Front.Comput.Sci.
DOI

RESEARCH ARTICLE

Where to go? Predicting next location in IoT environment

Hao LIN, Guannan LIU (✉), Fengzhi LI, Yuan ZUO

School of Economics and Management, Beihang University, Beijing 100191, China

Abstract Next location prediction has aroused great interests in the era of Internet of Things (IoT). With the ubiquitous deployment of sensor devices, *e.g.*, GPS and Wi-Fi, IoT environment offers new opportunities for proactively analyzing human mobility patterns and predicting user's future visit in low cost, no matter outdoor and indoor. In this paper, we consider the problem of next location prediction in IoT environment via a session-based manner. We suggest that user's future intention in each session can be better inferred for more accurate prediction if patterns hidden inside both trajectory and signal strength sequences collected from IoT devices can be jointly modeled, which however existing state-of-the-art methods have rarely addressed. To this end, we propose a Trajectory and SIgnal Sequence (TSIS) model, where the trajectory transition regularities and signal temporal dynamics are jointly embedded in a neural network based model. Specifically, we employ Gated Recurrent Unit (GRU) for capturing the temporal dynamics in the multivariate signal strength sequence. Moreover, we adapt gated Graph Neural Networks (gated GNNs) on location transition graphs to explicitly model the transition patterns of trajectories. Finally, both the low-dimensional representations learned from trajectory and signal sequence are jointly optimized to construct a session embedding, which is further employed to predict the next location. Extensive experiments on two real-world Wi-Fi based mobility datasets demonstrate that TSIS is effective and robust for next location prediction compared with other competitive baselines.

Keywords Internet of Things, Next Location Prediction, Neural Networks, Trajectory, Signal.

Received month dd, yyyy; accepted month dd, yyyy

E-mail: *liugn@buaa.edu.cn*

1 Introduction

Enabled by Internet of Things (IoT), human movement, no matter outdoor and indoor, can be well tracked by GPS, Wi-Fi access points, and other IoT devices in a ubiquitous way. These online footprints in IoT environment thus offer new opportunities for more accurately predicting user's next location without human intervention, which in turn helps to enhance system design [1] or improve location based services [2] in low cost. For example, with the indoor moving trajectories tracked by Wi-Fi facilities in a shopping mall, people's mobility patterns can be inferred and the next location they tend to visit can also be predicted, which can help manage the store locations by rearranging the frequently co-visited stores close to each other, and meanwhile design proactive personalized advertisements to increase the exposure of stores and products. Also, with human mobility patterns analyzed, smart energy scheduling can be better achieved [3]. Besides indoor movements, outdoor trajectories, *e.g.*, traces collected from wireless PDAs in a campus [1], can be utilized for determining user's locations and optimizing network load balancing.

In this paper, we study the problem of next location prediction in IoT environment, in which sensor signal information collected by IoT devices (*e.g.* Wi-Fi) can be easily obtained. The key aspect of this problem is how to effectively and accurately capture user's moving regularities in the noisy IoT environment. Prior state-of-the-art approaches usually make great efforts on revealing the underlying patterns in human's historical visited locations, *e.g.*, categorical location labels or real-valued check-in tuples associated with latitude and longitude, also known as

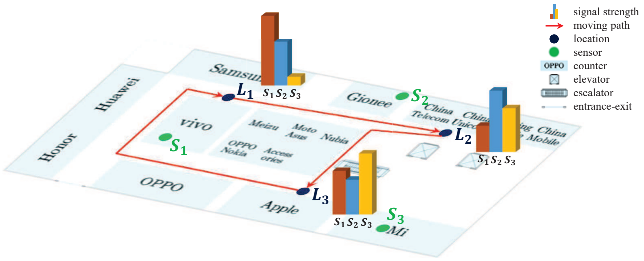


Fig. 1 An illustrative example of observed customer trajectory and signal strength detected by Wi-Fi sensors.

trajectories. For example, Yao et al. [4] proposed to jointly learn embeddings of multiple factors and transition patterns of a recurrent neural network for next location prediction in semantic trajectories. Feng et al. [5] designed a multi-modal embedding recurrent neural network with attention mechanisms for predicting human mobility. Feng et al. [6] proposed to address the drawbacks of matrix factorization by learning sequential and personalized patterns simultaneously on check-in sequences.

However, none of the above methods consider trajectory and the corresponding signal strength sequence simultaneously, which are in fact essential to accurately inferring user's next location to be visited in IoT environment. We show an illustrative example of observed customer trajectory and signal strength detected by three Wi-Fi sensors on a real-world dataset in Figure 1. We see that in this example the customer follows a walking path, denoted as "L_1->L_2->L_3->L_1". It is obvious to see that when the customer starts at location L_1, the signal strength received from sensor S_1 is the strongest among the three sensors, which indicates that the customer is currently near sensor S_1. However, one critical point that can easily be overlooked is that the second strongest signal strength received at location L_1, *i.e*, the signal detected by sensor S_2, which may indicate the next possible visited location is likely to be near sensor S_2 and it turns out to be the case in the example. As such, we suggest that the signal strength collected from IoT devices may contain useful information for understanding people's movement patterns, especially the intention of next visit, which makes signal strength at least as important as user's trajectories for accurately predicting the next location.

Along this line, we propose a neural network model, named Trajectory and SIgnal Sequence (TSIS), to jointly learning movement patterns hidden inside both trajectory and signal strength sequences collected from IoT devices for next location prediction. Concretely, we first split each user's activities into separate sessions and handle sessions of each user independently. By this means, we solve the next location prediction problem via a session-based manner. The next key aspect is to learn a good representation of each session tailored for effective next location prediction. Inspired by the recent work of session-based recommendation [7], we adapt gated Graph Neural Networks (gated GNNs) [8] for learning session representation, *i.e.*, explicitly capturing the transition regularities from trajectories. Moreover, considering the temporal dynamics in the session received signal strength, we employ Gated Recurrent Unit (GRU) [9] to encode the consecutive sequential patterns into a low-dimensional embedding vector. Subsequently, a joint session embedding is achieved to incorporate both the trajectory transition regularities and the signal temporal dynamics. Finally, the joint session representation is used for predicting probabilities of candidate locations of people's next visit in a session.

Our contributions are summarized as follows:

- We suggest that signal strength is crucial for next location prediction in IoT environment, which can help to uncover the intention of user's next visit.
- A novel predictive model, named TSIS, is proposed for capturing both transition regularities in trajectories and temporal dynamics in signal strength. To the best of our knowledge, TSIS is the first model that incorporates both trajectory and signal strength for accurate next location prediction in IoT environment.
- Extensive experiments are performed on two real-world Wi-Fi based mobility datasets (*i.e.*, indoor Store and outdoor WTD). The experimental results demonstrate that TSIS outperforms competitive baseline methods for next location prediction.

The rest of this paper is organized as follows. We begin to give a literature review in Section 2. Section 3 introduces the data description and problem definition. In Section 4, we give the technical details of the proposed model. Experimental results are shown in Section 5, and we finally conclude our work in Section 6.

2 Related Work

In this section, to deepen our understanding of aspects relevant to this paper, we review some existing literature on predictive analytics using Wi-Fi signals, session-based recommendation and next location prediction.

