International Journal of
Geo-Information

MDPI

Article

Indoor Location Prediction Method for Shopping Malls Based on Location Sequence Similarity

Peixiao Wang [1], **Sheng Wu** [1], **Hengcai Zhang** [2,3,*] and **Feng Lu** [2,3]

[1] The Academy of Digital China, Fuzhou University, Fuzhou 350002, China; peixiao_wang@163.com (P.W.); ws0110@163.com (S.W.)

[2] State Key Laboratory of Resources and Environmental Information System, IGSNRR, CAS, Beijing 100101, China; luf@lreis.ac.cn

[3] Fujian Collaborative Innovation Center for Big Data Applications in Governments, Fuzhou 350002, China

* Correspondence: zhanghc@lreis.ac.cn

Received: 7 September 2019; Accepted: 13 November 2019; Published: 15 November 2019

check for updates

Abstract: Fast and accurate indoor location prediction plays an important part in indoor location services. This work proposes an indoor location prediction framework named Indoor-WhereNext. First, a novel algorithm, "indoor spatiotemporal density-based spatial clustering of applications with noise" (Indoor-STDBSCAN), is proposed to detect the stay points in an indoor trajectory and convert them into a location sequence. Then, a spatial-semantic similarity (SSS) method for measuring the similarity between location sequences is defined. SSS comprehensively considers the spatial and semantic similarities between location sequences. Finally, a clustering algorithm is used to obtain similarity user groups based on SSS. These groups are used to train different prediction models to achieve improved results. Extensive experiments were conducted using real indoor Wi-Fi positioning datasets collected in a shopping mall. The results show that the Indoor-WhereNext model markedly outperforms the three existing baseline methods in terms of prediction accuracy and precision.

Keywords: indoor location prediction; sequence similarity; similar user clustering; indoor movement trajectory

1. Introduction

In recent years, with the rapid development of e-commerce, traditional "brick-and-mortar" industries have been severely affected [1,2]. These industries urgently need to develop ways to help merchants establish relations with customers and provide them with a personalized offline shopping experience in order to improve its marketing ability [3]. With the continuous development of indoor positioning technology and the popularization of mobile terminal equipment, indoor mobile user trajectory data have shown explosive growth [4–6]. Indoor trajectory data are an important basis for indoor location-based services and provide new opportunities for the development of "brick-and-mortar" industries [7–10].

Location prediction technology can infer the next location of a user according to the historical trajectory. It can thus provide flexible services for users, which has been a research concern in this field. Research shows that user behavior is predictable [11]. To date, location prediction technology has been widely used in trajectory reconstruction [12,13], location recommendation [14–16], intelligent transportation [17,18], and provision of security services [19]. Indoor location services, for example, can predict the next location of a user and push information about shops of interest to the user. This not only provides the user with a personalized shopping experience but also aids the merchant in earning profits [14–16].

Location prediction methods can be classified according to forecasting needs into the following two categories [20]: (1) those that predict the location that the user will visit next [10,21–25] and (2) those that predict the location of the user in the next time interval [26]. The main difference is that the former transforms an individual trajectory into a location sequence, while the latter treats it as a time interval sequence with relevant positions. In this study, only the first type of location prediction was considered. Existing research mainly uses data mining algorithms, such as the association rule [27,28], hidden Markov model (HMM) [29], or recurrent neural network (RNN) [30,31], to mine patterns between location sequences and then serve location predictions. In contrast with the existing research, the approach in this work focuses mainly on location prediction for indoor spaces, serving the location services of the offline industry, such as large shopping centers. Unlike outdoor location prediction, an indoor trajectory typically has three-dimensional features, which makes it difficult for existing stay point recognition algorithms to convert an indoor trajectory into a location sequence. Second, the user trajectory implies the user preference [32]. When there is a large number of users, it is easy to find similarity groups. It is easier to mine location patterns within similarity group [29].

Therefore, this paper proposes an indoor location prediction framework, called Indoor-WhereNext. The work makes a number of significant contributions, which are summarized as follows.

(1) A novel spatial-semantic similarity (SSS) method is defined. It combines spatial and semantic information to calculate the similarity between location sequences and find similarity groups of indoor users.
(2) Long short-term memory (LSTM) is used to model each group of users to improve the accuracy of indoor location prediction.
(3) The performance of the Indoor-WhereNext is evaluated using real indoor trajectories. The results demonstrate the advantages of our approach compared with baselines.

The rest of this work is organized as follows. In Section 2, the current literature focusing on models for location prediction from trajectories is reviewed. In Section 3, a new methodological framework for indoor location prediction is proposed. The performance of the method proposed in the current work is compared with that of methods proposed in previous research using real indoor Wi-Fi positioning data; these results are presented in Section 4. In Section 5, the work is summarized, and suggestions are made for possible future studies.

2. Related Work

Existing location prediction models that predict where users will visit can be roughly divided into two types: Individual-based and group-based prediction models.

Individual-based prediction models consider the movement behavior of each individual to be independent and, thus, use only the movement history of the user to predict her or his next location. The core issue of individual-based models is that they are mainly used to mine the periodic behavior of individual users. For instance, Lee et al. [22] proposed a spatiotemporal-periodic (STP) pattern to capture the periodic behavior of the individual. It used an association rule algorithm to mine periodic patterns in STP. Vu et al. [33,34] presented a novel framework, Jyotish, to find the periodic movement of people based on Wi-Fi/Bluetooth positioning data. Bayesian classifiers and support vector machines were used to give the most likely next location. Do et al. [35] redefined the location prediction problem from a new perspective and proposed a probabilistic kernel method for learning the dependence between user location and multivariate context variables from sparse data. Wu et al. [36] proposed a spatial-temporal-semantic neural network algorithm (STS-LSTM) for location prediction. Zhang et al. [17] combined the respective superiorities of support vector regression and deep learning to present a novel data embedding and ensemble learning method. Yang et al. [37] defined a novel Markov chain via Markov transition matrix multiplication and proposed the DestPD model. However, there are certain deficiencies in the individual-based models. First, these models require long-term movement trajectories of individual users, and these are difficult to obtain in practical applications. Second, individual-based models require an independent model to be built for each individual, which is also unrealistic in practical applications.

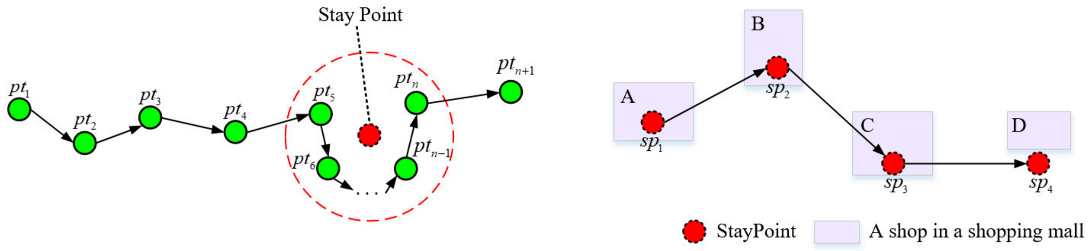
Group-based prediction models consider movement behavior to "follow the crowd" to some degree and, thus, use movement history of other users to predict a user's next location. These models are mainly used to mine similarity behaviors of groups of users. For example, Morzy [28] proposed an improved Apriori algorithm that uses association rules to predict the next location of a group of users. Ang et al. [38] utilized a Markov chain to convert location sequences into conversion probabilities for location prediction. Qiang et al. [30] proposed spatiotemporal RNN (ST-RNN) based on RNN [31] to model the location of groups of users. Ying et al. [23] proposed a geographic-temporal-semantic-based location prediction framework to predict the next location of a group of users. Unlike single-object models, group-based models can reveal the movement of a group of users in some scenarios [39]. In addition, group-based models do not need long-term movement trajectories of individual users. However, there are several shortcomings in the aforementioned group-based models. They build a model for all users, ignoring the existence of similarity subgroups. Therefore, some models were obtaining movement trajectories of only those who are in some way related to the user. Zhang et al. [40] found that there was a strong correlation between the calling patterns and co-cell patterns of users. Based on this finding, they proposed the NextCell model, which aims to enhance location prediction by harnessing the social interplay revealed in cellular call records. Wen et al. [41] proposed a fallback social-temporal-hierarchic Markov model (FSTHM), which introduced modified cross-sample entropy to quantify the similarities between the individual and his friends to enhance the predictive performance. Li et al. [42] used a linear regression model, which was constructed with a subset of users related to the predicted user, to predict the next location.

