World Wide Web (2018) 21:875–914
DOI 10.1007/s11280-017-0489-2

Personalized semantic trajectory privacy preservation through trajectory reconstruction

Yan Dai[1] · Jie Shao[1] · Chengbo Wei[1] · Dongxiang Zhang[1] · Heng Tao Shen[1]

Received: 6 October 2016 / Revised: 3 August 2017 / Accepted: 4 August 2017/
Published online: 26 August 2017

Abstract Trajectory data gathered by mobile positioning techniques and location-aware devices contain plenty of sensitive spatial-temporal and semantic information, and can support many applications through data analysing and mining. However, attribute-linkage and re-identification attacks on such data may cause privacy leakage, and lead to unexpected serious consequences. Existing privacy preserving techniques for trajectory data often ignore the different privacy requirements of different moving objects or largely scarify the availability of trajectory data. In view of these issues, we propose an effective personalized trajectory privacy preserving method which can strike a good balance between user-defined privacy requirement and data availability in off-line trajectory publishing scenario. The main idea is to firstly label semantic attributes of all sampling points on the trajectory and build a corresponding taxonomy tree, next extract sensitive stop points, then for different types of sensitive stop points, adopt different strategies to select the appropriate points of user interests to replace while considering user speed and avoiding reverse mutation, and finally publish the reconstructed trajectory. Besides, to make our method more realistic we further consider possible obstacles appeared in the user space environment. In

✉ Jie Shao
shaojie@uestc.edu.cn

Yan Dai
daiyan@std.uestc.edu.cn

Chengbo Wei
chengbowei@std.uestc.edu.cn

Dongxiang Zhang
zhangdo@uestc.edu.cn

Heng Tao Shen
shenhengtao@uestc.edu.cn

[1] Center for Future Media, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

the experiments, average identification possibility, trajectory semantic consistency and trajectory shape similarity are taken as evaluation criteria, and the performance of our method is comprehensively evaluated. The results show that our method can improve the user trajectory availability as much as possible, while effectively achieving the different trajectory privacy requirements.

Keywords Trajectory database · Privacy preservation · Semantic attributes · Replacement of stop points · Trajectory reconstruction

1 Introduction

In recent years, the widespread usage of mobile positioning techniques and location-aware devices, such as global positioning systems (GPS) and radio frequency identification devices (RFID), has made massive trajectory data easy to obtain. These trajectories can support many applications related to moving objects through data analysing and mining. For example, some mobile location-aware devices collect users' trajectory data, and release these data to business organizations for commercial applications (e.g., sending advertisements or pushing services). Besides, many research organizations track volunteers' position information everyday in order to collect data for research purposes (e.g., intelligent transportation system or city traffic planning). However, while published trajectory data can be utilized to bring people huge benefits, *attribute-linkage attack* and *re-identification attack* on such data may cause serious privacy leakage. Attribute-linkage attack can be explained as malicious attackers infer the personal privacy information of specific users, such as health condition and political belief, without obtaining the user's complete trajectory. Re-identification attack means to associate a unique trajectory data record with its corresponding moving object, and then infer the user's sensitive information. After obtaining the sensitive user information, attackers may send unsolicited advertising messages to users, or even threaten the security of user's life and property safety by utilizing the home address information inferred from the obtained trajectory. Therefore, privacy protection becomes an urgent and challenging problem for of trajectory data [2, 4, 14, 20].

Current studies on trajectory privacy protection mainly deal with the following two application scenarios.

– **On-line continuous querying:** In many location-based services, users issue query requests continuously, and they need to provide their location information all the time. Thus, we need to protect the privacy of these locations with a strong real-time processing ability. Since the privacy of user dynamic trajectory is protected before the whole trajectory is collected, the data collector can hardly obtain the correct trajectory database. It seems that protecting the entire trajectory database can mostly be achieved through real-time protection of user query location information. Therefore, in this case, existing studies pay more attentions to the privacy of user locations, such as k-anonymity or region cloaking for locations, which are *on-line* and *service-driven*. However, only protecting user location information cannot protect the real-time trajectory privacy, as attackers still can infer users' sensitive information through using cloaked regions of individual locations. For example, when we protect the privacy of users through using location k-anonymity model, the location and size of cloaking regions are updated continuously ensuring that each cloak region includes $k - 1$ other locations. However, if attackers connect the cloaking regions at different time, a rough

trajectory can be inferred [29]. Therefore, it is confronted with great challenges for preserving trajectory privacy in on-line continuous querying applications.

– **Off-line trajectory publishing:** In many applications, location service providers or other organizations usually collect a trajectory database which consists of moving objects' trajectory data records, and then publish it to third parties for various commercial purposes. For example, a drivers' trajectory database can help to analyse the transportation network, and improve the traffic of city. By analysing people's daily trajectory data, social scientists can study the behavior patterns of human. However, if attackers get the trajectory database directly, users' sensitive information will be fully exposed. Therefore, trajectory privacy can be protected after collectors obtaining the trajectory data, but before publishing, so that privacy preservation is *off-line* and *data-driven*. In this paper, we are committed to this off-line trajectory publishing scenario, and try to take a good balance of privacy protection and data availability.

Some protection methods have been proposed in trajectory publishing, such as suppression release or k-anonymity of trajectory, but most of them do not consider the different privacy protection requirements of different moving objects [19]. If we adopt some strategies that cannot reasonably meet the requirement of the user's privacy level, we cannot protect the user's trajectory privacy very well. If we adopt some strategies that have exceeded the user's privacy level, over-protection would lead to an increased loss of user sensitive information and trajectory data availability [7]. Moreover, the semantic attributes of trajectory have not been fully considered by the existing methods. If we analyse the published trajectory data with large semantic deviation, it might result in misleading analysis conclusions. In addition, for different moving objects, some semantic attributes are sensitive that cannot be leaked, while some are not sensitive that can be published directly. Therefore, *sensitive attribute settings and privacy protection requirements for different moving objects are not always the same*. In our work, users are allowed to define their own sensitive semantic attribute sets and privacy levels of trajectory protection.

Our method aims to extract out stop points among user trajectory and then choose appropriate points of interest (POIs), to replace the corresponding stop points, rather than all sampling points on the trajectory. This is because that users care more about long-stayed positions, frequently visited positions or positions associated with user sensitive semantic attribute (we mark all these positions as "stop points" among the user trajectory), instead of all positions where they just passed by. These stop points contain more sensitive information, and are more likely to reveal the purpose and significance of the user trajectory, so an attacker can easily infer the user's personal privacy through analysing these stop points. Therefore, trajectory privacy preservation can be realized through just protecting these stop points instead of all sampling points [26]. This method can not only ensure the level of privacy protection, but also prevent heavy trajectory information loss and decrease the calculation overhead. Moreover, the neighboring positions' distributions of stop points are different, so according to [16] these stop points can be divided into three types, namely *non-isolated stop point*, *isolated stop point* and *quite-isolated stop point*. We adopt different strategies to reasonably select the appropriate POIs considering the semantic and spatio-temporal attributes.

For trajectory reconstruction, most previous studies regard the user space region as an Euclidean space or a road network space, and directly publish trajectory after replacing stop points without considering the possible position mutations. However, if the reconstructed trajectory passes through some obstacles or cause some position mutations evidently, an attacker can easily find such special trajectory segments which have been modified, and

then the sensitive positions on the trajectory would be exposed. Thus, our work further considers the effect of obstacles presented in such space region and the position mutations while reconstructing the trajectory.

Our main idea is shown in Figure 1, where the solid curve is an initial user trajectory and the three dots on this curve are the extracted stop points. To protect the privacy of this user trajectory, these stop points can be replaced by the appropriate POIs with same or similar semantic attribute category of stop points in their selection regions, denoted by the three stars. The dashed curve represents the trajectory after reasonable reconstruction. Publishing the reconstructed trajectory can effectively protect user privacy. Suppose malicious attackers could gain the published trajectory, the sensitive stop points have been replaced, so the re-identification and attribute-linkage attacks can be prevented. Our work considers a number of other factors that are not adequately addressed by pervious studies, such as semantic attributes, user-defined privacy level, user speed and position mutation.

Our main contributions are summarized as follows:

– Firstly, we achieve personalized privacy preservation with full consideration of semantic attributes. We label the semantic attributes of all sampling point among the trajectory database, and build a corresponding taxonomy tree used for choosing appropriate POIs for replacement, while considering the different user-defined privacy levels.
– Secondly, to achieve a good balance of privacy protection and data availability, for different types of stop points, we propose different strategies to select appropriate POIs for replacement. In addition, we devise a double half-circle area as the proper selection region of each stop point, for considering user speed and avoiding reverse mutation.

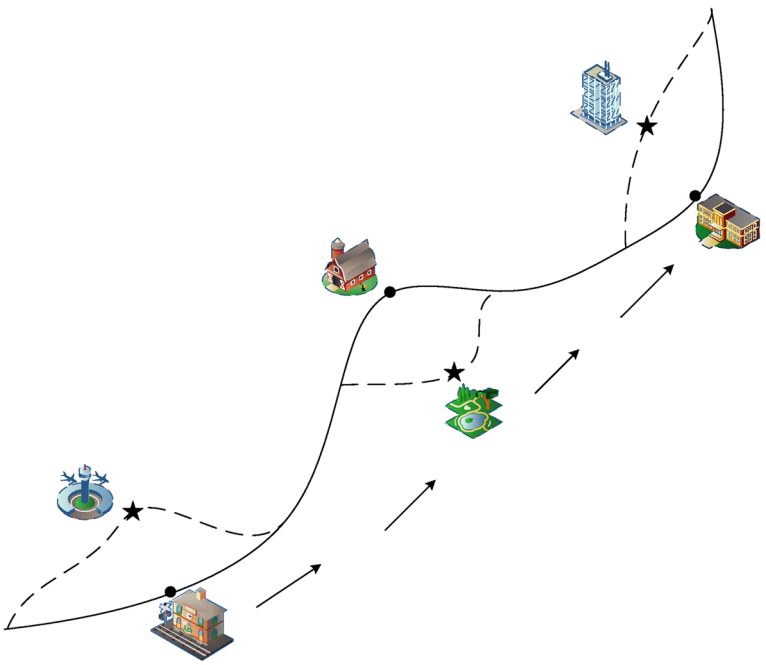


Figure 1 Example of a trajectory before and after replacing stop points

– Thirdly, during trajectory reconstruction for privacy protection, we further consider the effect of obstacles and position mutations, so that our method can reconstruct more realistic trajectories in line with actual situations.
– Finally, we propose effective evaluation criteria and conduct experiments on a trajectory database. The results show that our proposed method can improve the semantic consistency and shape similarity of the trajectory data as much as possible, and the reconstruction algorithm also can achieve the different privacy protection requirements while resisting to the attacks effectively.

The rest of the paper is organized as follows. In Section 2, we review related studies. We provide some notations and definitions in Section 3. Section 4 shows some assumptions and our system architecture. Section 5 introduces the attack models. In Section 6, we present our proposed method. Section 7 defines the evaluation criteria. The experimental study is reported in Sections 8 and 9 concludes with remarks on future work.

2 Related work

2.1 Categorization of existing techniques

To prevent personal privacy leakages, a large number of studies have been carried out for trajectory privacy preservation [2, 4, 14, 20]. Existing techniques can be roughly divided into four categories:

– The first category is *fake trajectory*. It means that the initial trajectory is published with several fake trajectories in order to confuse attackers. Note that however, the shape and semantic attributes of fake trajectories cannot deviate too much from original trajectory, because severe distortion may cause attackers infer users' true trajectory easily [24]. This method is simple but not very effective. There are three reasons. Firstly, the fake trajectory may pass through existing obstacles, and attackers can easily get rid of this obviously unreasonable trajectory. Secondly, the storage and computation of fake trajectories can cause a large expense. Thirdly, trajectory data availability is poor due to the published fake trajectories, and it will affect the quality of queries or applications based on these data.
– The second category is *the differential privacy*. Its main idea is to add noise to a database so that an adversary cannot decide weather a particular trajectory record is included in the database or not [5]. The first and mostly used method for achieving differential privacy is the Laplace mechanism. Although it provides provable guarantees independent of background knowledge obtained by an adversary and its computational power, it has some disadvantages. The added Laplace noise is unbounded and the variance of Laplace sampling is quite large due to the high sensitivity of trajectory publishing. Thus, the amount of noise to add could be too large to provide any information with good utility. A recent work in [21] tries to adopt a novel differentially private trajectory data publishing algorithm with a bounded noise generation algorithm and a trajectory merging algorithm.
– The third category is *trajectory k-anonymity*. It means to adapt the notion of k-anonymity to trajectory privacy preservation [1, 12, 27, 30]. This method can ensure that the published data are real, and also achieve a balance between privacy protection and data utility to some degree. In addition to directly releasing other $k - 1$

trajectories, other improved k-anonymity models are also proposed. Never Walk Alone (NWA) [17] introduces the concept of (k, δ)-anonymity for moving object database, where δ represents the possible location imprecision. In addition, two heuristics for trajectory anonymity are proposed in [3]: the first is based on trajectory micro-aggregation to achieve k-anonymity, while the second is based only on location permutation for location k-diversity considering readability constraints. For example, [29] also achieves a user trajectory k-anonymity based on the assistance of other historical trajectories, and the work in [31] defines the selection of trajectory k-anonymity set as graph partition problem and minimizes the partition cost according to the distances among trajectories so as to reduce information loss. Recently, there are some new considerations added to k-anonymity. For example, [23] focuses on providing a personalized service through a clustering trajectory preserving algorithm, [28] tries to consider the semantics of frequently-visited locations in the trajectory, and [13] aims to solve the problem of privacy-preserving publishing of spatiotemporal trajectories of mobile subscribers. However, almost all methods based on k-anonymity or its variants deal with the whole trajectory, and the generalization approach still does not well consider the semantic meanings of location points. In addition, trajectory k-anonymity will introduce noise and may lead to information loss.

– The fourth category is *selectively releasing trajectory*. It means to ignore those points with sensitive attributes or visited frequently in the trajectory publishing and only publish unsensitive sampling points [11, 15]. Another case is that once the user enters a sensitive area, location updates are suppressed at once. This method iteratively suppresses some trajectory segments or sensitive points until a probabilistic constraint of disclosing whole trajectories is satisfied, which seems simple and effective. However, it may cause a sudden change of trajectory and lead to severe data distortion. More importantly, as a result although the released trajectories do not contain the original sensitive information, it will cause serious distortion of the trajectory data, so the usability can be rapidly decreased leading to a very poor commercial value.

In addition, *encryption algorithm* can be used to protect trajectory privacy, such as the methods introduced in [6, 8, 22]. Although encryption can effectively protect trajectory privacy, the efficient query processing over encrypted trajectory data is a very challenging task.

2.2 Techniques protecting sensitive stop points

Protecting the whole user trajectory (those techniques belonging to the third category) usually leads to a large computational burden and a huge storage overhead. As an improvement of the fourth category, recently some researches [16, 18, 26] aim to protect sensitive stop points of a trajectory rather than the trajectory as a whole, because these stop points can reveal the purpose and meaning of a user trajectory more easily [25]. It is worth mentioning that the semantic attributes of these stop points can represent the requirements of trajectory protection to some degree as well. For example, if a stop point of the user trajectory locates at the relevant government department, it implies this user trajectory may contain sensitive political information, so it needs a higher degree of privacy protection.

