IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 14, 2019, accepted December 10, 2019, date of publication December 16, 2019,
date of current version December 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2960008

LRM: A Location Recombination Mechanism for Achieving Trajectory *k*-Anonymity Privacy Protection

YUNFENG WANG[1], **MINGZHEN LI**[1], **SHOUSHAN LUO**[1], **YANG XIN**[1,2], **HONGLIANG ZHU**[1], **YULING CHEN**[2], **GUANGCAN YANG**[1], **AND YIXIAN YANG**[1,2]

[1]National Engineering Laboratory for Disaster Backup and Recovery, Information Security Center, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China

Corresponding author: Yunfeng Wang (wangyunfeng@bupt.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802300, in part by the Major Scientific and Technological Special Project of Guizhou Province under Grant 20183001, and in part by the Foundation of Guizhou Provincial Key Laboratory of Public Big Data under Grant 2018BDKFJJ021.

ABSTRACT Trajectory *k*-anonymity is a prevalent technique for protecting trajectory privacy. However, the existing techniques for generating fake trajectories can be easily broken by an adversary because of the failure to capture the probabilistic features and geographic features of the trajectories. They also reduce data availability. Thus, this paper proposes a location recombination mechanism (*LRM*) for achieving trajectory *k*-anonymity privacy protection. First, we propose a metric that measures the location pair similarity between location pairs. Based on this metric, we select sampling locations and divide locations into different equivalent probability classes. Locations in one equivalent probability class have the same probability as one corresponding base location. Then, we also introduce two metrics that measure the probabilistic similarity and geographic similarity between locations. Based on these metrics, we design algorithms to generate fake trajectories. These algorithms can recombine locations sampled from each equivalent probability class into trajectories. All of these trajectories meet the privacy protection requirements for both base trajectories and sampling trajectories. Finally, we evaluate our scheme thoroughly with real-world data. The results show that our method can protect the privacy of base trajectories and sampling trajectories and achieve a better performance of service provider utility and data availability than other schemes.

INDEX TERMS Trajectory *k*-anonymity, trajectory privacy, privacy protection, location recombination mechanism, fake trajectories.

I. INTRODUCTION

Currently, the privacy issue in trajectory data publication is attracting increasing attention. In reality, some service providers regularly publish a large amount of trajectory data such as check-in data and taxi mobile data. Mining and analyzing of these data [1]–[3] can provide people with facilitation services such as advertising pushes [4] and traffic navigation [5]. However, inappropriate publishing methods may violate users' privacy [6]–[9]. For example, if the publishing trajectories are not anonymous, even when they are encrypted, the privacy of users may be breached. The cryptography-based approach may encrypt only part of a user's attributes, not achieving full privacy [8]. Thus, for trajectory data publication, fake data [10], sensitive location suppression [11], etc. are used to protect privacy. However, these techniques can reduce data availability in terms of available data and data accuracy. The former means that the published locations can be used for data analysis such as i.i.d. mining mobility patterns and recovering trajectory. The latter means that the proportions of different published locations are stable (i.e., the proportion of visiting a workplace or moving between home and a shop). For example, [12], [13] can generate hard-to-reach locations that cannot be used for data

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

analysis. For data accuracy, methods [14], [15] can always publish a workplace with a lower proportion instead of a shop with a higher proportion during a long period of time, thus altering the proportions of the workplace and the shop. Therefore, how to balance the privacy protection and data availability in data publication has been a topic of interest.

To address the above problems, many trajectory k-anonymity methods have been proposed for trajectory privacy protection [14]–[18]. They publish a k-anonymous group containing a real trajectory (called the base trajectory) and at least k-1 fake trajectories to achieve a k-anonymity level of privacy protection. Locations on the base trajectory or on the fake trajectory are called base locations and fake locations, respectively. The existing methods [10], [12], [14], [15], [17], [19], [20] believe that an adversary usually identifies the base trajectory by utilizing the mobility pattern (speed, region, etc.) of different users' trajectories. Therefore, these methods are mainly based on similar mobility patterns, such as the random walk model [12], historical trajectory sampling [15], [17] and the grid model [20], [21], to generate fake trajectories with high similarity to the mobility pattern of the base trajectory. Note that the adversary identifies a user's real trajectory by establishing the correct correspondence between the trajectory and the user. If the adversary has identified the base trajectory, he can also infer the corresponding user's real trajectory according to his own background knowledge [3], [7]. It also means that encrypting a user's name may still reveal her privacy [8]. Thus, the criterion for the user's trajectory privacy leakage discussed in this paper is that the adversary correctly identifies the base trajectory.

However, existing methods ignore the probabilistic features [2], [21] and the geographical features [22]–[24] of trajectories, so base trajectories can be easily identified by the adversary. The probabilistic feature is a set of probabilities with which users access a sequence of locations or move between them chronologically. The geographical feature is a set of geographical positions where users access a sequence of locations chronologically. For a trajectory L of user u consisting of n positions, let L be $l_1 \rightarrow l_2 \rightarrow \cdots \rightarrow l_n$. We use the vector $(P_1, P_2, \cdots, P_n)$ to denote the probabilistic feature of L, where $P_i (1 \leq i \leq n)$ is the probability of location l_i. For P_i, it consists of two probabilities: the probability π_i (access probability) that l_i was visited by u and the probability p_i (transition probability) that u moved from l_{i-1} to l_i in the past. Therefore, we also use $<\pi_i, p_i>$ to represent P_i. Similarly, we use the vector $(G_1, G_2, \cdots, G_n)$ to denote the geographical feature of L, where $G_i (1 \leq i \leq n)$ is the geographic location (geographic coordinates) of location l_i.

The adversary can obtain the probabilistic features and geographic features of trajectories in a variety of ways. For example, to publish trajectories in [10], the base trajectory can be obtained by connecting each cloaking region, since locations in a k-anonymous group are in a smaller cloaking region; the probability of each actual location can be obtained because all locations in a k-anonymous group have the same probability [21]. The adversary can also acquire the probabilistic features and geographic features through data mining [3].

Note that the mobility pattern is mostly specific to the individual and reflects the individual's lifestyle. This means that different users in a group have different mobility patterns. The probability of a location represents overall human behavior and reflects the common lifestyle of humans overall. This means that the access probability and the transition probability of different users in a crowd are the same. That is, the access probability and transition probability of a specific user u and the crowd are equal. Two locations with the same access probability and the transition probability have the same probability of locations. For two locations, based on [22], [23], if the geographic distance between them is geographically indistinguishable (less than a certain threshold), they have the same geographic locations. For l_i, if all locations within a region have the same geographic location as l_i, this region is indistinguishable from l_i. For two trajectories, if every corresponding location of them has the same probability of location and the geographic location, they have the same probability feature and geographic feature. Consequently, the attacker can identify the base trajectory by judging whether trajectories in a k-anonymous group have the same probability feature and geographic feature as the base trajectory. Fig. 1 shows an example to explain how the adversary identifies the base trajectory of Alice by analyzing probabilistic features and geographic features of different trajectories.

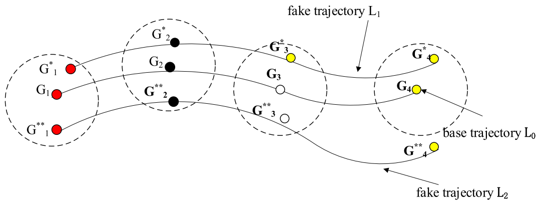


FIGURE 1. An example of identifying the base trajectory of Alice in a *k*-anonymous group.

In Fig. 1, differently colored small circles represent different probabilities of locations, and the large circle represents the indistinguishable region of the base location. In this example, L_0, L_1 and L_2 are trajectories in a 3-anonymous group, which is released for protecting the base trajectory L_0 of Alice. L_1 and L_2 are fake trajectories. As shown in Fig. 1, the probability of location G_3^* on L_1 and location G_3 on L_0 is different, and the geographic location of location G_4^{**} on L_2 and location G_4 on L_0 is different. Therefore, L_0 and L_1 have a different probability feature, and L_0 and L_2 have a different geographic feature. It means that the mobility patterns of L_1 and L_2 are different from those of L_0. The adversary can filter out L_1 and L_2 and correctly identify Alice's trajectory. (We assume that the adversary with background knowledge knows the probabilistic features and geographical features of Alice's mobility pattern.)

To prevent identification of base trajectories, existing methods contain at least k similar trajectories in a k-anonymous group. Methods [15], [17], [24] select trajectories (sampling trajectories) from historical trajectories as fake trajectories but ignore the privacy leakage of sampling trajectories. In this paper, both sampling trajectories and historical trajectories are real trajectories of users, and the locations in them are historical locations and sampling locations. Another problem is that fake trajectories in the k-anonymous group are noise, affecting the data availability for the published data. Some methods, such as [12], produce some hard-to-reach locations. However, these locations cannot be used for data analysis, which reduces the data availability in terms of available data. Other methods, e.g., [15], publish real locations without considering the data accuracy, which reduces the data availability, such as increasing the probability of individuals reaching a location.

In this paper, we propose a location recombination method, called *LRM*, to achieve trajectory k-anonymity privacy protection. The *LRM* selects and recombines different sampling locations to generate k-1 fake trajectories similar to the base trajectory in terms of probabilistic features and geographical features. In the *LRM*, we first construct a probabilistic model based on probabilistic features of trajectories to generate fake trajectories similar to probabilistic features of the base trajectory. However, it is difficult to select sample trajectories similar to the base trajectory as fake trajectories. Therefore, we propose a metric that measures the location pair similarity between location pairs (two adjacent locations in a trajectory). Using the metric, we select sampling trajectories and divide them into different classes (called equivalence probability classes) by the similarity to the base location. In the same class, the access probability of each sampling location is similar to the base location. Then, we introduce two metrics that measure the probabilistic similarity and geographic similarity between a fake trajectory and the base trajectory. Using them, we design algorithms to synthesize fake trajectories. For each fake trajectory, we conduct the privacy test to guarantee that the generated k-1 fake trajectories meet the privacy protection requirements for both base trajectories and sampling trajectories. Finally, we generate fake trajectories using real trajectories and evaluate the effectiveness of the *LRM* in terms of data availability and privacy level under inference attacks. The results show that the *LRM* has efficient data availability and better privacy protection than [14], [21], [25].

In this paper, we generate fake trajectories similar to the base trajectory based on four facts: (i) The access probability and the transition probability of each location will vary with different time periods, and such change has periodicity. Subway stations, for example, are visited by many more people in the morning on weekdays than in the evening. (ii) In a given time period, the access probability of each location is the probability of all users accessing that location. (iii) Both differences and similarities exist in the access probability and transition probability in different locations. Based on (i), (ii) and (iii), we can select locations with similar access probability and transition probability to replace the base location and guarantee that it is not recognized by the adversary in terms of probabilistic features. This is the basis for our choice of sampling locations to synthesize fake trajectories similar to the base trajectory in terms of probabilistic features. The differences and similarities in (iii) are the basis of the attack model and location classification, respectively. (iv) Geographical features in different locations exhibit a similarity. This similarity reflects that the geographical distance between the two locations is close enough to be indistinguishable. For two trajectories, if each corresponding location in them is geographically indistinguishable, their geographic features are indistinguishable.

To intuitively explain the advantages of the *LRM*, we assume that Alice and Bob are located in locations h_a and h_b ($h_a \neq h_b$) in time period T_1, and locations w_a and w_b ($w_a \neq w_b$) in time period T_2. If h_a and h_b, w_a and w_b, have similar access probabilities, and $h_a \rightarrow w_a$ and $h_b \rightarrow w_b$ also have similar transition probabilities, then $h_a \rightarrow w_a$ and $h_b \rightarrow w_b$ have similar probabilistic features. In this paper, two locations (e.g., h_a and h_b) satisfying the above relationship have probabilistic similarity and they are equivalent locations. The set of equivalent locations is an equivalence probability class. Furthermore, if h_a and h_b, w_a and w_b are geographically indistinguishable, they have geographic similarity. Therefore, we synthesize k-1 fake trajectories including two features: (1) Each trajectory consists of sampling locations selected from each equivalence probability class. (2) Each corresponding two adjacent locations (called location pair) between a fake trajectory and the base trajectory have probabilistic similarity and geographical similarity; thus, the base trajectory is unrecognized.

The *LRM* improves data availability. Compared with [25], the *LRM* generates fake trajectories combined by sampling locations. Thus, all published data by the *LRM* can be used effectively. In the *LRM*, the access probability and transition probability are derived from the statistics of a large number of locations visited by many users over a long period of time. Compared with [14], the two types of probabilities are stable and do not degrade the accuracy using these locations. These conclusions are shown in Fig. 12 and Fig. 13.

The *LRM* protects the privacy of both base trajectories and sample trajectories. For base trajectories, the *LRM* must satisfy two conditions before they are published: (1) The number of fake trajectories is at least k-1. (2) The base trajectory and the fake trajectories are similar both in probabilistic features and geographical features. Therefore, it ensures that base trajectories cannot be identified by inference attacks. For sampling trajectories, the *LRM* requires that the number of locations that are the same as those in the fake trajectory does not exceed the limit threshold. Therefore, sampling trajectories are not identified.

Specifically, the major contributions of this paper are as follows:

- *To protect the privacy of base trajectories against the adversary with a mobility pattern, we propose the LRM* *to generate k-1 fake trajectories similar to base trajectories in terms of probabilistic features and geographical features. The LRM also protects the privacy of sampling trajectories.*

- *We propose the location pair similarity. Using it, we can select sampling trajectories and divide them into different equivalence probability classes by the similarity to the base location.*
- *We propose the probabilistic similarity and the geographical similarity. Using them, we design algorithms to synthesize fake trajectories that are similar to base trajectories in terms of probabilistic features and geographical features.*
- *The validity of LRM in data availability and privacy protection is verified on real-world trajectories. Compared with [14], [21], [25], the results show that the LRM has efficient data availability and can meet the privacy requirements of both base trajectories and sample trajectories.*

The rest of the paper is organized as follows: In Section II, we discuss the related work. We present a sketch of our scheme and describe the main intuition behind our scheme for generating fake trajectories in Section III. We introduce the probabilistic model and define location pair similarity, probabilistic similarity and geographical similarity for analyzing the similarity of mobility patterns between the base trajectory and the fake trajectory in Section IV. We describe the detailed algorithms for generating k-1 fake trajectories. We evaluate the performance of our scheme in Section VI and Section VII. We conclude this paper in Section VII.

