Chinese Journal of Electronics
vol. 33, no. 3, pp. 708–720, May 2024
https://doi.org/10.23919/cje.2022.00.363

RESEARCH ARTICLE

Friendship Inference Based on Interest Trajectory Similarity and Co-occurrence

Junfeng TIAN[1,2] and Zhengqi HOU[1,2]

1. *School of Cyber Security and Computer, Hebei University, Baoding 071000, China*
2. *Hebei Key Laboratory of High Confidence Information Systems, Hebei University, Baoding 071000, China*

Corresponding author: Zhengqi HOU, Email: hozi_nl@hotmail.com
Manuscript Received October 27, 2022; Accepted March 14, 2023

Abstract — Most of the current research on user friendship speculation in location-based social networks is based on the co-occurrence characteristics of users, however, statistics find that co-occurrence is not common among all users; meanwhile, most of the existing work focuses on mining more features to improve the accuracy but ignoring the time complexity in practical applications. On this basis, a friendship inference model named ITSIC is proposed based on the similarity of user interest tracks and joint user location co-occurrence. By utilizing MeanShift clustering algorithm, ITSIC clustered and filtered user check-ins and divided the dataset into interesting, abnormal, and noise check-ins. User interest trajectories were constructed from user interest check-in data, which allows ITSIC to work efficiently even for users without co-occurrences. At the same time, by application of clustering, the single-moment multi-interest trajectory was further proposed, which increased the richness of the meaning of the trajectory moment. The extensive experiments on two real online social network datasets show that ITSIC outperforms existing methods in terms of AUC score and time efficiency compared to existing methods.

Keywords — Location social network, Interest clustering, Multi-interest track, Track similarity, Friendship prediction.

Citation — Junfeng TIAN and Zhengqi HOU, "Friendship Inference Based on Interest Trajectory Similarity and Co-occurrence," *Chinese Journal of Electronics*, vol. 33, no. 3, pp. 708–720, 2024. doi: 10.23919/cje.2022.00.363.

I. Introduction

In recent years, with the rapid development of intelligent devices and the Internet, more and more social applications provide users with location-sharing functions, allowing users to check in, sign in, etc. With the addition of location information, a new location-based social network is gradually formed, and location information has become an indispensable element in social networks [1]. For example, when users go to travel, they will share their location through social networks to clock in scenic spots. Similarly, when users attend a party, there will also be social news with location information published. Location sharing has become the instinct of every user in the location social network. Location-based social applications, on the other hand, are allowed to continuously read the location of the user to provide a better service to the user, which to the service provider is equivalent to the user checking in or clocking in real-time. Based on the above two aspects, it is easy for both data collectors and data managers to collect a large amount of check-in data from location-based social networks, which provides a basis for researchers to study social behaviors of users.

It has attracted wide attention from researchers to infer the relationship between users through their check-in information. According to the social homogeneity principle [2], friends tend to go to the same place for some social events, such as attending a mutual friend's wedding or eating together in a restaurant. Unlike strangers, friends often share the same interests and hobbies. These commonalities of users' interests are reflected in users' check-in behaviors, making it possible to infer users' relationships according to their check-in information. Algorithms for predicting user relationships have been widely used in friend recommendation [3], social influence analysis [4], and targeted marketing [5]. Although these studies have made fruitful progress, there are also some shortcomings, mainly in the following two aspects:

1) Excessive reliance on co-location reduces reasoning performance. Most of the previous work is to mine

Associate Editor: Prof. Debiao HE, Wuhan University.

the co-location between user check-in and then construct the feature of user co-location for friendship prediction. This method entirely relies on co-location between users, so the predicted performance will be greatly affected when there is little or no co-location between users. We refer to the statistical results of users co-located in the Gowalla and Brightkite data sets [6] in [7] to observe the proportion of user pairs with multiple co-locations under different thresholds. The results are shown in Table 1. When the thresholds were set to 120 min and 200 m, The number of users with more than one co-location in the two data sets only reached 30.25% and 47.17%, respectively, which indicates that the co-location between users is relatively small.

Table 1 The proportion of co-location users at different thresholds

Threshold value range	Gowalla	Brightkite
30 min, 0 m	19.04%	22.72%
30 min, 100 m	21.35%	31.44%
30 min, 200 m	23.03%	33.60%
60 min, 0 m	22.57%	33.13%
60 min, 100 m	24.92%	37.52%
60 min, 200 m	26.85%	40.20%
120 min, 0 m	25.73%	38.88%
120 min, 100 m	28.13%	44.05%
120 min, 200 m	30.25%	47.17%

2) Low efficiency of reasoning. Friendship reasoning has begun to be applied in practice. The existing research focuses on how to effectively construct and extract characteristic values to improve the accuracy of prediction, but the time efficiency of friendship reasoning in practical application is not taken into account in most cases. Often location-based social network user check-in data are enormous, generating multiple features based on large-scale data sets to conduct friendship reasoning. Its time efficiency is not ideal. In the past, most of the data sets were simply screened, and the data that had no practical effect on friendship prediction were removed as noise data, such as removing users whose check-in number was lower than a certain threshold or the number of check-in locations was lower than a certain threshold, and removing duplicate check-in data. As shown in Tables 2 and 3, the number of users and check-ins decreased after filtering by CNum (check num) and LNum (visit-location num), but the data set size was still large after simple filtering. In fact, for those users who meet the filtering conditions, there are multiple noise data in their multiple check-in data. Finally, the large-scale data set and the noise data in it reduce the efficiency and effect of the inference to a certain extent.

To solve the above problems, we propose the friendship inference method based on check-in clustering similarity and co-occurrence of interest trajectories (ITSIC) to infer friendship and design and implement two schemes,

Table 2 Comparison of Gowalla screening

Parameter	Origin	CNum > 5	LNum > 5	both
User_num	107k[1]	83k	79k	79k
Check_num	0.64M[2]	0.63M	0.63M	0.58M

Note: [1]k denotes thousand; [2]M denotes million.

Table 3 Comparison of Brightkite screening

Parameter	Origin	CNum > 5	LNum > 5	both
User_num	51k[1]	31k	21k	21k
Check_num	0.47M[2]	0.46M	0.37M	0.33M

Note: [1]k denotes thousand; [2]M denotes million.

i.e., sin_SCI and mul_SCI. First, we construct a single-interest trajectory scheme sin_SCI based on clustering and use the similarity of the trajectory as the inferred feature to test the effectiveness of the trajectory-based scheme to solve the above problems. Friendship inference is performed based on the similarity of the trajectory in conjunction with location diversity, location popularity, co-occurrence duration, and stability characteristics. The experimental results demonstrate the effectiveness of trajectory features, so we further propose a single-moment multi-interest trajectory scheme mul_SCI based on clustering. After clustering, we filter all clusters in the clustering results at a single moment. Finally, multiple interest clusters at a single moment are obtained, which are used as a moment representation of the trajectory, and then a trajectory containing more user interests is constructed. The similarity between trajectories is used as a trajectory feature and location co-occurrence to perform friendship prediction jointly. The experimental results show that a better prediction effect is achieved based on the single-moment multi-interest trajectory prediction.