To protect the stop points among trajectories, some researchers propose to coarsen the positions of stop points, such as [18]. A coarse zone is used to represent the position of a stop point. However, this method has some disadvantages. On the one hand, attackers can easily find the repeated moving objects of the coarse zones, so the re-identification attacks always

occur. On the another hand, fine granularity of protection also has a great influence on the leakage of information. Thus, another method is proposed to protect the stop points among trajectories in [26]. It replaces stop points with less sensitive POIs, and then reconstructs user trajectory. However, the sensitivity of sampling points and POIs is pre-defined as fix value, which may be not appropriate.

Instead simply choosing a less sensitive POI to replace corresponding stop point, in SST [16] a more reasonable method is introduced. It infers four privacy risk levels of stop points based on stop points' visiting status and the semantic place distribution in its neighboring region, and adapts different modification methods to replace corresponding stop points. Although this method allows personalized privacy requirements and takes environmental conditions such as speed into consideration (which are two advantages shared by our method), we further propose to choose POIs more reasonably through a built taxonomy tree. In addition, our trajectory reconstruction is more practical through dealing with the possible obstacles and position mutations.

2.3 Techniques considering semantic and other factors

Besides spatial-temporal attributes, the semantic meanings of positions become an emerging domain in the trajectory privacy protection. The first category of semantic works is applied to package real geographic coordinates to meet l-diversity principle. You Can Walk Alone (YCWA) [18] is a typical trajectory privacy preserving method considering the semantic attributes of sensitive stop points. Different from the usual cloaking methods, although the published trajectory consists of a series of cloaking regions, each region always needs to include at least l different types of semantic places. Although this can effectively ensure semantic attributes of stop points not to be disclosed, as the value of l grows, each region contains more and more semantic information. Releasing such trajectory data for analysis cannot guarantee the semantic consistency of trajectory very well, and the data availability could be very poor. The second category of semantic works pays attention to sensitive attribute generalization, which means to replace original semantic attributes with generalized semantic attributes. PPTD [19] is a representative work to decide the minimum amount of necessary generalization of each point. The key idea is "do not say something too specific". For example, a simple trajectory segment such as "WalMart → the People's Park" can be generalized to "Supermarket → park" instead. This generalization is done such that a desirable balance between information loss and privacy disclosure is achieved, but its process is more complex. Our method follows the idea of PPTD, but is more realistic. We randomly select an appropriate POI from the candidate POI set with the same or similar semantic attribute category of corresponding stop point. In addition, the selected POI is the real geographic coordinate and the category is a generalization depending on user-defined privacy level and the built taxonomy tree.

There are other algorithms that consider more factors, such as user speed, position mutation, moving direction of trajectory [10, 16]. The proposed method fully considers these related factors for more effective trajectory privacy preservation.

3 Notations and definitions

The definitions and notations we use throughout the following sections are defined as follows.

3.1 Trajectory database

A user trajectory consisting of a sequence of sampling points usually corresponds to a specific moving object, and the so-called *trajectory database* is a static set of all mobile users' trajectories within a certain time interval and area range. In our work, we focus on the context of trajectory database, so the privacy preservation is usually off-line and data-centric.

Definition 1 (Trajectory Database) Let O represent a set of moving objects denoted as $O = \{O_1, O_2, \cdots, O_n\}$, while T represents the obtained trajectory database and $T = \{T_1, T_2, \cdots, T_n\}$. A user trajectory in T denoted by T_i can be described as $T_i = \{id, (x_1, y_1, t_1), (x_2, y_2, t_2), \cdots, (x_n, y_n, t_n)\}$. id is a unique identifier of the trajectory, $(x_i, y_i, t_i)(1 \leq i \leq n)$ is a sampling point in the trajectory which is called as moving point, denoted by m_i, in which $l_i = (x_i, y_i)$ is the sampling position of this moving point, and t_i is the sampling timestamp.

Trajectory sequence can be defined in an ascending order by timestamps $\{t_1, t_2, \cdots, t_n\}$ [9]. In this paper, for simplicity we suppose each moving object O_i corresponds to only one trajectory record T_i among the whole trajectory database. Therefore, we introduce a function, denoted as $x : (T \rightarrow O)$, to assign only one trajectory record to a specific moving object. For each moving object, its trajectory data record is a sequence of moving points within a certain sampling interval. If the object has stopped, then the collected trajectory data are static. In contrast, if the object is moving, then the collected data are dynamic. As for trajectory publishing scenario, an off-line and static trajectory database is our main concern.

3.2 Semantic attribute

Besides the spatial and temporal attributes, we pay more attentions to the semantic attributes of trajectory data. For different moving objects, some semantic attributes are sensitive that cannot be leaked, while some are not sensitive that can be published directly. For example, if the moving objects are some patients, the health problems are their primary considerations, and the sensitive semantic attributes might be their health issues. The semantic attributes of different moving objects can de divided into two types: *sensitive semantic attributes* and *unsensitive semantic attributes*. Therefore, the trajectory database for different moving objects should be expressed as $T_i' =< id, (x_1, y_1, t_1), (x_2, y_2, t_2), \cdots, (x_n, y_n, t_n) >: s_1, s_2, \cdots, s_k : u_1, u_2, \cdots, u_m$. To be specific, $< id, (x_1, y_1, t_1), (x_2, y_2, t_2), \cdots, (x_n, y_n, t_n) >$ denotes the user trajectory record, $\{s_1, s_2, \cdots, s_k\}$ denotes sensitive semantic attributes of the user while $\{u_1, u_2, \cdots, u_m\}$ denotes unsensitive semantic attributes. Moreover, each moving object could define a semantic attribute is sensitive or not by itself. We adopt a set $S = \{s_1, s_2, \cdots, s_k\}$ to express all its sensitive semantic attributes. The elements of S entirely depend on the user's settings. For example, the user can set his own illness and income as sensitive attributes, but political status seems unsensitive to him, then the set of his sensitive semantic attributes could be denoted by $S = \{illness, income\}$, and his unsensitive semantic attributes could be denoted by $U = \{political\text{-}status\}$. In generally, the elements of S are usually some conceptual and high-level semantic attributes and can be divided into some more concrete and specific semantic attributes, which will be used subsequently.

Another definition about semantic attributes of POIs is given as follows. Suppose the set of POIs in the user space environment is denoted as $POIs = \{POI_1, POI_2, \cdots, POI_n\}$, and the semantic information is stored in a table called POI attribute table, abbreviated as PAT [16]. An example of PAT is shown as Table 1.

In this work, we assume each tagged position coordinate of POIs corresponds to only one semantic attribute. The element of the table is a triplet denoted as $POI_i = (x_i, y_i, a_i)$, where (x_i, y_i) is the position of POI_i and a_i is the corresponding semantic attribute. To be simple, for all moving points of trajectory database T, we simply associate its semantic attribute with the nearest POI_i and assign the semantic attribute a_i to the nearest moving point m_i. In this way, each moving points is related to one and only one corresponding semantic attribute. Therefore, we get a trajectory database with assigned semantic attributes, and a user trajectory is defined as $T_i' = \{id, (x_1, y_1, t_1, a_1), (x_2, y_2, t_2, a_2), \cdots, (x_n, y_n, t_n, a_n)\}$. The detailed labeling operation will be discussed later.

3.3 Taxonomy tree

To illustrate this concept, Figure 2 shows an example of the semantic attributes of patients, as the user sets his sensitive semantic set $S = \{illness\}$. The whole tree represents a specific classification for the sensitive semantic attribute *illness*, where leaf nodes represent the most common disease symptoms, and internal nodes represent generalized classes of some similar symptoms. Thus, the upper node closer to the root node *illness* indicts the generalization of more types of disease symptoms. For example, *mental disease* is a generalization of {*melancholia*, *autism*, *acrophobia*}.

Definition 2 (Taxonomy Tree) For the obtained static trajectory database, the set of all semantic attributes associated with all moving points is denoted by $A = a_1, a_2, \cdots, a_n$. Meanwhile, according to semantic attribute set A of the trajectory database, we build a taxonomy tree represented by $G = (V, E, l)$. V is a set of nodes whose values represent categories of basic semantic attributes in different grades, E is a set of edges between two nodes which represent a relationship between two nodes, and l is a labeling function to assign some semantic attributes of A to each node v_i in the node set V.

In general, as [19] indicates the taxonomy tree contains two types of nodes: *leaf nodes* and *internal nodes*. Specifically, our taxonomy tree is built through semantic attribute set A of the trajectory database such that all moving points' semantic attributes in A are regarded as leaf nodes. Moreover, the internal nodes have been uniquely labeled with a name to show the same semantic category of nodes of the lower layer in the sub-trees. Like the definitions of a common tree, we define the upper nodes as *parent nodes* of the lower layer nodes and use an edge E_i in E to express such relationship while defining these lower nodes as *child*

Table 1 POI attribute information

X	Y	Attribute
4421773.469888	697984.726873	A
4433640.998030	697213.359364	B
4429471.250742	698865.254019	C
...	...	...

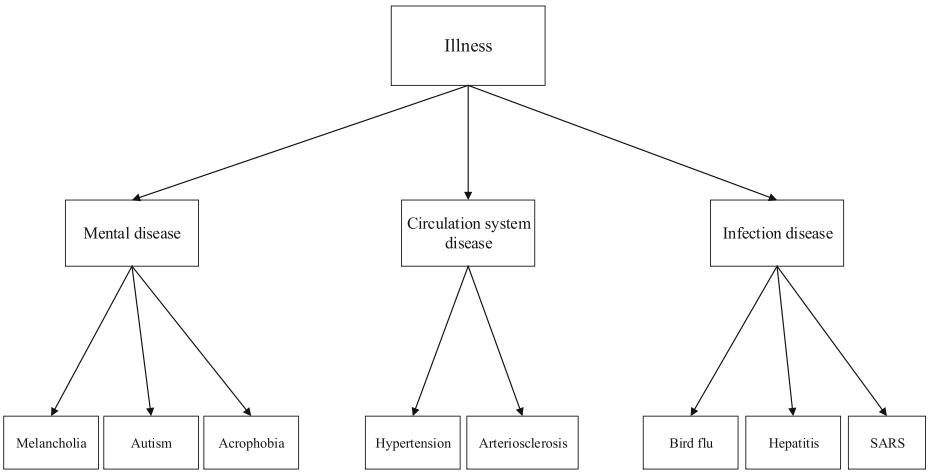


Figure 2 A taxonomy tree for patients

nodes of the upper node. We also call the adjacent nodes without an edge as a node's *sibling nodes*, if and only if they have the same parent node, and the summit node of a taxonomy tree is called the *root node*. We also should pay attention to the height of a taxonomy tree. We define the depth of all leaf nodes as level 0, and the depth of the upper nodes increases recursively until the root node. The height of each node is denoted by $h(v_i)$, the height of the root node is denoted as rh, so the set of heights for different layers of a taxonomy tree is $H = \{H_0, H_1, \cdots, H_n\}$, or actually we can say $H = \{0, 1, \cdots, rh\}$. Obviously, for each node, it has the same height as its sibling node while the height of its parent node increases by 1.

As shown in Figure 2, the level of leaf node is 0, e.g., $h(SARS) = 0$, while levels of internal nodes are increased, e.g., $h(\textit{circulation system disease}) = 1$ and $h(\textit{illness}) = 2$. For each v_i, the labeling function $l(v_i)$ is used to assign a subset of semantic attributes A to the node. For example, $l(\textit{autism}) = \{\textit{autism}\}$ while $l(\textit{mental disease}) = \{\textit{melancholia, autism, acrophobia}\}$ and $l(\textit{illness}) = \{\textit{melancholia, autism, acrophobia, hypertension, arteriosclerosis, bird_flu, hepatitis, SARS}\}$.

Definition 3 (Privacy Level) Privacy level is defined by users themselves. Let $P = \{no, p_0, p_1, \cdots, p_n\}$ represent a set of privacy levels users can choose form, where p_i is each user privacy level. It should be noted that *no* represents the user do not need any privacy protection.

Different users have different requirements of trajectory privacy protection, so our work allows each user to define his privacy levels to represent his own personal need. We introduce a function $y : (P \rightarrow O)$ to assign a privacy level p_i to a specific moving object O_i. As mentioned, each user corresponds to one trajectory data record of the trajectory database T, so a trajectory T_i naturally corresponds to a specific privacy level p_i. In addition, the user-defined privacy level is set in terms of the entire user trajectory database, which is important for the following discussion. However, it is worth noting that the taxonomy tree is built according to the semantic attributes of moving points among trajectory database, and the moving points are on the user trajectories, so we can map the user-defined privacy level to

the height of our taxonomy tree and define a mapping function $z : (P \rightarrow H)$ to assign the height of the taxonomy tree to a privacy level p_i. The height of the root node rh represents the user's highest privacy level p_n, denoted as $z(p_n) = rh$ and n equals to the layers of the taxonomy tree. Normally, we can assume $P = \{no, 0, 1, \cdots, rh\}$ is the set of privacy levels that users can choose form. Obviously, the higher the defined privacy level is, the higher the privacy protection of the user's trajectory should meet.

4 Assumptions and system architecture

4.1 Assumptions

As mentioned above, our work is on trajectory privacy preservation for off-line trajectory publishing. There are some basic assumptions about background knowledge in this scenario.

Firstly, we assume the location-based service providers and the organizations who have direct access to users' original trajectory database can be trusted, and the privacy protection is also completed by these trust mechanisms. The obtained trajectory database is static and off-line which consists of a series of user trajectories that need to be protected before publishing to third parties. Moreover, these trusted mechanisms can also acquire the related background knowledge of user space environment. The so-called space environment includes the distribution of obstacles (such as mountains, rivers, etc.) and the distribution of the POIs together with semantic attributes. In this paper, we also use a table (PAT) to store the geographical positions and semantic attributes of the set of POIs, and the table would be used to label the semantic attributes of all moving points among the trajectory database later. However, we assume each moving point among the trajectory database can only match to one semantic attribute to make our discussion simpler.

Secondly, for the moving objects, we assume that the different users have defined a set of semantic attributes S which are sensitive for them and a privacy level p_i for the trajectory privacy protection. In addition, each moving object can only match one trajectory in the obtained trajectory database.

Thirdly, attacks occur after publishing the protected trajectory database while the data transmission channel is secure. We assume that malicious attackers not only can obtain users' protected trajectory data published by the trusted mechanisms and the related user space environment, but also know the obtained user trajectory data may be changed, even the specific process of the trajectory privacy preservation method would be exposed to them. In our settings, the biggest barrier for attackers is the non-repeatability and randomness during the process of selecting an appropriate POI for replacement. On the other hand, in order to avoid the position mutations on the trajectory, some unsensitive moving points will also be replaced in the process of trajectory reconstruction, and the replacements of these moving points will also become a resistance for the attackers.

