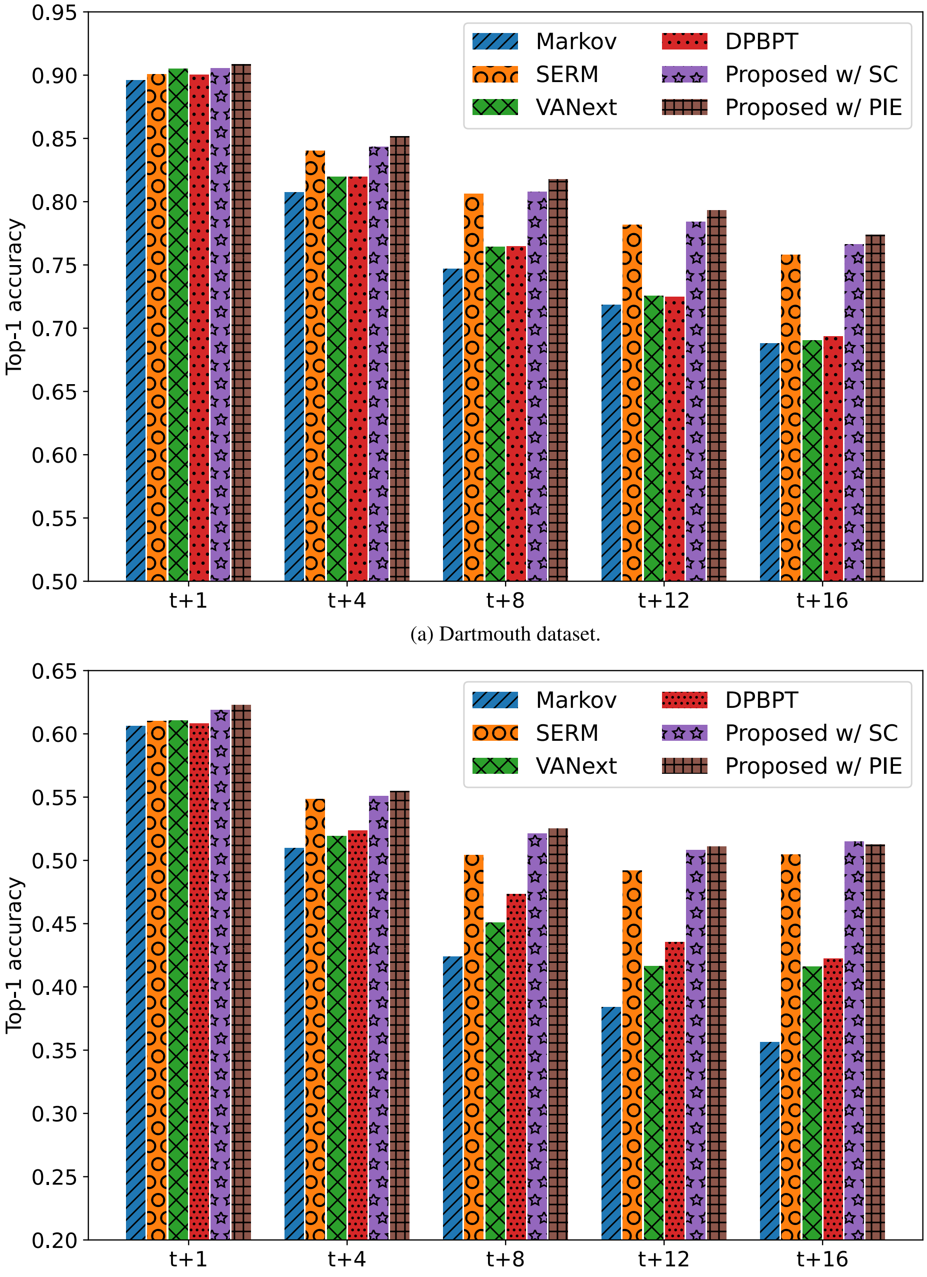
B. IMPLEMENTATION DETAILS

The proposed and baseline models were implemented using the TensorFlow Keras library on a four-core Xeon CPU with a single Titan-XP GPU. Each model was trained for 50 epochs, and the best set of hyper-parameters was chosen based on validation accuracy. The model is optimized by the RMSprop optimizer with a learning rate of 0.001, and the batch size is set to 1024. The baseline models were tuned to obtain the best performance with each dataset.