To sum up, the contributions of this paper are as follows:

1) Based on the MeanShift [8] clustering algorithm, the initial check-in data set is clustered and filtered, the user interest check-in, abnormal check-in, and noise data sets are extracted, and the influence of different types of check-in data on friendship speculation is explored.

2) The ITSIC model is proposed, and two schemes of the single-interest trajectory (sin_SCI) and multi-interest trajectory (mul_SCI) are proposed, respectively, which solves the problem that the previous methods cannot effectively predict user-pair relationships without co-occurrence. And the effectiveness of trajectory similarity for friendship reasoning is verified. Meanwhile, the mul_SCI scheme explores a finer-grained representation of user interest trajectories, increasing the effectiveness of trajectory similarity features.

3) Extensive experiments are conducted on two real datasets to evaluate the performance of ITSIC. The experimental results show that ITSIC outperforms the baseline methods with higher inference performance and

time efficiency.

The remaining chapters of this paper are as follows. Section II briefly introduces related work on social relational reasoning in social networks. Section III describes the method proposed in this paper in detail. Section IV presents the experimental procedure and results. Section V discusses the effect of clustering bandwidth and the value of the effective cluster parameter c_num on the inference effect. Section VI summarizes the work of this paper and provides an outlook for the future.

II. Related Works

This section introduces some relevant research on solving social relationship reasoning problems. According to the characteristics of the methods, they are mainly divided into co-location-based research methods and other methods.

1. Research methods based on co-location

Existing studies mainly extract co-location between users by setting distance and time thresholds and then performing feature mining on these co-locations. The obtained features are divided into deep features obtained through machine learning and shallow features carefully designed by humans.

Pham *et al.* proposed the EBM model [9] based on entropy, which designed two features: position diversity and weight frequency. Location diversity is used to measure the diversity of the co-location of a pair of users, and location entropy is used to measure the importance of the co-location of user pairs in weight frequency. However, they did not consider the influence of time information in co-location, resulting in many meaningless co-locations being considered in the inference process, reducing the prediction accuracy. To solve this problem and consider the factors affecting co-location, Wang *et al.* proposed the PGT [10] model, which believed that the co-location of user pairs was not equally important. It considered the personal, global, and time factors of co-location. It determined the significance of the co-location of two users at a specific location by using the probability distribution of personal location access. Subsequently, to further refine the impact of time factors on co-occurrence, SCI framework (social connection inference framework) [11] proposed by Njoo *et al.* derived two characteristics: stability and duration of co-occurrence time, which can reflect the consistency and total duration of co-location between two users. Njoo *et al.* then improved the SCI model and proposed SCI+ [12], four key features: co-location diversity, location prevalence, duration of co-location, and stability were used to predict friendship. Meanwhile, STIF framework [13] proposed by He *et al.* designed 12 spatio-temporal features from four aspects to infer friendship. As the number of features increases, the time efficiency of the friendship conjecture gradually decreases. So, Bayrak *et al.* [14] proposed the OSFLC model, which collects all features from past studies to generate a feature set and improves prediction efficiency without reducing prediction accuracy by manually selecting subsets from all feature sets. To explore more fine-grained co-occurrence, Wang *et al.* [15] collected access records of campus personnel based on the AP wireless network as user check-in, in this work, PMI point mutual information is used to measure the importance of the occurrence, end, and duration of co-occurrences. Previously proposed methods are based on the co-location of users to infer, but in fact, co-location between users does not exist in large numbers, so these methods have certain limitations. Subsequently, He *et al.* [7] proposed the CIFEF models, they believe that previous work used location entropy to measure the popularity of two users in public places without considering the time interval of co-location. If two users visit the same place but with long intervals between visits, they may not have any relationship, but if two users visit the same place with short intervals, they are likely to be friends. The time interval between visits to the same location by two users has a significant impact on judging friendship between users. Based on this, an explicit feature twice is introduced, which considers the influence brought by the time interval between two users in a certain place. Then, the embedding technology word2vec is used to learn the potential vector representation of user trajectory ($\boldsymbol{V}_u = [v_u^1, v_u^2, \dots, v_u^n]^{\mathrm{T}}$) by taking user trajectory as a word. Finally, the two features combined to predict friendship. The proposed implicit feature makes up for the deficiency of the method that only relies on co-location, improves the friendship reasoning function to a certain extent, and reduces the time efficiency of inference.

2. Other methods

In addition to artificially constructed co-location features, other methods measure the similarity of user pairs by carefully designed location features, such as location distance of home [16], Jaccard similarity of check-in sequence, and the number of common locations, etc. Other studies propose using machine learning algorithms to construct features and mine users' mobile features through machine learning algorithms to derive friendships between users, such as Backes *et al.* [17] proposed a two-step framework to mine the user mobile characteristics and then measure the user relationship. Gao *et al.* [18] embedded user check-ins as low-dimensional vectors, segmented user trajectories according to periods, and used autoencoders, variational autoencoders, and LSTM to learn and predict user trajectories and check-ins, respectively. Zhou *et al.* [19] proposed the vec2Link model to infer social connections from friendship and mobility data.

Recently, some scholars have studied graph-based friendship inference. Ren *et al.* [20] proposed a category-aware multi-bipartite graph embedding (CMGE) for cross-region friendship reasoning. The model exploits multi-bipartite graph embeddings to capture users' point-of-interest (POI) neighbor similarity and activity category similarity, and then learns the contribution of each POI and category through a category-aware heterogeneous graph attention network. Gao *et al.* [21] proposed a graph-based social circle inference framework (GSCI). This framework integrates the contextual information of user points of interest (POIs) into density representations from a graph learning perspective, introducing a mobile-specific end-to-end paradigm with varying attention. Based on this paradigm, it can more meaningfully encode discriminative patterns into the trajectory representation, and finally a graph neural network-based classifier to model the intrinsic correlation between user connections, which significantly improves the inference performance.

Notably, friendship reasoning also occurs in other scenarios. Li *et al.* [22], [23] based on the vehicle network dataset of a vehicle positioning system, friendship inference based on spatial and temporal co-occurrence features and friendship inference based on SEME framework learning user motion vectors are implemented. But this paper believes that vehicles can only walk on specified roads, and people can appear anywhere on the land. The Internet of vehicles scenario is different from ours, so we do not consider this work.

In the above methods, the methods based on co-occurrence focus on how to mine and design more effective co-occurrence features, to improve the reasoning effect of the proposed model. However, the fact that there is virtually no co-occurrence among a large number of users has limited the performance of co-occurrence-based methods. Although CIFEF considers co-occurrence, it uses embedded learning to obtain the latent vector representation of user trajectories to make up for the lack of co-occurrence, but the time complexity of machine learning also brings time efficiency problems, and with the increase of the number of features, the time cost of speculation is increasing. Combined with diversified module speculation, each module is equivalent to a separate speculation model. Although the combination of multiple modules, although the accuracy greatly improved, it also further increases the time overhead. To keep the time overhead low while ensuring the improvement of the inference effect, we propose data processing based on user interests, hoping to achieve better inference effect and inference efficiency by increasing the practicability of the data and reducing the data scale. At the same time, to get rid of the dependence of previous methods on co-occurrence, this paper proposes two representation schemes of user interest trajectories and infers the relationship of users by calculating the similarity between user interest trajectories.

