Yin et al. EURASIP Journal on Wireless Communications and Networking (2019) 2019:266
https://doi.org/10.1186/s13638-019-1606-y

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Location recommendation privacy protection method based on location sensitivity division

Chunyong Yin[1], Xiaokang Ju[1], Zhichao Yin[2] and Jin Wang[3*]

Abstract

Location-based recommendation services can provide users with convenient services, but this requires monitoring and collecting a large amount of location information. In order to prevent location information from being leaked after monitoring and collection, location privacy must be effectively protected. Therefore, this paper proposes a privacy protection method based on location sensitivity for location recommendation. This method uses location trajectories and check-in frequencies to set a threshold so as to classify location sensitivity levels. The corresponding privacy budget is then assigned based on the sensitivity to add Laplace noise that satisfies the differential privacy. Experimental results show that this method can effectively protect the user's location privacy and reduce the impact of differential privacy noise on service quality.

Keywords: Location information, Location recommendation, Differential privacy, Sensitivity, Laplace noise

1 Introduction

The development of mobile communication network provides users with a more colorful mobile network service platform, which enables users to obtain and push network information resources anytime and anywhere. This makes it possible to provide users with ubiquitous mobile network services. In particular, the rise of mobile social networks has greatly helped users in network information services [1]. At the same time, since it is necessary to collect a large amount of information from users while enjoying network services, how to ensure the data security of supervisory control and data acquisition (SCADA) system is worthy of attention. We must first ensure that the user's data is secure and then consider how to use this data to find the information resources that users are really interested in, so as to meet the personalized needs of mobile users [2].

As one of the solutions for personalized information services, the recommendation system has attracted wide attention in both industry and academia. Compared with the traditional search engine, the recommendation system not only pays attention to the relationship and ordering between search results but also focuses on the influence of the user's personalized preference model on the search results. In addition, the successful introduction of pervasive computing theory makes the traditional recommendation systems not only focus on the "user-project" binary relationship but also consider the context information of the user together to form a "context-user-project" system. This enables the system to automatically discover and utilize contextual information to meet the user's personalized information needs that change as contextual information changes [3]. For example, users are more willing to watch their favorite movies on commuting buses instead of at the office. Compared with the working office, users are more willing to know about the surrounding promotional advertisements in the leisure and entertainment plaza after work. This aspect truly satisfies the user experience and improves user satisfaction. On the other hand, the adaptability of the system and the accuracy of the recommendation are enhanced [4].

In recent years, with the further development of the mobile Internet and the maturation of geographical positioning technology, more and more scholars have begun to pay attention to the influence of location characteristics on the mobile users' cognition and behavior in the

* Correspondence: jinwang@csust.edu.cn
[3]School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, China
Full list of author information is available at the end of the article

context of the mobile Internet. At the same time, when researchers analyze the behavior of mobile users, the personal privacy protection of users becomes more and more important.

Most recommended services focus on social trust between users [5], while mobile location-based recommendations are mainly to send a location service request to the server through the user's operation so as to provide the user with personalized service. Location recommendation combines the user's interest preferences and geographic location and then analyzes the content of the service. Finally, the server returns the service information required by the user. In location-based recommendation, since the server is to provide personalized services to users, it will store a large number of users' evaluation and recommendation information. The server then mines hidden and useful information from it so that the requesting user can get valid recommendations. However, location-based recommendation services provide users with convenience and also pose a threat of privacy leakage. The service provider can collect sensitive information (such as location or point of interest (POI)) in the service request sent by the user through SCADA to obtain and infer more private information of the user. For example, service providers can analyze their travel patterns and predict their locations in the future by extracting the user's location characteristics [6]. In addition, the service provider can use the statistical data of the user's location information in certain time periods to infer user's home address and unit [7]. The leakage of the private information may cause the user to have an unpredictable loss. Therefore, how to ensure user data security in SCADA has become a problem that needs to be addressed in location-based recommendation services.

This paper proposes a location data security protection scheme under SCADA for user behavior patterns in location recommendation. The enterprise can grasp the user's behavioral regularity by analyzing the user's location trajectory so as to recommend the next location which the user may be interested in. Then the enterprise sends more targeted recommended information based on the distance information of the location where the mobile user is located or the next location where the user is likely to be interested. Therefore, we need to provide effective services while protecting the user's location information. On the one hand, the current researches add uniform noise to the user's location data, but this will cause excessive noise addition at some locations to reduce the quality of service. On the other hand, researchers add noise to certain sensitive locations, but this scheme ignores the user's interest preferences and does not meet the user's personalized privacy needs. In order to solve the above problems, this paper makes the following main contributions:

(1) We define the sensitivity level for the user's location based on check-in frequencies. When the number of check-in times reaches the threshold set in this paper, the sensitivity level of the location will change. Different levels of sensitivity indicate that the user has a different degree of preference for the location.
(2) We use the prefix tree structure to represent the user's location trajectory information. This structure shows the check-in statistics and the sensitivity level of the user's locations.
(3) According to the structure of the prefix tree, we propose a privacy budget allocation method based on location sensitivity. This paper assigns a corresponding privacy budget based on the sensitivity level of each location and then adds the corresponding noise. Through the experimental results in the real data set, we prove that this scheme can not only avoid adding too much noise but also meet the user's personalized privacy needs.

Section 2 of this paper compares domestic and foreign research status. Section 3 lists the relevant definitions of differential privacy and location recommendations. Section 4 describes a privacy protection method based on location sensitivity for location recommendation proposed in this paper. Section 5 analyzes the experimental results. Section 6 summarizes the full paper.

2 Related work

Implementing data privacy protection before data release has become a concern for researchers, individuals, and service providers [8]. In recent years, privacy protection for mobile locations has mostly been implemented on the basis of k-anonymity. Researchers usually cluster the movement trajectories first, and then they generalize the cluster group or limit its feature release [9]. In 2003, Gruteser et al. applied k-anonymity to location privacy protection for the first time [10]. The algorithm uses a quad-tree search method to construct an anonymous region that satisfies k-anonymity and whose area is not less than a certain value. Experiments show that this method makes it impossible for an attacker to effectively identify the user's real location. Chunyong Yin proposed an improved K-value location privacy protection method based on privacy level, which combines the k-anonymity method with the kana method [11]. The improved method can more reasonably select the K-value to meet different privacy level requirements. Miura proposed a hybrid method which introduced false nodes and anonymous areas to protect location privacy [12]. When the number of mobile users in the environment surrounding the user does not meet the anonymous demand, an anonymous area meeting the requirements is

constructed by creating fake nodes. The number of false nodes dynamically adjusts according to the number of users in the surrounding environment. Although the approach based on false nodes has various advantages in terms of implementation and the computational cost is low, when users submit continuous requests, they suffer from temporal and spatial correlation problems. To solve this problem, Nosouhi proposed a practical hybrid location privacy protection scheme [13]. The proposed method filters out relevant false location data before submission. Therefore, the attacker cannot identify the user's real location. Since privacy protection in the era of big data is more difficult than traditional information protection, Zhang Sun et al. proposed an improved model that combines k-anonymity with L-diversity. The *K*-member clustering algorithm can be used to transform the anonymity problem into a clustering problem to achieve an improved anonymity model. Improved anonymous models can reduce algorithm execution time and information loss, which is especially important for big data [14]. Most existing anonymous methods directly delete trajectories or locations that violate specific constraints, resulting in a large amount of information being lost. In response to this problem, Chen et al. proposed a trajectory privacy protection method based on 3D mesh partitioning [15]. This method first divides the trajectory area into a number of spatiotemporal units (represented as 3D units) and then performs location swapping or suppression in each spatiotemporal unit. Compared with other methods, this algorithm effectively preserves the trajectory data privacy and improves the availability of the data.

