Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8635275, 16 pages
https://doi.org/10.1155/2022/8635275

Research Article

A Trajectory Privacy Protect Method Based on Location Pair Reorganization

Wanqing Wu,[1,2] Wenlong Shang ,[1,2] Ruohe Lei,[1,2] and Xin Yang[1,2]

[1]*School of Cyber Security and Computer, Hebei University, Baoding, China*
[2]*Hebei Provincial Key Laboratory of High Credibility Information System, Baoding, China*

Correspondence should be addressed to Wenlong Shang; swl199698@163.com

Received 26 April 2022; Accepted 22 June 2022; Published 5 July 2022

Academic Editor: Dharmendra Singh Rajput

With the rapid development of mobile Internet and communication technology, location-based services (LBS) are widely used in our daily life. The server stores a large amount of user location data, and these location data constitute user trajectories. If trajectory information on the server is leaked, it will seriously endanger users' privacy. Trajectory k-anonymity technology is one of the most important methods to protect the privacy of user trajectory. However, current trajectory k-anonymity methods have less discussion on the semantic of stop point when selecting dummy trajectory, which leads to the fact that attacker can still exclude the dummy trajectory from the k-anonymity set and infer the real trajectory by combining background knowledge with the semantic information of stop points. To address this problem, this paper decomposes the real trajectory into location pairs set; the set consists of start-end points and stop points. According to the similarity of location pairs, the similar location pairs in history trajectory set are used to generate dummy trajectory: firstly, extracting the start-end points and stop points from real trajectory and assigning semantic to them. Then, based on the semantic, temporal, and geographical attributes, eligible location pairs are selected from history trajectory set to construct equivalence class. Finally, according to the location pairs in equivalence class, $k-1$ dummy trajectories are generated to form a k-anonymity set. We evaluate our method thoroughly with real dataset. The results show that our method achieve an effective data availability and higher privacy protection than other methods.

1. Introduction

In recent years, with the rapid development of 5G technology and the Internet of Things (IoT), smart city is gradually becoming a reality. As an important cornerstone of smart city, location-based services (LBS) are used in more and more areas, such as check-in, road conditions, and social networking [1, 2]. When a user requests an LBS service, he submits his current location to the service provider. The service provider stores user's location sequence on the server as a trajectory. Some service providers regularly release trajectory data to governments and research institutions for analysis and mining [3–5]. For example, the U.S. government updates and optimizes transportation facilities based on users' GPS trajectory data [6]; the Chinese government constructs epidemic prevention maps based on COVID-19 patients' trajectories, etc.

However, the servers that store trajectory data are not absolutely secure. If the service provider is attacked by attacker, the trajectory data may be leaked without any protection. By analysing the spatial-temporal information in user's trajectory, attacker combined with background knowledge can deduce user's hobbies, mobility patterns, health status, work and home address, and other personal information, which can cause economic losses and even threaten user's personal safety. Accordingly, scholars at home and abroad pay a large amount of attention to this thing: how to better protect the trajectory privacy.

The existing trajectory privacy protection methods can be divided into three categories: trajectory k-anonymity method [7–14], suppression [15, 16], and differential privacy [17–21]. The suppression method assumes that attacker has some specific background knowledge, protecting trajectory privacy by suppressing sensitive information in the trajectory. However, suppression method requires predetermined sensitive information; if sensitive information is not set properly, it can seriously damage data availability. Differential privacy ensures unconditional privacy, i.e., individual information cannot be obtained by analysing specified statistical data. However, differential privacy can only protect a limited amount of information. Compared to the first two methods, trajectory k-anonymity method transforms the $1:1$ relationship between user and trajectory into $k:1$ relationship between multiple trajectories and user by generating $k-1$ dummy trajectories. It is the main method to achieve trajectory privacy protection because of its simple implementation and flexible application scenarios.

How to choose the dummy trajectory is the key issue in trajectory k-anonymity. For availability purposes, dummy trajectory in k-anonymity set should contain the most possible available information. For privacy purposes, dummy trajectory and real trajectory must be indistinguishable. Based on these two demands, many trajectory k-anonymity methods are proposed [12, 14, 22, 23]. In the similarity calculation process, these methods calculate the similarity based on all location points in the trajectory, which leads to huge computational effort and the low data availability after privacy processing. In fact, not every location point in trajectory is necessary for privacy protection [24]. It is the stop points in the user's trajectory that really reveal the user's privacy. According to this idea, many methods calculate the similarity between trajectories based on the stop points in the trajectory, which can reduce the calculation volume while maintaining a level of privacy [13]. However, the above methods do not consider the semantic impact in trajectory privacy protection. In a trajectory, stop points combined with semantic attributes reveal the user's identity and action patterns. By analysing the semantic information of stop points, attacker can easily identify certain dummy trajectories from k-anonymity set or even obtain the user's real trajectories directly.

As shown in Figure 1, suppose Tom, an employee of a company, leaves the company at 18:00 on Wednesday to watch a movie at the cinema and then returns to his home. Tom's trajectory is represented by Tra_2. According to trajectory k-anonymity method, now, we generate the dummy trajectories Tra_1 and Tra_3 to protect Tom's trajectory. The three trajectories have greater similarity in trajectory shape, geographic location, and overall direction. However, by analysing trajectory's stop points, attacker can still find the difference between three trajectories. By extracting stop points from three trajectories, attacker can get the semantic trajectory of three trajectories as: $A_1(\text{Planetarium}) \longrightarrow C_1(\text{KFC}) \longrightarrow E_1(\text{Hostel})$; $A_2(\text{Company}) \longrightarrow C_2(\text{Cinema}) \longrightarrow E_2($ Community); $E_3(\text{Hospital})$. Compared with Tra_1 and Tra_2, Tra_3 has only one stop point, and hospital's working hours is from 8:00 to 18:00, so attacker speculates that Tra_3 is likely to be a dummy trajectory. In addition, although both Tra_1 and Tra_2 have three stop points, attacker knows that Tom is an employee of a company according to the background knowledge. So, attacker can infer that Tra_2 is more consistent with Tom's action pattern and then determine that Tra_2 is the real trajectory.

To solve the problem that low trajectory utilization and stop point's semantic lead to trajectory k-anonymity failed, this paper proposes a trajectory privacy protect method based on location pair reorganization (DSTPP). Specifically, we decompose the real trajectory into a set of location pairs consisting of start-end point and stop points. For each location pair, we select eligible location pairs from the history trajectory set for constructing candidate location set. Finally, we use the location pairs in candidate location set to generate dummy trajectories that satisfy the similarity measure; $k-1$ dummy trajectories are generated to form a k-anonymity set.

The main contributions of this paper are as follows:

(i) We design a candidate location set generation method. For each location pair in the real trajectory, according to the defined location pair similarity, the eligible location pairs are selected from the history trajectory dataset and added to the candidate location. Location pairs in the candidate location have high spatial-temporal and semantic similarity and can be used to generate dummy trajectory that match user action patterns

(ii) We design a dummy trajectory generation method that conforms to user action patterns. According to the similarity measures, dummy trajectory is similar to real trajectory in terms of geography, semantics, and direction

(iii) We evaluated the privacy and availability of DSTPP with two similar methods [13, 25] on real dataset [26–28]. The experimental results show that the trajectory k-anonymity set constructed by DSTPP meets the privacy protection requirements and has high data availability

The rest of this paper is structured as follows: relevant work is reviewed in Section 2. Section 3 provides a description of relevant concepts and measure standards. Section 4 elaborates the trajectory privacy protect method based on location pair reorganization. In Section 5, we compare with existing solutions in terms of availability and privacy, and this paper is concluded in Section 6.

2. Related Work

Typical privacy-preserving methods include suppression, generalization, and perturbation. Among them, k-anonymity technique based on generalization are widely used in trajectory privacy protection. The k-anonymity model was proposed by Sweeney [29] in 2002, which is the first complete model of privacy protection. This model prevents attacker from uniquely identifying a specific user in the dataset, making it impossible to obtain further accurate information about that user. Gruteser and Grunwald [30] first applied k-anonymity techniques to LBS services. For the purpose of protecting user privacy, they replace the user's exact location points with a location region that contains k location points, so the probability of users being identified is reduced. However, this method cannot resist the privacy leakage problem caused by trajectory

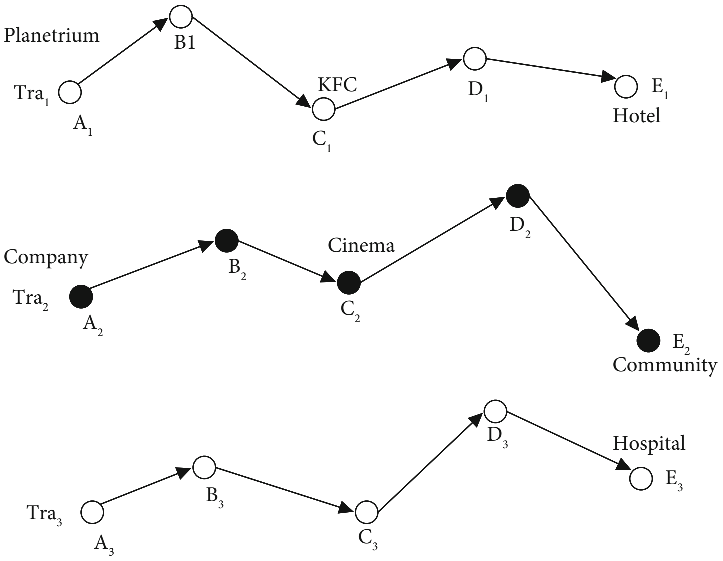


Figure 1: Example of identifying dummy trajectory.

data leakage, nor can it resist the background knowledge attack. For this reason, trajectory k-anonymity method was developed. This type of method constitutes a k-anonymity set that includes $k-1$ dummy trajectories and real trajectory. k trajectories in the anonymous set have indistinguishability, which reduces the identification probability of real trajectories to $1/k$. According to the way of generating dummy trajectory, the existing trajectories k-anonymity can be divided into two categories: local method and integral method.

