Front.Comput.Sci.
DOI

RESEARCH ARTICLE

Precise Sensitivity Recognizing, Privacy Preserving, Knowledge Graph-based Method for Trajectory Data Publication

Xianxian Li [1,2], **Bing Cai** [2], **Li-e Wang** (✉)[1,2], **Lei Lei**(✉)[3]

1 Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin, China
2 College of Computer Science and Information Engineering, Guangxi Normal University, Guilin, China
3 Nanning Tiancheng Zhiyuan Intellectual Property Service Co. Ltd, Nanning, Guangxi, China

Abstract Trajectory data can be widely collected via mobile devices and published for academic or commercial purposes. In our research, we noted that problems of privacy raised considerable concern among researchers. Most existing privacy protection methods seek to hide the user's accurate location by using perturbations or modifications, to ensure that the information is sufficiently imprecise to prevent re-identification of the published trajectory data; however, this reduces the availability and utility of the published trajectory data. Thus, the main challenge in publishing such datasets is striking a balance between the required privacy protection and data utility. To better balance the two, we propose a precise sensitivity recognition and privacy protection method for trajectory data publication($PSR\&PPM\_KG$ for short); this method constructs knowledge graphs, to automatically distinguish sensitive and non-sensitive information for each user. In this study, we first construct a trajectory knowledge graph from the original trajectory dataset by considering the users' attributes and position tags; then, we design a recognition method to accurately identify sensitive locations on each user's trajectory, using a knowledge graph; finally, we propose a personalized privacy protection method to ensure the privacy of each user's sensitive locations whilst improving the data utility. Experimental results show that our adaptive anonymization model can protect the privacy of users effectively and offer enhanced data utility.

Keywords Knowledge Graph, Trajectory Data Publication, Privacy Protection, Location-based Services, Anonymization, Recognition

Received month dd, yyyy; accepted month dd, yyyy

E-mail: *wanglie@gxnu.edu.cn,lainelei@hotmail.com*

1 Introduction

The emergence of various mobile location devices has produced large quantities of location data and generated numerous location-based service applications, such as in-car navigation devices and mobile phones embedded with global positioning systems. These applications allow users to send their location to a location server and thereby obtain convenient enquiry services. For example, when travelling, individuals can use Baidu maps for navigation services and Twitter for social location services. These applications are typically divided into two categories: online (in which users provide real-time locations) and offline applications. Offline applications are primarily used for data mining and analysis; they can be used to optimize transportation networks, plan road systems, detect hot-spots [3–5], and recognize human behaviour [6, 7] to support business decisions. Thus, these applications often consider only the locations visited with high frequency; these hot-spots or highly frequent roads in the trajectory data are published to facilitate data analysis and aid prediction. However, a user's trajectory data can reflect their interests and preferences, and private data can not be published directly without anonymization. This is because an attacker might infer a user's location information using the spatiotemporal correlation of some of their visited locations [1, 2]; this can compromise user privacy. Pre-publication data anonymization is required for two reasons: on the one hand, data owners can ensure that the published anonymized

data does not disclose the user's private information; on the other hand, the data is thereby permitted a high availability, allowing data analysts to analyze them more accurately. In this study, we restrict our focus to offline applications, and we propose a precise sensitivity recognition and privacy protection method based on knowledge graphs, integrating multiple types of attributes to ensure secure trajectory data publication.

Many anonymization approaches have been proposed for trajectory data publication; however, most of these result in a serious loss of information; the subsequent anonymized data are of low utility. Notably, previous studies have been unable to accurately identify users' sensitive locations, owing to the complex relationships between multiple types of entities; thus, these works have often applied a single unified anonymization mechanism. More specifically, they consider only the trajectories and delete non-frequent locations on each user's trajectory; then, they anonymize the frequent location by constructing an indistinguishable k-anonymous region or by adding noise, to protect privacy for trajectory data publication. However, this method neglects to integrate trajectory information with the attributes of users and the locations' tags; this integration can help identify the different sensitivities of locations and can better distinguish the specific sensitivities different locations have to different users. For example, a doctor visits the hospital every day, but this does not indicate that he/she is sick. If the same pattern was observed in another person who was not a doctor or hospital worker, we can readily infer that he/she has some form of the disease. Furthermore, by combining this information with some background knowledge we might learn more details about the individual; for instance, we might infer the severity of their disease. Moreover, some highly frequented trajectory positions do not indicate private locations or preferences; this can be inferred using the tag of the location. For instance, a user may travel frequently for work; thus, his/her trajectory data will frequently indicate locations or roads relevant to airports or train stations; these locations do not disclose the user's private information.

Thus, designing an adaptive anonymization procedure is crucial to increase data utility whilst maintaining user privacy. However, this requires that we accurately identify sensitive locations for each user; as described earlier, sensitivity varies from place to place and from place to user, which complicates the process. Therefore, we construct a knowledge graph incorporating multiple types of attribute information, and propose the *PSR&PPM_KG* method, to enhance data utility whilst guaranteeing privacy. The main contributions of this study are summarized as follows:

- To automatically identify sensitive relationships between users and locations, we construct a trajectory knowledge graph that integrates user attributes, location tags, and the relationships between them. This is because the relationships between entities in trajectory data are complex and diverse, for which knowledge graphs offer a better representation;
- We design an intelligent recognition algorithm that automatically identifies personalized sensitive location points for each user, using association rules extracted from the knowledge graph. We note that previous works have not successfully dealt with the privacy problems of trajectory data publication: either the highly sensitive information is insufficiently protected or the low-sensitivity information is over-protected. This causes excessive information loss through a unified anonymization standard;
- Aiming to improve the utility of data whilst maintaining privacy, we propose an adaptive anonymization method that processes the sensitive locations identified in the knowledge graph by the aforementioned intelligent recognition algorithm. We minimize information loss more effectively than existing anonymization technologies, whilst also providing sufficient privacy protection.