4.2 System architecture

In this section, we introduce the main steps of our proposed algorithm. Figure 3 is the system architecture of our proposed trajectory privacy preservation. Its inputs are the original trajectory database, the distribution and semantic attribute table of POIs, the user-defined sensitive semantic attribute set and its privacy level, and the distribution of obstacles is also included. The output is only the protected trajectory database which can maintain the semantic consistency and shape similarity with the original trajectory database while

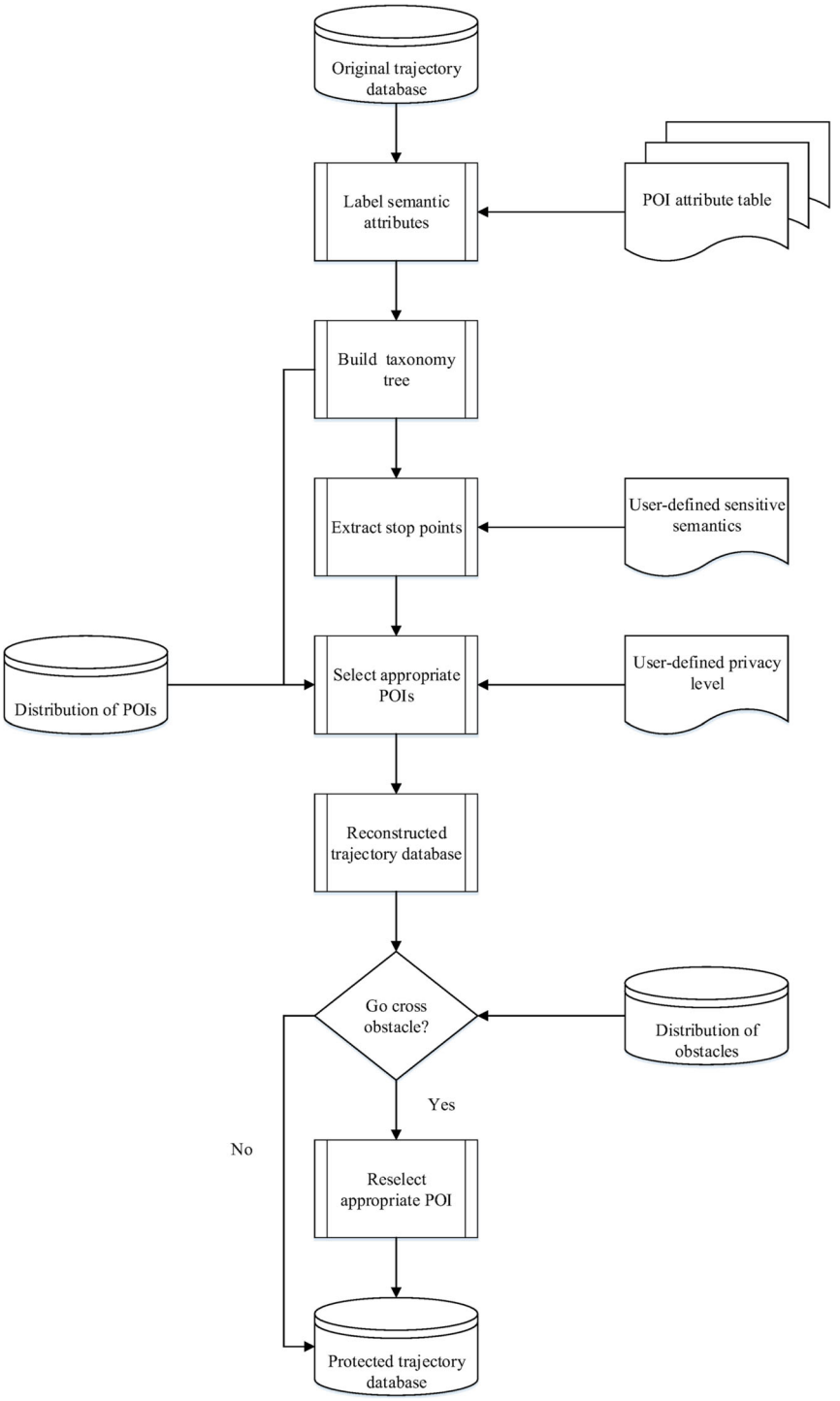


Figure 3 Architecture of the proposed trajectory privacy preserving system

meeting the different privacy requirements of users as much as possible, so that the published trajectory database can produce a good commercial value. Detailed steps are as follows.

First, according to the given trajectory database and the POI attribute table (PAT), we label the semantic attributes of all moving points by matching each moving point to its nearest POI, and assume it has the same semantic attribute with the corresponding POI. Next, we abstract and generalize the semantic attributes of all moving points to build our taxonomy tree.

After this, as our aim is to protect the stop points among trajectory, we extract the three kinds of stop points from all moving points [18], namely long-stayed points, wandering points and sensitive semantic points according to the user trajectories' time attribute, spatial attribute, and the user-defined sensitive semantic attributes. In addition, we divide these stop points into three types, namely *non-isolated stop point*, *isolated stop point* and *quite-isolated stop point*, according to the distribution of each stop point's neighboring positions.

Then, based on each extracted stop point, we determine its proper selection region (PSR) as a double half-circle area considering the user speed and reverse mutation, and adopt different strategies to select an appropriate POI for replacement. It means to randomly select a POI belongs to the same or similar semantic category of the stop point, to satisfy the different privacy protection requirements of users through the built taxonomy tree and user-defined privacy level.

Finally, it turns to reconstruct the user trajectory. The first step of this process is to replace all sensitive stop points with the selected POIs. In order to avoid position mutations, we need to reasonably replace some of unsensitive moving points at the same time. More importantly, we also need to check whether the reconstructed trajectory segment has crossed through obstacles. If yes, we will reselect another appropriate POI until the reconstructed trajectory does not cross through any obstacle. In this way, we publish the protected trajectory database in line with actual situations.

5 Attack models

Based on the relevant background knowledge that an attacker can obtain, the main goal of the attacker is to identify the user trajectory associated with the true identity or infer its corresponding sensitive information. We also assume that each trajectory has already been anonymized by replacing the true trajectory identifier with a random pseudonym. After this pre-protection step, threatens of trajectory privacy leakage still exist following two attacks:

– **Re-identification attack**: Based on the attacker's background knowledge, the trajectory is unique in the user's database, so attackers can easily identify the victim's trajectory. Such special side background knowledge, e.g., a causal talk eavesdropped by adversary, may expose its whereabouts totally. Suppose through obtaining some side information, the attackers know a specific user id would be at location l_i at time t_i, and meanwhile (l_i, t_i) happens to be a moving point on a trajectory in the published trajectory database. If this trajectory is the only one containing this moving point or sub-trajectory, obviously the attackers can re-identify the whole trajectory of the user id, and his other sensitive information as well. For this situation, attackers can match the obtained moving points or sub-trajectory to a specific trajectory and re-identify the user's id.

– **Attribute-linkage attack**: If there are more than one matched trajectories among the trajectory database through analysing specific sampling points and some side information, attackers cannot uniquely identify the trajectory record of the target victim. However for this situation, as discussed in [19], if some sensitive semantic attributes occur frequently on these matched trajectories, even though the record of the victim cannot be uniquely identified, the attackers can infer the probability that the victim has the sensitive semantic attribute. For example, a trajectory database is about medical records, and through some specific sampling points it can be inferred that there exist three trajectories belonging to a target user (e.g., Bill). Two of the three trajectories include a semantic attribute (e.g., SARS), so the attackers can infer that the probability the Bill has SARS is about 67%. Moreover, there is another situation. If the semantic attributes of the three trajectory records are *bird flu*, *hepatitis* and *SARS* respectively, the attackers then can infer Bill has infection disease with 100% confidence, because all the three semantic attributes are different types of the upper semantic attribute *infection disease*. It seems that although malicious attackers cannot identify the complete user trajectory associated with the true identity, they can associate the user with some sensitive semantic attributes.

In summary, although attackers can obtain a lot of side information about user's whereabouts through many ways, the obtained trajectories have already been changed. It is to say, the sensitive stop points on the trajectory obtained by attackers have been replaced by an appropriate POI with the same or similar semantic category before publishing. If attackers match the replaced trajectory sequence and its side information to re-identify the victim or to infer the sensitive value of the victim, such re-identification or attribute-linkage attacks will not cause too much damage.

6 The algorithm

Our trajectory privacy preserving algorithm contains four steps: labeling semantic attributes of moving points and building a taxonomy tree, extracting stop points among user trajectory from semantic and spatial-temporal attributes, selecting the appropriate POI for replacement according to user-defined privacy level, and reconstructing a trajectory with high data availability. This section describes the algorithm in detail.

6.1 Labeling semantic attributes and building a taxonomy tree

In our work, a trajectory database is static, so we can obtain all moving points of trajectories in advance, and the background knowledge includes a determined user space environment containing a set of all points of interest $POIs$ associated with a corresponding POI attribute table (PAT). While the semantic attributes of all POIs are available from the PAT, an urgent task is to label all moving points with semantic attributes. Then, we build a taxonomy tree by using the semantic attributes assigned to all the moving points among the user trajectory database.

First, for a specific moving point $m_i = (x_i, y_i, t_i)$, we calculate the Euclidean distance between it and its nearby $POI_j = (x_j, y_j, a_j)$, denoted by $\overrightarrow{m_i, POI_j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. We assume the nearest POI is the one whose Euclidean distance is shortest. Then, we simply assign the semantic attribute a_j of the nearest POI

POI_j to the corresponding moving point m_i, so $m_i' = (x_i, y_i, t_i, a_i)$ while a_i equals the value of a_j of POI_j. The rationale is that the closer the position of the moving point next to the POI is, the more similar the semantic attribute is. For example, the position of a moving point is much closer to a crossroad, while the corresponding semantic attribute of the crossroad is "congestion", then we can label the semantic attribute of the moving point as "congestion". In our work, we match each moving point to only one appropriate semantic attribute and obtain the set of all semantic attributes associated with all moving points denoted as $A = \{a_1, a_2, \cdots, a_n\}$, in which n represents the number of different semantic attributes of all moving points. Then, through the analysis, we can make these semantic attributes abstracted and generalized, and form a taxonomy tree recursively. Specifically, all elements in the set A are regarded as the leaf nodes of the taxonomy tree, then we divide these leaf nodes into different categories based on the semantic meanings, and the value of each abstracted and generalized category is used as the parent node of these leaf nodes which belong to the same generalized semantic attribute category. We continue to abstract and generalize these categories of leaf nodes with semantic similarity to form the upper nodes. This step is iterated until a root node is finally abstracted and generalized. In this way, a taxonomy tree $G = (V, E, l)$ mentioned above would be built. As Figure 2 shows, we can generalize the leaf nodes {*bird flu, hepatitis, SARS*}, and the internal node *infection disease* is obtained. Then, we generalize the internal nodes {*mental disease, circlation system disease, infection disease*} to obtain the root node *illness*. From the tree, we can see the semantic attributes *melancholia*, *autism*, *acrophobia*, *hypertension*, *arteriosclerosis*, *bird flu*, *hepatitis* and *SARS* all belong to a same semantic attribute category *illness*. Thus, after mapping the user-defined privacy level to the height of proper layer of the taxonomy tree and matching the stop point to the proper leaf node, we can determine the internal node *Internal_Candidate* and the set *Semantic_Candidate* corresponding to the stop point from the taxonomy tree, so as *Smiliar_Internal_Candidate* and *Smiliar_Semantic_Candidate*.

6.2 Extracting stop points

After labeling the semantic attributes of all moving points among trajectory database and building a taxonomy tree, we determine what kinds of moving points need to be protected most in terms of semantic and spatial-temporal attributes. The privacy preserving algorithm needs to pick up all stop points which need replacements. Hence, the corresponding long-stayed points, wandering points and sensitive points are identified as stop points. We adopt different methods to extract stop points of different types. In order to achieve the user-defined privacy level, the following part discusses how to extract stop points from a trajectory data record T_i corresponding to one specific moving object O_i. To make it clear, the background knowledge includes a taxonomy tree $G = (V, E, l)$ corresponding to the trajectory database T, a sensitive semantic attribute set S given by the specific moving object O_i, and we assume all moving points have appropriate semantic attributes.

For the first category, the long-stayed points considering time attribute, we adopt a duration-based method. For this kind of stop points, we consider two possible situations: one is the moving objects intentionally stay at one position for a long time (e.g., at user's home), and another is that the GPS device loses signals or is just turned off (e.g., in the buildings shielding signals). As it is common that if a person equipped with a GPS device gets into a building then the GPS device might lose signals and stop recording, or perhaps a driver stops his car, the onboard GPS device would also turn off. Thus, these positions contain more user sensitive information that need to be regarded as stop points

and need to be protected. In simple terms, these stop points are the moving points which have a larger time interval with their follow-up points but its sampling position remains unchanged. To extract them, a time threshold parameter th_{time} is introduced. If the time duration which a moving object stays at a certain position exceeds the given threshold th_{time}, all the moving points among this time interval are regarded as stop points. That is to say, for all moving points in a specific user trajectory, $T_i = \{(l_1, t_1), (l_2, t_2), \cdots, (l_n, t_n)\}$, if $|t_j - t_i| > th_{time}$ *and* $l_i = l_k = l_j (i < k < j)$, then l_i can be determined to be a sensitive position and $m_k (i \leq k \leq j)$ are all marked as stop points. In other words, such a stop point's position remain unchanged while time interval exceeds the threshold th_{time}.

After extracting long-stayed points, we continue to extract the second category of stop points, wandering points considering spatial attribute. We consider such a situation that within the scope of a longer period of time, if moving objects frequently access a position or just wander in a very small space area, then this small space region can also disclose user's privacy information through the analysis by attacker. For example, a person with a GPS device is wandering around a landmark, but the GPS neither loses signals nor turns off. These points are the wandering points, which also need to be marked as stop points. Under this situation, we introduce a distance threshold th_{dist} which is also companied with a time threshold parameter th_{time} defined previously. For some moving points among the user trajectory, if the Euclidean distance between any moving point pair is less than the given distance threshold th_{dist} while their time duration is larger than th_{time}, i.e., for some moving points denoted by $m_k (i \leq k \leq j)$, among the user trajectory $T_i = \{(l_1, t_1), (l_2, t_2), \cdots, (l_n, t_n)\}$, if $max\{\overline{m_k, m_l}\} < th_{dist} (i \leq k, l \leq j)$ while $|t_j - t_i| > th_{time}$, we can regard the spatial region formed by these moving points is sensitive and these points $m_k (i \leq k \leq j)$ should be marked as stop points. In other words, such stop points are those which are usually frequently accessed in a small space range while time interval exceeds the time threshold.

For the third category, the sensitive points considering semantic attribute, we adopt a matching-based method. Based on the background knowledge mentioned above, a moving point is identified as a sensitive point if it meets the following conditions: we first match the user-defined sensitive semantic attribute set S to the taxonomy tree, i.e., matching the user-defined set $S = \{s_1, s_2, \cdots, s_n\}$ to the node set $SN = \{node_1, node_2, \cdots, node_n\}$ while s_i equals to the value of $node_i$ $(1 \leq i \leq n)$. Then, we can use the labeling function l to find out all leaf nodes corresponding to the node set SN, denoted by $SLN = \{l(node_1), l(node_2), \cdots, l(node_n)\}$. We should understand leaf nodes are just the semantic attributes of the moving points, so the moving points are associated with these leaf nodes containing the user's sensitive information, which requires protection. To be simple, for the moving point $m_i = (l_i, t_i, a_i)$ if the semantic attribute $a_i \in SLN$, it is regarded as a stop point. It is not difficult to understand that these sensitive points' semantic attributes do not reach the user-defined privacy level, so they need to be labeled as stop points.