For a trajectory, user really cares about certain specific geographic location, not all locations. Based on this idea, the local method is proposed. The local method means only protecting trajectory's sensitive locations by k-anonymity, not the whole trajectory. Pan et al. [31] considered the user's movement direction and velocity when generating the generalized region, ensuring the user trajectory's privacy while improving the service quality. Zhang et al. [9] proposed a double-k mechanism to protect user's sensitive locations. They send the user's sensitive location and $k-1$ fake locations to k anonymizers. Then, each anonymizer is k-anonymized for a location. This method has a higher degree of privacy. However, this method has a high computational volume, and the trajectory availability is lower. To address this problem of high computational volume, Zhou and Wang [11] combined fog computing and k-anonymity to reduce the computational consumption of generating k -anonymity sets. Zhao et al. [10] consider the start-end points of user's trajectory are sensitive locations. Based on user's behaviour, they generate secure start-end candidate point set for constructing dummy trajectory. Ye et al. [32] propose protect location points within sensitive areas; they build the cloaking region for sensitive areas, which contains another k similar POIs to the sensitive place, and randomly select one to replace the sensitive place. However, the local method requires presetting sensitive locations. If the sensitive locations are set improperly, it can seriously affect data availability. In addition, user privacy can also be leaked based on location points in nonsensitive areas.

In order to solve the defects caused by the local method, the integral method is proposed. The integral method is to select $k-1$ dummy trajectories that are similar to real trajectory to form a k-anonymity set. Xu et al. [12] evaluated the trajectory similarity based on four features: angle, velocity, time, and space. Then, they selected $k-1$ historical trajectories that were similar to the real trajectories to form a k -anonymity set. Wang et al. [22] propose constituting k -anonymous set by exchanging the locations of neighbour nodes, protecting user privacy by interchanging the location between neighbour nodes on the K-core subnet of the relationship network. But this method ignores the privacy needs of different location points, which leads to insufficient availability. According to the spatial and temporal characteristics of trajectory data, Li et al. [23] propose a data partitioning method to store and calculate trajectory, which reduces computational volume. Liu et al. [25] generated $k-1$ dummy locations for each location point in the real trajectory and randomly generated dummy trajectory based on these dummy locations. The dummy trajectory generated by this method have some unreachable positions, so the data availability and privacy are insufficient. Dai et al. [33] proposed simplifying real trajectory into a trajectory that is only consisting of stop points. Then, each stop points constructs a k-anonymity set. The k-anonymity set contains $k-1$ semantically similar location points. According to these k-anonymity sets, generate dummy trajectory randomly. The dummy trajectory generated by this method only considers stop points, which may pass through unreachable locations. Meanwhile, it does not consider whether the direction of dummy trajectory is similar to real trajectory. Xu et al. [13] evaluated the trajectory similarity based on the number of stop points and average velocity. Then, select $k-1$ historical trajectories that are similar to the real trajectories to form a k -anonymity set. The integral method selects history trajectory or fake trajectory as dummy trajectory. However, although using fake trajectory can ensure the similarity requirement, the fake trajectory may pass through or be at

Table 1: Summary of notation.

Symbol	Notion
D Tra_i	Historical trajectory dataset/ith trajectory in historical trajectory dataset
$T = \{t_i \mid i = 1, 2, \cdots, n\}$	Real trajectory/ith location point
$\text{SEM} = \{sl_i \mid i = 1, 2, \cdots, n\}$	Semantic trajectory/ith semantic location point
$\text{TC} = \{tc_i \mid i = 1, 2, \cdots, n\}$	Trajectory equivalence class/ith trajectory in the class
$\text{STC} = \{stc_i \mid i = 1, 2, \cdots, n\}$	Semantic trajectory equivalence class/ith semantic trajectory
$\text{Can} = \{C_i \mid i = 1, 2, \cdots, n\}$	Set of candidate location/ith candidate location
$\text{FS} = \{fs_i \mid i = 1, 2, \cdots, n\}$	Fake semantic trajectory
FT	The set of $k - 1$ dummy trajectories forming the k-anonymity set
$F = \{f_i \mid i = 1, 2, \cdots, n\}$	Fake trajectory in FT
$\text{sim}_G/\text{sim}_S$ sim_A	Geographical similarity/semantic similarity/directional similarity

unreachable locations. Therefore, the availability and privacy are insufficient. If using history trajectory, the problem that the trajectory number is insufficient may be faced; meanwhile, although the whole trajectory does not satisfy the similarity requirement, some trajectory segments in these trajectories can still be utilized. But the integral method cannot utilize these trajectory segments.

In conclusion, existing dummy trajectory selection methods have the defects in low trajectory utilization and pay less attention to stop point's semantic, which leads to the fact that attacker combining with background knowledge can still exclude the dummy trajectory from the k-anonymity set. In order to solve above problem, this paper constructs dummy trajectory by the location pairs. Location pairs consist of stop points in history trajectory set. The history trajectory is taken from real user's trajectory, so dummy trajectory does not pass through or reach unreachable locations. At the same time, we defined semantic, geographical, and directional similarities, which ensure the similarity between dummy trajectory and real trajectory. This method can effectively resist background knowledge attack and improve data availability.

3. Preliminaries

To facilitate reader's understanding of the various system parameters used in this paper, Table 1 provides a description of the various system parameters used in this paper.

3.1. Related Concepts

Definition 1. GPS trajectory: the GPS trajectory can be represented as a polyline in three-dimensional space (two-dimensional coordinates and time dimension), denoted as $T = \{t_1, t_2, ,t_3, \cdots, t_{e-1}, t_e\}$, where $t_k = (x_k, y_k, \text{time}_k)$ indicates that the position (x_k, y_k) of the trajectory T at time time_k, $t_1 < t_2 < \cdots < t_e$, and e is the number of sampling points of real trajectory T.

The location points in real trajectory can be divided into two categories: moving point and stop point. Stop point is a location where the user stays (speed is 0 and lasts for a period of time) or visits (speed is not 0, but repeatedly wanders around a location). Moving point is a location that the user simply passes through. The specific description of stop point is given by Definition 2.

Definition 2. Stop point: a stop point is a geographic area where user stays within a distance threshold θ_d longer than a time threshold θ_t. In the real trajectory T, the subtrajectory is denoted as $\text{Sub}T = <t_i, t_{i+1}, \cdots, t_j>$, where, $\forall k \in (i, j)$, $\text{Dis}_{\text{geo}}(t_i, t_k) \le \theta_d$, and $\text{Dis}_{\text{time}}(t_i, t_j) \ge \theta_t$. Then, $< t_i, t_{i+1}, \cdots, t_j >$ can be combined into a stop point s. Denote as: $s{=}(x, y, \text{time}_a, \text{time}_l, \text{time}_d)$, where:

$$x = \frac{\sum_{k=i}^{j} t_k.x}{|\text{Sub}T|}, \tag{1}$$

$$y = \frac{\sum_{k=i}^{j} t_k.y}{|\text{Sub}T|}, \tag{2}$$

represent the latitude and longitude of stop point s, respectively. $t_k.x$ and $t_k.y$ denote the longitude and latitude of location point t_k.

time_a denotes the time of entering the stop point s, time_l denotes the time to leave the stop point s, and time_d denotes the stay time at stop point s. The values are calculated using the first position point t_i, the last position point t_j, and the time difference between t_j and t_i in the Sub$T$, respectively. According to Definition 1 and Definition 2, trajectory can be composed of stop points and moving points between top points.