C. EXPERIMENT RESULTS

Figure 7 shows the top-1 test accuracy from the proposed and the baseline models when predicting an individual's locations in the next time slots (e.g., $t+1$, $t+4$, $t+8$, $t+12$, and $t+16$). Comparisons of these methods resulted in the following.

RNN-based models (SERM, VANext, the proposed model with SC, and the proposed model with PIE) outperformed the Markov-based model by a large margin when predicting the target person's location in future time slots, owing to two factors. First, the Markov model is based on assumptions about



(a) Dartmouth dataset.

(b) UB dataset.

FIGURE 7. Top-1 accuracies on the two datasets from the different methods predicting POI locations in several time slots.

personal movement distributions. Second, Markov models can only model first-order dependencies for a person's movement characteristics, while RNN-based models can model long-term dependencies. Note that the performance of the Markov model when predicting the individual's location in the next time slot, $t+1$, was high thanks to the characteristics of the user's movements in the two datasets, where mobile users frequently stayed in one place for a couple of time slots.

The performance of the VANext model and the DPBPT when predicting an individual's location in distant time slots dropped significantly. The major issue is that both models do not consider the person ID, which has an important role in future location prediction. The person ID helps the model distinguish mobility characteristics between people, and makes for a more accurate prediction when estimating that person's future location. And SERM was a strong baseline when predicting a person's location in future time slots cause this model considered the person ID.

Models that utilized companion mobility information (e.g., the suggested model with SC and PIE) outperformed

TABLE 5. Average top-1, top-3, and top-5 accuracies with Dartmouth data.

Model	Avg. Top-1 Accuracy	Avg. Top-3 Accuracy	Avg. Top-5 Accuracy
Markov	0.7713	0.8703	0.8964
SERM	0.8174	0.9376	0.9573
VANext	0.7809	0.8865	0.9013
DPBPT	0.7805	0.8878	0.9052
Proposed model with SC	0.8214	0.9431	0.9582
Proposed model with PIE	0.8289	0.9462	0.9590

TABLE 6. Average top-1, top-3, and top-5 accuracies with UB data.

Model	Avg. Top-1 Accuracy	Avg. Top-3 Accuracy	Avg. Top-5 Accuracy
Markov	0.4560	0.6496	0.7089
SERM	0.5318	0.7401	0.7970
VANext	0.4824	0.6523	0.7126
DPBPT	0.4925	0.6648	0.7210
Proposed model with SC	0.5427	0.7466	0.8043
Proposed model with PIE	0.5452	0.7499	0.8095

models that do not use companion mobility information (Markov, SERM, VANext, and DPBPT). This demonstrates that using companion mobility information improves the model's accuracy. Models using the PIE approach surpassed the models that use the SC approach because the PIE approach incorporates both spatial and temporal aspects in people's movements, whereas the SC technique just considers the spatial component.

The average top-1, top-3 and top-5 test accuracies of the five outputs of different models are shown in Table 5 and Table 6. As can be seen, the proposed models that consider companion mobility (with SC and with PIE) outperformed baseline models that do not include companion information. Again, this proves that human movement is highly dependent on companions; consequently, incorporating companion mobility information into the prediction model enhances performance when predicting the POI's future locations.

Table 7 shows the number of parameters for each model on the two experimental datasets. In general, the proposed models (with SC and PIE) have fewer parameters than the SERM and DPBPT models, demonstrating the effectiveness of our proposed model. Although the VANext model has the fewest parameters, its performance in distant time slots is significantly lower than other models.

