



# DeepMove: Predicting Human Mobility with Attentional Recurrent Networks

Jie Feng<sup>1</sup>, Yong Li<sup>1</sup>, Chao Zhang<sup>2</sup>, Funing Sun<sup>3</sup>, Fanchao Meng<sup>3</sup>, Ang Guo<sup>3</sup>, Depeng Jin<sup>1</sup>

<sup>1</sup>Department of Electronic Engineering, Tsinghua University, Beijing, China

<sup>2</sup>Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA

<sup>3</sup>Tencent Inc., Beijing, China

liyong07@tsinghua.edu.cn

## ABSTRACT

Human mobility prediction is of great importance for a wide spectrum of location-based applications. However, predicting mobility is not trivial because of three challenges: 1) the complex sequential transition regularities exhibited with time-dependent and high-order nature; 2) the multi-level periodicity of human mobility; and 3) the heterogeneity and sparsity of the collected trajectory data. In this paper, we propose DeepMove, an attentional recurrent network for mobility prediction from lengthy and sparse trajectories. In DeepMove, we first design a multi-modal embedding recurrent neural network to capture the complicated sequential transitions by jointly embedding the multiple factors that govern the human mobility. Then, we propose a historical attention model with two mechanisms to capture the multi-level periodicity in a principle way, which effectively utilizes the periodicity nature to augment the recurrent neural network for mobility prediction. We perform experiments on three representative real-life mobility datasets, and extensive evaluation results demonstrate that our model outperforms the state-of-the-art models by more than 10%. Moreover, compared with the state-of-the-art neural network models, DeepMove provides intuitive explanations into the prediction and sheds light on interpretable mobility prediction.

## CCS CONCEPTS

• Information systems → Location based services; Data mining; • Human-centered computing → Ubiquitous and mobile computing design and evaluation methods;

## KEYWORDS

recurrent neural network; attention; human mobility

### ACM Reference Format:

Jie Feng<sup>1</sup>, Yong Li<sup>1</sup>, Chao Zhang<sup>2</sup>, Funing Sun<sup>3</sup>, Fanchao Meng<sup>3</sup>, Ang Guo<sup>3</sup>, Depeng Jin<sup>1</sup>. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3178876.3186058>

## 1 INTRODUCTION

Human mobility prediction is of great importance in a wide spectrum of applications, ranging from smart transportation and urban

planning, to resource management in mobile communications, personalized recommender systems, and mobile healthcare services. By predicting the future locations people tend to visit, governments can design better transportation planning and scheduling strategies to alleviate traffic jams and handle crowd aggregations. Ride-sharing platforms like Uber and Didi also heavily rely on accurate mobility prediction techniques, for better estimating the travel demands of their customers and scheduling resources to meet such demands accordingly. With the proliferation of such mobile applications, it has become a pressing need to understand the mobility patterns of people from their historical traces and foresee their future whereabouts.

By measuring the entropy of individual's trajectory, Song et al. [28] find remarkable stability in the predictability of human mobility — 93% human movements are predictable according to their study on a million-scale user base. So far, lots of research efforts [5, 10, 20, 22, 28, 43] have been taken to turn this identified predictability into actual mobility prediction models. Early methods for mobility prediction are mostly pattern-based [4, 18, 25, 26, 42]. They first discover pre-defined mobility patterns (e.g., sequential patterns, periodic patterns) from the trajectory traces, and then predict future locations based on these extracted patterns. Such methods, however, not only suffer from the one-sided nature of the pre-defined patterns, but also ignore personal preferences that are critical for mobility prediction. More recent developments turn to model-based methods [16, 22, 43] for mobility prediction. They leverage sequential statistical models (e.g., Markov chain or recurrent neural network) to capture the transition regularities of human movements and learn the model parameters from the given training corpus.

Despite the inspiring results of model-based mobility prediction, there are several key challenges that remain to be solved to realize the high potential predictability of human movements: (1) First, human mobility exhibits *complex sequential transition regularities*. In practice, the transitions between two arbitrary locations can be time-dependent and high-order. For instance, the probability of moving from home to office for a commuter is higher in work-day mornings but often low in weekend mornings. Meanwhile, the transition may not follow the simple and exact Markov chain assumption, as people can go to different places (e.g., breakfast places) in their commute routes, which lead to high-order and irregular transition patterns. (2) Second, there is often *multi-level periodicity* that governs human mobility. Periodicity has been demonstrated as an important factor that governs human movements [6, 41]. However, existing mobility prediction models are mostly sequential models that only capture the transitional regularities. Further, the

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW 2018, April 23–27, 2018, Lyons, France

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

ACM ISBN 978-1-4503-5639-8/18/04.

<https://doi.org/10.1145/3178876.3186058>

mobility periodicity is often complex and multi-level, involving daily routines, weekend leisure, yearly festivals, and even other personal periodic activities. All these periodic activities interweave with each other in complex ways and are difficult to be captured. (3) The third challenge is *heterogeneity and sparsity* in the data recording human mobility. Unlike intentionally collected tracking data like taxi trajectories, most data recording human mobility is low-sampling in nature, and the location information is recorded only when the user accesses the location service. Such sparsity makes it difficult for training a mobility model for each individual. Aggregating the data of all users, on the other hand, may face the problem of mixing personalized mobility patterns together and suffer from low prediction accuracy.

In this paper, we propose DeepMove, an attentional recurrent neural network model for predicting human mobility from lengthy and sparse trajectories. In DeepMove, we utilize a multi-modal recurrent neural network to capture the multiple factors that govern the transition regularities of human movements. Specifically, we design a multi-modal embedding module that converts sparse features (e.g., time of day, region, user) into dense representations, which are then fed into a recurrent neural network to model long-range and complex dependencies in a trajectory sequence. DeepMove is capable of discovering the transitional regularities that are shared by all the users, while flexibly leveraging user embeddings to capture personalized movement preferences. Another key component in DeepMove is a historical attention module, which captures the multi-level periodicity of human mobility in a principled way. The attention component is jointly trained to select historical records that are highly correlated with the current prediction timestamp. It thus flexibly utilizes periodic movement regularities to augment the recurrent neural network and improve prediction accuracy. Better still, the learned attention weights offer an easy-to-interpret way to understand which historical activities are emphasized in the prediction process.