Related pseudo code is shown in Algorithm 1. The inputs of the algorithm include all m moving points of the original user trajectory, a time threshold th_{time}, a distance threshold th_{dist}, a user-defined sensitive attribute set S and the built taxonomy tree $G = (V, E, l)$. The output is a set of stop points $Stop$ including three types of stop points. Lines 2-16 of Algorithm 1 adopt a duration-based method to extract the wandering points and long-stayed points, lines 17-48 are the process to map the sensitive attribute set S with the taxonomy

tree $G = (V, E, l)$ and get the sensitive semantic attributes SLN, then mark the sensitive stop points in lines 39-49.

Algorithm 1 Extracting stop points

Input: All m moving points of the original user trajectory, time threshold th_{time}, distance threshold th_{dist}, sensitive attribute set S, the taxonomy tree $G = (V, E, l)$
Output: The set of stop points $Stop$

```
1: i ← 1;
2: while i < SizeOf(m) and m[i] ∉ Stop do
3:     j ← i + 1;
4:     while j < SizeOf(m) do
5:         if (m[j][l] − m[i][l]) < th_dist then
6:             Δ_time ← m[j][t] − m[i][t];
7:             if Δ_time > th_time then
8:                 Stop ← AllPointsOf(m[i] → m[j]);
9:             end if
10:        end if
11:        j ← j + 1;
12:    end while
13:    i ← i + 1;
14: end while
15: j ← 1;
16: k ← 1;
17: i ← 1;
18: while j < SizeOf(S) do
19:    if S[j] == V(i) and V(i) ∈ G then
20:        SN[k] ← V[i];
21:        k ← k + 1;
22:    end if
23:    j ← j + 1;
24: end while
25: j ← 1;
26: i ← 1;
27: while j < SizeOf(SN) do
28:    sln = l(SN(j))
29:    k ← 1;
30:    while k < SizeOf(sln) do
31:        SLN[i] ← sln[k];
32:        k ← k + 1;
33:        i ← i + 1;
34:    end while
35:    j ← j + 1;
36: end while
37: i ← 1;
38: k ← SizeOf(Stop) + 1;
39: while i < SizeOf(m) and m[i] ∉ Stop do
40:    j ← 1;
41:    while j < SizeOf(SLN) do
42:        if m[i][a] == SLN[j] then
43:            Stop[k] ← m[i];
44:            k ← k + 1;
45:        end if
46:        j ← j + 1;
47:    end while
48:    i ← i + 1;
49: end while
```

6.3 Selecting appropriate POI

After labeling each stop point, the next task is to appropriately select a corresponding POI based on the user-defined privacy level P_i and the given set $POIs$ in the user space environment. In this part, we are most concerned about two issues: the space area from which we select POI, and how to select an appropriate POI. Therefore, we divide the POI selection process into two steps: first, we define a proper selection region of each stop point; then, we discuss how to randomly select an appropriate POI for replacement. According to the different stop points' neighboring positions, all stop points could be divided into three types, so we adopt three different strategies to deal with these cases.

6.3.1 Defining proper selection region of stop point

We define a proper selection region through considering user speed and avoiding reverse mutation. Existing studies related to the selection region searching can be divided into the following two categories [26]:

- The first kind is to search among the area determined by the entire user trajectory, which means to randomly select a POI with the same semantic category to replace the corresponding stop point from all the POIs. However, this large selection region will cause a very high computation burden to calculate candidate POI set *POI_Candidate*, and the large value of $|POI\_Candidate|$ will also increase the difficulty of selecting an appropriate POI, and the trajectory data availability would always be very poor as the selected POI might have a great shape deviation from original one. Even worse, this method does not consider user speed and reverse mutation factors at all.
- The second kind is to divide the user trajectory into different trajectory segments according to stop points. It means to use two adjacent stop points to form a trajectory segment in order to narrow down the area of the selection region. Then, it iteratively searches an appropriate POI in the selection region determined by each trajectory segment to replace the corresponding stop point. It seems easy for such method to calculate *POI_Candidate* and a simpler operation to select a POI *POI_selected* from the candidate POI set. However, if a trajectory segment determined by two stop points have a long distance, then it will also cause a high computation burden to calculate *POI_Candidate* and choose *POI_selected*, and in this case the selected POI might deviate too far from the corresponding stop point, which can lead to a poor data availability. In fact, the distance of two stop points is usually long, and we should pay particular attention to the user speed and reverse mutation occurred.

In our work, we propose another different searching method which is based on the stop point itself, and meanwhile considers the user speed and reverse mutation factors. The calculation and operation of this searching method are simpler, as the searching region would be smaller and more reasonable, so that *POI_selected* can be easily selected, and will not deviate too far from the corresponding stop point, and thus the protected trajectory can gain a higher trajectory data availability in terms of trajectory shape. In addition, the selected POI belongs to the same or similar semantic category of the stop point, so the semantic of protected trajectory would also maintain a high consistence. Besides, the selection process of

POI_selected can effectively meet the different user privacy requirements, so it can achieve a good balance between data availability and privacy protection. Detailed discussion on how to determine a proper selection region is given as follows.

As for the first factor, user speed, for a specific user trajectory, the moving object's velocity at different sampling positions usually varies. Because the sampling time interval of the trajectory sequence is consistent, the distance between the sampling points could be used to indicate the speed of the moving object, which is a proportional relationship. Moreover, from practical experience the high speed of a moving object suggests the moving object might in an underpopulated countryside. On the contrary, if the moving object is in a busy town street, its speed would be slow. Therefore, the selection region of the corresponding stop point seems to be larger in an open area, i.e., it is proportional to the speed of the moving object at the position. From this point of view, to take the user speed into account we use the distances formed by the stop point and its previous and next stop points on the trajectory to construct the proper selection region. Another factor to be considered is the trajectory reverse mutation, which is caused by the overlapping area of the selection region determined by the stop point and the selection region determined by its neighbor stop point. As choosing *POI_Candidate* from the overlapping area may cause a reverse mutation on trajectory with large possibility, such reverse mutation could be used by attackers to infer user sensitive information, which is known as reverse mutation attack. As Figure 4 shows, the selection regions determined by $stop_2$ and $stop_3$ are overlapped. If we choose POI_2

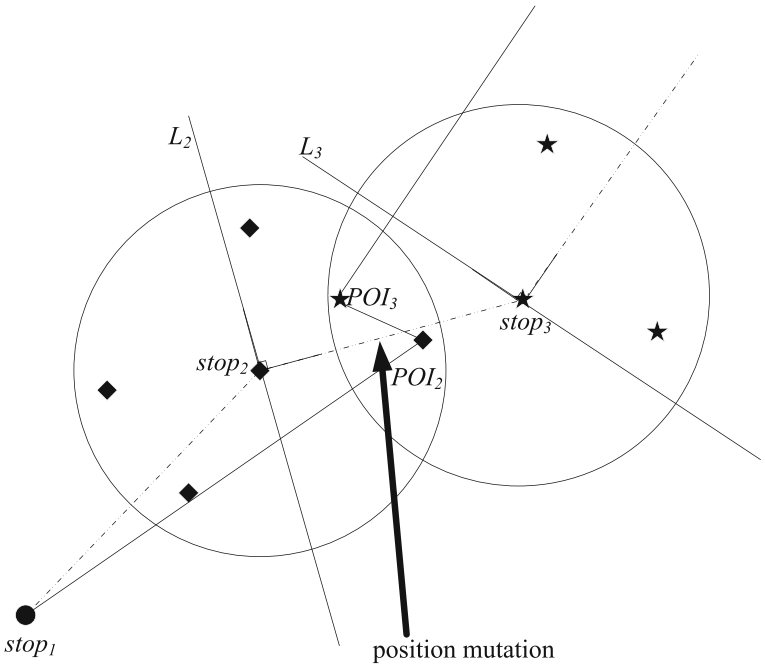


Figure 4 Mutation attack caused by overlapped selection regions

and POI_3 from the overlapping area as their *POI_selected*, and replace the two stop points respectively, as the figure shows, it would cause a reverse mutation attack.

Therefore, it is necessary to avoid the appearance of overlapping regions when determining the proper selection region. We follow [16] to use two asymmetric semi-circles to form a proper selection region (PSR) in the following discussion. Each corresponding PSR is composed of two semi-circles of different radii. First, we define the center of each semi-circle as the stop point. One radius of semi-circle is half-distance of the stop point and its previous stop point while the other radius is the half-distance of the stop point and its next stop point. Detailed region construction is discussed in the following.

As Figure 5 shows, for a specific stop point $stop_i$, its previous stop point is denoted as $stop_{i-1}$ and its next stop point is $stop_{i+1}$. We first connect $stop_i$ and $stop_{i-1}$ by a straight line to form a vector $\overrightarrow{stop_{i-1}stop_i}$, and form a vector $\overrightarrow{stop_i stop_{i+1}}$ in the same way. We regard the direction of the vector $\overrightarrow{stop_i stop_{i+1}}$ to be the velocity direction of the moving object at the sampling position, then we draw a line L_i perpendicular to the velocity direction at $stop_i$, and this vertical line L_i will serve as the dividing line between the two semi-circles. Next, in order to ensure that the adjacent PSRs do not overlap, the semi-circle on the left side of L_i regards $stop_i$ as center and $\frac{\overrightarrow{stop_{i-1}stop_i}}{2}$ as its radius. Likewise, the semi-circle on the right side of L_i regards $stop_i$ as center and $\frac{\overrightarrow{stop_{i+1}stop_i}}{2}$ as its radius. As a result, we successfully construct the PSR corresponding to $stop_i$ and ensure the selection regions of two adjacent stop points are not overlapped. It is worth mentioning that for the first stop point $stop_1$ of the user trajectory, its PSR is a circle centered at $stop_1$ and its radius is $\frac{\overrightarrow{stop_1 stop_2}}{2}$ while for the last stop point $stop_n$ of the user trajectory, its PSR is a circle centered at $stop_n$ and its

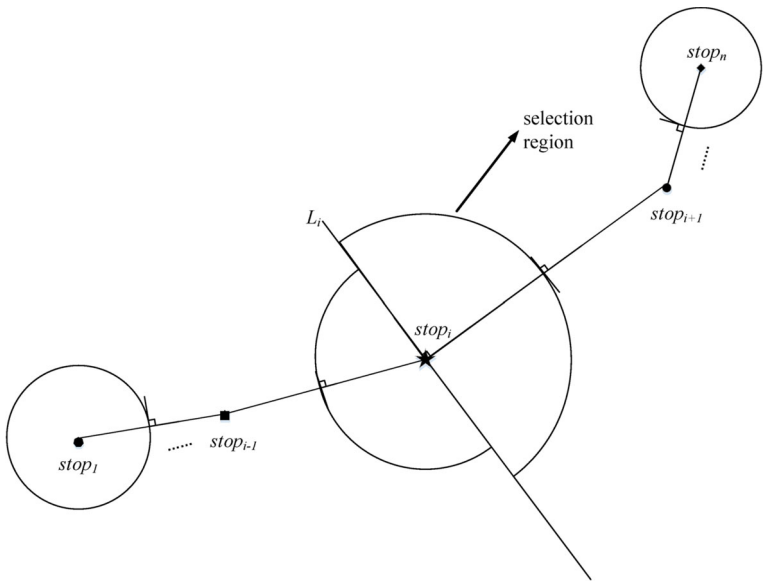


Figure 5 Example of forming a proper selection region

radius is $\frac{\overrightarrow{stop_{n-1}stop_n}}{2}$. After determining the PSR of each stop point, it is time to randomly select an appropriate POI from the PSR to replace itself.

Related pseudo code is shown in Algorithm 2. The inputs of the algorithm are the set of stop point $Stop$. The output is a set of PSR_i corresponding to each stop point. Lines 3–10 get the related center, radius and vertical line for constructing two asymmetric semicircles. Lines 11–14 handle the first and the last stop points, and we form their PSRs in the form a circle. Lines 15–17 construct PSRs for other stop points as component asymmetric semicircles.

Algorithm 2 Defining a proper selection region of the stop point

Input: The set of stop points $Stop$
Output: The proper selection region PSR_i
1: $i \leftarrow 1$;
2: **while** $1 \leq i \leq SizeOf(Stop)$ **do**
3: $\xi_1 \leftarrow \xi[Stop[i-1] : Stop[i]]$;
4: $\xi_2 \leftarrow \xi[Stop[i] : Stop[i+1]]$;
5: $A \leftarrow MidPointOf(\xi_1)$;
6: $B \leftarrow MidPointOf(\xi_2)$;
7: draw L_i at $Stop[i]$ making that L_i perpendicular to ξ_2;
8: $Center \leftarrow Stop[i]$;
9: $Radius_1 \leftarrow \xi[A : Stop[i]]$;
10: $Radius_2 \leftarrow \xi[Stop[i] : B]$;
11: **if** $i == 1$ **then**
12: $PSR_1 \leftarrow circle(Center, Radius_2)$;
13: **else if** $i == SizeOf(Stop)$ **then**
14: $PSR_{SizeOf(Stop)} \leftarrow circle(Center, Radius_1)$;
15: **else**
16: $PSR_i \leftarrow asymmetric\_semicircles(L_i, Center, Radius_i, Radius_2)$;
17: **end if**
18: $i \leftarrow i + 1$;
19: **end while**

6.3.2 Selecting appropriate POI for replacement

After the above steps, we have got all stop points each associated with a PSR and labeled semantic attribute, and we regard a set $POIs$ with semantic attributes and a user-defined privacy level P_i as our background knowledge. As mentioned, the given user-defined privacy level P_i could be mapped to the corresponding height of the taxonomy tree h_i. First, we map the semantic attributes A of stop point set $Stop$ to the corresponding leaf nodes of the taxonomy tree $G = (V, E, l)$, denoted by ALN. For each stop point mapped to leaf node ALN_i, we can find out its specific internal node, denoted by *Internal_Candidate*, whose height is h_i and $l(ALN_i) \in l(Internal\_Candidate)$, then get the set of all possible semantic attributes *Semantic_Candidate* which equals to the set $l(Internal\_Candidate)$. We assume that the elements of the set $l(Internal\_Candidate)$ all have the same semantic category with the specific stop point while meeting the user-defined privacy level P_i. The parent node of *Internal_Candidate* can be denoted by *Similiar_Internal_Candidate*, so the elements of the set $l(Similiar\_Internal\_Candidate)$ could be said to have the similar semantic category with the stop point, and it corresponds to a set *Similiar_Semantic_Candidate*.

After this step, we can find a set of POIs which have the same or similar semantic category, *Semantic_Candidate* or *Similiar_Semantic_Candidate*, as the stop point meeting user privacy level. Then, we map the semantic attributes of the set $POIs$ to the set *Semantic_Candidate* or the set *Similiar_Semantic_Candidate*, and obtain the candidate POIs named *POI_Candidate*. However, each stop point's position has different isolation degrees, which could be divided into three cases according to distribution of neighbor POIs. Therefore, we use different methods to randomly select an appropriate POI, denoted by *POI_selected*, for the replacement process within the PSR. The three cases are as follows.