In addition to the above methods, differential privacy has also achieved remarkable results in location protection research in recent years. Differential privacy is a new privacy protection model proposed by Dwork in 2006 [16]. This model is based on data distortion. This method can solve two major defects of the traditional privacy protection model. One is that it defines a fairly strict attack model. It does not care how much background the attacker has. Even if the attacker has mastered all record information except a certain record, the privacy of the record cannot be disclosed. The other is that it gives a rigorous definition and quantitative assessment of the level of privacy protection. Implementing differential privacy mainly considers the following two issues. The first is to design algorithms that satisfy differential privacy to ensure the privacy security. The second is how to reduce errors caused by data distortion to improve data availability. Wang et al. combined the concepts of differential privacy and k-anonymity to propose a differential private k-anonymous concept (DPkA) for LBS query privacy [17]. They also proposed an algorithm to implement DPkA with a minimized ε. Experiments show that DPkA effectively improves the level of privacy protection while ensuring query efficiency. Chunyong Yin et al. proposed a location privacy protection method that satisfies differential privacy constraints to protect location data privacy [18]. First of all, they built a multilevel location information tree model and then selected data according to the tree node access frequency. Finally, the Laplace mechanism was used to add noise to the access frequency of the selected data. Experiments show that this scheme can effectively blur the access frequency of sensitive locations and maintain high data availability. In order to solve the problem that the traditional grid noise adding method leads to high error, Zhou proposed a differential privacy noise dynamic allocation algorithm based on the standard deviation circle radius (called SDC-DP algorithm) [19]. The strength of the privacy protection requirements of the SDC-DP algorithm is defined by the divergence of each grid. For different privacy protection requirements, the SDC-DP algorithm dynamically adds noise of different scale to the count query results of each grid. Experiments show that the SDC-DP algorithm effectively reduces the relative error and improves the query accuracy. In addition, in order to protect real-time trajectory data, Ma et al. proposed a privacy protection mechanism based on differential privacy called RPTR [20]. RPTR uses an aggregate Kalman filter based on the user-location transfer probability matrix to ensure data availability. In addition, they have built a privacy budget allocation method based on regional privacy weights to provide better protection for areas with high user density. Experiments show that RPTR cannot only protect the privacy of real-time trajectory data but also ensure the availability of data.

With the popularity of recommendation systems, location recommendation services are playing an increasingly important role in people's daily lives. At the same time, the issue of location privacy protection in location recommendation services has also received increasing attention. Hao proposed a differential privacy trajectory analysis algorithm [21]. The algorithm first converts the original trajectory data set into a bipartite graph and then extracts the correlation matrix representing the bipartite graph to inject carefully calibrated noise to satisfy the difference privacy. A large number of experiments on real trajectory data sets show that the algorithm demonstrates high recommendation accuracy while meeting the required differential privacy guarantees. Polatidis proposes a multilevel privacy protection method for collaborative filtering systems that perturbs each rating before submitting it to the server [22]. The perturbation method is based on multiple levels of each level and random values of different ranges. Before we submit each rating, the privacy level and perturbation range are randomly selected from a fixed level of privacy. The results show that this scheme can provide different levels of

privacy protection and can achieve a more satisfactory recommendation. Xue proposed a new destination prediction method specifically for registration services for geographic social networks [23]. This method not only solves the problem of data sparsity faced by common destination prediction methods but also utilizes common background information such as social structure, road network and speed limit. The experimental results show that the destination prediction method has strong predictive ability and has effective protection against destination inference attacks. Zhang S proposed an enhanced user privacy scheme through caching and spatial-anonymity (CSKA) in continuous LBS [24]. It uses multilevel caching to reduce the risk of user information being exposed to untrusted location service providers. Simulation results show that the CSKA scheme can provide higher privacy protection than the previous methods and can minimize the overhead of the LBS server. In order to improve the availability of data, Wei proposed a trajectory community recommendation (DPTCR) scheme based on differential privacy [25]. First, DPTCR converts the position of the real trajectory into a noise feature location based on a proprietary semantic expectation method. Second, DPTCR uses a proprietary geographic distance approach to construct a noise trajectory that has the smallest geographic distance from the actual trajectory. DPTCR ensures that the real trajectory is highly similar to the constructed noise trajectory. Zhang et al. proposed a new Privacy-preserving LOcation REcommendation framework [26]. They use n-order additive Markov chains to use the user's sequence pattern for location recommendation. In addition, they designed a probabilistic differential privacy mechanism to effectively protect the user's location privacy. This solution addresses two key challenges of recommendation accuracy and location privacy caused by high sensitivity and small counting issues in a personalized and fine-grained location recommendation environment.

However, the above privacy protection methods for the location recommendation service ignore the user's interest preference, resulting in a failure to more rationally allocate the privacy budget and add noise. Therefore, this paper proposes a privacy protection method based on location sensitivity for location recommendation.

3 Preliminary knowledge

3.1 Related definitions of difference privacy

3.1.1 Definition 1 (differential privacy)

There are two data sets$D1$and$D2$. The difference between them is at most one record. Range (K) represents the range of a random function (algorithm) K. $P_r[Es]$ represents the disclosure risk of event Es. If the random function (algorithm) K satisfies Eq. (1) for any query result S Range (K), the algorithm K satisfies the ε-difference privacy protection, where the parameter ε is the privacy protection budget [27]. In practice, ε usually takes a small value, such as 0.01, 0.1, or ln2. If the value of ε is smaller, it means that the level of privacy protection is higher.

$$P_r(K(D_1)\in S)\le e^{\varepsilon}P_r(K(D_2)\in S) \tag{1}$$

The definition shows that for any possible outcome of the algorithm output, the probability ratio of the algorithm to the same result on data sets$D1$and$D2$is a constantexp(ε) less than ε.

As shown in Fig. 1, algorithm K randomizes the output to achieve the effect of privacy protection. The parameter ε quantifies the probability of outputting the same result when any record is deleted or added in the data set.

3.1.2 Definition 2 (global sensitivity)

Suppose there is a function$f: D \to R^d$. The input is a data set and the output is a d-dimensional real number vector. For any adjacent data set$D$andD', if it satisfies Eq. (2), it is called the global sensitivity of function f, where$\|f(D) - f(D')\|_1$ is the one-order norm distance between and$f(D)$and$f(D')$.

$$GS_f = \max_{D,D'}\|f(D)-f(D')\|_1 \tag{2}$$

The global sensitivity of a function is determined by the function itself, and different functions will have different global sensitivities. Some functions have less global sensitivity, so only a small amount of noise can be added to mask the effect of a record being deleted on the query results.