III. The Proposed Method

This section details the two implementation schemes of the proposed ITSIC inference model. First, it introduces the clustering operation of the user check-in data and describes the process of the clustering algorithm; Then, it introduces the screening of clustering results, that is, extracting valid check-in data; Then, the construction of the single-interest trajectory and the construction of single-moment multi-interest trajectory are described respectively. Finally, the similarity calculation under the two trajectory construction methods is introduced.

1. Inference model

To solve the issue of degraded inference performance and efficiency due to over-reliance on co-occurrence in friendship speculation, this paper proposes an ITSIC inference model, as shown in Figure 1. First, the check-in data is clustered and filtered to obtain valid check-ins among users, improve the data quality, and reduce the data set. Then, for the problem that the previous methods that only consider co-occurrence cannot effectively infer friendship when facing users without co-occurrence, ITSIC improves the SCI+ [12] model by constructing user interest trajectories and adding interest trajectory similarity features, which can effectively handle each pair of users regardless of whether there is co-occurrence between pairs of users.

In general, by clustering and filtering the data set, higher-quality check-in data can be obtained first, thereby improving the effect of friendship speculation and reducing the time overhead of speculation; Secondly, the user interest trajectory is constructed, and the similarity feature of user interest trajectory is proposed to deal with those users without co-occurrence, and then the co-occurrence feature is combined as the input of the random forest classifier to infer user friendship.

2. Check-in clustering and screening

In the existing friendship reasoning research, most researchers filter out some noise check-in from the surface of the data set. In fact, for user check-ins in the dataset, not all check-in data has a positive meaning for inferring friendship between users, but there is a part of check-in data that can effectively express the relationship between users. We call this part of the check-in data as effective check-in of users, and the rest of the check-in that has no positive meaning to friendship speculation of user is the noise check-in. Utilizing effective user check-ins for friendship prediction enhances prediction accuracy and time efficiency due to reduced data volume.

Effective check-ins are obtained through clustering and screening. MeanShift [8] migration clustering algorithm is used to cluster users' check-ins into a series of check-in clusters and their corresponding cluster centers (cluster centers are represented by latitude and longitude, for example (−45.789, 122.3456)). The input of the algorithm is check-in data of users, clustering bandwidth parameter bandwidth, and the output is a series of clusters and their cluster centers. The algorithm we use is algorithm 1, in which the MeanShift migration clustering algorithm is included. The algorithm input includes user

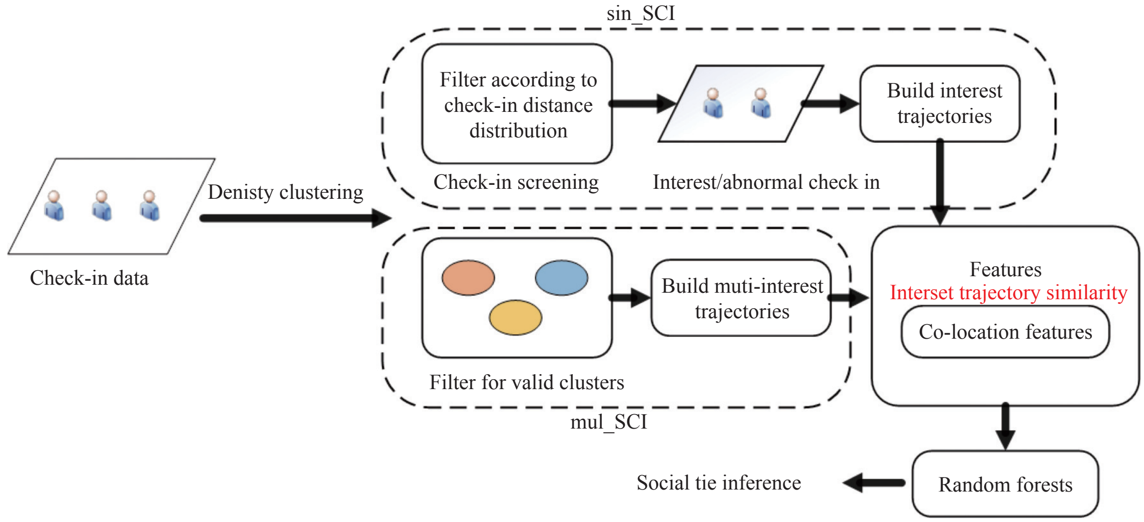

Figure 1 Proposed reasoning model.

check-ins D, user list U_1, clustering bandwidth b, check-ins partition threshold R_h, R_{irr}, and the algorithm output is user interest check-in, abnormal check-in, and noise data.

Algorithm 1 Check-in clustering

Input: Check-in data D, user list U_1, bandwidth b, Threshold R_h, R_{irr}.

Output: Interest/Irregular check-in.

```
1: for user u in U_1 do
2:     Get single user check-in Du and remove duplicate data;
3:     //clustering of user check-in in each period
4:     for time t in (0,23) do
5:       df_time = D_u. timestamp(t, t+1);
6:       if len(df_time)>5 then
7:         df_time[irr] = 0;
8:         Get clustering results df_time according b;
9:         Select effective sign in according to the distance from the sign in point to the cluster center;
10:        for row in df_time do
11:          dis = distance(cen[0][0], cen[0][1], row.latitude, row.longitude);
12:          if R_irr>dis>R_h then
13:            df_time.loc[row.Index, irr] = 1;
14:          end if
15:        end for
16:      else if df_time.empty then
17:        continue
18:      else
19:        temp_df = df_time;
20:      end if
21:      df_user = df_temp.append(temp_df);
22:    end for
23:    D_check = D_check. append(df_user);
24: end for
25: return D_check.
```

The specific processing process: firstly, all check-ins are grouped according to users in line 3. Then, in lines 5 to 6, the user check-in is divided into 24 groups according to the time. The time interval of each group is 1 h. The MeanShift algorithm is used to process each group of data. Then the cluster center O_m of the largest cluster is selected from the output clusters as the interest center for user check-in. With O_m as the center, two thresholds R_h and R_{irr} are set simultaneously (the specific settings will be explained below) to divide check-in data of users, where R_h stands for the threshold of interest check-in. In lines 12 to 15, we calculate the distance from each check-in to the cluster center and divide the interest check-in, irregular sign in, and noise data according to this distance. Specifically, in line 14, we get the corresponding data by setting the if condition. When the condition is set to $dis < R_h$, we get the interest check-in; When the condition is set to $dis > R_h$ and $dis > R_{irr}$, we get irregular check-in; When $dis > R_{irr}$ is finally set, noise data is obtained, but this is unnecessary. Finally, we stored the corresponding data of the corresponding user in D_{check}.

3. User trajectory construction

Existing works dealing with user trajectories either use dwell points as single-temporal locations of user trajectories or use embedded learning to learn user trajectory representations. The former uses a single stop point to express only part of the meaning of the trajectory, while the latter construct the trajectory with the context, but the construction is complex. To more comprehensively describe the meaning of each moment position in the trajectory of users and reduce the complexity of trajectory construction. This paper proposes two kinds of user interest point trajectory representations based on clustering and predicts the relationship between two users based on the similarity feature of the trajectory.