II. RELATED WORK

In data publication, the trajectory k-anonymity has been widely used to protect trajectory privacy in many different applications such as IoT [24], [26], [27], sensor networks [18], [28], search engines [29], mobile social networks [30], traffic management [15], [31], etc. In these applications, methods, such as the random walk model, historical trajectory sampling and grid model, are used to achieve privacy protection by adding noise to the base trajectory to generate k-1 fake trajectories similar to the base trajectory. In the existing methods, [12], [13] randomly generated noise, and some noise may be of hard-to-reach fake locations. [14], [15] generated noise based on historical locations, and all these locations are reachable. According to different ways of generating noise, existing trajectory k-anonymity can be divided into two types: the random method and the historical method.

Random methods generate noise randomly. For example, such a method can be accomplished by randomly selecting the locations near base locations [12], [32], rotating the base trajectory by a certain angle as a fake trajectory [10], [13], [19], and randomly selecting locations from grids satisfying some constraints [21]. There are two shortcomings in this method: (1) Some locations are hard to reach, causing fake trajectories to be easily recognized. (2) Hard-to-reach locations for data analysis reduce data availability.

Historical methods mainly select the locations from historical locations as noise. For example, historical trajectories were selected as fake trajectories [15], [31], and some segments of historical trajectories were selected to combine into fake trajectories [17], [18]. However, they did not consider probabilistic features and geographic features of trajectories, leading to two disadvantages: (1) The adversary can identify base trajectories by analyzing the difference between the fake trajectories and base trajectories in terms of probabilistic features and geographic features. (2) Fake locations increase without considering the data accuracy, causing changes in access probability and transition probability and degrading the accuracy using these locations.

For data availability, although the *LRM* also generates fake trajectories by adding noise, this method is significantly different from the above methods. Compared with the random method, the *LRM* selects locations from historical trajectories as noise, which does not generate hard-to-reach locations. In the *LRM*, the access probability and transition probability are based on the statistics of a large number of locations visited by many users over a long period of time. Thus, they exhibit stability [33], [34]. However, locations in the historical method increase without considering the data accuracy. Hence, it causes changes in access probability and transition probability of locations.

The cryptography-based approaches, such as [35], [36], are mainly used to make the user's key attributes invisible to the adversary. However, efficient cryptography-based approaches that provide full privacy do not exist, and the remaining approaches are unable to make a trade-off between privacy and loss of quality of service [8].

In particular, the existing methods fail to protect the probabilistic features and geographical features of trajectories. Thus, these methods cause base trajectories to be identified and data availability to be reduced. Aiming at the above problem, we propose the *LRM* to generate fake trajectories that are difficult to be identified and improve data availability.

Furthermore, although some methods are not k-anonymous, they are the theoretical basis of this paper. In [7], Du et al. demonstrated that the adversary can identify users' anonymous ID of published data by analyzing the quasi-identifier and attribute information. Xu *et al.* [3] proved that users' anonymous ID can be identified by using statistical attributes of trajectories. Therefore, the work of this paper focuses on protecting trajectory privacy from the perspective of mobility patterns. In [2], Noulas et al. analyzed the geographic feature of the location. In [22], Andres et al. proposed the concept of geographical indistinguishability and proved that the farther the geographical distance between two locations is, the greater the similarity is. Inspired by schemes [37]–[39], which recover an image from n shadows, we implemented this inspiration to reconstruct the trajectory from historical locations.

III. OUR SCHEME

In this section, we present a sketch of the *LRM*, as shown in Fig. 2, and describe the main intuition behind our scheme for generating fake trajectories. In our scheme, we generate k-1 fake trajectories through a 6-step process. We first compute the probability for each location of each historical trajectory, including the base trajectory. Then, we build probabilistic models for all historical trajectories and select sampling trajectories based on the base trajectory. Next, we partition locations from the base trajectory and sampling trajectories into distinct equivalence probability classes and build a location pair graph. Using the location pair graph, the fake trajectories have probabilistic and geographic similarity to the base trajectory. Table 1 provides a list of the notations used in this paper.

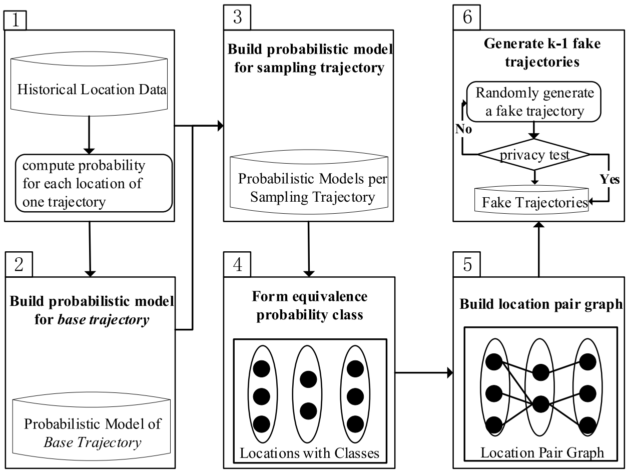

1
Historical Location Data
compute probability for each location of one trajectory
2
Build probabilistic model for *base trajectory*
Probabilistic Model of *Base Trajectory*
3
Build probabilistic model for sampling trajectory
Probabilistic Models per Sampling Trajectory
4
Form equivalence probability class
Locations with Classes
5
Build location pair graph
Location Pair Graph
6
Generate k-1 fake trajectories
Randomly generate a fake trajectory
No
privacy test
Yes
Fake Trajectories

FIGURE 2. Sketch of the ***LRM***.

A. COMPUTING THE ACCESS PROBABILITY AND THE TRANSITION PROBABILITY

Different locations have different properties, e.g., time period, time, geographical location, access probability, and transition probability. We use them as properties of locations. Therefore, the access probability for location l in time period T is defined as the ratio of the number of times all users access l in T to the number of times all users access all historical locations. For the transition probability, we assume users could have transformed to l', l'', and l''' from l in the historical locations. The transition probability of l to l' is defined as the ratio of the number of times that users could have transformed to l' from l to the sum of times that users could have transformed to l', l'', and l''' from l. Note that the access probability we discuss here is based on trajectory rather than on a discussion of individual locations.

B. PROBABILISTIC MODEL

A trajectory is a discrete-time sequence of locations. As probability feature is one of the mobility patterns, we construct a probabilistic model for the trajectory.

TABLE 1. List of notations.

Symbol	Notion
$A = \{a_i \| i = 1, 2, ..., n_a\}$	The minimum access period during which users successively visit the same locations.
a_i	The i^{th} time period of A
$L = \{l_i \| i = 1, 2, ..., n\}$	A trajectory that consists of n locations
$A(l_i)$	The time period of l_i
$T(l_i)$	The time of l_i
$L - model = (P, \Pi)$	Probabilistic model of L
$P = \{P(l_i) \| l_i \in L\}$	Set of transition probabilities in the probabilistic model of L
$\Pi = \{\pi(l_i) \| l_i \in L\}$	Set of access probabilities in the probabilistic model of L
$P(l_i)$	Probability that a user moves to location l_i from location l_{i-1}
$\pi(l_i)$	Probability that a user is in location l_i
$B = \{b_i \| i = 1, 2, ..., n\}$	Base trajectory that consists of n locations
$\mathcal{A} = \{B_1, B_2, ..., B_n\}$	Set of equivalent probability classes
B_i	equivalent probability class of b_i
$\mathcal{F}$	Set of k-1 fake trajectories in k-anonymous group
$F = \{f_i \| i = 1, 2, ..., n\}$	A fake trajectory in $\mathcal{F}$
$\mathcal{C}$	The set of trajectories that are possible combinations of n locations from n equivalence probability classes
$C = \{c_i \| i = 1, 2, ..., n\}$	A trajectory in $\mathcal{C}$
$\mathcal{H}$	The set of historical trajectories
$H = \{h_i \| i = 1, 2, ..., n_H\}$	A trajectory in $\mathcal{H}$
$\mathcal{L}$	Set of nonrepetitive locations in $\mathcal{H}$
$P_C(c_i)$	Probability that attackers believe the probabilistic features of two locations are similar
$d_P(c_i, b_i)$	Euclidean distance between probabilistic features of c_i and b_i
$sim_P(B, C)$	Probabilistic similarity between B and C
$d_G(c_i, b_i)$	Euclidean distance between coordinates of c_i and b_i
$sim_G(B, C)$	Geographic similarity between B and C
$Q = \{q_1, q_2, ..., q_{n_n}\}$	Index number of the cell of the quadtree

Consider the trajectory L of user u: for a location l on L, its access probability only depends on l. Therefore, the access probability distribution of L is an independent distribution. For two adjacent locations l and l', we find that the transition probability of l to l' only depends on l, i.e., the transition probability of a location only depends on the location in its last instant. Therefore, the transition probability distribution of L is a first-order Markov chain. Based on it, we model the probabilistic model of L as $\langle P, \Pi \rangle$, where P is the transition probability distribution of locations and Π is the access probability distribution of locations.

Note that users may visit multiple locations during the same time period. Because of the time dependence of these locations, the trajectory still applies to the above model.

C. TRAJECTORY SAMPLING

The *LRM* selects sampling locations from sampling trajectories sampled from historical trajectories and recombines them into fake trajectories. To generate at least k-1 different fake trajectories, sampling trajectories must satisfy three conditions: for each base location, there are at least k-1 sampling locations with similar access probabilities (1) and transition probabilities (2) to each base location. Furthermore, sampling locations satisfying (1) and (2) and their corresponding base locations are in the same time period and almost at the same time (3).

D. SAMPLING LOCATION CLUSTERING

For a base trajectory, if k-1 sampling trajectories are selected as fake trajectories, each one must meet the condition that the access probability and transition probability of each sampling location are almost the same as its corresponding base location. In fact, such sampling trajectories are difficult to select and reveal the privacy of them. Even so, several sampling locations in each sampling trajectory may still satisfy the condition. If we select such sampling locations for each base location and reconstruct them into a fake trajectory, the access probability of each location on the fake trajectory must be similar to its corresponding base location. Nevertheless, there is no guarantee that the transition probability satisfies the condition.

Considering the example about Alice and Bob in the INTRODUCTION, if each location pair on sampling trajectories (called sampling location pair) and its corresponding location pair on base trajectories (base location pair) satisfy the above 3 conditions, reconstructed fake trajectories also necessarily satisfy the conditions. Therefore, we propose the location pair similarity to measure how similar location pairs are.

In the above example, h_a and h_b, w_a and w_b are in the same equivalence probability class. To illustrate the necessity of clustering locations in this way, we assume that $h_a \rightarrow w_a$ and $h_b \rightarrow w_b$ are the base trajectory and the sampling trajectory respectively. Each of them contains two locations. We find that selecting $h_a \rightarrow w_b$ or $h_b \rightarrow w_a$ may be better than $h_b \rightarrow w_b$ to protect the privacy of the base trajectory (assuming that $h_b \rightarrow w_b$ or $h_b \rightarrow w_a$ has probabilistic similarity to the base trajectory). Therefore, in this paper, we can divide sampling locations into different equivalence probability classes by base locations and select a sampling location from each class to reconstruct a fake trajectory.

E. CONSTRUCTING THE LOCATION PAIR GRAPH

The similarity of mobility patterns for trajectories reflects the similarity of both probabilistic features and geographical features of trajectories. To this end, we propose the probabilistic similarity and the geographic similarity to measure the similarity of probabilistic features and geographical features between two trajectories. To obtain a fake trajectory similar to the base trajectory in the mobility pattern, one method is to select a fake trajectory from all reconstructed trajectories. However, this method has a higher complexity. Assuming that there are n equivalence probability classes, the number of locations in each class is m_i, $0 \leq i \leq n$; then, the number of fake trajectories that can be synthesized is $\prod_{i=0}^{n} m_i$. Hence, we propose an algorithm to construct a location pair graph. In the graph, each node represents a location, locations in adjacent classes are connected by edges, and the weight of the edge is represented by a binary group consisting of the probabilistic similarity and geographical similarity. Through the location pair graph, we can optimize the reconstruction process of fake trajectories.

F. GENERATING FAKE TRAJECTORIES

After constructing the location pair graph, we consider how to generate k-1 fake trajectories. In this process, fake trajectories must be similar to the base trajectory in the mobility pattern, and the privacy of sampling trajectories must be protected. That is, a fake trajectory and a sampling trajectory should have a similarity that is as high as possible and should have as few as possible of the same location. Therefore, the process needs to ensure that each fake trajectory satisfies two conditions: (1) Protect the privacy of the base trajectory. (2) Protect the privacy of sampling trajectories. To this end, we design algorithms to generate k-1 fake trajectories. The algorithms first generate the first trajectory similar to the base trajectory in the mobility pattern and conduct a privacy test to determine whether it satisfies conditions (1) and (2). If so, it is regarded as the first fake trajectory. Otherwise, the second trajectory is generated according to the above step until the first fake trajectory is generated. Similarly, we generate all k-1 fake trajectories in turn. In this process, k-1 fake trajectories satisfy trajectory k-anonymity, so it can meet condition (1). All trajectories pass the privacy test; thus, they satisfy condition (2).

IV. TRAJECTORY SIMILARITY METRICS

In this section, we introduce the probabilistic model and two metrics to analyze the probabilistic and geographic similarity of trajectories.