D. SENSITIVITY OF HYPER-PARAMETERS

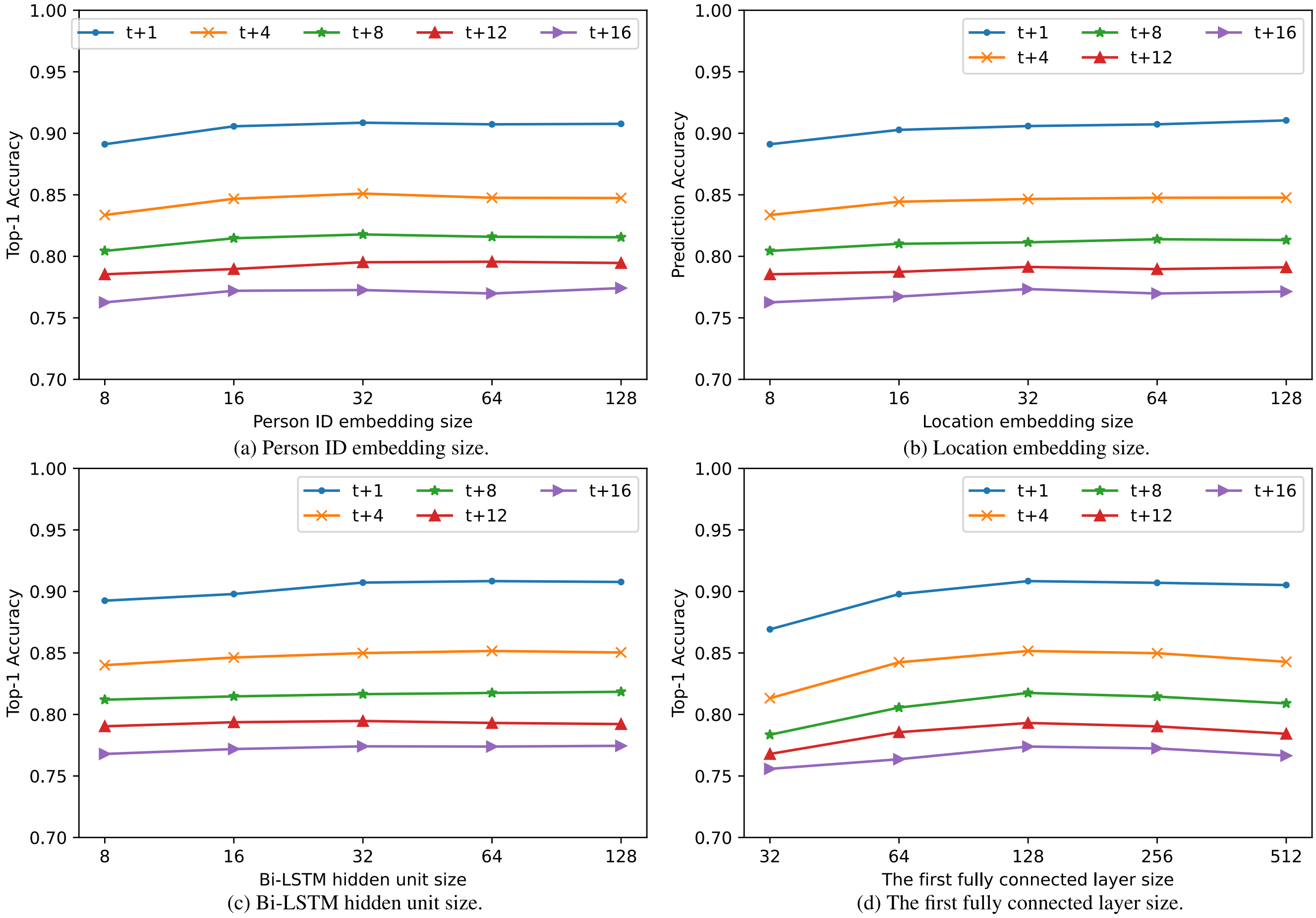
The impact of different model hyper-parameter settings on location prediction performance is investigated in this subsection. Specifically, we evaluate the sensitivity of four hyper-parameters: person ID embedding size, the location embedding size, the number of bi-LSTM hidden units, and the first fully connected layer size. We set the person ID embedding size and the location embedding size to {8, 16, 32, 64, 128}, the number of bi-LSTM hidden units to {8, 16, 32, 64, 128}, and the first fully connected layer size to {32, 64, 128, 256, 512}. Except for the parameters under test, all other parameters were left at their default settings. Figure 8 shows the performance comparison when we vary the values of model parameters on the Dartmouth dataset. In general, we find that using a larger value of the hyper-parameters improves performance by allowing for a more powerful representation. However, this increases the model's complexity and makes it prone to overfitting. In our experiments, based on the results in Fig. 8, we chose the following hyper-parameters to account for the effectiveness vs. computational cost trade-off: the location and user embedding sizes are set to 64, the number of bi-LSTM hidden units is set to 64, and the size of the first fully connected layer is set to 128.