In this study, a novel indoor location prediction framework, Indoor-WhereNext, was developed. In the proposed framework, previously collected historical location sequences are first grouped according to their characteristics (i.e., according to the similarity of historical location sequences). Afterward, different user groups are used to train different prediction models to achieve improved results.

3. Methodology

Definition 1 (trajectory): A trajectory $traj = \{pt_i\}_{i=1}^{n}$ is an ordered sequence of points $pt_i = (id, t_i, x_i, y_i, f_i)$, where id is a unique user identifier; t_i is the time at which pt_i was collected; and x_i, y_i, f_i correspond to the longitude, latitude, and floor, respectively, of the user at time t_i.

Definition 2 (stay point): In general, a stay point $sp^{id} = (x, y, f, arrT, levT)$ stands for a geographic region where a user stayed over a certain time interval, where id is a unique user identifier; x, y, f correspond to the longitude, latitude, and floor, respectively, of the user's stay; and $(arrT, levT)$ represent the user's arrival and departure times, respectively, with respect to the geographic region. For example, the stay point of a user u shown in Figure 1a is expressed as $sp^u = \left(\sum_{i=5}^{n} pt_i.x/(n-4),\ \sum_{i=5}^{n} pt_i.y/(n-4),\ 3, pt_5.t,\ pt_n.t\right)$.



(a) (b)

Figure 1. Stay point and location sequence: (**a**) the movement of a user on the third floor and (**b**) the location sequence of a user in an indoor space.

Definition 3 (stay point sequence): A stay point sequence $S^{id} = \left\{sp_i^{id}\right\}_{i=1}^{k}$ is an ordered set of stay points detected in a user trajectory for $sp_i^{id}.levT < sp_{i+1}^{id}.arrT$, where id is a unique user identifier, and k represents the number of stay points in the user trajectory. For example, the stay point sequence of a user u shown in Figure 1b is represented as $S^u = \left\{sp_1^u, sp_2^u, sp_3^u, sp_4^u\right\}$.

Definition 4 (location sequence): A location sequence $locSeq^{id} = \left\{shop_i^{id}\right\}_{i=1}^{k}$ is an ordered sequence of locations visited by a user, where $shop_i^{id}$ represents the shop visited at the stay point sp_i^{id}. For example, the location sequence of user u shown in Figure 1b is represented as $locSeq^u = \{A, B, C, D\}$.

In this section, the Indoor-WhereNext framework for indoor location prediction is constructed. The overall architecture is shown in Figure 2. The framework is based on the bottom-up design principle and is divided into two modules: SSS-based location modeling and SSS-based location prediction. In the SSS-based location modeling phase, user trajectories are converted to location sequences via the Indoor-STDBSCAN algorithm. The user location sequences are clustered to obtain similarity user groups based on the SSS, and each group trains a model. An exemplar is also generated by each group to represent itself. In the SSS-based location prediction phase, the similarity matrix between the location sequence and each group is calculated by exemplars, and then different models are used to predict the next possible location according to the similarity matrix.

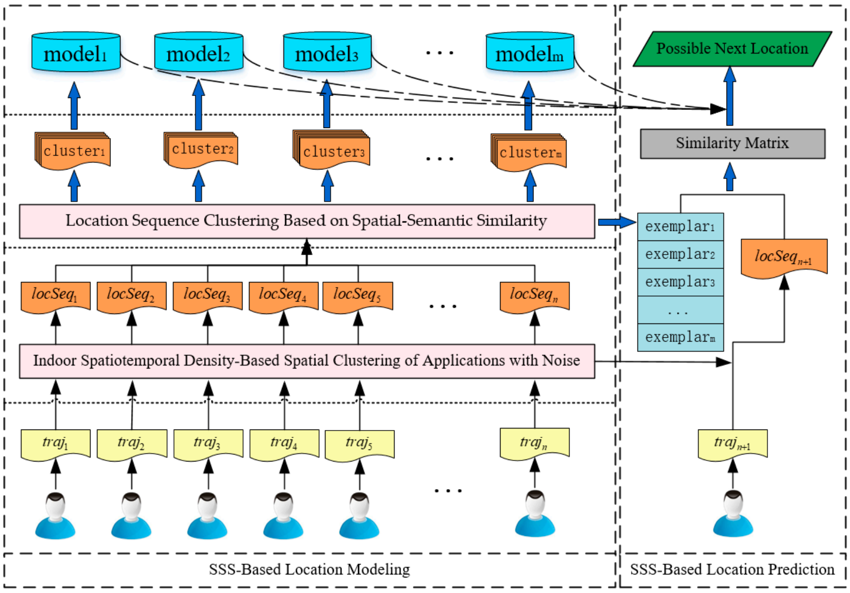


Figure 2. Flow chart of Indoor-WhereNext framework.

3.1. Location Sequence Detection Method

3.1.1. Stay Point Detection

When a user stays at a particular location, there is a greater probability that the user will view the location service information [43]. Therefore, stay point detection is a key step in location sequence conversion. When the user stays in a certain place for a certain length of time, the mobile terminal records more trajectory points in a limited area. This results in clustering of trajectory points. Therefore, a clustering algorithm is applied to detect the stay points. However, in contrast with an outdoor trajectory, an indoor trajectory typically has three-dimensional characteristics, which makes existing outdoor detection algorithms difficult to apply to the indoor trajectory. Therefore, a novel indoor trajectory stay point detection algorithm, Indoor-STDBSCAN, is proposed.

The Indoor-STDBSCAN algorithm is an improved version of the spatiotemporal density-based spatial clustering of applications with noise (ST-DBSCAN) algorithm [44,45], which aims to divide the individual user trajectory $traj = \{pt_i\}_{i=1}^{n}$ into k disjoint order clusters $\{C_1, C_2, \ldots, C_k\}$. Each cluster C_i

generates a stay point sp_i^{id}, and k clusters generate k stay points, i.e., $\left\{sp_i^{id}\right\}_{i=1}^{k}$. The Indoor-STDBSCAN algorithm makes two improvements to the ST-DBSCAN algorithm: (1) adding floor constraints, so that the trajectory point pt_j in the spatiotemporal neighborhood is on the same floor as pt_i, and (2) adding order constraints, so that the index of the last trajectory point of cluster C_i serves as the search benchmark of cluster C_{i+1}, ensuring that the stopping times of sp_{i+1}^{id} and sp_i^{id} do not overlap along the timeline—that is, $sp_i^{id}.\,levT < sp_{i+1}^{id}.arrT$. The overall process for Indoor-STDBSCAN is shown in Algorithm 1.

Algorithm 1. Indoor trajectory stay point detection algorithm.

```
Require: Individual trajectory: $traj = \{pt_i\}_{i=1}^{n}$
Radius: $\epsilon_1$
Time window: $\epsilon_2$
Neighborhood density threshold: $MinPts$
Ensure: Individual stay point sequence: $S^{id} = \left\{sp_i^{id}\right\}$
function Indoor-STDBSCAN ($traj,\ \epsilon_1,\ \epsilon_2,\ MinPts$)
    $clusterId = 0;\ start = 0$
    for next unprocessed $pt \in traj$ do
        if $pt.index < start$ then
            continue
        $N = getNeighbors\ (pt,\ \epsilon_1,\ \epsilon_2)$
        if $|N| > MinPts$ then
            $pt.processed = true; pt.clusterId = clusterId$
            $Seeds = []$
            $add\ unprocessed\ p \in N\ to\ Seeds$
            for next $q \in Seeds$ do
                $N\prime = getNeighbors(q,\ \epsilon_1,\ \epsilon_2)$
                $q.processed = true; q.clusterId = clusterId$
                if $|N\prime| > MinPts$ then
                    $add\ unprocessed\ p \in N\prime\ to\ Seeds$
        $start = traj.find(p.clusterId == clusterId)[-1].index$
        $clusterId = clusterId + 1$
    for next $i \in [0,\ 1,\ \ldots,\ clusterId - 1]$ do
        $ptArr\ =\ traj.find(p.clusterId == i)$
        $sp^{id}.arrT = ptArr[0].t; sp^{id}.levT = ptArr[len(ptArr) - 1].t$
        $sp^{id}.x, sp^{id}.y\ =\ computeMeanCoord\ (ptArr); sp^{id}.f = ptArr[0].f$
        $S^{id}.add\left(sp^{id}\right)$
    return $S^{id}$
```

