JCSE Journal of Computing Science and Engineering

Regular Paper
Journal of Computing Science and Engineering,
Vol. 14, No. 2, June 2020, pp. 52-65

Trajectory Pattern Construction and Next Location Prediction of Individual Human Mobility with Deep Learning Models

Dabin You
Shinhan Investment Corp., Seoul, Korea
mongxmongx2@gmail.com

Ha Yoon Song*
Department of Computer Engineering, Hongik University, Seoul, Korea
hayoon@hongik.ac.kr

Abstract
Many modern portable devices, especially smartphones, are equipped with positioning functionality. The rapid growth in the use of such devices has allowed for the accumulation of a vast amount of positioning data. Combined with deep learning methods, these data may be used for many novel applications. Herein, a trajectory pattern tree generation method via deep learning is proposed. The convolutional neural network (CNN) and recurrent neural network (RNN) model of deep learning were applied for trajectory generation and prediction. Several volunteers provided their raw positioning data. The trajectory generation and prediction are for individual mobility patterns and were performed for every volunteer. We present the results obtained from seven volunteers. The preciseness of prediction can be measured both for CNN and RNN. Consequently, we can predict an individual's location with 32.98% accuracy, and predict the top-five up to 69.22% for unit area size of 0.030 km^2.

Category: Information Retrieval / Web

Keywords: Next location prediction; Mobility model; Deep learning; Convolution neural network; Recurrent neural network; Trajectory pattern

I. INTRODUCTION

Technological advances have enabled the near-ubiquitous presence of the global positioning system (GPS) in personal hand-held devices, including smartphones and smartwatches. This implies that enterprises can acquire consumers' positioning and mobility data. Mobility data are raw data comprising latitude and longitude information and time. Therefore, such geopositioning datasets are utilized in functions such as recommendations of nearby restaurants, navigation, and sports.

Location-based service (LBS) provides various services and information based on locations of objects. LBS includes navigation, location search, location-based advertisements, infotainment, senior or disabled person care, disaster situation control, finance, logistics, shopping, game, public transportation services, and so on. The availability of positional and mobility data also invites social and engineering analysis of the trajectory patterns. Especially, predicting the next location of objects based on trajectory history can improve the classes of LBS and the quality of LBS.

Open Access http://dx.doi.org/10.5626/JCSE.2020.14.2.52 http://jcse.kiise.org

Received 20 May 2020; **Accepted** 29 May 2020
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

As discussed in Section II-A, several related kinds of research have been reported in the area of location prediction with the mining of mobility data, analysis of mobility data, and trajectory pattern generation. Most of the reported studies demonstrate the generation of the mobility pattern tree based on the visited locations and the frequency of visit and subsequently predict the future possible location of the visit of the objects. In addition, mobility patterns can be represented in a Markov chain with probabilistic approaches. Deep learning-based research exists for extracting mobility patterns.

However, the previous trajectory pattern prediction focused on the whole trajectory pattern of mobile objects, and could not predict the next location of objects in real-time.

In this study, we generate a model that can predict the next location in real-time using deep learning technology. Particularly, we will use convolutional neural networks (CNNs) and recurrent neural networks (RNNs) that are classification models in deep learning. CNNs and RNNs demonstrate good classification performance and they are generally used in many other fields.

In the case of the next location prediction from the raw geopositioning dataset, the two notable properties are the order of location transition and transition to the adjacent location. CNN and RNN are such models that can cope with these two properties.

CNN can be used as a learning model because it learns data in association with adjacent information.

In our case, as location prediction with adjacent location information is of clear benefit, we have introduced the CNN. Another candidate of the learning model is the RNN. RNNs are typically used for sequential data with dynamic input and prediction. In our case, sequential-data-oriented processing is another benefit because a mobility pattern is generated sequentially over time. Although deep learning models such as the CNN or RNN require a long time for the training process, prediction with CNN and RNN requires less processing time. Therefore, our model can demonstrate real-time service for users. Previous methods using the trajectory pattern tree for the next location prediction can also provide services in real-time [1]. In the new trajectory, the pattern is absent in the tree structure, hence no predictions can be performed. However, probabilistic prediction models rather than static structure models such as the deep learning model can cope with such situations whereby prediction cannot be performed by the static structure model. In the current work, we aim to create a more flexible and error-tolerant model compared to previous works.

It is necessary to build individual mobility models for each volunteer and to predict for each volunteer, as the LBS is essentially individual-oriented. Even though individuals can move within a group, they possess their favorite locations and their characteristics of mobility. Thus, we focused on individual prediction in this research. In addition, we will prepare the methods to measure the accuracy of the next location prediction and measure the accuracy of trajectory prediction and trajectory pattern generated by a trajectory prediction model.

Section II discusses the classification models of deep learning as a core background of our approach, and the related works in detail. In Section III, we present the algorithm to predict the next location and to generate a trajectory pattern. Section IV describes the method of accuracy measurement for the predicted trajectory. In Section V, the positioning data collection, experimental environment, and detailed structure of network models are discussed. Section VI presents the result of trajectory prediction and trajectory pattern generation with deep learning models. Section VII presents conclusions and discusses the possible related topics for future research.

II. RELATED WORKS

Previous investigations on this subject focused on the tree generation of trajectory patterns by data mining techniques to predict the next location or trajectory pattern generation by establishing a Markov chain from the probabilistic approach.

Predicting the next location of objects with mobility sequence tree generation by pattern mining the objects' mobility is a typical approach in this area [1-3]. An incremental approach exists where initially, the mobility tree is expanded continuously by pattern mining [4], followed by continuous pattern mining. Generating a trajectory with pattern mining is based on an *a priori* algorithm. Transitions from the start location to end location can form linked lists. For all the previous locations, highly visited transitions can be generated by removing less frequently visited transitions. Therefore, trees can be generated with highly visited transitions. With the generated tree, the most frequently visited location could be predicted.

The LBS can also be utilized to generate mobility patterns [5]. Such methods, including mobility pattern tree generation and next location prediction, are typically combinations of the probabilistic approach and data mining, as shown in [6, 7]. Some researchers have generated graphs representing the mining results of frequent past trajectories to predict the next location based on the graph [8, 9].

The trajectory pattern itself can be utilized for the mining trajectory without trees, graphs, or probability models [10].

Location prediction in combination with frequent trajectory and mobility rule can be achieved [11]. For example, the measure of the frequency of location visits can be defined as shown in [12].

From the view of location prediction, various methods and situations can be assumed such as a disaster. Mobility prediction in case of disaster using particle filter has been demonstrated [13]. Contrary to our research, this research deals with group mobility. A similar mobility pattern between two users can be used to predict the location of the user using collaborative filtering as shown in [14].

Moreover, research results exist about group mobility as shown in [15] and [16]. The results were of general prediction for the wider area and massive group mobility.

Markov chain based results can be found in [17-19] which represents the human mobility model in a form of Markov chain to show continuous human mobility pattern. A Markov chain-based approach allows prediction of the next location based on human mobility pattern, as previously demonstrated [20]. The hidden Markov model (HMM)-based prediction shown in [17] utilizes a transition matrix with seven locations and nine combinations of pairs of previous locations and employment of 20 locations [18]. As discussed, the trajectory pattern tree, Markov chain, and data mining are major tools for past research.