3.1.3 Definition 3 (Laplace mechanism)

In practical applications, we usually use the Laplace mechanism for the protection of location privacy [28]. The idea of Laplace mechanism is to satisfy ε-differential privacy protection by adding random noise to the query

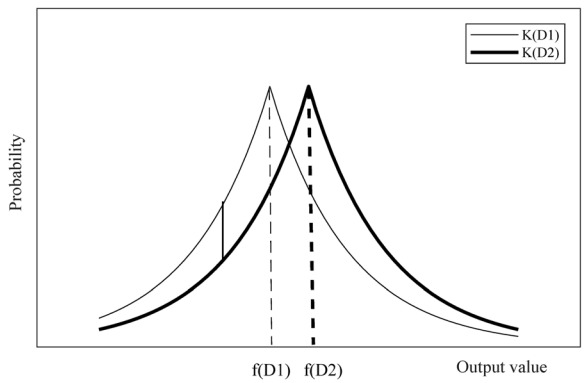

Fig. 1 Random algorithm output probabilities on neighboring data sets

result$f(D)$. And the noise is subject to the Laplace distribution.

We give the data set D and assume that there is a function$f: D \to R^d$ with a sensitivity of Δf. Then the random algorithm$M(D) = f(D) + Y$ provides ε-differential privacy protection where$Y \sim Lap(\Delta f/\varepsilon)$ is random noise and obeys the Laplace distribution with scale parameter $\Delta f/\varepsilon$.

$$P(x) = \frac{1}{2b} \exp\left(-\frac{|x-u|}{b}\right) \tag{3}$$

When $u = 0$, the probability density of Laplace under different scale parameters is shown in Fig. 2. As can be seen in the Fig. 2, the scale parameter increases, and the probability distribution becomes more uniform. On the contrary, the scale parameter decreases and the probability distribution are more concentrated.

The amount of noise added is positively related to the size of $\Delta f$and negatively related to ε. When $\Delta f$is small, less noise is added and the algorithm shows better effect.

3.2 Related definitions based on location recommendation

At present, scholars have conducted extensive research on location recommendation. Recommendations based on collaborative filtering are also often used for location recommendations [29]. The collaborative filtering method uses the user's rating matrix for the project to calculate the similarity and compares the similarity to select the most similar set of neighbors. It then builds a user-location scoring matrix and calculates the similarity between user locations. In general, the collaborative filtering matrix is based directly on the user's scoring value for the project. For example, in movie recommendations, users have an intuitive score rating for movies they have watched. However, such explicit scoring evaluation is not included in the user's historical location information. Therefore, in the location recommendation based on collaborative filtering, a value of interest that can reflect the location of the user is often extracted according to the location information of the user, and this is used as an implicit rating evaluation of the location by the user. For example, the frequency of user's appearance in a certain location reflects the user's interest in this location. Therefore, we can predict the user's next action by analyzing the frequency characteristics of the user's trajectory data [30].

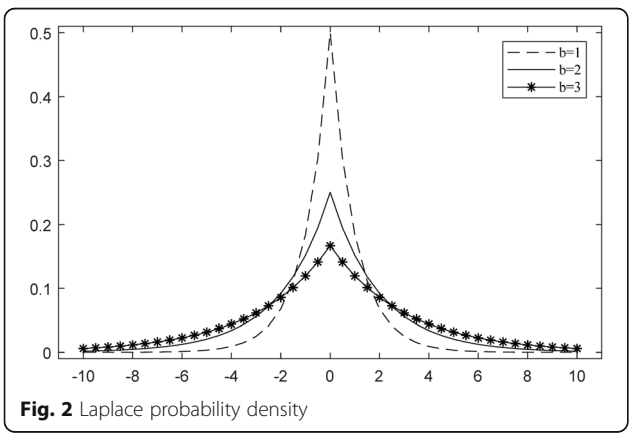

Fig. 2 Laplace probability density

3.2.1 Definition 4

All the set of check-in locations of user u are $SL(u) = <l1, l2, \ldots ln>$, where the number of check-in times of the user at the location $li(1 \le i \le n)$isαi.

3.2.2 Definition 5

The set of all users checked in at location l is $Su(l) = <u_1, u_2, \ldots u_m>$, where the number of check-in times at the location l of the user $ui(1 \le i \le m)$ is β_i.

3.2.3 Definition 6

The user u scores the location in a way that combines the number of check-in times and the weighting of the sensitivity. As shown in Eq. (4):

$$Score(u, l) = \alpha_l \times w_l \tag{4}$$

$\overline{Score}(u, l)$ represents the user's average score for all locations. Equation (5) is used to calculate the location similarity$sim(u_i, u_j)$ of two users u_i and u_j.

$$sim(u_i, u_j) = \frac{\sum_{l \in S_{L(u_i, u_j)}} \left(Score(u_i, l) - \overline{Score}(u_i, l)\right)\left(Score(u_j, l) - \overline{Score}(u_j, l)\right)}{\sqrt{\sum_{l \in S_{L(u_i)}} \left(Score(u_i, l) - \overline{Score}(u_i, l)\right)^2} \sqrt{\sum_{l \in S_{L(u_j)}} \left(Score(u_j, l) - \overline{Score}(u_j, l)\right)^2}} \tag{5}$$

4 Methods

Location-based recommendations calculate location similarity based on the user's rating of the location. The rating is based on the user's set of check-in locations, which includes features such as trajectory sequence, check-in times, and location sensitivity. In order to intuitively reflect these features, this paper considers the use of a prefix tree to organize the user's check-in location information.

Differential privacy protection primarily adds noise based on privacy budget allocation results. The random allocation of privacy budgets will distort the user's original trajectory sequence and location sensitivity features, thereby affecting the location similarity calculation. Therefore, in order to reduce the errors caused by noise, this paper uses the method based on the location

sensitivity to allocate the privacy budget. The overall flow chart of this scheme is shown in the Fig. 3.

As shown in the flow chart, we use the prefix tree to represent the location trajectory and check-in times in the location information after collecting the location data. Then we divide sensitivity levels for locations by check-in frequency. The corresponding privacy budget is assigned according to different sensitivity levels. Then we add the corresponding Laplace noise to the check-in statistics according to the assigned privacy budget. Finally, we use the noisy location data for location recommendation.

4.1 Prefix tree structure

The basic unit of location information is location point information. It represents the location information of a user at a certain time. Although the user's location points are discrete in geospatial, the discrete locations in the space can be effectively connected according to the order of time points. They are then combined into a sequence of continuous linear trajectories that describes the location information of the user over a certain period of time. We abstract the user's all trajectories into a prefix tree whose root node is root. Each node in the prefix tree describes a location, and each branch represents a check-in trajectory. All of the user's trajectories start from root and merge the same subtracks. Then we count the number of location check-in times and use this to classify the sensitivity levels. Table 1 shows the trajectory sequence owned by the user u.