3.1.2. Location Sequence Conversion

The stay point obtained by the Indoor-STDBSCAN algorithm only contains the spatial information and does not contain semantic information, so it is necessary to assign semantics to the stay point. For this, a matching method is defined. There are two spatial relationships between a stay point and a shop: Inside and outside the shop. For example, the stay point sequence of a user u shown in Figure 3 contains two stay points $S^u = \left\{sp_1^u, sp_2^u\right\}$: stay point sp_1^u inside the shop and stay point sp_2^u outside the shop. For an inside stay point such as sp_1^u, the intersection method is used to obtain the shop that user u visited at stay point sp_1^u—that is, user u at stay point sp_1^u visited shop B. For an outside stay point such as sp_2^u, the concentric circle tangent method is used to calculate the shop that user u visited at stay point sp_2^u. With sp_2^u as the center, the radius $r_i\ (i\ =\ 1,\ 2,\ 3,\ \ldots)$ draws a circle, and the first shop to be tangent to the circle is shop C, so user u stayed at stay point sp_2^u and visited shop C. When all of the stay points in S^u are matched to a shop, the location sequence of user u can be obtained, and the location sequence is represented as $locSeq^u = \{B, C\}$.

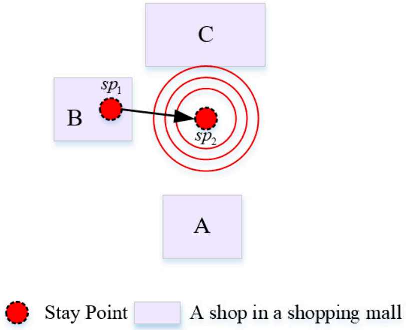


Figure 3. Stay point semantic matching.

3.2. Location Sequence Similarity Calculation Method

The core of the Indoor-WhereNext framework is to cluster the location sequences better. Users whose location sequences fall within the same group have high similarity and vice versa. Hence, the location patterns of users with location sequences falling within the same group are easier to mine. The Indoor-WhereNext framework achieves improved prediction accuracy by modeling similar users. The user location sequence implies the spatial information and semantic information of the shop. The spatial information of the shop describes the spatial location of the shop inside the shopping mall, which restricts the user's range of movement in the indoor space. The semantic information of the shop describes the semantic characteristics of the shop, which to some extent reflect the shopping habits of the user. Spatial information and semantic information are comprehensively considered to define a novel SSS method to measure the similarity between location sequences for cluster formation. The SSS method is divided into two parts: Spatial similarity and semantic similarity.

Spatial similarity mainly calculates the similarity of spatial information implied in the location sequence and describes the similarity of the movement trajectories of the two sequences in geospatial space. When users stay in the same shop, they show a certain degree of spatial similarity. The more shops there are between the location sequences, the higher the spatial similarity. Therefore, the longest common subsequence (LCSS) [46] is used to calculate the spatial similarity between location sequences. The spatial similarity between user u and user v is calculated as defined in Formulas (1) and (2).

$$LCSS\left(\left\{shop_i^u\right\}_{i=1}^{n},\left\{shop_j^v\right\}_{j=1}^{m}\right)=\begin{cases} 0 \;\; if\ n=0\ or\ m=0 \\ 1+LCSS\left(\left\{shop_i^u\right\}_{i=1}^{n-1},\left\{shop_j^v\right\}_{j=1}^{m-1}\right)\;\; if\ shop_n^u=shop_m^v \\ max\left(LCSS\left(\left\{shop_i^u\right\}_{i=1}^{n},\left\{shop_j^v\right\}_{j=1}^{m-1}\right),LCSS\left(\left\{shop_i^u\right\}_{i=1}^{n-1},\left\{shop_j^v\right\}_{j=1}^{m}\right)\right)\;\; otherwise \end{cases} \tag{1}$$

$$\boldsymbol{SM}_{uv}^{spa}=1-\frac{LCSS\left(\left\{shop_i^u\right\}_{i=1}^{n},\left\{shop_j^v\right\}_{j=1}^{m}\right)}{min(n,m)} \tag{2}$$

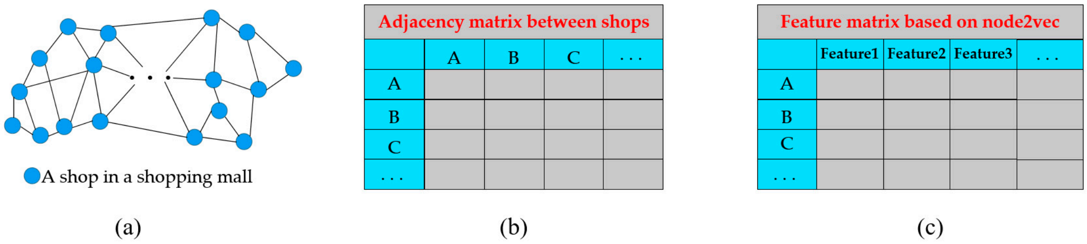
where $\left\{shop_i^u\right\}_{i=1}^{n}$ and $\left\{shop_j^v\right\}_{j=1}^{m}$ represent the location sequences of users u and v, respectively; n and m represent the numbers of shops visited by users; $max(x,y)$ is a function for obtaining the maximum of the values x and y; $\boldsymbol{SM}^{spa}$ is the spatial similarity matrix; and $\boldsymbol{SM}_{uv}^{spa}$ represents the spatial similarity between users u and v.

Semantic similarity mainly calculates the similarity of semantic information implicit in the location sequence and describes the degree of similarity between two users in interests and behaviors. In this paper, semantic information is not a categorical attribute of the shops, because we believe that their attribute information is artificially specified and subjective. The semantic information to which we refer is an implicit message that is expressed through user behavior. Generally, users hop more frequently between the same types of shop (purposeful consumption), which reflects the semantic similarity between those shops. In view of this, the location sequences of all users are constructed into a weighted network $G(V, E, W)$, where V represents the shop set, E represents the transfer set between shops, and W represents the transfer times between shops. With an increase in the number of location sequences, the weight between shops can reflect the similarity between them; that is, the higher the similarity, the higher the weight. Based on these characteristics, the node2vec [47] method is used to vectorize the shops. As shown in Figure 4, when the weight between shops is larger, the distance between the shops' corresponding vectors is less. After vectorization by node2vec, each shop uniquely corresponds to a vector, and the semantic similarity between location sequences can be calculated by the corresponding vector sequences. In this work, the dynamic time warping (DTW) [48–50] algorithm is used to calculate the semantic similarity between location sequences. The semantic similarity between user u and user v is calculated as defined in Formulas (3) and (4).

$$DTW\left(\left\{shop_i^u\right\}_{i=1}^{n},\left\{shop_j^v\right\}_{j=1}^{m}\right)= \begin{cases} 0 & if\ m=0\ n=0 \\ \infty & if\ m=0\ or\ n=0 \\ dish\Big(shop_n^u, shop_m^v + min\Big(DTW\Big(\left\{shop_i^u\right\}_{i=1}^{n-1},\left\{shop_j^v\right\}_{j=1}^{m-1}\Big), & \\ \qquad DTW\Big(\left\{shop_i^u\right\}_{i=1}^{n},\left\{shop_j^v\right\}_{j=1}^{m-1}\Big), & otherwise \\ \qquad DTW\Big(\left\{shop_i^u\right\}_{i=1}^{n-1},\left\{shop_j^v\right\}_{j=1}^{m}\Big)\Big) & \end{cases} \tag{3}$$

$$\boldsymbol{SM}_{uv}^{sem} = DTW\left(\left\{shop_i^u\right\}_{i=1}^{n},\left\{shop_j^v\right\}_{j=1}^{m}\right) \tag{4}$$

where $\left\{shop_i^u\right\}_{i=1}^{n}$ and $\left\{shop_j^v\right\}_{j=1}^{m}$ represent the location sequences of users u and v, respectively; n and m represent the numbers of shops visited by users; *dist* is used to calculate the Euclidean distance between the corresponding vectors of shops $shop_n^u$ and $shop_m^v$; $min(x, y, z)$ is a function for obtaining the minimum of the values x, y, and z; $\boldsymbol{SM}^{sem}$ is the semantic similarity matrix; and $\boldsymbol{SM}_{uv}^{sem}$ represents the semantic similarity between users u and v.