Definition 3. Semantic category: the semantic category of location point can be represented by points of interest

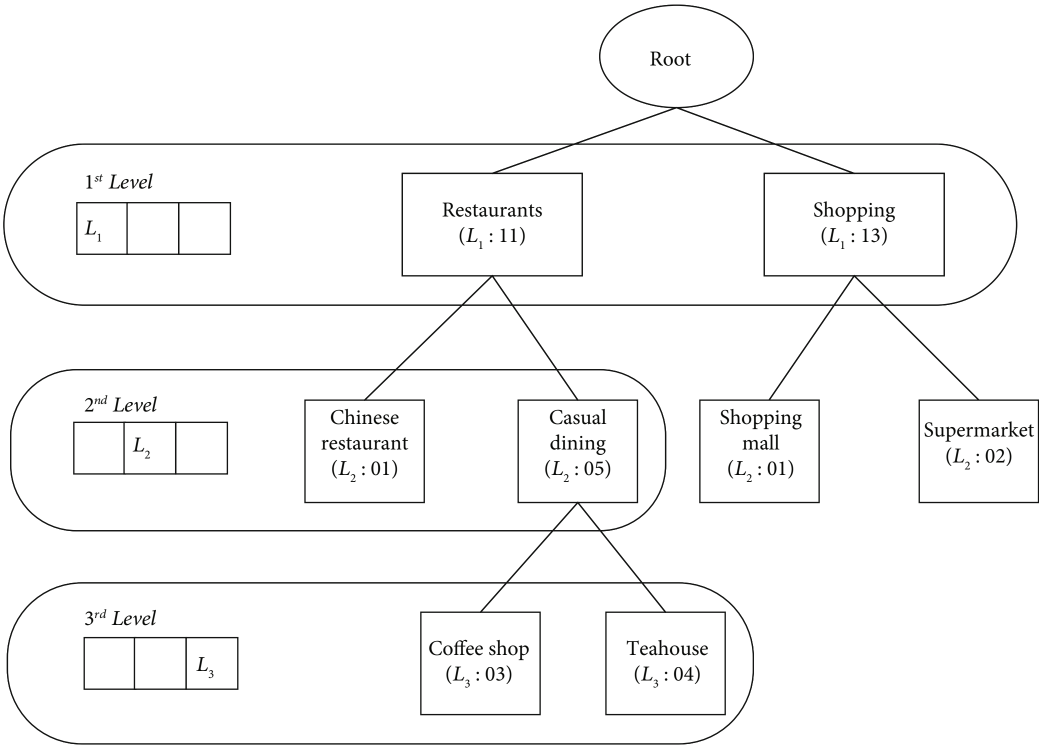


Figure 2: Example of the semantic information levels and the semantic codes.

(POI). Each location point (x_i, y_i) has an independent POI, and the semantic category can be obtained through Chinese POI standard [34].

According to Chinese POI standard, POI semantic category is divided into three levels: major category, middle category, and minor category. Each level consists of a 2-digit code. The codes L_1 for the major category, L_2 for the middle category, and L_3 for the minor category are arranged in order to form a fixed 6-bit semantic code (L_1, L_2, and L_3), as shown in Figure 2.

Definition 4. Semantic location point: the semantic location point sl can be represented as $<\mathrm{loc}, \mathrm{cat}, \mathrm{time}>$, where loc denotes the center coordinates of semantic location point; cat denotes the semantic category of semantic location point, consisting of the triplet $<L_1, L_2, L_3>$; and time represents the temporal characteristics of semantic location point, which consists of the triplet $<\mathrm{time}_a, \mathrm{time}_l, \mathrm{time}_d>$. The three attributes indicate visit time, leave time, and stay time, respectively.

Definition 5. Semantic trajectory: a semantic trajectory SEM is an ordered list of start-end points and a set of semantic points:

$$\mathrm{SEM} = \{\mathrm{sl}_1, \mathrm{sl}_2, \cdots, \mathrm{sl}_n\}. \tag{3}$$

Definition 6. Semantic distance d_{sem}: d_{sem} refers to the semantic difference between two semantic location points. It is expressed by semantic encoding difference of two semantic location points:

$$\begin{aligned} d_{\mathrm{sem}} &= \left|(l_1, l_2, l_3)_{\mathrm{SEM}_1} - \left(l'_1, l'_2, l'_3\right)_{\mathrm{SEM}_2}\right| \\ &= \left|10000\left(l_1 - l'_1\right) + 100\left(l_2 - l'_2\right) + \left(l_3 - l'_3\right)\right|. \end{aligned} \tag{4}$$

According to the specification:

When the semantic distance $d_{\mathrm{sem}} \in (0,100)$, it means that two semantics differ only in minor category. The two points have a high degree of similarity.

When the semantic distance $d_{\mathrm{sem}} \in [100, 9999]$, it means that two semantics differ only in middle category. The two points have a low degree of similarity.

When the semantic distance $d_{\mathrm{sem}} \in [100, 9999]$, it means that two semantics differ only in major category. No similarity between the two points at all.

Definition 7. Geographical distance d_{geo}: d_{geo} refers to the geographical difference between two semantic location points, which can be calculated by the Euclidean distance:

$$d_{\mathrm{geo}} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \tag{5}$$

Definition 8. Location pair similarity [14]: suppose $<sl_{i-1}, sl_i>$ and $<cl_{i-1}, cl_i>$ are the location pairs in real semantic trajectory SEM and semantic trajectory equivalence class STC, respectively. Two location pairs satisfy the location pair similarity if they satisfy the following conditions:

(1) $d_{\text{sem}}(sl_{i-1}, cl_{i-1}) < \delta_{\text{sem}}, d_{\text{sem}}(sl_i, cl_i) < \delta_{\text{sem}}$

(2) $d_{\text{geo}}(sl_{i-1}, cl_{i-1}) < \delta_{\text{geo}}, d_{\text{sem}}(sl_i, cl_i) < \delta_{\text{geo}}$

(3) $d_{\text{time}}(sl_{i-1}, cl_{i-1}) < \delta_{\text{time}}, d_{\text{time}}(sl_i, cl_i) < \delta_{\text{time}}$

where, δ_{sem}, δ_{geo}, and δ_{time} are semantic distance threshold, geographic distance threshold, and time threshold, respectively. Equations (1) and (2) ensure that location pairs are semantically and geographically similar. Equation (6) ensures that two location pairs are similar in velocity and time period by calculating the location time difference.

3.2. Similarity Measure. In the k-anonymity set, we want the generated dummy trajectory to be semantically and geographically indistinguishable from real trajectories. In this subsection, geographic similarity and semantic similarity are proposed to evaluate the differences between dummy trajectory and real trajectory.

We first give two theorems and utilize them to prove that the difference between two probability distributions is the expectation of the difference between the features of the corresponding positions.

Theorem 9. *Suppose X_0 is a random variable with probability distribution $P(X_0)$. $X = \{X_i | i = 1, 2, \cdots, n\}$ is a set of random variables, the corresponding probability distribution is $P(X_i)$. For each X_i, there is a probability distribution difference with X_0, the probability distribution difference between $P(X_0)$ and $P(X_i)$ is denoted by $d(X_0, X_i)$. The probability of the probability distribution difference between $P(X_0)$ and $P(X_i)$ is $P_{X_i}(X_i)$. Therefore, the probability distribution difference $d(X_0, X_1, \cdots, X_i)$ between $X_0, X_1, \cdots, X_i$ and X_0 is the expectation of each difference, i.e.,*

$$d(X_0, X_1, \cdots, X_i) = \sum_{i=1}^{n} P_{X_i}(X_i) \bullet d(X_0, X_i). \tag{6}$$

Proof. $P_{X_i}(X_i)$ and $d(X_0, X_i)$ are functions of variables X_0 and X_i. When the value of X_0 remains unchanged, the results of two functions depend only on X_i. If X_i is also deterministic, then both $P_{X_i}(X_i)$ and $d(X_0, X_i)$ are constants. That is, there is a one-to-one correspondence between $P_{X_i}(X_i)$ and $d(X_0, X_i)$. Therefore, $P_{X_i}(X_i)$ can be regarded as the probability of $d(X_0, X_i)$ in all probability distributions difference. Accordingly, $d(X_0, X_1, \cdots, X_i)$ is the expectation of probability distribution difference $d = \{d(X_0, X_i) | i = 1, 2, \cdots, n\}$ and its probability $d = \{P_{X_i}(X_i) | i = 1, 2, \cdots, n\}$. □

Theorem 10. *Suppose X' and Y' are two sets of random variables with probability distributions $P(X' = x_i)$ and $P(Y' = y_i)$, where $X' = \{x_i | i = 1, 2, \cdots, m\}$ and $Y' = \{y_i | i = 1, 2, \cdots, m\}$. For $\forall x_i$ and y_i, the probability distributions difference of $P(X' = x_i)$ and $P(Y' = y_i)$ is $d(x_i, y_i)$. The probability of probability distributions difference between $P(X' = x_i)$ and $P(Y' = y_i)$ is $P_{Y'}(Y' = y_i)$. The probability distribution difference between X' and Y' is the expectation $d(X', Y')$ of all $d(x_i, y_i)$, i.e.,*

$$d\left(X', Y'\right) = \sum_{i=1}^{m} P_{Y'}\left(Y' = y_i\right) \bullet d(x_i, y_i). \tag{7}$$

Proof. For $\forall x_{i'} \in X'$, according to Theorem 9, the difference between $x_{i'}$ and Y' is $\sum_{i=1}^{m} P_{Y'}(Y' = y_i, X' = x_{i'}) \bullet d(x_{i'}, y_i)$. Then, $d(X', Y') = \sum_{i=1}^{m} \sum_{i=1}^{m} P_{Y'}(Y' = y_i, X' = x_{i'}) \bullet d(x_{i'}, y_i)$. However, for $d(X', Y')$, if $i \neq i'$, then $P_{Y'}(Y' = y_i, X' = 0)$ and $d(x_{i'}, y_i)$ are 0. That means $d(X', Y') = \sum_{i=1}^{m} P_{Y'}(Y' = y_i) \bullet d(x_i, y_i)$. Therefore, the probability distribution difference of X' and Y' is the expectation $d(X', Y')$ of all $d(x_i, y_i)$. □

For $\forall F \in \text{FT}$ and real trajectory T, semantic distance, geographic distance, and its corresponding probability distribution (the probability of distance difference) depend only on F. Therefore, according to Theorem 10, we define geographical similarity and semantic similarity between F and T as the expectation of distance difference.

Definition 11. Visit probability: $P_v(p)$ represents the probability of a user visiting location P, which is calculated as follows:

$$P_v(p) = \frac{\text{Num}(p)}{N}, \tag{8}$$

where $\text{Num}(p)$ represents the number of people visiting location P and N denotes the total number of people.

