

Received July 29, 2019, accepted October 10, 2019, date of publication October 31, 2019, date of current version November 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2950370

Passenger Behavior Prediction With Semantic and Multi-Pattern LSTM Model

HAIQUAN WANG^{1,2}, XIN WU¹, LEILEI SUN³, AND BOWEN DU³¹School of Software, Beihang University, Beijing 100083, China²Yunnan Innovation Institute, Beihang University, Kunming 650000, China³School of Computer Science and Engineering, Beihang University, Beijing 100083, China

Corresponding author: Leilei Sun (leileisun@buaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51822802, Grant 51778033, Grant 71901011, and Grant U1811463, in part by the National Key Research and Development Program under Grant 2018YFB2100100, in part by the Beijing Municipal Science and Technology Project under Grant Z171100005117001, and in part by the China Geological Survey Project under Grant DD20190637.

ABSTRACT Understanding passenger behaviors is of great importance in intelligent transportation and infrastructure planning. However, the passenger trajectories are actually complex temporal data, which consist of rich spatial and temporal information. What's more, the observed passenger trajectories may be a mixture of different types of passengers with various travelling purposes. These difficulties make the prediction of passenger behaviors a challenging work. To address these problems, this paper improves the existing passenger behavior prediction methods from the following two aspects: 1) Encoding the travelling sequence with personalized semantic sensing, and 2) constructing multi-pattern prediction models to capture multiple travelling purposes and dynamics. Along this line, this paper provides a novel passenger behavior prediction model, namely, Semantic and multi-Pattern Long Short-Term Memory (SP-LSTM) model. Particularly, 1) a translation unit is designed, which is able to encode an observed travelling sequence into a structured sequence with consideration of individual travelling semantics; 2) a multi-pattern learning schematic is proposed, which first identifies the travelling patterns of passengers and then handles different patterns with different learning modules; 3) a unified learning framework is provided to integrate the semantic sensing module and multi-pattern learning module together, and present the final prediction results. To evaluate the proposed method, this paper conducts experiments on real-world passenger travelling data. Results demonstrate the superiority of SP-LSTM over both classical and the state-of-the-art methods.

INDEX TERMS Behavioral sciences, big data applications, predictive models, public transportation.

I. INTRODUCTION

Public transportation plays an important role in daily life of residents, especially in metropolises such as New York, USA and Beijing, China. On the one hand, data mining and machine learning have undergone a rapid development in the recent years, machine-learning technology powers many aspects of modern society [1]; on the other hand, we have accumulated a large amount of transportation data, such as NYC Taxi Open Data [2], Uber Trip Data, Taxi Trip Records and so on. Hence, an opportunity of improving the public transportation service by data-driven solutions and strategies has been witnessed by many scholars such as [12]–[16].

This paper focuses on modelling passenger behaviors and predicting passengers' next stations. By doing this, we can


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



FIGURE 1. An illustration of next-station prediction.

help the government and enterprises optimize to dispatch the transportation resources, and also help the citizens to circumvent the crowded stations. As shown in Fig. 1, If travelling behaviors of passengers can be predicted precisely, a better traffic route planning can be made for workers with regular moving patterns, such as P_1 and P_3 , and recommended routes or tourism products can be provided for travelers shown as P_2 . It is also helpful for making better city

planning to facilitate people who come to cities for business such as P_4 .

Essentially, the prediction of passengers' next stations could be formulated as a sequence prediction problem, which has been studied in both academia and industry for many years. Many methods have been proposed to model the relationships of historical sequences with future events. For instance, Hidden Markov Model (HMM) [3] first recognized the current hidden state which was not easy to be observed directly in sequences, then presented the prediction according to the current state. Another approaches based-on Markov chain also have been widely applied for sequence prediction. S. Rendle *et al.* proposed Factorizing Personalized Markov Chain (FPMC) which combined Markov chains with matrix factorization to achieve the target of next-basketball prediction [4]. C. Chen *et al.* proposed an embed personalized Markov chain method based on FPMC in order to solve the next POI recommendation [18]. Markov Renewal Process (MRP) [5] also aimed at modeling sequential processes with time intervals using Markov-based method.

With the development of deep learning in recent years, Recurrent Neural Network (RNN) [6] was proposed for modeling sequences by recursive unit, and its variants have also become popular state-of-the-art methods on this field. For example, Y. Zhang *et al.* showed how to use recurrent neural networks to handle sequential click prediction for sponsored search [7]. B. Hidasi *et al.* proposed a RNN solution to solve the session-bases recommendation task [8]. Long Short-Term Memory (LSTM) [9], an important variant of RNN architectures which can handle the gradient vanishing problem, has been applied to many sequence prediction problems. Phased LSTM added the time information to prediction by spatial-designed time gates [10]. Gated Recurrent Unit (GRU) [11] simplifies the LSTM to achieve higher efficiency.

Even so, the passenger behavior prediction is still a challenging problem due to the next two facts: firstly, a same station may have different semantics for different passengers. Therefore passenger behavior prediction without taking station semantics into consideration often fails to get the correct prediction results. Secondly, the collected passenger trajectories consist of multi-various travelling patterns and one model can not uncover all these patterns, if we don't divide data into different parts and train multiple models according to their own patterns, the results of prediction will be definitely incorrect. Due to the above difficulties, most of the existing sequence prediction methods can not provide precise next-station prediction.

To fill these research gaps, we propose a novel passenger behavior prediction method named Semantic and multi-Pattern Long Short-Term Memory (SP-LSTM) model. This model contains two modules called Semantic Sensing module (SS-LSTM) and multi-Pattern Learning module (PL-LSTM) respectively. Correspondingly, two particularly designed structures are added into the classical LSTM: one is to encode the observed travelling sequence into a structured sequence with consideration of individual travelling

semantics then decode it back; another is to ensemble multiple LSTMs with different patterns to predict behaviors of passengers with various travelling purposes.

The main contributions of the paper can be summarized as follows:

- 1) A novel semantic sensing module is proposed for LSTM-based sequence prediction, which improves the prediction performance by uncovering semantics hidden in the observed sequences.
- 2) A multi-pattern learning schematic is provided, which is able to identify the travelling purposes and then handle multiple travelling patterns by different learning units.
- 3) A unifying learning framework is designed to incorporate the above two modules with the basic LSTM model.

The rest of this paper is organized as follows. In Section II, the related work about sequence prediction and recently invented models are reviewed briefly. The details of our methods are presented in Section III. Experiments results on three real-world data sets are shown in Section IV. Finally, we summarize our work in Section V.

II. RELATED WORK

This section provides a three-fold introduction of the related researches, which includes travelling behavior analysis, sequence prediction problem, and recently invented models for sequence prediction.