Adjacency matrix between shops

	A	B	C	. . .
A				
B				
C				
. . .				

Feature matrix based on node2vec

	Feature1	Feature2	Feature3	. . .
A				
B				
C				
. . .				

(a) (b) (c)

Figure 4. Process of turning shops into vectors: (**a**) weighted network formed by the transfer of shops, (**b**) adjacency matrix representation network, and (**c**) feature matrix encoded by node2vec.

After the semantic and spatial similarities between sequences have been calculated, the final location sequence similarity is superimposed by two parts, as defined in Formula (5).

$$\boldsymbol{SSM} = \alpha * \frac{\boldsymbol{SM}^{spa} - min(\boldsymbol{SM}^{spa})}{max(\boldsymbol{SM}^{spa}) - min(\boldsymbol{SM}^{spa})} + (1-\alpha) * \frac{\boldsymbol{SM}^{sem} - min(\boldsymbol{SM}^{sem})}{max(\boldsymbol{SM}^{sem}) - min(\boldsymbol{SM}^{sem})} \tag{5}$$

where $\boldsymbol{SM}^{spa}$, $\boldsymbol{SM}^{sem}$, and $\boldsymbol{SSM}$ represent the spatial similarity matrix, semantic similarity matrix, and spatial-semantic similarity matrix, respectively; $min(\boldsymbol{X})$ and $max(\boldsymbol{X})$ are functions for obtaining the minimum and maximum values, respectively, in matrix $\boldsymbol{X}$; and α is a weight coefficient that represents the contribution of spatial similarity to the location sequence similarity. The default value of α is 0.5—that is, the contributions of semantic similarity and spatial similarity to the location sequence similarity are equal.

3.3. Indoor User Location Prediction Framework

3.3.1. SSS-Based Location Modeling

After the SSS method has been defined, to divide users into different groups, several requirements should be considered. First, the number of groups cannot be known in advance. Second, each group needs to have a representative user. The representative user mainly helps new users know which model they use. Based on the above two points, the affinity propagation (AP) [51] algorithm is used to cluster the location sequences of all users. After clustering, LSTM [52] is used to train the prediction model for users in each group. The training process of the Indoor-WhereNext framework is shown in Algorithm 2.

Algorithm 2. Training process of Indoor-WhereNext framework.

```
Require: Trajectories of All Users: $trajArr = \{traj_i\}$
Hyperparameters of Indoor-STDBSCAN: $\epsilon_1, \epsilon_2, MinPts$
Weight coefficient: $\alpha$
Ensure: Prediction models: $\{model_i\}$
Cluster centers: $\{exemplar_i\}$
for next $traj \in trajArr$ do
    $\left\{sp_i^{id}\right\} = Indoor - STDBSCAN(traj, \epsilon_1, \epsilon_2, MinPts)$
    $locSeq^{id} = Conversion\left(\left\{sp_i^{id}\right\}\right)$
    $locSeqArr.add\left(locSeq^{id}\right)$
$\boldsymbol{SSM} = SSS(locSeqArr, \alpha)$
$\{locSeqSubArr_i\}, \{exemplar_i\} = AffinityPropagation(\boldsymbol{SSM})$
for $locSeqSubArr \in \{locSeqSubArr_i\}$ do
    $model = LSTM(locSeqSubArr)$
    $models.add(model)$
return $models, \{exemplar_i\}$
```

3.3.2. SSS-Based Location Prediction

After modeling, there is a one-to-one correspondence between models and exemplars—that is, $model_i$ corresponds to $exemplar_i$. Given a new user trajectory, the goal is to determine where the user is likely to visit next. First, a group is found that is more likely to be associated with the particular sequence of visits being considered in the forecasting task, and then the corresponding LSTM model is used to predict the most likely location. The prediction process of the Indoor-WhereNext framework is shown in Algorithm 3. Here, $\{examplari\}$ is used to determine to which group the new user belongs. In essence, exemplars are specific location sequences, so the similarity between the new location sequence and the exemplars is calculated. Then, the model with the highest similarity is chosen for location prediction.

Algorithm 3. Prediction process of Indoor-WhereNext framework.

```
Require: New user trajectory: $traj = \{pt_i\}$
Hyperparameters of Indoor-STDBSCAN: $\epsilon_1, \epsilon_2, MinPts$
Weight coefficient: $\alpha$
Prediction models: $\{model_i\}$
Cluster centers: $\{exemplar_i\}$
Ensure: $Set\ of\ predicted\ locations$
1: $\{sp_i^{id}\} = Indoor-STDBSCAN\ (traj, \epsilon_1, \epsilon_2, MinPts)$
2: $locSeq^{id} = Conversion\ (\{sp_i^{id}\})$
3: $\boldsymbol{SSM} = SSS(locSeq^{id}, \{exemplar_i\}, \alpha)$
4: $idx = argmax(\boldsymbol{SSM})$
5: $model = models[idx]$
6: $Set\ of\ predicted\ locations = model\ (locSeq^{id})$
7: return $Set\ of\ predicted\ locations$
```

4. Experimental Results and Analysis

4.1. Data Preparation

4.1.1. Data Sources

The experimental data consist mainly of Wi-Fi positioning data and shop data for a shopping mall in Jinan City, China. The total area of the mall is about 350,000 m^2. The indoor Wi-Fi positioning data cover the eight floors of the shopping mall from 23 December 2017 to 6 January 2018. The positioning accuracy was approximately 3 m, the total number of trajectories was more than 20 million, and the total number of trajectory points was 129,070,836. As shown in Table 1, the data field included the unique identifier of the user, the record upload time, the user's X,Y coordinates, and the unique identifier of the floor. As shown in Table 2, there are 489 shops in the mall. The average shop size is about 40 m^2. The data for each shop included the shop's unique ID, the shape of the shop (a polygon composed of the coordinate sequence), the shop name, and the floor ID.

Table 1. Sample of user trajectory data.

User ID	Date and Time	X (m)	Y (m)	Floor ID
0000CE ***	2017-12-31 10:46:45	130,219 ***	43,904 ***	1
0000CE ***	2017-12-31 10:46:57	130,219 ***	43,903 ***	1
0000CE ***	2017-12-31 10:47:05	130,219 ***	43,904 ***	1
……	……	……	……	……
0000CE ***	2017-12-31 19:20:33	130,219 ***	43,904 ***	4
0000CE ***	2017-12-31 19:20:45	130,219 ***	43,904 ***	4

Note: In order to protect the privacy of the user, the user's XY coordinates are represented by ***.

Table 2. Sample of shopping mall data.

Shop ID	Shape	Name	Floor ID
1	Polygon	***	2
2	Polygon	***	2
3	Polygon	***	6
……	……	……	……
488	Polygon	***	4
489	Polygon	***	3

Note: *** indicates the name of the specific shop.

4.1.2. Data Preprocessing

The indoor users' original trajectory data were collected via Wi-Fi positioning. Due to the instability of the mobile terminal signal and an artificial shutdown of the Wi-Fi signal, abnormal, erroneous, and invalid data was easily generated. The statistical characteristics of the users' original trajectories are shown in Figure 5. After data preprocessing, a total of 345, 824 user trajectories were obtained.

(1) The sampling interval for trajectory points was mostly concentrated between 1 and 5 s, accounting for approximately 82.5%, but there still were abnormal data with large sampling intervals and sampling intervals of 0 s. For example, trajectory points with sampling intervals of 0 s accounted for approximately 7.3%.

(2) The number of trajectory points contained in a trajectory was between 1 and 7 in most sets, accounting for more than 97%. In other words, a large number of trajectories contained only a few trajectory points and could not be used to train the model. In our work, trajectories where the number of trajectory points was less than 50 were deleted.

(3) The time span for trajectory points recorded in the shopping mall was 24 h—that is, there were records generated even during nonbusiness hours for the shopping mall, and the records generated in this process were invalid.