The larger the value of $P_v(p)$, the more people visit location P, and the higher the probability that location P will be visited.

Definition 12. Geographical similarity [14]: for $\forall F \in \text{FT}$ and real trajectory T, $D_G(t_i, f_i)$ denotes the geographic distance between t_i and f_i, and the probability distribution of geographic distance is denoted by $P_G(f_i)$. The geographical similarity between T and F can be defined as the expectation of all $D_G(t_i, f_i)$:

$$\sum_{i=1}^{n} P_G(f_i) \bullet D_G(t_i, f_i). \tag{9}$$

Attacker's goal is to infer user's real trajectory from the k-anonymity set. In order to achieve this goal, attacker usually assumes a location (called hypothetical location) to be the real location and evaluate the probability that this location is the true location by background knowledge. In the candidate location C_i, we assume that e_i is the hypothetical location. Without considering background knowledge, the probability that e_i is the true location is denoted by $P(e_i | C_i)$. In this paper, there is an equal probability that any location in C_i is assumed to be the true location, so, $P(e_i|C_i) = 1/k$. For f_i, the background knowledge that f_i is the hypothetical location e_i is attacker believes f_i is the true location t_i, i.e., the joint probability $P(f_i, t_i)$. So, the probability that f_i is assumed to be t_i is $P(f_i, t_i)\bullet P(e_i|C_i)$. The higher the probability, the higher the probability that e_i is t_i, which means that attacker believes that f_i and t_i are more similar. Therefore, we use this probability to calculate the probability distribution of geographic distance $P_G(f_i)$:

$$P_G(f_i) = 1 - P(f_i, t_i)\bullet P(e_i|C_i) = 1 - P_v(f_i)\bullet P_v(t_i)\bullet P(e_i|C_i), \tag{10}$$

According to the above formula, the geographical similarity $\mathrm{sim}_G(F, T)$ between F and T is calculated as follows:

$$\mathrm{sim}_G(F, T) = \frac{1}{z_g}\sum_{i=1}^{n}(1 - P_G(f_i)\bullet D_G(t_i, f_i)), \tag{11}$$

where $1/z_g$ is a constant used to normalize the geographical similarity value to lie in the range [0,1]. z_g is the sum of the maximum location distance difference in each candidate location C_i and user location t_i:

$$z_g = \sum_{i=1}^{n}\sum_{j=1}^{|C_i|}\max\left\{D_G\left(t_i, f_j\right)\right\}. \tag{12}$$

Definition 13. Semantic similarity: for $\forall F \in \mathrm{FT}$ and real trajectory T, $D_S(t_i, f_i)$ denotes the semantic distance between t_i and f_i, and the probability distribution of the semantic distance is denoted by $P_S(f_i)$. The semantic similarity between T and F can be defined as the expectation of all $D_S(t_i, f_i)$:

$$\sum_{i=1}^{n} P_S(f_i)\bullet D_S(t_i, f_i). \tag{13}$$

Similar to Definition 12, for Equation (13), we still calculate $P_S(f_i)$ by $P(f_i, t_i)\bullet P(e_i|C_i)$:

$$P_S(f_i) = 1 - P_v(f_i)\bullet P_v(t_i)\bullet P(e_i|C_i). \tag{14}$$

According to the above formula, the semantic similarity $\mathrm{sim}_S(F, T)$ between F and T is calculated as follows:

$$\mathrm{sim}_S(F, T) = \frac{1}{z_s}\sum_{i=1}^{n}(1 - P_S(f_i)\bullet D_S(t_i, f_i)), \tag{15}$$

where $1/z_S$ is a constant used to normalize the geographical similarity value to lie in the range [0,1]. z_S is the sum of the maximum semantic distance difference in each candidate location C_i and user location t_i:

$$z_s = \sum_{i=1}^{n}\sum_{j=1}^{|C_i|}\max\left\{D_S\left(t_i, f_j\right)\right\}. \tag{16}$$

Attacker can also analyse the trajectory's movement direction in the published trajectory set. If a trajectory differs from other trajectories, attacker deduces that this trajectory is likely to be a dummy trajectory. Therefore, it is necessary to ensure the movement direction similarity between dummy trajectory and real trajectory T. We use the least squares method to fit the slope of the overall trajectory movement direction and determine whether dummy trajectory direction is similar to real trajectory direction by the slope ratio. The slope is calculated as follows:

$$l = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \tag{17}$$

where $\bar{x} = 1/n\sum_{i=1}^{n} x_i$; $\bar{y} = 1/n\sum_{i=1}^{n} y_i$.

Definition 14. Directional similarity: for $\forall F \in \mathrm{FT}$ and real trajectory T, according to Equation (17), calculate the slope of two trajectories l_F and l_T. The directional similarity between T and F is calculated as follows:

$$\mathrm{sim}_A(F, T) = \frac{l_F}{l_T}, \mathrm{sim}_A(F, T) \in [0, 1]. \tag{18}$$

The larger the value of $\mathrm{sim}_A(F, T)$, the more similar the overall direction of the two trajectories.

4. Scheme Description

4.1. Scheme Framework. This scheme is designed to protect user's trajectory privacy. To prevent attacker from identifying user's real trajectory from the k-anonymity set. In this paper, we construct $k - 1$ dummy trajectories based on stop point location pairs to form k-anonymity set, thus protecting user privacy and security. The scheme is divided into three steps:

Trajectory preprocessing stage: based on the start time and end time of real trajectory, the trajectories with similar time periods are selected from the history trajectory dataset D. The trajectory equivalence class TC consists of these historical trajectories. According to Definition 2 and Definition 3, we extract stop points in real trajectory T and the equivalent class TC, then assign semantic to start-end points of T and all stop points to generate the corresponding semantic trajectory SEM and semantic trajectory equivalent class STC.

Selection of candidate location stage. For each location point sl_i in semantic trajectory SEM, we select the location points that satisfy Definition 8 from STC and then add them to candidate location C_i to form the candidate location set Can.

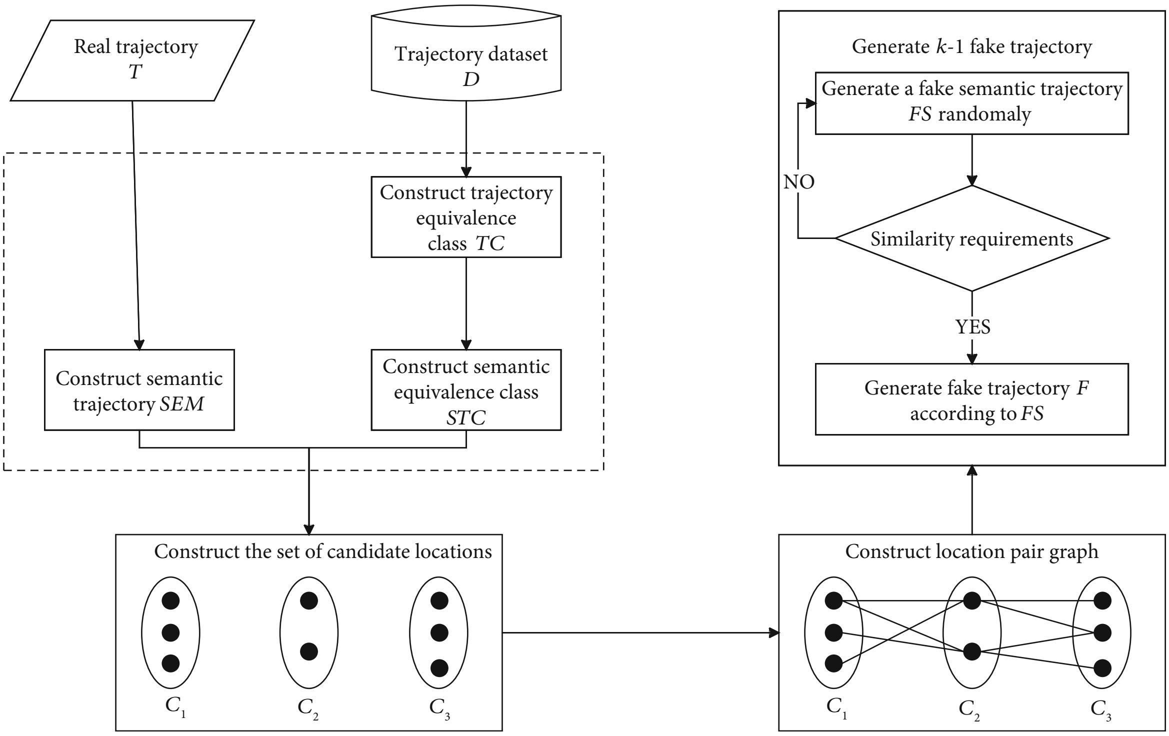

Real trajectory
T
Trajectory dataset
D
Construct trajectory equivalence class TC
Construct semantic trajectory SEM
Construct semantic equivalence class STC
Construct the set of candidate locations
C1
C2
C3
Construct location pair graph
Generate k-1 fake trajectory
Generate a fake semantic trajectory FS randomaly
NO
Similarity requirements
YES
Generate fake trajectory F according to FS

Figure 3: System overview of scheme.