2.1 Predictive Analytics using Wi-Fi Signals

Based on Wi-Fi signals, the most studied topic in the computer science community is indoor localization. Over the years, various solutions have been proposed for indoor localization [10–12]. Traditional localization usually relies on the trilateration and triangulation [13, 14] while the majority of recent approaches take a Wi-Fi fingerprinting process for processing Wi-Fi signals and associating them with locations [15–17]. The main idea of Wi-Fi fingerprinting is to characterize a position by signals, *e.g.*, vectors of RSSIs (Received Signal Strength Indicators) [18] from different Wi-Fi Access Points (APs) [19]. Besides indoor localization, other device-free sensing tasks based on Wi-Fi signals have recently attracted lots of attention. For instance, Li et al. [20] proposed to monitor human's vitality, *i.e.*, the information in which area the target is staying and whether the target is still or non-still. Sapiezynski et al. [21] inferred person-to-person physical proximity from the lists of Wi-Fi access points. Zhang et al. [22] presented a novel system for two Wi-Fi sensing problem, *i.e.*, gait identification and gesture recognition. Guo et al. [23] proposed to use existing Wi-Fi infrastructures to provide smart human dynamics monitoring. Kim and Lee [24] utilized in-store sensors to predict the revisit intention of customers.

2.2 Session-based Recommendation

Session-based recommendation shares the common characteristic with our mobility prediction scenario. That is the technologies such as MAC address of mobile devices or websites cookies are sometimes not reliable enough for accurate user identification and may raise great privacy concerns [25]. Additionally, user's short time behaviors are considered as more important than long ranged behaviors for understanding user's complex time-variant intentions. As such, user's sequential behaviors should be split into different sessions and each session can be handled without emphasis of its corresponding user. By this means, session-based traits modeling is recognized as the key for effective recommendation. Recently, session-based recommendation has been extensively studied and Recurrent Neural Networks (RNNs) have been popular in achieving state-of-the-art performance. Hidasi et al. [25] was the first to introduce RNN based approach, namely GRU4Rec, for session-based recommendation. More recently, several neural networks based-models are proposed in order to further improve the session-based recommendation performance [7, 26, 27]. Jannach and Ludewig [26] adapted k-nearest neighbor (kNN) for session-based recommendation and found that the best results were often achieved by combining kNN approach with GRU4Rec. Yuan et al. [27] introduced a convolutional generative model to address the modeling of long-range dependencies in session-based next item recommendation. Wu et al. [7] proposed to employ Graph Neural Networks (GNN) [28] for capturing complex transitions of items and achieved state-of-the-art performance compared to other competitive baselines. Along this line, we are motivated to model the trajectory sessions as graph data with GNN-based model [7].

2.3 Next Location Prediction

Typical next location prediction approaches aim at characterizing user movement regularity from user's historical visited locations, *e.g.*, categorical location labels or real-valued check-in tuples associated with latitude and longitude, also known as *trajectories*. Various predictive models have been proposed, including Matrix Factorization (MF) based methods [29], Markov Chain (MC) based models [30], hidden Markov models (HMMs) [31], periodic mobility models [32] and Recurrent Neural Network (RNN) based methods [4, 5, 33]. Duong-Trung et al. [29] developed a generative content-based regression model via matrix factorization to tackle the near real-time geolocation prediction problem. Rendle et al. [30] introduced personalized Markov chains relying on personalized transition matrices, which is expected to be able to capture both sequential effects and long term user interests. Mathew et al. [31] accounted for both location characteristics and each individual's previous actions for predicting users' future locations. It first clustered the locations from the trajectories and then employed a Hidden Markov Model for capturing each user's sequential patterns. Liu et al. [33] proposed to use RNN to predict the next location with spatial and temporal contexts considered. Yao et al. [4] proposed to jointly learn embeddings of multiple factors and transition patterns in semantic trajectories. Feng et al. [5] designed a multi-modal embedding recurrent neural network with attention mechanisms. From the application view, while next location prediction problem has been extensively studied on different kinds of mobility data, next POI recommendation [6, 34–36] remains promising. However, none of the above work considers both trajectory and signal strength, which are both essential for accurately inferring user's next location to be visited in IoT environment.

3 Preliminaries

Some preliminaries including data description and problem definition are given in this section.

3.1 Data Description

In this paper, we aim to leverage both the trajectories and signal strength collected from sensors for next location prediction. The deployed sensors enable us to collect Wi-Fi signals sent from any mobile devices (*e.g.*, PDAs, smart mobile phones or iPads). Each transaction record corresponds to one single Wi-Fi signal instance which includes anonymized device ID, sensor ID, timestamp and RSSI (Received Signal Strength Indicator) level. Wi-Fi signals are collected continuously from each mobile device at regular intervals.

3.2 Problem Definition

In what follows, we formally introduce the main concepts in our paper and formulate the next location prediction problem in IoT environment. Since the activity sequence of each mobile device (also regarded as an individual user) is usually very long, *e.g.*, the full length of each sequence might be several days or months, we follow the fashion of session-based recommendation to split each long sequence into multiple shorter subsequences, which we denote as *sessions*. This requires a conversion process, which is a key data preprocessing step in our settings. The general logic of the conversion is that a new session starts when the time interval between the current and the last detected visit is greater than a predefined threshold denoted as η. Since the values of η are slightly varied on different datasets, detailed discussion can be found in Section 5.1.

After obtaining each session s with n visits, we construct two types of sequences which are assumed to be very important in revealing user's mobility patterns, including *received signal strength sequence* RSS_s and *visited location sequence* L_s.

Definition 1. *A* ***received signal strength sequence*** RSS_s *is a sequence of received signal strength vectors ordered by timestamps,* $\mathsf{RSS}_s = \{rss_{si} | i \in \mathbb{Z}, 1 \leqslant i \leqslant n\}$*, and* $rss_{si} = \{rss_{sij} | j \in \mathbb{Z}, 1 \leqslant j \leqslant m\}$*, where* rss_{sij} *is the received signal strength (RSS) value of the j-th sensor observed by a mobile device at time slot i for session s. Note that m is the total number of sensors which is a fixed value in our settings and n, the number of visits, varies in different sessions. It can be seen that* RSS_s *can be viewed as a multivariate time series. Hereinafter, the terms such as signal, signal strength or signal sequence can be short for received signal strength sequence and we would interchangeably use these terms.*

Definition 2. *A* ***visited location sequence*** L_s *is an anonymous sequence of visited locations ordered by timestamps,* $\mathsf{L}_s = \{l_{si} | i \in \mathbb{Z}, 1 \leqslant i \leqslant n\}$*. Note that* $l_{si} \in V$ *represents a visited location of the user at time slot i in session s, where* $V = \{l_1, l_2, ..., l_v\}$ *denotes the set of unique locations in all sessions. It is obvious that a visited location sequence is exactly the same as trajectories mentioned in previous literatures. We would interchangeably use these two terms hereinafter.*

The goal of our paper is to predict user's next visited location in IoT environment via a session-based manner. Formally, given the records of users' traits in the current session s, including received signal strength sequence RSS_s and visited location sequence L_s, the task is to predict where a user will go next within this session.

4 Proposed Model

As discussed previously, both the trajectories and signal strength may indicate user's intention of future movement, and thus could be utilized for revealing user movement patterns. As such, the general idea of our proposed model is to effectively learn movement patterns hidden inside trajectories and signal strength simultaneously. Specifically, concerning trajectory modeling, previous methods [33] have suggested that location transition is important, however complex location transitions among sessions can not be well-identified and explicitly modeled by existing methods. Inspired by the recent advance of graph neural networks (GNNs) in session-based recommendation [7], we innovatively adapt GNNs in modeling sessions of trajectories, since GNNs are well-suited for modeling graph structured location transitions of trajectories with rich location connections considered. As for signal representation, Gated Recurrent Unit (GRU), a more elaborate model of RNN unit, is employed due to its great success in modeling variable-length sequence data. Finally, we build a joint session embedding from the two low-dimensional representations of trajectories and signal strength, which is optimized under the supervision of cross-entropy loss. We show the network architecture of