TABLE 7. Number of parameters of different models on the two datasets.

Model	Dartmouth	UB
SERM	442,572	607,336
VANext	168,451	341,140
DPBPT	586,672	1,026,252
Proposed model with SC	354,096	554,576
Proposed model with PIE	414,144	521,788

E. EFFECTIVENESS OF WEIGHTED LOSS

In this subsection, the effectiveness of using weighted loss is validated. Figure 9 depicts the performance of the proposed model with PIE as trained on the two experimental datasets using normal loss and weighted loss. The weighted loss, as shown in the figure, improves the models' accuracy when predicting the POI's location in near-future time slots (e.g., $t+1$ and $t+4$). As previously stated, adopting weighted loss forces the model to be tuned to predict locations in



(a) Person ID embedding size.

(b) Location embedding size.

(c) Bi-LSTM hidden unit size.

(d) The first fully connected layer size.

FIGURE 8. Performance with varying parameters on Dartmouth dataset. (a) Person ID embedding size. (b) Location embedding size. (c) Bi-LSTM hidden unit size. (d) The first fully connected layer size.

near-future time slots, which improves performance for these time slots. It is worth noting, however, that the model's performance when predicting locations in distant-future time slots (e.g., $t + 8$, $t + 12$, and $t + 16$) improved with the two experimental datasets. One possible explanation is that assigning different weights to the loss term of each model's output impacts the learning rate while tuning the model parameters upon output. This is an example of an adaptive learning rate that increases the accuracy of a multi-output model with every output.

F. PERSON ID EMBEDDING COMPANION SELECTION VALIDATION

In this subsection, the proposed PIE-based companion selection method is validated. This step's purpose is to demonstrate that the PIE vectors of two people with similar mobility are close to each other in the embedding space. Otherwise, their PIE vectors remain apart. For the demonstration, the t-SNE technique [46] was employed to visualize the PIE vector.

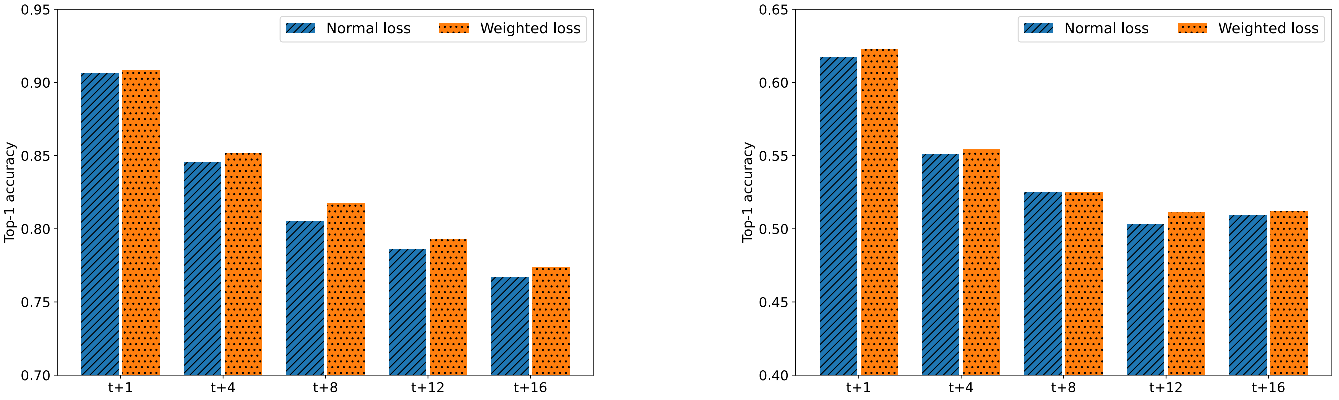
The Dartmouth dataset was used for this subsection. Besides the original data from 50 people, synthetic data of four new people (IDs 51, 52, 53, and 54) were generated from the original data. The data preparation is described as follows.