1) Single-interest trajectory construction

A single-interest trajectory is a fixed-length trajectory obtained based on all the check-in information of a single user. There are two forms of single-interest trajectory. The first is point-based trajectory. We divide the check-in information of a single user after clustering and filtering into 24 segments according to time (i.e., $[(0,1),(1,2),\ldots,(23,0)]$). Then, the interest location of a user in each period is calculated separately. For example, from 12 o'clock to 13 o'clock, the experiment sets the average of all check-in latitudes and longitude in this period as the latitude and longitude of the user interest point, and its calculation is expressed as

$$l_t=\left(\frac{\sum_{i=1}^{n}P_{i,\text{lon}}}{n},\frac{\sum_{i=1}^{n}P_{i,\text{lat}}}{n}\right) \tag{1}$$

where t represents the t-th period, $P_{i,\text{lon}}$ represents the longitude of the i-th check-in at this time, and $P_{i,\text{lat}}$ represents the latitude of the i-th check-in at this time. It should be noted that when the user does not check in at this period, the position of this period will be set to empty. After counting the position of each moment, sort the positions in order of period to obtain the interest trajectory of users, which is expressed as $S_u=\{l_0,l_1,l_2,\ldots,l_{23}\}$.

2) Multi-interest trajectory construction

The multi-interest trajectory is also of fixed length. Different from the single-interest trajectory, it has multiple clusters at a single moment as the interest clusters at that moment. First, the initial check-in data is clustered, and each user will get multiple clusters $c\in C_t$ at a single time t. To reduce the number of clusters and reduce the computational complexity, we set the proportional threshold c_num to filter the clusters at a single moment of the user when the ratio of the number of check-ins in the user current cluster to the total number of check-ins at the current moment is less than c_num, the cluster is not enough to represent the interest of the user, so it is discarded. Conversely, when the check-in ratio of a cluster is greater than or equal to c_num, the cluster is considered the interesting cluster of users. At the same time, we set an interesting weight for each considered cluster to measure the importance of the current cluster to the user at the current moment. The trajectory of the user u at a single moment is expressed as $T_u=\{(c_1^1,p_1^1;c_1^2,p_1^2;\ldots;c_1^k,p_1^k),\ldots,(c_n^1,p_n^1;c_n^2,p_n^2;\ldots;c_n^m,p_n^m)\}$, $c_j^i(1<i<k,0<j<23)$ is expressed as the i-th cluster that satisfies the screening conditions in the j-th period, p_j^i is the weight of the corresponding cluster, and its calculation formula is

$$p=\frac{\text{num}_\text{c}}{\text{num}_\text{all}} \tag{2}$$

where num_c is the number of positions in the current cluster, and num_all is the total number of positions in all valid interest clusters at the current moment.

4. Track similarity calculation

1) Single-interest track similarity

The single interest trajectory is of equal length, and there is one interest point at each moment. Therefore, when calculating the similarity of two interest trajectories, we use Euclidean distance to calculate the distance between two positions at the same time in the two trajectories respectively, then accumulate all the distance values, and finally calculate the average distance according to the number of periods. The calculation process is shown in (3).

$$\text{Sim\_tra}=\frac{\sum_{i=0}^{23}\text{ED}(P_i^{u1},P_i^{u2})}{24} \tag{3}$$

where ED represents the Euclidean distance calculation function, P_i^{u1} represents the i-th position of user u_1, and P_i^{u2} represents the i-th position of user u_2.

2) Multi-interest trajectory similarity

The multi-interest trajectory construction process shows that the multi-interest trajectory has multiple interest clusters in a single moment, as shown in Figure 2. The trajectory of user i at time t is a set of clusters denoted as $C_t=(c_t^1,p_t^1;c_t^2,p_t^2;\ldots;c_t^k,p_t^k)$, where (c_t^1,p_t^1) represents an interesting cluster and weight of the user at time t.

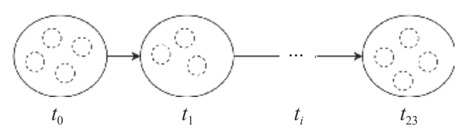

Figure 2 Multi-interest track.

When calculating the similarity of two users, the similarity of the two trajectories at a single moment must be calculated individually. Then the similarity of all moments is accumulated, and finally, the trajectory similarity of the two users is obtained.

We use the following method to calculate the trajectory similarity of users at a single moment: Given the trajectories C_t^{ui}, C_t^{uj} of two users at a certain moment. First, traverse the interest clusters of user i, find the closest interest clusters in the cluster set of user j respectively, calculate the cluster center distance, add the weights of the two clusters, and finally accumulate the weighted scores to obtain W_{ui}. Then traverse the interest clusters of user j, find the nearest interest clusters in the cluster set of user i, calculate the cluster center distance and add weights for calculation, and get W_{uj}. The final interest similarity at this moment is $\min(W_{ui}, W_{uj})$. Since the two-way friendship authentication is considered, the minimum value of the two similarities is taken as the final moment similarity of the two users. The similarity calculation process at the last t moment is as shown in (4) and (5).

$$W_{u1}=\sum_{i=0}^{k}(\min(\text{ED}(c_i^{u1},c^{u2}))\times p_i\times p_j) \tag{4}$$

$$\text{sim}_t=\min(W_{u1},W_{u2}) \tag{5}$$

where k and m are the numbers of clusters of the two users at this moment, respectively, $i \in (0, k)$, $j \in (0, m)$, c_i^{u1} is the i-th interest cluster of user u_1 at this moment, and c_j^{u2} is the j-th interest cluster of user u_2 at this moment. p^i, p^j are the weights corresponding to the clusters of interest, respectively.

After obtaining the similarity of user moments, the similarity of each moment is accumulated, divided by the number of moments, and finally, the trajectory similarity of two users is obtained. The calculation process is shown in the following equation:

$$\mathrm{Sim}_t = \frac{\sum_0^{24} \mathrm{sim}_t}{\mathrm{num\_}t} \tag{6}$$

where sim_t represents the similarity of two users at time t, and $\mathrm{num\_}t$ represents the total number of times, which is set to 24 here.

IV. Experiments

This section first introduces the data sets and parameter settings used in the experiments and compares the information of the original data and the filtered data sets. Then, by comparing the two proposed schemes with related research methods, explore the influence of different categories of data sets generated under different distance parameters in the sin_SCI scheme on the results of friendship reasoning, and the effect of different combinations of co-occurrence features in the mul_SCI scheme. Finally, the AUC score was used as the judging standard.

1. The data set

The experiments use two real datasets: Gowalla and Brightkite [6]. These two datasets are location-based social network datasets from SNAP Stanford and have been widely used in related research. Table 4 presents some statistics on the original and filtered datasets.

Table 4 Statistics of the dataset

Statistics	GA	G(0–20)	G(1–15)	BA	B(0–15)	B(1–50)
User	79k[1]	79k	79k	21k	21k	21k
Checkins	0.58M[2]	0.46M	0.34M	0.33M	0.20M	0.19M
Friends	362k	362k	359k	107k	102k	103k
Avg.Checkins	73	57	44	154	97	90
Location	779k	658K	555k	331k	220k	103k

Note: [1]k denotes thousand; [2]M denotes million.