However, the big data analytics approach is now required as the volume of mobility data increases daily. A current candidate for big data processing is deep learning [21]. Deep learning is a branch of machine learning, especially with neural networks. Research regarding trajectory pattern generation using a deep autoencoder has been reported [22]. Further, trajectory pattern prediction with network structures has been investigated [23, 24]. A reported work demonstrates the generation of a matrix of mobility flow [25]. This work utilizes CNN to predict group mobility in urban areas such as the central part of New York. A study used exogenous variables at present and past time and tried to predict the variable at a given time [26]. On the contrary, a study removed present time data of prediction time and added convolution layer to their model for prediction, thereby resulting in time-shifted training of actual data [27]. Other works based on deep learning [28] pertain to mobility prediction under disaster situations. The time, location, and disaster information are used for the prediction. In our research, we used continuous geolocation data apart from the above research results which show the time-shifted results for prediction of continuous trajectory or predicted discontinuous trajectory.

The purpose of our research is to predict individual locations in typical and general situations. We will solve these problems using the general deep learning classification model such as CNN and RNN. With the smallest preprocessing of data, the CNN and RNN based prediction of the objects' trajectories will be presented.

III. BACKGROUNDS

A. Mobility Data and Trajectory Pattern

Mobility data are data containing the location or positioning information of objects. For example, GPS data contains the latitude, longitude, and time information. Positioning devices such as a GPS receiver or smartphones can collect mobility data. Several positioning mechanisms exist such as GPS, GLONASS, WPS (Wi-Fi-based positioning system), and Bluetooth-based positioning systems. The trajectory is defined as shown in Definition 1 [6].

DEFINITION 1. *A Trajectory or Spatio-Temporal Sequence is a Sequence of Triples*

$$T = \langle x_0, y_0, t_0 \rangle, \ldots, \langle x_n, y_n, t_n \rangle \tag{1}$$

where (x_i, y_i) *are points in* R^2 *and* $t_i < t_{i+1}$, $(0 \le i \le n)$.

In this research, $\langle x_i, y_i \rangle$ denotes geoposition (location) and t_i denotes time.

Fig. 1 shows how Definition 1 can be visualized concerning the change in location as a function of timestamp. The locations are changed according to the timestamp. The trajectory contains the positioning and location information, for example, restaurant, home, school, etc. The trajectory represents the mobility of objects in the form of location data or positioning data sequences like Fig. 2.

The trajectory pattern is a pattern that can be observed from the typical or repetitive trajectory of multiple objects. The trajectory pattern can be generated by various methods that can subsequently generate different trajectory patterns from the same trajectory set.

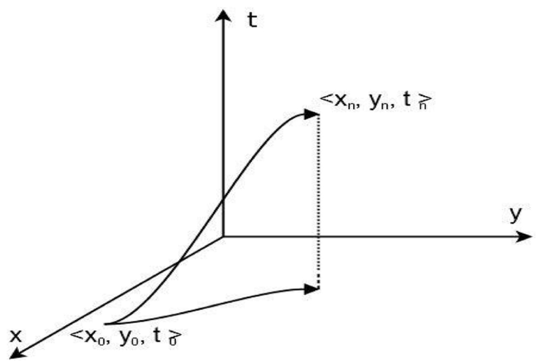

Fig. 1. The three-dimensional (3D) positioning data.

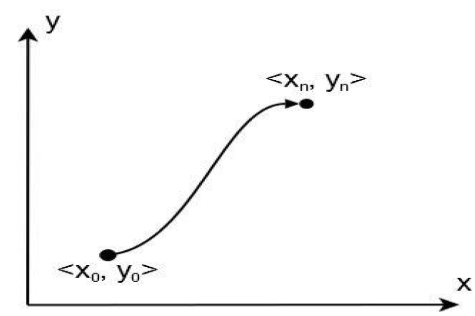

Fig. 2. The two-dimensional (2D) positioning data.

B. Classification Model of Deep Learning

Deep learning originates from neural networks in the artificial intelligence field. Multiple stacks of neural networks comprised of deep layers of networks and learning on this network are called deep learning. It is used primarily in image classification, speech recognition, and natural language processing because it can excellently perform in the fields that require classification. Deep learning models can be classified according to their usage. A classification model classifies the input data according to the pre-trained results based on the input data for training. A generative model generates new data based on trained data. To predict the object's next location, a classification model is useful to establish a trajectory pattern because we wish to predict the object's next location about the new input data using a model that is trained by the previous data. For the deep learning classification model, several models exist. The distinguished models are class-deep belief networks [29], feedforward neural networks [30], CNNs [31], and RNNs [32]. The previous research [22] described in Section II-A utilizes the generative model as a deep autoencoder for trajectory pattern mining.

However, we will utilize the CNN and RNN in this research. As shown in [31] and [32], CNNs and RNNs are excellent models that are widely used in both research and real-world applications; they are used for image classification and sequential data processing. We will process the trajectory data according to the properties of these models for model training and prediction.

1) Convolution Neural Network

CNN is a model for deep learning that classifies input data through training. For example, CNN can classify images such as the classification of a cat from a dog or classification of numbers from manuscripts. With input data which is not used to train, the CNN classifies such input data based on the trained information. The CNN is used widely compared to other deep learning models as it performs better than other models, and is used primarily for ImageNet, which is an image classification contest.

CNN filters the characteristics of neighboring data especially in big data such as images; thus, a small amount of data with characteristics can be fed to the fully connected neural network (FNN), which is a core part of CNN. The classification process of CNN is also good for mobility data processing towards trajectory classification. Particularly, the segmentation of geographical areas for mobility pattern processing is suitable for CNN. Our mobility trajectory is preprocessed such that the trajectory can be represented in a two-dimensional matrix form reflecting the mobility on the map. Therefore, the trajectory over time remains in the preprocessed data, and the trajectory is a set of adjacent locations of mobility. Such adjacency of location and the relationship with adjacency renders the CNN as a good candidate to be applied.

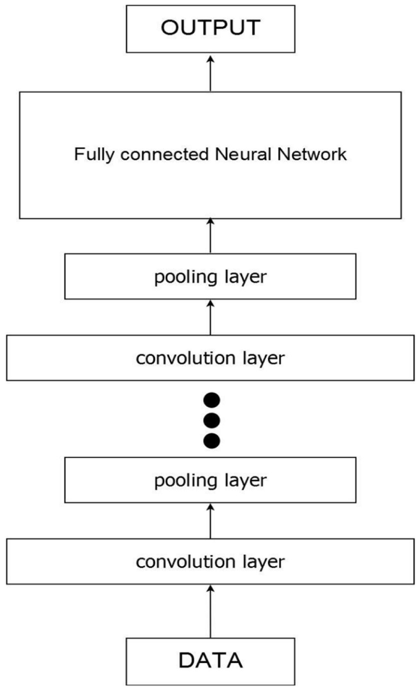

Fig. 3. Convolutional neural networks.