A. PROBABILISTIC MODEL

In reality, people tend to move along a fixed route. Thus, locations are visited periodically, e.g., working in a company from 9 a.m. to 11 a.m., visiting restaurants from 12 p.m. to 1 p.m. and remaining at home from 12 a.m. to 6 a.m. People have different access probabilities for different locations within the same time period and have the same location within different time periods. Therefore, we can establish the probabilistic model consisting of three factors: the time period, coordinates and time. Before building the model, we first introduce some concepts.

Definition 1 (Access Period): The physical meaning of the access period refers to the time it takes for people to repeatedly access some of the same locations in the same order. Among them, the time of accessing these locations once is called the minimum access period A. In this paper, A is represented as a set consisting of n_a ordered discrete-time segments. That is, $A = \{a_i | i = 1, 2, \ldots, n_a\}$.

Definition 2 (Trajectory): For a trajectory L composed of n locations, we formalize it as $L = \{l_i | i = 1, 2, \ldots, n\}$. For $\forall l_i \in L$, $A(l_i) \in A$ represents the time period in which l_i is located; $T(l_i)$ indicates the time of l_i being accessed.

For L, obviously, the access probability of l_i only depends on l_i, so the access probability distribution of the trajectory is an independent distribution. For the transition probability, whether the next location is l_i only depends on l_{i-1}, but has nothing to do with the previous location of l_{i-1}. It indicates that the transition probability distribution of the trajectory is a first-order Markov chain. Therefore, we define the probabilistic model shown in **Definition 3**.

Definition 3 (Probabilistic Model): For the trajectory L established on A, its probabilistic model is a binary group $L - model = (P, \Pi)$.

- *$P = \{P(l_i) | l_i \in L\}$ is the transition probability distribution of $L - model$, and $P(l_i)$ is the transition probability of l_{i-1} to l_i.*
- *$\Pi = \{\pi(l_i) | l_i \in L\}$ is the set of the access probability distributions of positions in the $L - model$, and $\pi(l_i)$ is the access probability of l_i.*

For the $L - model$, $P(l_i)$ is the ratio of $m(l_{i-1}, l_i)$ to $\sum_{l \in S(A(l_{i-1}))} m(l, l_i)$, where $m(l_{i-1}, l_i)$ is the number of times that users move from l_{i-1} to l_i, and $S(A(l_{i-1}))$ is the set of locations that will move to l_i during $A(l_{i-1})$. $\pi(l_i)$ is the ratio of $m(l_i)$ to $m(\cdot)$, where m_i is the number of times that l_i appears in the historical locations and $m(\cdot)$ is the total number of historical locations. Assume that there are 10 historical locations, including a,b,c,d,e, and f, and users can only move from a, b, c, and d to f. The numbers of times that all historical locations and f are visited are 100 and 40, respectively. The number of times that users moved from a, b, c, d to f are 2, 3, 5 and 7 respectively. Then, $m(\cdot)$ is 100, $m(f)$ is 40 and $\pi_f = 0.4$. Assume $A(a) = A(b) = A(c)$; then, $S(A(a)) = \{a, b, c\}$, $m(a, f) = 2$, $m(b, f) = 3$, $m(c, f) = 5$ and $P(f) = 0.2$. Note that the statistics here are not the minimum period of data, but the average of multiple periods to make the statistics more accurate. In the $L - model$, the first location l_i does not depend on any location, that is, the transition probability $P(l_i)$ from any position to l_i is equal to the access probability $\pi(l_i)$, while any other location l_i only depends on l_{i-1}. Therefore, the transition probability $P(l_i)$ and the access probability $\pi(l_i)$ are calculated as follows.

$$
\begin{aligned}
P(l_1) &= \pi(l_1) \\
P(l_i) &= \frac{\sum_{l \in S(A(l_{i-1}))} m(l, l_i)}{m(l_{i-1}, l_i)} \\
\pi(l_i) &= \frac{m(l_i)}{m(\cdot)}
\end{aligned} \tag{1}
$$

B. TRAJECTORY SIMILARITY METRICS

The *LRM* reconstructs fake trajectories by recombining locations. In this section, we propose three metrics to measure the similarity of location pairs, probabilistic features, and geographic features of two trajectories, which are also the basis of algorithms generating fake trajectories in Section V.

In our scheme, there are two ways to generate fake trajectories: (1) Sampling trajectory. A sampling trajectory is selected as a fake trajectory, but the differences from the base trajectory in terms of probabilistic features and the geographic features make the base trajectory easy to be identified. (2) Recombining locations. Sampling locations similar to the access probability, transition probability and geographic features of base locations are selected and recombined into fake trajectories, which can ensure the probabilistic similarity and geographic similarity with the base trajectory. Therefore, this paper reconstructs fake trajectories by recombining locations. The basic idea is to divide sampling locations into different equivalence probability classes based on the location pair similarity and finally reconstruct fake trajectories similar to the base trajectory in the probabilistic features and geographical features.

Location pair similarity measures the relevance of two location pairs, reflecting whether two users visit two locations with a similar access probability and move to the next location with a similar transition probability. For two location pairs, if the access probability and transition probability are the same, they have maximum uncertainty. Thus, the sampling locations can be sequentially divided into different classes, and the access probabilities of the locations in the same class are similar.

The probabilistic similarity measures the correlation between two probability models, reflecting whether two trajectories have similar probabilistic features. Although the access probability of each location in a fake trajectory of recombined locations from each equivalence probability class is similar to its corresponding base location, it is difficult to ensure that the transition probabilities are completely similar. Therefore, we use probabilistic similarity to measure the similarity of the access probability and the transition probability between the fake trajectory and the base trajectory. If two trajectories exhibit probabilistic similarity, their access probability and transition probability are similar.

Similarly, the geographic locations of two trajectories with similar probabilities are difficult to guarantee to be completely similar. Therefore, we consider measuring the geographic differences of two trajectories with similar probabilistic features. Geographic similarity measures the geographical difference between two trajectories. If two trajectories with similar probabilistic features meet the requirement of geographic similarity, they are geographically similar.

C. LOCATION PAIR SIMILARITY METRIC

Measuring the location pair similarity is accomplished by selecting similar location pairs from sampling locations for each base location pair and dividing them into different equivalence probability classes. Moreover, the criterion that two locations are similar is that they are almost the same in access probability, transition probability, time period, and access time (see the example in Section III).

For the base trajectory $B = \{b_i | i = 1, 2, \ldots, n\}$, the server needs to publish a k-anonymous group composed of B and k-1 fake trajectories. Let the set of these fake trajectories be $\mathcal{F}$, and $F = \{f_i | i = 1, 2, \ldots, n\}$ is a fake trajectory in $\mathcal{F}$. According to **Definition 2**, we use $\mathcal{A} = \{B_i | i = 1, 2, \ldots, \}$ to represent equivalence probability classes of base locations in the k-anonymous group, where $B_i \in \mathcal{A}$ represents the equivalence probability classes of b_i. Additionally, we use $\mathcal{C}$ to denote the set of trajectories that are possible combinations of n locations from n equivalence probability classes. $C = \{c_i | i = 1, 2, \ldots, n\}$ is a trajectory in $\mathcal{C}$ and $\mathcal{F} \subsetneqq \mathcal{C}$. We also use $\mathcal{H}$ to express the set of historical trajectories and $H = \{h_i | i = 1, 2, \ldots, n_H\}$, where H is a historical trajectory in $\mathcal{H}$. $\mathcal{L} = \left\{\mathcal{L}_l | \mathcal{L}_l \in \bigcup_{\mathcal{H}} h_i\right\}$ represents the set of all nonrepetitive locations in $\mathcal{H}$.

Based on the above analysis, we provide a formal definition of location pair similarity.

Definition 4 (Location Pair Similarity): Suppose $\langle b_{i-1}, b_i \rangle$ and $\left\langle h_{j-1}, h_j \right\rangle$ are two location pairs on B and H, respectively; if they can satisfy the following conditions, they have location pair similarity:

(1) $\left|\pi\left(b_{i-1}\right) - \pi\left(h_{j-1}\right)\right| \le \delta_\pi, \left|\pi\left(b_i\right) - \pi\left(h_j\right)\right| \le \delta_\pi$
(2) $\left|P\left(b_i\right) - P\left(h_j\right)\right| \le \delta_P$
(3) $\left|T\left(b_{i-1}\right) - T\left(h_{j-1}\right)\right| \le \delta_T, \left|T\left(b_i\right) - T\left(h_j\right)\right| \le \delta_T$
(4) $A\left(b_{i-1}\right) = A\left(h_{j-1}\right), A\left(b_i\right) = A\left(h_j\right)$

where δ_π, δ_P and δ_T are thresholds of the difference of the access probability, transition probability, and time between two location pairs, respectively. Formulas (1), (2) and (3) ensure that the access probability, transfer probability and access time of two location pairs are close enough, and formula (4) ensures that corresponding locations of two location pairs are in the same time period. In particular, if δ_π, δ_P and δ_T are zero, the two location pairs reach the maximum similarity.

D. PROBABILISTIC SIMILARITY METRIC

In a k-anonymous group, intuitively, optimal fake trajectories should exactly have the same probabilistic model as the base trajectory. We propose the probabilistic similarity to measure the difference of probability distributions between the base trajectory and a fake trajectory. The smaller the difference is, the more similar the probabilistic similarity is. In this section, we first give two theorems and then utilize them to prove that the difference between two probability distributions is the expectation of differences between probabilistic features of their corresponding locations.

Theorem 1: Let X_0 be a random variable with the probability distribution $P\left(X_0\right)$. $X = \{X_i | i = 1, 2, \ldots, n\}$ is a set of random variables with the probability distribution $P\left(X_i\right)$. For each X_i, it has a probability distribution difference with X_0. Among them, the probability distribution difference between $P\left(X_i\right)$ and $P\left(X_0\right)$ is $d\left(X_0, X_i\right)$, and the probability of the probability distribution difference between $P\left(X_i\right)$ and $P\left(X_0\right)$ is $P_{X_i}\left(X_i\right)$. Therefore, the probability distribution difference $d\left(X_0, X_1, \ldots, X_i\right)$ between $X_0, X_1, \ldots, X_i$ and X_0 is the expectation of each difference, which is represented as follows.

$$d(X_0, X_1, \ldots, X_i) = \sum_{i=1}^{n} P_{X_i}(X_i) \cdot d(X_0, X_i) \tag{2}$$

Proof: $P_{X_i}\left(X_i\right)$ and $d\left(X_0, X_i\right)$ are functions of variables X_0 and X_i. For a given X_0, the two functions only depend on X_i. In this case, if X_i is also a given variable, then both $P_{X_i}\left(X_i\right)$ and $d\left(X_0, X_i\right)$ are constants. That is, if X_0 is a given variable, there is a one-to-one correspondence between $P_{X_i}\left(X_i\right)$ and $d\left(X_0, X_i\right)$. Thus, $P_{X_i}\left(X_i\right)$ can be seen as the probability of $d\left(X_0, X_i\right)$ in all probability distribution differences. Therefore, $d\left(X_0, X_1, \ldots, X_i\right)$ is the expectation of the probability distribution difference $d = \left\{d\left(X_0, X_i\right) | i = 1, 2, \ldots, n\right\}$ and the probability $d = \left\{P_{X_i}\left(X_i\right) | i = 1, 2, \ldots, n\right\}$.

Theorem 2: Let X' and Y' be two sets of random variables with the probability distribution $P(X' = x_i)$ and $P(Y' = y_i)$, where $X' = \{x_i | i = 1, 2, \ldots, m\}$ and $Y' = \{y_i | i = 1, 2, \ldots, m\}$. For x_i and y_i, the probability distribution difference between $P(X' = x_i)$ and $P(Y' = y_i)$ is $d\left(x_i, y_i\right)$, and the probability of the probability distribution difference between $P(X' = x_i)$ and $P(Y' = y_i)$ is $P_{Y'}(Y' = y_i)$. The difference in the probability distribution between X' and Y' is the expectation $d(X', Y')$ of all $d\left(x_i, y_i\right)$, where $d(X', Y')$ is represented as follows:

$$d(X', Y') = \sum_{i=1}^{m} P_{Y'}(Y' = y_i) \cdot d\left(x_i, y_i\right) \tag{3}$$

Proof: For $\forall x_{i'} \in X'$, according to **Theorem 1**, the difference between $x_{i'}$ and Y' is $\sum_{i=1}^{m} P_{Y'}(Y' = y_i, X' = x_{i'}) \cdot d\left(x_{i'}, y_i\right)$. Then, $d(X', Y') = \sum_{i=1}^{m} \sum_{i=1}^{m} P_{Y'}(Y' = y_i, X' = x_{i'}) \cdot d\left(x_{i'}, y_i\right)$. However, for $d(X', Y')$, if $i' \neq i$, $P_{Y'}(Y' = y_i, X' = 0)$ and $d\left(x_{i'}, y_i\right) = 0$. That is, $d(X', Y') = \sum_{i=1}^{m} P_{Y'}(Y' = y_i) \cdot d\left(x_i, y_i\right)$. Thus, we prove that the difference of probability distributions between X' and Y' is the expectation $d(X', Y')$ of all $d\left(x_i, y_i\right)$.

For $\forall C \in \mathcal{C}$ and a given B, the probability distribution difference and its probability between B and C only depend on C. Therefore, we define the probabilistic similarity between B and C as the expectation of the difference of probability distributions between B and C based on **Theorem 2**.