In Table 4, User represents the number of users, Checkins indicate the number of check-ins, Friends indicate the number of friends, Avg.Checkins indicate the average number of check-ins, and Location indicates the number of locations. GA and BA represent the complete Gowalla and Brightkite datasets respectively. G(0–20) represents the Gowalla data set obtained by clustering screening when the range of distance parameter is set to [0, 20]. Similarly, B(0–15) represents the Brightkite data set obtained by clustering screening when the range of distance parameter is set to [0, 15].

The table reveals that Gowalla has approximately 4 times more users than Brightkite, yet its check-ins are only twice as many. Brightkite, on the other hand, exhibits a higher average check-in density. It's worth noting that the clustering screening algorithm concentrates on users' check-ins, preserving the total user count and friend count after screening.

2. Baseline and evaluation indicators

1) Baseline

As far as we know, under the same scenario, the latest research is SCI+ [12] and CIFEF [7]. Therefore, this paper takes these two methods as the baseline of this experiment.

SCI+ [12]: This model considers the co-location characteristics among users to speculate friendship and proposes four co-location characteristics: co-location diversity, location prevalence, duration of co-location, and stability. The diversity of co-location is based on the location entropy of co-location sites to help measure the coincidence of users' co-location. Location popularity is based on the global location entropy of a location to measure the popularity of the location in the global. co-location duration indicates the co-location duration between two users. Co-location stability indicates the stability of the duration of user co occurrence.

CIFEF [7]: The model uses embedding technology to learn the implicit weekday trajectory features and weekend trajectory features from the user check-in trajectory sequence, respectively. At the same time, a new explicit feature is proposed to capture the explicit information of co-occurring user pairs. This explicit feature measures the importance of each common location of user pairs by introducing the time interval of check-in into location entropy.

2) Model feature comparison

In order to more clearly present the differences between the model in this paper and the existing schemes, we list the features used in our scheme and the baseline scheme in Table 5. It should be noted that the first five features in the table can be explained in the baseline description and the last features are the features of this paper; Table 5 only distinguishes the differences of various models from the perspective of features, and does not involve the specific feature formation process of the models, for example, the interest track feature in our scheme is based on the clustering of multiple interest tracks.

Table 5 Feature comparison

Method	Diversity	Popular	Duration	Stability	Embedded track	Interest track
SCI+	✓	✓	✓	✓		
CIFEF		✓		✓	✓	
sin_SCI	✓	✓	✓	✓		✓
mul_SCI	✓	✓	✓			✓

3) Evaluation indicators

The model evaluation metrics used in this paper mainly include the following metrics.

- AUC: We use AUC score as the main evaluation index of friendship prediction model, which can be directly called by Scikit-learn [24]. AUC is calculated using the area of the receiver operating characteristic (ROC) curve. Higher AUC represents that the query retrieves less non-relevant data.
- Precision, Recall, and F1: These are commonly used evaluation indicators in classification models. Wherein accuracy represents the probability of actually positive samples among all predicted positive samples; The recall rate represents the probability that the actual positive samples are predicted to be positive samples. In practical application, F1 value is used to comprehensively evaluate the accuracy rate and recall rate. When both the accuracy rate and recall rate are high, F1 will also be high.

3. Cluster processing

For all the check-ins of each user in the initial data set, use MeanShift to perform clustering processing, and specify the algorithm parameter bandwidth, where bandwidth is the search radius during clustering. At the same time, set *bin_seeding* to True and set *bin_seeding* to True, which optimizes the initialization of the kernel position and speeds up the algorithm. The output result of the algorithm is the cluster center O of the classification cluster with decreasing density, where $O = [o_1, o_2, \ldots, o_n]$, where n is the number of clusters obtained by clustering, and o_1 is the cluster center with the highest density. It can be known that $O_m = o_1$. To more effectively divide all the check-ins of a single user in the initial data set, with O_m as the center, a series of r intervals are set to divide the data set. Then, the distance distribution of all check-ins to the largest cluster center in the two data sets is calculated to set the interval of r more reasonably, as shown in Figures 3 and 4, where the abscissa is the distance between a single check-in data and the center of the current check-in cluster of users, in-unit of meter. The ordinate is the proportion of the number of check-ins within a certain distance (i.e., the abscissa range) in the total number of check-ins.

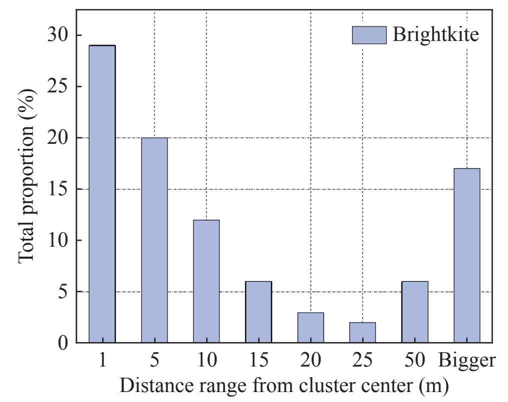

Figure 3 Brightkite check-in distance statistics.

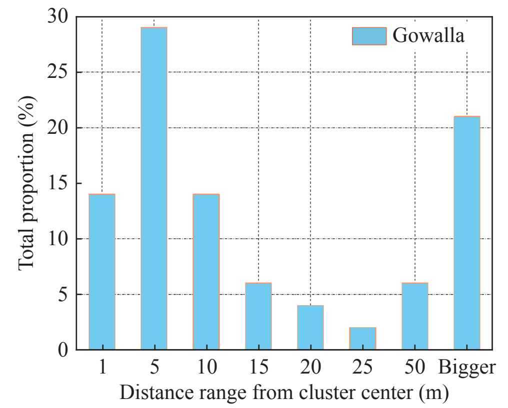

Figure 4 Gowalla check-in distance statistics.

According to the statistics in the above figure, it can be observed that starting from the cluster center, the check-in concentration in the Brightkite dataset is distributed near the largest cluster center. With the increase of the distance range, the number of check-ins decreases outward. The check-ins with a check-in distance of fewer than 50 m account for 83% of the total, and the check-ins less than 5 m account for 49% of the total. Half of the check-in data are concentrated within 5 m of the cluster center; In the Gowalla dataset, as the distance range increases, the number of check-ins increases first and then decreases. Check-ins less than 50 m account for 80% of the total, and check-ins less than 10 m account for 57% of the total. Over half of the check-ins are distributed within 10 m of the cluster center. It can be seen that the user check-in intensity of the two datasets is different. Specifically, the activity range of Gowalla users is wider than that of Brightkite. In general, starting from the cluster center, the number of check-ins in the two datasets gradually decreases as the distance increases, which also confirms the principle of MeanShift clustering, and the density gradually decreases from the cluster center to the outside.