We create a prefix tree from the user u's trajectory sequence and then merge the subtracks. After the establishment of the tree, we need to calculate the number of check-in times for each tree node and assign a sensitivity level to the node. Figure 4 shows the prefix tree structure for user u.

As shown in Fig. 4, the first parameter in parentheses indicates the number of statistics for this location on the trace. The second parameter indicates the sensitivity level of this location on this trajectory. The number of visits to the location point reflects the user's level of interest in this location. The more users visit this location, the more interested the user is in this location. It is not difficult to find from Fig. 4 that user u has the greatest interest in location l_3. And we can see that the prefix tree path structure keeps the chronological order of the user's original location from top to bottom. At the same time, it visually and clearly reflects the user's trajectory law and the frequency characteristics of the passing location. In this paper, the height of the prefix tree structure is determined by the maximum length of the trajectory sequence. Since the abstract root node does not belong to any real trajectory sequence, the height of the prefix tree in Fig. 4 is 4. The first layer of the structure indicates that the user's first passing location is l_1 or l_3.Moreover, the substructure of the prefix tree also represents a collection of all trajectory sequences with the same prefix subsequence.

4.2 Privacy budget allocation plan based on location sensitivity

The traditional privacy budget allocation method uses a uniform distribution strategy. However, this allocation strategy can result in some of the privacy budget being wasted. If the substructure of the tree is highly unbalanced, it can lead to a tilt in the distribution of privacy budgets. And it does not take into account the check-in features and sensitivity of the location. However, the recommendation system believes that users with the same hobbies have certain similarities in behavior patterns. For example, if some users have similar interests in a location, then there is a certain similarity in the recorded location trajectory sequence. In order to maintain this similarity, this paper proposes a privacy budget allocation method based on location sensitivity. We first redefine the location sensitivity and location weight.

4.2.1 Definition 7 (location sensitivity)

The number of check-in times of user u at location l can indicate the user's degree of preference for this location.

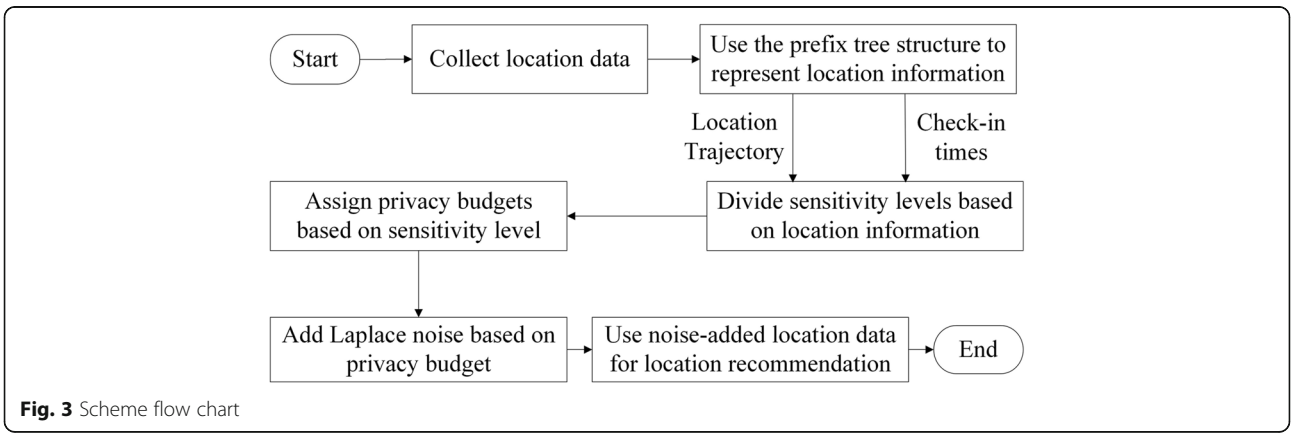

Fig. 3 Scheme flow chart

Table 1 The trajectory sequences of the user u

Sequence	Track path
T_1	l_1-l_2-l_3
T_2	l_1-l_2
T_3	l_3-l_2-l_1
T_4	l_1-l_2-l_3
T_5	l_3-l_1-l_4
T_6	l_3-l_2
T_7	l_3-l_1-l_4
T_8	l_3-l_1-l_4-l_5
T_8	l_1-l_2
T_{10}	l_3-l_2-l_1

The more the user checks in, the higher the user prefers to this location. The attacker can easily grasp the user's preference by counting the number of check-in times for a certain location. Therefore, the user's privacy is vulnerable to leakage. In order to solve this problem, this paper defines the sensitivity level for the user's check-in location. If the user has more check-in times at a certain location, the sensitivity level of this location is higher.

We define the location sensitivity level as $p_r = r(r = 1, 2, \ldots, n)$. The sensitivity level of location is determined by the number of check-in times of user u at this location. The more users check in at this location, the more sensitive this location is. This paper sets a threshold θ for the number of check-in times. When the number of check-in times reaches the threshold, the sensitivity level of this location will change. For example, we assume that the sensitivity level is the lowest when the number of check-in times is less than 50. Then we assume that the threshold interval is 50, and when the number of check-in times is between 50 and 100, the sensitivity level is increased by one level. By analogy, when the number of check-in times exceeds 200, the sensitivity level is the highest. In this paper, we define the highest sensitivity level when $p_r = 1$. As the value of p_r increases, the level of location sensitivity decreases. The reasons for this definition will be explained at the end of this section.

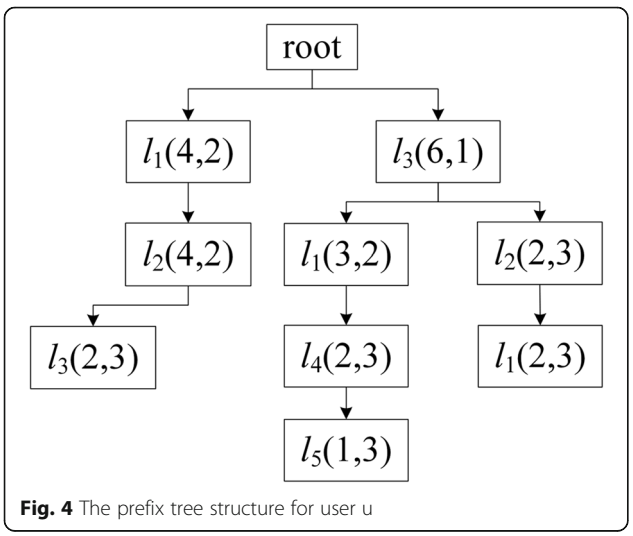

Fig. 4 The prefix tree structure for user u

4.2.2 Definition 8 (location weight)

α_l represents the number of times the user u checked in at location l, and w_l represents the sensitivity weight of location l. In definition 7, we define that the location l has the highest sensitivity when the value of p_{rl} is the smallest. However, when we calculate the location score, the weight of this location increases as the sensitivity of this location increases. Therefore, we use Eq. (6) to represent the weight of location l.