(1) Dummy trajectory generation stage. We generate dummy semantic trajectory FS based on candidate location set Can. If FS meets similarity measure, moving points between stop points in FS are also added to the trajectory FS to generate corresponding dummy trajectory F. Repeat this step until $k-1$ dummy trajectories are generated to form a k-anonymity set. The overall process is shown in Figure 3. We will then describe each step separately in chapter 4.

4.2. Trajectory Preprocessing. This subsection preprocesses real trajectory. It can be divided into the following two steps: (1) construct a trajectory equivalence class TC for T and add trajectories with similar time periods to TC; (2) extract stop points in real trajectory T and the equivalent class TC and assign semantic to them, and construct the corresponding semantic trajectory SEM and semantic trajectory equivalent class STC.

4.2.1. Construct Trajectory Equivalence Class. In k-anonymity set, if the time period of a trajectory is different from other trajectories, attacker may infer that this trajectory is a dummy trajectory. Therefore, we need to ensure that the trajectory in the equivalent class TC is similar to real trajectory T in terms of time period. According to the set time offset Δ_t, if a trajectory Tra in history trajectory dataset D satisfies the following conditions:

$$\begin{cases} \text{Tra}.t_s \in [T.t_s - \Delta_t, T.t_s + \Delta_t], \\ \text{Tra}.t_e \in [T.t_s - \Delta_t, T.t_s + \Delta_t]. \end{cases} \tag{19}$$

Then, Tra is added to the equivalent class TC, where t_s and t_e denote the start time and end time of trajectory Tra, respectively.

The specific steps are shown in Algorithm 1.

Algorithm 1 traverses the trajectories in the trajectory dataset D by a for-loop and calculates whether they satisfy Equation (19). The time cost of Algorithm 1 is num(D), i.e., the algorithm time complexity is $O(n)$.

4.2.2. Construct Trajectory Equivalence Class. Stop point contains richer information than moving point, and attacker can identify k-anonymity set's dummy trajectory by analysing stop point. The dummy trajectory that constitutes the k -anonymity set have to be semantically similar to the real trajectories T. So, in this subsection, we extract stop points in real trajectories T and equivalence class TC and assign semantic to start-end points of T and all stop points to construct the corresponding semantic trajectory and semantic trajectory equivalence class STC.

When the distance between a segment of consecutive location points t_i and $t_j (i \leq j)$ is less than the distance threshold θ_d and the interval time is greater than the time threshold θ_t, the location points from t_i to t_j are aggregated into a stop point. If the Euclidean distance between t_i and t_j is greater than the distance threshold θ_d, then t_i and t_j cannot be aggregated into a stop point. Next, the interval time between t_i and t_{j-1} is calculated to determine whether it is greater than the time threshold θ_t, if the interval time is greater than the time threshold θ_t, then t_i to t_{j-1} is aggregated as a stop point. If the interval time is less than the time

```
Input: Real trajectory $T$; time offset $\Delta_t$; history trajectory dataset $D$
Output: Trajectory equivalent class TC
TC $\longleftarrow \varnothing$
forall Tra $\in D$do
  ifTra meets the candidate of Formula (19) then
    $TC \longleftarrow TC \cup$ Tra
  returnTC
```

Algorithm 1: Construct trajectory equivalence class

threshold θ_t, the location points t_i to t_j are all moving points and repeat this operation from t_{j+1}.

For example, in the subtrajectory $tr = \{t_1, t_2, \cdots, t_5, t_6\}$, $\mathrm{Dis}_{\mathrm{geo}}(t_1, t_5) \leq \theta_d$, $\mathrm{Dis}_{\mathrm{geo}}(t_1, t_6)$, and $\mathrm{Dis}_{\mathrm{time}}(t_1, t_5) \geq \theta_t$. We can aggregate the location points t_1 to t_5 as a stop point. After acquiring stop points, we assign semantic of the closest POI to stop point based on the PAT. The step is repeated, and the semantic trajectory is finally generated.

The specific steps are shown in Algorithm 2.

In Algorithm 2, lines 2-5 assign semantic to start-end points of real trajectory T and add them to semantic trajectory SEM. Lines 6-15 extract stop points in real trajectory T based on the threshold θ_t and distance threshold θ_d and assign semantic. Then, add these points to SEM. Lines 15-27 traverse the trajectories in the equivalence class TC, convert the trajectory to semantic trajectory, and add them to semantic trajectory equivalence class STC.

Algorithm 2 transforms real trajectory T into semantic trajectory SEM by two layers of while-loop. In the worst case, the time consumption is $\mathrm{num}(T) * \mathrm{num}(T)$, and the time complexity is $O(n^2)$. Second, the trajectories in TC are traversed by a for-loop, and then, the equivalence class is transformed into semantic trajectory equivalence class STC by two layers of while-loop. In the worst case, the time consumption is $\mathrm{num}(\mathrm{TC}) * \mathrm{num}(T) * \mathrm{num}(T)$, and the time complexity is $O(n^3)$. Therefore, the total time consumption of Algorithm 2 is $\mathrm{num}(\mathrm{TC} * T * T) + \mathrm{num}(T^2)$, and the total time complexity is $O(n^3)$.

4.3. Construct Candidate Location Set. The purpose of this subsection is to select the location pairs used to construct the dummy trajectory. We traverse semantic trajectory SEM and compute the similarity of location pairs in STC according to Definition 8. If there is a location pair that satisfies the condition, this location pair is added to the candidate location C_i and C_{i+1}. After traversing all position pairs, the composition of $\mathrm{Can} = \{C_1, C_2, \cdots, C_n,\} = \{<C_1, C_2>, <C_2, C_3>, \cdots, <C_{n-1}, C_n>\}$.

The specific steps are shown in Algorithm 3.

For the location pair $<\mathrm{sl}_i, \mathrm{sl}_{i+1}>$ in the semantic trajectory SEM, Algorithm 3 traverses each trajectory in STC. If there exists position pair similar to $<\mathrm{sl}_i, \mathrm{sl}_{i+1}>$ in this trajectory, this location pair will be added to C_i and C_{i+1}. Then, traverse the next trajectory until all trajectories have been traversed. Algorithm 3 finally returns the candidate location set Can.

Algorithm 3 constructs the candidate location set Can by a three-level for-loop. The first level for-loop traverses the location pairs in SEM with a time consumption of num(SEM). The second level for-loop traverses the trajectory in *STC* with a time consumption of num(STC). The third-level for-loop traverses the location pairs in the trajectory stc with a time consumption of num(stc). Therefore, the total time consumption of Algorithm 3 is $\mathrm{num}(\mathrm{SEM} * \mathrm{STC} * \mathrm{stc})$, and the time complexity is $O(n^3)$.

4.4. Construct Location Pair Graph. In the previous subsection, we picked candidate location for each semantic location point in SEM. The next goal is to generate $k-1$ dummy trajectories based on these candidate positions to form a k-anonymity set. If location points are randomly selected from each candidate location and combined into a dummy trajectory, the generated dummy trajectory may have some unreachable locations. Specifically, as shown in Figure 4, the solid line and dotted line represent the location pair that exist in Can and the location pair that do not exist in Can, respectively. A dummy trajectory consisting of location pairs $<A_1, B_2>$ and $<B_2, A_3>$ will pass through unreachable locations. So, attacker can easily identify this trajectory as dummy trajectory.

This subsection constructs the location pair graph $G = (V, E, W)$. Among them, V consists of all semantic location points in Can, and each semantic location point represents a node in G. c_i is a location in the candidate position C_i and $E(c_i, c_{i+1})$ is a directed edge connecting c_i and c_{i+1}. $W(c_i, c_{i+1})$ is the weight of edge $E(c_i, c_{i+1})$, and the value is a binary consisting of geographical similarity and semantic similarity. Algorithm 4 describes the graph generation process as follows.

Algorithm 4 first traverses each candidate location C_i and picks locations from C_i and C_{i+1} respectively to form the location pair $<c_j, c_k'>$. Next, determine whether the location pair $<c_j, c_k'>$ exists in STC. If it exists, add $E(c_i, c_{i+1})$ and $W(c_i, c_{i+1})$ to the graph G.