The CNN structure is shown in Fig. 3. The convolution layer, pooling layer, and feedforward neural network are combined to structure the CNN. Once the training data are fed to the CNN, repetitive layers of the convolution layer and pooling layer decrease the data size. On the convolution layer of the CNN, various filters are applied to predict the connectivity of the adjacent data and subsequently passed to the next layer. Input data are reduced by max-pooling and feeds the data to the next layer. The decreased data are subsequently fed to the neural networks. The error function can be utilized to the interlayer parameter, W (weight) and b (bias). After the training, the trained CNN can classify images with adjusted parameters. The CNN adjusts the parameters between layers from the input data, which constitutes training the input data. It classifies another input data based on the trained model and adjusted parameters. Therefore, CNN requires data for the training and classification of other data.

The prediction procedure of CNN trained by data is as follows:

- The new input data are reduced by the convolution layer and pooling layer, similar to the training procedure, and inputted to the FNN.
- Bypassing the output layer of the FNN, the data are classified to the maximum likelihood label with the softmax function.
- Labels can represent prediction.

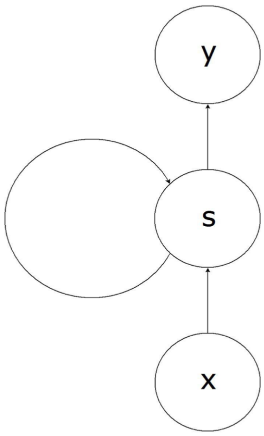

Fig. 4. Recurrent neural networks.

In this research, we have used the trajectory data as inputs for the CNN training and established the predicted locations and trajectory patterns with the new mobility data.

2) Recurrent Neural Network

The RNN is suitable for sequential data processing such as mobility trajectory identification and serves as a classification model. Thus, RNN is another candidate for our research because the mobility trajectory exists on the time domain, and time adjacent movements compose the sequential pattern. Both the past trajectory and current movement can be used for trajectory prediction.

The RNN structure is shown in Fig. 4. In Fig. 4, past data x and current data s can be fed as inputs to the current state, i.e., memory exists in the RNN. Among various RNN models, we have chosen the long short-term memory (LSTM) model [32]. LSTM has a long-term memory compared to the basic RNN.

IV. PREDICTION OF OBJECT'S TRAJECTORY AND CREATION OF TRAJECTORY PATTERN

A. Preprocessing Mobility Data

As the dataset was inappropriate for training in its raw form, certain preprocessing was required. The trajectory data was unstructured and could not use the directory for the deep learning model because they contain no data format in their initial form. Therefore, they are required to be preprocessed before applying the trajectory data to use for the deep learning model.

As using three-dimensional data increases the data volume significantly, an area partitioning method was used to reduce the data. To reduce the data volume, the first-hand approach is area partitioning. The map was partitioned and labeled as shown in Fig. 5; subsequently, the movement trajectory was represented as sequences of

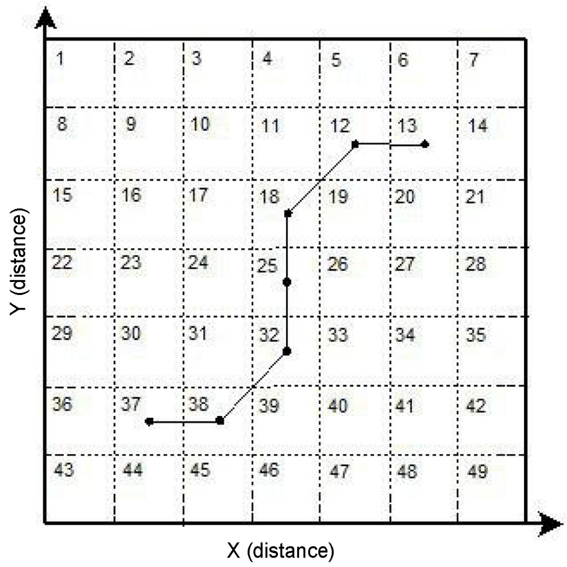

Fig. 5. Area partitioning and labeling.

label changes. The trajectory that is a sequence of labels can be applied as the training data.

Training with part of the trajectory instead of the whole trajectory was performed to understand the mobility based on our models. Consequently, it was required to partition the whole trajectory into multiple sets. In detail, a sequence of movements from the starting point composes one partition. Each data was labeled as the next location. The trajectory length affects mobility data preprocessing. Naturally, a large number of visits compose a longer trajectory and larger input data.

EXAMPLE 1. Suppose we have the following set of labels representing movements.

[1, 2, 3, 4, 5, 6]

Table 1 shows an example of the preprocessing of mobility data. Once trajectory [1, 2, 3, 4, 5, 6] is obtained, the mobility data can be preprocessed to contain one trajectory and the next location as labels. In this case, [1, 2, 3, 4, 5, 6] can be preprocessed as Table 1 entries. For example, [1, 0, 0, 0, 0, 0] is the first trajectory of the object's trajectory, and it will move to area label 2. Therefore, the number of preprocessed data from one mobile data will be preprocessed as much as the length of the mobile data. The preprocessing procedure is formalized as shown in Algorithm 1.

- Path is the input variable representing trajectory path.
- Line 1: ProcessedPath is defined as the variable of $N \times N$ matrix, where N is the square root of the possible longest number of trajectory or the number of maximum prediction. For example, N is 3 once we obtain a trajectory length of 9. Initially, they are

Table 1. Example of mobility data preprocessing

Trajectory	Predict
(1, 0, 0, 0, 0, 0)	2
(1, 2, 0, 0, 0, 0)	3
(1, 2, 3, 0, 0, 0)	4
(1, 2, 3, 4, 0, 0)	5
(1, 2, 3, 4, 5, 0)	6
(1, 2, 3, 4, 5, 6)	0

filled with zeros.

- Line 2: Repetition will be performed to line 8, until the length of Path.
- Line 3: Generate ProcessedPath with Path until the length of Path to line 5.
- Line 6: The generated ProcessedPath is appended to ProcessedPathSet.
- Line 7: The next Path will be appended to the Label set.
- Line 9: ProcessedPathSet and Label will be returned.

Algorithm 1 PreProcessing

Input: Trajectory Path $Path$
Output: A Set of Trajectory ProcessedPathSet $ProcessedPathSet$, A Set of Label $Label$
PreProcessing($Path$) {

1: $ProcessedPath = \text{array.zeros}(N * N)$
2: **for** $(j = 0;\ len(Path);\ j{+}{+})$ **do**
3:     **for** $(i = 0;\ i < j;\ i{+}{+})$ **do**
4:         $ProcessedPath[i] = Path[j]$
5:     **end for**
6:     $ProcessedPathSet.\text{append}(ProcessedPath)$
7:     $Label.\text{append}(Path[i + 1])$
8: **end for**
9: **return** $[ProcessedPathSet, Label]$
10: }

Thus, the preprocessing Algorithm 1 prepares the matrix variable of ProcessedPathSet and a set of Label's for further processing can be padded as zero (0) to satisfy the size restriction of the input matrix. On the contrary, in the case of an RNN and a short trajectory, no padding is required as the RNN is not concerned with the input length.