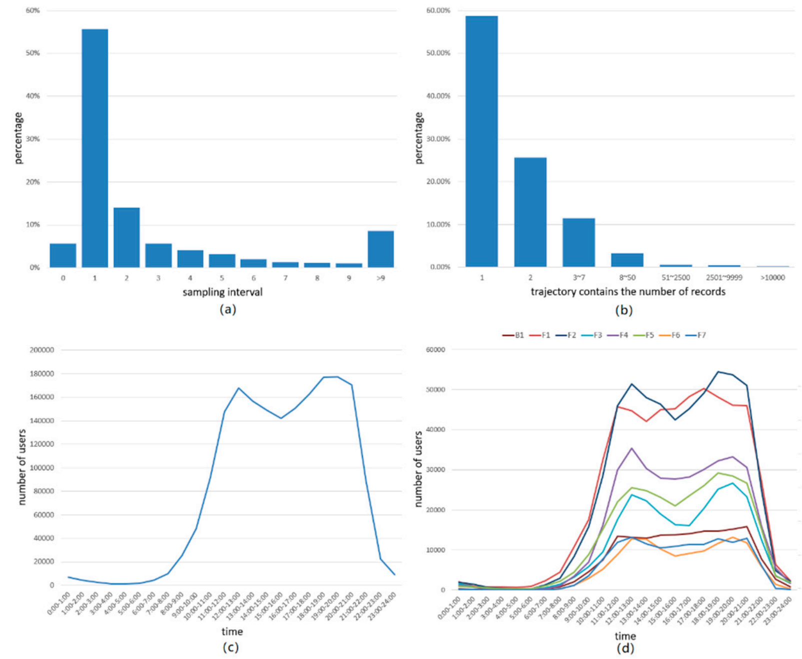


Figure 5. Statistical characteristics of the original trajectories of indoor users: (**a**) distribution of sampling intervals in the two weeks of data, (**b**) distribution of the numbers of trajectory points in the two weeks of data, (**c**) change in the number of users over time during a day, and (**d**) changes in the number of users on each floor over time during a day.

4.2. Evaluation Metrics

In this work, *Accuracy@k* and *Precision@k* (top k locations) were used as quantitative indicators of the evaluation model. *Accuracy@k* is used to evaluate the top-*k* prediction locations, to determine if they represent real locations. *Precision@X* uses macro-averaging to evaluate the performance of models from the perspective of multiple classifications—that is, indoor location prediction problems. *Accuracy@k*, and *Precision@k* are defined in Equations (6) and (7).

$$Accuracy@k = \frac{number\ of\ samples\ correctly\ predicted}{total\ number\ of\ test\ samples} \quad k \in \{1,\ 3,\ 5,\ 7,\ \ldots\} \tag{6}$$

$$Precision@k = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FP_i} \quad k \in \{1,\ 3,\ 5,\ 7,\ \ldots\} \tag{7}$$

where N represents the total number of locations and the number of shops; TP_i represents the number of samples in which the model correctly predicts that a user will visit location $shop_i$; FN_i represents the number of samples in which the model incorrectly predicts that a user will not visit location $shop_i$.

4.3. Variable Estimation

The value of hyperparameters has a considerable impact on the predictive performance of the model. When the value of the hyperparameters is not suitable, the model exhibits poor prediction performance. In this section, we calibrate the hyperparameters in the framework and analyze the impact of the hyperparameters on the prediction performance. The main hyperparameters of the Indoor-WhereNext framework are the radius ϵ_1, the time window ϵ_2, the minimum number of points *MinPts*, and the weight coefficient α. To determine the optimal hyperparameter of the framework, the control variable method was used to obtain the combination of parameter values with the best prediction accuracy. In the parameter estimation stage, first ϵ_1, ϵ_2, and *MinPts* in the Indoor-STDBSCAN algorithm were determined. Then, using these values, the weight coefficient α was adjusted to test the influence of semantic and spatial similarities on prediction accuracy.

4.3.1. Calibrating the Parameters of Indoor-STDBSCAN

In the Indoor-STDBSCAN algorithm, the main test time window ϵ_2 influences the prediction accuracy—that is, the test stay time influences the prediction result. In the parameter calibration process, the weight coefficient α was fixed to 0.5, the space radius ϵ_1 was fixed to 5 m with reference to the average distance between indoor shops. The time window ϵ_2 was the best parameter found in $[1\text{ min}, 3\text{ min}, 5\text{ min}, \ldots, 13\text{ min}]$. The minimum number *MinPts* was set to a fixed value according to the data average sampling interval, and the time window ϵ_2—that is, $MinPts = \frac{\epsilon_2}{average\ sampling\ interval}$. The effect of the time window ϵ_2 on the prediction accuracy *Accuracy@k* is shown in Figure 6. When $k \in \{1,\ 3,\ 5,\ 7,\ 9\}$, *Accuracy@k* increased initially and then became stable. When $\epsilon_2 \geq 5$ min, the prediction accuracy of the framework did not change much. However, as the time window increased, the number of location sequences tended to decrease—that is, the number of training data decreased. To ensure the prediction accuracy and the number of training data at the same time, the time window ϵ_2 was set to 7 min. After the Indoor-STDBSCAN parameter was calibrated, we further filtered the trajectory with too few stay points. A total of 45,315 trajectories was finally used for the experiment.

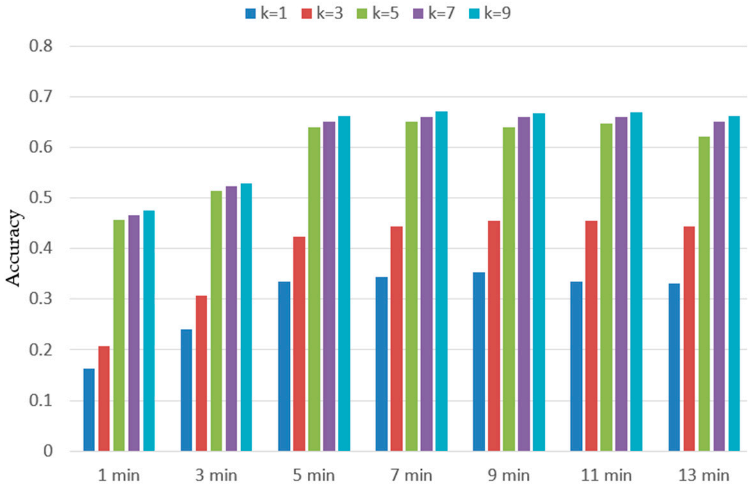


Figure 6. Impact of parameters ϵ_1, ϵ_2, and *MinPts* on prediction accuracy.

4.3.2. Calibrating the Weight Coefficient

The weight coefficient α mainly tests the influence of spatial similarity and semantic similarity on prediction accuracy. First, the hyperparameters in the Indoor-STDBSCAN algorithm are fixed. Then, α finds the optimal parameter from $[0, 0.1, 0.2, \ldots, 1]$. When α is set to 0 or 1, it means that only one similarity is considered to affect the accuracy of the prediction. The influence of weight coefficient α on prediction *Accuracy@k* is shown in Figure 7. When $k \in \{1, 3, 5, 7, 9\}$, *Accuracy@k* showed a trend of first increasing and then decreasing. When $0.3 \leq \alpha \leq 0.6$, *Accuracy@k* of the framework was relatively high. When $\alpha = 0.4$, *Accuracy*@5 reached 67.6%, an improvement of 17.6 and 24.1 percentage points, respectively, over that with $\alpha = 0$ and $\alpha = 1$. This indicates that both semantic and spatial similarity contributed to the accuracy of the model.

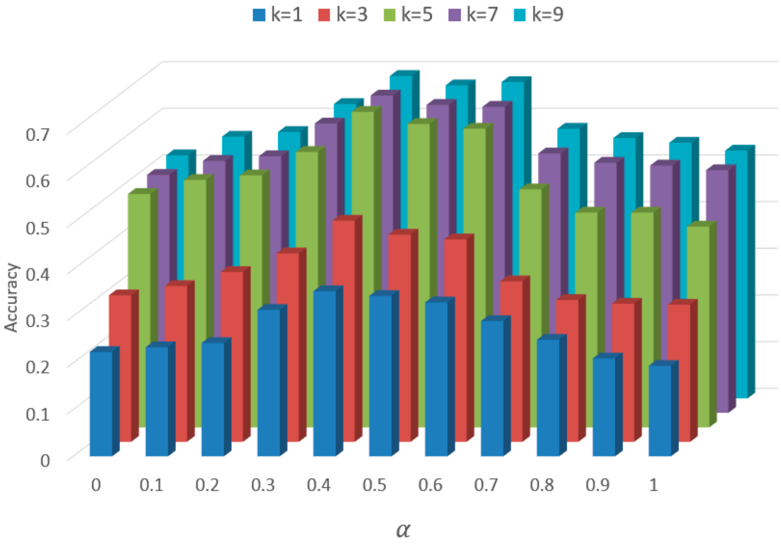


Figure 7. Impact of weight coefficient α on prediction accuracy.