Algorithm 4 constructs the location pair graph G by a three-level for-loop. The first level for-loop traverses the candidate location of Can, and the time consumption is $\mathrm{num}(\mathrm{Can})$. The second level for-loop traverses location point c_j in candidate position C_i, and the time consumption is $\mathrm{num}(C_i)$. The third level for-loop traverses the location point c_k' in the candidate position C_{i+1}, and the time consumption of $\mathrm{num}(C_{i+1})$. The total time consumption of

```
Input: Real trajectory $T$; time threshold $\theta_t$; distance threshold $\theta_d$; point-of-interest set
       PAT
Output: Semantic trajectory Sem; Semantic trajectory equivalence class STC
$\mathrm{SEM} \longleftarrow \varnothing, \mathrm{STC} \longleftarrow \varnothing$
$\mathrm{sl} = \mathrm{assign\_sem}(t_1, \mathrm{PAT})$
$\mathrm{SEM} \longleftarrow \mathrm{SEM} \cup \mathrm{sl}$
$\mathrm{sl} = \mathrm{assign\_sem}(t_e, \mathrm{PAT})$
$\mathrm{SEM} \longleftarrow \mathrm{SEM} \cup \mathrm{sl}$
while all $t_i \in \{t_2, t_3, \cdots, t_{e-1}\}$ do
$j \longleftarrow i + 1$
while $t_j \in \{t_2, t_3, \cdots, t_{e-1}\}$ do
  if Dis_geo$(t_i, t_j) < \theta_d$ then
            if Dis_time$(t_i, t_j) > \theta_d$ then
                  $\mathrm{sp} = \mathrm{gen\_sp}(t_i, t_j)$
                  $\mathrm{sl} = \mathrm{assign\_sem}(\mathrm{sp}, \mathrm{PAT})$
                  $\mathrm{SEM} \longleftarrow \mathrm{SEM} \cup \mathrm{sl}$
        $j \longleftarrow j + 1$
      $i \longleftarrow j + 1$
for all $\mathrm{Tc} \in \mathrm{TC}$ do
      while all $\mathrm{tc}_i \in \mathrm{Tc}$ do
            $j \longleftarrow i + 1$
            while $\mathrm{tc}_j \in \mathrm{Tc}$ do
                  if Dis_geo$(\mathrm{tc}_i, \mathrm{tc}_j) < \theta_d$ then
                  if Dis_time$(\mathrm{tc}_i, \mathrm{tc}_j) > \theta_d$ then
                        $\mathrm{sp} = \mathrm{gen\_sp}(\mathrm{tc}_i, \mathrm{tc}_j)$
                        $\mathrm{cl} = \mathrm{assign\_sem}(\mathrm{sp}, \mathrm{PAT})$
                        $\mathrm{stc} \longleftarrow \mathrm{stc} \cup \mathrm{cl}$
                  $j \longleftarrow j + 1$
      $i \longleftarrow j + 1$
   $\mathrm{STC} \longleftarrow \mathrm{STC} \cup \mathrm{stc}$
return SEM, STC
```

Algorithm 2: Construct semantic trajectory

```
Input: Semantic trajectory SEM; semantic trajectory equivalence class STC; location
       pair similarity threshold $\delta_{\mathrm{sem}}, \delta_{\mathrm{geo}}, \delta_{\mathrm{time}}$
Output: Candidate location set Can
$\mathrm{Can} \longleftarrow \varnothing$
for all $< \mathrm{sl}_i, \mathrm{sl}_{i+1} > \in \mathrm{SEM}$ do
$C_i \longleftarrow \mathrm{sl}_i, C_{i+1} \longleftarrow \mathrm{sl}_{i+1}$
for all $\mathrm{stc} \in \mathrm{STC}$ do
          for all $cl_j \in \mathrm{stc}$ do
                if $<\mathrm{sl}_i, \mathrm{sl}_i>$ and $<cl_j, cl_{j+1}>$
              meets the candidate of
              Definition8 then
                      $C_i \longleftarrow cl_j, C_{i+1} \longleftarrow cl_{j+1}$
                      exit
$\mathrm{Can} \longleftarrow C_i, \mathrm{Can} \longleftarrow C_{i+1}$
return Can
```

Algorithm 3: Construct candidate location set

Algorithm 4 is $\mathrm{num}(\mathrm{Can} * C_i * C_{i+1})$, and the total time complexity is $O(n^3)$.

4.5. Construct k-Anonymity Set. The goal of this subsection is to generate $k-1$ dummy trajectories to form k-anonymity sets. These dummy trajectories are semantically, geographically, and directionally similar with real trajectories T. To achieve this goal, dummy semantic trajectory FS is first generated. If FS satisfies the requirements in semantic similarity and geographic and directional similarity, a dummy

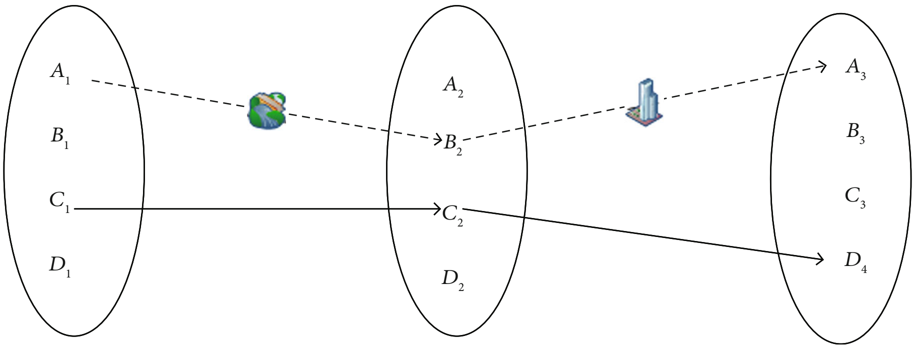


Figure 4: Example of constructing fake trajectory.

```
Input: Semantic trajectory SEM; location candidate set Can; semantic trajectory
        equivalence class STC
Output: Location pair graph $G$
$E \longleftarrow \varnothing$, $W \longleftarrow \varnothing$
**for**all $C_i \in$ Can**do**
    **for**all $c_j \in C_i$**do**
        **for**all $c'_k \in C_{i+1}$**do**
            **if**$<c_j, c'_k> \in$STC**then**
                $E \longleftarrow E(c_j, c'_k)$
                $W \longleftarrow W(c_j, c'_k)$
**return**$G$
```

Algorithm 4: Construct location pair graph

trajectory F is generated based on FS. The dummy trajectory F consists of each semantic position pair in FS and the moving points between corresponding location pairs.

Algorithm 5 describes the generation process of dummy trajectory. First, the directed edges that do not satisfy the threshold requirement are removed from the location pair graph G. Then, iteratively generate dummy trajectory up to $k-1$ trajectories. Lines 3-12 show the detailed steps of dummy trajectory generation. First, a dummy semantic trajectory is randomly generated (line 5). If the generated trajectory happens to be the user semantic trajectory, a new semantic trajectory is regenerated (lines 5-6). If it is not the user semantic trajectory, determine whether FS satisfies the similarity requirement (line 8). If the requirement is satisfied, iteratively traverse the location pair in FS, find the location pair in STC, then add stop points and moving points contained in location pair to the dummy trajectory F (lines 9 to 11). Finally, the eligible dummy trajectories are added to FT. The specific algorithm is as follows.

Algorithm 5 generates $k-1$ dummy trajectories by while-loop, and the time consumption is $k-1$. In the dummy trajectory generation process, the dummy trajectory F is constructed by two-level for-loop. In the worst case, the time consumption is $\text{num}(\text{FS}) * \text{num}(\text{STC})$. Therefore, the total time consumption of Algorithm 5 is $((k-1) * \text{num}(\text{FS}) * \text{num}(\text{STC})$, and the total time complexity is $O(n^3)$.

According to the above analysis, the total time complexity of this scheme is $O(n^3)$.

5. Experiment Analysis

5.1. Data Set and Experimental Environment. In this paper, we use the GeoLife dataset [26–28] to evaluate the performance of the trajectory privacy protect method based on location pair reorganization. The dataset collected 5 years of trajectories of 182 volunteers, and this dataset contains 17,621 trajectories. Most of the trajectories in GeoLife dataset are recorded in Beijing, China. Therefore, this paper extracts the trajectory data of the Beijing area for experimental analysis. We select 5 attributes in GeoLife to compose the history trajectory dataset D. The 5 attributes are user ID, longitude, latitude, date, and time. We generate the trajectory k-anonymity set on the historical trajectory dataset D.

In this paper, the point of interest set (PAT) includes a total of 211,615 POIs within the 6th Ring Road of Beijing. The semantic categories adopt the three-level classification of Chinese POI standard, including 15 major categories, 51 middle categories, and 145 minor categories.

The hardware environment of the experiment is: Intel i7-8750H 2.20 GHz, 16.00 GB memory, the operating system is Microsoft Windows 10, and the algorithms are all implemented under Pycharm2020.

5.2. Experimental Parameter and Evaluation Indicator. To verify the performance of the DSTPP algorithm, we randomly select 10 user trajectories from the trajectory dataset D for the experiment. Each user's semantic trajectory

```
Input: Semantic trajectory SEM; location pair graph $G$; semantic similarity threshold $\delta_S$;
       Geographical similarity threshold $\delta_G$; directional similarity threshold $\delta_A$;
       Semantic trajectory equivalence class STC
Output: Dummy trajectory set FT
FT $\longleftarrow \varnothing$, $n(\mathrm{FT}) \longleftarrow 0$
Calculate the slope $l_{\mathrm{sem}}$ of SEM
Remove all $E(c_i, c_{i+1})$ which $W(c_i, c_{i+1})$ do not satisfy Def(10) and Def(11) from G
while $n(\mathrm{FT}) < k-1$ do
    Random generates a fake semantic trajectory FS
    if FS = SEM then
        $n(\mathrm{FT}) = n(\mathrm{FT})$
    else
        if $\mathrm{sim}_S(\mathrm{SEM}, \mathrm{FS}) < \delta_S$ and $\mathrm{sim}_G(\mathrm{SEM}) < \delta_G$ and $\mathrm{sim}_A(\mathrm{SEM}, \mathrm{FS}) > \delta_A$
         then
            for all $< fs_i, fs_{i+1} > \in$ FS
                for all stc $\in$ STC do
                    if $< fs_i, fs_{i+1} > \in$ stc then
                        all point between $fs_i$ and $fs_{i+1}$ join $F$
            FT $\longleftarrow$ FT $\cup$ $F$
return FT
```

Algorithm 5: Generate fake trajectory set

contains at least 4 semantic location points. The experimental parameters are set as shown in Table 2.