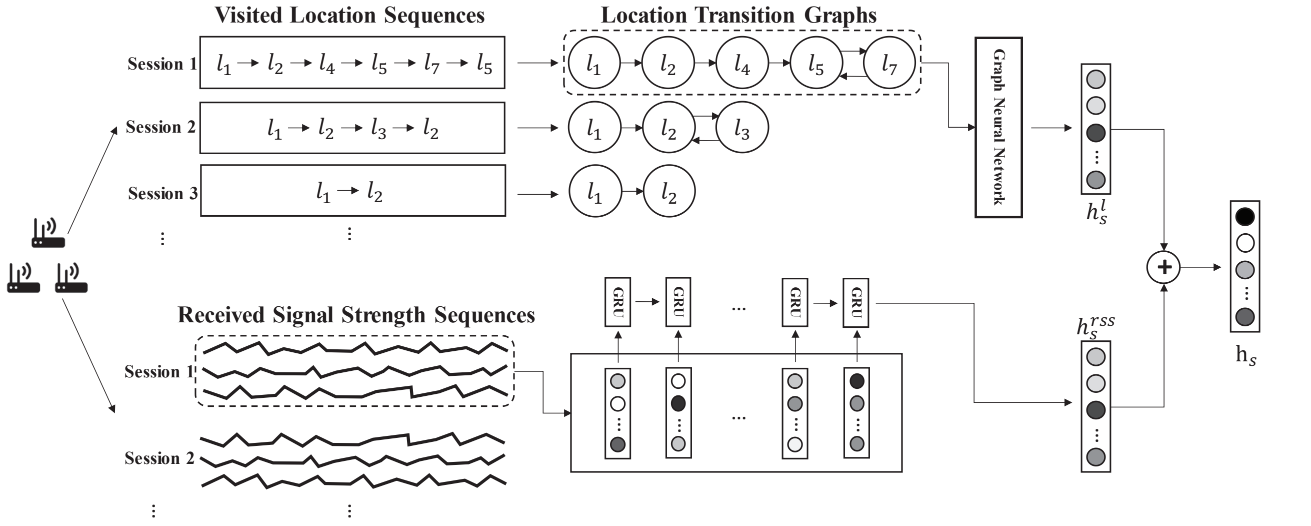

Visited Location Sequences
Location Transition Graphs
Session 1
$l_1 \rightarrow l_2 \rightarrow l_4 \rightarrow l_5 \rightarrow l_7 \rightarrow l_5$
Session 2
$l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_2$
Session 3
$l_1 \rightarrow l_2$
l_1
l_2
l_3
l_4
l_5
l_7
Graph Neural Network
h_s^l
Received Signal Strength Sequences
Session 1
Session 2
GRU
GRU
GRU
GRU
h_s^{rss}
h_s

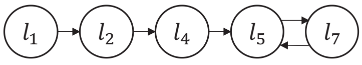
Fig. 2 The network architecture of our proposed TSIS model. At the bottom, the received signal strength sequence is encoded with GRU units (h_s^{rss}). At the top, the visited location sequence is encoded via gated GNNs (h_s^l). A joint session embedding is then learned under the supervision of the cross-entropy loss.

TSIS in Figure 2 and introduce its key components in the following.

4.1 Visited Location Sequence Representation

We view the visited location sequence of each session s as a directed subgraph $\mathcal{G}_s = (\mathcal{V}_s, \varepsilon_s)$, where node corresponds to an visited location $l_{si} \in V$ and directed edge $(l_{s,i-1}, l_{si}) \in \varepsilon_s$ indicates that location l_{si} is visited after $l_{s,i-1}$ in session s.

To consider the location transitions in session subgraphs, we first construct a location transition graph for each session, which is denoted as $\hat{\mathcal{G}}_s = (\hat{\mathcal{V}}_s, \hat{\varepsilon}_s)$. In the location transition graph $\hat{\mathcal{V}}_s$, each node represents a unique location and each edge indicates that there exists a transition between the two locations.



	Outgoing edges					Incoming edges				
	1	2	4	5	7	1	2	4	5	7
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	1	0	0	0	0
4	0	0	0	1	0	0	1	0	0	0
5	0	0	0	0	1	0	0	1	0	1
7	0	0	0	1	0	0	0	0	1	0

Fig. 3 An example of trajectory session subgraph and its corresponding adjacency matrix $\mathbf{A}_s$.

To explicitly capture transition regularities from trajectory session subgraphs, we employ gated Graph Neural Networks (gated GNNs) [8], adaptation of GNNs [28], on the built location transition graphs. As variant of standard GNNs, gated GNNs use Gated Recurrent Units [9] for propagating node features. Firstly, for each session s, a visited location l_{si} at time slot i can be represented as a v-dimensional one-hot vector, where non-zero entry denotes the index for the corresponding location in $V = \{l_1, l_2, ..., l_v\}$. An location embedding module is then applied to learn a $d \times v$ transformation matrix E_l, such that each location l_{si} can be embedded into a d-dimensional dense feature vector h^l_{si} according to the following equation,

$$h^l_{si} = E_l l_{si}. \tag{1}$$

We use Equation 1 as the initialized node vectors, *i.e.* ${h^l_{si}}^{(1)} = E_l l_{si}$. Subsequently, the information contained in the node vectors h^l_{si} is passed through the graph via the adjacency matrix A in an iterative manner as follows:

$$a_{s,:}^{(t)} = \mathbf{A}_s^\top \Big[{h^l_{s1}}^{(t-1)}, \ldots, {h^l_{s,|\mathcal{V}_s|}}^{(t-1)} \Big]^\top + b, \tag{2}$$

where the adjacency matrix $\mathbf{A}_s$ reveals how locations in $\mathcal{G}_s$ interact with each other based on the location transition graph $\hat{\mathcal{G}}_s$. As shown in Figure 3, the adjacency matrix $\mathbf{A}_s$ utilizes both outgoing edges and incoming edges for characterizing the network structure of location transitions. As such, the activations $a_{s,\cdot}^{(t)}$ at t-th iteration contain information from edges in both directions. The remaining updates are conducted in an GRU-like fashion,

$$z_{si}^{(t)} = \sigma(W_z a_{si}^{(t)} + U_z {h_{si}^l}^{(t-1)}), \tag{3}$$

$$r_{si}^{(t)} = \sigma(W_r a_{si}^{(t)} + U_r {h_{si}^l}^{(t-1)}), \tag{4}$$

$$\widetilde{{h_{si}^l}^{(t)}} = tanh(W_o a_{si}^{(t)} + U_o(r_{si}^{(t)} \odot {h_{si}^l}^{(t-1)})), \tag{5}$$

$${h_{si}^l}^{(t)} = (1 - z_{si}^{(t)}) \odot {h_{si}^l}^{(t-1)} + z_{si}^{(t)} \odot \widetilde{{h_{si}^l}^{(t)}}, \tag{6}$$

where z_{si} and r_{si} are the reset and update gates respectively.

After updating all nodes in session subgraphs until convergence, we obtain each node's final hidden vectors h_{si}^l. To represent each session s, we use the vectors of the last visited location, *i.e.*, h_{sn}^l, as the session embedding h_s^l from the perspective of trajectory, $h_s^l = h_{sn}^l$.