However, in CNN, the input data is limited to the same size for each matrix, thus the maximum length must be restricted. The same restriction is also applied to the RNN model with the restriction of the maximum length. In the case of a CNN and a short trajectory, the non-existing values can be padded as zero (0) to satisfy the size restriction of the input matrix. On the contrary, in case of an RNN and a short trajectory, no padding is required as the RNN is not concerned with the input length.

B. Prediction of Object's Trajectory and Creation of Trajectory Pattern

Based on the mobility data input, the optimized next location will be predicted by the trained model. Algorithm 2 shows the procedure of the next location prediction. For the trajectory input, the model predicts the next location. The predicted location will be appended to the rear of the first input trajectory. Additionally, the prediction can be repeated as desired. The repetition of the next location eventually generates the trajectory pattern that contains both the observed location and predicted location.

- Line 1: Append Path to existing trajectory PredictedPath.
- Line 2: Repetitions will be made n times where n is the desired number of prediction.
- Line 3: Using the prediction function of the CNN, predict the next location for Path.
- Line 4: Append label of the predicted path to the existing trajectory PredictedPath. PredictedPath can be reused for the next iteration.

Algorithm 2 PredictedPath

Input: Existing Trajectory Path $Path$, Desired length of prediction n, Existing Trajectory PredictedPath
Output: Predicted Trajectory $PredictedPath$
PredictPath(Path){

1: $PredictedPath.\text{append}(Path)$
2: **for** n **do**
3:     $PredictedLabel = \text{CNN.Predict}(PredictedPath)$
4:     $PredictedPath.\text{append}(PredictLabel)$
5: **end for**
6: **return** $PredictedPath$
7: }

The mechanism of this algorithm is depicted in Fig. 6. The inputs and outputs of the prediction by CNN are included in Fig. 6, where the values of the input and output are actual values observed during the experiment. The trajectory of [1, 2, 3, 4, 5] was used for the input of the CNN, where the reserved space was filled with zeros for the predicted locations. Therefore, the label of the current location is 6. The actions will be repeated by Algorithm 2 for the desired times.

Apart from the next location prediction of CNN, the prediction by the RNN uses the structural prediction process. Fig. 7 depicts the prediction mechanism by the RNN. The RNN's next location prediction is performed with the current location and past location.

V. ACCURACY MEASUREMENT OF PREDICTED TRAJECTORY PATTERN

For the measurement of the accuracy of prediction in

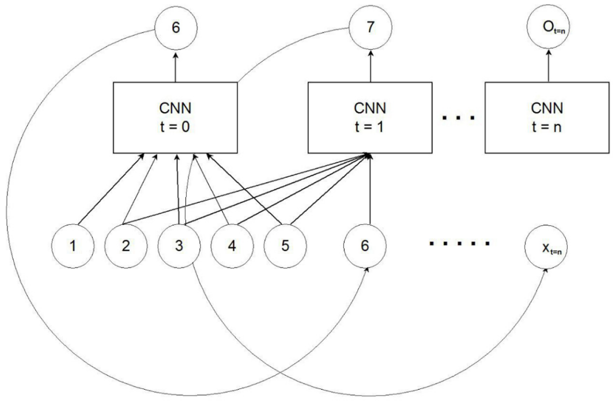

Fig. 6. Prediction in convolutional neural networks.

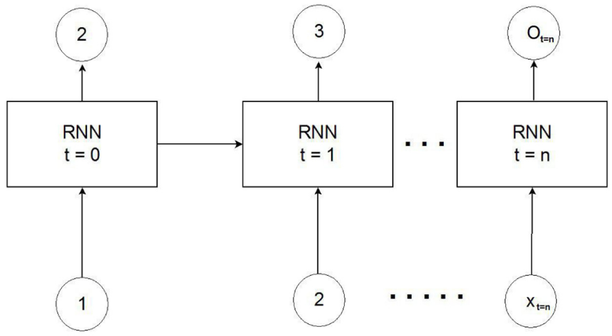

Fig. 7. Prediction in recurrent neural networks.

our research, two methods were selected to measure the accuracy of three distinguished cases of location prediction.

- CASE 1: Actual location and predicted location are the same.
- CASE 2: Predicted location is not the same as the actual location but the same as the adjacent location of the actual location.
- CASE 3: Predicted location is not related to the actual location.

Considering various cases, the accuracy measure, which covers only CASE 1, is insufficient to present the model performance. An additional method is required to represent the prediction performance that also includes CASE 2 and CASE 3. To include such cases of location prediction, a measure called Top-k is introduced as a generalized method for evaluating the performance of the classification model [31]. Top-k, where k is a predefined constant, verifies if the actual data is present in a set of predicted data, and the set of predicted data is selected according to the highest prediction probability where the cardinality of the set is k, i.e., Top-k verifies if the predicted data with the k topmost prediction probability matches the actual data. Thus, Top-k demonstrates a wider prediction performance of the model instead of the accuracy only. We set k as five and, along with the accuracy, measured the performance of prediction by verifying if the five predicted locations can contain the actual location in trajectory.

VI. PREPARATION OF EXPERIMENT

Concerning the frequent location of the volunteers, we trained the model for the mobility data of one whole day. Seven volunteers carried their own positioning devices for the individual positioning data collection.

The area was limited to the size of 3.61 km in vertical length and 4.82 km in horizontal length. We partitioned the location area as 24 by 24 rectangles. Each partitioned location area was in the range of 150.42 m (vertical) × 200.83 m (horizontal). A total of 576 partitions were generated and the label for each partition was assigned. Based on the presence of a specific area where volunteers were typically located; we chose that area. Virtually, every possible size can be used for partitioning the area, and we estimated that less than 200 m is of sufficient resolution for our purpose.

The data used in this research are primarily the positioning data. The positioning data are also called as the geolocation data. To collect the positioning dataset, devices such as a GPS receiver or smartphone with applications of such functionality are required. In our research, we used smartphones with the application known as “Sports Tracker”. The volunteers carried their smartphones with functioning Sports Tracker application. Each volunteer’s dataset contains the mobility state of walking, running, public transportation, etc.; therefore, the dataset contains nearly every possible status of mobility. The Sports Tracker collects one or two positioning data in one second when it senses high mobility; otherwise, it collects one positioning datum in three seconds at the maximum. The frequency of positioning data collection affects the learning process of the mobility trajectory. However, our method is not affected by positioning frequency as we treat mobility as changes in position regardless of time span. The total sum of the collected positioning data is 1.5 GB. Even though we assumed continuous collection of positioning data, it cannot be accomplished, owing to various reasons such as insufficient battery level of the smartphone, personal life protection, forgetting the activation of “Sports Tracker”, urban canyon, and underground transportation. We used such non-continuous dataset as it is, because it is reasonable to use the data collected in a real situation. Services based on our research might encounter a similar situation of data shortage, which is almost inevitable. We assume that the experimental result will indicate better performance once we obtain the ideal 24-hour daily dataset. For the mobility model of seven volunteers, each model was trained by each volunteer’s data, independently. In other words, each volunteer had an individually trained dedicated model. The positioning dataset from only one individual is sufficient to verify the location prediction by deep learning if the dataset is continuous and contains sufficient data. It is important to possess sufficient data to train deep learning models. In our case of supervised learning, the difference between real data and predicted data is the key for learning. However, we used the datasets from seven volunteers as the sets were intermittently collected and contained a relatively smaller number of data than expected, which is typical in an actual situation. In such situations, we can compare datasets from different volunteers and analyze the result from each dataset in terms of accuracy and Top-k, to determine if the deficient dataset is suitable for prediction. It is interesting that even with a smaller number of positioning data for training, the prediction sometimes achieved better performance compared to a larger number of training sets.