Definition 5: For $\forall C \in \mathcal{C}$ and a given B, the difference between the probabilities of c_i and b_i is $d_P\left(c_i, b_i\right)$, and the probability of the difference is $P_C(c_i)$. Then, the probabilistic similarity between B and C is defined as the expectation of the difference of all $d_P\left(c_i, b_i\right)$, namely,

$$\sum_{i=1}^{n} P_C(c_i) \cdot d_P\left(c_i, b_i\right) \tag{4}$$

The adversary's goal is to infer the user's real trajectories from published trajectories. To achieve this goal, in each equivalence probability class, the adversary always estimates a location (called estimation location) as the real location and calculates the possibility that it is the real location by exploiting the side information. In B_i, we assume that $\hat{c}_i$ is the estimation location. Without considering side information, the probability that c_i is the real location is $P(\hat{c}_i|c_i)$. In this case, the possibility that each location in B_i is estimated as the real location is the same. Thus, $P(\hat{c}_i|c_i) = \frac{1}{k}$. For c_i, the side information that c_i is the estimation location is that c_i is estimated as b_i, namely, the joint probability $P(c_i, b_i)$. Therefore, the possibility that c_i is estimated as b_i is $P(c_i, b_i) \cdot P(\hat{c}_i|c_i)$. The greater the probability is, the greater the probability that $\hat{c}_i$ is b_i. This means that the adversary thinks c_i is more similar to b_i. Therefore, we use this probability to calculate $P_C(c_i)$:

$$\begin{aligned} P_C(c_i) &= 1 - P(c_i, b_i) \cdot P(\hat{c}_i|c_i) \\ &= 1 - \pi(b_i) \cdot \pi(c_i) \cdot P(\hat{c}_i|c_i) \end{aligned} \tag{5}$$

For c_i and b_i, the closer their access probability and transition probability are, the more similar they are, and the smaller $d_P\,(c_i, b_i)$ is. We use Euclidean distance to measure the difference between them.

$$d_P\,(c_i, b_i) = \sqrt{(\pi(c_i) - \pi(b_i))^2 + (P(c_i) - P(b_i))^2} \tag{6}$$

According to the above formulas, we finally obtain the probabilistic similarity between B and C, which is represented as $sim_P\,(B, C)$.

$$sim_P\,(B, C) = \frac{1}{\sqrt{2n}} \cdot \sum_{i=1}^{n} P_C(c_i) \cdot d_P\,(c_i, b_i) \tag{7}$$

where $\frac{1}{\sqrt{2n}}$ is a constant used to normalize the probabilistic similarity such that each $sim_P\,(B, C)$ lies in [0, 1] in C.

E. GEOGRAPHIC SIMILARITY METRIC

Intuitively, we hope the published fake trajectories will be geographically indistinguishable from the base trajectory, i.e., the geographic distances between them are too close to be distinguished. We introduce the geographic similarity to measure the geographic distance between the base trajectory and the fake trajectory. The smaller the distance is, the more difficult it is to distinguish them geographically.

For $\forall C \in \mathcal{C}$ and given B, the geographical distance between B and C and its probability distribution (essentially, it is the probability of the geographical difference, called the geographical difference probability distribution) depend on C. Referring to **Definition 5**, we define the geographic similarity between B and C as the expectation of geographic distance between them.

Definition 6: For $\forall C \in \mathcal{C}$ and a given B, the geographic distance between c_i and b_i is $d_G(c_i, b_i)$, and the probability distribution of the geographical difference between them is $P_{[C,G]}(c_i)$. Then, the geographic similarity between B and C is defined as the expectation of the difference of all $d_G(c_i, b_i)$, namely,

$$\sum_{i=1}^{n} P_{[C,G]}(c_i) \cdot d_G\,(c_i, b_i) \tag{8}$$

For formula (8), we still compute $P_{[C,G]}(c_i)$ by $P(c_i, b_i) \cdot P(\hat{c}_i|c_i)$, which is expressed as follows.

$$P_{[C,G]}(c_i) = 1 - \pi(b_i) \cdot \pi(c_i) \cdot P(\hat{c}_i|c_i) \tag{9}$$

According to the above formulas, we obtain the geographic similarity $sim_G\,(B, C)$ between B and C.

$$sim_G\,(B, C) = \frac{1}{z_g} \cdot \sum_{i=1}^{n} (1 - P_{[C,G]}(c_i)) \cdot d_G\,(c_i, b_i) \tag{10}$$

where $\frac{1}{z_g}$ is a constant used to normalize the geographic similarity such that each $sim_G\,(B, C)$ lies in [0, 1]. z_g is the sum of geographic distances between locations in each equivalence probability class that has the maximum geographic distance from the base location and the base location.

$$z_g = \sum_{i=1}^{n} \underbrace{max}_{c_i \in B_i} \{d_G\,(c_i, b_i)\} \tag{11}$$

V. GENERATING FAKE TRAJECTORIES BY RECOMBINING LOCATIONS

This section mainly describes the detailed algorithms for generating k-1 fake trajectories by using the *PLM*.

A. SAMPLING HISTORICAL TRAJECTORIES

In this section, we select sampling trajectories meeting the conditions of **Definition 4** from historical trajectories. To improve retrieval efficiency, we partition the database into two areas: location area and historical trajectory area. The location area stores location data represented by $\mathcal{L}$, and the historical trajectory area stores users' historical trajectories, as shown in Fig. 3 and Fig. 4. We divide the location area into a full quad-tree with n_a layers and retrieve it using a grid-based approach.

In Fig. 3, vector $Q = (q_1, q_2, \ldots, q_{n_a})$ represents the index of the block in the quad-tree. In Q, q_i represents the i^{th} layer area of the quad-tree, and stores locations in the time period a_i. The value of q_i is 1, 2, 3 or 4, indicating that it can store the locations of their access probabilities that lie in $[0, 0.25)$, $[0.25, 0.5)$, $[0.5, 0.75)$ and $[0.75, 1]$ during a_i. The index of area A is $Q = (1, 2, \ldots, 0, 0)$, which stores the location of their access probabilities that lie in $[0.25, 0.5)$ during a_2. Moreover, $Q(\mathcal{L}_l)$ denotes the index of the block where $\mathcal{L}_l$ locate, $\hat{Q}(\mathcal{L}_l)$ expresses the subblock with the same father block as $Q(\mathcal{L}_l)$, and $\hat{Q}_P(\mathcal{L}_l)$ represents the probability interval of $\hat{Q}(\mathcal{L}_l)$, such as $\hat{Q}_P(\mathcal{L}_l) = [0, 0.25)$.

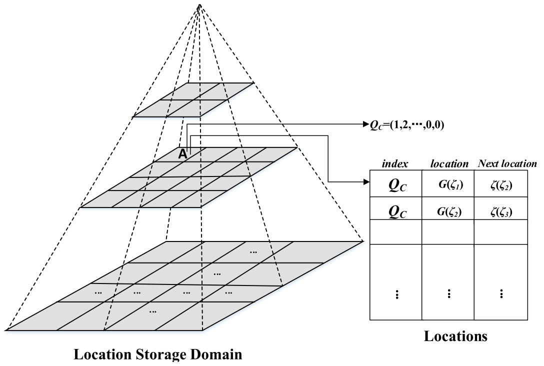


FIGURE 3. Location domain.

In the location area, $(Q(\mathcal{L}_l), G(\mathcal{L}_l), \mathcal{L}(\mathcal{L}_{l+1}))$ denotes the data structure of $\mathcal{L}_l$, where $G(\mathcal{L}_l)$ is the geographic coordinate of $\mathcal{L}_l$, and $\mathcal{L}(\mathcal{L}_{l+1})$ is the set of locations where $\mathcal{L}_l$ has arrived at the next instant. In the historical trajectory area, $(Q_c(h_j), h_j)$ represents locations on trajectory H, as shown in Fig. 4.

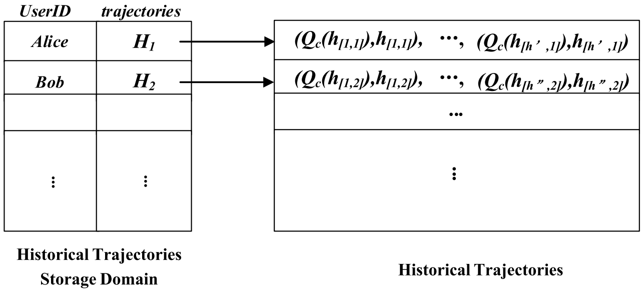

UserID
trajectories
Alice
H_1
Bob
H_2
⋮
⋮
$(Q_c(h_{[1,1]}),h_{[1,1]})$, ⋯, $(Q_c(h_{[h',1]}),h_{[h',1]})$
$(Q_c(h_{[1,2]}),h_{[1,2]})$, ⋯, $(Q_c(h_{[h'',2]}),h_{[h'',2]})$
⋯
⋮
Historical Trajectories Storage Domain
Historical Trajectories

FIGURE 4. Historical trajectories domain.

For B, we select sampling trajectories meeting the conditions of **Definition 4** from the two areas. Algorithm 1 shows the formal description of trajectory sampling. It first selects a location pair $\langle b_i, b_{i+1}\rangle$ and then determines the location area $\hat{Q}(b_i)$ based on b_i, where locations in $\hat{Q}(b_i)$ have the same time period as b_i and satisfy formula (1) of **Definition 4**. It limits the search range to quad-tree blocks that are the same time period as b_i and whose access probability of locations meets the conditions of **Definition 4** and reduces unnecessary retrieval. Next, it selects the location $\mathcal{L}_l$ from $\hat{Q}(b_i)$ and $Q(b_i)$, and selects $\mathcal{L}_{l+1}$ to form the location pair $\langle \mathcal{L}_l, \mathcal{L}_{l+1}\rangle$ to be similar to $\langle b_i, b_{i+1}\rangle$. This process mainly selects a location pair similar to $\langle b_i, b_{i+1}\rangle$ from the search block. Then, a trajectory containing a location pair $\langle \mathcal{L}_l, \mathcal{L}_{l+1}\rangle$ similar to $\langle b_i, b_{i+1}\rangle$ is selected from the historical trajectories block as a sampling trajectory. The above steps are repeated until each location pair on B has at least k-1 similar location pairs, and the obtained trajectories are sampling trajectories.

B. CLUSTERING THE SAMPLING LOCATION

We cluster the sampling locations from sampling trajectories set S in Algorithm 1 into different equivalent probability classes by the base location. Algorithm 2 shows the formal description of location sampling. First, the sampling locations in the same equivalent probability class as b_1 and b_2 are aggregated into B_1 and B_2, respectively. For $\forall H \in S$, Algorithm 2 traverses each location pair in H starting from $\langle h_1, h_2\rangle$. If a location pair is similar to $\langle b_i, b_{i+1}\rangle$ or there is no such location pair, the next trajectory is retrieved until all trajectories in S are traversed, and the sampling locations are added to B_i and B_{i+1}, respectively. Second, to aggregate the sampling locations into B_i and $B_{i+1} (i \geq 2)$, Algorithm 2 no longer traverses all trajectories in S, but only those that contain locations in B_i. During the process, Algorithm 2 traverses each location pair on H. Once a location pair is similar to $\langle b_i, b_{i+1}\rangle$ or there is no such location pair, the next trajectory is retrieved until all trajectories are traversed and sampling locations are added to B_i and B_{i+1}, respectively. The above process is repeated until B_n is aggregated. Finally, Algorithm 2 returns the set of equivalence probability classes B.

Algorithm 1 Trajectory Sampling

```
Input: base trajectory $B$, historical trajectory set $\mathcal{H}$,
       nonrepetitive location set $\mathcal{L}$, location pair
       similarity threshold $\delta_\pi$, $\delta_P$, $\delta_T$
Output: sampling trajectory set $S$
$S \leftarrow \varnothing$;
for all $b_i \in B$ do
    Num $(\langle b_i, b_{i+1}\rangle) = 0$;
    Calculate $\hat{Q}(b_i)$ that
    $|\pi(b_i) - \delta_\pi| \in \hat{Q}_P(b_i)$ or $\pi(b_i) + \delta_\pi \in \hat{Q}_P(b_i)$;
    for all $\mathcal{L}_l \in Q(b_i) \cup \hat{Q}(b_i)$ do
        if Num $(\langle b_i, b_{i+1}\rangle) \geqslant k - 1$ then
            exit
        else if $\langle b_i, b_{i+1}\rangle$ and $\langle \mathcal{L}_l, \mathcal{L}_{l+1}\rangle$ meet the
        condition of Definition 4 then
            for all $H \in \mathcal{H}$ do
                if $\langle b_i, b_{i+1}\rangle$ and $\langle \mathcal{L}_i, \mathcal{L}_{i+1}\rangle \in H$ meet the
                condition of Definition 4 then
                    $S \leftarrow H$;
                    Num $(\langle b_i, b_{i+1}\rangle) += 1$;
                    exit
return $S$
```

C. CONSTRUCTING THE LOCATION PAIR GRAPH

We need to randomly select a sampling location from each equivalence probability class and combine them into a fake trajectory. However, this method has a higher complexity. The reason is that some nonexistent location pairs in historical trajectories may be selected to construct trajectories. As shown in Fig. 5, the solid lines and the dotted lines, respectively, represent existing location pairs and nonexistent location pairs in the historical trajectories. To this end, we build the location pair graph $G = (\mathcal{A}, E, W)$.

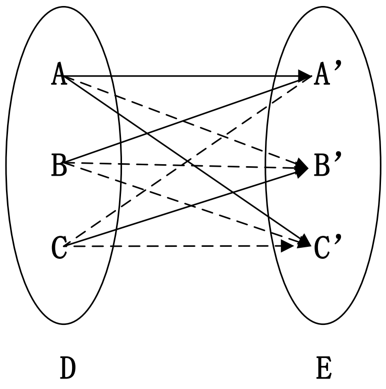


FIGURE 5. Example of constructing fake trajectories.