4.2 Received Signal Strength Sequence Representation

To perform next location prediction, it is essential to learn good representation for received signal strength sequence, which captures the temporal dynamics of user movement from the perspective of signal strength. For each session s, we aim to learn a non-linear mapping $\mathcal{F}$ from the variable-length multivariate time series RSS_s to a d-dimensional hidden feature vector $h_s^{rss} \in \mathbb{R}^d$,

$$h_s^{rss} = \mathcal{F}(\mathsf{RSS}_s). \tag{7}$$

Due to the widely use and success of RNN-based models in modeling variable-length sequence data for Natural Language Processing (NLP) [37–39], we employ RNN-based models for modeling temporal dynamics in received signal strength sequence. Standard RNNs update their hidden state h using the following update function,

$$h_{si}^{rss} = g(W\mathsf{RSS}_{si} + Uh_{s,i-1}^{rss}), \tag{8}$$

where g is the activation function, usually chosen as a logistic sigmoid function and RSS_{si} is the input at time slot i for session s. However, standard RNNs suffer from the vanishing gradient problem. As such, we employ a Gated Recurrent Unit (GRU) [9], a more elaborate model of an RNN unit, as $\mathcal{F}$ to deal with this problem and better capture long-term temporal dynamics. GRU gates essentially learn when and by how much to update the hidden state of the unit. The activation of the GRU is a linear interpolation between the previous activation $h_{s,i-1}^{rss}$ and the candidate activation $\hat{h}_{si}^{rss}$,

$$h_{si}^{rss} = (1 - z_{si})h_{s,i-1}^{rss} + z_{si}\hat{h}_{si}^{rss}. \tag{9}$$

The update gate z_{si} is calculated with

$$z_{si} = \sigma(W_z\mathsf{RSS}_{si} + U_z h_{s,i-1}^{rss}). \tag{10}$$

The candidate activation function $\hat{h}_{si}^{rss}$ is computed in a similar manner,

$$\hat{h}_{si}^{rss} = tanh(W\mathsf{RSS}_{si} + U(r_{si} \odot h_{s,i-1}^{rss})). \tag{11}$$

Finaly the reset gate r_{si} is given by:

$$r_{si} = \sigma(W_r\mathsf{RSS}_{si} + U_r h_{s,i-1}^{rss}). \tag{12}$$

Note that σ and $\odot$ are a logistic sigmoid function and an element-wise multiplication operator, *i.e.*, Hadamard product, respectively.

As shown in Figure 2, the last hidden state of GRU units, *i.e.*, h_{sn}^{rss}, is used as the representation for the received signal strength sequence of session s, since it encodes the temporal dynamic information of the entire sequence.

4.3 Joint Session Embedding

Given the representation for a received signal strength sequence h_s^{rss} as well as the representation for the visited location sequence h_s^l, we concatenate them together as $[h_s^{rss}; h_s^l] \in \mathbb{R}^{2d}$. We aim to learn a joint session embedding via a mapping function $\mathcal{H}$, such that the $2d$-dimensional input can be compressed into a d-dimensional vector h_s. In this work, we take linear transformation $W_j \in \mathbb{R}^{d\times 2d}$ as the mapping function $\mathcal{H}$ and calculate the joint session embedding vector h_s as,

$$h_s = W_j[h_s^{rss}; h_s^l]. \tag{13}$$

With the joint session embedding vector h_s and the location embedding matrix E_l, we obtain the v-dimensional predicted score vector o_s as follows,

$$o_s = E_l^\top h_s. \tag{14}$$

Then we derive the distribution over the v candidate locations with the softmax function,

$$\hat{l}_s = softmax(o_s), \tag{15}$$

Table 1 Statistics of the datasets.

Statistics	WTD	Store
# of visits	181,004	4,825,999
# of training sessions	12,581	757,160
# of test sessions	3,415	114,346
# of locations	220	200
average length per session	11.32	5.54
range of signal strength value	0–31	0–100

where $\hat{l}_s \in \mathbb{R}^v$ denotes the predicted probabilities of the last visited location within the session.

We use the cross-entropy of the prediction and the ground truth as the loss function. Given a training set with S sessions, the loss can be written as,

$$\mathcal{L} = -\sum_{s=1}^{S} l_{s,I_s} \log \hat{l}_s, \tag{16}$$

where I_s is the length of session s and l_{s,I_s} denotes the v-dimensional one-hot vector of the last visited location for session s. We use Stochastic Gradient Descent and the Back Propagation Through Time algorithm to learn the parameters of the model until convergence.

5 Experimental Results

In this section, we first introduce the datasets, baseline methods and evaluation metrics. Then we compare our proposed method with other state-of-the-art baselines for next location prediction on two real-world datasets. Moreover, by ablation experiment, we validate that both trajectories and signal strength are essential for next location prediction. The robustness of TSIS is also checked with we different parameter settings and under different session lengths. Finally, a case study of session clustering is conducted to show the effectiveness of TSIS in revealing human mobility patterns.

5.1 Datasets

We evaluate TSIS against popular baselines on two real-world datasets, of which the statistics are summarized in Table 1.

The first dataset is UCSD Wireless Topology Discovery Trace [1], which is publicly available. This dataset contains traces of 275 students with PDAs in the UCSD campus for an 11-week period. While the device of each student was powered on, the background data collection tool Wireless Topology Discovery, *i.e.*, WTD, collected every 20 seconds for all APs (access points) that it could sense across all frequencies. The transactional information of this trace data used in our experiment include unique device identifier, sampled time, unique identifier of the detected AP and strength of AP signals received by device. We remove the APs with low visited frequency in all the trace and finally obtain 220 APs. This dataset can be referred to as WTD hereinafter.

The second dataset is collected from one of the brick-and-mortar stores of a Chinese household electrical appliance retailers. 17 sensors were deployed in this store and the signal strength received by customer mobile devices were collected every 60 to 70 seconds. The time range of this dataset is 69 days, from October 18, 2017 to December 25, 2017. The information used in the experiments is similar to that of WTD dataset. We refer to this dataset as Store hereinafter.

For all these two datasets, we treat the unique device identifier as the unique identifier for each user. Generally, at each time slot, we use the nearest AP identifier indicated by signal strength as the corresponding location label. However, as for Store dataset, there are only 17 sensors in the raw data, which might be inadequate to assign each trace record a precise and fine-grained position. As such, we cluster the signal strength vectors of all trace records into 200 groups and use the cluster label as the location label of each record. By this means we could extract each user's moving activities, *i.e.*, sequences of signal strength and visited locations. Note that, the range of signal strength value for WTD is 0–31 while that for Store is 0–100. For the ease of training, we take a min-max normalization method to scale all signal strength values into [0, 1].

The next key aspect of data preprocessing is to divide each activity sequence into multiple sessions for each user. Revisit that the general logic of session splitting is that a new session starts when the time interval between the current and the last detected visit is greater than a predefined threshold denoted as η. For WTD dataset, we followed the authors that released this dataset [1] to use a simple heuristic for determining the value of η. That is, requiring $\eta = 1.5$ minutes between sessions in which session boundary signal strength was adequate ($> RSSI_{MAX}/3$), and $\eta = 4$ minutes between sessions where session boundary signal strength was low ($< RSSI_{MAX}/3$). This heuristic accounted for signal fluctuations detected by sensors. For Store dataset, we set η to 5 minutes as the default settings and also evaluate our proposed model under different values of η in Section 5.5 to study the effect of η to the prediction performance. After obtaining sessions, we filtered out all sessions of length 1.

For fair comparison, in each session, we use the last visited location as the next visited location, *i.e.*, the label of the session, while the previous signal strength and location records are treated as the input data. On Store dataset, the last session of each user is reserved for testing while the previous sessions of each user are for training. However, on WTD dataset, the number of users is quite small, thus we reserve the sessions of the last four weeks of each user as testing. We follow the fashion of session-based recommendation [25] and do not split the data mid-session, which means each session is assigned to either the training or the testing set.

5.2 Baseline Algorithms

To validate the effectiveness of our proposed method, we compare it with competitive baseline methods as follows:

- **POP**: POP ranks all locations by their popularity on the training set. It is a strong baseline despite its simplicity, thus we include it for comparison.
- **S-POP**: S-POP ranks locations according to their popularity in the current session. Ties are broken up using global popularity values.
- **MC**: The markov chain model is a classical model, which captures the consecutive sequential patterns.
- **FPMC** [30]: It is a sequential prediction method for next-basket recommendation.
- **PRME** [6]: PRME is a recent state-of-the-art model for next new POI recommendation.
- **NextItNet** [27]: NextItNet is a convolutional generative network based model for session-based next item recommendation. We use it as a baseline for next location prediction.
- **SR-GNN** [7]: SR-GNN is currently the state-of-the-art method for session-based recommendation. We use it as a competitive baseline for next location prediction.