Compared to the previous methods, 576 locations were used in our research and such several data renders our model more general for prediction. Our deep learning model-based approach demonstrates prediction in terms of a large number of locations, non-restrictive location coverage, and flexibility of location size, all of which can be accomplished by a simple setup of partition size of a given area.

On the networks for deep learning, the architectures for three CNN models and three RNN models were proposed, and every model was pre-tested. From the pre-test experiment, the accuracy and top-5 were measured using the weekday data of volunteer 3 with input size 576 and output size 576. Table 2 shows the model name, the architecture of each model, accuracy, and the top-5. CNN1 and RNN2 demonstrate the best performance in their groups. Thus, CNN1 and RNN2 will be our architecture henceforth, and more experiments will be carried out based on these two models.

- Learning rate: 0.01
- Activation function: tanh (hyperbolic tangent)
- Mini-batch size: 100
- Gradient descent optimization: ADAM (adaptive moment estimation) optimizer

The experiment was performed separately for weekdays and weekends. The result can be visualized in maps. On the machine having a GPU of GTX980 on Ubuntu 15.04, Python and Theano were used for the actual experiment.

Regarding the prediction speed of our model, 0.10 second was required to predict data size one for both CNN and RNN. For the prediction of 24,419 data size, it took 1.38 seconds. We concluded that the time required for prediction was short enough for the actual implementation of location prediction service. It enables real-time service for prediction. The execution time and the prediction size are not linearly proportional because of data transfer time from the main memory to GPU memory. To use GPU, data must be transferred from main memory to GPU memory preceding to GPU's processing which spends most of the execution time in case of a small number of prediction size.

VII. RESULTS

A. One Step Prediction

The detailed results are summarized in Tables 3, 4, and Fig. 8. For seven volunteers, the details of the data and results are for only one next location prediction.

The one-step prediction does not simply predict the eight neighboring cells. Non-continuity could be present in the positioning data collection. However, it is also possible in our method to predict the location even when a gap exists in the mobile trajectory. Therefore, even a random guess of the one-step prediction does not satisfy the probability of prediction as 0.125.

Table 3 shows the ID of the volunteers, the number of train sets, the number of test sets, the accuracy of prediction, and the top-5 of the prediction. Apart from the accuracy, the misprediction ratio ignores the near prediction but pertains only to the correct prediction compared to the actual trajectory. Similarly, Table 4 shows the details of the training data and the prediction result for movements on weekends. The result for Volunteer 5 could not be observed because the volunteer had no movement on weekends.

The case of weekdays for Volunteer 1 demonstrates the CNN accuracy of 34.05% and the top-5 of 83.07%. For

Table 2. Accuracy of predicted trajectory on model architecture

Model	Architecture	Accuracy	Top-5
CNN1	Cov(5×5) - MaxPool(2v2) - Cov(5×5) - MaxPool(2v2) - FFN(576)	22.17	64.54
CNN2	Cov(5×5) - Cov(3×3) - MaxPool(2×2) - FFN(576)	16.70	40.13
CNN3	Cov(5×5) - Cov(5×5) - MaxPool(2×2) - FFN(576)	18.93	51.60
RNN1	LSTM(24×24) - LSTM(300) - FFN(576)	20.05	48.95
RNN2	LSTM(24×24) - LSTM(24×24) - FFN(576)	20.05	50.00
RNN3	LSTM(24×24) - LSTM(24×24) - FFN(576)	20.05	48.95

Table 3. Accuracy of predicted trajectory in weekdays

Volunteer ID	Data size (day)	CNN		RNN	
		Accuracy	Top-5	Accuracy	Top-5
Volunteer 1	24,419 (452)	34.05	83.07	23.35	60.60
Volunteer 2	19,117 (372)	34.34	81.25	34.16	81.08
volunteer 3	11,885 (141)	22.17	64.54	20.05	48.95
Volunteer 4	2,701 (117)	29.31	79.04	27.60	71.60
Volunteer 5	4,511 (177)	24.50	63.30	21.44	51.56
Volunteer 6	878 (143)	24.34	68.18	20.00	60.63
Volunteer 7	1,786 (99)	18.77	49.30	18.67	53.67

Table 4. Accuracy of predicted trajectory on weekends

Volunteer ID	Data size (day)	CNN		RNN	
		Accuracy	Top-5	Accuracy	Top-5
Volunteer 1	6,239 (166)	31.49	76.28	25.25	57.75
Volunteer 2	4,669 (150)	33.40	84.48	34.11	81.00
volunteer 3	2,174 (52)	11.03	36.78	12.75	45.50
Volunteer 4	937 (25)	27.18	75.40	20.00	58.00
Volunteer 5	61 (38)	30.77	69.23	10.00	90.00
Volunteer 6	354 (43)	12.68	29.58	14.29	34.29
Volunteer 7	6,239 (166)	31.49	76.28	25.25	57.75

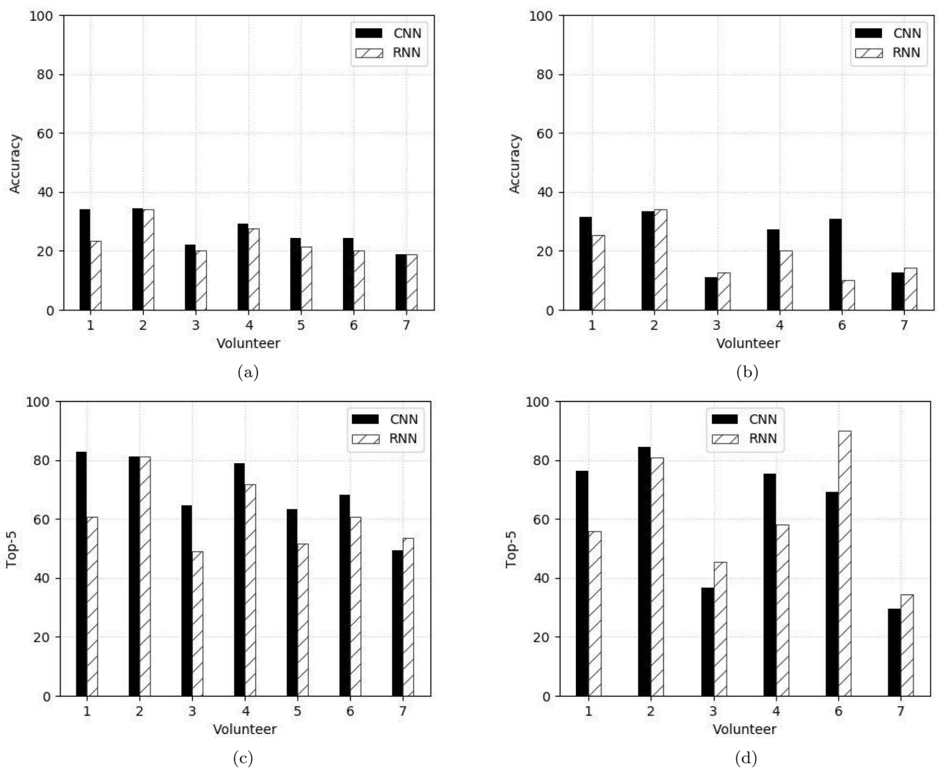

Fig. 8. Comparison of accuracy and top-5 for CNN and RNN: accuracy for weekday data (a) and weekend data (b); top-5 for weekday data (c) and weekend data (d).