Our contributions can be summarized as follows:

- We propose an attentional recurrent model, DeepMove, to predict human mobility from long-range and sparse trajectories. Our model combines two regularities in a principled way: heterogeneous transition regularity and multi-level periodicity. To the best of our knowledge, DeepMove is the first model that simultaneously combines these two important regularities for accurate mobility prediction.
- We design two attention mechanisms that are tailored to cooperate with the recurrent module. The first is to directly embed historical record into independent latent vectors and use the current status to selectively focus on relevant historical steps; while the second preserves the sequential information among historical records. Both unveil the periodicity of human mobility by matching historical records with the current status, and rationalize the prediction making process.
- We perform extensive experiments on three representative real-life mobility datasets. Our results demonstrate that DeepMove outperforms state-of-the-art mobility prediction models by more than 10%. DeepMove shows outstanding generalization ability and is robust across trajectory datasets that

have different natures. Furthermore, compared with existing RNN models, DeepMove provides intuitive explanations into the prediction and sheds light on interpretable mobility prediction.

The rest of this paper is organized as follows. We first formulate the problem and discuss the motivation of our work in Section 2. Following the motivation, we introduce details of the architecture of DeepMove in Section 3. After that, we apply our model on three real-world mobility datasets and conduct extensive analysis on the performance in Section 4. After systematically reviewing the related works in Section 5, we finally conclude our paper in Section 6.

## 2 PRELIMINARIES

In this section, we first formally formulate the mobility prediction problem, and then briefly introduce the recurrent neural network. Finally, we discuss the motivation and overview our solution.

### 2.1 Problem Formulation

**DEFINITION 1 (TRAJECTORY SEQUENCE).** *Spatiotemporal point  $q$  is a tuple of time stamp  $t$  and location identification  $l$ , i.e.,  $q = (t, l)$ . Given a user identification  $u$ , trajectory sequence  $S$  is a spatiotemporal point sequence, i.e.,  $S^u = q_1 q_2 \dots q_n$ .*

**DEFINITION 2 (TRAJECTORY).** *Given a trajectory sequence  $S^u$  and a time window  $t_w$ , trajectory is a subsequence  $S_{t_w}^u = q_i q_{i+1} \dots q_{i+k}$  of  $S^u$  in the time window  $t_w$ , if  $\forall 1 < j \leq k, t_{q_j}$  belongs to  $t_w$ .*

At the  $m$ -th time window  $t_{w_m}$ , the current trajectory of user  $u$  can be defined as  $S_{t_{w_m}}^u$  and the trajectory history can be denoted as  $S_{t_{w_1}}^u S_{t_{w_2}}^u \dots S_{t_{w_{m-1}}}^u$ , where  $t_w$  can be one specific day, one week or one month in the year.

**PROBLEM 1 (MOBILITY PREDICTION).** *Given the current trajectory  $S_{t_{w_m}}^u = q_1 q_2 \dots q_n$  and the corresponding trajectory history  $S_{t_{w_1}}^u S_{t_{w_2}}^u \dots S_{t_{w_{m-1}}}^u$ , predict the next spatiotemporal point  $q_{n+1}$  in the trajectory.*

In this paper, we quantify the time interval of the spatiotemporal point into a fixed value, i.e., 30 minutes. Thus, the mobility prediction is simplified to predict the next location identification  $l$  in the next time interval.

### 2.2 Recurrent Neural Network

Recurrent Neural Network [11] is a class of neural networks with cycle and internal memory units to capture sequential information. Long short-term memory (LSTM) [14] and gated recurrent unit (GRU) [7] are widely used recurrent units. LSTM consists of one cell state and three controlled *gates* to keep and update the cell state. Based on the input and last cell state, LSTM first updates the cell state with parts to keep and parts to drop. Then, LSTM generates the output from the cell state with learnable weight. GRU is a popular variant of LSTM which replaces the forget gate and the input gate with only one update gate. The updating formulations of GRU are as follows,

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \quad (1)$$

$$r_t = \sigma(W_{rx}x_t + W_{rh}h_{t-1} + b_r), \quad (2)$$

$$c_t = \tanh(W_{cx}x_t + r_t * (W_{ch}h_{t-1} + b_c), \quad (3)$$

$$h_t = (1 - f_t) * c_t + f_t * h_{t-1}, \quad (4)$$

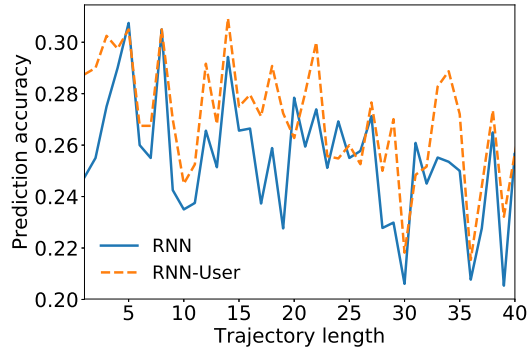


Figure 1: Performance varies with trajectory length.

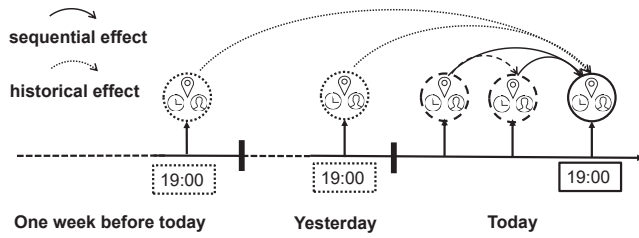


Figure 2: Motivation and Intuition of our solution.

where  $x_t$  is the input in time  $t$ ,  $h_{t-1}$  is the last output of GRU unit, multiple matrix  $W$  are different gate parameters, multiple vectors  $b$  are the bias vectors for different part,  $*$  means element-wise multiplication,  $f_t$  is the update weight,  $r_t$  is the reset gates,  $c_t$  is the candidate and  $h_t$  is the output result. According to Chung et al. [7], GRU achieves the similar performance in multiple tasks with less computation, which is used as the basic recurrent unit in our proposed model.

### 2.3 Overview

As a powerful sequence modeling tool, the recurrent neural network can capture long-range dependencies of sequential information. However, when the sequence is too long, i.e., a long sentence with more than 20 words, its performance will degrade rapidly [1]. According to the typical mobility datasets, the average length of one day's trajectory for mobile application data varies from 20 to 100, which obviously exceeds the processing ability of recurrent neural network. Figure 1 plots the prediction accuracy obtained by a simple recurrent neural network. It shows that the prediction accuracy varies significantly with the testing trajectory. The longer the time extends, the worse performance the prediction achieves. Thus, with the recurrent neural network, we can only process limited length trajectory with a short duration of one day or even shorter.