4.4. Performance of Indoor-WhereNext

After calibration of the framework parameters, the change in the prediction accuracy of the Indoor-WhereNext framework with the number of iterations was analyzed. The results are shown in Figure 8.

(1) For the training dataset, the prediction accuracy showed a continuous upward trend with the increase in the number of iterations.

(2) For the test dataset, the prediction accuracy increased initially, then remained constant and finally decreased as the number of iterations increased. The framework tended to overfit as the number of iterations increased, improving the prediction accuracy of the model in the training dataset while worsening the prediction accuracy in the test dataset.

(3) Comparing *Accuracy@k* on the test dataset, when $k \in \{1, 3, 5\}$, the prediction accuracy of the model was greatly improved; at *Accuracy*@5, the prediction accuracy was 67.6%. Compared with *Accuracy*@1 and *Accuracy*@3, the prediction accuracy increased by 32.5% and 22.1%, respectively. However, as k continued to increase, the prediction accuracy of the model increased slowly. Compared with *Accuracy*@5, *Accuracy*@7 and *Accuracy*@9 only increased by 0.9% and 1.5%, respectively, because the shop that the next user visits in the mall is often a collection of shops rather than a specific shop. In the predicted set of shops, the user destination has a certain randomness.

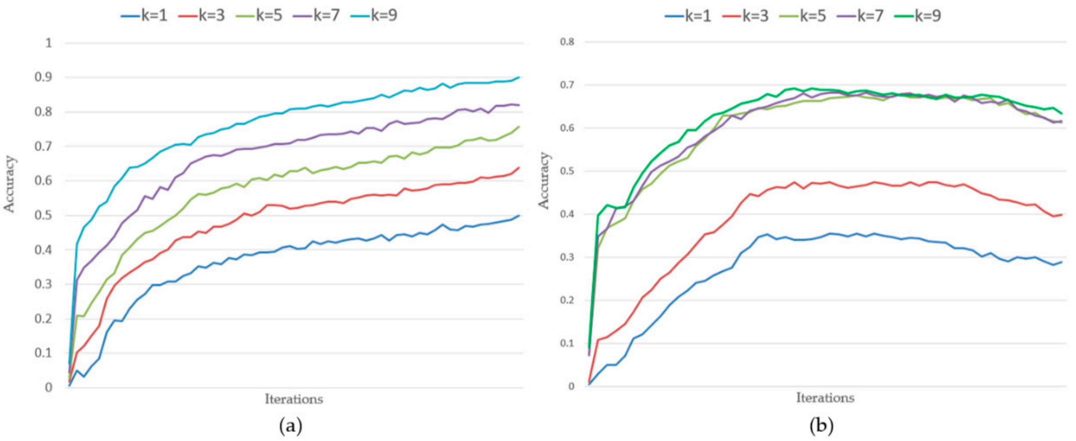


Figure 8. Indoor-WhereNext frame location prediction accuracy: (**a**) accuracy of location prediction on training dataset and (**b**) accuracy of location prediction on test dataset.

4.5. Comparison with Baselines

To ascertain the efficiency of the proposed Indoor-WhereNext framework, it was compared with three existing prediction models for datasets: HMM (original-HMM), the improved hidden Markov model (improved-HMM), and the LSTM model (original-LSTM). Of these, original-HMM and original-LSTM use the shop sequences to build a model to predict the next location. Improved-HMM replaces LSTM in the Indoor-WhereNext framework with HMM and builds models based on the SSS to predict the next location. The prediction accuracy of HMM is related to the number of states. In the comparison experiment, the number of states in HMM was varied among 10, 20, 30, and 40 states.

Figure 9 shows the prediction accuracy of the four models. It can be seen that, because the original-HMM and the original-LSTM models consider location prediction as a time series modeling problem, they ignore the influence of the similarity between location sequences on the location prediction. Therefore, their predictive performance was worse than those of improved-HMM and the proposed Indoor-WhereNext framework. In particular, when the number of states in HMM was 10, the *Accuracy*@5 of the Indoor-WhereNext framework was 31.2% higher than that of original-HMM and 23.8% higher than that of original-LSTM. Improved-HMM accounts for the similarities between location sequences and builds a model for similar users. However, when the number of states in HMM was 40, the *Accuracy*@1, *Accuracy*@3, and *Accuracy*@5 values of the Indoor-WhereNext framework were still 3.2%, 2.5%, and 13.8% higher, respectively, than those of Improved-HMM. The reason is that the LSTM model in the Indoor-WhereNext framework is used to model the location sequences, which makes it easier to capture the movement patterns in a long location sequence. In general, the Indoor-WhereNext framework greatly improved indoor location prediction by enhancing the *Accuracy*@1 by between 3.2% and 15.1%, the *Accuracy*@3 by between 2.4% and 18.3%, and the *Accuracy*@5 by between 13.8% and 31.9%.

Figure 10 compares the prediction precision of the four models. As in the case of accuracy, the precision of the framework can be improved by 7–27.3%, 17.8–20.9%, and 6.9–14.7% compared with the baseline experiments. In particular, when $k = 5$, the prediction precision of the model was 61.6%. However, compared to the accuracy of the framework, the precision of the Indoor-WhereNext framework is reduced by 6%. This reduction in accuracy can be attributed to the fact that the indicator precision regards location prediction as a multi-classification problem, and the test samples in each classification are unbalanced, resulting in a slight reduction in the precision. Overall, the Indoor-WhereNext framework significantly outperforms the three existing baseline methods in terms of prediction precision.

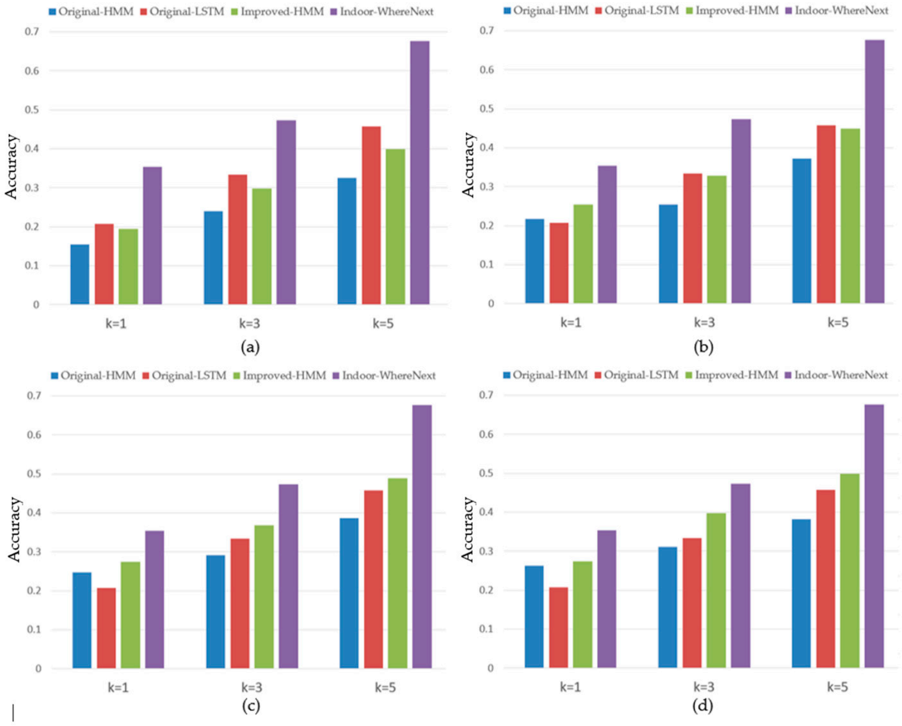


Figure 9. Accuracy using the baseline of the dataset: (**a**) the number of states in the hidden Markov model (HMM) is 10, (**b**) the number of states in the HMM is 20, (**c**) the number of states in the HMM is 30, and (**d**) the number of states in the HMM is 40.