$$w_l = \frac{1 + p_{(n-p_{rl})}}{\sum_{r=1}^{n} p_r} \tag{6}$$

Since the abstract root node in the prefix tree is not the actual check-in location, the root node will not consume privacy budget. As shown in Fig. 4 above, we take the trajectory which starts at $l3$ as an example. The track contains two sub-tracks and its track height is four layers. We first calculate the sum of the sensitivity levels of all the nodes in the trajectory. Then we calculate the sum of the sensitivity levels for each layer. Finally, we allocate the privacy budget to each node at each layer based on the ratio of sensitivity weights. The specific privacy budget allocation algorithm steps based on location sensitivity is as follows:

Step 1: Input: privacy budget ε, prefix tree.
Step 2: Calculate the sum of the sensitivity levels of all location nodes.
Step 3: Calculate the sum of the location node sensitivity levels for each layer.
Step 4: Each layer is assigned a privacy budget by calculating the weight of each layer's location sensitivity.
Step 5: The privacy budget is assigned to each location node by calculating the weight of each location node at this layer.
Step 6: Output: Location set after assigning privacy budget.

In the above steps, the location node's sensitivity weight is calculated as shown in Eq. (7):

$$w' = \frac{p_r}{\sum_{r=1}^{n} p_r} \tag{7}$$

In definition 7, if the value of p_r defined in this paper is smaller, the sensitivity of the location is higher.

Therefore, the more sensitive the location l is, the smaller the weight it will have. As a result, the privacy budget allocated by location l is small. In differential privacy protection, if the ε assigned to location l is smaller, the added noise is greater, and the privacy protection is higher. Figure 5 shows the results of the privacy budget allocation with $l3$ as the root node of the subtree.

The same user may check in at the same location at different times, so the same location will appear multiple times in the prefix tree structure because of multiple check-in times. And because of the different location of the check-in location, the location of the node will be different. Therefore, the allocation of privacy budgets may cause the same location to be assigned to different privacy budgets. We need to merge the results of different allocations at the same location by location identifiers. From Fig. 5, we can see that l_1 appears in two subtracks. Its assigned privacy budgets are $2\varepsilon/15$ and $\varepsilon/15$, respectively. Therefore, the final allocated private budget for l_1 is $\varepsilon/3$.

4.3 Location recommendation under differential privacy protection

We get the privacy budget for each location of the user through Sect. 4.2 above. Then we use the Laplace mechanism to add appropriate noise to the location's check-in statistics to change the location sensitivity. With changes in location sensitivity, it is difficult for an attacker to discover the user's true preference for the location.

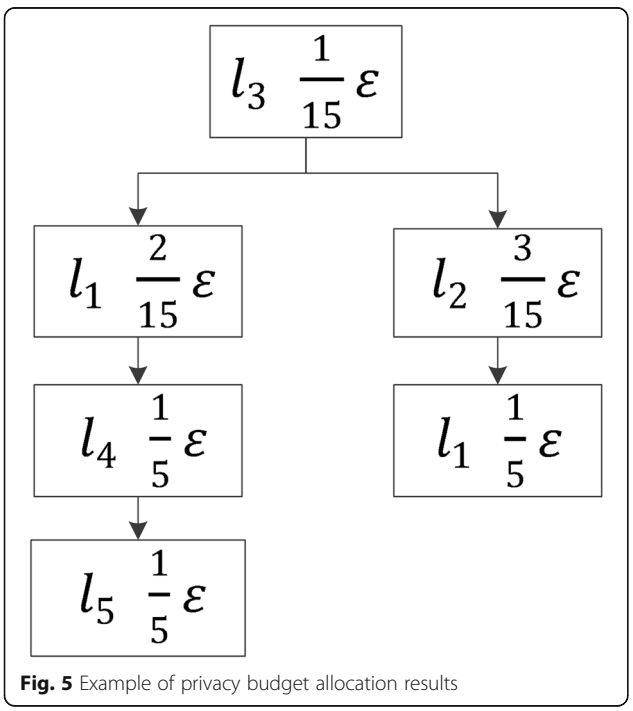

Fig. 5 Example of privacy budget allocation results

Step 1: Input: location set A. It includes check-in statistics and privacy budget for each location point.
Step 2: Add noise to its check-in statistics based on the privacy budget of each location point.
Step 3: Calculate the sensitivity level of each location point after adding noise.
Step 4: Output: location set B. It includes check-in statistics and sensitivity levels after adding noise at each location point.

After the set B is obtained, we calculate the interest score Eij of the user i on the location j by the equations in the definition 7 and construct the interest degree rating matrix ME of the user and the location. As shown in Eq. (8):

$$ME = \begin{bmatrix} E11 & E12 & \dots & E1j \\ E21 & \dots & \dots & E2j \\ \vdots & \vdots & \vdots & \vdots \\ Ei1 & \dots & \dots & Eij \end{bmatrix} \tag{8}$$

Then we use Eq. (5) to calculate the similarity of locations to construct a similarity matrix Ms, where $sim(i, j)$ represents the location similarity of users u_i and u_j. As shown in Eq. (9):

$$ME = \begin{bmatrix} sim(1,1) & sim(1,2) & \dots & sim(1,j) \\ sim(2,1) & \dots & \dots & sim(2,j) \\ \vdots & \vdots & \vdots & \vdots \\ sim(i,1) & \dots & \dots & sim(i,j) \end{bmatrix} \tag{9}$$

Finally, for the target user (the recommended user), the k users with the highest similarity are selected as similar neighbors. And based on the location set of similar neighbors, the locations that the user may be interested in are selected for recommendation.

5 Results and discussion

5.1 Experimental setup

All experiments of this paper are conducted on a computer with Intel i5-6500 3.20 GHz CPU and 16 GB RAM, running 64-bit Windows 10 OS. The algorithm is implemented in Python. This experiment uses the real public location data set Gowalla. The data set contains 107,092 users and 1,280,970 different locations with a total of 6,428,892 location check-in records [31]. In order to obtain better experimental results, we select the check-in records of the 20,000 users who are more active. The specific data format is shown in Table 2. Each piece of data consists of the user's unique identifier, time, and location information.