Directly applying the recurrent neural network to solve the mobility prediction problem is intuitive but inefficient. Except the long-term nature mentioned above, some other challenges also make it failed. The first is the multi-level periodicity of human mobility. Generally, there exists several periodicities in human activities: day, week, month and even other personal periodicities, and thus the multi-level periodicity becomes a universe pattern that

governs the human mobility. However, the general recurrent neural network can do little to handle this because of the long-term effect and the complicated influencing factors from the trajectory. Besides, because users will not report their activities in every location (unlike uniform sampling traces of taxi [31]), most collected mobility data is sparse and incomplete, which cannot record the periodical nature of human mobility. In general, the data quality problem degrades the performance in two ways. The first one is that the missing data will puzzle the recurrent neural network, and induce it to learn the wrong transition information. The other one is that the sparse data makes it difficult to train model for every individual. Even for capturing the transition relations, the recurrent neural network faces the problem of the time-dependent and high-order nature of human mobility. In conclusion, the recurrent neural network faces the problem of periodicity, data sparsity and complicated transitions, which prevent it to achieve high prediction accuracy for human mobility.

Based on the above observations, we propose DeepMove, an attentional recurrent neural network for predicting human mobility from the lengthy, periodical and incomplete trajectories. Figure 2 presents the intuition behind our solution: **not only the sequential information from the current activities decides the next mobility status but also the periodical information from the trajectory history takes effects**. In DeepMove, we first involve multi-modal recurrent neural network to capture the complicated transition relationship. In the multi-modal recurrent neural network, we design a multi-modal embedding module to convert the sparse features (e.g., user, location, time of day) into dense representations, which are more expressing and computable. Then, they are jointly fed into a recurrent neural network to capture the complicated transition relationship. With the help of user embedding, DeepMove can distinguish every user and learn the personal preference while training a single model for all users to learn and share similar mobility regularities. Besides, the time representation involved in the embedding gives recurrent neural network the ability to model the time-dependent nature.

Another key component of DeepMove is the historical attention module, which is designed to capture the multi-level periodical nature of human mobility by jointly selecting the most related historical trajectories under the current mobility status. The historical attention module first extracts spatiotemporal features from the historical trajectories by an extractor. Then, these features are selected by the current mobility status based on the spatiotemporal relations to generate the most related context. By combining this context with the current mobility status, we could predict the mobility based on not only the sequential relation but also the multi-level periodical regularity.

## 3 THE DEEPMOVE MODEL

### 3.1 Multi-modal Recurrent Prediction Framework

Figure 3 presents the architecture of Deepmove. It consists of three major components: 1) feature extracting and embedding; 2) recurrent module and historical attention; and 3) prediction.

**3.1.1 Feature Extracting and Embedding.** The trajectories are partitioned into two parts: current trajectory and historical trajectory. The current trajectory is processed by the recurrent layer

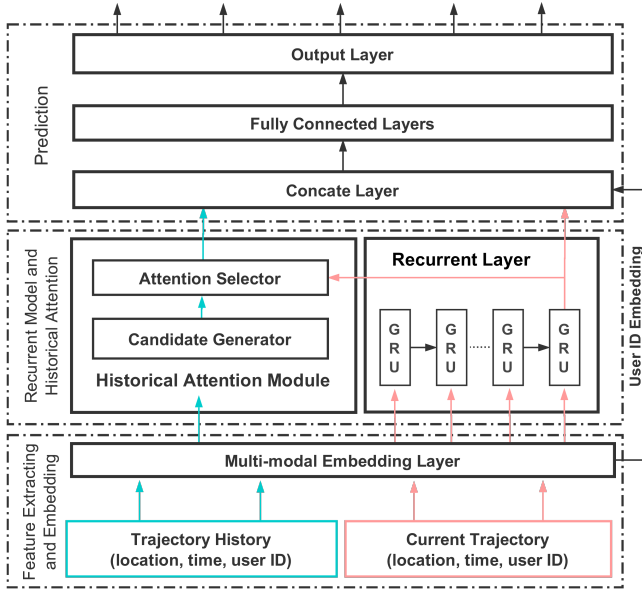


Figure 3: Main architecture of DeepMove.

to model the complicated sequential information. The trajectory history is handled by the historical attention module to extract the regularity of mobility. Before that, all the trajectories are first embedded by the multi-modal embedding module. Simple model like Markov chains can only describe the transitions between independent states like locations. However, mobility transitions are governed by multiple factors like time of day and user preference. Thus, we design a multi-modal embedding module to jointly embed the spatiotemporal features and the personal features into dense representations to help model the complicated transitions. In practice, all the available features of one trajectory point including time, location, user ID can be numbered. Then, the numbered features are translated into one-hot vectors and inputted to the multi-modal embedding module. The ID number of any new user, which is not appeared in the training set, is fixed as 0. According to the word2vec project [24], compared with the limited one-hot representation, the dense representation can better capture the precise semantic spatiotemporal relationship. Another advantage is that this dense representation is always lower dimension, which benefits the follow-up computation.

**3.1.2 Recurrent Module and Historical Attention.** The recurrent module aims to capture the complicated sequential information or long-range dependencies contained in the current trajectory. We select GRU as the basic recurrent unit because of its computation efficiency without performance decay. The recurrent layer takes the spatiotemporal vector sequence embedded by the multi-modal embedding layer as input and outputs the hidden state step by step. These hidden states are called as the current status of the mobility. In every step, the output hidden state flows to historical attention module and prediction module. Paralleled with the recurrent module is the historical attention module, which is designed to capture mobility regularity from the lengthy historical records. It takes the historical trajectory as the input and outputs the most related context vector when queried by a query vector from the recurrent

module. More details about the historical attention module are discussed in the next section.

**3.1.3 Prediction.** The prediction module is the final component that combines the context from different modules to complete the prediction task. It consists of a concatenate layer, several fully connected layers and an output layer. The concatenate layer combines all the features from the historical attention module, recurrent module, and embedding module into a new vector. Following the concatenate layer, fully connected layers further process the feature vector into a smaller and more expressing vector. Finally, the output layer consists of a soft-max layer with negative sampling. Negative sampling [24] can approximately maximize the log probability of the soft-max, which is widely used in natural language processing because of its large vocabulary. In our problem, the size of the location candidate set can also be up to ten thousand, which makes the location representation sparse and the system difficult to train. With the help of negative sampling, our model can converge rapidly. In the practice, the instances of negative sampling are generated by following the uniform distribution.

## 3.2 Historical Attention Module

To capture the multi-level periodicity of human mobility, we need an auto-selector to choose the most related historical records of current mobility status from the trajectory history as the periodicity representation. Inspired by the human visual attention nature and the attention mechanism widely used in natural language translation [1], we design a historical attention module to implement the auto-selector. As Figure 3 presents, it is comprised of two components: 1) an attention candidate generator to generate the candidates, which are exactly the regularities of the mobility; 2) an attention selector to match the candidate vectors with the query vector, i.e., the current mobility status. We first introduce the basic formulation of attention module, and then discuss two specific candidate generation mechanisms.