Algorithm 2 Sampling Location Clustering

```
Input: base trajectory $B$, sampling trajectory set $S$,
       location pair similarity threshold $\delta_\pi$, $\delta_P$, $\delta_T$
Output: equivalent probability class set $\mathcal{A}$
$\mathcal{A} \leftarrow \varnothing$;
for all $b_i \in B$ do
    if $i = 1$ then
        $B_1 \leftarrow b_1, B_2 \leftarrow b_2$;
        for all $H \in S$ do
            for all $h_j \in H$ do
                if $\langle h_j, h_{j+1} \rangle$ and $\langle b_1, b_2 \rangle$ meets the
                condition of Definition 4 then
                    $B_1 \leftarrow h_j, B_2 \leftarrow h_{j+1}$;
                    exit
        $\mathcal{A} \leftarrow B_1, \mathcal{A} \leftarrow B_2$;
    else if $i \geqslant 2$ then
        $B_i \leftarrow b_i, B_{i+1} \leftarrow b_{i+1}$;
        for all $H \in S$ do
            for all $h_j \in (H \cap B_i)$ do
                if $\langle h_j, h_{j+1} \rangle$ and $\langle b_i, b_{i+1} \rangle$ meets the
                condition of Definition 4 then
                    $B_i \leftarrow h_j, B_{i+1} \leftarrow h_{j+1}$;
                    exit
    $\mathcal{A} \leftarrow B_i, \mathcal{A} \leftarrow B_{i+1}$;
return $\mathcal{A}$
```

In G, $\mathcal{A}$ is the set of all equivalent probability locations from Algorithm 2, and each location is represented as a node of G; b'_i is a location in B_i; $E(b'_i, b'_{i+1})$ is a directed edge connecting b'_i and b'_{i+1}; $W(b'_i, b'_{i+1})$ is the weight of the edge $E(b'_i, b'_{i+1})$, which is a binary group composed of probabilistic similarity and geographic similarity between $< b'_i, b'_{i+1} >$ and $\langle b_i, b_{i+1} \rangle$.

We construct G by analyzing E and W among $\mathcal{A}$. Algorithm 3 shows the formal description of constructing the location pair graph. First, it traverses each class B_i in $\mathcal{A}$, and selects a location from B_i and B_{i+1}, respectively, to constitute a location pair $< b'_i, b'_{i+1} >$. Then, Algorithm 3 judges whether the location pair is in historical trajectories. If so, Algorithm 3 adds $E(b'_i, b'_{i+1})$ and $W(b'_i, b'_{i+1})$ to G. This process is repeated until it traverses each class in $\mathcal{A}$. Finally, Algorithm 3 returns the location pair graph G.

D. GENERATING FAKE TRAJECTORIES

In this section, we generate k-1 fake trajectories similar to the base trajectory in the mobility pattern. A fake trajectory is a sequence of locations that are selected from each equivalence probability class in G, and two adjacent locations are connected by an adjacent directed edge. For each sequence, its i^{th} node is a location in the probability class B_i. In this way, we turn the goal of generating k-1 fake trajectories into the problems reconstructing sequences in G. Among them, the efficiency of sequence reconstruction and the effect of privacy protection are the problems that need to be solved.

Algorithm 3 Location Pair Graph Construction

```
Input: base trajectory $B$, equivalent probability class set
       $\mathcal{A}$
Output: location pair graph $G$
$E \leftarrow \varnothing, W \leftarrow \varnothing$;
$G = (\mathcal{A}, E, W)$ for all $b_i \in B$ do
    for all $B_i \in B$ do
        for each $b'_i \in B_i$ do
            if $< b'_i, b'_{i+1} > \in \mathcal{H}$ then
                $E \leftarrow E(b'_i, b'_{i+1})$;
                $W \leftarrow W(b'_i, b'_{i+1})$;
return $G$
```

We consider improving the efficiency of sequence reconstruction from two aspects: (1) Delete locations that do not meet the privacy protection requirements for the base trajectory (Section V-F). In G, there may be some directed edges with a weight greater than δ_B or δ_G. Then, any sequence of locations containing the directed edges cannot satisfy the privacy protection requirement for the base trajectory, so we need to delete these locations before the sequence reconstruction. (2) Delete sampling trajectories. Some reconstructed sequences are also sampling trajectories, which obviously do not meet the privacy protection requirement for the sampling trajectory, so we need to delete them.

After obtaining a sequence of locations, we need to delete those that do not satisfy the privacy protection requirements for the base trajectory or the sampling trajectory. If the probabilistic similarity of a trajectory does not satisfy formula (12) and the geographic similarity of a trajectory does not satisfy formula (13), it does not satisfy the privacy protection requirement for the base trajectory. If the trajectory does not satisfy formula (14), it does not satisfy the privacy protection requirement for the sampling trajectory.

According to the above analysis, we design an algorithm to generate k-1 fake trajectories. Algorithm 4 shows the formal description of generating fake trajectories. First, it retrieves all directed edges in G. If the probabilistic similarity of the weight of $E(b'_i, b'_{i+1})$ is greater than δ_B or the geographic similarity is greater than δ_G, $E(b'_i, b'_{i+1})$ is deleted. Second, after obtaining the deleted location pair graph, it randomly generates a fake trajectory F. If F is B or a sampling trajectory, it continues to randomly generate another fake trajectory. Then, Algorithm 4 uses the Privacy Test 1 algorithm and Privacy Test 2 algorithm to conduct a privacy test for F. If the test results are all *True*, it selects F as a fake trajectory. Finally, Algorithm 4 repeats the above process until k-1 fake trajectories are generated.

Algorithm 4 Generating Fake Trajectories

```
Input: base trajectory $B$, location pair graph $G$, sampling
       trajectory set $S$, probabilistic similarity threshold
       $\delta_B$, geographic similarity threshold $\delta_G$, sampling
       trajectory privacy requirement threshold $\delta_S$
Output: fake trajectory set $\mathcal{F}$
$\mathcal{F} \leftarrow \varnothing, G' \leftarrow \varnothing, n(\mathcal{F}) \leftarrow 0$;
Remove all $E(b'_i, b'_{i+1})$ for which $W(b'_i, b'_{i+1})$ do not
satisfy conditions (12) or (13);
$G' \leftarrow G$;
while $n(\mathcal{F}) < k - 1$ do
    Randomly generates a trajectory $F$;
    if $F = B$ or $F \in S$ then
        $n(\mathcal{F}) = n(\mathcal{F})$
    else
        $\Omega \leftarrow PrivacyTest1, \Omega' \leftarrow PrivacyTest2$;
        if $\Omega = True$ and $\Omega' = True$ then
            $\mathcal{F} \leftarrow \mathcal{F} \cup F$;
            $n(\mathcal{F}) = n(\mathcal{F}) + 1$
return $\mathcal{F}$
```

E. PRIVACY THREAT MODEL

In this paper, we focus on the scenario of anonymous trajectory publication. The scenario has three features: (1) All fake trajectories in the published k-anonymous group have the same ID as the base trajectory. (2) Side information grasped by the adversary is the trajectory mobility patterns of all users. (3) The adversary can obtain the published k-anonymous group through various channels. Thus, we consider two types of privacy threats faced by users. The first type is against the inference attack on the base trajectory. By comparing the mobility pattern of the base trajectory with that of all of the trajectories in the k-anonymous group, the adversary matches trajectories similar to the base trajectory in terms of probabilistic features and geographical features of the base trajectory.

The second type is against the inference attack on the sampling trajectory. Since locations on a fake trajectory come from sampling trajectories, there may be some locations that are the same as the locations on a sampling trajectory. Moreover, the more such locations on a fake trajectory there are, the more privacy a sampling trajectory leaks. Therefore, it is necessary to protect sampling trajectories by limiting the number of such locations.

To address the above two threats, we define two privacy protection requirements for both base trajectories and sampling trajectories (see V-F, G). In the *LRM*, we need to test the privacy requirement of the fake trajectory to meet these two requirements.

F. PRIVACY PROTECTION REQUIREMENTS FOR THE BASE TRAJECTORY

Generally, when we say that fake trajectories and base trajectories have the same mobility pattern, we mean that the probabilistic similarity and geographic similarity of two trajectories reach a certain range. However, the LRM may release some fake trajectories that are different from the base trajectory in the mobility pattern, causing the base trajectory to be identified. Therefore, we need to generate fake trajectories that are the same mobility pattern as the base trajectory to protect the base trajectory privacy. We call it the privacy protection requirement for the base trajectory.

In the attack model, the privacy protection requirements for the base trajectory need to consider two aspects: (1) The probabilistic similarity between the base trajectory and each fake trajectory F is less than the threshold δ_B. The rationale is that the smaller the probabilistic similarity is, the more difficult it is to distinguish the probabilistic features of them. (2) The geographic similarity between the base trajectory and each of the fake trajectories F is smaller than the threshold δ_G. Similarly, the rationale is that the smaller the geographic similarity is, the more difficult it is to distinguish the geographic features of them. In addition, F should not be the same as the base trajectory in geographic features. That is, δ_G is not equal to 0. We ensure that the mobility pattern between the base trajectory and each fake trajectory is similar by setting the above aspects (1) and (2).

$$sim_P(B, F) \leq \delta_B \tag{12}$$

$$0 < sim_G(B, F) \leq \delta_G \tag{13}$$

Formula (12) and (13) are criteria for verifying the similarity of the mobility pattern. Therefore, we can construct a fake trajectory meeting these criteria to solve the privacy threat of the base trajectory.

We design a privacy test algorithm to ensure that fake trajectories meet this standard. Algorithm 5 shows the formal description of it. It judges the relationships of $sim_P(B, F)$ and $sim_G(B, F)$. If $sim_P(B, F) \leq \delta_B$ and $0 < sim_G(B, F) \leq \delta_G$, it satisfies the privacy protection requirement for the base trajectory, so the privacy test result Ω is *True*.

Algorithm 5 Privacy Test 1

```
Input: probabilistic similarity $sim_P(B, F)$,
       geographic similarity $sim_G(B, F)$,
       probabilistic similarity threshold $\delta_B$,
       geographic similarity threshold $\delta_G$
Output: test result $\Omega$
if $sim_P(B, F) \leq \delta_B$ and $0 < sim_G(B, F) \leq \delta_G$ then
    $\Omega \leftarrow True$
return $\Omega$
```

G. PRIVACY PROTECTION REQUIREMENT FOR THE SAMPLING TRAJECTORY

In the *LRM*, the sampling trajectory also faces a privacy threat. To explain the reason, we consider a 2-anonymous group containing the base trajectory L_a and the sampling trajectory L'_a for Alice and another 2-anonymous group containing the base trajectory L_b and the sampling trajectory L'_b

for Bob. We assume that L_b is a sampling trajectory in the process of protecting L_a. Then, some locations on L'_a may be from L_b. In this case, L'_a and L_b will have some of the same locations. Although L_b cannot be obtained by the adversary using the first privacy threat model, it can suffer from the inference attack on the sampling trajectory. If the number of the same locations of L'_a and L_b reaches the threshold, especially when they are completely the same, the adversary can identify L_b. Therefore, we must ensure that the adversary cannot identify the sampling trajectory through the published k-anonymous group. We call it the privacy protection requirement for the sampling trajectory.

For a fake trajectory and a sampling trajectory, the greater the proportion is of the same locations contained in the fake trajectory, the more likely the fake trajectory is identified as the sampling trajectory. Therefore, for each F, it should satisfy the relationship:

$$\frac{n(F,H)}{n(F)} \leq \delta_{\mathcal{S}} \tag{14}$$

In formula (14), $n(F, H)$ is the number of the same locations of F and H, $n(F)$ is the number of locations in F, and $\delta_{\mathcal{S}}$ is the threshold of the privacy requirement protection for the sampling trajectory and is constant. In other words, when F and all H satisfy formula (14), the adversary is not able to identify the sampling trajectory through F.

Similarly, we design a privacy test algorithm to ensure that fake trajectories meet this standard. Algorithm 6 shows the formal description of it. It determines whether an F and all trajectories H in S meet condition (14). If they satisfy the privacy protection requirement for the sampling trajectory, the privacy test result Ω' is *True*.

Algorithm 6 Privacy Test 2

```
Input: sampling trajectory set $\mathcal{S}$, fake trajectory $F$,
       sampling trajectory privacy requirement
       threshold $\delta_{\mathcal{S}}$
Output: test result $\Omega'$
if all $H \in \mathcal{S}$ meets condition of $\frac{n(F,H)}{n(F)} \leq \delta_{\mathcal{S}}$ then
    $\Omega' \leftarrow True$
return $\Omega'$
```

H. TIME COMPLEXITY

In our scheme, most of the computation time is spent on Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4. For a base trajectory B, we assume that the number of historical trajectories and locations stored in $Q(b_i) \cup \hat{Q}(b_i)$ is m_H and m_i, respectively. Here, $Q(b_i) \cup \hat{Q}(b_i)$ are the blocks that meet conditions $|\pi(b_i) - \delta_\pi| \in \hat{Q}_P(b_i)$ or $\pi(b_i) + \delta_\pi \in \hat{Q}_P(b_i)$. If B requests a k-anonymous privacy protection, for $\langle b_i, b_{i+1} \rangle$, Algorithm 1 needs to select at least $k-1$ of $m_i \times m_{i+1}$ location pairs and select one out of m_H historical trajectories for each selected location pair. In the worst case, the time complexity of Algorithm 1 is $O(m_H \times \sum_{i=1}^{n-1} m_i \times m_{i+1})$. Assume that Algorithm 1 returns h sampling trajectories and the j^{th} trajectory h_j contains n_{h_j} sampling locations. In the worst case, Algorithm 2 needs to traverse each location in every sampling trajectory and cluster the sampling locations into n equivalent probability classes. Therefore, the time complexity of Algorithm 2 is $O((n-1) \times h \times n_{h_j})$. Assume that B_i that Algorithm 2 returns contains n_{B_i} locations. For the location pair graph, in the worst case, each location in B_i connects to every location in B_{i+1}. Consequently, the time complexity of Algorithm 3 is $O(\sum_{i=1}^{n-1} n_{B_i} \times n_{B_{i+1}})$. For the location pair graph that Algorithm 3 returns, we need to randomly select a sampling location from each equivalence probability class and combine them into a fake trajectory. Thus, in the worst case, the time complexity of Algorithm 4 is $\prod_{i=1}^{n} n_{B_i}$.