- **Non-isolated stop point:** A stop point is defined to be non-isolated if there exist some POIs belonging to the same semantic category with it within the corresponding PSR. These points are marked as the stop points among moving points of the trajectory, so they need to be replaced. Because the position is not that isolated, we could randomly select a POI of the same semantic directly for replacement. That is to randomly choose an element from the set *POI_Candidate* of the set *Semantic_Candidate*, while it is contained in the region PSR, as the appropriate *POI_selected*. After replacing the sensitive stop point, it not only satisfies the user requirements of privacy protection, but also keeps the semantic consistency of the protected trajectory and the original trajectory very well.
- **Isolated stop point:** A stop point is identified as isolated point while the corresponding PSR does not contain any POI of the same semantic category. This situation is for the stop point which is more unique on the map, then we adopt an approximate replacement method. That is to say, we select a POI of the similar semantic category within the PSR. We randomly choose an element of the set *POI_Candidate* of the set *Similiar_Semantic_Candidate* in the PSR as the appropriate *POI_selected*. This method can guarantee to meet the user requirements of privacy protection, and the selection is still carried in the PSR, so trajectory shape would not deviate too much. Unfortunately, the protected trajectory cannot keep high semantic consistency with the original trajectory. That means, we sacrifice some semantic consistency to ensure small shape deviation of trajectory data.
- **Quite-isolated stop point:** A stop point is identified as quite-isolated point if and only if there is not any appropriate POI to be chosen within the PSR, no matter from the same semantic category or the similar semantic category. It indicates that it is the only one with sensitive semantic attribute in a certain geographical area. Because the point was identified as the stop point containing the sensitive attributes that needs to be protected, direct release cannot meet the user requirements of privacy protection. There are two solutions: first, we use the dynamic expansion of PSR forcing to select an appropriate *POI_Candidate* that might be a little far away from the stop point; second, we just directly publish the stop point without any replacement. In the experiments, we will compare the performance of the two approaches in terms of average identification possibility and trajectory data availability. For the first approach, we first define a static extension step size, denoted by *expansion-step*. Then, *expansion-step* is utilized to expand the areas of both semi-circles. Specifically, expand the radius to $\frac{\overrightarrow{stop_{i+1}stop_i}}{2} + expansion\text{-}step$ and $\frac{\overrightarrow{stop_{i-1}stop_i}}{2} + expansion\text{-}step$ respectively, and select *POI_selected* to replace the stop point. That is to randomly choose an element of the set *POI_Candidate* of the set *Semantic_Candidate*, while it is contained in the expanded

selection region. However, we should take notice of another selection principle, which is to avoid choosing a POI from the overlapping area between the expanded selection region and its neighbor selection region. This checkup step to avoid reverse mutation is repeated until we find an appropriate *POI_selected* for replacement. This method would replace the stop point successfully, which satisfies the user requirements of privacy protection while maintaining the semantic consistency of the original trajectory well. However, the region expansion will lead to a larger degree of deviation from the original trajectory, which might lead to poor availability of trajectory data.

As shown in Figure 6, sampling points belonging to the same semantic category are represented by the same shape while sampling points belonging to the similar semantic category are represented by the similar shape. In the example, there are four semantic categories depicted by shapes like seven-corner-star, four-corner-star, triangle and circle. The seven-corner-star and four-corner-star represent the similar semantic categories. From this figure, we can see that the stop points $stop_1$ and $stop_2$ belong to non-isolated stop points depicted by triangle and circle, and we can randomly choose the *POI_selected* of same category for replacement within its PSR. $stop_3$ depicted by seven-corner-star can be regarded as isolated stop point, as in its PSR, there does not exist POIs shaped as seven-corner-star, but exist POIs shaped as four-corner-star. That means we can choose a *POI_selected* of similar semantic category within its PSR to replace it. Obviously, $stop_4$ depicted by four-corner-star belongs to quite-isolated sop points, so we use the process of dynamic expansion, but pay special attention not to choose a POI in the overlapping area of its expanded selection region and the neighbor selection region, as the figure shows. This part is used to explain how to select the appropriate POI for replacement. Next, we will discuss the reconstruction process of trajectory in detail, and finally release the protected trajectory.

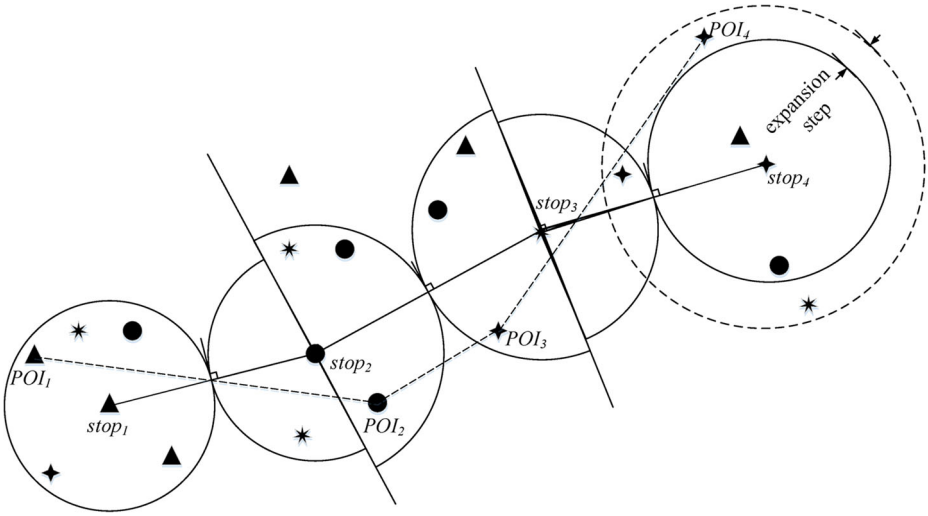


Figure 6 Example of choosing an appropriate POI

Algorithm 3 Selecting an appropriate POI

Input: The set of stop point $Stop$, the corresponding PSR, the user-defined privacy level P, the taxonomy tree $G = (V, E, l)$, the POI set $POIs$

Output: The set of POIs *POI_selected* for replacement of all stop points

$h \leftarrow map(P \rightarrow G)$;
$i \leftarrow 1$;
$k \leftarrow 1$;
$m \leftarrow 1$;
while $i < SizeOf(V)$ **do**
  if $h(V_i) == h$ **then**
    $HN[k] \leftarrow V[i]$;
    $k \leftarrow k + 1$;
  end if
  if $h(V_i) == h + 1$ **then**
    $SHN[m] \leftarrow V[i]$;
    $m \leftarrow m + 1$;
  end if
  $i \leftarrow i + 1$;
end while
$i \leftarrow 1$;
while $i \leq SizeOf(Stop)$ **do**
  while $k \leq SizeOf(HN)$ **do**
    if $Stop_i \in l(HN[k])$ **then**
      Semantic_Candidate $\leftarrow l(HN[k])$;
    end if
  end while
  while $k \leq SizeOf(SHN)$ **do**
    if $Stop_i \in l(SHN[k])$ **then**
      Similiar_Semantic_Candidate $\leftarrow l(SHN[k])$;
    end if
  end while
  $j \leftarrow 1$;
  $m \leftarrow 1$;
  $k \leftarrow 1$;
  while $j \leq SizeOf(POIs)$ **do**
    if $POIs[j][a] \in$ *Semantic_Candidate* **then**
      POI_Candidate$[m] \leftarrow POIs[j]$;
      $m \leftarrow m + 1$;
    end if
    if $POIs[j][a] \in$ *Similiar_Semantic_Candidate* **then**
      Similiar_POI_Candidate$[k] \leftarrow POIs[j]$;
      $k \leftarrow k + 1$;
    end if
    $j \leftarrow j + 1$;
  end while
  if $Stop_i$ is a non-isolated stop point **then**
    $POI\_seleted_i \leftarrow Randomly\_select$(*POI_Candidate*, PSR_i);
  end if
  if $Stop_i$ is an isolated stop point **then**
    $POI\_seleted_i \leftarrow Randomly\_select$(*Similiar_POI_Candidate*, PSR_i);
  end if
  if $Stop_i$ is a quite-isolated stop point **then**
    $POI\_seleted_i \leftarrow Randomly\_select$(*POI_Candidate*, PSR_i, *expansion-step*);
  end if
  $i \leftarrow i + 1$;
end while

Related pseudo code is shown in Algorithm 3. The inputs of the algorithm are the set of stop points $Stop$, PSR_i corresponding to each $Stop_i$, the user-defined privacy level P, the taxonomy tree $G = (V, E, l)$, and the POI set $POIs$ in the user space environment. The output is the set of POIs *POI_selected* for replacement of all stop points. Lines 4-15 pay attention the user-defined privacy level P and get the internal nodes from the taxonomy tree while its height equals to P. Lines 11-27 find *Semantic_Candidate* and *Similiar_Semantic_Candidate* while lines 28-41 find the candidate POIs for selection. Then, lines 42-50 adopt different methods to choose appropriate *POI_selected* for different types of stop points.

6.4 Reconstructing trajectory

This section discusses the detailed reconstruction process of trajectory considering some necessary factors. Our goal is that the published trajectory should maintain the maximum semantic consistency and the minimum shape deviation from the original trajectory. Moreover, reconstruction of trajectory has two main factors to be considered in terms of improving the trajectory data availability as much as possible. One is that we replace not only stop points, but also some other moving points on the trajectory as well, in order to avoid the sudden change of positions on the reconstructed trajectory. Another is that we take obstacles in the user space environment into consideration, in order to produce a more realistic trajectory. The published trajectory is also a sequence of sampling points at every sampling time.

As Figure 7 shows, for ease of description we label the stop point $stop_i$ as O, and label the selected POI *POI_selected* as O'. Similarly, we mark the midpoint between $stop_i$ and $stop_{i-1}$ as A and the midpoint between $stop_i$ and $stop_{i+1}$ as B. Our proposed trajectory reconstruction algorithm can be expressed as follows: firstly, we try to find a moving point in the trajectory segment $A \rightarrow O$, namely C. The point C must satisfy that the difference between the length of trajectory segment $C \rightarrow O$ and the length of trajectory segment $C \rightarrow O'$ is minimal. After selecting C, we uniformly generate points on the segment $C \rightarrow O'$ with the same number of moving points on the trajectory segment $C \rightarrow O$. The reason is that this method can avoid the sudden position change of velocity. Similarly, we find a point D on the segment $O \rightarrow B$ and generate a certain number of points on the segment $O' \rightarrow D$. Finally, for the stop point O, we use the trajectory segment $A \rightarrow C \rightarrow O' \rightarrow D \rightarrow B$ to take the place of the original trajectory segment $A \rightarrow O \rightarrow B$. This reconstruction process is applied to every stop point until the reconstruction of the whole user trajectory is completed.

From Figure 7, we can easily find that the trajectory reconstruction algorithm not only replaces stop points, but also replaces some moving points on the trajectory. This can effectively avoid position mutations on the reconstructed trajectory. Otherwise, third party or malicious attackers can easily find such sudden changed positions on the published trajectory, and then they can infer users' personal privacy according to the sensitive information of these special positions. As Figure 8 shows, O' is the selected POI and O is the corresponding stop point. In the case that O' deviates too far from O, if we directly connect O' with the previous moving point and the next moving point, as the figure shows, then the reconstructed trajectory segment will cause a serious position mutation. Therefore, we should replace some moving points as well to avoid this problem.

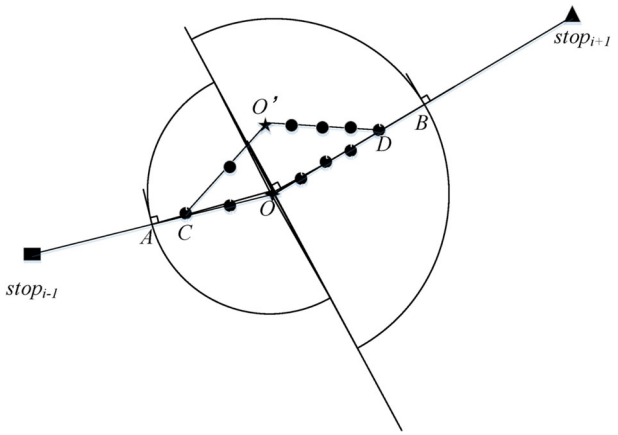


Figure 7 Example of trajectory reconstruction

What we also should take note is that there are often many obstacles in the real physical environment. Therefore, our proposed algorithm needs to detect whether the reconstructed trajectory segment goes through obstacles. If it does, then we need to reselect an appropriate POI. This step can avoid that malicious attackers easily find the trajectory segment passing through the obstacles and the replacement of sensitive stop points is disclosed, and guarantee the published protected trajectory data with a high utility to be in line with actual situations. Assume that we directly publish the trajectory which goes through an obstacle, then the malicious attackers can quickly find this special trajectory segment. As Figure 9 shows, the attackers can easily find that the trajectory segment $C \rightarrow O'$ goes through an obstacle, then they can infer that the location O' may be a substitute and then infer the real location of stop point $stop_i$. Thus, it is necessary to consider the spatial distribution of obstacles when reconstructing trajectory.