**3.2.1 Attention Selector.** The goal of attention module is to calculate the *similarity* between the query vector (i.e., the current mobility status) and the candidate vectors to generate the context vector. The attention module is parametrized as a feed-forward neural network that can be trained with the whole neural network. Figure 4(a) presents the framework of this neural network. The attention computation formulations are as follows,

$$c_t = \sum \alpha_i s_i, \quad (5)$$

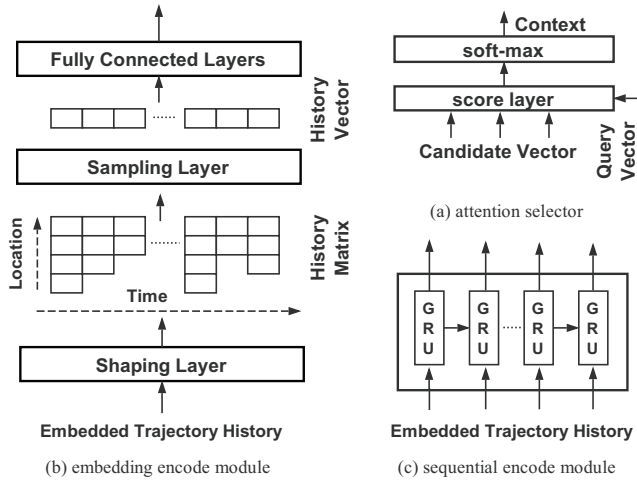
$$\alpha_i = \sigma(f(h_t, s_i)), \quad (6)$$

$$f(h_t, s) = \tanh(h_t W s), \quad (7)$$

where  $s$  represents the historical features,  $W$  is the learnable parameters,  $h_t$  is the query vector which denotes current mobility status from the recurrent layer,  $f$  represents the score function,  $\sigma$  is the soft-max function and  $c_t$  is the context output representing the periodicity related to the current mobility status. While there are many other variations of attention model [21], we choose the original one for its simplicity and general expressions.

**3.2.2 Attention Candidate Generator.** To provide the candidate vectors for the attention selector, we discuss two specific generation mechanisms.





**Figure 4: Architecture of the historical attention module.**

1) *Embedding Encode Module.* The first is the embedding encode mechanism, whose implementation structure is presented in Figure 4(b). The embedding encode module directly embeds the historical records into independent latent vectors as the candidate vectors. It is composed of three components: 1) a shaping layer for disorganizing the ordered trajectory sequence into a history matrix with fixed-length temporal dimension and variable-length spatial dimension; 2) a sampling layer for the location sampling; 3) fully connected layers. The shaping layer is a fixed layer, whose structure and parameters are manual assigned. In this layer, we reorganize the trajectory vectors into a two dimension matrix (for the convenience of discussion, we omit the embedding dimension logically for the time being). In the temporal dimension, we align all the time of trajectory into one week or two days, which is designed to simulate the periodical nature of human mobility. In the spatial dimension, we collect all the locations appeared in the same time period to keep a visited location set for every time slot. Following the shaping layer is a sampling layer which is designed to sample location from the visited location set in every time slot. We design three kinds of sampling strategies: 1) average sampling; 2) maximum sampling; 3) none sampling. Average sampling strategy adds up all the location embedding vectors in the set at every time slot and calculates the mean value as their representation. In this way, all the historical information can be reserved. The maximum sampling strategy is based on the periodical assumption of human mobility: the most frequently visited location means a lot to the user. It works by selecting the most frequent location embedding vector as the representation for every time slot. None sampling strategy is to reserve all the location and flatten them along the temporal dimension without any processing. In the last of the paper, the sampling layer with average sampling strategy is the default settings for the embedding encode mechanism. The final fully connected layers further process the historical spatiotemporal vectors into the appropriate shape.

2) *Sequential Encode Module.* The second mechanism is the sequential encode mechanism, whose implementation structure is presented in Figure 4(c). It consists of a recurrent neural network.

The sequential encode module takes the historical records as input and keeps the intermediate outputs of every step as the candidate vectors. Different from the embedding encode module, it does not directly simulate the periodicity and reserves all the spatiotemporal information. Based on multi-modal embedding method mentioned above, the recurrent neural network can extract complex sequential information from the historical records. Compared with the embedding encode module, sequential encode module relies on the follow-up attention selector to capture the periodical information. Besides, the sequential encode module projects the historical records into a latent space which is similar to the current mobility status in. This similar projection result also benefits the follow-up attentional selection.

### 3.3 Training Algorithm

Algorithm 1 outlines the training process of DeepMove. DeepMove works in an end-to-end manner without requiring hand-crafting features. In general, next location prediction from the limited discrete location list can be regarded as a multi-classification problem, we choose the cross-entropy loss as our loss function. In practice, we use Backward Propagation Through Time (BPTT) and Adam [17] to train it. The historical attention module is parameterized to a feed-forward neural network that can be jointly trained with the whole recurrent neural network.

---

#### Algorithm 1: Training algorithm for DeepMove

---

**Input:** Time window:  $t_w$ ; Trajectory:  $\{S^{u_1}, S^{u_2}, \dots, S^{u_n}\}$ .

**Output:** Trained Model  $M$ .

//construct training instances

$\mathcal{D} \leftarrow \emptyset$

**for**  $u \in \{u_1, \dots, u_n\}$  **do**

**for**  $m \in \{1, 2, \dots, M\}$  **do**

$S_c^u = S_{t_w m}^u; S_h^u = S_{t_w 1}^u S_{t_w 2}^u \dots S_{t_w m-1}^u$   
        put a training instance  $(u, m, S_c^u, S_h^u)$  into  $\mathcal{D}$

initialize the parameters  $\theta$

**for**  $i \in \{1, 2, \dots, EPOCH\}$  **do**

    select instance  $\mathcal{D}_m^u$  with  $m$  in order for user  $u$  from  $\mathcal{D}$

    update  $\theta$  by minimizing the objective with  $\mathcal{D}_m^u$

    stop training when criteria is met

output trained model  $M$

---

## 4 PERFORMANCE EVALUATION

### 4.1 Datasets

We collect three representative real-life mobility datasets to evaluate the performance of our proposed model. The first one is the public Foursquare check-in data, the second one is a mobile application location data from a popular social network vendor, and the last one is call detail records (CDR) data from a major cellular network operator. The generation mechanism of trajectory records of three data is totally different, which represent three main location generation mechanisms in the reality.

- Call detail records data with location records generates in the base station of cellular network when users access it for communication and data accessing.

- Mobile application data with location records generates in the application servers when users request location service in the application like search, check-in and so on.
- In Foursquare, users always intentionally publish their location information to share with other friends and the public, which is the check-in location.