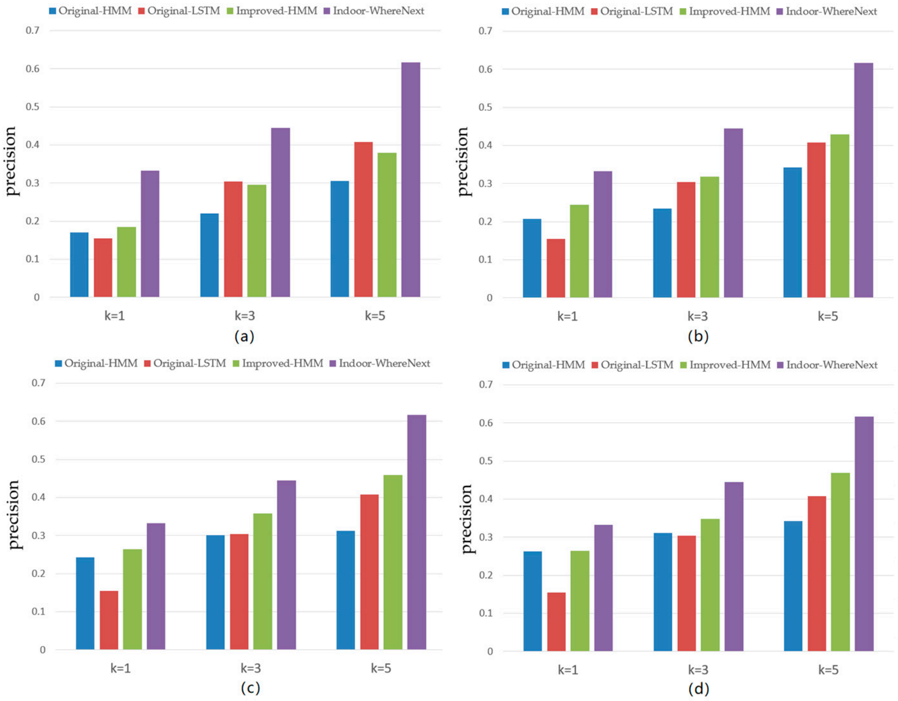


Figure 10. Precision using the baseline of the dataset: (**a**) the number of states in the hidden Markov model (HMM) is 10, (**b**) the number of states in the HMM is 20, (**c**) the number of states in the HMM is 30, and (**d**) the number of states in the HMM is 40.

5. Conclusions and Future Work

The Indoor-WhereNext framework was proposed for indoor location prediction. First, considering the three-dimensional characteristics and the relative error of indoor trajectories, the Indoor-STDBSCAN algorithm was proposed in order to identify the stay points of the indoor user and convert the user trajectory into a location sequence, thereby overcoming the problem that it is difficult to identify indoor stay points using the existing methods. Then, considering the spatial and semantic similarities of location sequences, the SSS method was defined to obtain the similarity matrix between location sequences. Finally, the AP algorithm was used to obtain similarity user groups based on the similarity matrix, and the groups were used to train different prediction models to improve the accuracy of location prediction.

In the experimental section, a two-week period of real indoor trajectories was used to verify the efficiency of the proposed framework. First, the control variable method was used to obtain the combination of parameter values with the best prediction accuracy. When the optimal parameters were used, the *Accuracy*@5 reached 67.6%. Then, a comparison with three existing baseline methods was conducted. Compared with original-HMM, original-LSTM, and improved-HMM, the proposed framework delivered improved accuracy and precision, with *Accuracy*@5 increasing by 31.2%, 23.8%, and 13.8%, and *Precision*@5 increasing by 27.3%, 20.9%, and 14.7%, respectively. This demonstrates the efficiency of the Indoor-WhereNext framework.

The following aspects can potentially be investigated further in future work: (1) further validation of the proposed framework with more types of data such as hospital indoor trajectories and airport indoor trajectories, (2) comprehensive comparison with other location prediction models such as ST-RNN and Markov chain, (3) comparison of the models with more comprehensive evaluation indicators, and (4) integration of more factors into Indoor-WhereNext to achieve a more robust model that further improves the accuracy of indoor location prediction.

Author Contributions: P.W. contributed to data preprocessing, the experiment, and the writing of the manuscript; S.W. gave advice on the experimental discussion and materials; H.Z. formulated the general research idea and contributed to writing the manuscript; F.L. contributed to the manuscript revision.

Funding: This project was supported by the National Natural Science Foundation of China (Grant Nos. 41771436 and 41701521), the National Key Research and Development Program of China (Grant Nos. 2016YFB0502104 and 2017YFB0503500), and the Digital Fujian Program (Grant No. 2016-23).

Acknowledgments: We are grateful to Shanghai Palmap Science & Technology Company Limited for providing indoor trajectory data support, which made this research possible.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, Y.-S.; Lin, S.-J.; Li, C.-R.; Tseng, T.; Li, H.-T.; Lee, J.-Y. Developing and validating a physical product e-tailing systems success model. *Inf. Technol. Manag.* **2018**, *19*, 245–257. [CrossRef]
2. Hajli, N.; Featherman, M.S. Social commerce and new development in e-commerce technologies. *Int. J. Inf. Manag.* **2017**, *37*, 177–178. [CrossRef]
3. Liu, Y.; Cheng, D.; Pei, T.; Shu, H.; Ge, X.; Ma, T.; Du, Y.; Ou, Y.; Wang, M.; Xu, L. Inferring gender and age of customers in shopping malls via indoor positioning data. *Environ. Plan. Urban Anal. City Sci.* **2019**. [CrossRef]
4. Zhang, H.; Wang, Z.; Chen, S.; Guo, C. Product recommendation in online social networking communities: An empirical study of antecedents and a mediator. *Inf. Manag.* **2018**, *56*. [CrossRef]
5. Dixit, V.S.; Gupta, S.; Jain, P. A Propound Hybrid Approach for Personalized Online Product Recommendations. *Appl. Artif. Intell.* **2018**, *32*, 785–801. [CrossRef]
6. Chan, N.N.; Gaaloul, W.; Tata, S. A recommender system based on historical usage data for web service discovery. *Serv. Oriented Comput. Appl.* **2012**, *6*, 51–63. [CrossRef]
7. Tomazic, S.; Dovzan, D.; Škrjanc, I. Confidence-Interval-Fuzzy-Model-Based Indoor Localization. *IEEE Trans. Ind. Electron.* **2018**, *66*, 2015–2024. [CrossRef]