the RNN, an accuracy of 23.35% and top-5 of 60.60% are demonstrated. Both the CNN and RNN results are credible as this experiment was performed with sufficient data.

Table 3 shows that the top-5 of the next one location prediction of the weekday's movement is in the range of 49%–83%. Table 4 also shows that the top-5 of the next one location prediction of the weekend's movement is in the range of 29%–84%, while the lowest accuracy is not credible owing to the small number of train sets. The RNN-based prediction was in the top-5 with 51%–81% during the weekdays and top-5 with 34%–90% in the weekend.

A noticeable tendency is that a higher number of train set implies a higher accuracy. It implies that proper training can be performed based on a sufficient number of samples. Tables 3 and 4 can be compared in this regard. The accuracy is similar while the top-5 is vastly different although they have a similar number of train sets and test sets. This implies the arbitrary nature of human mobility. Volunteer 6 exhibits more randomness in terms of mobility.

Fig. 8 shows the comparison accuracy and top-5 by the CNN and RNN. In Fig. 8(a), the weekday case indicates a higher accuracy of the CNN-based prediction compared to the RNN-based prediction. The weekend case in Fig. 8(b) shows that a similar accuracy pattern can be found for both CNN- and RNN-based predictions. In general, the CNN-based prediction demonstrates a higher accuracy than the RNN-based prediction. In terms of the top-5, Fig. 8(c) and 8(d) show a better top-5 by the CNN-based prediction compared to the RNN-based prediction except for Volunteers 3, 6, and 7.

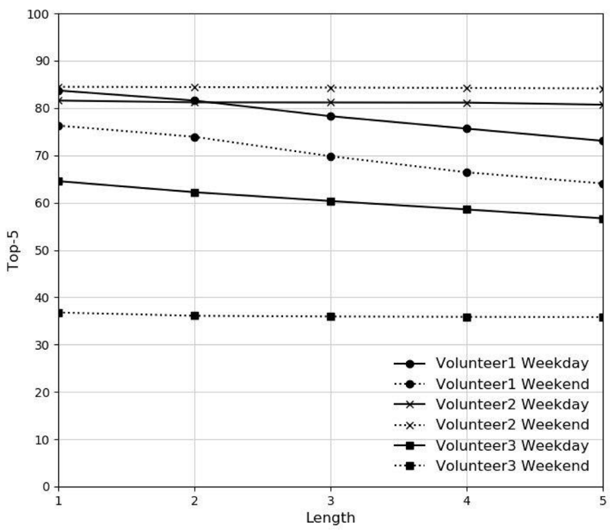

Fig. 9. Accuracy of predicted trajectory with multiple prediction depth.

In summary, CNN is proposed as a better model for trajectory prediction than the RNN with the top-5 of 49%–83%.

B. Multiple Step Prediction

The results in the previous subsection indicate better accuracy of the CNN in general compared to the RNN for the next location prediction. Subsequently, the result of the long-term prediction is interesting. Based on the basic results in Section VII-A, we conducted the prediction of trajectory pattern with multiple depths of up to five, only with the CNN. Volunteers 1 and 2 were chosen owing to their high one-step prediction top-5 and large number of train data, while Volunteer 3 was chosen to compare the effect of the number of train data on the multiple depth prediction.

As shown in Fig. 9 and Table 5, the top-5 of prediction exhibited a decrease, as expected, according to the degree of prediction.

However, the top-5 of prediction with multiple depths depends completely on the nature of the volunteers. Regardless of the number of data for training, Volunteers 1, 2, and 3 exhibited different accuracies of prediction. We may conclude this phenomenon as follows: The mobility of Volunteer 2 is stable both on the weekday's data and weekend's data. The mobility of Volunteer 1 tends to wander. The mobility of Volunteer 3 is rather stable on the weekend's data but active on the weekday's data.

For the five depth predictions, some of the results are visualized on the map. Figs. 10–13 show the results of the depth prediction of Volunteers 1 and 3 on weekdays. Each subfigure shows the map and trajectories over 24 × 24 grid partitions. In each subfigure, the number 1 dot

Table 5. Accuracy of predicted trajectory with multiple degrees of prediction

Volunteer ID	Degree	1	2	3	4	5
Volunteer 1	Weekday	83.70	81.57	78.26	75.64	73.06
	Weekend	76.28	73.91	69.81	66.41	64.04
Volunteer 2	Weekday	81.57	81.19	81.17	81.14	80.70
	Weekend	84.48	84.42	84.32	84.23	84.13
Volunteer 3	Weekday	64.54	62.20	60.34	58.56	56.67
	Weekend	36.78	36.09	35.94	35.86	35.81

represents the starting point of the actual trajectory, the last number dot represents the endpoint of the actual trajectory (current position), and the other number dot represents the past visited location. Figs. 10 and 11 are for Volunteer 1 and Figs. 4 and 6 are for Volunteer 3. Figs. 12 and 13 show the graphical result of the location prediction of Volunteer 1. As shown in Fig. 9, the overall prediction is fairly inaccurate owing to the misprediction

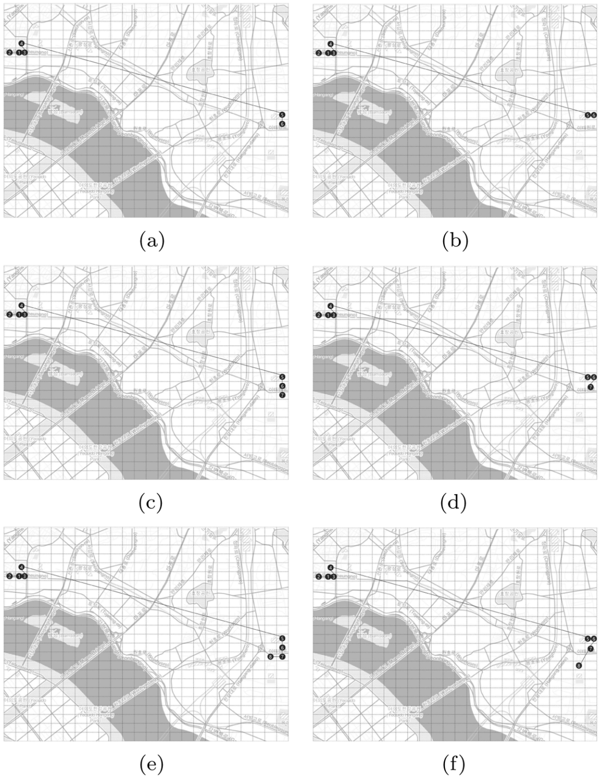

Fig. 10. Depth prediction of Volunteer 1 on weekday: (a, c, e) actual trajectory of depth 1, 2, and 3; (b, d, f) predicted trajectory of depth 1, 2, and 3.