4. Performance evaluation of the sin-SCI scheme

To achieve the purpose of screening out the effective check-in, after many experiments with different distance parameters, the final experiment set up 6 groups of r ranges to divide the data set, which are [0, 5], [0, 10], [0, 15], [0, 20], [0, 50], [1, 25], [1, 50]. In the experiment, SCI+ [12] was set as the baseline method. The experimental group using the trajectory similarity feature was set as a control experimental group to observe the contribution of the interest similarity feature to improving the performance of friendship reasoning. A series of experimental groups with different distance screening ranges were set up to observe the impact of check-in data in different ranges on friendship reasoning under check-in clustering. The experiments use AUC scores to evaluate the effect of friendship inference under different datasets. Table 6 shows the experimental results of the baseline method and the method based on clustering screening check-in.

Table 6 AUC scores from different data sets

Experimental group	BA	GA
SCI+	0.901	0.924
ori+simi	0.938	0.959
$0 < d < 5$	0.943	0.965
$0 < d < 10$	0.943	0.964
$0 < d < 15$	**0.944**	0.963
$0 < d < 20$	0.940	0.965
$0 < d < 50$	0.939	0.964
$1 < d < 25$	0.938	**0.970**
$1 < d < 50$	0.937	0.969

In Table 6, BA represents the Brightkite dataset, and GA also represents the Gowalla dataset. The leftmost column of the table represents different experimental groups, SCI+ represents the baseline grouping of the experiment SCI+ [12], original+simi is the experimental grouping that adds the trajectory similarity feature based on the baseline, and the others are experimental groupings based on the similarity of sin_SCI trajectories with different distance range filtering conditions. Data in bold in the table represents the highest AUC value achieved.

The observations of the experimental results are as follows. First, adding interest trajectory similarity does help friendship prediction. Under all data groups of the two datasets, after adding the similarity feature of interest trajectory, the friendship prediction effect of the model is improved. Among them, the effect of friendship reasoning is significantly improved on Brightkite, and the AUC score reaches 0.938, which is 4.1% higher than that of SCI+, and the AUC score is 3.7% higher on Gowalla. We speculate that this is due to the higher quality of the check-in data in Brightkite, thus forming a more complete trajectory, while in Gowalla. However, the number of check-ins is much larger than that of Brightkite, the check-in data is more scattered. The quality of check-in is far inferior to the former, so the composing trajectory is relatively vague, so it cannot improve the friendship inference effect between users, which is consistent with the conclusions obtained in Figures 3 and 4.

Secondly, on the whole, the results are consistent with expectations, and the results brought by the check-in data under different distance constraints are different. Specifically, in the Brightkite dataset, when r takes [0, 15], the AUC score of the BA group reaches 0.944, which is 4.7% higher than the baseline; In the Gowalla dataset, when r is taken as [1, 25], the GA grouping AUC score reaches 0.970, which is 4.9% higher than the baseline score. Overall all screening-based experimental group results are better than the baseline results.

5. Performance evaluation of mul-SCI scheme

Considering the results obtained in the previous part of the experiment, the two data sets obtained the best results under different screening distances. Although it is proved that there is indeed some valid check-in data that can improve the inference effect, this makes our proposed model not scalable; On the other hand, it can be seen from the previous results that the trajectory similarity is effective, so we then propose a mul_SCI trajectory construction scheme to strengthen the trajectory similarity feature, and we use multi-interest trajectory similarity for friendship inference and compare with existing SCI+ [12], CIFEF [7]. The result is shown in Table 7.

Table 7 AUC score under different methods

Method	Brightkite	Gowalla	Average
SCI+	0.901	0.924	0.9125
CIFEF	0.903	0.939	0.921
sin_SCI	0.944	0.965	0.9545
mul_SCI	0.965	0.975	0.97

The proposed mul_SCI scheme further strengthens the effectiveness of the trajectory similarity feature. This is because using the mul_SCI scheme considers more trajectory meanings at a single moment, enhancing the effectiveness of trajectory similarity, and the final effect is better than sin_SCI.

However, it can be seen from the multi-interest trajectory construction process that the mul_SCI scheme is very computationally complex. It considers all the contributing features and multiple interest clusters at a single moment of the trajectory, which brings a large overhead to the similarity calculation. Therefore, we conducted a separate reasoning test for all features to see their contribution to friendship reasoning and then selected the optimal combination to reduce the time overhead while maintaining a certain performance. The following experimental tests are carried out on the Gowalla dataset, and the experimental results are shown in the Figure 5.

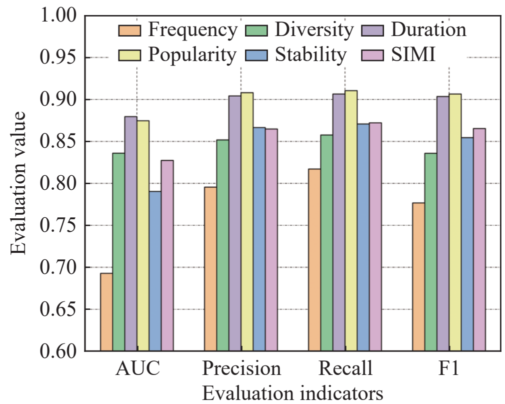

Figure 5 Single feature effect test.

As can be seen from the figure, under the indicators of precision, recall, and F1, the most effective features for distinguishing users are Duration and Population, followed by SIMI trajectory similarity feature. Under the AUC evaluation, SIMI ranks fourth, after the Diversity feature. Then we further test the inference effect of all features combined with SIMI to see the degree of interaction between co-occurrence features and SIMI. The results are shown in Figure 6.

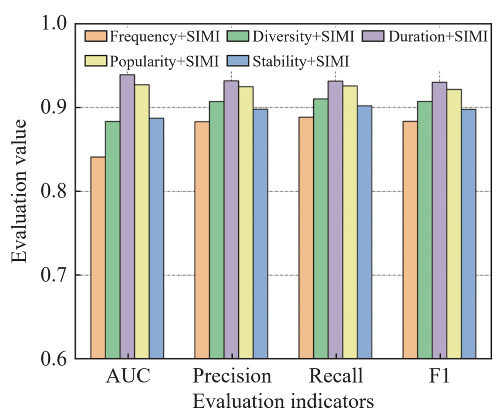

Figure 6 Feature combination effect.

As can be seen from the above figure, after Duration is combined with SIMI, its AUC value exceeds all combinations, followed by the Popularity+SIMI combination and the Diversity+SIMI combination. Therefore, we select the optimal Duration, Popularity, Diversity, and SIMI random combination for inference, the result is shown in Figure 7.

In the above figure, TD represents the Duration feature, P represents the Popularity feature, D represents the Diversity feature, SIMI represents the trajectory similarity feature, and mul_SCIP represents the combination of the previous four features. First of all, it can be seen that the optimal combination is the mul_SCI experimental group, followed by the mul_SCIP experimental group, followed by TD+P+SIMI, and finally, the combination of TD+D+SIMI and P+D+SIMI. The combination of SIMI multi-interest trajectory features and Duration, Popularity, and Diversity features we propose achieves the effect second only to the mul_SCI scheme. From the point of view of AUC, mul_SCIP is reduced by 0.06 relative to mul_SCI. However, compared with the CIFEF scheme, the numerical value has increased by 0.29, and other indicators are based on the same level as mul_SCI. Our scheme has reached the optimum.