Travelling behavior analysis has attracted a lot of attentions in urban data analysis. Early study [12] proposed multiple dimensions to analyze people's travel demands based on their traveling behavior. In recent years, travelling behavior analysis is frequently used for Point-Of-Interest (POI) recommendation and urban function perception. The work in [13] utilized taxi drop-offs to profile temporal popularity patterns of POIs for improving the performance of POI recommendations. Y. Zhu *et al.* modelled user behaviors with consideration of time intervals to predict people's next behavior [14]. P. Zhao *et al.* gave the list of users' next locations with considering spatio-temporal factors [15]. Human mobility patterns were extracted from taxi trajectories to help understand zones' function [16]. And the work in [17] proposed a time-aware metric embedding approach with asymmetric projection for successive POI recommendations. FPMC-LR, a tensor-based model, got Markov chain of transitions with distance together [18]. A personalized ranking metric embedding method (PRME) was proposed to model personalized check-in sequences for next POI recommendation [19]. Z. Zhang *et al.* tried to learn user's next movement intention and incorporated different contextual factors to improve next POI recommendation [20].

Sequence prediction has become a popular research spot in recent years and been applied to multiple situations. A representative application is prediction of next basket. S. Rendle *et al.* proposed a method based on personalized transition graphs over underlying Markov chains to

recommend items to users that they might want to buy next time [4]. F. Yu *et al.* proposed a model based on RNN for next basket recommendation [21]. And R. Guidott *et al.* defined a method which was able to understand the level of the customer's stocks and recommended the set of most necessary items [22]. Sequence prediction is also applied to health care. C. Estebann *et al.* modelled clinical processes based on clinical data to develop a clinical decision support system [23]. The work in [24] achieved predictive clinical decision support system based on RNN encoding and tensor decoding. I. M. Baytas *et al.* proposed a patient subtyping model using an improved version of LSTM [25]. L. Li *et al.* introduced a novel application to next career move prediction with a contextual LSTM model [26]. C. Yang *et al.* jointly modeled a social network structure and users' trajectory behaviors with a neural network model named JNTM [27].

Sequence prediction is a very important, but still challenging problem. To satisfy the demands of multi-behavior prediction, Q. Liu *et al.* proposed a method based on LBL [29] which could model multiple types of behaviors in historical sequences with behavior-specific transition matrices [28]. To provide users POI recommendations for a specific period, Y. Liu *et al.* proposed a time-aware model to integrate the users' interests and their evolving sequential preferences with temporal interval assessment [30]. To take spatial and temporal contexts into consideration at the same time, Q. Liu *et al.* extended RNN and proposed a method which could model local temporal and spatial contexts in each layer [31]. Considering that users' long-term preferences might keep evolving over time, H. Ying *et al.* proposed a two-layer hierarchical network with attention module [32]. However, to the best of our knowledge, all of these methods have not solved the following two problems: semantic understanding of sequences and multi-pattern learning of sequences, which absolutely have significant impact on the prediction results and should be taken into consideration definitely. This paper aims to fill these two research gaps, and provides a new passenger behavior prediction model.

III. PROPOSED METHODOLOGY

This section is organized as follows. We will introduce problem statement at first. Then we will review basic LSTM and explain its usage in this problem. Next we presents the passenger behavior prediction method proposed in this paper, namely Semantic and multi-Pattern Long Short-Term Memory (SP-LSTM) model. It consists of two key learning modules, semantic sensing module and multi-pattern learning module. At last we will give theoretical analysis about relationships between our modules and basic LSTM.

A. PROBLEM STATEMENT

The travelling behavior prediction problem studied in this work can be formulated as follows:

Let $\mathbb{P} = \{p_1, p_2, \dots\}$ be a set of passengers and $\mathbb{S} = \{s_1, s_2, \dots\}$ be a set of stations. For each passenger p ,

his travelling behavior sequence T^p is given by $T^p := [(s_1^p, t_1^p), (s_2^p, t_2^p), \dots, (s_{n_p}^p, t_{n_p}^p)]$, where (s_m^p, t_m^p) means that passenger p gets on or off at his m -th station s_m^p at time t_m^p and $(s_{n_p}^p, t_{n_p}^p)$ means passenger p 's last trip. Our target is to predict the next station $s_{n_p+1}^p \in S$ of a certain passenger p .

B. PREDICTION WITH BASIC LSTM

In this subsection, we plan to make a quick review about Long Short-Term Memory network, which is the most popular method adopted to sequence prediction problem in recent years. Then we will introduce two ways in which it can be used in this problem.

LSTM [9] was proposed with special designed memory cells and gate unites to solve the problem of insufficient decaying error back flow in 1997. We combine the commonly-used equations [33] with our settings. The updated equations of basic LSTM can be put as follows:

$$\begin{aligned} i_m &= \sigma_i(W_{xi}T_m + W_{hi}h_{m-1} + b_i), \\ f_m &= \sigma_f(W_{xf}T_m + W_{hf}h_{m-1} + b_f), \\ \tilde{c}_m &= \sigma_c(W_{xc}T_m + W_{hc}h_{m-1} + b_c), \\ c_m &= f_m \odot c_{m-1} + i_m \odot \tilde{c}_m, \\ o_m &= \sigma_o(W_{xo}T_m + W_{ho}h_{m-1} + b_o), \\ h_m &= o_m \odot \sigma_h(c_m), \end{aligned} \quad (1)$$

where i_m, f_m, o_m represent the *input*, *forget* and *output* gates of the m -th object respectively. T_m represents the passengers' m -th travel vector and h_m is the hidden output vector. And \odot represents for the element-wise (Hadamard) product. The update of cell state c_m has two parts. The former part is the previous cell state c_{m-1} that is controlled by forget gate f_m , and the latter part is the new candidate value scaled by how much we decided to add state value. $W_{hi}, W_{hf}, W_{hc}, W_{ho}, W_{xi}, W_{xf}, W_{xc}$, and W_{xo} are weight parameters connecting different inputs and gates. b_i, b_f, b_c , and b_o are corresponding biases.

And the loss function is often defined as:

$$\mathcal{L} = \sum_{p=1}^N \left\| h_{n_p}^p - y_{n_p}^p \right\|, \quad (2)$$

where N is the count of sequences and $y_{n_p}^p$ is the last station $s_{n_p}^p$ of passenger p which is also called ground truth.

For our task, we can use basic LSTM in the following two ways. The first one is to extract station sequence of passenger p from T^p and simply regard the m -th station label as T_m , then put it into (1) after one-hot encoding process. We mark this way as LSTM_1. The second way is to add an embedding layers to the original sequence so as to transform T^p to vectors as the input data. And this way is called LSTM_2. We will use the first way as baseline in section IV.