Besides, three datasets are collected among three different cities during different time period. All of these features ensure the representativeness of our data. The basic information of three mobility datasets is presented in Table 1. Figure 5 shows the spatiotemporal features of three mobility data. The details about the datasets and basic preprocessing procedure are discussed as follows.

| Dataset   | Foursquare | Mobile Application | Cellular Network |
|-----------|------------|--------------------|------------------|
| City      | New York   | Beijing            | Shanghai         |
| Duration  | 1 year     | 1 month            | 1 month          |
| Users     | 15639      | 5000               | 1075             |
| Records   | 293559     | 15007511           | 491077           |
| Locations | 43379      | 31522              | 17785            |
| Loc./User | 40         | 48                 | 40               |

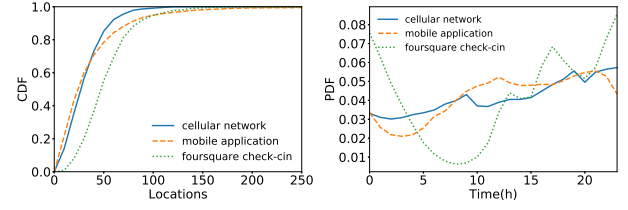
**Table 1: Basic statistics of mobility datasets.**

**Foursquare Check-in Data:** This data is collected from Foursquare API from Feb. 2010 to Jan. 2011 in New York. Every record in the data consists of user ID, timestamp, GPS location and POI ID. The data is sparse and the average length of records for one user is about 18 during one year. Thus, we filter out the users with less than 10 records and then cut the left trajectories into several sessions based on the interval between two neighbor records. Further, we filter out the sessions with less than 5 records and the users with less than 5 sessions. Here, we choose 72 hours as the default interval threshold based on the practice. Besides, we normalize the time stamp into one week with keeping the original order of the trajectory.

**Mobile Application Data:** This data is collected from the most popular social network vendor in China. It records the location of users whenever they request the localization service in the applications. The data is collected from 17 Nov. 2016 to 31 Oct. 2016. The localization of the records is mainly achieved by GPS modules on the mobile phone and enhanced by other possible sensors. For the convenience of representation and computation, the GPS location is projected into street block, which can be represented as a street block ID.

**Call Detail Records Data:** This data is collected from one major cellular network operator in China. It records the spatiotemporal information of users when they access the cellular networks (i.e., making phone calls, sending short messages, or consuming data plan). The data is collected from 1 Jan. 2016 to 31 Jan. 2016. The spatial granularity of it is the cellular base station, which is similar to street block.

Different from the sparse Foursquare check-in data, mobile application data and call detail records data are both dense daily mobility data [34, 35]. In order to obtain meaningful trajectory from them, we first split the whole trajectory into different sessions by the date. Further, we split one day into 48 pieces and aggregate the records in the same time slot into one record. In practice, because of the duplication of the raw mobility data, we filter out these records during the same period of time.



**Figure 5: Spatiotemporal features of three mobility data.**

Other information about three mobility data can refer to Table 1 and Figure 5. To protect the privacy of the users, the base station ID, the street block ID and the user ID are all anonymous. Meanwhile, we want to point out that only the core researchers can access to the data with the strict non-disclosure agreements. Besides, all the data are stored in a secure local server. After processing data without leaking user privacy, we will open and publish these datasets and codes for the community.

## 4.2 Experimental Setup

To evaluate the accuracy of our predictive model, we compared the proposed model with several most updated methods: (1) **Markov model** is widely used to predict human prediction [16, 22] for a long time. They regard all the visited locations as states and build a transition matrix to capture the first order transition probabilities between them. (2) **PMM** [6], which is recently proposed, assumes that mobility location follows a spatiotemporal mixture model and predicts the next locations with periodicity into consideration. (3) **RNN-based model** [11, 20] can be regarded as a simplification version of our model without historical attention module. We adapt the ST-RNN [20] to our scene where only anonymous location ID is known, while the original version focuses on modeling the continuous spatiotemporal information. In the experiments, RNN-Short means that the trajectory is fed into the model day by day, while RNN-Long means that the whole trajectory lasting one month is directly fed into the model. The recurrent module of RNN-based model is GRU, which is the same with our model.

| Training Settings      | Value | Feature (Size) | Input           | Output |
|------------------------|-------|----------------|-----------------|--------|
| learning rate ( $lr$ ) | 1e-3  | location       | $\approx 10000$ | 256    |
| the decay of $lr$      | 0.1   | time           | {48,168}        | 16     |
| L2 penalty             | 1e-5  | user ID        | $\approx 1000$  | 16     |
| gradient clip          | 1.0   | hidden state   | 300             | 256    |

**Table 2: The default parameter settings for DeepMove.**

The default training settings for our proposed model are presented in Table 2, while the same parameters are used to train and evaluate the baseline RNN model. The experiments are conducted in terms of test-train mode, where the first 80% of each users' trajectory are selected as training data, the remaining 20% as testing data. In the following experiments, we utilize GRU as the default recurrent module of our model. In practice, the performance of GRU and LSTM are close, which are much better than the performance of vanilla RNN [7]. We repeat our experiments at least 3 times with different random seed.

### 4.3 Overall Performance

We evaluate our model with the baseline methods on three mobility datasets to present the performance of our model. We rank the candidate locations by the probabilities generated from the model, and check whether the ground-truth location  $v$  appears in the top- $k$  candidate locations. The results of top-1 and top-5 prediction accuracy are presented in Figure 6.

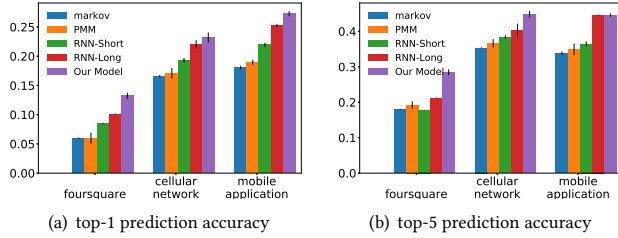


Figure 6: Performance comparison with baselines.

We first analyze the result of Foursquare check-in data in Figure 6. In baseline methods, general RNN model works better than others significantly because of its powerful sequence modeling ability. Compared with the performance of the general RNN, we find that the prediction accuracy of our model is about 31.63% better on average. This suggests that there indeed exist periodical regularity in the human mobility, which helps to improve the prediction accuracy. As the general RNN captures the complex sequential transition from the current trajectory, recurrent part of our model can also do like this. Nevertheless, our model utilizes the historical attention module to capture the periodical regularity from the lengthy trajectory history. Such attention mechanism on the trajectory helps our model understand the human mobility and achieve much better prediction accuracy.