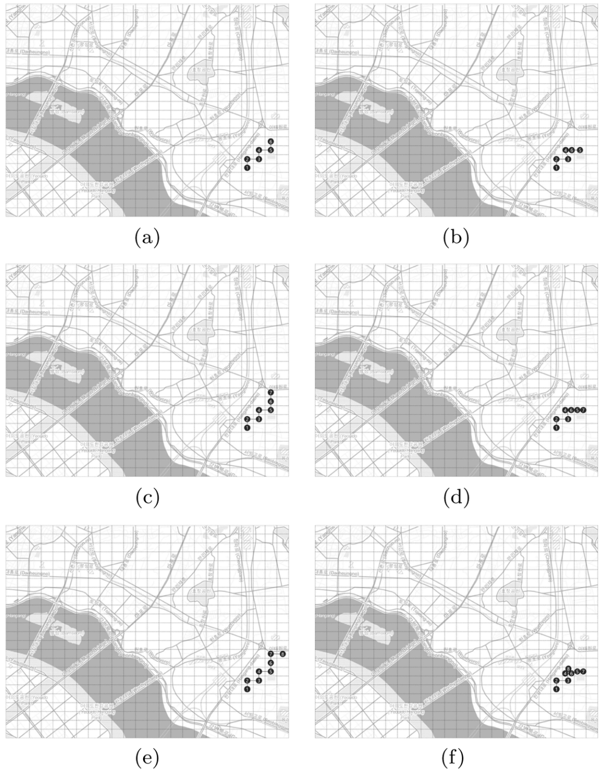

Fig. 12. Depth prediction of Volunteer 3 on weekday: (a, c, e) actual trajectory of depth 1, 2, and 3; (b, d, f) predicted trajectory of depth 1, 2, and 3.

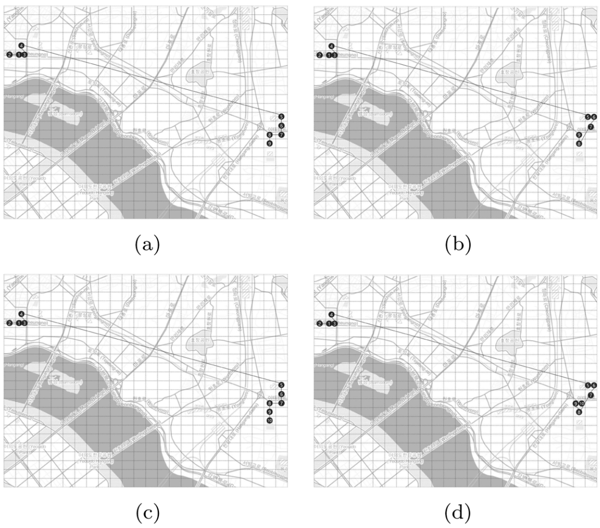

Fig. 11. Depth prediction of Volunteer 1 on weekday: (a, c) actual trajectory of depth 4 and 5; (b, d) predicted trajectory of depth 4 and 5.

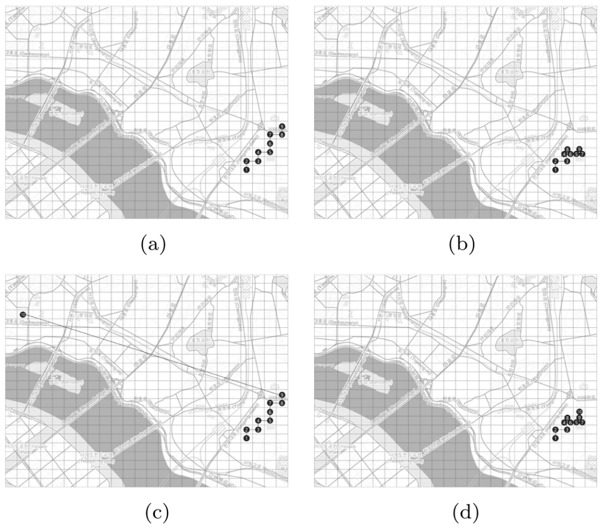

Fig. 13. Depth prediction of Volunteer 3 on weekday: (a, c) actual trajectory of depth 4 and 5; (b, d) predicted trajectory of depth 4 and 5.

in the first step, as depicted in the depth one case of Fig. 10. However, it is interesting that the trend of prediction is reasonable. The similar phenomena can be found for the prediction of Volunteer 3's location, as shown in Figs. 12 and 13.

Figs. 10 and 11 show non-continuous mobility. This is due to the problem of the positioning data collection because of the underground transportation, urban canyon, etc. In such an environment, the positioning data cannot be collected normally. However, prediction can be made regardless of the continuity of the positioning data.

VII. CONCLUSIONS

Based on the assumption of the trends of individual human mobility, we developed a method of human location prediction using a classification model.

We generated the trajectory pattern of the objects' trajectory and predicted the next location of objects using the CNN and RNN, which are classification models of deep learning. Instead of handling the map data of the positioning data, the trajectory information was used for training.

We verified the objects' trajectory generation and predicted the trajectory pattern. Additionally, the method of measuring the accuracy of model was provided and the model accuracy was measured. The essence of our research is that trajectory-based information can generate trajectory patterns using deep learning. However, the randomness of human mobility led to a relatively low accuracy of trajectory prediction. Also, the lack of human mobility data caused drawbacks in model training.

In our experiment, not all the volunteers gathered sufficient mobility data to train the model in a single day. In other words, the prediction of human location based on the daily positioning dataset would yield a lower accuracy of prediction. Further, some events caused significant human mobility such as moving in and out, newly constructed highway, and even Pokemon GO, thus hindering prediction based on the past mobility data. Even though we studied the human mobility data, studies on freight traffic will be able to show much more accurate results, as such mobility patterns may not include the randomness of human mobility.

The largeness of positioning data requiring big data processing could be reduced to the essential trajectory data by excluding meaningless positioning data; thus, our method results in less noise in the trajectory pattern. Therefore, we reduced the processing time of big data and the amount of data to be processed, implying that our method could be implemented on portable devices with typically low computational capability and smaller battery consumption. The next location might be able to be predicted in real-time with our method on portable devices, thereby widening the application area of LBSs.

We only focused on a single, individual object for the trajectory pattern and location prediction. One candidate for future research would be the trajectory pattern and location prediction of object groups. Additionally, the application of other deep learning models for better prediction will be another candidate.