5.2 Evaluation index

In order to effectively evaluate the impact of differential privacy on the final recommendation effect, the experiment

Table 2 The samples of the user check-in data

User	Time	Latitude	Longitude	Location id
469	2010-01-16T06:08:14Z	36.02070054	− 115.0905554	154706
469	2010-03-14T23:56:47Z	36.09091582	− 115.179323	154743
469	2010-03-21T19:04:20Z	36.06759061	− 115.1785076	138981
469	2010-04-04T04:35:43Z	36.02070054	− 115.0905554	154706
469	2010-04-09T23:04:19Z	36.02070054	− 115.0905554	154706

selects the evaluation indicators commonly used in the recommendation system: precision, recall, F-Score, and MAE (mean absolute error) [32]. The definitions of precision and recall are respectively shown in Eqs. (10) and (11):

$$Precison = \frac{The\ number\ of\ effective\ recommended\ sets}{The\ number\ of\ recommended\ sets} \tag{10}$$

$$Recall = \frac{The\ number\ of\ effective\ recommended\ sets}{The\ number\ of\ test\ sets} \tag{11}$$

F-Score indicates the comprehensive recommendation quality. Precision and recall are metrics that measure recommendations from two different perspectives. Therefore, F-Score is the result of weighting and reassessing precision and recall. If the value of F-Score is larger, it means that the recommended quality is higher. As shown in Eq. (12):

$$F\text{–}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

MAE truly reflects the error between the test set and the predicted data. If the value of MAE is smaller, it means that the predicted performance is better. As shown in Eq. (13):

$$MAE = \frac{\sum_N \mid Score(u,l) - \widetilde{Score}(u,l) \mid}{\mid N \mid} \tag{13}$$

$Score(u,l)$ represents the true score of the user u for location l.$\widetilde{Score}(u,l)$ indicates the score of the user u for location a after adding noise. $|N|$ represents the number of test set samples.

In order to prove the effectiveness of this paper's scheme, we will compare it with uniform distribution (UD) method and private neighbor collaborative filtering (PNCF) method [33]. In this experiment, we set a total of five levels of sensitivity. The initial threshold θ for this paper is set to 30. When the number of check-in times is less than 30, the location sensitivity level p_r reaches a minimum of 5. The threshold interval is then gradually increased at intervals of 50. When the number of check-in times is between 30 and 80, the location sensitivity level p_r is 4. By analogy, when the number of check-in times exceeds 180, the location sensitivity level p_r reaches a maximum of 1. We use top-k (k = 10) to filter out the set of candidate locations with the highest similarity. The experiment uses fivefold cross-validation and then takes the average of precision and recall. The impacts of differential privacy protection on precision and recall are respectively shown in Figs. 6 and 7.

According to the above experimental results, it can be found whether it is precision or recall; the recommended quality of the UD and PNCF and the method based on the sensitivity level are lower than the recommended quality before protection. It can be seen that differential privacy protection has caused a certain loss of recommended quality. The reason for this is that the differential privacy

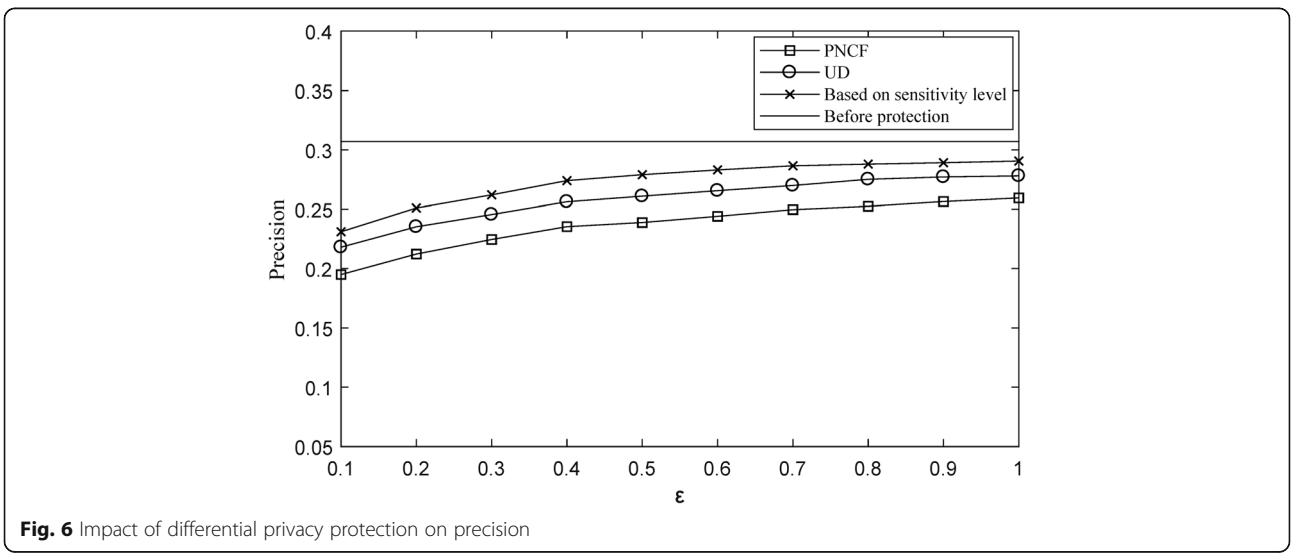

Fig. 6 Impact of differential privacy protection on precision

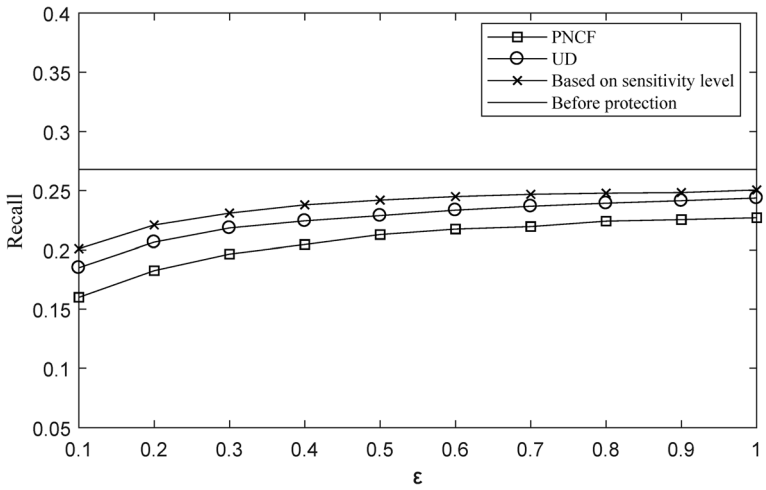

Fig. 7 Impact of differential privacy protection on recall

protection method needs to add Laplace noise to the original data set in order to provide privacy protection. This process inevitably causes a certain degree of noise error, which affects the final recommended quality. But in general, the method proposed in this paper is better than UD and PNCF. The F-Score comparison of the three methods is shown in Fig. 8.

As can be seen from Fig. 8, the differential privacy method based on location sensitivity level will result in a certain loss of recommendation quality. And because the user's check-in data is sparse, it will have a certain impact on the results of the experiment. However, it still can be seen from the figure that the F-Score of the privacy budget allocation method proposed in this paper is better than UD and PNCF. This is because UD does not consider the user's preference for location and the difference in subtrack length. The user's preference for locations can result in different levels of sensitivity at various locations. And different track lengths can cause privacy budgets to lean. However, the privacy budget allocation method based on location sensitivity level proposed in this paper maintains the user's original trajectory law and the frequency characteristics of the passing location. It better maintains the similarity between locations and reduces the similarity error caused by noise addition. But PNCF achieves the privacy protection effect in the recommendation process by adding Laplace noise based on the similarity. This directly obscures the similarity between users' locations, so PNCF results in a greater loss of recommended quality compared to the method proposed in this paper.