Evaluation results of the other two mobility datasets also demonstrate the superiority and generalization of our model. Compared with the Foursquare check-in data, cellular neural network data and mobile application data completely record human's daily life. As Figure 6 presents, the performance of our historical attention model outperforms the general RNN model over 8.04% on average in mobile application data. The performance gain is 5.16% on average in cellular network data, which demonstrates the generalization of our model. Compared with the general RNN, the advantage of our model is that it can capture the periodical regularity of human mobility from trajectory history. In general, our model significantly outperforms all the baseline methods on three different mobility data in terms of prediction accuracy.

Besides, we compare our models with the baseline methods on the raw schema, i.e., without deleting the duplication, of two daily mobility data. In Table 3, we can observe that our model outperforms the general RNN model by only about 1%, which achieves the prediction accuracy of 69.4%. Meanwhile, even the prediction accuracy of Markov model comes up to 50% in two mobility data. According to a further analysis of the data, we find that many users always stay in a location for several hours during the day. For this kind of trajectory, we can achieve pretty well results in location

prediction only by simply copying the current input, where complex methods like attention mechanism takes minor effect because of the principal influence of the current input. Apparently, the performance gain of our DeepMove will be limited. However, it will work well on the mobility data where users move around different places.

|           | cellular network | mobile application |
|-----------|------------------|--------------------|
| Markov    | 0.459            | 0.595              |
| RNN       | 0.595            | 0.690              |
| Our model | 0.593            | 0.694              |

Table 3: Prediction performance on dense mobility data.

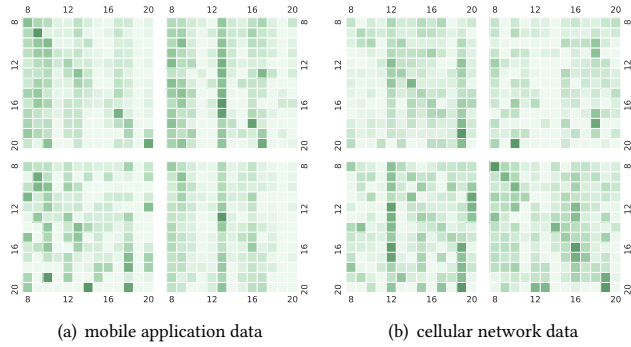
### 4.4 Reason Interpretation: Visualization of Historical Attention Weights

Because of the importance of periodicity of human mobility, our model, especially the historical attention module, is designed to capture the periodicity of human mobility. Thus, in this section, we discuss whether our periodicity assumption appears and whether our model really captures it.

In Figure 7, we visualize the output of historical attention module to demonstrate this. To obtain the visualization, we first collect the normalized weight of historical attention module for a little seed users, and align them together on the time dimension. Then, we re-normalize the weights and draw them in Figure 7 in terms of the heatmap. The horizontal axis and vertical axis of every square matrix in Figure 7 are both time period, the shade of the grids describe the weight, where the deeper green means the larger weight. For example, the top-left square matrix in Figure 7(a) shows us the distribution of the historical attention weight from 8.am. to 8. pm. during the weekday via the weekday's historical trajectories in mobile application data. The diagonal entries of it are remarkably larger than other entries, which shows the day-level regularity of human mobility in the different workday. The top-left square matrix in Figure 7(b) shows the similar result, while it is based on another cellular network data. The bottom-right square matrix in Figure 7(b) shows the attention distribution in the weekend in cellular network data, which also reveals a remarkably day-level regularity. In general, the results of Figure 7 show that our model indeed captures the regularity and periodicity from the historical trajectory. Meanwhile, our historical attention module not only improves the prediction accuracy but also offers an easy-to-interpret way to understand which historical activities are emphasized in the future mobility.

### 4.5 Model Variations

In order to present the efficiency of historical attention module, we first compare two proposed historical attention modules in terms of prediction accuracy and computation efficiency, and then discuss how different sampling strategies in the embedding encode attention module can influence the final results. Finally, we discuss the effect of user embedding and present the effectiveness of our model in describing personal preference.



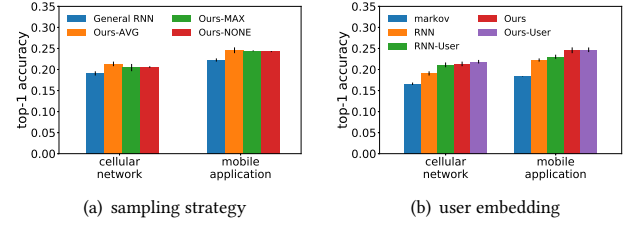
**Figure 7: Visualization of the output of historical attention module.** Every matrix presents the correlations between current trajectory and historical trajectory. The diagonal entries of matrix present the correlations of trajectories in the same time period of different days. The shade of the grids describe the weight where the deeper green means the larger weight. For example, the top-left matrix in (a) shows the correlations of the current trajectory and the historical trajectory on workdays on cellular network data. The highlight diagonal entries tell us that trajectories during the workday are periodical. The bottom-right matrix in (b) tells us that trajectories during the weekend are even more periodical than the workday.

We compare the performance and efficiency of our proposed two historical attention module on two datasets. The results are presented in Table 4. The sequential encode attention module works better than the embedding encode attention module in most of the time especially in mobile application data, while the latter one computes more efficiently. Two reasons may account for the better performance of the sequential encode attention module: 1) it captures sequential information along the lengthy trajectory to some extent, while the embedding encoder cannot; 2) the latent space of output of it is more similar to the current mobility status’s because of the similar generation structure.

Besides, we evaluate the system performance of different sampling strategies in the sampling layer of embedding encode attention module. As mentioned in the model section, we design three kinds of sampling strategies in the historical attention module: average sampling, maximum sampling, and none sampling. Figure 8(a) shows the evaluation results of three different samplings in two datasets in terms of top-1 prediction accuracy. In general, the average sampling strategy works better among three strategies, while the maximum sampling strategy performs a little worse, the result shows us that most people own regular mobility pattern and limited locations to visit. The performance gap of three strategies on cellular network data is larger than the mobile application data.

Finally, we identify every single user with a user ID and add user ID embedding feature to the model to capture the personality. The results are presented in Figure 8(b). Obvious performance gain can be observed in the general RNN model after adding the user ID embedding. However, the performance gain of our model can be omitted, which demonstrates that our proposed model not only

capture deeper periodical pattern but also characterize personal regularities.