VI. EVALUATION SETUP

In our experiment, we evaluate the performance of the *LRM* through the evaluation setup (Section VI) and evaluation results (Section VII).

A. DATASET

The data we use for the evaluation are from a real GPS trajectory dataset, called GeoLife. The GeoLife dataset was collected in (Microsoft Research Asia) the GeoLife project (see [40]), in which GPS was used to collect the data, such as latitude, longitude and altitude, etc., and has been widely used in many studies, such as location privacy [41], data mining [42], and location recommendation [43], etc. It has recorded the GPS trajectory of 182 users with 17,621 trajectories in a period of over five years (from April 2007 to August 2012). In this paper, we select five valid fields—user ID, longitude, longitude, date and time—from GeoLife as our dataset, called the Raw dataset. We run our algorithms on the Raw dataset to generate a trajectory k-anonymous group for data release.

In the Raw dataset, each trajectory is a sequence of timestamped points, each of which contains all five fields. For all trajectories, 91.2 percent of locations are positioned every 5∼10 seconds. That is, multiple consecutive GPS points may refer to the same place, which causes that a place that is actually only accessed once is mistaken for frequent access. Thus, we add two fields to the Raw dataset, location and street, which are the house number of a location and the street where it is located, respectively. In addition, for multiple consecutive GPS points representing one place, we sample the one with the smallest difference from the average time of all points for our evaluation, and reduce the others. Then, we extract approximately 30 days of trajectories of 22 users (the days of a small number of users are more than 30 days or less than 30 days) from the Raw dataset and obtain a new dataset called the Preprocessed Dataset for the evaluation.

B. EXPERIMENTAL SETTINGS

To set the access period, we need to know how many days each user visited his most frequently visited street in approximately 30 days. Considering the different sampling days for each user in the Preprocessed Dataset, we use the frequency $f \in [0, 1]$ to evaluate the access period. For example, $f = 0.5$ means a user visits his most frequently accessed street every other day. Due to different positioning times, however, simply summing up the time of each location in the sequence of locations periodically accessed is insufficient because it is difficult for the same sequence of locations to repeatedly appear. For example, one location accessed by a user with less than the positioning time will not be positioned. Thus, we use the time of the street accessed instead of the sequence of locations to evaluate the access period. The result is shown in Fig. 6.

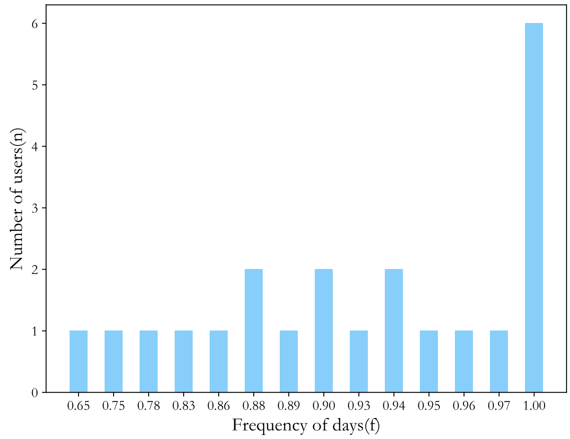
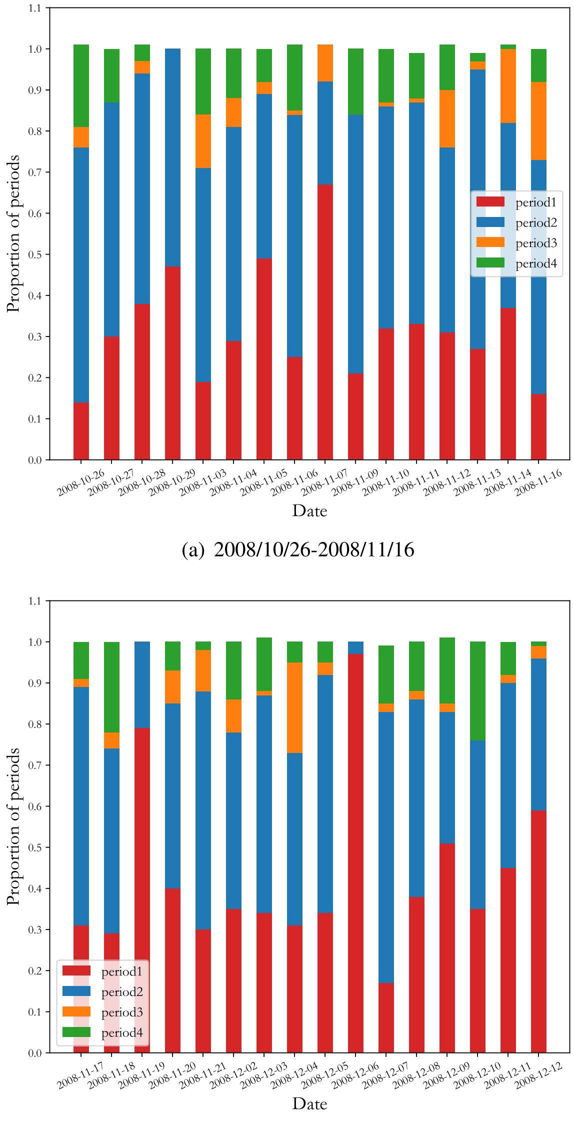


FIGURE 6. Statistics of the frequency of days for which the street was accessed most frequently by users.

Fig. 6 shows that frequencies of all users are in the interval [0.65, 1], and 21 users are in the interval [0.75, 1]. It means the number of users with an access period of no more than 1.5 days is 22, and the number of users with a maximum access period of 1.3 days is 21. Thus, we set the access period as 1 day.

Intuitively, people's activity patterns during a period in an access period are stable. Based on it, we analyze the distribution of users' locations in different time periods of every access period, as shown in Fig. 7. In Fig. 7, there are 4 different time periods—period 1, period 2, period 3, and period 4—which represent that their time periods in an access period are between 12 a.m. and 6 a.m., between 6 a.m. and 3 p.m., between 3 p.m. and 8 p.m. and between 8 p.m. and 12 a.m. respectively. For most of the access periods, because people go out to work during the day, period 2 is the time period in which users have the most frequent activity, and its ratio steadily falls in the range [0.4, 0.6]. In addition, because people reduce their activities in the afternoon and evening, ratios of period 1, period 3, and period4 steadily fall in the range [0.1, 0.3], [0, 0.1] and [0.1, 0.2], respectively, and all are smaller than period 2. Thus, we partition the time in every access period into 4 different time periods: period 1-period 2-period 3-period 4.



(a) 2008/10/26-2008/11/16

(b) 2008/11/17-2008/12/12

FIGURE 7. Distribution of users' locations in different time periods of every access period.

In the Preprocessed Dataset, each trajectory is a user's sequence of GPS points within one day or more. We consider dividing each trajectory into multiple trajectories with fewer locations. For each user, there are two states: uniform motion and staying in one place. We consider the time t_0, the time at which a user stays in one place, as the criterion for dividing trajectories. If the time interval of two adjacent locations $t \leq t_0$, they belong to the same trajectory; otherwise, they do not. Suppose that n locations are divided into different trajectories; then, the number of trajectories is the function $f(t_0)$ for t_0. For the trajectory containing the most locations among $f(t_0)$ trajectories, the number of its locations is the function $g(t_0)$ for t_0.

The division of trajectories needs to balance the privacy level and the utility. The privacy level measures privacy that trajectories leak and the utility measures privacy that trajectories contain. The smaller t_0 is, the larger $f(t_0)$ is, and the less privacy that each trajectory leaks. We assume that the average privacy of trajectories is 1, and then its average privacy level is $1-\frac{1}{f(t_0)}$. In addition, the larger t_0 is, the larger $g(t_0)$ is, and the more average utility that trajectories contain. We use $\frac{1}{g(t_0)}$ to represent the average utility of trajectories. If $1-\frac{1}{f(t_0)}=\frac{1}{g(t_0)}$, the privacy level and the utility are balanced. Thus, we define the time interval in which the privacy level and the utility are balanced as t_0. In Fig. 8, the time interval of balance t_0 is 51 s. Based on t_0, we divide the trajectory on 12/10/2018 into 51 trajectories, and the trajectory that has the most locations contains 51 locations. Analogous to Fig. 8, we divide all the trajectories in the Preprocessed Dataset and obtain the trajectory set $\mathcal{H}$.

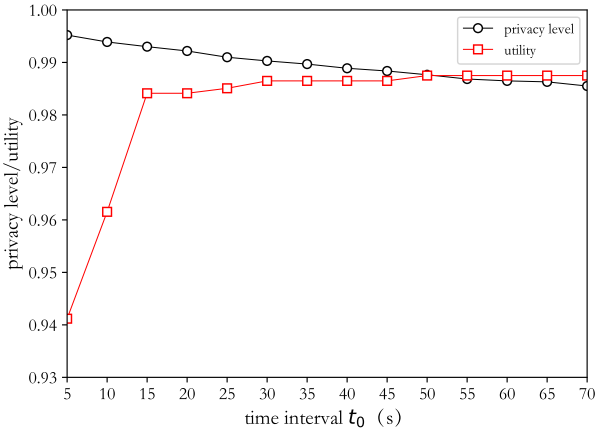


FIGURE 8. Division of user trajectories on 12/10/2018.

In our experiment, we randomly sampled 10 trajectories, which are in the same access period, as base trajectories from the Preprocessed Dataset. For each base trajectory, we run Algorithm 1 to select sampling trajectories and build probabilistic models for them. For all sampling trajectories and the base trajectory, we use Algorithm 2 to partition locations from them into distinct equivalence probability classes, and Algorithm 3 is implemented to build a location pair graph for them. Based on the location pair graph, we run Algorithm 4 to randomly generate trajectories. For each trajectory, if two results from Algorithm 5 and Algorithm 6 are True, it is a fake trajectory that meets the privacy requirements of the base trajectory and sampling trajectories. The implementation procedure of our experiment is shown in Fig 9. Finally, we generate 10 to 12 k-anonymous groups with an anonymity level of 4 to 15 by setting the parameters in Table 2.

VII. EVALUATION RESULTS

A. THREAT SCENARIO SETUP

In this scenario, users u and u', respectively, send their real trajectories l and l' to a service provider and share a location-based service. The service provider receives l and l' and releases a k-anonymous group for each of them. In addition, l' is a sampling trajectory exactly sourced from the process of privacy protection for l.

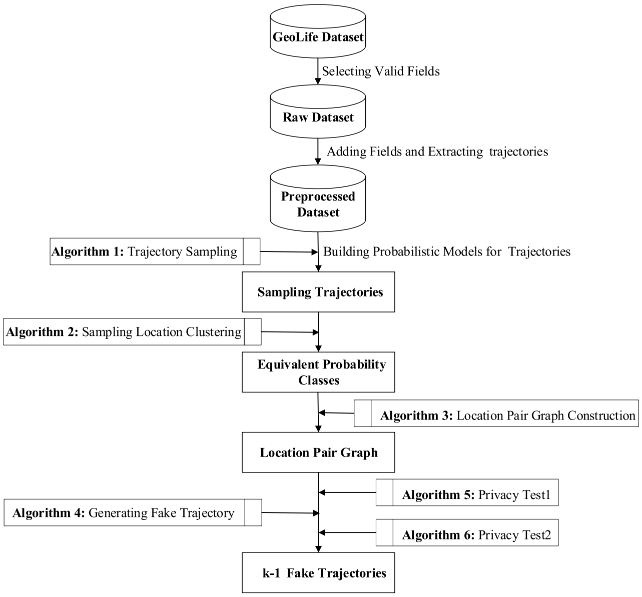

GeoLife Dataset
Selecting Valid Fields
Raw Dataset
Adding Fields and Extracting trajectories
Preprocessed Dataset
Algorithm 1: Trajectory Sampling
Building Probabilistic Models for Trajectories
Sampling Trajectories
Algorithm 2: Sampling Location Clustering
Equivalent Probability Classes
Algorithm 3: Location Pair Graph Construction
Location Pair Graph
Algorithm 5: Privacy Test1
Algorithm 4: Generating Fake Trajectory
Algorithm 6: Privacy Test2
k-1 Fake Trajectories

FIGURE 9. Implementation procedure of our experiment.

TABLE 2. Experimental parameters.

parameters	range
δ_π	0.004
δ_P	0.5
δ_t	30 min
δ_B	0.3
δ_G	0.3
$\delta_\mathcal{S}$	0.5
Number of base trajectories	10
Anonymity level	4-15

For u, the adversary can legally obtain his k-anonymous groups and all sampling trajectories such as l'. Analogous to [3], [7], we assume that the adversary knows the user's mobility pattern and the *LRM* mechanism. For example, u is a real user and his real trajectory l is hidden in the k-anonymous groups. However, he does not know how does the *LRM* works (we assume that service providers are trustworthy).

To resist the inference attacks on the base trajectory, it is necessary to ensure that the mobility patterns of all fake trajectories and the base trajectory in a k-anonymous group are indistinguishable. Suppose that a, b, c, d is a k-anonymous group, where a is the base trajectory and others are fake trajectories. Although the adversary cannot identify the base trajectory by analyzing its mobility pattern, he may identify fake trajectories in the k-anonymous group (e.g., no similarity to the base trajectory in the mobility pattern is considered a fake trajectory). It is the reason why we propose the geographic similarity and the probabilistic similarity. They ensure that each fake trajectory is similar to the base trajectory in the mobility pattern so that the *LRM* has a higher privacy level (see Fig. 10).