C. SEMANTIC SENSING AND SS-LSTM

The same stations may have different semantics for different passengers, the superficial labels are very likely to mislead

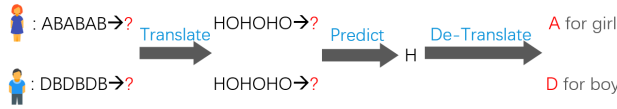


FIGURE 2. Sketch map of semantic sensing. Through this process, station **A** is marked as “**H**” (Home) and station **B** as “**O**” (Office) for girl, station **D** is marked as “**H**” and station **B** as “**O**” for boy according to their sequences. After prediction, the translation label **H** will be decoded for each passenger.

the learning models. For example, as shown in Fig. 2, if we directly predict the next stations of these two sequences, the next stations after station **B** are difficult to predict, because station **A** and **D** have the same probability. However, if we could sense semantics of stations for each user and “translate” them to their semantic structure data, such as “Home”, “Office,” “Mall” and so on, it is not difficult to predict the next-station as **H**, which is abbreviation of “Home,” that is, **A** for girl and **D** for boy. In this case, a sequence prediction model can predict the next-station correctly.

To achieve this goal, we design two units to encode an observed travelling sequence into a structured sequence with consideration of individual travelling semantics, and recover the travelling sequence by a co-trained decoder. These two processes are named “Translation” and “De-Translation” separately. Because of the Semantic Sensing ability, this module is named as SS-LSTM.

To design a semantic sensing module, we extract d -dimensional features of a station from transition records. We use vector f_m^p grouped by features to replace the tuple (s_m^p, t_m^p) , and f_m^p means the m -th station information of passenger p which is extracted from both spatial data s_m^p and temporal data t_m^p . This process should consider reducing the loss of information. Travelling behavior sequence T^p can be updated as:

$$T^{p'} = [f_1^p, f_2^p, \dots, f_{n_p}^p]. \quad (3)$$

Then we extend the LSTM by adding two trainable matrices W_{in} and W_{out} to achieve semantic understanding.

After multiplying the matrix W_{in} with input feature vectors to simulate the process of translation, we get the updated equations of (1) as follows:

$$\begin{aligned} i_m &= \sigma_i(W_{xi}f_m^p W_{in} + W_{hi}h_{m-1} + b_i), \\ f_m &= \sigma_f(W_{xf}f_m^p W_{in} + W_{hf}h_{m-1} + b_f), \\ \tilde{c}_m &= \sigma_c(W_{xc}f_m^p W_{in} + W_{hc}h_{m-1} + b_c), \\ c_m &= f_m \odot c_{m-1} + i_m \odot \tilde{c}_m, \\ o_m &= \sigma_o(W_{xo}f_m^p W_{in} + W_{ho}h_{m-1} + b_o), \end{aligned} \quad (4)$$

$$h_m = o_m \odot \sigma_h(c_m), \quad (5)$$

where i_m, f_m, o_m represent the *input*, *forget*, and *output* gates of the m -th object respectively. c_m is the cell activation vector. f_m^p and h_m represent the input feature vector and the hidden output vector respectively. $W_{hi}, W_{hf}, W_{hc}, W_{ho}, W_{xi}, W_{xf}, W_{xc}$, and W_{xo} are weight parameters connecting different inputs and gates. b_i, b_f, b_c , and b_o are corresponding biases.

We add a matrix W_{out} to achieve De-translation based on hidden output vector in (5) and use output vector $\hat{y}_{n_p}^p$ to represent the prediction results of passenger p as shown in (6).

$$\hat{y}_{n_p}^p = h_{n_p}^p W_{out}. \quad (6)$$

Euclidean distance between the last output vector $\hat{y}_{n_p}^p$ of passenger p and true label vector $y_{n_p}^p$ is used to measure performance of training. Loss function \mathcal{L} in (2) can be updated to follows:

$$\mathcal{L} = \sum_{p=1}^N \left\| \hat{y}_{n_p}^p - y_{n_p}^p \right\| = \sum_{p=1}^N \left\| h_{n_p}^p W_{out} - y_{n_p}^p \right\|. \quad (7)$$

Our goal is to minimize the loss \mathcal{L} through training weighted parameters, biases and two translation matrices.

For each epoch, we update two translation matrices as the following process. Some simplifications are made to accelerate our training process. Firstly, translation and de-translation are two reversible processes, so we set W_{in} and W_{out} with the relationship in (8).

$$W_{in} = W_{out}^{-1}. \quad (8)$$

Secondly, we redefine (9) based on (7) and the updating process of W_{out} is shown in (10).

$$g(W_{out}) = \sum_{p=1}^N \frac{1}{2} \left\| h_{n_p}^p W_{out} - y_{n_p}^p \right\|^2, \quad (9)$$

$$\begin{aligned} \nabla_{W_{out}} g &= \frac{\partial g(W_{out})}{\partial W_{out}} = \sum_{p=1}^N h_{n_p}^{pT} (h_{n_p}^p W_{out} - y_{n_p}^p), \\ W_{out}' &= W_{out} - lr \nabla_{W_{out}} g, \end{aligned} \quad (10)$$

where $0 < lr < 1$ means learning-rate factor, $h_{n_p}^{pT}$ is the transposed matrix of $h_{n_p}^p$ and W_{out} becomes W_{out}' after update. Through limited training process, we could improve the prediction performance according to above operations.

D. MULTI-PATTERN LEARNING AND PL-LSTM

There is no doubt that a LSTM model could uncover the pattern by mining data generated by a same pattern, but what will happen if the collected data are generated by multiple patterns? Is it a reasonable way to fit these data with one single model? Actually most of our data collected in real life are generated by different patterns. Taking the passenger trajectories for example, some of the passengers are citizens, they travel regularly to or from work, but the others are travellers, they come to this city for business or tourism. It is obvious that the travelling dynamics of different people are quite different. In this case, it is necessary to employ multiple learning models to handle the multiple dynamics. This is the motivation of designing multi-pattern LSTM model to improve the prediction performance. In this paper, we design a novel model called PL-LSTM with multiple cores which identifies the travelling patterns of passengers at first and then handles different patterns with different learning modules. Fig. 3 shows its sketch map.

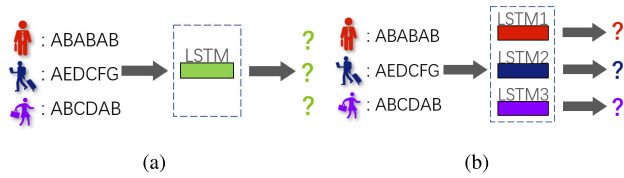


FIGURE 3. Sketch map of PL-LSTM. (a) Using one model to uncover all patterns. (b) Identifying passengers' patterns then handle different patterns with different modules.