**Figure 8: Performance variation with the model design.**

| Dataset            | Model       | Top-1 Accuracy | Overhead(s) |
|--------------------|-------------|----------------|-------------|
| cellular network   | attention 1 | 0.22           | ≈600        |
|                    | attention 2 | 0.24           | ≈1600       |
| mobile application | attention 1 | 0.24           | ≈2600       |
|                    | attention 2 | 0.27           | ≈11200      |

**Table 4: Efficiency of two attention models.**

#### 4.6 Evaluation on User Groups

In order to evaluate the variation of performance gain among different users, we cluster them into different groups based on three rules: mobility entropy [8], explore ratio, radius of gyration [10]. Mobility entropy calculates the entropy of locations in trajectory, which is related to the regularity level of human mobility. Explore ratio represents the fraction of new locations in test data, which do not exist during the training. Thus, one person with more regular behaviors should have lower mobility entropy and lower explore ratio. The final rule is the radius of gyration which describes the spatial range of the mobility.

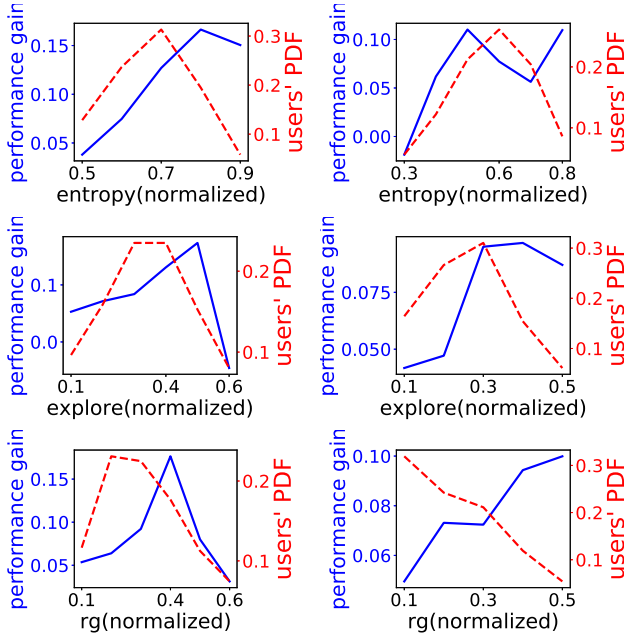
The evaluation results are presented in Figure 9, where the vertical axis shows the performance gain compared with the baseline method-general RNN. There are two interesting insights from the result: 1) our model outperforms the baseline method on almost all kinds of users; 2) our model predicts non-regular users better than baseline method that meets the goal of our historical attention module. For example, in the top-left image of Figure 9, our model’s prediction accuracy increases when the mobility entropy of users increases. With the effective usage of lengthy historical trajectory, our model captures the underlying and deeper periodical patterns of human mobility.

### 5 RELATED WORK

Works close to our work can be classified into two categories: model-based methods and pattern-based methods. Besides, we introduce related works on recurrent neural network and attention model.

*Model-based methods:* Markov model and its variations are common models of this approach. In Markov-based models [3, 16], they model the probability of the future action by building a transition matrix between several locations based on the past trajectories.





**Figure 9: Performance varies with mobility entropy/explore ratio/radius of gyration (rg) on cellular network data (left) and mobile application data (right).**

To capture the unobserved characteristics between location transition, Mathew et al. [22] cluster the locations from the trajectories and then train a Hidden Markov Model for each user. Considering the mobility similarity between user group, Zhang et al. [43] propose GMove: a group-level mobility modeling method to share significant movement regularity. Different from existing Markov-based models, our model can model time-dependent and high order transitions.

*Pattern-based methods:* Pattern-based methods [25, 26, 34, 40, 42] first discover the popular sequential mobility patterns from the trajectory, and then try to predict the mobility based on these popular patterns. Matrix factorization can also be regarded as a kind of pattern discovered method. Matrix factorization (MF) [18, 29] emerges from recommendation system and the basic idea of it is to factorize the users-items matrix into two latent matrices which represent the users and items characteristics. Cheng et al. [4] fuse MF with the geographical influence by modeling the location probability as a multi-center Gaussian Model. Combining Markov model with matrix factorization, Rendle et al. [27] propose factorized personalized Markov model (FPMC) to do item recommendation. Based on FPMC, Cheng et al. [5] propose a matrix factorization method named FPMC-LR to capture the sequence transition with Markov chain while considering the localized region constraint. Compared with pattern-based methods, our model can not only model the transitional regularities shared by all the users but also model the personal preference based on the user embedding and personal historical trajectory.

*Recurrent neural network:* Recurrent Neural Network (RNN) [12, 19] is a powerful tool to capture the long-range dependencies in the sequence and have achieved success in Natural Language Processing (NLP) [1, 30], Image Caption [36], etc. Because of its powerful

representation ability, RNN have been applied to many fields like click prediction [44], recommendation system [13] and mobility prediction [9, 15, 20, 23, 37, 38]. RNN-based models is also a kind of model-based method. Liu et al. [20] propose Spatial Temporal Recurrent Neural Networks (ST-RNN) to model temporal and spatial contexts. However, the proposed model is too complicated to train and apply with so many parameters. Besides, it can not be applied into the discrete location prediction scene because of its continuous spatial modeling method. Du et al. [9] propose Recurrent Marked Temporal Point Process (RMTTP) to learn the conditional intensity function automatically from the history. However, this model is not specific for the trajectory prediction and do not consider the natural characteristics of trajectory like multi-level periodicity. By coupling convolutional and recurrent neural network, the Yao et al. [39] propose DeepSense: a unified deep learning framework for mobile sensing data. However, this model needs uniform sampling data and also does not consider the multi-level periodicity of trajectory.

*Attention model:* Based on the seq2seq model [30], Bahdanau et al. [1] introduce attention mechanism into neural machine translation task. The attention mechanism strengthens not only the ability of RNN in capturing the long-range dependencies but also the interpretability. Attention mechanism has been applied into many other fields such as image caption [36] and recommendation system [2], which achieves satisfactory result. To the best of our knowledge, we are the first to propose the attention mechanism into mobility prediction to model human mobility.

## 6 CONCLUSION

In this paper, we investigated the problem of mobility prediction from the sparse and lengthy trajectories. We proposed an attentional mobility model DeepMove, which enjoys two novel characteristics compared to previous methods: 1) a multi-modal embedding recurrent neural network for capturing multiple factors that govern the transition regularities of human mobility; and 2) a historical attention module for modeling the multi-level periodicity of human mobility. Extensive experiments on three real-life mobility datasets show that DeepMove significantly outperforms all the baselines. Meanwhile, the visualization of historical attention weights shows that DeepMove is able to effectively capture meaningful periodicities for mobility prediction.