5.3 Evaluation Metrics

Several popular ranking-based metrics were adopted for comparing next location prediction performance of different models.

- **Recall@k** is the proportion of sessions for correctly predicting next locations amongst the top-K predicted locations in all test sessions. Note that the rank of the groundtruth location is neglected by recall@k as long as the location is amongst the top-K predicted locations.
- **MRR@k** (Mean Reciprocal Rank) is the average reciprocal ranks of the groundtruth locations in the predicted list. The reciprocal rank is 0 if the rank exceeds k. With MRR, the order of locations is considered, where large MRR value indicates the correct prediction in the top of ranking list.

Note that we report Recall@k and MRR@k with k={5,10,15,20} in our experiments for comparison. For all these two metrics, the larger the value, the better the performance.

5.4 Parameter Setup & Model Optimization

For comparing next location prediction performance, we set the dimensionality of latent vectors $d = 100$ on all datasets. The sensitivity of TSIS to latent dimensionality d was given later in Section 5.7. Other hyper-parameters was selected on a validation dataset which is a random 10% subset of the training set. All parameters of the network in TSIS were initialized with a Gaussian distribution with a mean of 0 and a standard deviation 0.1. Adam optimizer was employed for parameter learning [40] and the learning rate was set to 0.01. Mini-batch training technique was adopted and the batch size was set to 1000. The max number of training epochs was 30 with early-stopping employed.

For representing received signal strength sequence, we use one layer of GRU. Other than GRU unit introduced in Section 4.2, we also briefly experimented with different network architectures, including (i) sequence to sequence autoencoder [38], (ii) using sequence to sequence autoencoder to first pretrain the parameters, and then applying GRU units to train the model [41, 42]. Empirically we found that our simple GRU based model achieves comparable performance with the above two methods, but with less computational costs. As such, we keep the simple GRU unit in our architecture.

We experimented with different loss functions in TSIS, including pointwise ranking based loss *cross-entropy* (*i.e.*, the one introduced in Section 4.3) and pairwise ranking based loss *Bayesian Personalized Ranking* [25, 43, 44] (BPR). We found BPR to perform worse than cross-entropy. Therefore, we use cross-entropy loss in our model.

The proposed TSIS model is implemented in PyTorch and all experiments were carried out on a Linux Server with Intel Xeon E5-2609 v4 @ 1.70GHz CPU, 126 GB memory and 4 GeForce GTX Titan 1080 Ti GPUs.

Table 2 Next location prediction performance on WTD dataset. The best results are in bold and the second best underlined.

Model	WTD							
	Recall@5	MRR@5	Recall@10	MRR@10	Recall@15	MRR@15	Recall@20	MRR@20
POP	0.2688	0.1102	0.5467	0.1450	0.7687	0.1653	0.8173	0.1659
S-POP	0.8073	0.6046	0.8673	0.6128	0.9280	0.6176	0.9353	0.6180
MC	0.7522	0.6787	0.8281	0.6887	0.9080	0.6951	0.9179	0.6957
FPMC	0.8492	0.6185	0.9042	0.6256	0.9236	0.6272	0.9455	0.6284
PRME	0.5979	0.2633	0.8287	0.2943	0.9309	0.3027	0.9558	0.3041
NextItNet	0.8764	0.7184	0.9290	0.7260	0.9442	0.7272	0.9525	0.7277
SR-GNN	0.9198	0.7649	0.9595	0.7706	0.9706	0.7715	0.9755	0.7717
TSIS	**0.9340**	**0.7913**	**0.9664**	**0.7959**	**0.9765**	**0.7967**	**0.9817**	**0.7970**

Table 3 Next location prediction performance on Store dataset. The best results are in bold and the second best underlined.

Model	Store							
	Recall@5	MRR@5	Recall@10	MRR@10	Recall@15	MRR@15	Recall@20	MRR@20
POP	0.6481	0.2747	0.6871	0.2799	0.8061	0.2886	0.8405	0.2907
S-POP	0.6592	0.3920	0.7304	0.4022	0.8170	0.4086	0.8556	0.4109
MC	0.6164	0.3373	0.6770	0.3457	0.7622	0.3520	0.8007	0.3542
FPMC	0.7357	0.4580	0.8441	0.4748	0.8944	0.4789	0.9206	0.4806
PRME	0.5959	0.4563	0.7139	0.4720	0.7795	0.4772	0.8252	0.4798
NextItNet	0.7254	0.4582	0.8338	0.4730	0.8861	0.4771	0.9152	0.4788
SR-GNN	0.7517	0.4942	0.8585	0.5088	0.9047	0.5124	0.9309	0.5140
TSIS	**0.7745**	**0.5129**	**0.8706**	**0.5259**	**0.9142**	**0.5293**	**0.9379**	**0.5307**

Table 4 Next location prediction performance on StoreRaw dataset. The best results are in bold and the second best underlined.

Model	StoreRaw							
	Recall@5	MRR@5	Recall@10	MRR@10	Recall@15	MRR@15	Recall@20	MRR@20
POP	0.6595	0.3832	0.7961	0.4031	0.9990	0.4190	1.0000	0.4191
S-POP	0.8110	0.5474	0.9094	0.5616	0.9994	0.5688	1.0000	0.5688
MC	0.8078	0.5584	0.9771	0.5813	0.9987	0.5830	1.0000	0.5831
FPMC	0.8864	0.6085	0.9911	0.6235	0.9996	0.6243	1.0000	0.6243
PRME	0.8322	0.6023	0.9865	0.6233	0.9995	0.6305	1.0000	0.6322
NextItNet	0.8870	0.6185	0.9872	0.6326	0.9995	0.6336	1.0000	0.6336
SR-GNN	0.8938	0.6323	0.9914	0.6460	0.9996	0.6468	1.0000	0.6469
TSIS	**0.9122**	**0.6577**	**0.9931**	**0.6690**	**0.9997**	**0.6696**	**1.0000**	**0.6696**

Table 5 Next location prediction performance on Store-3 dataset. The best results are in bold and the second best underlined.

Model	Store-3							
	Recall@5	MRR@5	Recall@10	MRR@10	Recall@15	MRR@15	Recall@20	MRR@20
POP	0.6762	0.2962	0.7076	0.3004	0.7456	0.3033	0.8250	0.3080
S-POP	0.6799	0.4123	0.7498	0.4222	0.7833	0.4248	0.8430	0.4283
MC	0.6318	0.3489	0.6910	0.3571	0.7297	0.3600	0.7912	0.3636
FPMC	0.7472	0.4795	0.8443	0.4921	0.8937	0.4961	0.9227	0.4977
PRME	0.5825	0.4710	0.6972	0.4868	0.7718	0.4927	0.8204	0.4955
NextItNet	0.7374	0.4737	0.8273	0.4858	0.8822	0.4901	0.9135	0.4919
SR-GNN	0.7649	0.5152	0.8540	0.5271	0.9050	0.5311	0.9296	0.5325
TSIS	**0.7815**	**0.5349**	**0.8712**	**0.5478**	**0.9144**	**0.5503**	**0.9382**	**0.5517**

5.5 Next Location Prediction Performance Comparison

The next location prediction performance in terms of 8 evaluation metrics on two datasets is shown in Table 2 and Table 3, respectively. Note that all the reported results are averaged over 10 times independent run. Also, we conduct T-test between TSIS and each baseline for statistical significance testing on each dataset. Since TSIS incorporates both trajectories and signal strength for revealing user mobility patterns, it is obvious to see that TSIS consistently outperforms other competitive baseline methods on both datasets in terms of all evaluation metrics. Moreover, the advantages of TSIS over other baselines are statistically significant (p-value < 0.001 by T-test) on both

Table 6 Next location prediction performance on Store-10 dataset. The best results are in bold and the second best underlined.