In this paper, we verify the performance of DSTPP by comparing with Random algorithm [25] and MTPPA [13] algorithm in terms of both privacy and availability. Random algorithm randomly selects $k-1$ locations from the candidate location to generate dummy trajectory. MTPPA algorithm selects $k-1$ history trajectories to form k-anonymity set. Under the same conditions, we run 3 algorithms on 10 user trajectories. To ensure the accuracy of the experimental results, each group of experiments was measured 100 times. Finally, the average of 100 results was taken as the final result.

5.3. Privacy Analysis. In this section, we use identification probability (IP) to evaluate k-anonymity set's privacy. In k-anonymity set, there are m trajectories that are similar to T. The identification probability is $1/m$. The larger the value of m, the more difficult it is for attacker to identify true trajectory T from k-anonymity set and the better privacy. The identification probability is calculated by the following equation.

$$\mathrm{IP} = \frac{1}{\mathrm{tra\_num}(k_{\mathrm{set}}, \delta_G, \delta_A, \delta_S)}, \tag{20}$$

where $\mathrm{tra\_num}(k_{\mathrm{set}}, \delta_G, \delta_A, \delta_S)$ is used to calculate the number of trajectories that satisfy the similarity measure in the trajectory k-anonymity set.

We evaluate the privacy by the average identification probability of 10 trajectories. Figure 5 shows the three methods' average identification probability under different values of k. As shown in Figure 5, the average identification probability of all methods tends to decrease when the k value increases. This is because as the k value increases, the more trajectories in k-anonymity set and the more trajectories that are similar to the real trajectory. Under the same k value,

Table 2: Experimental parameters.

Parameter	Range
θ_t	10 min
θ_d	100 m
δ_{geo}	200 m
δ_{sem}	98
δ_{time}	10 min
δ_G	0.3
δ_S	0.3
δ_A	0.85
Anonymity level	3-10

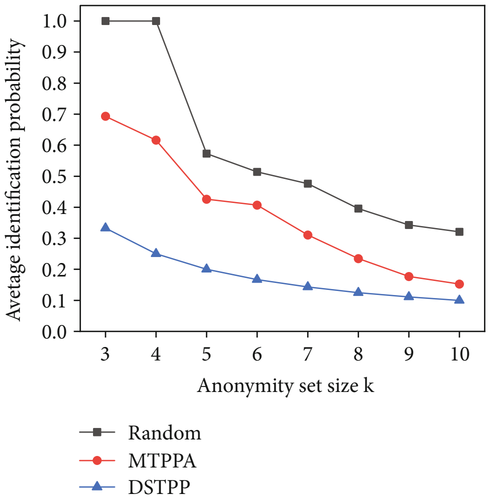


Figure 5: Privacy vs. k.

random algorithm has the highest average identification probability; this is because (1) the dummy trajectory passes through some unreachable locations and (2) the dummy trajectory do not consider geographic and semantic attribute. Attacker can easily filter dummy trajectories in k-anonymity set by analysing the stop points; (3) the dummy trajectory may differ significantly from the real trajectory in the overall direction.

The average identification probability of MTPPA algorithm is lower than random algorithm. This is because MTPPA algorithm selects history trajectories to form a k -anonymous set, so there are no unreachable locations. Meanwhile, dummy trajectory is selected based on the number of stop point, which can ensure the semantic similarity of trajectories in the k-anonymity set to a certain degree.

Compared to the above two methods, DSTPP has a lower average identification probability. This is because DSTPP constructs dummy trajectory based on real position pairs, so no unreachable locations appear. In addition, dummy trajectory and real trajectories are similar in terms of overall direction, semantic property, and geographical property.

5.4. Availability Analysis. After the service provider publishes the k-anonymity set, researchers can conduct statistics, research, and other studies by mining the trajectories in the anonymity set. Therefore, the k-anonymity set's availability is to be evaluated. According to the research needs, availability can be divided into two categories: trajectory availability and data availability.

5.4.1. Trajectory Availability Analysis. Trajectory availability means: for traffic optimization, logistics management, and other needs, researchers want to use trajectory data to evaluate traffic flow. This requires that the dummy trajectory must have a high degree of shape similarity and geographic similarity to the real trajectory.

In this subsection, we evaluate the trajectory availability of k-anonymous sets by trajectory difference (TD). The trajectory difference is calculated by the following equation.

$$\mathrm{TD} = \frac{\sum_{i=1}^{|F|} F(T, F_i)}{k}, \tag{21}$$

where $F(T, F_i)$ is the Fréchet distance between two trajectories. Fréchet distance is calculated by the following equation.

$$F(A, B) = \inf_{\alpha,\beta,t\in[0,1]} \max \{d(A(\alpha(t)), B(\beta(t)))\}. \tag{22}$$

A and B are two trajectories for comparison, with t representing a specific moment and $A(\alpha(t)), B(\beta(t))$ representing A and B's location point at moment t, respectively. $d(A(\alpha(t)), B(\beta(t)))$ denotes the Euclidean distance between two location points.

We use the average trajectory difference of the 10 trajectories to evaluate the k-anonymity set's trajectory availability. For the k-anonymous set, larger TD values represent weaker trajectory availability. Figure 6 shows three methods' average trajectory difference of under different k values. As shown in Figure 6, three methods' trajectory difference shows an increasing trend when the k value increases. This is because, as the k value becomes larger, the number of required dummy trajectories increases. So, the dummy trajectories' overall quality decreases. Under the same k value, DSTPP has the lowest trajectory difference. This is because DSTPP considers the overall direction and geographic location, and the generated dummy trajectory is more similar to the real trajectory. Random's average trajectory difference is higher than DSTPP. This is because Random randomly generates dummy trajectory based on fake location points. The generated dummy trajectory has some geographical similarity, but the overall direction is not considered. The MTPPA algorithm has the largest average trajectory difference. This is because MTPPA does not consider trajectory shape similarity and overall direction at all when selecting dummy trajectory.

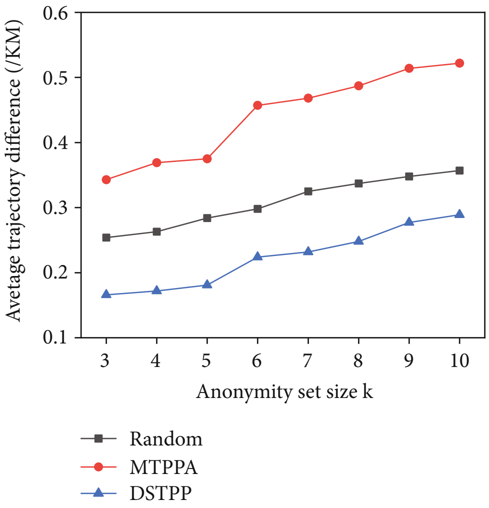


Figure 6: Trajectory difference vs. k.

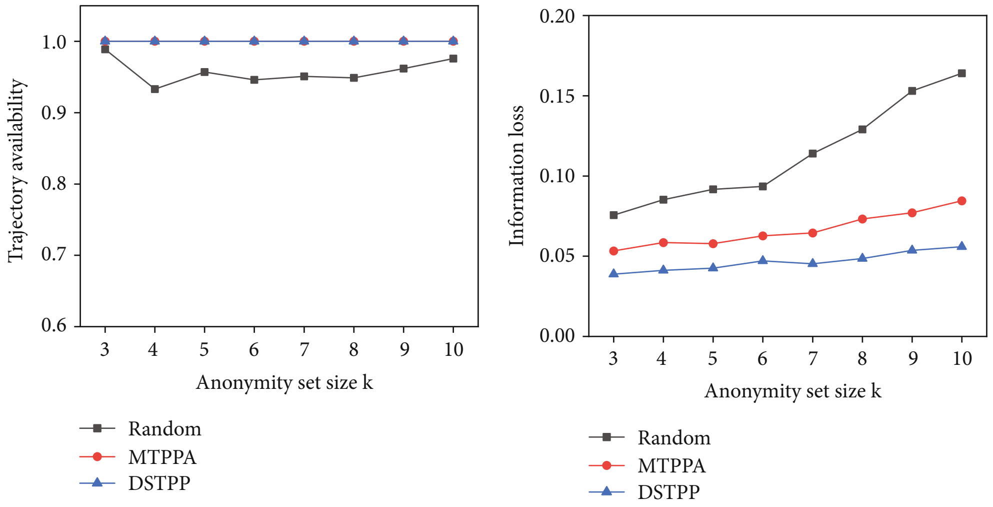
5.4.2. Data Availability Analysis. Data availability is pointed out for needs such as interest recommendation and location prediction [26, 35]; researchers want to analyse semantic properties in trajectories. This requires that there must be enough available data in the k-anonymity set and the data must be accurate.

In this paper, the available data rate (AD) is used to evaluate the amount of available data in the anonymous set, which is calculated as follows.

$$\mathrm{AD} = \frac{\mathrm{num\_reach}(k_{\mathrm{set}})}{\mathrm{num\_all}(k_{\mathrm{set}})}, \tag{23}$$

where $\mathrm{num\_all}(k_{\mathrm{set}})$ denotes all the location points in the k -anonymous set and $\mathrm{num\_reach}(k_{\mathrm{set}})$ denotes the reachable location points in the k-anonymous set.