There are several future directions for our work. First, we currently only predict the next location because we fixed the time interval in practice. In the future, we plan to expand the location prediction into the spatiotemporal point prediction by taking the potential duration into consideration. Second, our current work does not consider the semantic context in the trajectory like point of interests [32, 33] and user's tweets because of the limitation of data. In the future, we plan to add these semantic information into the model to predict not only the location but also the underlying motivation of user's movement.

## ACKNOWLEDGMENTS

This work is supported by research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

## REFERENCES

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *Computer Science* (2014).
- [2] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 335–344. <https://doi.org/10.1145/3077136.3080797>
- [3] Meng Chen, Yang Liu, and Xiaohui Yu. 2014. NLPMM: A Next Location Predictor with Markov Modeling. 8444 (2014), 186–197.
- [4] C. Cheng, H. Yang, I. King, and M. R. Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI Conference on Artificial Intelligence*.
- [5] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: successive point-of-interest recommendation. In *International Joint Conference on Artificial Intelligence*.
- [6] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 1082–1090. <https://doi.org/10.1145/2020408.2020579>
- [7] Junyoung Chung, Çağlar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555 (2014). <http://arxiv.org/abs/1412.3555>
- [8] Justin Cranshaw, Eran Toch, Jason Hong, Aniket Kittur, and Norman Sadeh. 2010. Bridging the Gap Between Physical Location and Online Social Networks. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp '10)*. ACM, New York, NY, USA, 119–128. <https://doi.org/10.1145/1864349.1864380>
- [9] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent Marked Temporal Point Processes: Embedding Event History to Vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 1555–1564. <https://doi.org/10.1145/2939672.2939875>
- [10] M. C. González, C. A. Hidalgo, and A. L. Barabási. 2008. Understanding individual human mobility patterns. *Nature* 453, 7196 (2008), 779.
- [11] Alex Graves. 2008. Supervised Sequence Labelling with Recurrent Neural Networks. *Studies in Computational Intelligence* 385 (2008).
- [12] Alex Graves. 2013. Generating Sequences With Recurrent Neural Networks. *Computer Science* (2013).
- [13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. *Computer Science* (2015).
- [14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [15] Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, Huaxiu Yao, Fei Wu, and Zhenhui Li. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. In *Proceedings of 2018 AAAI Conference on Artificial Intelligence (AAAI '18)*.
- [16] Marc Olivier Killijian. 2012. Next place prediction using mobility Markov chains. In *The Workshop on Measurement, Privacy, and Mobility*. 3.
- [17] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). <http://arxiv.org/abs/1412.6980>
- [18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.
- [19] Zachary C Lipton, John Berkowitz, and Charles Elkan. 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. *Computer Science* (2015).
- [20] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: a recurrent model with spatial and temporal contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*. 194–200.
- [21] Minh Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *Computer Science* (2015).
- [22] Wesley Mathew, Ruben Raposo, and Bruno Martins. 2012. Predicting future locations with hidden Markov models. In *ACM Conference on Ubiquitous Computing*. 911–918.
- [23] Hongyuan Mei and Jason Eisner. 2017. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. In *Advances in Neural Information Processing Systems*. Long Beach. <https://arxiv.org/abs/1612.09328>
- [24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems* 26 (2013), 3111–3119.
- [25] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. 2009. WhereNext: a location predictor on trajectory pattern mining. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July. 637–646*.
- [26] Fabio Pinelli, Fabio Pinelli, Fabio Pinelli, and Dino Pedreschi. 2007. Trajectory pattern mining. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 330–339.
- [27] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *International Conference on World Wide Web*. 811–820.
- [28] C. Song, Z. Qu, N. Blumm, and A. L. Barabási. 2010. Limits of predictability in human mobility. *Science* 327, 5968 (2010), 1018.
- [29] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. *A survey of collaborative filtering techniques*. Hindawi Publishing Corp. 4 pages.
- [30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. 4 (2014), 3104–3112.
- [31] Hongjian Wang and Zhenhui Li. 2017. Region Representation Learning via Mobility Flow. In *CIKM*.
- [32] Fei Wu and Zhenhui Li. 2016. Where Did You Go: Personalized Annotation of Mobility Records. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM '16)*. ACM, New York, NY, USA, 589–598. <https://doi.org/10.1145/2983323.2983845>
- [33] Fei Wu, Zhenhui Li, Wang-Chien Lee, Hongjian Wang, and Zhuojie Huang. 2015. Semantic Annotation of Mobility Data Using Social Media. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1253–1263. <https://doi.org/10.1145/2736277.2741675>
- [34] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. 2017. Trajectory Recovery From Ash: User Privacy Is NOT Preserved in Aggregated Mobility Data. In *International Conference on World Wide Web*. 1241–1250.
- [35] Fengli Xu, Pengyu Zhang, and Yong Li. 2016. Context-aware Real-time Population Estimation for Metropolis. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 1064–1075. <https://doi.org/10.1145/2971648.2971673>
- [36] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *Computer Science* (2015), 2048–2057.
- [37] Cheng Yang, Maosong Sun, Wayne Xin Zhao, Zhiyuan Liu, and Edward Y. Chang. 2017. A Neural Network Approach to Jointly Modeling Social Networks and Mobile Trajectories. *ACM Trans. Inf. Syst.* 35, 4, Article 36 (Aug. 2017), 28 pages. <https://doi.org/10.1145/3041658>
- [38] Di Yao, Chao Zhang, Jianhui Huang, and Jingping bi. 2017. SERM: A Recurrent Model for Next Location Prediction in Semantic Trajectories. In *ACM International Conference on Information and Knowledge Management*. 2411–2414.
- [39] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 351–360. <https://doi.org/10.1145/3038912.3052577>
- [40] Jia Ching Ying, Wang Chien Lee, Tz Chiao Weng, and Vincent S. Tseng. 2011. Semantic trajectory mining for location prediction. In *ACM Sigspatial International Conference on Advances in Geographic Information Systems*. 34–43.
- [41] Quan Yuan, Wei Zhang, Chao Zhang, Xinhe Geng, Gao Cong, and Jiawei Han. 2017. PRED: Periodic Region Detection for Mobility Modeling of Social Media Users. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, New York, NY, USA, 263–272. <https://doi.org/10.1145/3018661.3018680>
- [42] Chao Zhang, Jiawei Han, Lidan Shou, Jiajun Lu, and Thomas La Porta. 2014. Splitter: Mining Fine-grained Sequential Patterns in Semantic Trajectories. *Proc. VLDB Endow.* 7, 9 (May 2014), 769–780. <https://doi.org/10.14778/2732939.2732949>
- [43] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han. 2016. GMove: Group-Level Mobility Modeling Using Geo-Tagged Social Media. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1305–1314.
- [44] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie Yan Liu. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*. 1369–1375.