There are two types of fake trajectories used by the adversary to identify sampling trajectories. One is from the same

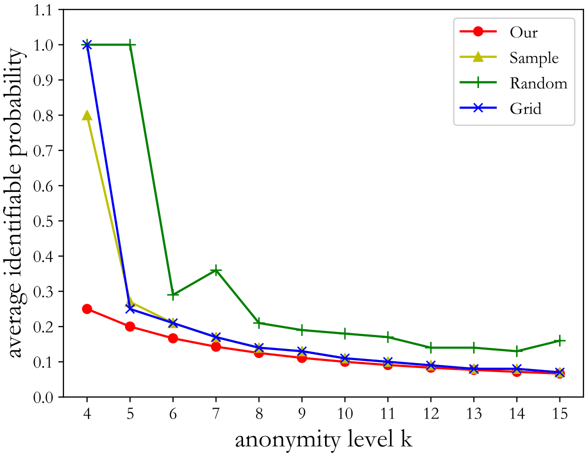


FIGURE 10. **The privacy level of the base trajectory *vs*. *k*.**

process as the sampling trajectory and the other is not, but its part regarding locations is the same as that in the sampling trajectory. Both of them can be used by the adversary to launch inference attacks. For the second, all historical trajectories are required to satisfy the sampling trajectory privacy requirements. Therefore, we only consider the sampling trajectory privacy requirements that satisfy the first.

B. EXISTING TRAJECTORY *k*-ANONYMITY METHODS

In this section, we study existing typical trajectory k-anonymity methods [12], [14], [15], [17], [21], [25], [31], [44], which protect trajectory privacy by generating k-anonymous groups. However, these methods differ in how fake trajectories are generated. According to the difference, existing typical methods can be divided into three categories.

- *Random [12], [25]: Generate a fake trajectory by randomly rotating the base trajectory [14] or randomly generating k-1 fake locations for each base location [36]. Thus, some fake locations in a fake trajectory may be hard-to-reach locations.*
- *Sample [14], [15], [17], [31]: Select one historical trajectory of a user as a fake trajectory. Thus, all fake trajectories in the k-anonymous group are historical trajectories.*
- *Grid [21], [44]: For each base location, randomly select k-1 locations that have the same probability as it does as the fake locations. In [21], [44], each location is a grid and is given a query probability of a user. Thus, all fake locations are historical locations.*

We respectively select [14], [21], [25] from the three methods to compare with our methods (LRM). Reference [25] randomly generates $k - 1$ fake locations without the spatiotemporal correlation for each base location. Thus, some fake locations may still be hard-to-reach locations. Reference [14] selects $k - 1$ independent historical trajectories that satisfy the privacy criterion from real-world datasets as fake trajectories. Reference [21] selects $k-1$ real locations that have the highest uncertainty of identifying a user as fake locations for each base location independently. Therefore, the fake locations corresponding to different base locations are uncorrelated. For comparison purposes, these methods use the same 10 base trajectories as the *LRM* for k-anonymity protection. For the same base trajectory, we run three algorithms separately to generate k-anonymous groups with the same number and anonymity level as the *LRM*. In particular, for [21], we chose k fake locations for each base location in each base trajectory and recombine locations that sample from each k fake location into a trajectory.

We then use these trajectories to evaluate the performance of the four methods in terms of four metrics (see Section VII-C). (i) Base trajectory privacy. Counting the number of base trajectories identified under the inference attack on the base trajectory to evaluate the capability of the four methods for protecting the base trajectory privacy. (ii) Sampling trajectory privacy. Counting the number of sampling trajectories unidentified under the inference attack on sampling trajectories to evaluate the capability of the *LRM* and the *Sample* method for protecting sampling trajectory privacy. (iii) Service provider utility. Counting the number of trajectories released at the same level of privacy for all methods to measure the service provider utility. (iv) Data availability. Analyze the hard-to-reach locations in k-anonymous groups and the change of probabilities of locations to evaluate the data availability for all methods. (Section VII-D presents the evaluation results.)

C. PRIVACY METRIC

The metric to quantify the privacy level of the base trajectory is the probability that the base trajectory is identified (called *identifiable probability*). Although the adversary can easily obtain k-anonymous groups, he cannot identify fake trajectories that are similar to the base trajectory in the mobility pattern. For a base trajectory, if there are $m(m \leq k)$ trajectories similar to it in its k-anonymous group, the *identifiable probability* is $\frac{1}{m}$. The larger m is, the more difficult identifying the base trajectory is, and the higher the privacy level of the base trajectory is. We evaluated the privacy level of the *LRM* and all three methods.

We use the ratio of sampling trajectories that are not identified (called *unidentifiable ratio*) to quantify the privacy level of sampling trajectories. Assume that there are n_0 sampling trajectories in the process of trajectory k-anonymity, and $n(n \leq n_0)$ among them cannot be identified by the adversary; then, the *unidentifiable ratio* is $\frac{n}{n_0}$. In our experiments, we only evaluated the privacy level of sampling trajectories of the *LRM* and the *Sample* scheme. (*Random* and *Grid* do not use sampling trajectories to generate fake trajectories). Unlike the *LRM*, each fake trajectory in the *Sample* scheme is a sampling trajectory. The more locations in each fake trajectory that are released, the more likely it is to be identified.

D. UTILITY METRIC

Different users need different utilities of location data. For example, users wish to leak their privacy as little as possible but obtain a higher quality of services, service providers want to release more locations with the lowest load, and researchers hope to acquire the most accurate location data for data analysis. We measure the utility of our approach in terms of the privacy level (Section VII-C), load, and data availability.

For the service provider, the more trajectories the k-anonymous group has, the greater the overhead (e.g., storage, computation) is, and the lower the utility of the service provider is. Fig. 10 shows that different methods need to release different numbers of trajectories to achieve the same privacy level for the base trajectory. Thus, we measure the utility of the service provider using the minimum number of trajectories that need to be released to achieve a certain level of privacy.

The utility of researchers refers to data availability in terms of available data and data accuracy. We use the available *data ratio* to measure the available data and the variance of the difference before and after the change in access probability (called *change variance*) to measure the data accuracy. For a k-anonymous group, the more accessible locations there are, the greater the available data ratio is. For data accuracy, change of locations probability can affect data accuracy. The smaller the access probability of locations changes, the lower the data accuracy is. The smaller the variance is, the smaller the *change variance* is, and the higher the data accuracy is.

E. RESULTS

1) PRIVACY LEVEL OF THE BASE TRAJECTORY *vs*. *k*

We evaluate the relationship between the privacy level of the base trajectory and anonymity level k. The privacy level is measured by the average *identifiable probability* of 10 base trajectories. Fig. 10 shows the privacy level in terms of the average *identifiable probability* of different methods. Generally, the average *identifiable probability* decreases with k. This occurs because more trajectories in a k-anonymous group mean more trajectories are similar to the basic trajectory. Among these schemes, *Random* has the highest average *identifiable probability* since fake trajectories contain some hard-to-reach locations. It also ignores the probabilistic and geographical features of these trajectories. As a result, it has the most fake trajectories filtered out by the adversary and has the easiest identification of the base trajectory. The privacy level of *Grid* and *Sample* are higher than that of *Random*, since both *Grid* and *Sample* select historical locations to synthesize fake trajectories instead of hard-to-reach locations. The privacy level of *Grid* and *Sample* are also similar. The reason is that the $k-1$ fake locations chosen by *Sample* and their corresponding base location have the same geographic location, and the locations of the *Grid* scheme have the same access probability. Compared with the three schemes, we can see that the *LRM* can achieve a much greater privacy level. This occurs because fake locations in our scheme are not hard-to-reach locations, and fake trajectories are similar to the basic trajectory in terms of the probabilistic and geographic features.

2) PRIVACY LEVEL OF SAMPLING TRAJECTORIES *vs*. *k*

We evaluate the relationship between the privacy level of sampling trajectories and anonymity level k. Their privacy level is measured by the average *unidentifiable ratio* of sampling trajectories corresponding to 10 base trajectories. Fig. 11 shows the privacy level in terms of the average *unidentifiable ratio* of two schemes. Obviously, the *LRM* has the higher average *unidentifiable ratio* than the *Sample* scheme. In the *LRM*, locations in a fake trajectory are sampled from different sampling trajectories, which ensures that each sampling trajectory cannot be identified due to the lower similarity to this fake trajectory. This is why all average *unidentifiable ratio* in the *LRM* are 1. For the *Sample* scheme, each fake trajectory is also a sampling trajectory. This means that some sampling trajectories cannot satisfy the privacy protection requirement of the sampling trajectory. However, to ensure the similarity between the base trajectory and sampling trajectories, only locations with the same geographic location as the base location are released. This is why no average *unidentifiable ratio* in the *Sample* are zero in Fig. 11.

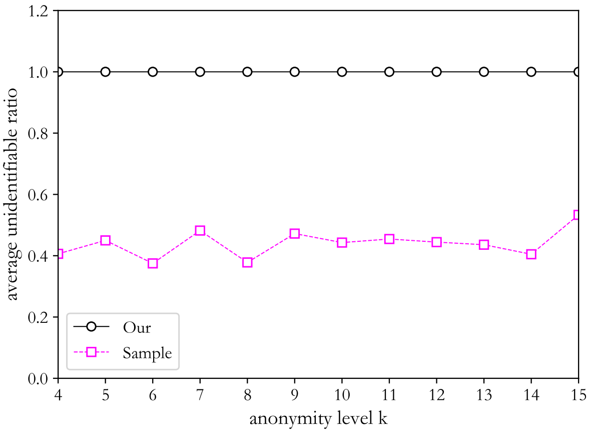


FIGURE 11. The privacy level of sampling trajectories *vs*. *k*.

3) UTILITY OF THE SERVICE PROVIDER *vs*. PRIVACY LEVEL OF THE BASE TRAJECTORY

Fig. 12 shows the relationship between the minimum number of trajectories and *identifiable probability* for different methods. We evaluate the utility of the service provider at different *identifiable probability*: 0.2, 0.4, 0.6 and 0.8. Generally, the utility decreases with the *identifiable probability* since reducing the privacy level results in the decrease of fake trajectories needed to be released. Among these methods, the *LRM* is optimal because it is more effective against the inference attacks on the base trajectory, which causes it to require fewer fake trajectories to achieve the same privacy level as other methods. It also means that the *LRM* requires releasing the least locations to achieve the same privacy level. The reason is that the *LRM* selects locations similar to the base location from the smaller areas to ensure the similarity to the base trajectory in the mobility pattern. As a result, only fewer locations can be selected, which requires the *LRM* to repeatedly use some location pairs to

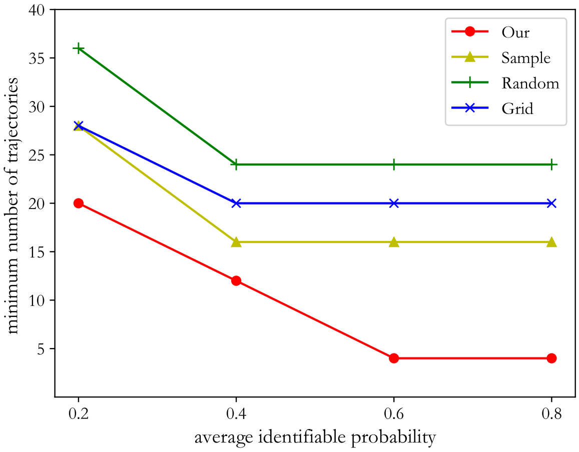
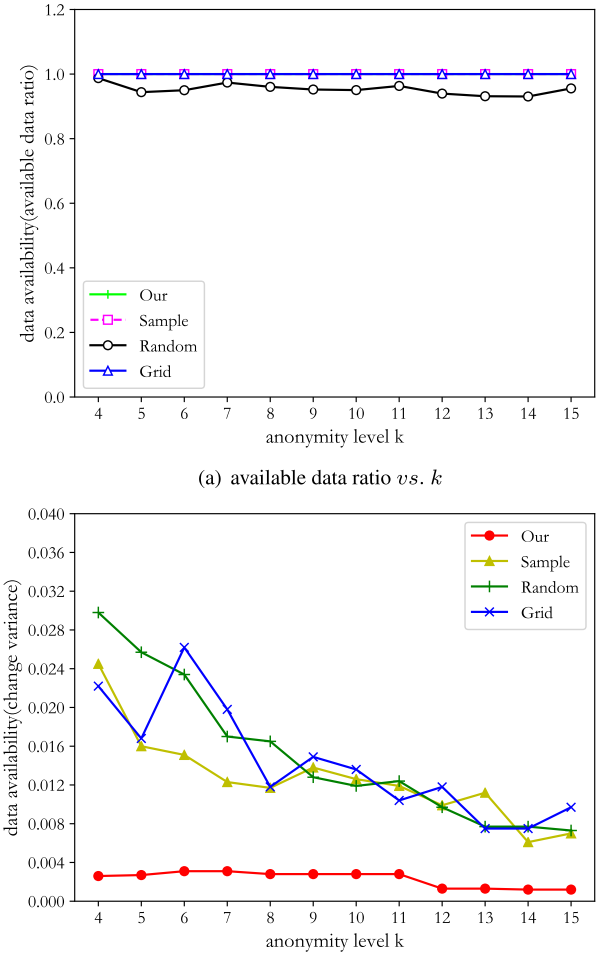


FIGURE 12. **The utility of the service provider *vs*. the privacy level of the base trajectory.**

meet the privacy protection requirement of the base trajectory. The *Grid* and the *Sample* are slightly worse than the *LRM*. The reason is that both the *Grid* and the *Sample* methods release more trajectories than the *LRM* at the same privacy level. *Random* is the worst since it contains some hard-to-reach locations and ignores the probabilistic and geographical features. As a result, it requires the most fake trajectories to achieve the same privacy level as other methods.