We use information loss (IL) to evaluate the accuracy of the anonymous set. Less information loss indicates more accurate data and better data availability. Information loss depends on the size of the anonymity zone; the method



(a) Available data vs. k

(b) Data availability vs. k

Figure 7: Availability vs. k.

was similar with Xu et al. [13]. The calculation formula is as follows.

$$\mathrm{IL} = \frac{\sum_{i=1}^{|\mathrm{SEM}|} \mathrm{Area}(\mathrm{loc}_i, \mathrm{time}_i)/\mathrm{Max\_Area}(k_{\mathrm{set}})}{|\mathrm{SEM}|}, \qquad (24)$$

where $\mathrm{Area}(\mathrm{loc}_i, \mathrm{time}_i)$ denotes the ith semantic location point's anonymous set size, and $\mathrm{Max\_Area}(k_{\mathrm{set}})$ denotes the size of the trajectory k-anonymous set, and $|\mathrm{SEM}|$ denotes the number of semantic location points.

We evaluate the k-anonymity set's data availability by the average available data rate and the average loss information for the 10 trajectories. Figure 7 illustrates the effect of k value on data availability.

Figure 7(a) evaluates three methods' available data rate. Among them, Random has the lowest available data rate. This is because, this method contains or passes through some unreachable locations that are not available for data analysis. The available data rate of both MTPPA and DSTPP is 1. This is because the dummy trajectory of MTPPA is taken from history trajectories set, and the dummy trajectory by DSTPP is also taken from history trajectories. Therefore, no unreachable positions will be present or passed.

Figure 7(b) evaluates three methods' data accuracy. When k value increases, the information loss of all three methods shows an increasing trend. This is because the quality of dummy trajectory decreases as the k-anonymity set increases. When k values are constant, Random has the greatest loss information. This is because the dummy trajectory generated by Random does not consider semantic similarity. So, there is too much noise in the query results of each query. MTPPA's information loss is in the middle; MTPPA considers the number of stop point between trajectories. Therefore, the query result of each query contains a high amount of semantic information, which can be used for data analysis. DSTPP has minimal information loss. This is because the dummy trajectory generated by DSTPP not only considers the number of stop point but also considers the semantics of stop point are similar to the real trajectory. The query results contain rich semantic information for researchers to analyse.

6. Conclusions

With the rapid development of information technology and 5G technology, human society has entered the era of big data. Analysing the user's daily action trajectory is of great help in optimizing national resource scheduling and improving public facilities. User trajectories contain sensitive information: how to ensure the availability of trajectories on the premise of ensuring user privacy and security. Aiming at this problem, this paper proposes a trajectory privacy protect method based on location pair reorganization. The real trajectory is decomposed into location pairs consisting of start-send point and stop points. Then, $k-1$ dummy trajectories are generated by selecting eligible location pairs from the historical trajectory set to form trajectory k-anonymous. Finally, this paper experiments privacy and availability on real dataset. By comparing with MTPPA algorithm and Random algorithm, our method reduces information loss and improves privacy protection. Overall, MTPPA algorithm is better than MTPPA algorithm and Random algorithm.

Data Availability

The trajectory data used to support the findings of this study can be downloaded from https://www.microsoft.com/en-us/download/details.aspx?id=52367 and the detailed instructions can be found in https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the Natural Science Foundation of Heibei Province (F2019201361)

References

[1] I. A. Junglas and R. T. Watson, "Location-based services," *Communications of the ACM*, vol. 51, no. 3, pp. 65–69, 2008.

[2] J. Krumm, "A survey of computational location privacy," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, 2009.

[3] C. Y. Chow and M. F. Mokbel, "Trajectory privacy in location-based services and data publication," *ACM Sigkdd Explorations Newsletter*, vol. 13, no. 1, pp. 19–29, 2011.

[4] A. Altomare, E. Cesario, C. Comito, F. Marozzo, and D. Talia, "Trajectory pattern mining for urban computing in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 1–599, 2016.

[5] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *2012 IEEE 12th international conference on data mining*, pp. 1038–1043, Brussels, Belgium, 2012.

[6] D. Luper, D. Cameron, J. A. Miller, and H. R. Arabnia, "Spatial and temporal target association through semantic analysis and GPS data mining," *IKE*, vol. 7, pp. 25–28, 2007.

[7] X. Shen, L. Wang, Q. Pei, Y. Liu, and M. Li, "Location privacy-preserving in online taxi-hailing services," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 69–81, 2021.

[8] A. M. Sazdar, S. A. Ghorashi, V. Moghtadaiee, A. Khonsari, and D. Windridge, "A low-complexity trajectory privacy preservation approach for indoor fingerprinting positioning systems," *Journal of Information Security and Applications*, vol. 53, article ???, 2020.

[9] S. Zhang, X. Mao, K. K. R. Choo, T. Peng, and G. Wang, "A trajectory privacy-preserving scheme based on a dual-K mechanism for continuous location-based services," *Information Sciences*, vol. 527, pp. 406–419, 2020.

[10] Y. Zhao, Y. Luo, Q. Yu, and Z. Hu, "A privacy-preserving trajectory publication method based on secure start-points and end-points," *Mobile Information Systems*, vol. 2020, 12 pages, 2020.

[11] K. Zhou and J. Wang, "Trajectory protection scheme based on fog computing and K-anonymity in IoT," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–6, Matsue, Japan, 2019.

[12] H. J. Xu, Q. H. Wu, and X. M. Hu, "Privacy protection algorithm based on multi-characteristics of trajectory," *Computer Science*, vol. 46, pp. 190–195, 2019.

[13] Z. Xu, J. Zhang, P. Tsai, L. Lin, and C. Zhuo, "Spatiotemporal mobility based trajectory privacy-preserving algorithm in location-based services," *Sensors*, vol. 21, no. 6, p. 2021, 2021.

[14] Y. Wang, M. Li, S. Luo et al., "LRM: a location recombination mechanism for achieving trajectory k-anonymity privacy protection," *IEEE Access*, vol. 7, pp. 182886–182905, 2019.

[15] C. Y. Lin, "Suppression techniques for privacy-preserving trajectory data publishing," *Knowledge-Based Systems*, vol. 206, article ???, 2020.

[16] C. S. H. Eom, C. C. Lee, W. Lee, and C. K. Leung, "Effective privacy preserving data publishing by vectorization," *Information Sciences*, vol. 527, pp. 311–328, 2020.

[17] Z. Hu and J. Yang, "Differential privacy protection method based on published trajectory cross-correlation constraint," *Plos one*, vol. 15, no. 8, article e0237158, 2020.

[18] X. Zhao, D. Pi, and J. Chen, "Novel trajectory privacy-preserving method based on clustering using differential privacy," *Expert Systems with Applications*, vol. 149, article 113241, 2020.

[19] Y. Li, D. Yang, and X. Hu, "A differential privacy-based privacy-preserving data publishing algorithm for transit smart card data," *Transportation Research Part C: Emerging Technologies*, vol. 115, article 102634, 2020.

[20] X. Zhao, D. Pi, and J. Chen, "Novel trajectory privacy-preserving method based on prefix tree using differential privacy," *Knowledge-Based Systems*, vol. 198, article ???, 2020.

[21] W. Wu, Y. Zhao, and Q. Wang, "A safe storage and release method of trajectory data satisfying differential privacy," *Journal of Computer Research and Development*, vol. 58, no. 11, p. 2430, 2021.

[22] S. Wang, C. Chen, G. Zhang, and Y. Xin, "Interchange-based privacy protection for publishing trajectories," *IEEE Access*, vol. 7, pp. 138299–138314, 2019.

[23] S. Li, H. Shen, Y. Sang, and H. Tian, "An efficient method for privacy-preserving trajectory data publishing based on data partitioning," *The Journal of Supercomputing*, vol. 76, no. 7, pp. 5276–5300, 2020.

[24] Z. Huo, X. Meng, H. Hu, and Y. Huang, "You can walk alone: trajectory privacy-preserving through significant stays protection," in *International conference on database systems for advanced applications*, pp. 351–366, Berlin, Heidelberg, 2012.

[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 *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.

[26] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proceedings of International conference on World Wild Web (WWW 2009)*, pp. 791–800, Madrid Spain, 2009.

[27] Z. Yu, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proceedings of CM conference on Ubiquitous Computing (UbiComp 2008)*, pp. 312–321, Seoul, Korea, 2008.

[28] Y. Zheng, X. Xie, and W.-Y. Ma, "GeoLife: a collaborative social networking service among user, location and trajectory," *IEEE Data Engineering Bulletin*, vol. 33, no. 2, pp. 32–40, 2010.

[29] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[30] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, pp. 31–42, 2003.

[31] X. Pan, X. Meng, and J. Xu, "Distortion-based anonymity for continuous queries in location-based mobile services," in *Proceedings of the 17th ACM SIGSPATIAL International*

Conference on Advances in Geographic Information Systems, pp. 256–265, 2009.

[32] A. Ye, Q. Zhang, Y. Diao, J. Zhang, H. Deng, and B. Cheng, "A semantic-based approach for privacy-preserving in trajectory publishing," *IEEE Access*, vol. 8, pp. 184965–184975, 2020.

[33] Y. Dai, J. Shao, C. Wei, D. Zhang, and H. T. Shen, "Personalized semantic trajectory privacy preservation through trajectory reconstruction," *World Wide Web*, vol. 21, no. 4, pp. 875–914, 2018.

[34] *Classification and Coding of Geographic Information Points of Interest*, Standard GB/T 35648-2017, 2017.

[35] X. Cao, G. Cong, and C. S. Jensen, "Mining significant semantic locations from GPS data," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1009–1020, 2010.