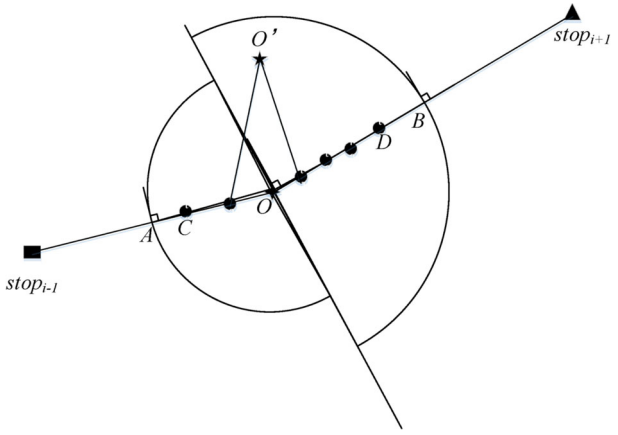


Figure 8 Location mutation in trajectory reconstruction

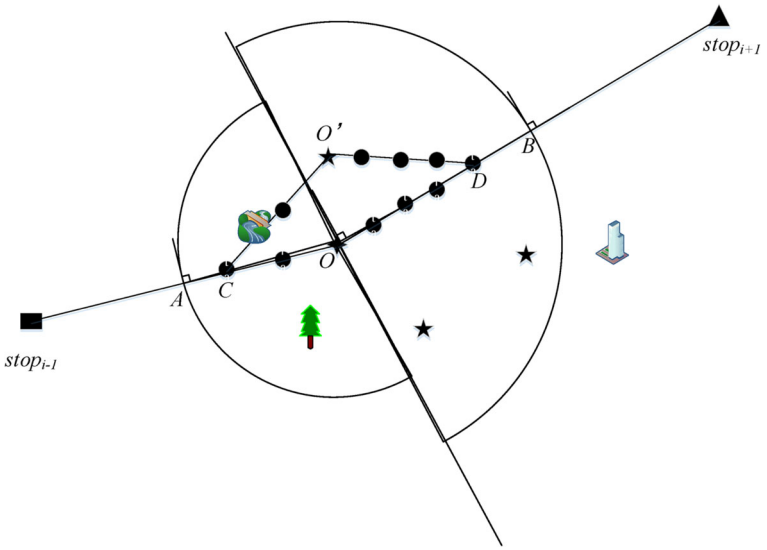


Figure 9 Trajectory going through obstacle region

Algorithm 4 Reconstructing the user trajectory

Input: The original trajectory T_i of a specific user, the set of stop points $Stop$ among T_i, the selected POI $POI\_selected$ corresponding to each $Stop_i$, the set of obstacles χ within the user space environment
Output: The reconstructed trajectory T_i^*

1: $i \leftarrow 2$;
2: **while** $i < SizeOf(Stop)$ **do**
3: $\xi_1 \leftarrow \xi[stop[i-1] : stop[i]]$;
4: $\xi_2 \leftarrow \xi[stop[i] : stop[i+1]]$;
5: $A \leftarrow MidPointOf(\xi_1)$;
6: $B \leftarrow MidPointOf(\xi_2)$;
7: $\xi_{seg1} \leftarrow \xi[A : Stop[i]]$;
8: $\xi_{seg2} \leftarrow \xi[Stop[i] : B]$;
9: select a point $C \in \xi_{seg1}$ making that $DisDeviation \leftarrow DistanceOf(C, POI\_selected) - DistanceOf(C, Stop[i])$ is minimal;
10: select a point $D \in \xi_{seg2}$ making that $DisDeviation \leftarrow DistanceOf(POI\_selected, D) - DistanceOf(Stop[i], D)$ is minimal;
11: $sum1 \leftarrow SumPointBetween(C, Stop[i])$;
12: $sum2 \leftarrow SumPointBetween(Stop[i], D)$;
13: generate $sum1$ moving points between C and $POI\_selected$ uniformly;
14: generate $sum2$ moving points between $POI\_selected$ and D uniformly;
15: $\xi^* \leftarrow [A : C : POI\_selected : D : B]$;
16: **if** $\xi^* \cap \chi \neq \varnothing$ **then**
17: reselect $POI\_selected$;
18: **end if**
19: **end while**

Related pseudo code is shown in Algorithm 4. The inputs include the original trajectory T_i of a specific user, the set of stop points $Stop$ among T_i, the selected POI $POI\_selected$ corresponding to each $Stop_i$, the set of obstacles χ within the user space environment. The output is the reconstructed trajectory T_i^*. Lines 5-6 look for the points A and B. Lines 7-14 uniformly generate moving points on the reconstructed trajectory segments. Lines 15-18 detect whether the reconstructed trajectory segment goes through obstacles. If yes, then it is needed to reselect another suitable POI and reconstruct the trajectory segments.

7 Evaluation criteria

As discussed, trajectory protection requires to meet different privacy requirements of users and also maintain a relatively high trajectory data availability of semantic consistency and shape similarity. Therefore, we evaluate the performance of our algorithm in terms of three aspects, namely *average identification possibility*, *trajectory shape similarity* and *trajectory semantic consistency*.

7.1 Average identification possibility

In this paper, we pay attention to all stop points among the user trajectory, to prevent attackers to identify the user's complete trajectory sequence or infer its corresponding sensitive information.

For each stop point among a specific trajectory, we get its candidate POI set *POI_Candidate* (abbreviated as PC), and randomly select a POI *POI_selected* from the candidate POI set to replace the stop point. The selection process cannot be repeated, so the possibility that the attackers can identify current position of the stop point is inversely proportional to the size of candidate POI set. That is to say, the larger the value of $|PC|$ is, the smaller identification possibility of the current sensitive position is, and the higher the privacy protection degree can achieve. Therefore, for a trajectory database, we first investigate each stop point on each user's trajectory. Let IP_i be the identification possibility of the stop point, it can be calculated by

$$IP_i = \frac{1}{|PC|} \tag{1}$$

We can see that, for each stop point, usually the identification possibility $IP_i < 1$, and $IP_i = 1$ only if we directly publish the stop point without replacement (as the value of $|PC| = 1$). It is worth noting that the process to obtain PC of each stop point is firstly to map the privacy level and its semantic attribute to the taxonomy tree, then get the value of an internal node which represents the same category of the stop point, next we can get the possible semantic attributes meeting the user protection requirements, and finally match POIs to these semantic attributes and form PC. There are some calculation differences for three types of stop points. For the first type *non-isolated stop point* and the third type *quite-isolated stop point*, we choose *POI_selected* from PC belonging to same semantic category, but for the second type *isolated stop point*, we choose from its similar semantic category. Overall, the value of $|PC|$ indicates the number of POIs we can choose from the selection region with the same or similar semantic category. Obviously, the calculation of PC considers many factors, such as user-defined privacy level, semantic attributes of stop points and POIs, selection region and so on. In general, although calculation method is slightly different for different types of stop points, the value of IP_i can be a good indication of the re-identification possibility of this sensitive position that attackers can achieve.

Let T_i^* represent the reconstructed trajectory, $Stop$ be the set of all stop points and AIP_i denote the average identification possibility of a user trajectory data record. In addition, let T represent the whole trajectory database, and O represent the set of all moving objects, so the average identification possibility of the whole trajectory database T is set as AIP. We calculate AIP as

$$AIP = \frac{\sum_1^{|O|}(\frac{\sum_1^{|Stop|} IP_i}{|Stop|})}{|O|} \tag{2}$$

In this equation, the average identification possibility of a user trajectory is an average value of all the stop points' identification possibilities. The average identification possibility of the whole trajectory database is an average value of all the user trajectory records' average identification possibilities. In general, the smaller the value of AIP is, the higher degree the privacy protection is, i.e., the more effective an algorithm is.

7.2 Trajectory shape similarity

Since normally it is required to publish the protected trajectory database to third parties for a variety of purposes, we need to ensure the shape deviation of the reconstructed trajectory and the original user trajectory is as small as possible. In this paper, we take the distance and angle into account. Following [10] we use the two criteria, namely *trajectory direction deviation* and *trajectory distance utility*, to describe the shape similarity of the two trajectories. We introduce their definitions as follows.

7.2.1 Trajectory direction deviation

Definition 4 (Trajectory Angle) Let T_1 and T_2 be two trajectories with n moving points, that is to say each trajectory has $n-1$ trajectory segments. The trajectory segment denoted by $\overrightarrow{T_i^k}$ in the time interval $[t_i, t_{i+1}]$ of $T_k (k = 1, 2)$ is from (x_i^k, y_i^k) to (x_{i+1}^k, y_{i+1}^k) $(1 \le i \le n-1)$, in which (x_i^k, y_i^k) represents the position of the trajectory T_k at time t_i, so as (x_{i+1}^k, y_{i+1}^k). The trajectory segment angle $\theta_i (\theta_i \in [0, \pi])$ can be calculated by

$$\begin{aligned} cos\theta_i &= \frac{\overrightarrow{T_i^1} \cdot \overrightarrow{T_i^2}}{\left|\overrightarrow{T_i^1}\right| \cdot \left|\overrightarrow{T_i^2}\right|} \\ &= \frac{(x_{i+1}^1 - x_i^1) \cdot (x_{i+1}^2 - x_i^2) + (y_{i+1}^1 - y_i^1) \cdot (y_{i+1}^2 - y_i^2)}{\sqrt{(x_{i+1}^1 - x_i^1)^2 + (y_{i+1}^1 - y_i^1)^2} \cdot \sqrt{(x_{i+1}^2 - x_i^2)^2 + (y_{i+1}^2 - y_i^2)^2}} \end{aligned} \tag{3}$$

Because cosine function is monotonically decreasing in the interval $[0, \pi]$, a larger value of $cos\theta_i$ indicates the smaller angle of θ_i, so as the direction deviation of the two trajectory segments, and hence the larger the shape similarity of these two trajectory segments is. Therefore, we define the direction deviation of two user trajectory data records as TDD, which can be expressed as

$$TDD(T_1, T_2) = \frac{\sum_{i=1}^{n-1} cos\theta_i}{n-1} \tag{4}$$

In this equation, the trajectory direction deviation of two trajectories is the average value of all $cos\theta_i (1 \le i \le n-1)$ of the corresponding $n-1$ trajectory segments. In our work, we use these two formulas to describe the angle similarity between the reconstructed trajectory and the original trajectory. Let T_i represent one specific trajectory of a moving object, and its reconstructed trajectory denoted by T_i^*, so we can use $TDD(T_i, T_i^*)$ to represent the trajectory direction deviation. In addition, if the speed direction of the two trajectory segments is different, then the angle θ_i between the two trajectory segments will be within the range $[\frac{\pi}{2}, \pi]$ leading to $cos\theta_i \le 0$. In this case, we suppose $cos\theta_i = 0$. In fact, the definition of selection region and trajectory reconstruction have already avoided this problem, so we do not need to take such trajectory segments into consideration. Obviously, the larger

$TDD(T_i, T_i^*)$ is, the smaller the angle of two trajectories, then the more similar between the reconstructed trajectory and the original trajectory in terms of direction shape, and hence an algorithm performs better.

7.2.2 Trajectory distance utility

Definition 5 (Trajectory Distance) Let T_1 and T_2 be two trajectories with n moving points, and the trajectory sampling time range is $[t_1, t_n]$. We use the average of the Euclidean distances of corresponding moving point pair on the two trajectories to represent their trajectory distance. The so-called moving point pair is the positions of two trajectories at the same sampling time t_i, namely (x_i^1, y_i^1), (x_i^2, y_i^2). Therefore, the whole trajectory distance can be calculated by

$$Distance(T_1, T_2) = \frac{\sum_{i=1}^{n} \sqrt{(x_i^2 - x_i^1)^2 + (y_i^2 - y_i^1)^2}}{n} \tag{5}$$

Let T_i represent one specific trajectory of a moving object, with its reconstructed trajectory denoted by T_i^*. We use $Diatance(T_i, T_i^*)$ to represent the trajectory distance, and introduce the maximum radius of all the selection regions produced in the selection process, denoted as $MaxRad$. Finally, we calculate the *trajectory distance utility* through

$$TDU(T_i, T_i^*) = 1 - \frac{Diatance(T_i, T_i^*)}{MaxRad} \tag{6}$$

Note that, while calculating the distance similarity of two trajectories, we consider all the moving points among the published trajectory, rather than only the stop points that have been replaced, as we also replace some moving points to avoid position mutations. Obviously, the value of $Diatance(T_i, T_i^*)$ represents the average distance deviation of all moving points among the trajectory, so the smaller the value is, the better an algorithm is. However, the value might be a large quantized distance, e.g., 10 kilometers. These large values cannot be an evaluation to compare the trajectory distance utility directly, so we define the formula as above to measure the results. Through this normalization, the trajectory distance utility will be in the range of [0, 1]. The value of utility is larger, then the distance deviation is smaller.

In general, trajectory direction deviation can represent the direction similarity between the original trajectory and the reconstructed trajectory while trajectory distance utility represents the distance similarity. In addition, for the trajectory database, we use an average value of the trajectory direction deviation between each user trajectory and its protected trajectory to express the whole trajectory database. We use the same method to represent trajectory database's trajectory distance deviation.

7.3 Trajectory semantic consistency

We also need to consider the trajectory semantic consistency (TSC), between a user trajectory and its protected trajectory. The smaller the semantic deviation degree is, the higher value of the published trajectory data will have. We firstly consider individual stop point. If the selected POI belongs to the same semantic category of the stop point, the semantic is totally kept meeting the user-defined settings. In this case, its semantic consistency, denoted by TSC_i, equals to 1. This occurs for the first type *non-isolated stop point* and the third type *quite-isolated stop point*, but for the second type *isolated stop point*, we choose from its similar semantic category, so it cannot keep the semantic consistency very well. We

calculate its TSC_i considering the amount of internal nodes representing all similar semantic categories of the stop point. According to the previous definition, the sibling nodes of *Internal_Candidate* or the child nodes of *Similiar_Internal_Candidate* represent all the similar semantic categories of the stop point. If *Internal_Candidate* means the same category to keep the semantic consistency, then for the second type stop point, we can choose form the similar semantic categories which could be calculated as the number of all sibling nodes of *Internal_Candidate*, abbreviated as SIC. Thus, each stop point's semantic consistency TSC_i can be calculated as

$$TSC_i = \begin{cases} 1 & for\ non\text{-}isolated\ stop\ point\ or\ quite\text{-}isolated\ stop\ point \\ \frac{1}{|SIC|} & for\ isolated\ stop\ point \end{cases} \tag{7}$$

Then, we calculate the semantic consistency TSC of a trajectory corresponding to a moving object as

$$TSC = \frac{\sum_1^{|Stop|} TSC_i}{|Stop|} \tag{8}$$

Obviously, the semantic consistency of a trajectory database is an average value of all user trajectories' semantic consistencies. In general, the higher the value of TSC is, the more similar of semantic between the original trajectory and its protected trajectory, which means, the more effective an algorithm is.

8 Experiments

In this section, we first report our experimental data set, parameters and compared approaches. Then, we follow the three evaluation criteria to analyze the results of our experiments.

8.1 Experimental setup

We run our experiments on a synthetic trajectory database based on real GPS trajectories and with some simulated data. The real GPS trajectory data set was collected in *Geolife* project [32] by 182 users in a period of over five years (from April 2007 to August 2012), which contains 17,621 trajectories with a total distance of 1,292,951 kilometers and a total duration of 50,176 hours. The majority of the data was created in Beijing, China. The positions of all moving points are represented by altitude and longitude coordinates, so we transform these GPS coordinates into two-dimensional plane coordinates in the pre-processing. Moreover, we would generate a certain number of simulated POIs and obstacles in the user space environment, together with the semantic attributes of POIs which form the POI attribute tables later.

Besides, we simulate and define the set of sensitive semantic attributes for different users in the beginning. During the process of extracting stop points, we set the duration threshold th_{time} as 30 minutes, and also set the distance threshold th_{dist} as 100 meters. As for the process of selecting the appropriate POI, in particular for the third type *quite-isolated stop point*, there are two different solutions: one approach is we use the dynamic expansion of the selection region forcing to select an appropriate POI that might be a little far away from the stop point; another approach is we just directly publish the stop point without replacement. In our experiment, we set the value of *expansion-step* as 100 meters.

The main purpose of our experiments is to show verify the performance of our algorithm and compared two different approaches mentioned above during the process of POI

selection. We call the approach which directly publishes the sensitive stop points as *no replacement* while the approach expanding the selection region as *expansion replacement*. In order to compare them under different conditions, we test with different numbers of POIs and different user-defined privacy levels, respectively. For each setting, we run the experiments 20 times to report average result. In addition, we test the effect of obstacles. Therefore, the experiment is also performed by changing the number of obstacles.