This design is derived from a famous unsupervised learning method, namely, Self-Organizing Maps (SOM) [34]. The classical SOM is a two-dimensional array of neurons, indicated by

$$\mathbb{N} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_K\}, \quad (11)$$

where K is the number of neurons, \mathbf{n}_k is the vector representation of k -th patterns. When there comes a new data object \mathbf{x}_j , which first finds the nearest neurons w.r.t Euclidean distance $\text{argmin}_k \|\mathbf{x}_j - \mathbf{n}_k\|$, $k \in 1, 2, \dots, K$, then updates the \mathbf{n}_k according to previous representation and \mathbf{x}_j .

Different from the existing SOM model, the problem studied in this paper is a supervised learning task. Therefore, the above fitting measure needs to be changed. Assume we have K single LSTM models in our model pool. The i -th model is marked as m_i and the j -th observation sequence is marked as Seq^j . So we have the following equation:

$$m_i^j = f(m_i^{j-1}, Seq^j, lr_{c_j,i}), \quad (12)$$

which means the j -th state of m_i is defined by its $(j-1)$ -th state, the j -th sequence Seq^j , and learning rate function $lr_{c_j,i}$, which is used to determine the impact of sequence Seq^j on the model m_i . The first subscript c_j is defined by the following condition,

$$\forall i, \|Seq^j - m_c^j\| \leq \|Seq^j - m_i^j\|. \quad (13)$$

m_c is the model that matches best with Seq^j and we call it *Winner Model*. c_j refers to the Winner Model m_c for the j -th sequence. The “distances” between sequence Seq^j and models are put into set \mathbb{L}^j as shown in (14).

$$\mathbb{L}^j := \{\|Seq^j - m_1^j\|, \dots, \|Seq^j - m_K^j\|\}. \quad (14)$$

To follow the principle that sequence Seq^j has the greatest contribution to winner model and declining contribution to the others, we define the learning rate function $lr_{c_j,i}$ as follows:

$$lr_{c_j,i} = LR^* * \exp\left(-\frac{\|Seq^j - m_i^j\| - \|Seq^j - m_c^j\|}{\max \mathbb{L}^j - \|Seq^j - m_i^j\| + a}\right), \quad (15)$$

where $0 < LR^* < 1$ means the learning-rate constant which should be defined before experiments and the range of $lr_{c_j,i}$ is between 0 and 1. a is set to be an extremely small number to avoid any divided by zero or overflow. This function can

measure whether the sequence is applicable to the model. For example, if $i = c$, then $lr_{c_j,c} = LR^*$, which means this model m_c^j can get biggest learning rate LR^* for sequence Seq^j ; if $i = z$ and z satisfies that $\|Seq^j - m_z^j\| = \max \mathbb{L}^j$, then $lr_{c_j,z} = 0$, means this sequence has no contribution to the model and will not be fed into it. With these settings and conditions above, each sequence can find its own pattern and choose most suitable model for itself.

Different datasets may have different pattern numbers, and the choice of core number K should be decided according to the specific dataset. Considering the balance between computational complexity and prediction performance, K can be chosen from 2 to 10 based on the experiment results.

E. OVERVIEW OF SP-LSTM

To integrate the semantic sensing module and multi-pattern learning module introduced above together, we provide a unifying learning framework named Semantic and multi-Pattern Long Short-Term Memory (SP-LSTM) model to predict passengers' next behaviors. The proposed framework is illustrated in Fig. 4.

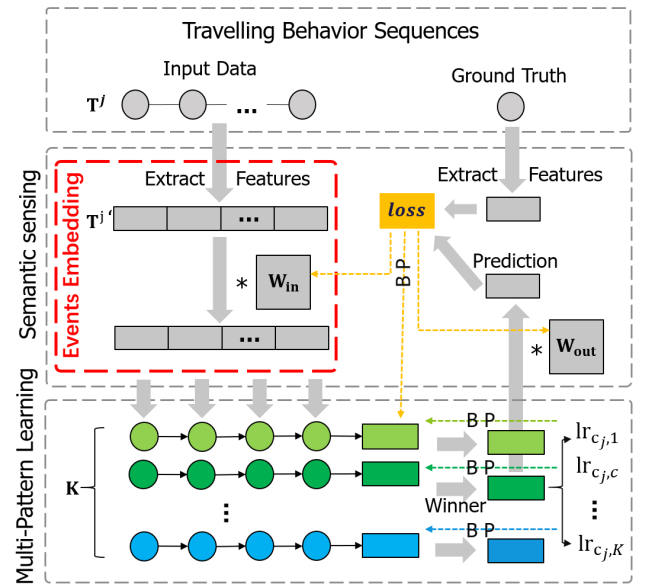


FIGURE 4. Architecture of SP-LSTM. It contains three modules which are travelling behavior sequences module, semantic sensing module and multi-pattern learning module, respectively.

As shown in Fig. 4, this structure has three modules, travelling behavior sequences module, semantic sensing module and multi-pattern learning module, respectively. The process of training is as follows: firstly, feature vectors will be extracted from sequence T^j to $T^{j'}$ before it is fed into this model and then multiply with a matrix W_{in} . This procedure is called “Events Embedding”; secondly, this matrix sequence will be sent into each LSTM in PL-LSTM to judge which model is the winner model, and their learning rate function $lr_{c_j,i}$ will be calculated; thirdly, the last hidden state of winner model will be sent back to semantic sensing module

and translate back to prediction result through the matrix W_{out} ; finally, the loss between ground truth's feature vector and prediction result will achieve the process of back propagation [35] with updating W_{in} , W_{out} of SS-LSTM and PL-LSTM.

F. THEORETICAL ANALYSIS

1) RELATIONSHIP BETWEEN SS-LSTM AND BASIC LSTM

As we can see from (4-6), SS-LSTM has two matrices W_{in} and W_{out} more than basic LSTM in structure. And if we set these two matrices both equal to unit matrix I , (4) and (6) can be simplified to the following equations:

$$\begin{aligned}
 i_m &= \sigma_i(W_{xi}f_m^p + W_{hi}h_{m-1} + b_i), \\
 f_m &= \sigma_f(W_{xf}f_m^p + W_{hf}h_{m-1} + b_f), \\
 \tilde{c}_m &= \sigma_c(W_{xc}f_m^p + W_{hc}h_{m-1} + b_c), \\
 c_m &= f_m \odot c_{m-1} + i_m \odot \tilde{c}_m, \\
 o_m &= \sigma_o(W_{xo}f_m^p + W_{ho}h_{m-1} + b_o), \\
 \hat{y}_{np}^p &= h_{np}^p.
 \end{aligned}
 \tag{16}$$

And if we regard f_m^p as the input vector of subsection III-B, SS-LSTM can be simplified as the basic LSTM, which means that the basic LSTM is a special case of SS-LSTM.