Model	Store-10							
	Recall@5	MRR@5	Recall@10	MRR@10	Recall@15	MRR@15	Recall@20	MRR@20
POP	0.6594	0.4414	0.7108	0.4478	0.7910	0.4548	0.8502	0.4583
S-POP	0.6791	0.4294	0.7483	0.4391	0.8132	0.4446	0.8630	0.4475
MC	0.6412	0.4321	0.6888	0.4384	0.7628	0.4447	0.8264	0.4483
FPMC	0.7435	0.4894	0.8526	0.5049	0.9035	0.5095	0.9303	0.5110
PRME	0.6018	0.4541	0.7143	0.4691	0.7791	0.4742	0.8275	0.4769
NextItNet	0.7421	0.4869	0.8506	0.5016	0.9012	0.5056	0.9288	0.5072
SR-GNN	0.7643	0.5190	0.8698	0.5337	0.9163	0.5374	0.9409	0.5388
TSIS	**0.7846**	**0.5371**	**0.8798**	**0.5500**	**0.9236**	**0.5535**	**0.9467**	**0.5548**

Table 7 Next location prediction performance on Store-15 dataset. The best results are in bold and the second best underlined.

Model	Store-15							
	Recall@5	MRR@5	Recall@10	MRR@10	Recall@15	MRR@15	Recall@20	MRR@20
POP	0.6075	0.3930	0.7405	0.4145	0.8224	0.4215	0.8631	0.4238
S-POP	0.6689	0.4187	0.7738	0.4347	0.8375	0.4399	0.8717	0.4418
MC	0.6234	0.4158	0.7099	0.4293	0.7850	0.4355	0.8367	0.4385
FPMC	0.7503	0.4705	0.8534	0.4841	0.8984	0.4875	0.9282	0.4892
PRME	0.5984	0.4409	0.7150	0.4567	0.7816	0.4620	0.8311	0.4647
NextItNet	0.7552	0.4783	0.8507	0.4912	0.8985	0.4950	0.9260	0.4966
SR-GNN	0.7733	0.5061	0.8687	0.5190	0.9133	0.5225	0.9400	0.5241
TSIS	**0.7884**	**0.5206**	**0.8778**	**0.5327**	**0.9205**	**0.5361**	**0.9441**	**0.5374**

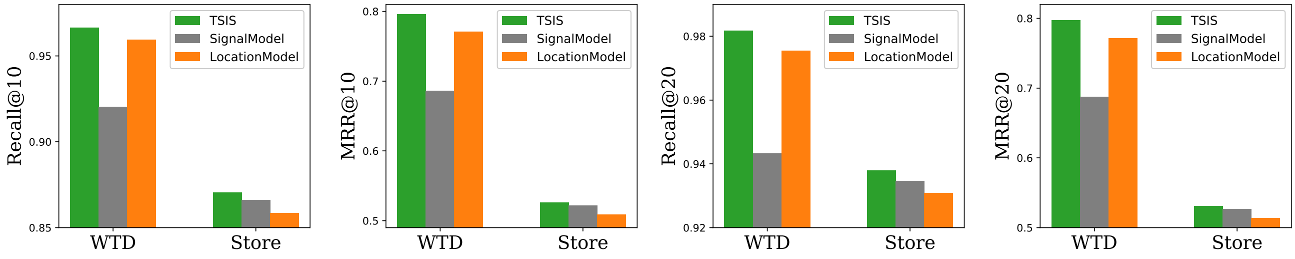


Fig. 4 Next location prediction performance of our proposed method compared with its two submodels.

Table 8 Prediction performance with different session lengths (MRR@20).

Model	WTD		Store	
	Short	Long	Short	Long
PRME	0.3788	0.3104	0.4263	0.4101
NextItNet	0.6820	**0.8343**	0.5210	0.3739
SR-GNN	0.7215	0.7438	0.5223	0.4148
TSIS	**0.7291**	0.7814	**0.5402**	**0.4233**

datasets. This guarantees the outperformance of our proposed method over baselines does not happen by chance. It is worthwhile to see that SR-GNN, which models the location transition patterns in separated sessions using graph neural networks, achieves second best prediction performance, which demonstrates the power of graph neural networks in session representations. By making predictions based on repetitive co-occurred locations in current sessions, S-POP makes great improvement compared with POP.

We also evaluate the next location prediction performance on the original coarse grained Store dataset with 17 unique locations, which we refer to as StoreRaw dataset. We can see from Table 4 that all methods show improved prediction performance than those on Store dataset in terms of all evaluation metrics and our proposed method still consistently outperforms other competitive baseline methods. Due to the imprecise positioning issue on StoreRaw dataset, we resort to Store instead of StoreRaw dataset for the following experiments.

Additionally, to further study the effect of η to the next location prediction performance, we varied η in {3, 10, 15} minutes on Store dataset, which generated three variants of Store dataset, denoted as Store-3, Store-10 and Store-15, respectively. We compare TSIS with all baselines on these datasets and show the results in Table 5, Table 6, Table 7, respectively. From the results, we can see that the prediction

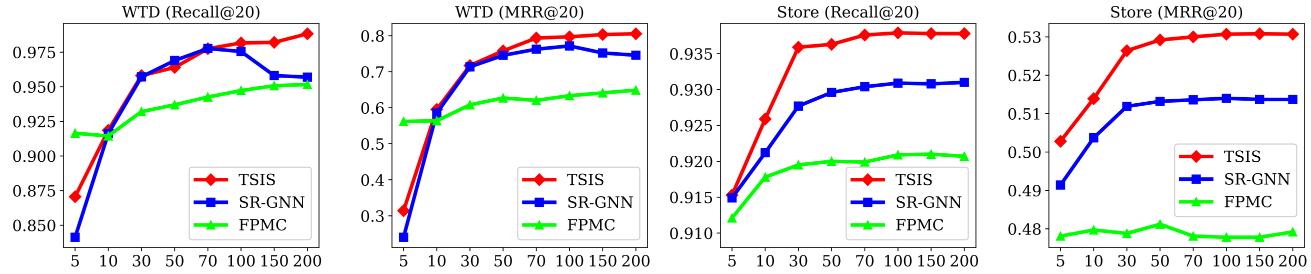


Fig. 5 Influence of latent dimensionality to the prediction performance of our proposed model compared to SR-GNN and FPMC on both datasets. Recall@20 or MRR@20 (y-axis) vs. the number of latent dimensions (x-axis).

performance of TSIS did not fluctuate too much and our model shows its consistent outperformance over all baselines under different settings of η. This also validates our proposed model is robust.

5.6 Ablation Experiment

To study the contribution of different information to TSIS, we eliminate the impact of signal strength and visited location sequence from TSIS, respectively. As such two submodels are introduced, which differ in the manner of session representation with our proposed model, but share the same output layer.

- **SignalModel** uses GRU to encode only the signal strength sequence of all sessions. Then the last hidden vector of each session represents the session embedding.
- **LocationModel** uses gated GNNs [8] to directly model the location sequences and treat the last hidden vector of each session as the representation vector.

The prediction performance of TSIS and its two submodels in terms of Recall@10, MRR@10, Recall@20, MRR@20 are shown in Figure 4. As for all four evaluation metrics, it is obvious to see that the predictive performance on both datasets degrades when the impact of either signal strength or visited locations are eliminated from the model, which indicates that signal strength and visited locations are both indispensible for next location prediction. It is also interesting to see that on Store dataset the signal strength is more important than trajectory whereas on WTD dataset trajectory is more important. This might be owing to that the outdoor WTD dataset allows for larger space, which brings more noisy information to signal strength.