- Step 1: Pick training data of five people at random from a pool of 50 original people. Here, randomly chosen people are the persons with IDs 1, 34, 44, 10, and 50.
- Step 2: Create data of the four new people by randomly combining data samples of the five people chosen in Step 1. The synthetic data construction of the four new people (IDs 51, 52, 53, and 54) is shown in Fig. 10. Please note that the data of each chosen person were randomly selected at the sample level. For example, to create data for person 51, 50% of the sample data of person 1, and 50% of the sample data of person 34 were randomly chosen and combined.

Given the data from 54 people, a PIE model was trained to get the PIE vectors, which were then visualized into two-dimensional space using the t-SNE technique. Figure 9 shows the t-SNE plot of the PIE vectors for all 54 people. Each point in the plot indicates a PIE vector. We focused on the PIE vectors of the following IDs: 1, 10, 34, 44, 50, 51, 52, 53, and 54.

Some observation can be made from the t-SNE plot:

- The mobility of person 50 and person 54 are the same, and thus, the PIE vectors of that pair (50 and 54) are close



(a) Dartmouth dataset

(b) UB dataset

FIGURE 9. **Performance comparison of the proposed model with PIE, trained using normal loss and weighted loss.**

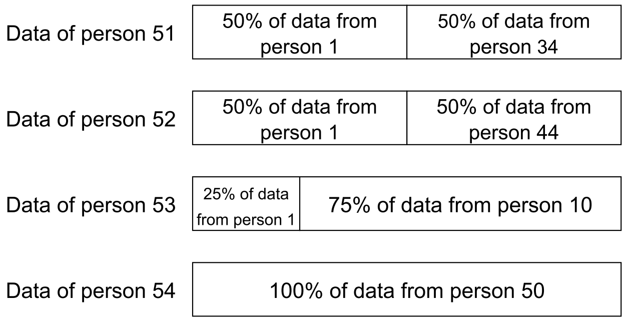

Data of person 51
50% of data from person 1
50% of data from person 34
Data of person 52
50% of data from person 1
50% of data from person 44
Data of person 53
25% of data from person 1
75% of data from person 10
Data of person 54
100% of data from person 50

FIGURE 10. **Synthetic data construction for four new people with IDs 51, 52, 53, and 54.**

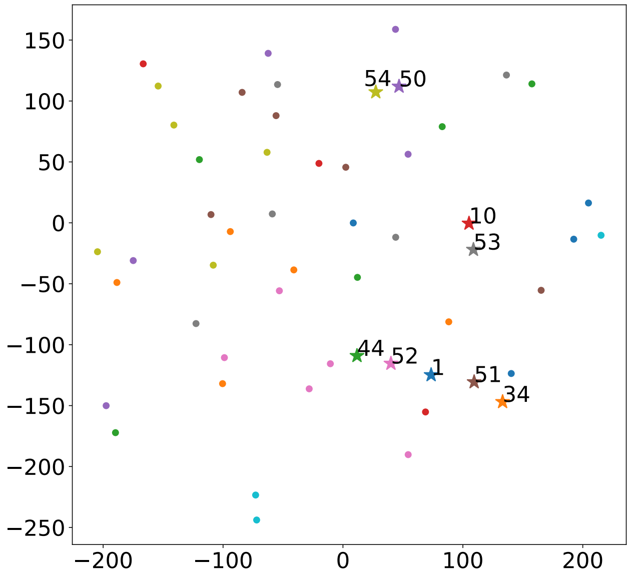


FIGURE 11. **t-SNE plot of the PIE vectors for all 54 people. Each point in the plot indicates a PIE vector.**

TABLE 8. **The cosine similarity scores between different people's PIE vectors.**

Person ID	51	52	53	54
1	**0.7127**	**0.6954**	**0.5522**	**0.0601**
34	**0.7913**	0.2666	-0.0189	0.0693
44	0.1681	**0.7480**	-0.1289	0.3359
10	-0.0254	-0.0302	**0.8447**	0.1488
50	0.1167	0.2555	0.1442	**0.9999**

to each other. It is worth noting that, despite the fact that the PIE vectors of 50 and 54 are highly similar, the two vector points that represent them do not overlap. This is due to the randomness in the t-SNE algorithm [46].