2) RELATIONSHIP BETWEEN PL-LSTM AND BASIC LSTM

We set k LSTMs in PL-LSTM which means dividing data into k patterns, and if we set $k = 1$, it is can be inferred from (13) that the only one model is the Winner Model, and the learning rate function $lr_{c_j,i}$ can be simplified to the next equations:

$$\begin{aligned}
 lr_{c_j,c} &= LR^* * \exp\left(-\frac{\|Seq^j - m_c^j\| - \|Seq^j - m_c^i\|}{\|Seq^j - m_c^j\| - \|Seq^j - m_c^i\| + a}\right) \\
 &= LR^*,
 \end{aligned}
 \tag{17}$$

and PL-LSTM changes back into basic LSTM with learning rate LR^* . That also means basic LSTM is one of special cases of PL-LSTM.

IV. EXPERIMENTS

In this section, we evaluate the proposed methods by experiments on real-world data sets. We first validate the design of our methods partly, then compare our methods with baselines by prediction accuracy.

A. VERIFICATION EXPERIMENTS

1) IMPORTANCE OF SEMANTIC SENSING

To prove the importance of semantic sensing in the process of sequence prediction, we design a verification experiment.

We select 22,556 travel sequences with only two different stations in each sequence to form test data set 1, and 44,921 sequences with three different stations to form test data set 2. We design the verification experiment as following steps: in test data set 1, we mark the station appearing more in each sequence as ‘‘H,’’ while appearing less as ‘‘O’’.

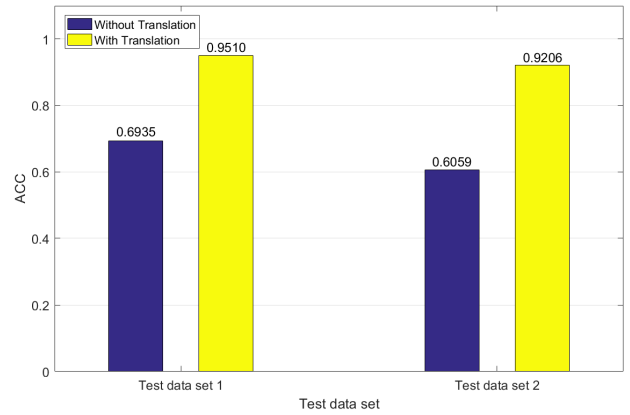


FIGURE 5. Performance comparisons on test data sets with translating process or not. ACC is used to measure the predicting accuracy of results.

And in test data set 2, the station which appears most in each sequence is marked as ‘‘H,’’ the minimal station is marked as ‘‘T’’, the others is ‘‘O’’. And these translation processes are also recorded. Then we feed these translated sequences into basic LSTM. After training process, we translate the prediction results back to their original stations based on the translation records. Of course we also feed original travel sequences into basic LSTM model as controlled experiment. ACC(Accuracy) is used to measure the predicting performance after all of these process. As we can see from Fig.5, the accuracy of prediction can be improved by 30% after translation, and that also proves the process of translation can be very helpful for prediction.

Therefore, a prediction model without semantic understanding is not able to achieve desired prediction results.

2) IMPORTANCE OF CONSIDERING MULTIPLE PATTERNS

To prove that model’s performance on prediction task will be misled by multi-pattern data, we design a set of experiments using three kinds of chaotic time series to model behavior sequences with multiple patterns. Mackey Glass, Lorenz, and Rossler are selected to simulate three models of travel data and their mixed data analogy the real-world data we collected.

In the first experiment, we select 10,000 sequences of length 100 as experimental data for each kind of chaotic time sequences and make them to three data sets. Then we choose 3,000 sequences of Glass, 4,000 of Lorenz, and 3,000 of Rossler to build up the fourth data set. For each data set we split 80% of sequences as training set and the rest as testing set. We add different levels of noise to simulate real-world data set before feed them to the basic LSTM model. RMSE(Root Mean Square Error) is used to measure the losses of predicting results with the true labels. The results is shown as Fig.6. As we can see from this figure, with the increase of noise, LSTM performs much worse on mixed data set than any other separate data set. As collected real-world data often have many noises and contains more than one pattern, it is not strange that LSTM has such bad performance on real-world data.

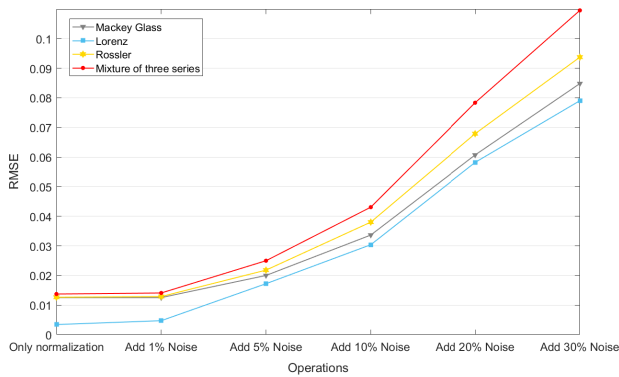


FIGURE 6. Performance comparisons on different data sets with LSTM model. The RMSE of model which is trained on mixture of three series is much bigger than that on the other separate data set.

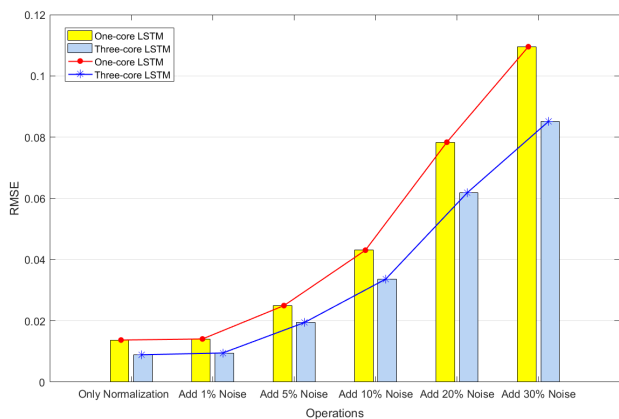


FIGURE 7. Performance comparisons on data set with one-core or three-core LSTM models. The bar chart can reflect the gap between two models under the same operation and line chart can show the trend of RMSE with adding increasing noise.

Then we do our second experiment and this experiment is based on the fourth data set. Because these three kinds of series have different regulations which one-core LSTM(basic LSTM) model may has difficulty to handle, we design a special LSTM model with three cores which can divide different kinds of sequences to different models. In order to prove experiment results batter, we also make one-core LSTM as controlled experiment which is trained at the same experiment conditions. RMSE is still our target to measure the losses. The results is shown as Fig.7 and it can be inferred definitely that three-core LSTM has batter performance on multi-pattern data set. That also shows the necessity of considering multi-mode factors.