5.7 Analysis of Latent Dimensionality and Session Lengths

We analyze the influence of latent dimensionality to the prediction performance of TSIS compared to SR-GNN and FPMC on both datasets. Figure 5 shows the Recall@20 and MRR@20 for various latent dimensionality while keeping other optimal hyperparameters unchanged. It is clear that in terms of all evaluation metrics on both datasets the performance of our proposed model first improves with increasing number of latent dimensions, then reaches to a stable level with further increasing number of latent dimensions. It can also be seen that when the number of latent dimensions is 100 our proposed model achieves reasonable performance. This validates the settings of number of latent dimensions in performance comparison experiments. Moreover, even not with the best dimensionality, our proposed model still outperforms other compared methods in a large range. We can say that our proposed model is not very sensitive to the number of latent dimensions and can be well employed for practical applications.

Further, we evaluate the prediction performance of TSIS and three baselines in dealing with different lengths of sessions. Specifically, we group each of the WTD and Store datasets into two categories, *i.e.*, Short and Long. Short means that the length of sessions is less than or equal to 5 while Long means that the length of sessions is more than 5. The percentages of session belonging to short group and long group are 0.57 and 0.43 on WTD dataset while the percentages on Store dataset are 0.79 and 0.21. We show the prediction performance evaluated by MRR@20 in Table 8. We can see that our proposed method can consistently outperform the competitive baseline on two datasets with different session lengths except for the case on WTD Long. This validates the robustness of our proposed method.

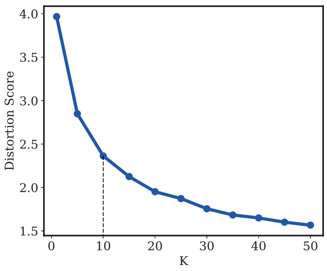


Fig. 6 Elbow method for choosing number of clusters K.

5.8 Case Study on Session Clustering

In order to show the interpretability of our proposed model, we finally perform session clustering on StoreRaw dataset. All the parameters used in this experiment are the same as the default settings used in previous comparison experiments. Specifically, for clustering sessions on StoreRaw dataset, we first utilize our framework to generate the joint embedding of each session, which is a 100 dimensional vector. Then K-means is employed to generate clusters on these session embeddings.

As the number of clusters K needs to be initially specified but remains unknown in prior, we use the popular Elbow method to select the optimal number of clusters by fitting the model with a range of values for K. As in this case, we fit the K-means model with the values of K varying from 1 to 50 with step-size 5. For each K, we compute the sum of squared distances from each point to its assigned center, denoted as distortion score, as shown in Figure 6. When the model is fitted with 10 clusters, we can see an “elbow” in the graph, which in this case is assumed to be the optimal number of clusters.

We choose to set K to 10 and extract 10 clusters for all sessions on StoreRaw dataset. To better understand the moving behaviors for each cluster of sessions, we further conduct FP-Growth analysis on the raw trajectories within each cluster to extract most frequent moving patterns to provide explanations for the cluster. For each cluster, we plot the session embeddings using t-SNE as well as the most two representative patterns of each cluster in Figure 7. We can see that in some clusters, *e.g.*, Cluster 1 and 7, trajectories are distributed only in one floor, while for others, *e.g*, Cluster 3, 4, 5 and 8, trajectories are distributed across different floors, which also involves transitions between different floors. Among cross-floor trajectories, Cluster 4, 5, and 8 contain round trips between two floors. For example, Cluster 4 shows a “floor 1 → floor 2 → floor 1 → floor 2” moving path, whereas Cluster 8 shows a “floor 1 → floor 2 → floor 1” moving path. This experiment validates that the embeddings generated by our framework can be utilized to capture the movement patterns of users and the trajectories in each cluster are meaningful. This might also explain in partial that why TSIS could beat some state-of-the-art baselines for next location prediction.

Moreover, we plot the transition patterns with respect to each cluster in Figure 8, where the entry (i, j) in the transition matrix represents the observed frequency of current location i transiting to location j. Note that the transition matrices in Figure 8 are calculated over the Top-200 frequent patterns mined by FP-Growth for each cluster. Obviously, the transition patterns of different clusters are also distinct. For example, in Cluster 2 little probability mass lies in transition across different floors while the other clusters contain high probability of cross floor transition. This might be owing to the explicit modeling of location transitions in our proposed model.

6 Conclusion

In this paper, we investigated the problem of next location prediction in IoT environment. We proposed a TSIS model, which learned good representation of each session by jointly modeling transition regularities in trajectories and temporal dynamics in signal strength. Extensive experiments on two real-world Wi-Fi based mobility datasets demonstrated that TSIS consistently outperformed competitive baseline methods for next location prediction. Especially, by ablation study, we empirically validated that both trajectories and signal strength were essential to the success of accurate prediction. In the future, we plan to extend our model by incorporating auxiliary information such as position and distance between locations to further enhance the next location prediction performance.

Acknowledgements Dr. Guannan Liu was supported by National Natural Science Foundation of China (NSFC) under the grant number 71701007 and 71531001.

References

1. McNett M, Voelker G M. Access and mobility of wireless pda users. ACM SIGMOBILE Mobile Computing and Communications Review, 2005, 9(2): 40–55
2. Leu J S, Yu M C, Tzeng H J. Improving indoor positioning precision by using received signal strength fingerprint and footprint based

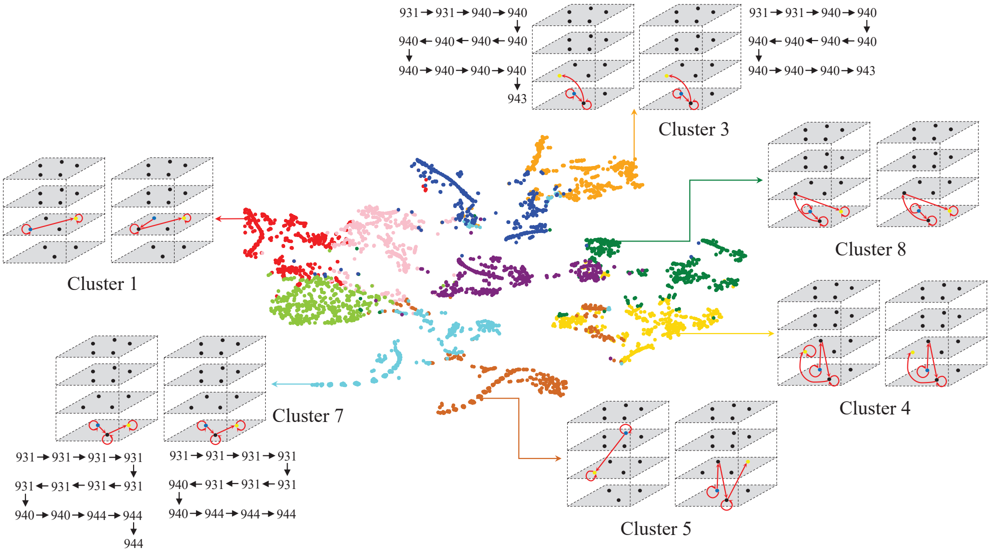


Fig. 7 Visualization of the embedding of 10 session clusters. For each cluster, the top 2 representative trajectories mined by FP-Growth are also illustrated. For Cluster 3 and 7, examples of visited sequences of the two representative trajectories are also shown. For each trajectory, the blue points denote the start and the yellow points denote the end.