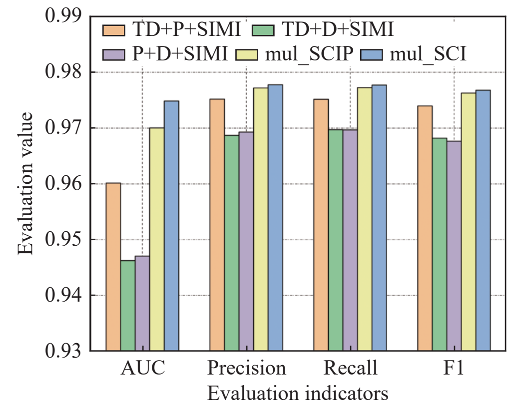

Figure 7 Combination effect comparison.

6. Time cost analysis

In this section, we test and analyze the time consumption of friendship inference between the proposed and comparison schemes on the two datasets. We take the time of the SCI+ [12] scheme as the basic time unit and then express the time consumption of other schemes respectively. The results are shown in Figure 8.

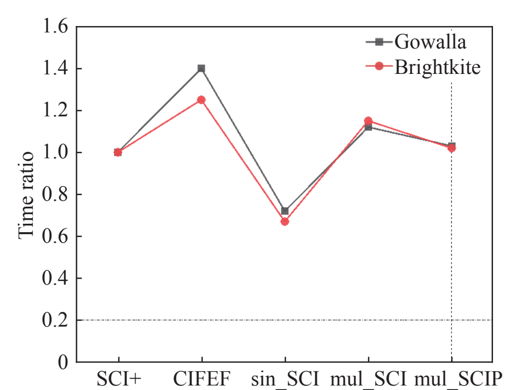

Figure 8 Time-consuming analysis.

It can be seen from Figure 8 that by signing in the sin_SCI-based sign-in screening scheme, the time overhead on the two datasets is reduced by 28% and 33% compared with SCI+ [12]. The mul_SCI scheme increases the time overhead of the two datasets by 12% and 21%, respectively, compared with SCI+ [12] due to the complexity of trajectory construction is higher than that of sin_SCI and no data screening is performed. Still, compared with CIFEF [7], the time overhead is reduced by 20% and 8% respectively. The final combined feature scheme mul_SCIP reduces the time overhead by 8% and 11% respectively compared with the mul_SCI scheme. We believe that there are two main reasons for this result. First, we reduced the size of the dataset through clustering. It can be seen from Table 4 in Section IV.1 that after screening, the number of sign ins in the two datasets has been reduced to varying degrees, which ultimately reduces the time spent on model processing; Secondly, we reduce the time consumption of feature calculation by reducing the number of co-occurrence features; The reason why the time overhead has not decreased significantly is that we use clustering to generate more clusters and calculate the similarity of interest clusters in these clusters, which will undoubtedly increase the time overhead of the model. In general, the purpose of maintaining the performance and reducing the time overhead is achieved. (Note that the mul_SCIP here is the optimization of the mul_SCI scheme. All mul_SCI mentioned in the following refers to the mul_SCIP grouping here.)

In general, our clustering and filtering of the dataset based on single-interest trajectories can indeed reduce the time overhead of the prediction algorithm; while using multi-interest trajectories to improve performance leads to an increase in time overhead. Then, further reducing the number of co-occurring features reduces the time overhead to a certain extent.

V. Discussion

In this section, the sensitivity of the sin_SCI scheme to the clustering bandwidth parameter bandwidth is explored, followed by the discussion of the influence of the c_num parameter on the mul_SCI scheme, and then some topics related to but beyond the scope of this study are discussed.

First, the experiment analyzes the degree of influence of the size of the MeanShift clustering parameter bandwidth on the dataset of interest. Several bandwidth sizes are set in the check-in clustering: {0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}. In Figure 9, it can be observed that sin_SCI achieves the best performance when the bandwidth size is set to 0.6 on the Gowalla dataset and 0.4 on the Brightkite dataset. With the bandwidth size set to 0.5, this seems to be a good balance of the two datasets.

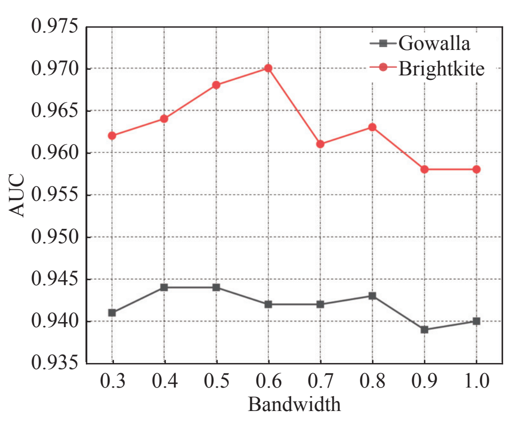

Figure 9 AUC under different clustering bandwidths.

Secondly, in order to explore the impact of parameter c_num on friendship inference based on multiple interest trajectories, we set a series of meaningful c_num values, $c\_num = [0.1, 0.2, 0.3, 0.4, 0.5]$. When $c\_num = 0.1$, it means that as long as the number of positions in a cluster reaches one-tenth of the current moment, it can be determined as a valid cluster, and there are at most ten valid interest clusters at a moment. When $c\_num = 0.5$, it means that only when the number of check-ins in the cluster accounts for half of the total number of check-ins at a time, it will be determined as a valid cluster. Note that in this case, there are at most two such clusters at a time. Finally, our test results are shown in Figure 10.

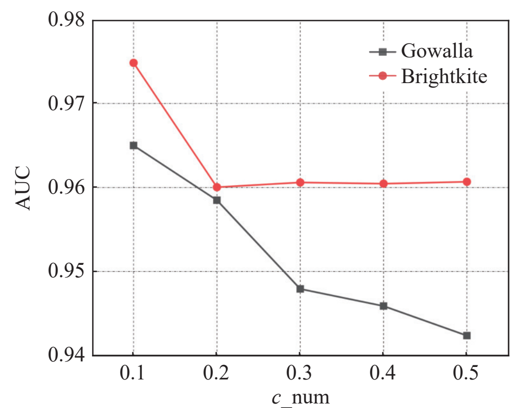

Figure 10 AUC under different c_num.

With the increase of c_num, AUC generally shows a downward trend. This is because the larger c_num is, the fewer valid clusters there are in a single trajectory moment, and the less useful information the trajectory represents, thus causing performance degradation. Specifically, there has been a downward trend in the Brightkite dataset, while the Gowalla dataset is a flat curve after 0.2. We speculate that this is because the check-ins in the Brightkite dataset are all centrally distributed, making clusters of different magnitudes distinct, however, in Gowalla, due to the similarity in cluster size, the number of interest clusters changes less when c_num is greater than 0.2, ultimately leading to a flat trend in AUC.

In addition, as the collection of personal location information becomes more convenient, there is also a risk of user privacy leakage. At present, there are many mature schemes for location privacy protection, such as the differential privacy-based location protection scheme [25], [26], and the k-anonymity-based location protection scheme [27]. When the location of the user cannot be accurately obtained, the real interest location and trajectory of users cannot be obtained, which increases the difficulty of inferring user similarity. Therefore, the focus of future defense should continue to be placed on the protection of the location of the user, and at the same time, attention should be paid to the convenience of the user to obtain services.