B. DATASETS AND EXPERIMENT SETTINGS

The datasets are collected from citizen transition records of buses and railways in Beijing, China, dating from June to September 2017. To observe the performance of the prediction models in different cases, the datasets are divided into three subsets according to the number of stations in passenger

TABLE 1. Statistics of three datasets.

	TBD1	TBD2	TBD3
counts of sequences	4,500	4,500	4,500
counts of stations	2,813	4,306	4,762
kinds of stations in each sequence	1-8	9-16	17-24

travelling sequences. They are named as TBD1, TBD2, and TBD3 which are abbreviations of the 1-th, 2-th, and 3-th Traveling Behavior Dataset, respectively. Table 1 presents the details of them. Each subset consists of 4,500 traveling behavior sequences which present 4,500 people’s traveling behavior records with different lengths from 20 to 400. For example, passenger p only has been to three different stations and his travelling sequence is longer than 20 but shorter than 400, his sequence will be divided into TBD1.

Each travelling sequence contains a series of tuples $\langle p_id, s_id, t \rangle$, where p_id is the passenger ID, s_id is the station ID, t is the time label.

For each dataset, 80% of the travelling sequences are used to train prediction models. 20% of them are used as testing sequences. For a testing sequence $T^u := [(s_1^u, t_1^u), (s_2^u, t_2^u), \dots, (s_{n^u}^u, t_{n^u}^u)]$, the tuples with index from 1 to $(n^u - 1)$ are assumed to be known to us, $s_{n^u}^u$ is the station needing to be predicted by our model.

At the beginning of semantic sensing, we extract 11-dimensional features from T^u to ensure our translating matrices can get inner laws better. And these features are shown in Table 2. We select these features based on our general knowledge and each of them can reflect the function of station from one aspect.

TABLE 2. Features extracted from data.

Index	Features
f^1	Frequency of check-in from 7:00 to 10:00
f^2	Frequency of check-out from 17:00 to 20:00
f^3	Frequency of check-in as the first station in one day
f^4	Frequency of check-out as the last station in one day
f^5	Frequency of check-out from 7:00 to 10:00
f^6	Frequency of check-in from 17:00 to 20:00
f^7	Frequency of check-out in half hour before next check-in
f^8	Frequency of check-in in half hour after last check-out
f^9	If station appears only once as check-out station
f^{10}	Frequency of check-in after 21:00
f^{11}	Frequency of check-in in 0.5-3 hours after last check-out

C. BASELINES

Both the classical and state-of-the-art methods are employed to provide benchmark performance. The classical sequence prediction methods include:

- 1) **Most Pop** This method regards the most popular station of each passenger as the prediction result.
- 2) **Markov Model (MM)** It is a Markov chain model based on passenger travelling prediction method.

Deep learning based sequence prediction methods include:

- 1) **RNN** We feed sequences handled by one-hot encoding to basic RNN with two superimposed layers.

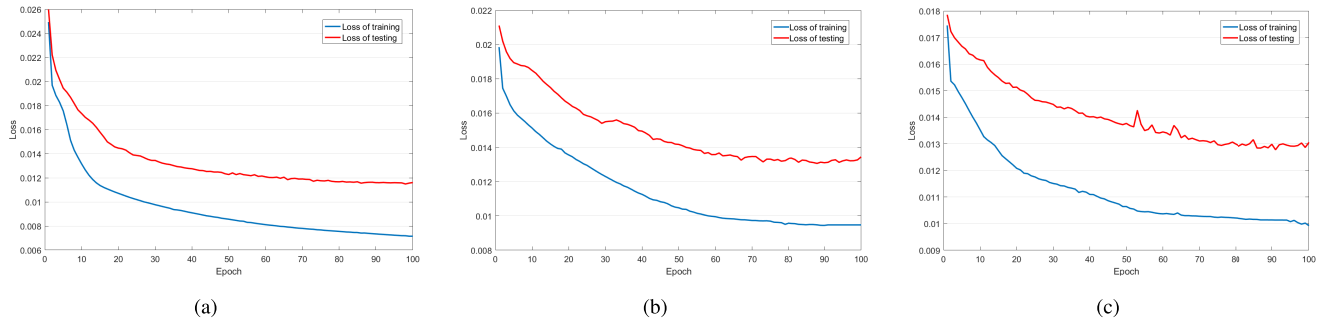


FIGURE 8. The losses of training and testing on three data set. (a) Losses on TBD1. (b) Losses on TBD2. (c) Losses on TBD3.

- 2) **LSTM_1** This method has been introduced in section III-B, which is widely applied in prediction field.
- 3) **GRU** As another version of LSTM, GRU has some advantages in precision and the speed of training process.

D. EVALUATIONS

To get a comprehensive evaluation of the proposed method, the following four metrics are used to measure the consistency of prediction results with the ground-truth.

- 1) **Precision (PRE)** is the ratio of correct prediction to total prediction, which measures the accuracy of prediction results.

$$PRE_n = \frac{\alpha_{nn}}{\sum_{i=1}^I \alpha_{in}}$$

where PRE_n denotes prediction precision of the n -th ground truth, $\{\alpha_{ij}\}$ is the confusion matrix, α_{ij} is the number of cases that, prediction result is e_j but ground truth is e_i , I is the total number of related stations.

- 2) **Recall (REC)** is the ratio of correct prediction to total ground truth, which measures the coverage rate of prediction.

$$REC_n = \frac{\alpha_{nn}}{\sum_{j=1}^J \alpha_{nj}}$$

where REC_n means prediction recall of the n -th ground truth. J is the total number of predicted stations.

- 3) **F1-Score (F1)** is a comprehensive measurement to evaluate model’s performance, which is defined as

$$F1_n = \frac{2PRE_n \times REC_n}{PRE_n + REC_n}$$

where $F1_n$, PRE_n , and REC_n represent the F1-Score, Precision, and Recall of the n -th ground truth, respectively. In the experiments, the average of $\{F1_1, F1_2, \dots\}$ is used to compare the prediction methods.

- 4) **Accuracy (ACC)** is the most intuitive evaluation which can show the ratio of correct prediction to all prediction,

$$ACC = \frac{\sum_{i=1}^I \alpha_{ii}}{\sum_{i=1}^I \sum_{j=1}^J \alpha_{ij}}$$

TABLE 3. Comparison of the proposed method with baselines.