4) THE DATA AVAILABILITY *vs*. k

Data availability and anonymity level k. Fig. 13 shows the relationship between data availability and k for different methods. In Fig. 13(a), we evaluate the *available data ratio* of different schemes. Among these schemes, *Random* has the lowest *available data ratio* since it contains some hard-to-reach locations that cannot be used for data analysis. Compared with *Random*, the others have a much higher *available data ratio* because they select historical locations to synthesize fake trajectories. All these historical locations can be used for data analysis. It is also the reason that all of the available data ratios of the *LRM*, the *Grid* scheme and the *Sample* scheme are 1.0. In Fig. 13(b), we evaluate the minimum *change variance* of different schemes. Among all the schemes, the access probabilities in the *LRM* are the most stable. Given that the access probability is a statistic based on how times many people visit a location over a long period of time, it has stability. However, other methods do not consider the stability, resulting in a more remarkable change of access probabilities for some or fewer released locations than that in the *LRM*. *Grid* and *Random* randomly select locations to synthesize fake trajectories, especially *Random* selects some hard-to-reach locations, resulting in greater randomness and an increase in the instability of the access probability of locations. *Sample* needs to select $k - 1$ historical locations that are closer to a base location for each base location. Thus, it has less randomness than *Grid* and *Random*. This is why *Sample* is superior to *Grid* and *Random*, and *Random* has the worst data accuracy.



(a) available data ratio $vs.$ k

(b) minimum change variance $vs.$ k

FIGURE 13. **Data availability *vs*. k.**

VIII. CONCLUSION

This paper proposes a location recombination mechanism based on trajectory k-anonymity to protect the trajectory privacy in the scenario of the data release. We introduce probabilistic similarity and geographic similarity to synthesize fake trajectories that satisfy the privacy protection requirements for both the base trajectory and the sampling trajectory. The verification results on real-world data show that our scheme is more effective than other trajectory k-anonymity schemes against the inference attacks on both the base trajectory and the sampling trajectory. Our method also has a better effect on the privacy protection trajectory. Compared with other approaches, our method achieves better utility for service providers and researchers.

REFERENCES

[1] A. Altomare, E. Cesario, C. Comito, F. Marozzo, and D. Talia, "Trajectory pattern mining for urban computing in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 2, pp. 586–599, Feb. 2017.

[2] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *Proc. ICDM*, Brussels, Belgium, Dec. 2012, pp. 1038–1043.

[3] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data," in *Proc. WWW.*, Perth, WA, Australia, 2017, pp. 1241–1250.

[4] K. Ammar, A. Elsayed, M. M. Sabri, and M. Terry, "BusMate: Understanding mobility behavior for trajectory-based advertising," in *Proc. MDM*, Pittsburgh, PA, USA, vol. 2, Jun. 2015, pp. 74–79.

[5] S. B. Zhang, G. J. Wang, Q. Liu, and J. H. Abawajy, "A trajectory privacy-preserving scheme based on query exchange in mobile social networks," *Soft Comput.*, vol. 22, no. 18, pp. 6121–6133, 2018.

[6] Y. Xin, Z.-Q. Xie, and J. Yang, "The privacy preserving method for dynamic trajectory releasing based on adaptive clustering," *Inf. Sci.*, vol. 378, pp. 131–143, Feb. 2017.

[7] F. Amiri, N. Yazdani, A. Shakery, and A. H. Chinaei, "Hierarchical anonymization algorithms against background knowledge attack in data releasing," *Knowl.-Based Syst.*, vol. 101, pp. 71–89, Jun. 2016.

[8] B. Yao, F. Li, and X. Xiao, "Secure nearest neighbor revisited," in *Proc. ICDE*, Brisbane, QLD, Australia, Apr. 2013, pp. 733–744.

[9] D. Chen, P. Zhang, C. C. Hu, H. Z. Wang, S. Wu, and N. Z. Xing, "Private and precise range search for location based services," in *Proc. ICC*, London, U.K., Jun. 2015, pp. 7347–7352.

[10] Z. Huo, X. Meng, H. Hu, and Y. Huang, "You can walk alone: Trajectory privacy-preserving through significant stays protection," in *Proc. DASFAA*, Busan, South Korea, 2012, pp. 351–366.

[11] R. Chen, B. C. M. Fung, N. Mohammed, B. C. Desai, and K. Wang, "Privacy-preserving trajectory data publishing by local suppression," *Inf. Sci.*, vol. 231, pp. 83–97, May 2013.

[12] R. Kato, M. Iwata, T. Hara, A. Suzuki, X. Xie, Y. Arase, and S. Nishio, "A dummy-based anonymization method based on user trajectory with pauses," in *Proc. GIS*, Redondo Beach, CA, USA, 2012, pp. 249–258.

[13] X. Wu and G. Sun, "A novel dummy-based mechanism to protect privacy on trajectories," in *Proc. ICDMW*, Shenzhen, China, Dec. 2014, pp. 1120–1125.

[14] M. Gramaglia, M. Fiore, A. Tarable, and A. Banchs, "Preserving mobile subscriber privacy in open datasets of spatiotemporal trajectories," in *Proc. INFOCOM*, Atlanta, GA, USA, May 2017, pp. 1–9.

[15] T. Peng, Q. Liu, D. Meng, and G. Wang, "Collaborative trajectory privacy preserving scheme in location-based services," *Inf. Sci.*, vol. 387, pp. 165–179, May 2017.

[16] S. Gao, J. Ma, C. Sun, and X. Li, "Balancing trajectory privacy and data utility using a personalized anonymization model," *J. Netw. Comput. Appl.*, vol. 38, pp. 125–134, Feb. 2014.

[17] A. Y. Ye, Y. Li, L. Xu, Q. Li, and H. Lin, "A trajectory privacy-preserving algorithm based on road networks in continuous location-based services," in *Proc. ICESS*, Sydney, NSW, Australia, vol. 1, Aug. 2017, pp. 510–516.

[18] S. Gao, J. Ma, W. Shi, G. Zhan, and C. Sun, "TrPF: A trajectory privacy-preserving framework for participatory sensing," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 6, pp. 874–887, Jun. 2013.

[19] P.-R. Lei, W.-C. Peng, I.-J. Su, and C.-P. Chang, "Dummy-based schemes for protecting movement trajectories," *J. Inf. Sci. Eng.*, vol. 28, no. 2, pp. 335–350, 2012.

[20] Y. M. Sun, M. Chen, L. Hu, Y. F. Qian, and M. M. Hassan, "ASA: Against statistical attacks for privacy-aware users in location based service," *Future Gener. Comput. Syst.*, vol. 70, no. 2017, pp. 48–58, May 2017.

[21] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proc. INFOCOM*, Toronto, ON, Canada, Apr./May 2014, pp. 754–762.

[22] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," 2012, *arXiv:1212.1984*. [Online]. Available: https://arxiv.org/abs/1212.1984

[23] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Proc. SP*, San Jose, CA, USA, May 2016, pp. 546–563.

[24] W. Qiong, H. Liu, C. Zhang, Q. Fan, Z. Li, and K. Wang, "Trajectory protection schemes based on a gravity mobility model in IoT," *Electronics*, vol. 8, no. 2, p. 148, 2019.

[25] H. Liu, X. Li, H. Li, J. Ma, and X. Ma, "Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services," in *Proc. INFOCOM*, Atlanta, GA, USA, May 2017, pp. 1–9.

[26] L. Farhan, A. E. Alissa, S. T. Shukur, M. Hammoudeh, and R. Kharel, "An energy efficient long hop (lh) first scheduling algorithm for scalable Internet of things (iot) networks," in *Proc. ICST*, Sydney, NSW, Australia, Dec. 2017, pp. 1–6.

[27] D. Liao, G. Sun, H. Li, H. Yu, and V. Chang, "The framework and algorithm for preserving user trajectory while using location-based services in IoT-cloud systems," *Cluster Comput.*, vol. 20, no. 3, pp. 2283–2297, 2017.

[28] Y. Feng, P. Liu, and J. Zhang, "A mobile terminal based trajectory preserving strategy for continuous querying LBS users," in *Proc. DCOSS*, Hangzhou, China, May 2012, pp. 92–98.

[29] A. Pingley, N. Zhang, X. Fu, H.-A. Choi, S. Subramaniam, and W. Zhao, "Protection of query privacy for continuous location based services," in *Proc. INFOCOM*, Shanghai, China, Apr. 2011, pp. 1710–1718.

[30] R. Lu, X. Lin, Z. Shi, and J. Shao, "Plam: A privacy-preserving framework for local-area mobile social networks," in *Proc. INFOCOM*, Toronto, ON, Canada, Apr./May 2014, pp. 763–771.

[31] M. Ghasemzadeh, B. C. Fung, R. Chen, and A. Awasthi, "Anonymizing trajectory data for passenger flow analysis," *Transp. Res. C, Emerg. Technol.*, vol. 39, pp. 63–79, 2014.

[32] B. Niu, X. Y. Zhu, Q. H. Li, J. Chen, and H. Li, "A novel attack to spatial cloaking schemes in location-based services," *Future Gener. Comput. Syst.*, vol. 49, no. 2015, pp. 125–132, Aug. 2015.

[33] I. Palomares, L. Martínez, and F. Herrera, "MENTOR: A graphical monitoring tool of preferences evolution in large-scale group decision making," *Knowl.-Based Syst.*, vol. 58, pp. 66–74, Mar. 2014.

[34] Á. Labella, Y. Liu, R. M. Rodríguez, and L. Martínez, "Analyzing the performance of classical consensus models in large scale group decision making: A comparative study," *Appl. Soft Comput.*, vol. 67, pp. 677–690, Jun. 2018.

[35] X. Yi, R. Paulet, E. Bertino, and V. Varadharajan, "Practical k nearest neighbor queries with location privacy," in *Proc. DE*, Chicago, IL, USA, Mar./Apr. 2014, pp. 640–651.

[36] F. Olumofin and I. Goldberg, "Revisiting the computational practicality of private information retrieval," in *Proc. FCDS*, Berlin, Germany: Springer, 2012, pp. 158–172.

[37] Y.-X. Liu, C.-N. Yang, C.-M. Wu, Q.-D. Sun, and W. Bi, "Threshold changeable secret image sharing scheme based on interpolation polynomial," *Multimedia Tools Appl.*, vol. 78, no. 13, pp. 18653–18667, 2019.

[38] Y. Liu and C. Yang, "Scalable secret image sharing scheme with essential shadows," *Signal Process., Image Commun.*, vol. 58, pp. 49–55, Oct. 2017.

[39] Y. X. Liu, C. N. Yang, Q. D. Sun, S. Y. Wu, S. S. Lin, and Y. S. Chou, "Enhanced embedding capacity for the SMSD-based data-hiding method," *Signal Process. Image Commun.*, vol. 78, pp. 216–222, Oct. 2019.

[40] Y. Zheng, H. Fu, and X. Xie. (Jul. 2012). *Geolife GPS Trajectory Dataset-User Guide Microsoft Research*. [Online]. Available: https://www.microsoft. com/enus/research/publication/geolife-gps-trajectory-dataset-user-guide

[41] T. Le and I. Echizen, "Lightweight collaborative semantic scheme for generating an obfuscated region to ensure location privacy," in *Proc. IEEE SMC*, Miyazaki, Japan, Oct. 2018, pp. 2844–2849.

[42] V. Kulkarni, A. Mahalunkar, B. Garbinato, and J. D. Kelleher, "Examining the limits of predictability of human mobility," *Entropy*, vol. 21, no. 4, p. 432, 2019.

[43] G. Qiu, K. Cheng, L. Liu, and S. Zeng, "Tmarkov: Lbs trajectory prediction for crowdsourcing recommendation," in *Proc. NaNA*, Xi'an, China, Oct. 2018, pp. 153–158.

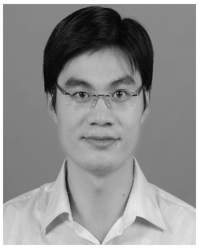
[44] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *Proc. INFOCOM*, Hong Kong, Apr./May 2015, pp. 1017–1025.

YUNFENG WANG was born in 1987. He received the B.Sc. degree in public service management and the M.Sc. degree in management science and engineering from Henan Polytechnic University, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include network and information security and privacy protection.

MINGZHEN LI was born in 1986. She received the B.Sc. degree in information and computing science from Luoyang Normal University, in 2009, and the M.Sc. degree in computer application technology from the Guilin University of Electronic Technology, in 2012. She is currently pursuing the Ph.D. degree with the School of Cyberspace Security, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. She is also a Lecturer with the School of Computer and Information Engineering, Hechi University. Her research interests include network and information security and privacy protection.

SHOUSHAN LUO received the B.Sc. degree in mathematics from Beijing Normal University, in 1985, and the M.Sc. degree in applied mathematics and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications, in 1994 and 2001, respectively. He is currently a Professor with the School of Cyberspace Security, BUPT, Beijing, China. His research interests include cryptography and information security.

YANG XIN was born in 1977. He received the B.Sc. degree in signal and information system and the M.Sc. degree in circuits and systems from Shandong University, in 1999 and 2002, respectively, and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications, in 2005. He is currently a Professor with the School of Cyberspace Security, BUPT, Beijing, China. His research interests include big data security, cloud computing security, and network security.



HONGLIANG ZHU was born in 1982. He received the Ph.D. degree in information security from the Beijing University of Posts and Telecommunications (BUPT), in 2010. He is a Master Supervisor with BUPT, meanwhile, serves as the Vice Director of the Beijing Engineering Lab for CloudSecurity and Information Security Center of BUPT. His current research interests include network security and big data security.

YULING CHEN was born in 1983. She received the B.Sc. degree in applied mathematics from Taishan University, in 2006, and the M.Sc. degree in computer application technology from Guizhou University, in 2009. She is currently an Associate Professor with the Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University. Her research interests include cryptography and information safety.

GUANGCAN YANG was born in 1986. He received the B.Sc. degree in network engineering and the M.Sc. degree in computer application technology from Henan Polytechnic University, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include network and information security and privacy protection.

YIXIAN YANG was born in 1961. He received the B.Sc. degree in applied mathematics from the University of Electronic Science and Technology, in 1983, and the M.Sc. degree in applied mathematics and the Ph.D. degree in electronic and communication system from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1986 and 1988, respectively. He is currently a Professor with the School of Cyberspace Security, BUPT. His research interests include modern cryptography, security of network and information, information hiding and digital water-marking, big data, and cloud computing security.