Figure 9 shows the comparison of MAE using three methods under different differential budgets. It can be seen from the figure that under different differential budgets, MAE based on sensitivity level is significantly smaller than the other two methods. Since PNCF masks

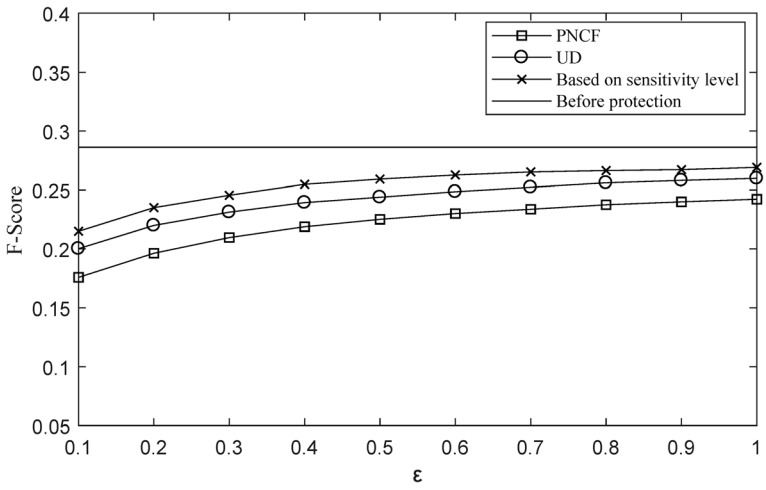

Fig. 8 F-Score of three different methods

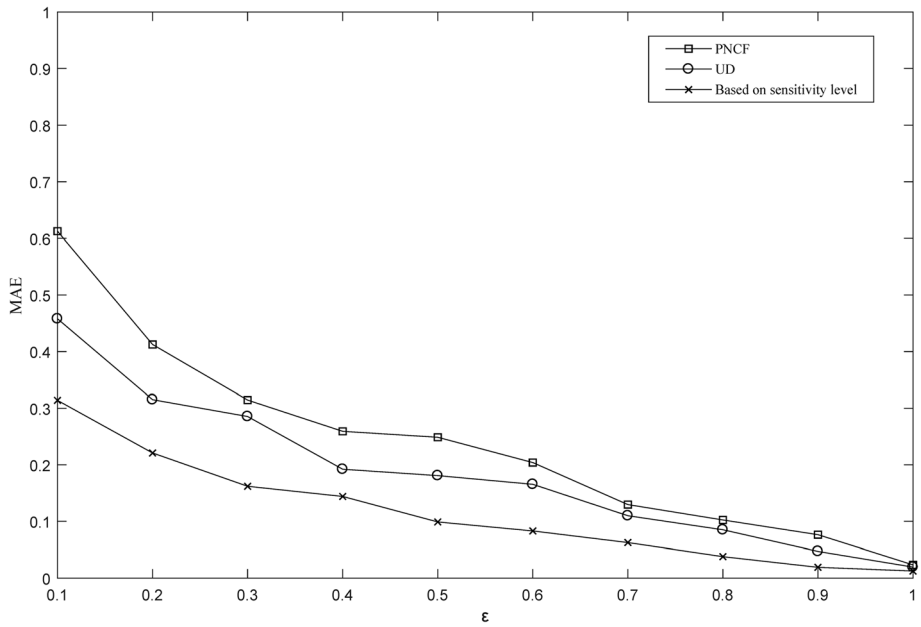

Fig. 9 MAE of three different methods

the similarity between the locations, its MAE is relatively large. And UD is to add noise evenly. Compared to the method proposed in this paper, UD ignores the user's preference for the location and thus cannot maintain the check-in frequency characteristics of the location. Therefore, its MAE is also relatively higher than the method proposed in this paper. It can be seen that the scheme of this paper can effectively reduce the MAE and improve the data availability.

Since the location sensitivity is determined by the number of check-in times in this paper, the range setting of the thresholds will also have an impact on the final result. Therefore, in the case where the location sensitivity level is a total of 5 levels and the privacy budget is 0.5, this paper compares the relationship between F-Score and threshold range. We set the initial threshold to 30 and the threshold interval between each sensitivity level to be 30. The interval in the experiment is increased by 20 each time. The result is shown in Fig. 10.

As shown, as the threshold interval increases, the threshold range between each sensitivity level becomes larger, and F-Score also increases slightly. This is because as the threshold range increases, the number of locations with higher sensitivity levels decreases, and the noise added to them is also relatively reduced. However, a decrease in the number of locations with a higher

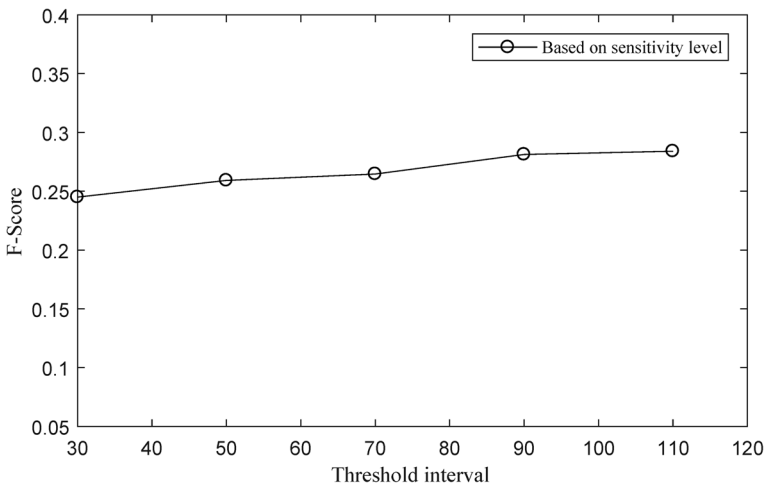

Fig. 10 Impact of threshold range on F-Score

sensitivity level means a reduction in the privacy protection strength. We need to set the threshold range reasonably to balance the relationship between recommendation quality and privacy protection. In summary, the differential privacy protection method proposed in this paper can effectively protect the user's location privacy in the location recommendation process and can effectively reduce the impact of differential privacy noise on the recommendation effect.

6 Conclusions

In order to protect user's location data security in SCADA under location-based recommendation, this paper proposes a protection method based on location sensitivity. This method stores the trajectory sequence through a prefix tree structure and uses the check-in statistics to divide the location sensitivity level. Then according to the allocated budget based on sensitivity level, appropriate differential privacy noise is added to the user's location check-in statistics to achieve privacy protection effect. By analyzing the experimental results on the real location data set, the proposed method can effectively protect the user's location privacy and reduce the impact of differential privacy noise on service quality. However, there are still some shortcomings in the research of this paper. The privacy protection method studied in this paper is based on collected static location data. Since SCADA monitors and collects data in real time, in order to protect the user's real-time location information, a dynamic privacy protection processing method is required. Therefore, the research on location privacy protection in dynamic environment is one of the important contents of the next step.

Acknowledgements

It was also supported by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions, Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX18_1032), Natural Science Foundation of Jiangsu Province (BK20150460), and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET). It was also funded by the open research fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education.