8. Li, H.; Lu, H.; Shou, L.; Chen, G.; Chen, K. In Search of Indoor Dense Regions: An Approach Using Indoor Positioning Data. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1481–1495. [CrossRef]
9. Guo, S.; Xiong, H.; Zheng, X.; Zhou, Y. Activity Recognition and Semantic Description for Indoor Mobile Localization. *Sensors* **2017**, *17*, 649. [CrossRef]
10. Koehler, C.; Banovic, N.; Oakley, I.; Mankoff, J.; Dey, A.K. Indoor-ALPS:an adaptive indoor location prediction system. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Seattle, WA, USA, 13–17 September 2014; ACM: New York, NY, USA, 2014; pp. 171–181. [CrossRef]
11. Song, C.; Qu, Z.; Blumm, N.; Barabási, A.-L. Limits of Predictability in Human Mobility. *Science* **2010**, *327*, 1018–1021. [CrossRef]
12. Lu, X.; Wetter, E.; Bharti, N.; Tatem, A.J.; Bengtsson, L. Approaching the Limit of Predictability in Human Mobility. *Sci. Rep.* **2013**, *3*, 2923. [CrossRef] [PubMed]
13. Mingxiao, L.; Song, G.; Feng, L.; Hengcai, Z. Reconstruction of human movement trajectories from large-scale low-frequency mobile phone data. *Comput. Environ. Urban Syst.* **2019**, *77*, 101346. [CrossRef]
14. Ye, M.; Yin, P.; Lee, W.-C. Location recommendation for location-based social networks. In Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, San Jose, CA, USA, 2–5 November 2010; ACM: New York, NY, USA, 2010; pp. 458–461. [CrossRef]
15. Kuang, L.; Yu, L.; Huang, L.; Wang, Y.; Ma, P.; Li, C.; Zhu, Y. A Personalized QoS Prediction Approach for CPS Service Recommendation Based on Reputation and Location-Aware Collaborative Filtering. *Sensors* **2018**, *18*, 1556. [CrossRef] [PubMed]
16. Bao, J.; Zheng, Y.; Mokbel, M.F. Location-based and preference-aware recommendation using sparse geo-social networking data. In *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*; ACM: New York, NY, USA, 2012; pp. 199–208. [CrossRef]
17. Zhang, X.; Zhao, Z.; Zheng, Y.; Li, J. Prediction of Taxi Destinations Using a Novel Data Embedding Method and Ensemble Learning. *IEEE Trans. Intell. Transp. Syst.* **2019**, 1–11. [CrossRef]
18. Li, X.; Li, M.; Gong, Y.-J.; Zhang, X.-L.; Yin, J. T-DesP: Destination Prediction Based on Big Trajectory Data. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2344–2354. [CrossRef]
19. Bogomolov, A.; Lepri, B.; Staiano, J.; Oliver, N.; Pianesi, F.; Pentland, A. Once Upon a Crime: Towards Crime Prediction from Demographics and Mobile Data. In Proceedings of the 16th International Conference on Multimodal Interaction, Istanbul, Turkey, 12–16 November 2014; ACM: New York, NY, USA, 2014.
20. Zhao, Z.; Koutsopoulos, H.N.; Zhao, J. Individual mobility prediction using transit smart card data. *Transp. Res. Part Emerg. Technol.* **2018**, *89*, 19–34. [CrossRef]
21. Monreale, A.; Pinelli, F.; Trasarti, R.; Giannotti, F. WhereNext: A location predictor on trajectory pattern mining. In Proceedings of the Acm Sigkdd International Conference on Knowledge Discovery & Data Mining, Paris, France, 28 June–1 July 2009; ACM: New York, NY, USA, 2009; pp. 637–646. [CrossRef]
22. Lee, S.; Lim, J.; Park, J.; Kim, K. Next Place Prediction Based on Spatiotemporal Pattern Mining of Mobile Device Logs. *Sensors* **2016**, *16*, 145. [CrossRef]
23. Ying, J.C.; Lee, W.C.; Tseng, V.S. Mining Geographic-Temporal-Semantic Patterns in Trajectories for Location Prediction. *ACM Trans. Intell. Syst. Technol.* **2014**, *5*, 1–33. [CrossRef]
24. Wu, R.; Luo, G.; Yang, Q.; Shao, J. Learning Individual Moving Preference and Social Interaction for Location Prediction. *IEEE Access* **2018**, *6*, 10675–10687. [CrossRef]
25. Gambs, S.; Killijian, M.-O.; del Prado Cortez, M.N. Next Place Prediction using Mobility Markov Chains. In Proceedings of the 1st Workshop on Measurement, Privacy, and Mobility, MPM'12, Bern, Switzerland, 10 April 2012; ACM: New York, NY, USA, 2012. [CrossRef]
26. Hawelka, B.; Sitko, I.; Kazakopoulos, P.; Beinat, E. Collective Prediction of Individual Mobility Traces for Users with Short Data History. *PLoS ONE* **2017**, *12*, e0170907. [CrossRef]
27. Keles, T.I.; Ozer, M.; Toroslu, I.; Karagoz, P. Location prediction of mobile phone users using apriori-based sequence mining with multiple support. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8983, pp. 179–193. [CrossRef]
28. Morzy, M. *Prediction of Moving Object Location Based on Frequent Trajectories*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 583–592. [CrossRef]

29. Mathew, W.; Raposo, R.; Martins, B. Predicting future locations with hidden Markov models. In Proceedings of the UbiComp'12—2012 ACM Conference on Ubiquitous Computing, Pittsburgh, Pennsylvania, 5–8 September 2012; ACM: New York, NY, USA, 2012; pp. 911–918. [CrossRef]
30. Qiang, L.; Shu, W.; Liang, W.; Tan, T. Predicting the next location: a recurrent model with spatial and temporal contexts. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; AAAI Press: Palo Alto, CA, USA, 2016; pp. 194–200.
31. Di, Y.; Chao, Z.; Huang, J.; Bi, J. *SERM: A Recurrent Model for Next Location Prediction in Semantic Trajectories*; ACM: New York, NY, USA, 2017; pp. 2411–2414. [CrossRef]
32. Ye, Y.; Zheng, Y.; Chen, Y.; Feng, J.; Xie, X. Mining Individual Life Pattern Based on Location History. In Proceedings of the IEEE International Conference on Mobile Data Management, Taipei, Taiwan, 18–21 May 2009; pp. 1–10.
33. Vu, L.; Do, Q.; Nahrstedt, K. Jyotish: Constructive approach for context predictions of people movement from joint Wifi/Bluetooth trace. *Pervasive Mob. Comput.* **2011**, *7*, 690–704. [CrossRef]
34. Vu, L.; Nguyen, P.; Nahrstedt, K.; Richerzhagen, B. Characterizing and modeling people movement from mobile phone sensing traces. *Pervasive Mob. Comput.* **2015**, *17*, 220–235. [CrossRef]
35. Do, T.M.T.; Dousse, O.; Miettinen, M.; Gatica-Perez, D. A Probabilistic Kernel Method for Human Mobility Prediction with Smartphones. *Pervasive Mobile Comput.* **2014**, *20*, 13–28. [CrossRef]
36. Wu, F.; Fu, K.; Wang, Y.; Xiao, Z.; Fu, X. A Spatial-Temporal-Semantic Neural Network Algorithm for Location Prediction on Moving Objects. *Algorithms* **2017**, *10*, 37. [CrossRef]
37. Zhou, Y.; Sun, H.; Huang, J.; Jia, X.; Zhao, Z. Efficient Destination Prediction Based on Route Choices with Transition Matrix Optimization. *IEEE Trans. Knowl. Data Eng.* **2018**, *14*. [CrossRef]
38. Ang, B.-K.; Dahlmeier, D.; Lin, Z.; Huang, J.; Seeto, M.-L.; Shi, H. Indoor Next Location Prediction with Wi-Fi. In Proceedings of the Fourth International Conference on Digital Information Processing and Communications, Kuala Lumpur, Malaysia, 18–20 March 2014.
39. Sepahkar, M.; Khayyambashi, M.R. A novel collaborative approach for location prediction in mobile networks. *Wirel. Netw.* **2018**, *24*, 283–294. [CrossRef]
40. Zhang, D.; Zhang, D.; Xiong, H.; Yang, L.T.; Gauthier, V. NextCell: Predicting Location Using Social Interplay from Cell Phone Traces. *IEEE Trans. Comput.* **2015**, *64*, 452–463. [CrossRef]
41. Wen, L.; Shi-Xiong, X.; Feng, L.; Lei, Z. Improving Location Prediction by Exploring Spatial-Temporal-Social Ties. *Math. Probl. Eng.* **2014**, *2014*, 1–7. [CrossRef]
42. Li, J.; Brugere, I.; Ziebart, B.; Bergerwolf, T.; Crofoot, M.; Farine, D. Social Information Improves Location Prediction in the Wild. In Proceedings of the Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015. [CrossRef]
43. Spaccapietra, S.; Parent, C.; Damiani, M.L.; De Macedo, J.A.; Porto, F.; Vangenot, C. A conceptual view on trajectories. *Data Knowl. Eng.* **2008**, *65*, 126–146. [CrossRef]
44. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the International Conference on Knowledge Discovery & Data Mining, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
45. Birant, D.; Kut, A. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data Knowl. Eng.* **2007**, *60*, 208–221. [CrossRef]
46. Iliopoulos, C.S.; Rahman, M.S. A New Efficient Algorithm for Computing the Longest Common Subsequence. *Theory Comput. Syst.* **2009**, *45*, 355–371. [CrossRef]
47. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. In Proceedings of the KDD: International Conference on Knowledge Discovery & Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
48. Yuan, G.; Sun, P.; Zhao, J.; Li, D.; Wang, C. A review of moving object trajectory clustering algorithms. *Artif. Intell. Rev.* **2016**, *47*, 123–144. [CrossRef]
49. Chen, L.; Özsu, M.T.; Oria, V. Robust and fast similarity search for moving object trajectories. In Proceedings of the 2005 ACM SIGMOD international conference, Baltimore, MD, USA, 14–16 June 2005; p. 491.
50. Sankoff, B.D.; Kruskal, J.B. Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Reading, MA: Addison-Wesley. *J. Logic Comput.* **1983**, *11*, 356.

51. Frey, B.J.; Dueck, D. Clustering by Passing Messages Between Data Points. *Science* **2007**, *315*, 972–976. [CrossRef] [PubMed]
52. Sepp, H.; Jürgen, S. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]