- The same observation can be seen for the PIE vectors of the following pairs: 44 and 52, 10 and 53, and 51 and 34.
- The PIE vectors of group 51, 52, and 53 are near the PIE vector of person 1 because their mobility is similar to that of person 1.

The cosine similarity scores of the PIE vectors of the four new people and the five chosen people are shown in Table 8. Some conclusions can be drawn from the table.

- The cosine similarity score between person 50 and person 54 PIE vectors is approximately 1 (e.g., 0.9999). This is because the mobility of these two people is the same. Similarly, the cosine similarity score between PIE vectors of 44 and 52, 10 and 53, and 51 and 34 are relatively high.
- The cosine similarity score between PIE vectors for person 53 and person 1 is lower than the scores for person 51 and person 1 and for person 52 and person 1 (e.g., 0.5522 vs. 0.7127 and 0.6954, respectively). This is because the mobility data for person 53 used only 25% of the mobility data from person 1, whereas the mobility data for persons 51 and 52 used 50% of the mobility data from person 1.
- The mobility data for persons 54 and 1 are unrelated. Thus, the cosine similarity score between their PIE vectors is relatively low (e.g., 0.0601).

- The mobility data for person 53 used 75% of the mobility data from person 10. As the result, the cosine similarity score between their PIE vectors is high (e.g., 0.8447).

VII. CONCLUSION

In this paper, a two-phase framework for predicting a person's future locations is proposed. Our framework uses mobility information from the POI and her/his companions to predict the POI's future locations. Two companion selection methods were used, which efficiently determined POI companions whose mobility information aided in accurately predicting the POI's future locations. In particular, a novel companion selection method was proposed, which is based on the embedding technique. Furthermore, the autoencoder architecture is used to learn the latent vector with multiple factors (location, day of the week, time slot in the day, and person ID) underlying human motion and to tackle the curse of dimensionality. Lastly, a BRNN-based multi-output model was trained using mobility information of the POI and his/her companions under the supervision of weighted loss to predict the location of the POI in the next several time slots. It is worth noting that the companion selection methods in our framework use on the similarity in movement behavior between persons rather than their true social ties. Thus, the proposed approach is broadly applicable to various types of spatio-temporal datasets (e.g., the framework does not require the social relationship information between people). We evaluated our model with two real-world datasets, and the results showed that it outperformed baseline methodologies when predicting an individual's future locations.

Due to data limitations, our work does not consider the semantic context of individual mobility. For future work, we plan to include this semantic context in the model in order to predict not just the person's movement, but also the purpose behind it. Many other factors, such as time (workdays vs. weekends), and personal preferences, will also be taken into account to improve the model's accuracy.

REFERENCES

[1] F. Asgari, V. Gauthier, and M. Becker, "A survey on human mobility and its applications," 2013, *arXiv:1307.0814*.

[2] D. K. Rossmo, *Geographic Profiling*. Boca Raton, FL, USA: CRC Press, 1999.

[3] A. Wesolowski, N. Eagle, A. J. Tatem, D. L. Smith, A. M. Noor, R. W. Snow, and O. C. Buckee, "Quantifying the impact of human mobility on malaria," *Science*, vol. 338, no. 6104, pp. 267–270, 2012.

[4] R. Wu, G. Luo, J. Shao, L. Tian, and C. Peng, "Location prediction on trajectory data: A review," *Big Data Mining Anal.*, vol. 1, no. 2, pp. 108–127, 2018.

[5] M. Chen, Q. Liu, W. Huang, T. Zhang, Y. Zuo, and X. Yu, "Origin-aware location prediction based on historical vehicle trajectories," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 1, pp. 1–18, Feb. 2022.