Authors' contributions

CY contributed for the equation of methodology and experiments, XJ for algorithm design and data sources, ZY for data analysis and performance analysis, and JW for overview of the proposed approach and decision analysis. All the authors conceived the idea, developed the method, and conducted the experiment. All authors read and approved the final manuscript.

Author's information

Nil.

Funding

This work was funded by the National Natural Science Foundation of China (61772282, 61772454, 61811530332, 61811540410).

Availability of data and materials

The data is available on request with prior concern to the first author of this paper.

Ethics approval and consent to participate

All the authors of this manuscript would like to declare that mutually agreed no conflict of interest.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China. [2]College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China. [3]School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, China.

Received: 2 April 2019 Accepted: 18 November 2019
Published online: 10 December 2019

References

1. M. Eirinaki, J. Gao, I. Varlamis, et al., Recommender systems for large-scale social networks: a review of challenges and solutions. Future Gener. Comp. Sy. **78**(1), 413–418 (2018)
2. I.B. Sassi, S. Mellouli, S.B. Yahia, Context-aware recommender systems in mobile environment: on the road of future research. Inform. Syst. **72**, 27–61 (2017)
3. Q. Fang, C. Xu, M.S. Hossain, et al., Stcaplrs: a spatial-temporal context-aware personalized location recommendation system. ACM T. Inter. Syst. Tec. **7**(4), 1–30 (2016)
4. M.F. Alhamid, M. Rawashdeh, H. Dong, et al., Exploring latent preferences for context-aware personalized recommendation systems. IEEE T. Hum-mach. Syst. **46**(4), 615–623 (2016)
5. C. Yin, J. Wang, J.H. Park, An improved recommendation algorithm for big data cloud service based on the trust in sociology. Neurocomputing. **256**, 49–55 (2017)
6. D. Zeng, Y. Dai, F. Li, et al., Adversarial learning for distant supervised relation extraction. CMC-Comput. Mater. Con. **55**(1), 121–136 (2018)
7. J.D. Zhang, C.Y. Chow, Ticrec: a probabilistic framework to utilize temporal influence correlations for time-aware location recommendations. IEEE T. Serv. Comput. **9**(4), 633–646 (2016)
8. V. Kumari, S. Chakravarthy, Cooperative privacy game: a novel strategy for preserving privacy in data publishing. Hum-cent. Comput. Info. **6**(1), 12 (2016)
9. C. Yin, L. Xia, S. Zhang, R. Sun, et al., Improved clustering algorithm based on high-speed network data stream. Soft Comput. **22**(13), 4185–4195 (2018)
10. M. Gruteser, D. Grunwald, in *Proceedings of the 1st international conference on Mobile systems, applications and services*. Anonymous usage of location-based services through spatial and temporal cloaking (2003), pp. 31–42
11. Yin, J. Xi, R Sun. Location privacy protection based on improved k-value method in augmented reality on mobile devices. Mob Inf Syst. 2017;12:1–7.
12. K. Miura, F. Sato, in *Proceedings of the 2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems*. A hybrid method of user privacy protection for location based services (2013), pp. 434–439
13. M.R. Nosouhi, V.V.H. Pham, S. Yu, et al., in *Proceedings of GLOBECOM 2017-2017 IEEE Global Communications Conference*. A hybrid location privacy protection scheme in big data environment (2017), pp. 1–6
14. C. Yin, S. Zhang, J. Xi, et al., An improved anonymity model for big data security based on clustering algorithm. Concurr. Comp-pract. E. **29**(7), e3902 (2017)
15. C. Chen, Y. Luo, Q. Yu, et al., TPPG: Privacy-preserving trajectory data publication based on 3D-Grid partition. Intell. Data Anal. **23**(3), 503–533 (2019)
16. C. Dwork, F. McSherry, K. Nissim, et al., in *Proceedings of Theory of cryptography conference*. Calibrating noise to sensitivity in private data analysis (2006), pp. 265–284

17. J. Wang, Z. Cai, Y. Li, et al., Protecting query privacy with differentially private k-anonymity in location-based services. Pers. Ubiquit. Comput. **22**(3), 453–469 (2018)
18. C. Yin, J. Xi, R. Sun, Location privacy protection based on differential privacy strategy for big data in industrial internet-of-things. IEEE T. Ind. Inform. **14**(8), 3628–3636 (2018)
19. G. Zhou, S. Qin, H. Zhou, et al., A differential privacy noise dynamic allocation algorithm for big multimedia data. Multimed Tools Appl. **78**(3), 3747–3765 (2019)
20. Z. Ma, T. Zhang, X. Liu, et al., Real-time privacy-preserving data release over vehicle trajectory. IEEE T. Ven. Technol. **68**(8), 8091–8102 (2019)
21. F. Hao, D.S. Park, D.S. Sim, et al., An efficient approach to understanding social evolution of location-focused online communities in location-based services. Soft Comput. **22**(13), 4169–4174 (2018)
22. N. Polatidis, C.K. Georgiadis, E. Pimenidis, et al., Privacy-preserving collaborative recommendations based on random perturbations. Expert Syst. Appl. **71**, 18–25 (2017)
23. D. Xue, L.F. Wu, H.B. Li, et al., A novel destination prediction attack and corresponding location privacy protection method in geo-social networks. Int. J. Distrib. Sens. N. **13**(1), 1550147716685421 (2017)
24. S. Zhang, X. Li, Z. Tan, et al., A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services. Future Gener. Comp. Sy. **94**, 40–50 (2019)
25. J. Wei, Y. Lin, X. Yao, et al., Differential privacy-based trajectory community recommendation in social network. J. Parallel Distr. Com. **133**, 136–148 (2019)
26. J.D. Zhang, C.Y. Chow, Enabling probabilistic differential privacy protection for location recommendations. IEEE T. Serv. Comput. (2018). https://doi.org/10.1109/TSC.2018.2810890
27. Z. Wang, J. Hu, R. Lv, et al., Personalized privacy-preserving task allocation for mobile crowdsensing. IEEE T. Mobile Comput. **18**(6), 1330–1341 (2018)
28. G. Quan, P. Viswanath, The optimal noise-adding mechanism in differential privacy. IEEE T. Inform. Theory. **62**(2), 925–951 (2016)
29. D. Lian, Y. Ge, F. Zhang, et al., Scalable content-aware collaborative filtering for location recommendation. IEEE T. Knowl. Data En. **30**(6), 1122–1135 (2018)
30. H. Han, S. Park, Traffic information service model considering personal driving trajectories. J. Inf. Proc. Syst. **13**(4), 951–969 (2017)
31. P. Mazumdar, B.K. Patra, K.S. Babu, et al., Hidden location prediction using check-in patterns in location-based social networks. Knowl. Inf. Syst. **57**(3), 571–601 (2018)
32. M. Aliannejadi, F. Crestani, Personalized context-aware point of interest recommendation. ACM T. Inform. Syst. **36**(4), 1–28 (2018)
33. T. Zhu, G. Li, Y. Ren, et al., in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. Differential privacy for neighborhood-based collaborative filtering (2013), pp. 752–759

Publisher's Note