In the experiments, we assume the user-defined privacy levels to be 1, 2, 3, and discuss the performance of our proposal for the different user requirements. Then, considering the effect of obstacles, we represent each obstacle's shape as a circle. The center of each circle is generated at random in the area, while the radius is set to a random number between 10m and 100m. According to our settings, there are two conditions to discuss. One is we assume there is no obstacles in user space environment, which is the most ideal but less realistic. Another is we assume there are some obstacles in order to make our algorithm more realistic, and we set the number of obstacles as 200. Under each user-defined privacy level and corresponding number of obstacles, we generate POIs at random in the area determined by trajectory data set. The number of POIs is set to 2000, 4000, $\cdots$, 10000 respectively, with a fixed step size as 2000. We test the performance of our proposal in terms of the three evaluation criteria *average identification possibility*, *trajectory semantic consistency* and *trajectory shape similarity*. In general, we study how the three variables, namely privacy level, obstacles and POIs affect these evaluation criteria of the two approaches, and to verify the effectiveness of the enhanced approach *expansion replacement*.

8.2 Experimental results

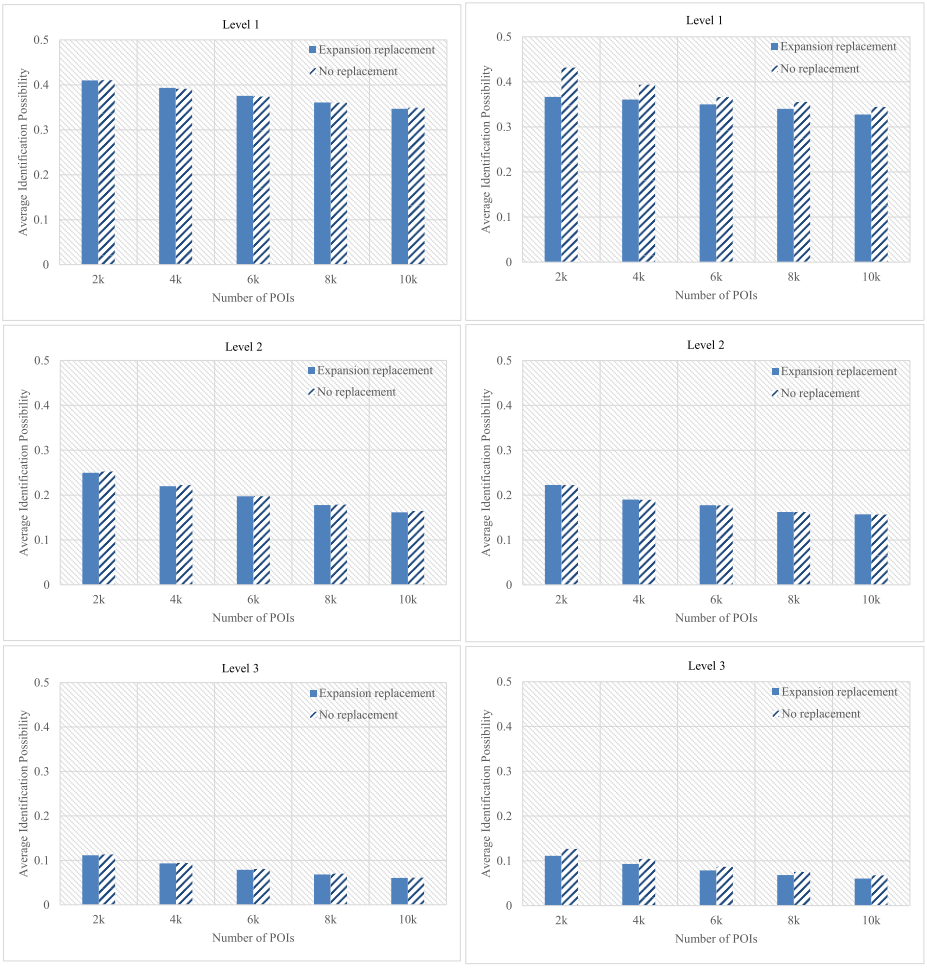
In this part, we conduct the controlled experiments and analyse the results of two different approaches from different aspects according to the three evaluation criteria.

8.2.1 Study on average identification possibility

Figure 10 shows the achieved average identification possibility of the two approaches with the different numbers of POIs under different settings about privacy level and obstacles. Figure 10a represents the experimental results for different privacy levels while there are no obstacles in the user space environment, and Figure 10b shows the experimental results for different privacy levels while there are 200 obstacles in the user space environment.

First, from these figures, we can see the enhanced *expansion replacement* approach can achieve a lower average identification possibility than the *no replacement* approach. The lower average identification possibility is, the higher protection degree we can achieve. However, as POIs around sensitive stop points are densely distributed, there are not many third type of stop points, so both two approaches achieve a small average identification possibility which is about 0.1 0.4.

Then, we should pay attention to how the three variables affect our experimental results. With the increase of the number of POIs, the average identification possibility is reduced. This is because the larger number of POIs is, the more POIs belong to the same or similar semantic category of each stop point, i.e., *POI_Candidate* becomes larger, and the average identification possibility would be lower. Note that with the increase of user-defined privacy level, the average identification possibility is reduced for both approaches regardless of the existence of obstacles. When the user-defined privacy level equals to 3, the average identification possibility even reduces to 0.1. When the privacy level increases, the POI candidate set corresponding to each stop point will certainly increase, and average



(a) No obstacle (b) 200 obstacles

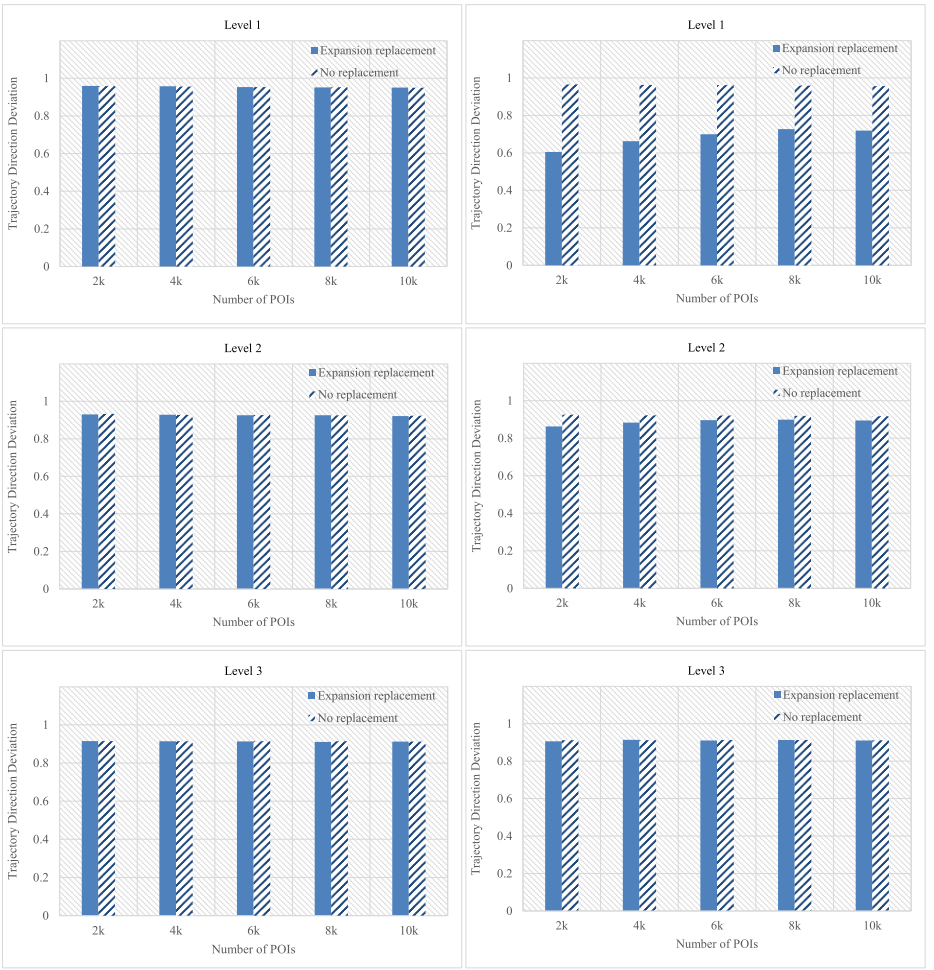
Figure 10 Comparison of average identification possibility with different values of privacy levels, obstacles and POIs for two approaches

identification possibility decreases. From another point of view, by comparing the right and left parts of Figure 10, we can see the influence of obstacles on the two approaches is not very significant. This is a good news for us because it shows that we can reconstruct much more realistic trajectories while achieving almost the same privacy protection degree.

8.2.2 Study on trajectory shape similarity

As for the discussion of trajectory shape similarity, we use two evaluation criteria to indict its performance, namely *trajectory direction deviation* and *trajectory distance utility*.

Figure 11 shows the trajectory direction deviation of the two approaches with the different numbers of POIs under different settings about privacy level and obstacles. Figure 11a represents the experimental results for different privacy levels while there are no obstacles



(a) No obstacle (b) 200 obstacles

Figure 11 Comparison of trajectory direction deviation with different values of privacy levels, obstacles and POIs for two approaches

in the user space environment, while Figure 11b shows the experimental results for different privacy levels while there are 200 obstacles in the user space environment.

As mentioned above, the larger TDD is, the more similar between the reconstructed trajectory and the original trajectory in terms of direction shape, and hence the better performance. From these figures, the experimental results show that values of TDD are in the range of 0.9 to 1, i.e., the average angle deviation of the original trajectory and the reconstructed trajectory is between 0 and 25 degrees. This result is quite promising, considering that we randomly select an appropriate POI from the selection region which meets the requirements. Random selection may cause a large angle deviation, but our algorithm effectively avoids the problem by reasonably determining the area of the selection region.

We can see the trajectory direction deviation of our enhanced *expansion replacement* approach is a little larger than the *no replacement* approach, as *expansion replacement*

would somehow expand its selection region in order to find an appropriate POI. However, the deviation from the angle of change range is constrained in a very small range (the maximum deviation of the two approaches is only 6 degrees). This means that our enhanced approach is able to achieve a smaller degree of angle deviation from the perspective of maintaining a much higher degree of privacy protection.

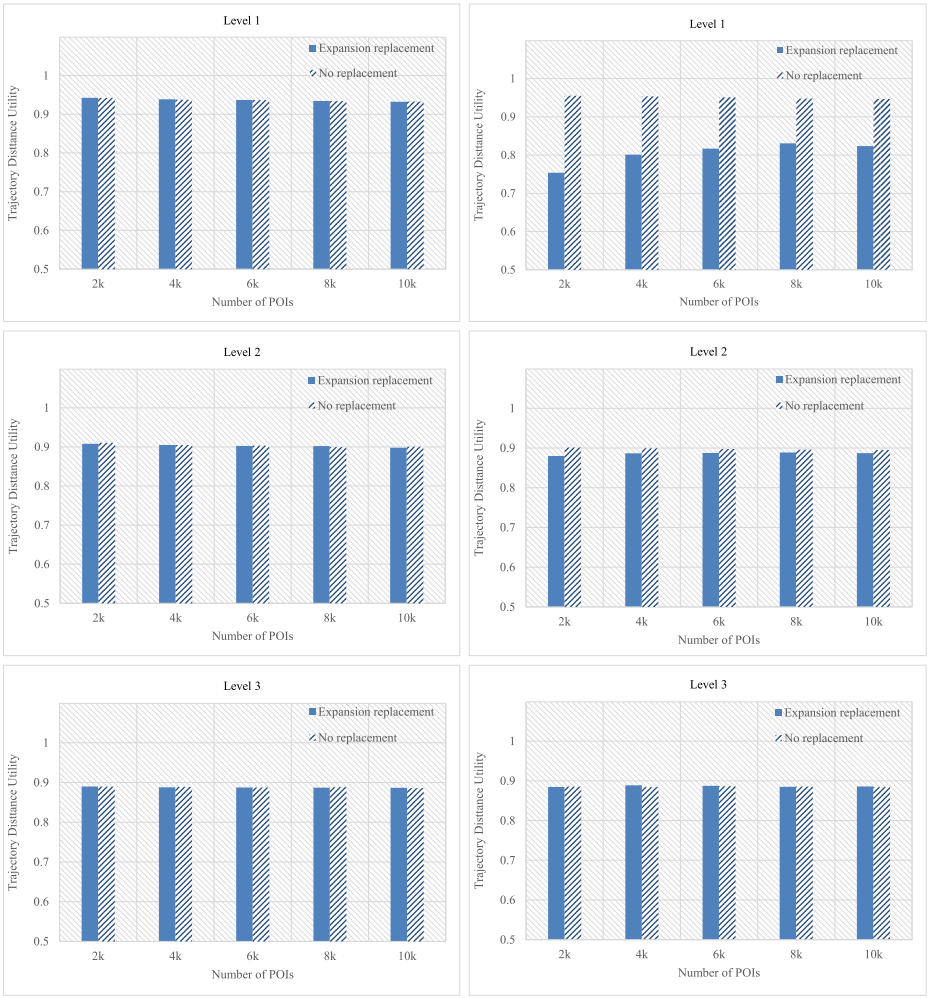
We also exam how the three variables affect the experimental results. Because the direction deviation of our reconstructed trajectory largely depends on the proper selection region, in our proposal we reasonably define the selection region considering user speed and reverse mutation. As the change of the number of POIs or the user-defined privacy level and whether there are obstacles have little effect on a selection region, these three factors have little influence on the trajectory direction deviation due to the random selection. This is a good news for us. We can ensure to achieve a smaller trajectory direction deviation and a higher degree of privacy protection with changes of external conditions.

Figure 12 shows the achieved trajectory distance utility of the two approaches with the different numbers of POIs under different settings about privacy level and obstacles. Figure 12a represents the experimental results for different privacy levels while there is no obstacles in the user space environment, while Figure 12b shows the experimental results for different privacy levels while there are 200 obstacles in the user space environment.

According to the definition of *trajectory distance utility*, the value of utility is larger, then the deviation is smaller. The higher the degree of similarity of trajectory distance shape is, the performance is better. First, from these figures, we can see although the enhanced *expansion replacement* approach gains lower trajectory distance utility compared with the *no replacement* approach when the user-defined privacy level equals to 1, they both achieve a high trajectory distance utility, which almost maintains at a high range of 0.85 to 0.9. Our proposal takes an expense of a very small trajectory distance availability to maintain a higher degree of privacy protection, and the distance utility mostly relies on the position of selected POI, so the utility also depends on its selection region to a great extent. As the enhanced approach would somehow expand the selection region to get candidate POIs, the selected POI would be a little far from the original stop point for the quite-isolated stop point, then the trajectory distance utility would be a little lower than the approach which directly publishes the original position.

If the number of POIs increases or privacy level increases, the number of candidate POIs would also increase, and note that, to deal with obstacles, we need to reselect the POI, so it is very likely that we need to expand the selection region, which leads to a decline in the trajectory distance utility. However, the number of POIs and obstacles and privacy level are not the decisive factors here, as distance utility mainly depends on the selection region and random selection. Thus, theoretically the effect caused by these three factors will be uncertain. In addition, we can guarantee that our trajectory reconstruction always keeps the high distance utility with the increase of these three variables.

Through the above discussion, we can see that the trajectory shape similarity largely depends on the position of the selected POI, as selection region and random selection are pivotal. Regardless of the number of POIs and obstacles and privacy level, although the enhanced *expansion replacement* approach may cause some decrease of trajectory shape similarity compared with *no replacement*, it still keeps a high value of shape similarity. Our proposal aims to slightly sacrifice shape similarity to greatly improve the degree of privacy protection, and we also construct more realistic trajectories considering obstacles in user space environment.