[6] A. K. Laha and S. Putatunda, "Real time location prediction with taxi-GPS data streams," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 298–322, Jul. 2018.

[7] N. Meghanathan, "A location prediction based routing protocol and its extensions for multicast and multi-path routing in mobile ad hoc networks," *Ad Hoc Netw.*, vol. 9, no. 7, pp. 1104–1126, Sep. 2011.

[8] L. N. Balico, A. A. F. Loureiro, E. F. Nakamura, R. S. Barreto, R. W. Pazzi, and H. A. B. F. Oliveira, "Localization prediction in vehicular ad Hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2784–2803, 4th quart., 2018.

[9] S. G. Siddharth, G. M. Tamilselvan, and C. Venkatesh, "Location prediction for improved human safety at complex environments," *Comput., Mater. Continua*, vol. 71, no. 3, pp. 5219–5234, 2022.

[10] H. Bi, W.-L. Shang, and Y. Chen, "Cooperative and energy-efficient strategies in emergency navigation using edge computing," *IEEE Access*, vol. 8, pp. 5455–54441, 2020.

[11] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Next place prediction using mobility Markov chains," in *Proc. 1st Workshop Meas., Privacy, Mobility (MPM)*, 2012, pp. 1–6.

[12] 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.

[13] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Predicting human mobility via variational attention," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 2750–2756.

[14] P. Ebel, I. E. Gol, C. Lingenfelder, and A. Vogelsang, "Destination prediction based on partial trajectory data," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1149–1155.

[15] F. Alhasoun, M. Alhazzani, F. Aleissa, R. Alnasser, and M. González, "City scale next place prediction from sparse data through similar strangers," in *Proc. ACM KDD Workshop*, vol. 2017, pp. 191–196.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[17] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo. (Sep. 2009). *CRAWDAD Dataset Dartmouth/Campus*. (Sep. 9, 2009). Downloaded from. [Online]. Available: https://crawdad.org/dartmouth/campus/20090909

[18] J. Shi, C. Qiao, D. Koutsonikolas, and G. Challen. (Mar. 2016). *CRAWDAD Dataset Buffalo/Phonelab-WiFi*. (Mar. 9, 2016). [Online]. Available: https://crawdad.org/buffalo/phonelab-wifi/20160309

[19] S. Jiang, G. A. Fiore, Y. Yang, J. Ferreira, Jr., E. Frazzoli, and M. C. A. G. Lez, "A review of urban computing for mobile phone traces: Current methods, challenges and opportunities," in *Proc. 2nd ACM SIGKDD Int. Workshop Urban Comput.*, 2013, pp. 1–9.

[20] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proc. 10th Int. Conf. Ubiquitous Comput. (UbiComp)*, 2008, pp. 312–321.

[21] Y. Zhang, L. Wang, Y.-Q. Zhang, and X. Li, "Towards a temporal network analysis of interactive WiFi users," *Europhys. Lett.*, vol. 98, no. 6, p. 68002, Jun. 2012.

[22] J. Fournet and A. Barrat, "Contact patterns among high school students," *PLoS ONE*, vol. 9, no. 9, Sep. 2014, Art. no. e107878.

[23] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella, "Human mobility models for opportunistic networks," *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 157–165, Dec. 2011.

[24] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[25] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "WhereNext: A location predictor on trajectory pattern mining," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2009, pp. 637–646.

[26] J. J.-C. Ying, W.-C. Lee, and V. S. Tseng, "Mining geographic-temporal-semantic patterns in trajectories for location prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, pp. 1–33, Dec. 2013.

[27] H. Wang, Z. Yang, and Y. Shi, "Next location prediction based on an AdaBoost-Markov model of mobile users," *Sensors*, vol. 19, no. 6, p. 1475, Mar. 2019.

[28] M. Chen, Y. Liu, and X. Yu, "NLPMM: A next location predictor with Markov modeling," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2014, pp. 186–197.

[29] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, "Next location prediction within a smart office building," Cognit. Sci. Res. Paper-Univ. Sussex CSRP, Brighton, U.K., Tech. Rep. 577, 2005, p. 69.

[30] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[31] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[32] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 194–200.