Dataset	Method	REC	PRE	F1	ACC
TBD1	Most Pop	0.2223	0.2340	0.2176	0.3711
	MM	0.3112	0.3234	0.3033	0.4873
	RNN	0.0027	0.0030	0.0004	0.0322
	LSTM_1	0.0022	0.0101	0.0010	0.0422
	GRU	0.0029	0.0144	0.0017	0.0433
	SP-LSTM	0.5860	0.5828	0.5618	0.7267
TBD2	Most Pop	0.1466	0.1467	0.1378	0.3087
	MM	0.1955	0.1987	0.1856	0.3587
	RNN	0.0012	0.0012	0.0001	0.0158
	LSTM_1	0.0019	0.0021	0.0003	0.0278
	GRU	0.0018	0.0005	0.0002	0.0233
	SP-LSTM	0.2899	0.2835	0.2677	0.4256
TBD3	Most Pop	0.1114	0.1148	0.1036	0.2224
	MM	0.1462	0.1515	0.1371	0.2558
	RNN	0.0009	0.0001	0.0001	0.0089
	LSTM_1	0.0010	0.0018	0.0002	0.0133
	GRU	0.0012	0.0018	0.0002	0.0178
	SP-LSTM	0.2213	0.2260	0.2155	0.3022

E. RESULTS AND DISCUSSIONS

Table 3 presents the comparison of our method with baselines. It can be seen that our method outperforms other baselines significantly on all the three data sets in terms of all the four performance evaluation metrics. Compared with the classical deep learning methods, the prediction accuracy of our method has been improved from less than 1% to 72.67%, 42.56%, and 30.33% respectively, the other evaluation metrics have also witnessed a significant improvement of our method in prediction performance. An interesting observation is that the performance of statistics-based sequence prediction methods is also better than the deep learning baselines, the reason is that statistics-based models are implemented at an individual level, and therefore, they are not affected by the confusion of multiple travelling patterns. Figure 8 shows the losses of SP-LSTM training on three data sets and we can conclude that SP-LSTM model can be trained to convergence in our experiments. Fig. 9 presents the contributions of the two learning modules to SP-LSTM. As shown

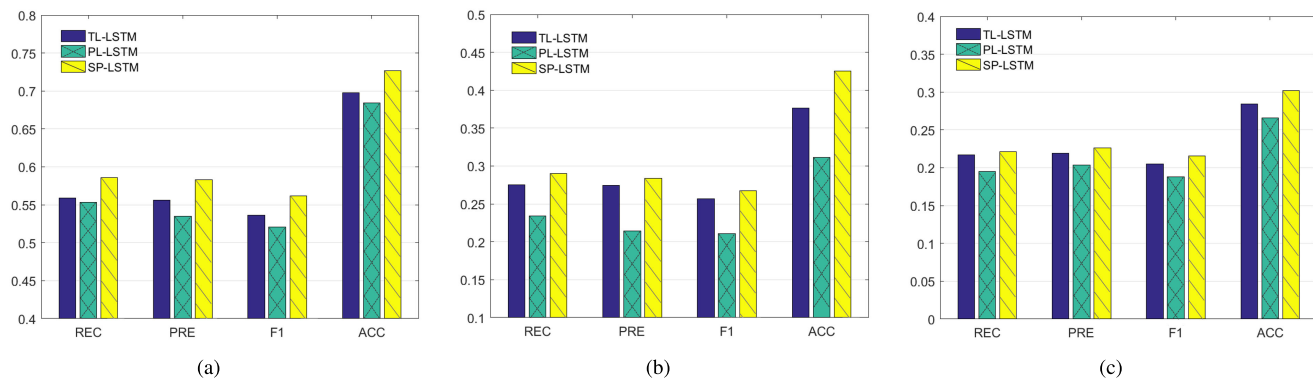


FIGURE 9. The comparison of variants of our method. (a) Performance on TBD1. (b) Performance on TBD2. (c) Performance on TBD3.

in each of sub-graph, the performance of SP-LSTM is the best among the three models, which suggests the effectiveness of both two learning modules. Additionally, the performance of SS-LSTM performance is always better than PL-LSTM, which demonstrates that the semantic sensing module is more useful than multi-pattern learning module.

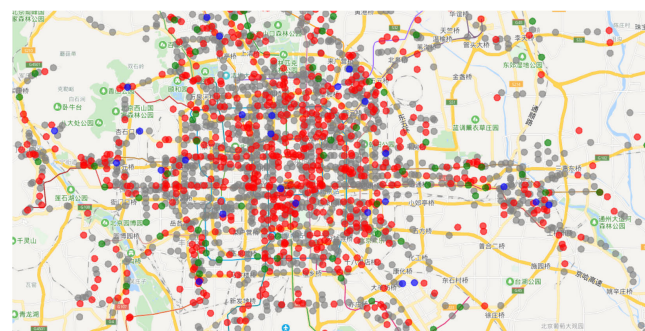


FIGURE 10. Semantic understanding shown in map. Stations with same color are regarded as belonging to the same semantic.

To illustrate how the semantic sensing module works, the stations are represented by vectors with consideration of individual travelling semantics, and then the stations are divided into four clusters. Fig. 10 presents the distributions of these stations belonging to different clusters, where gray points are “Home” stations, red points denote “Office” stations, blue points indicate tourist spots, and green points could be the traffic transfer stations. To the best of our knowledge, the travelling sequences have not been decomposed semantically as our method did in Fig. 10. Obviously, the semantic sensing module can help improve the interpretability of the sequence prediction models.

Compared with the existing sequence prediction models, the proposed SP-LSTM is equipped with two novel learning modules: semantic sensing module which translates the raw travelling sequence into a semantic sequence, and multi-pattern learning module which is designed to capture the multiple travelling patterns of passengers with various travelling purposes. All the above facts demonstrate the effectiveness of the two proposed learning modules.

V. CONCLUSION

This paper aims to improve the passenger behavior prediction method from two perspectives: 1) uncovering the semantics of travelling sequence at an individual level, and 2) discovering the multiple travelling patterns and modeling them simultaneously. Correspondingly, two novel learning modules were proposed: 1) an semantic sensing module that translates the raw travelling sequence into a semantic representation, and 2) a multi-pattern learning module that constructs exclusive prediction model for passengers with different travelling purposes. Finally, a unifying learning framework was designed to incorporate the above two learning modules with the LSTM model, the proposed method for passenger behavior prediction method was named SP-LSTM. Experiments were conducted on real-world data sets, SP-LSTM demonstrates significant superiority over the existing methods, which validates the effectiveness of the proposed learning modules.