VI. Conclusions

In this paper, friendship inference based on interest check-in data and interest trajectory similarity in location social networks is studied. The proposed ITSIC model uses the Meanshift clustering algorithm to cluster user check-in data and tests the performance of friendship reasoning under two schemes, sin_SCI, and mul_SCI. This paper verifies the effectiveness of the trajectory interest similarity feature and explores the performance of prediction based on different check-in data under a single interest trajectory. At the same time, it further mines the interesting trajectory of users and proposes a multi-interest trajectory construction scheme. Co-occurrence features are combined to select the combination with the best performance. Finally, extensive experiments are carried out on two public datasets. The experimental results show that the mul_SCI scheme of the ITSIC model outperforms existing methods in inference and reduces the time overhead of friendship inference.

For future work, we plan to design privacy protection algorithms according to the characteristics of users' check-in data in different periods to ensure data distribution and limit users' privacy leakage to a certain extent.

Acknowledgements

This work was supported by the Natural Science Foundation of Hebei Province (Grant No. F2021201058).

References

[1] E. Cho, S. A. Myers, and J. Leskovec, "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*, San Diego, CA, USA, pp.1082–1090, 2011

[2] G. Kossinets and D. Watts, "Origins of homophily in an evolving social network," *American Journal of Sociology*, vol. 115, no. 2, pp. 405–450, 2009.

[3] D. Z. Ding, M. Zhang, S. Y. Li, *et al.*, "BayDNN: Friend recommendation with Bayesian personalized ranking deep neural network," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, Singapore, pp.1479–1488, 2017.

[4] S. C. Peng, A. M. Yang, L. H. Cao, *et al.*, "Social influence modeling using information theory in mobile social networks," *Information Sciences*, vol. 379, pp. 146–159, 2017.

[5] Z. H. Hu, C. Peng, C. He, *et al.*, "IO-aware factorization machine for user response prediction," in *Proceedings of 2020 International Joint Conference on Neural Networks*, Glasgow, UK, pp.1–8, 2020.

[6] J. Leskovec and A. Krevl, "SNAP datasets: Stanford large network dataset collection," Available at: http://snap.stanford.edu/data, 2014.

[7] C. He, C. Peng, N. Li, *et al.*, "CIFEF: Combining implicit and explicit features for friendship inference in location-based social networks, " in *Proceedings of the 13th International Conference on Knowledge Science, Engineering and Management*, Hangzhou, China, pp. 168–180, 2020.

[8] Y. Z. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[9] H. Pham, C. Shahabi, and Y. Liu, "EBM: An entropy-based model to infer social strength from spatiotemporal data," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, pp.265–276, 2013.

[10] H. J. Wang, Z. H. Li, and W. C. Lee, "PGT: Measuring mobility relationship using personal, global and temporal factors," in *Proceedings of 2014 IEEE International Conference on Data Mining*, Shenzhen, China, pp.570–579, 2014.

[11] G. S. Njoo, M. C. Kao, K. W. Hsu, *et al.*, "Exploring check-in data to infer social ties in location based social networks," in *Proceedings of the 21st Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Jeju, South Korea, pp.460–471, 2017.

[12] G. S. Njoo, K. W. Hsu, and W. C. Peng, "Distinguishing friends from strangers in location-based social networks using co-location," *Pervasive and Mobile Computing*, vol. 50, pp. 114–123, 2018.

[13] C. He, C. Peng, N. Li, *et al.*, "Exploiting spatiotemporal features to infer friendship in location-based social networks," in *Proceedings of the 15th Pacific Rim International Conference on Artificial Intelligence*, Nanjing, China, pp.395–403, 2018.

[14] A. E. Bayrak and F. Polat, "Mining individual features to enhance link prediction efficiency in location based social networks," in *Proceedings of 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Barcelona, Spain, pp.920–925, 2018.

[15] P. H. Wang, F. Y. Sun, D. Wang, *et al.*, "Predicting attributes and friends of mobile users from AP-Trajectories," *Information Sciences*, vol. 463-464, pp. 110–128, 2018.

[16] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, USA, pp.1046–1054, 2011.

[17] M. Backes, M. Humbert, J. Pang, *et al.*, "Walk2friends: Inferring social links from mobility profiles," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas, TX, USA, pp.1943–1957, 2017.

[18] Q. Gao, G. Trajcevski, F. Zhou, *et al.*, "Trajectory-based social circle inference," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle, WA, USA, pp.369–378, 2018.

[19] F. Zhou, B. Y. Wu, Y. Yang, *et al.*, "Vec2Link: Unifying heterogeneous data for social link prediction," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Torino, Italy, pp.1843–1846, 2018.

[20] L. F. Ren, R. M. Hu, D. S. Li, *et al.*, "Cross-regional friendship inference via category-aware multi-bipartite graph embedding," in *Proceedings of 2022 IEEE 47th Conference on Local Computer Networks*, Edmonton, Canada, pp.73–80, 2022.

[21] Q. Gao, F. Zhou, X. Yang, *et al.*, "When friendship meets sequential human check-ins: Inferring social circles with variational mobility," *Neurocomputing*, vol. 518, pp. 174–189,

2023.

[22] J. Li, F. Z. Zeng, Z. Xiao, *et al.*, "Drive2friends: Inferring social relationships from individual vehicle mobility data," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5116–5127, 2020.

[23] J. Li, F. Z. Zeng, Z. Xiao, *et al.*, "Social relationship inference over private vehicle mobility data," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5221–5233, 2021.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] J. D. Zhang and C. Y. Chow, "Enabling probabilistic differential privacy protection for location recommendations," *IEEE Transactions on Services Computing*, vol. 14, no. 2, pp. 426–440, 2021.

[26] J. Wang, F. Wang, and H. T. Li, "Differential privacy location protection scheme based on Hilbert curve," *Security and Communication Networks*, vol. 2021, article no. 5574415, 2021.

[27] H. T. Li, L. X. Gong, B. Wang, *et al.*, "*k*-anonymity based location data query privacy protection method in mobile social networks," in *Proceedings of 2020 International Conference on Networking and Network Applications*, Haikou City, China, pp.326–334, 2020.

Junfeng TIAN received the B.S. degree in radio from Hebei University in 1986, M.S. degree in educational technology from Hebei University in 1994, and the Ph.D. degree in computer science and technology from the University of Science and Technology of China in 2005. He has been engaged in teaching and research in the fields of distributed computing and network technology for many years. Currently, he is a Professor, and Deputy Secretary General of Open System Specialized Committee of China Computer Federation; Young and Middle-aged Specialist with Outstanding Contribution in Hebei Province; Young and Middle-aged Teacher with Outstanding Contribution in Hebei Province. (Email: tjf@hbu.edu.cn)

Zhengqi HOU received the B.E. degree in information security from School of Cyberspace Security and Computer, Hebei University, China, in 2020, and is currently studying for the M.S. degree in cyberspace security and computer at Hebei University. His main research interests include information security and social network location privacy security. (Email: hozi_nl@hotmail.com)