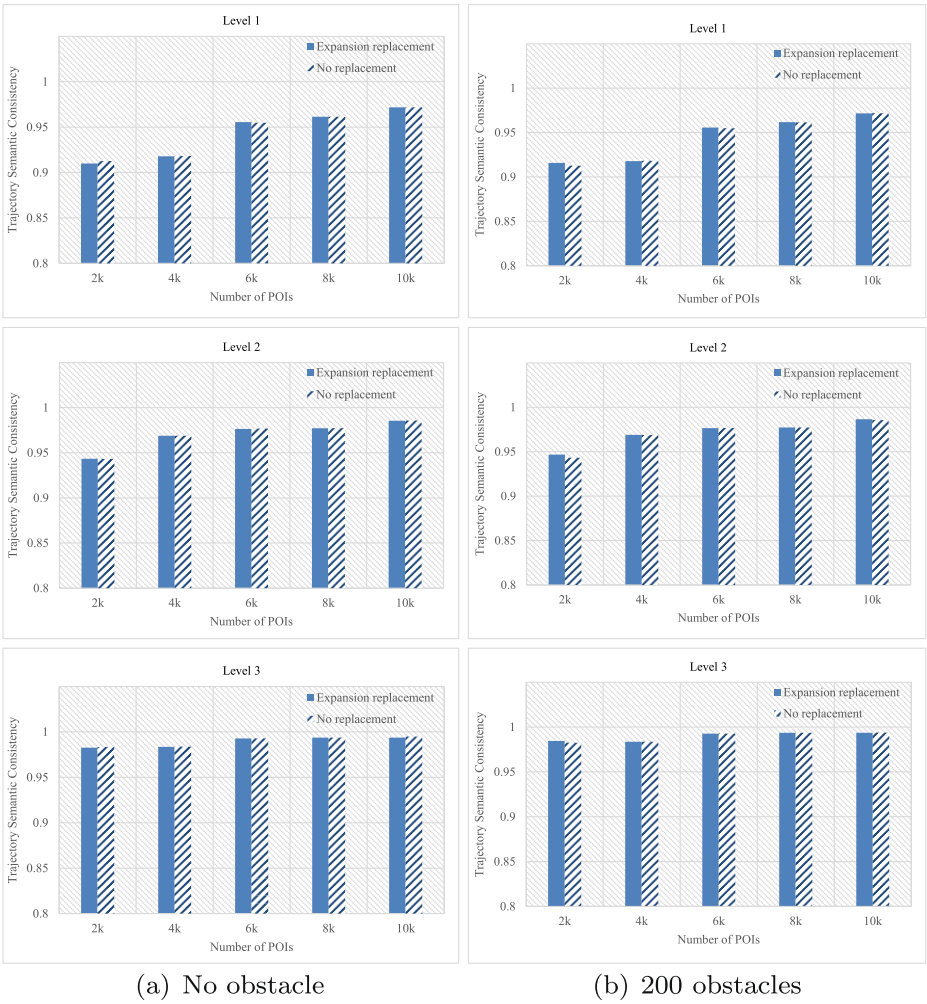
(a) No obstacle (b) 200 obstacles

Figure 12 Comparison of trajectory distance utility with different values of privacy levels, obstacles and POIs for two approaches

8.2.3 Study on trajectory semantic consistency

Figure 13 shows the achieved trajectory semantic consistency of the two approaches with the different numbers of POIs under different settings about privacy level and obstacles. Figure 13a represents the experimental results for different privacy levels while there are no obstacles in the user space environment, while Figure 13b shows the experimental results for different privacy levels while there are 200 obstacles in the user space environment.

One novelty of our work is to fully consider the semantic attributes, so during the discussion of reconstructing trajectories, we should not only make sure the reconstructed trajectory to keep the maximum similarity in shape with the original trajectory, but also to make sure the reconstructed trajectory to maintain consistency in the semantic attributes as much as



(a) No obstacle (b) 200 obstacles

Figure 13 Comparison of trajectory semantic consistency with different values of privacy levels, obstacles and POIs for two approaches

possible. From the above definition of trajectory semantic consistency, if we choose an appropriate POI from its same semantic category, we define it fully maintains the semantic consistency and set the value of TSC as 1. This case is mainly for the first and third types of stop points, i.e., non-isolated and quite-isolated stop points, as both approaches would choose the POI from the same semantic category or directly publish. For the second type, isolated stop point, we choose from the similar semantic category, so the semantic consistency depends on the number of sibling nodes of its *Internal_Candidate*. The more sibling nodes are, the lower the trajectory semantic consistency is. Then, we can say the value of trajectory semantic consistency is most related to the taxonomy tree built according to semantic attributes of all moving points among the trajectory database and the number of the second type of stop points.

First, from these figures, we can see the enhanced *expansion replacement* approach and the *no replacement* approach both achieve a very high trajectory semantic consistency (almost 0.9 to 1). For the user-defined privacy level, our proposal ensures that trajectory semantic consistency can reach nearly 1, which means the reconstructed trajectory can almost completely keep the semantic consistency under the premise of satisfying the user's privacy requirements. This result is consistent with our core idea, which is designed to meet the needs of different users' privacy level (as long as the requirements of user privacy level are met, we regard the trajectory semantic consistency is completely kept). Only for the second type of stop point, we make small sacrifice on privacy protection to ensure the trajectory similarity. Overall, we will keep a very high semantic consistency. As the difference of two compared approaches in the experiments lies in the third type of stop point, the two approaches should achieve almost the same performance of the trajectory semantic consistency, which can be observed from the experimental results.

Then, we investigate how the three variables affect the experimental results. In our experiments, we simulate the number of POIs' attributes to be about 240, and the number of POIs from 2K to 10K. In addition, each user trajectory contains at least 2K moving points. After matching the nearest POI's semantic attribute to the sampling point, the trajectory database contains almost all the simulated attributes, so the taxonomy tree is almost the same for a specific user regardless of the value of variables. Thus, we pay more attention to the number of the second type of stop points, isolated stop point. As for the increase of the number of POIs, the experimental results show that the trajectory semantic consistency will increase somehow. This is because the more POIs are, the more candidate POIs belong to the same semantic category of each stop point form the selection region. It can be seen that the number of isolated stop points would decrease, then the value of SIC becomes 1, and thus the trajectory semantic consistency would be larger. Note that, with the increase of user-defined privacy level, the trajectory semantic consistency also increases for both two approaches, regardless of the existence of obstacles. This is because when the privacy level increases, the height of *Internal_Candidate* increases as well, and according to the definition of taxonomy tree, the number of sibling nodes of *Internal_Candidate* would decrease, so as SIC. Thus, for the isolated stop point, the value of TSC_i would increase. In addition, the impact of obstacles is minimal, so we can construct trajectories more realistically.

9 Conclusion and future work

Leakage of trajectories may pose serious threats to users' personal privacy since not only their temporal and spatial attributes, but also semantic attributes could be exposed. Most previous studies consider a same level of privacy protection in trajectory database for all moving objects. In this paper, we present a novel method for privacy preservation in trajectory data publishing scenario, through trajectory reconstruction after reasonably replacing sensitive stop points. In order to achieve a good balance of data availability and privacy protection, we first select the sensitive stop points among the user trajectory, and for different types of stop points we propose different methods to select an appropriate POI for replacement. In addition, the environment conditions are taken into consideration, such as user speed and reverse mutation, and we consider the position mutations and obstacles in trajectory reconstruction. Finally, the performance of our proposal is comprehensively evaluated. The results show that the *expansion replacement* approach can improve the trajectory semantic consistency and shape similarity as much as possible, while effectively achieving the different trajectory privacy protection requirements of users.

We have mentioned that the reconstruction of trajectory can prevent the attribute-linkage attack and re-identification attack. However, it has not been fully verified with real environmental conditions, as the POIs, semantic attributes and obstacles are generated by simulation. In the future, we will develop a suitable model on the reconstruction of trajectory and conduct related experiments in real user space environment.

Acknowledgements This work is supported by the National Nature Science Foundation of China (grants No. 61672133, No. 61602087 and No. 61632007), the Fundamental Research Funds for the Central Universities (grants No. ZYGX2015J058 and No. ZYGX2014Z007), and a project funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology.

References

1. Abul, O., Bonchi, F., Nanni, M.: Never walk alone: uncertainty for anonymity in moving objects databases. In: Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7–12, 2008, Cancún, México, pp. 376–385 (2008)
2. Beresford, A.R., Stajano, F.: Location privacy in pervasive computing. IEEE Pervasive Comput. **2**(1), 46–55 (2003)
3. Domingo-Ferrer, J., Trujillo-Rasua, R.: Microaggregation- and permutation-based anonymization of movement data. Inf. Sci. **208**, 55–80 (2012)
4. Duckham, M., Kulik, L.: A formal model of obfuscation and negotiation for location privacy. In: Pervasive Computing, Third International Conference, PERVASIVE 2005, Munich, Germany, May 8–13, 2005, Proceedings, pp. 152–170 (2005)
5. Dwork, C.: Differential privacy. In: Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10–14, 2006, Proceedings, Part II, pp. 1–12 (2006)
6. Fu, Z., Huang, F., Ren, K., Weng, J., Wang, C.: Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data. IEEE Trans. Inf. Forensics Secur. **12**(8), 1874–1884 (2017)
7. Fu, Z., Ren, K., Shu, J., Sun, X., Huang, F.: Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans. Parallel Distrib. Syst. **27**(9), 2546–2559 (2016)
8. Fu, Z., Wu, X., Guan, C., Sun, X., Ren, K.: Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans. Inf. Forensics Secur. **11**(12), 2706–2716 (2016)
9. Gao, S., Ma, J., Shi, W., Zhan, G., Sun, C.: Trpf: a trajectory privacy-preserving framework for participatory sensing. IEEE Trans. Inf. Forensics Secur. **8**(6), 874–887 (2013)
10. Gao, S., Ma, J., Sun, C., Li, X.: Balancing trajectory privacy and data utility using a personalized anonymization model. J. Netw. Comput. Appl. **38**, 125–134 (2014)
11. Gidófalvi, G., Huang, X., Pedersen, T.B.: Privacy: preserving trajectory collection. In: 16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2008, November 5-7, 2008, Irvine, California, USA, Proceedings, p. 46 (2008)
12. Gkoulalas-Divanis, A., Verykios, V.S., Mokbel, M.F.: Identifying unsafe routes for network-based trajectory privacy. In: Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30–May 2, 2009, Sparks, Nevada, USA, pp. 942–953 (2009)
13. Gramaglia, M., Fiore, M., Tarable, A., Banchs, A.: $k^{\tau,\epsilon}$-anonymity: towards privacy-preserving publishing of spatiotemporal trajectory data. arXiv:1701.02243 (2017)
14. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the First International Conference on Mobile Systems, Applications, and Services, Mobisys 2003, San Francisco, CA, USA, May 5–8, 2003 (2003)
15. Gruteser, M., Liu, X.: Protecting privacy in continuous location-tracking applications. IEEE Secur. Priv. **2**(2), 28–34 (2004)
16. Han, P., Tsai, H.: SST: privacy preserving for semantic trajectories. In: 16Th IEEE International Conference on Mobile Data Management, MDM 2015, Pittsburgh, PA, USA, June 15–18, 2015, vol. 2, pp. 80–85 (2015)
17. Hazzard, A., Benford, S., Burnett, G.E.: You'll never walk alone: composing location-based soundtracks. In: 14th International Conference on New Interfaces for Musical Expression, NIME 2014, London, United Kingdom, June 30–July 4, 2014, pp. 411–414 (2014)

18. Huo, Z., Meng, X., Hu, H., Huang, Y.: You can walk alone: trajectory privacy-preserving through significant stays protection. In: Database Systems for Advanced Applications - 17th International Conference, DASFAA 2012, Busan, South Korea, April 15–19, 2012, Proceedings, Part I, pp. 351–366 (2012)
19. Komishani, E.G., Abadi, M., Deldar, F.: PPTD: Preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression. Knowl.-Based Syst. **94**, 43–59 (2016)
20. Krumm, J.: A survey of computational location privacy. Pers. Ubiquit. Comput. **13**(6), 391–399 (2009)
21. Li, M., Zhu, L., Zhang, Z., Xu, R.: Achieving differential privacy of trajectory data publishing in participatory sensing. Inf. Sci. **400**, 1–13 (2017)
22. Liu, A., Zheng, K., Li, L., Liu, G., Zhao, L., Zhou, X.: Efficient secure similarity computation on encrypted trajectory data. In: 31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13–17, 2015, pp. 66–77 (2015)
23. Liu, X., Xie, Q., Wang, L.: Personalized extended (α, k)-anonymity model for privacy-preserving data publishing. Concurrency and Computation: Practice and Experience **29**(6) (2017)
24. Luper, D., Cameron, D., Miller, J., Arabnia, H.R.: Spatial and temporal target association through semantic analysis and gps data mining. In: Proceedings of the 2007 International Conference on Information & Knowledge Engineering, IKE 2007, June 25–28, 2007, Las Vegas, Nevada, USA, pp. 251–257 (2007)
25. Monreale, A., Trasarti, R., Renso, C., Pedreschi, D., Bogorny, V.: Preserving privacy in semantic-rich trajectories of human mobility. In: Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS, SPRINGL 2010, November 2, 2010, San Jose, California, USA, Proceedings, pp. 47–54 (2010)
26. Naghizade, E., Kulik, L., Tanin, E.: Protection of sensitive trajectory datasets through spatial and temporal exchange. In: Conference on Scientific and Statistical Database Management, SSDBM '14, Aalborg, Denmark, June 30–July 02, 2014, pp. 40:1–40:4 (2014)
27. Nergiz, M.E., Atzori, M., Saygin, Y., Güç, B.: Towards trajectory anonymization: a generalization-based approach. Transactions on Data Privacy **2**(1), 47–75 (2009)
28. Tu, Z., Zhao, K., Xu, F., Li, Y., Su, L., Jin, D.: Beyond k-anonymity: protect your trajectory from semantic attack. In: 14th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON 2017, San Diego, CA, USA, June 12–14, 2017, pp. 1–9 (2017)
29. Xu, T., Cai, Y.: Exploring historical location data for anonymity preservation in location-based services. In: INFOCOM 2008. 27tH IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13–18 April 2008, Phoenix, AZ, USA, pp. 547–555 (2008)
30. Yarovoy, R., Bonchi, F., Lakshmanan, L.V.S., Wang, W.H.: Anonymizing moving objects: how to hide a MOB in a crowd? In: EDBT 2009, 12th International Conference on Extending Database Technology, Saint Petersburg, Russia, March 24-26, 2009, Proceedings, pp. 72–83 (2009)
31. Yurtsever, E., Takeda, K., Miyajima, C.: Traffic trajectory history and drive path generation using GPS data cloud. In: 2015 IEEE Intelligent Vehicles Symposium, IV 2015, Seoul, South Korea, June 28–July 1, 2015, pp. 229–234 (2015)
32. Zheng, Y., Xie, X., Ma, W.: Geolife: a collaborative social networking service among user, location and trajectory. IEEE Data Eng. Bull. **33**(2), 32–39 (2010)