REFERENCES

- [1] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [2] C. Whong. *NYC Taxi Data (FOI Led 2013-14)*. Accessed: 2019. [Online]. Available: http://chriswhong.com/open-data/foi_nyc_taxi/
- [3] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state Markov chains,” *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 1966.
- [4] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized Markov chains for next-basket recommendation,” in *Proc. WWW*, 2010, pp. 811–820.
- [5] J. Janssen and N. Limnios. *Semi-Markov Models and Applications*. Springer, 2013.
- [6] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. “Recurrent neural network based language model,” in *Proc. INTER-SPEECH*, 2010, pp. 1045–1048.
- [7] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu, “Sequential click prediction for sponsored search with recurrent neural networks,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1–7.
- [8] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, “Session-based recommendations with recurrent neural networks,” 2015, *arXiv:1511.06939*. [Online]. Available: <https://arxiv.org/abs/1511.06939>
- [9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, pp. 1735–1780, Dec. 1997.
- [10] D. Neil, M. Pfeiffer, and S. C. Liu, “Phased LSTM: Accelerating recurrent network training for long or event-based sequences,” in *Proc. NIPS*, 2016, pp. 3882–3890.

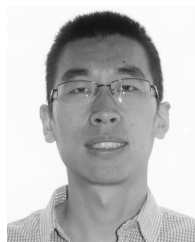
- [11] K. Cho, B. van Merriënboer, C. C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. EMNLP*, 2014, pp. 1724–1734.
- [12] D. Mcfadden, "The measurement of urban travel demand," *J. Public Econ.*, vol. 3, no. 4, pp. 303–328, 1974.
- [13] Z. Yao, Y. Fu, B. Liu, Y. Liu, and H. Xiong, "POI recommendation: A temporal matching between POI popularity and user regularity," in *Proc. IEEE ICDM*, Dec. 2016, pp. 549–558.
- [14] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai, "What to do next: Modeling user behaviors by time-LSTM," in *Proc. IJCAI*, 2017, pp. 3602–3608.
- [15] P. Zhao, H. Zhu, Y. Liu, Z. Li, J. Xu, and V. S. Sheng, "Where to go next: A spatio-temporal LSTM model for next POI recommendation," 2018, *arXiv:1806.06671*. [Online]. Available: <https://arxiv.org/abs/1806.06671>
- [16] Z. Yao, Y. Fu, B. Liu, W. Hu, and H. Xiong, "Representing urban functions through zone embedding with human mobility patterns," in *Proc. IJCAI*, 2018, pp. 3919–3925.
- [17] H. Ying, J. Wu, G. Xu, Y. Liu, T. Liang, X. Zhang, and H. Xiong, "Time-aware metric embedding with asymmetric projection for successive POI recommendation," *World Wide Web*, vol. 22, no. 5, pp. 2209–2224, Jun. 2018.
- [18] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proc. IJCAI*, vol. 13, 2013, pp. 2605–2611.
- [19] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI recommendation," in *Proc. IJCAI*, 2015, pp. 2069–2075.
- [20] Z. Zhang, C. Li, Z. Wu, A. Sun, D. Ye, and X. Luo, "NEXT: A neural network framework for next POI recommendation," 2017, *arXiv:1704.04576*. [Online]. Available: <https://arxiv.org/abs/1704.04576>
- [21] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proc. ACM SIGIR*, 2016, pp. 729–732.
- [22] R. Guidotti, G. Rossetti, L. Pappalardo, F. Giannotti, and D. Pedreschi, "Next basket prediction using recurring sequential patterns," 2017, *arXiv:1702.07158*. [Online]. Available: <https://arxiv.org/abs/1702.07158>
- [23] C. Estebann, D. Schmidt, D. Krompaß, and V. Tresp, "Predicting sequences of clinical events by using a personalized temporal latent embedding model," in *Proc. ICHI*, 2015, pp. 130–139.
- [24] Y. Yang, P. A. Fasching, M. Wallwiener, T. N. Fehm, S. Y. Brucker, and V. Tresp, "Predictive clinical decision support system with RNN encoding and tensor decoding," 2016, *arXiv:1612.00611*. [Online]. Available: <https://arxiv.org/abs/1612.00611>
- [25] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware LSTM networks," in *Proc. ACM SIGKDD*, 2017, pp. 65–74.
- [26] L. Li, H. Jing, H. Tong, J. Yang, Q. He, and B. Chen, "NEMO: Next career move prediction with contextual embedding," in *Proc. WWW*, 2017, pp. 505–513.
- [27] C. Yang, M. Sun, W. Zhao, Z. Liu, and E. Y. Chang, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, pp. 36:1–36:28, 2017.
- [28] Q. Liu, S. Wu, and L. Wang, "Multi-behavioral sequential prediction with recurrent log-bilinear model," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1254–1267, Jan. 2017.
- [29] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, Corvallis, OR, USA, Jun. 2007, pp. 641–648.
- [30] Y. Liu, C. Liu, B. Liu, M. Qu, and H. Xiong, "Unified point-of-interest recommendation with temporal interval assessment," in *Proc. ACM SIGKDD*, 2016, pp. 1015–1024.
- [31] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. AAAI*, 2016, pp. 194–200.
- [32] H. Ying, F. Zhuang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention networks," in *Proc. IJCAI*, 2018, pp. 3926–3932.
- [33] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*. [Online]. Available: <https://arxiv.org/abs/1308.0850>
- [34] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.
- [35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.



HAIQUAN WANG received the Ph.D. degree in computer science from Beihang University, in 2013. He is currently an Associate Professor with Beihang University, Beijing, China. His research interests include intelligent transport systems and software engineering. He has been conducting studies on intelligent transport systems in recent years, hosting, or participating in many national projects including the National Nature Science Foundation of China, National High Technology Research and Development Program of China.



XIN WU received the B.E. degree from the College of Science, Beijing Forestry University, Beijing, China, in 2016. She is currently pursuing the master's degree with the School of Software, Beihang University. Her research interests include sequence prediction, behavior analysis, data mining, and mining regulations hidden in travel behavior sequences and use them to guide people to a better life with intelligent transportation.



LEILEI SUN received the B.S. and M.S. degrees from the School of Control Theory and Control Engineering, Dalian University of Technology, in 2009 and 2012, respectively, and the Ph.D. degree from the Institute of Systems Engineering, Dalian University of Technology, in 2017. He is currently an Assistant Professor with the State Key Laboratory of Software Development Environment and Big Data Brain Computing Laboratory (SKLSDE and BDBC Lab), Beihang University, Beijing, China. He was a Postdoctoral Research Fellow of the School of Economics and Management, Tsinghua University. He has published in the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). He has published many articles in the IEEE TRANSACTIONS ON DATA AND KNOWLEDGE ENGINEERING (TKDE) and *Knowledge and Information Systems* (KAIS). His research interests include machine learning and data mining.



BOWEN DU received the Ph.D. degree in computer science and engineering from Beihang University, Beijing, China, in 2013. He is currently an Assistant Professor with the State Key Laboratory of Software Development Environment, Beihang University. His research interests include smart city technology, multisource data fusion, and traffic data mining.

...