Another combination of human mobility and human personality exists. Even though we only considered the individual's trajectory pattern, other factors for an individual exist such as age, gender, income, and personality. Among these factors, the human mobility model, which can be represented with the big five factors (BFF) personality model, may be significantly related to human mobility. For example, our result implies that Volunteer 1 is much more active, or may exhibit a higher Openness than Volunteer 3 in terms of the BFF. Volunteer 2 may exhibit a higher Agreeableness than Volunteer 1 in terms of the BFF. Otherwise, the effects of age, gender, marriage, or family matters might be imposed on the volunteers' mobility. The research that combines human personality and other personal factors with the topic of this research may result in the personality prediction from the mobility prediction method, as shown in [33]. Additionally, we can use another source data like from the SNS and identify the purpose of movement and action at a location. It is hypothesized that once this information is combined with mobility patterns, a more precise prediction could be accomplished.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. NRF- 2017R1D1A1B03029788).

References

1. Y. Lee and H. Ko, "Efficient STMPM (spatio-temporal moving pattern mining) using moving sequence tree," in *Proceedings of 2008 4th International Conference on Networked Computing and Advanced Information Management*, Gyeongju, South Korea, 2008, pp. 432-437.
2. H. Y. Song, "Probabilistic space-time analysis of human mobility patterns," *WSEAS Transactions on Computers*, vol. 12, pp. 222-238, 2016.
3. M. Gorawski and P. Jureczek, "Continuous pattern mining using the FCPGrowth algorithm in trajectory data warehouses," in *Hybrid Artificial Intelligence Systems*. Heidelberg: Springer, 2010, pp. 187-195.
4. J. W. Lee, O. H. Paek, and K. H. Ryu, "Temporal moving pattern mining for location-based service," *Journal of Systems and Software*, vol. 73, no. 3, pp. 481-490, 2004.
5. A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "Wherenext: a location predictor on trajectory pattern mining," in *Proceedings of the 15th ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, Paris, France, 2009, pp. 637-646.
6. F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, CA, 2007, pp. 330-339.
7. H. Jeung, Q. Liu, H. T. Shen, and X. Zhou, "A hybrid prediction model for moving objects," in *Proceedings of 2008 IEEE 24th International Conference on Data Engineering*, Cancun, Mexico, 2008, pp. 70-79.
8. M. Morzy, "Prediction of moving object location based on frequent trajectories," in *Computer and Information Sciences – ISCIS 2006*. Heidelberg: Springer, 2006, pp. 583-592.
9. M. Morzy, "Mining frequent trajectories of moving objects for location prediction," in *Machine Learning and Data Mining in Pattern Recognition*. Heidelberg: Springer, 2007, pp. 667-680.
10. V. T. H. Nhan and K. H. Ryu, "Future location prediction of moving objects based on movement rules," In *Intelligent Control and Automation*. Heidelberg: Springer, 2006, pp. 875-881.
11. D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel approaches to the indexing of moving object trajectories," in *Proceedings of the 26th VLDB Conference*, Cairo, Egypt, 2000, pp. 395-406.
12. J. J. C. Ying, W. C. Lee, T. C. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Chicago, IL, 2001, pp. 34-43.
13. S. Gambs, M. O. Killijian, and M. N. del Prado Cortez, "Next place prediction using mobility Markov chains," in *Proceedings of the 1st Workshop on Measurement, Privacy, and Mobility*, Bern, Switzerland, 2012, pp. 1-6.
14. W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden Markov models," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, Pittsburgh, PA, 2012, pp. 911-918.
15. M. Baratchi, N. Meratnia, P. J. Havinga, A. K. Skidmore, and B. A. Toxopeus, "A hierarchical hidden semi-Markov model for modeling mobility data," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Seattle, WA, 2014, pp. 401-412.
16. W. He, H. Liu, J. He, S. Tang, and X. Du, "Extracting interest tags for non-famous users in social network," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, Melbourne, Australia, 2015, pp. 861-870.
17. M. Elleuch, H. Kaaniche, and M. Ayadi, "Exploiting neuro-fuzzy system for mobility prediction in wireless ad-hoc networks," in *Advances in Computational Intelligence*. Cham: Springer, 2015, pp. 536-548.
18. M. F. Hassanin and A. Badr, "Mobility prediction using modified RBF network," *International Journal of Computer Applications*, vol. 118, no. 25, pp. 1-4, 2015.
19. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012.
20. H. Y. Song and D. Y. Choi, "Defining measures for location visiting preference," *Procedia Computer Science*, vol. 63, pp. 142-147, 2015.
21. L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3-4, pp. 197-387, 2014.
22. G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527-1554, 2006.
23. D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43-62, 1997.
24. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
25. Convolutional Neural Networks (LeNet), 2010, http://deeplearning.net/tutorial/lenet.html.
26. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
27. J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," 2017, https://arxiv.org/abs/1610.00081.
28. X. Song, R. Shibasaki, N. J. Yuan, X. Xie, T. Li, and R. Adachi, "DeepMob: learning deep knowledge of human emergency behavior and mobility from big and heterogeneous data," *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 4, pp. 1-19, 2017.
29. A. Sudo, T. Kashiyama, T. Yabe, H. Kanasugi, X. Song, T. Higuchi, S. Nakano, M. Saito, and Y. Sekimoto, "Particle filter for real-time human mobility prediction following unprecedented disaster," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Burlingame, CA, 2016, pp. 1-10.
30. Y. Wang, N. J. Yuan, D. Lian, L. Xu, X. Xie, E. Chen, and Y. Rui, "Regularity and conformity: location prediction using heterogeneous mobility data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, Australia, 2015, pp. 1275-1284.
31. X. Song, Q. Zhang, Y. Sekimoto, R. Shibasaki, N. J. Yuan, and X. Xie, "Prediction and simulation of human mobility following natural disasters," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, article no. 29, 2016.
32. M. X. Hoang, Y. Zheng, and A. K. Singh, "FCCF: forecasting citywide crowd flows based on big data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Burlingame, CA, 2016, pp. 1-10.
33. D. Y. Kim and H. Y. Song, "Method of predicting human mobility patterns using deep learning," *Neurocomputing*, vol. 280, pp. 56-64, 2018.

Dabin You https://orcid.org/0000-0001-8323-7916

Dabin You received his B.S. degree in Computer Engineering in 2016 and received his M.S. degree in Computer Engineering in 2018, respectively, both from Hongik University, Seoul, Korea. His research interests are in the areas of deep learning and next location prediction.

Ha Yoon Song https://orcid.org/0000-0002-4934-5072

Ha Yoon Song received his B.S. degree in Computer Science and Statistics in 1991 and received his M.S. degree in Computer Science in 1993, both from Seoul National University, Seoul, Korea. He received Ph.D. degree in Computer Science from University of California at Los Angeles, USA in 2001. From 2001 he has worked at Department of Computer Engineering, Hongik University, Seoul, Korea and is now a full professor. In his sabbatical year 2009, he worked at Institute of Computer Technology, Vienna University of Technology, Austria as a visiting scholar and in the year 2018, he worked at Free University Berlin as a visiting scholar. Prof. Song's research interests are in the areas of data science, mobile computing, application of machine learning, location based service and human mobility modeling.