[33] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "GMOVE: Group-level mobility modeling using geo-tagged social media," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1305–1314.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[36] M. Granovetter, "Economic action and social structure: The problem of embeddedness," *Amer. J. Sociol.*, vol. 91, no. 3, pp. 481–510, 1985.

[37] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 1082–1090.

[38] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and K. Jon, "Inferring social ties from geographic coincidences," in *Proc. Nat. Acad. Sci. USA*, vol. 107, no. 52, pp. 22436–22441, 2010.

[39] J. L. Toole, C. Herrera-Yaqüe, C. M. Schneider, and M. C. González, "Coupling human mobility and social ties," *J. Roy. Soc. Interface*, vol. 12, no. 105, Apr. 2015, Art. no. 20141128.

[40] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, 2001.

[41] D. Boston, S. Mardenfeld, J. Pan, Q. Jones, A. Iamnitchi, and C. Borcea, "Leveraging Bluetooth co-location traces in group discovery algorithms," *Pervas. Mobile Comput.*, vol. 11, pp. 88–105, Apr. 2014.

[42] N. Eagle and A. S. Pentland, "Eigenbehaviors: Identifying structure in routine," *Behav. Ecol. Sociobiol.*, vol. 63, no. 7, pp. 1057–1066, May 2009.

[43] C.-B. Nguyen, S. Yoon, and J. Kim, "Discovering social community structures based on human mobility traces," *Mobile Inf. Syst.*, vol. 2017, pp. 1–17, Jul. 2017.

[44] I. O. Nunes, C. Celes, P. O. S. V. de Melo, and A. A. F. Loureiro, "GROUPS-NET: Group meetings aware routing in multi-hop D2D networks," *Comput. Netw.*, vol. 127, pp. 94–108, Nov. 2017.

[45] E. Bulut and B. K. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2254–2265, Dec. 2012.

[46] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.

QUAN T. NGO received the B.S. degree in electronics and telecommunications engineering from The University of Danang–University of Science and Technology, Da Nang, Vietnam, in 2016. He is currently pursuing the M.S./Ph.D. degree with the Department of Electrical, Electronic and Computer Engineering, University of Ulsan, South Korea. His research interests include computer vision and human mobility prediction.

DOI THI LAN received the B.S. and M.S. degrees in electrical and electronic engineering from Le Quy Don Technical University, Hanoi, Vietnam, in 2013 and 2018, respectively. She is currently pursuing the Ph.D. degree in electrical, electronic and computer engineering with the University of Ulsan, South Korea. Her research interest includes applying machine learning to human mobility prediction.

SEOKHOON YOON (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science and engineering from the State University of New York at Buffalo (SUNY Buffalo), in 2005 and 2009, respectively. After receiving the Ph.D. degree, he has worked as a Senior Research Engineer in the defense industry, where he designed several tactical wireless network solutions. He is currently a Professor at the University of Ulsan, South Korea, where he leads the Advanced Mobile Networks and Intelligent Systems Laboratory. His research interests include opportunistic networking, human mobility, intelligence defined networking, and machine learning-based IoT services.

WOO-SUNG JUNG (Member, IEEE) received the dual B.S. degree in electrical and computer engineering and information and computer engineering and the M.S. and Ph.D. degrees from the School of Computer Engineering, Ajou University, South Korea, in 2007, 2009, and 2015, respectively. He is currently a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His current research interests include wireless networking, the Internet of Things, device-to-device communication, and embedded systems.

TAEHYUN YOON received the B.S., M.S., and Ph.D. degrees from the School of Electronics Engineering, Kyungpook National University, in 2005, 2007, and 2015, respectively. Since 2016, he has been a Researcher with the Intelligent Robotics Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His current research interests include applied mobile communication, ship-IT convergence, and LoRa networks.

DAESEUNG YOO received the B.S., M.S., and Ph.D. degrees from the Department of Computer Engineering and Information Technology, University of Ulsan, in 1998, 2001, and 2011, respectively. Since 2009, he has been a Research Engineer with the Intelligent Robotics Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His current research interests include applied software engineering, mobile communication, ship-IT convergence, and *ad-hoc* networks.