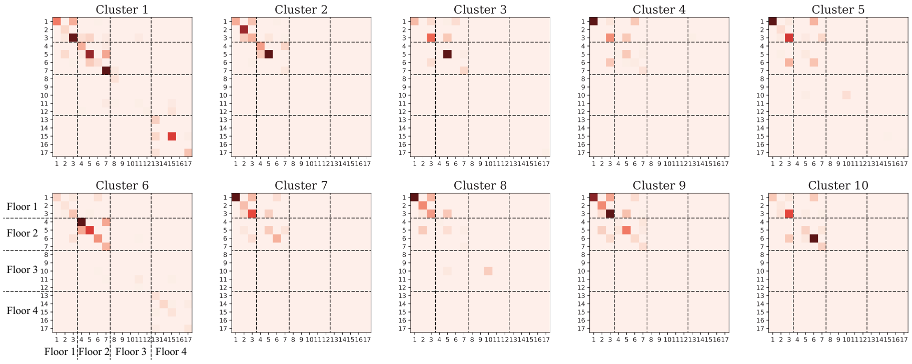


Fig. 8 Transition matrix for each cluster. The y-axis represents the location that user transits from, and the x-axis is the location that user transits to.

on weighted ambient wi-fi signals. Computer Networks, 2015, 91(C): 329–340

3. Li D, Balaji B, Jiang Y, Singh K. A wi-fi based occupancy sensing approach to smart energy in commercial office buildings. In: Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings. 2012, 197–198

4. Yao D, Zhang C, Huang J, Bi J. Serm: A recurrent model for next location prediction in semantic trajectories. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 2017, 2411–2414

5. Feng J, Li Y, Zhang C, Sun F, Meng F, Guo A, Jin D. Deepmove: Predicting human mobility with attentional recurrent networks. In: Proceedings of the 2018 World Wide Web Conference. 2018, 1459–1468

6. Feng S, Li X, Zeng Y, Cong G, Chee Y M, Yuan Q. Personalized ranking metric embedding for next new poi recommendation. In: Proceedings of the 24th International Conference on Artificial Intelligence. 2015, 2069–2075

7. Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T. Session-based recommendation with graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 346–353

8. Li Y, Tarlow D, Brockschmidt M, Zemel R S. Gated graph sequence neural networks. In: 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016

9. Cho K, Merrienboer v B, Bahdanau D, Bengio Y. On the properties of neural machine translation: encoder-decoder approaches. CoRR, 2014, abs/1409.1259

10. Feng C, Au W S A, Valaee S, Tan Z. Received-signal-strength-based indoor positioning using compressive sensing. IEEE Transactions on Mobile Computing, 2012, 11(12): 1983–1993

11. Zhu X, Feng Y. Rssi-based algorithm for indoor localization. Communications and Network, 2013, 5(02): 37

12. He S, Chan S G. Wi-fi fingerprint-based indoor positioning: recent advances and comparisons. IEEE Communications Surveys Tutorials,

2016, 18(1): 466–490

13. Liu Y, Yang Z. Location, Localization, and Localizability: Location-awareness Technology for Wireless Networks. Springer Publishing Company, Incorporated, 2014
14. Gentile C, Alsindi N, Raulefs R, Teolis C. Geolocation Techniques: Principles and Applications. Springer Publishing Company, Incorporated, 2012
15. Wu C, Yang Z, Liu Y, Xi W. Will: wireless indoor localization without site survey. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(4): 839–848
16. Liu H, Gan Y, Yang J, Sidhom S, Wang Y, Chen Y, Ye F. Push the limit of wifi based localization for smartphones. In: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking. 2012, 305–316
17. Jiang Y, Pan X, Li K, Lv Q, Dick R P, Hannigan M, Shang L. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. 2012, 441–450
18. Bahl P, Padmanabhan V N. Radar: an in-building rf-based user location and tracking system. In: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064). March 2000, 775–784
19. Farshad A, Li J, Marina M K, Garcia F J. A microscopic look at wifi fingerprinting for indoor mobile phone localization in diverse environments. In: International Conference on Indoor Positioning and Indoor Navigation. 2013, 1–10
20. Li X, Zhang D, Xiong J, Zhang Y, Li S, Wang Y, Mei H. Training-free human vitality monitoring using commodity wi-fi devices. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2018, 2(3): 121:1–121:25
21. Sapiezynski P, Stopczynski A, Wind D K, Leskovec J, Lehmann S. Inferring person-to-person proximity using wifi signals. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2017, 1(2): 24:1–24:20
22. Zhang J, Tang Z, Li M, Fang D, Nurmi P, Wang Z. Crosssense: Towards cross-site and large-scale wifi sensing. In: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking. 2018, 305–320
23. Guo X, Liu B, Shi C, Liu H, Chen Y, Chuah M C. Wifi-enabled smart human dynamics monitoring. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems. 2017, 16:1–16:13
24. Kim S, Lee J G. Utilizing in-store sensors for revisit prediction. In: 2018 IEEE International Conference on Data Mining (ICDM). 2018, 217–226
25. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based recommendations with recurrent neural networks. In: 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016
26. Jannach D, Ludewig M. When recurrent neural networks meet the neighborhood for session-based recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems. 2017, 306–310
27. Yuan F, Karatzoglou A, Arapakis I, Jose J M, He X. A simple convolutional generative network for next item recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. 2019, 582–590
28. Scarselli F, Gori M, Tsoi A C, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE Transactions on Neural Networks, 2009, 20(1): 61–80
29. Duong-Trung N, Schilling N, Schmidt-Thieme L. Near real-time geolocation prediction in twitter streams via matrix factorization based regression. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. 2016, 1973–1976
30. Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web. 2010, 811–820
31. Mathew W, Raposo R, Martins B. Predicting future locations with hidden markov models. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. 2012, 911–918
32. Cho E, Myers S A, Leskovec J. 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. 2011, 1082–1090
33. Liu Q, Wu S, Wang L, Tan T. Predicting the next location: A recurrent model with spatial and temporal contexts. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. 2016, 194–200
34. Feng S, Cong G, An B, Chee Y M. Poi2vec: Geographical latent representation for predicting future visitors. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. 2017, 102–108
35. Zhao P, Xu X, Liu Y, Zhou Z, Zheng K, Sheng V S, Xiong H. Exploiting hierarchical structures for poi recommendation. In: 2017 IEEE International Conference on Data Mining (ICDM). Nov 2017, 655–664
36. Zhao P, Zhu H, Liu Y, Xu J, Li Z, Zhuang F, Sheng V S, Zhou X. Where to go next: a spatio-temporal gated network for next poi recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 5877–5884
37. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015
38. Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. 2014, 3104–3112
39. Mikolov T, Karafiát M, Burget L, Černockỳ J, Khudanpur S. Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association. 2010
40. Kingma D P, Ba J. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015

41. Dai A M, Le Q V. Semi-supervised sequence learning. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2. 2015, 3079–3087
42. Ramachandran P, Liu P J, Le Q V. Unsupervised pretraining for sequence to sequence learning. arXiv preprint arXiv:1611.02683, 2016
43. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. 2009, 452–461
44. Hidasi B, Karatzoglou A. Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018, 843–852

Hao Lin received the Bachelor's degree in management information systems from Beihang University in 2013. He is currently working toward the Ph.D. degree in the School of Economics and Management at Beihang University, China. His research interests generally lie in the areas of data mining and machine learning, with special interests in user modeling and heterogeneous data fusion.

Guannan Liu is currently an Assistant Professor in the Department of Information Systems with Beihang University, Beijing, China. He received the Ph.D. degree from Tsinghua University, China. His research interests include data mining, social networks, and business intelligence. His work has been published in the journal of IEEE TKDE, ACM TKDD, ACM TIST, Decision Support Systems, Neurocomputing, etc, and also in the conference proceedings such as KDD, ICDM, SDM etc.

Fengzhi Li received his Bachelor's degree in applied physics from Beihang University, Beijing, China, in 2018. He is currently a MS candidate in the School of Economics and Management at Beihang University. His research interests include information extraction, topic model and multi-label learning.

Yuan Zuo received his Ph.D. degree from Beihang University, Beijing, China, in 2017. He is currently a postdoctor in Information Systems Department of Beihang University. His research interests include topic modeling and social computing.