The remainder of this paper is structured as follows: Section 2 presents a review of the relevant literature, to highlight how our work builds upon state-of-the-art research; Section 3 presents the preliminary concepts and problem definition; Section 4 presents our solution and explains the proposed privacy protection algorithm in detail; Section 5 reports upon our experiments, in which we extensively compared our method against competing ones; finally, Section 6 concludes the paper and suggests future research.

2 Related works

In this section, we review the existing research regarding privacy protection and the task of distinguishing location sensitivities before publishing trajectory data, both of which are directly relevant to our work. Also, we discuss the differences between our method and existing methods.

2.1 Generalization-based methods

Generalization-based methods extract positions from the trajectory data, group these positions into clusters, and use the cluster centres as the points of production for the anonymous region.

The current k-anonymous trajectory technology belongs to this category. K-anonymity [11] is a typical generalization technique that was proposed by Sweeney. The k-anonymity trajectory model was first proposed by Grutester and Grunwald in [12]. In the anonymous trajectory dataset, the probability that a trajectory or location is accurately identifiable is less than $1/k$. To reduce information loss, Abuul et al. [13] proposed a model referred to as (k, m)-anonymous and designed the so-called "Never Walk Alone" method. It applies the Euclidean distance metric, and the time must be synchronized between different trajectories. However, though these trajectories are similar in Euclidean space, they may differ in an actual road network space; this can result in anonymization failure. Harnsamut et al. proposed a look-up table bruteforce algorithm [14] based on the LKC-privacy model, to ensure data privacy and quality when implementing generalization technologies to anonymize trajectory data. In [10], traditional trajectory privacy protection algorithms were used to treat tasks as single-layer problems. To solve complex problems in layers, they proposed a two-level layered granularity model. However, this method still reduces the availability of the published trajectory data. In [16], Geo et al. proposed a personalized anonymous model to select the k-anonymous trajectory set by considering the various privacy and utility requirements of trajectory data in different scenarios. In [20], Zhang et al. employed sequential retention symmetric encryption and k-anonymity to protect users' location privacy. Under this method, an attacker can only perform simple matching and comparison operations. However, if the user consistently uses the same key to encrypt their coordinates in successive queries, the key's security cannot be guaranteed. In [25], Sui et al. proposed a new trajectory anonymity model, in which the degree of correlation between the parking location and user was accurately expressed as $LF\text{-}IUF$. This approach achieved a better trade-off between privacy and utility; however, it defined each user's mobile preferences by access frequency. In [28], a new privacy protection algorithm was proposed, based on frequent-path trajectory data publishing; First, the method removed infrequently visited roads from each trajectory and used a novel method to divide the trajectories into candidate groups; then, they proposed another novel method to identify the most-frequent paths; finally, a representative trajectory was selected to represent all trajectories within the group. This approach only protects the highly frequent roads instead of all the original trajectory data. However, this method suffers from the problem that the frequent roads identified do not necessarily disclose the user's sensitive information.

2.2 Suppression-based methods

This method is used to selectively publish trajectory data. Before the dataset is released, sensitive data and frequently occurring locations are suppressed or removed. This method is relatively simple, though it produces a large information loss and results in low data availability.

In [15], Chen et al. employed the $(K, C)_L$-privacy model to anonymize trajectory data by considering both identity and attribute link attacks thereupon. Furthermore, they proposed an anonymization framework to remove all privacy threats from the trajectory database by using local and global suppression methods. In [21], Komishani et al. proposed a new method for trajectory data publication, based on the concept of personalized privacy. This was the first study to implement a personalized privacy model for trajectory data distributions by combining sensitive attribute generalizations with local trajectory suppression; however, adding noise to high-dimensional trajectory data reduces the data availability. In [17], Huo et al. proposed a novel method to solve the case in which an attacker has as his/her background knowledge the actual location of a mobile user. Therefore, they proposed the so-called "You Can Walk Alone" method; this method protects privacy by generalizing the stopping points on the trajectory; therefore, an attacker can still analyze the movement patterns of the trajectory to obtain background knowledge. In [19], Li et al. proposed a privacy protection algorithm based on segmented clustering. The anonymization results of this method were diverse though not conducive to data analysis. Existing studies of transaction data privacy protection have predominately focused on the single-mode dataset; thus, they cannot be directly implemented to manage the privacy problems of multimodal data integration. Therefore, a k^m-anonymity-ρ-uncertainty privacy model was proposed in [30], to solve the privacy protection problems of transaction data and their integration. They used suppression techniques to eliminate sensitive association rules and solve the leakage problems of sensitive items. In [24], Chen et al. designed a trajectory privacy protection method based on a $3D$ grid division, to reduce information losses during the anonymization of trajectory data.

2.3 Perturbation-based methods

These methods are typically applied to replace real trajectory data with false data, or to add false trajectory data to the original trajectory dataset and thereby protect the real data. This approach protects published trajectory data but results in low data availability.

In [22], Zhao et al. proposed a *S equence R* (*S R*) tree structure based on *R*-trees, to satisfy differential privacy; they applied differential privacy technology to add noise to the location data and non-location-sensitive data, to prevent non-location-sensitive information attacks; In [8], Zhou et al. proposed an anchor generation method based on the diversity of sensitive locations. It selected sensitive places according to the number of visitors and the peak hours, to form a diversity area; then, it used the centroid of this area as the anchor position, thereby increasing the user's location diversity. From this anchor point, an interest point query algorithm was proposed; this produced accurate results using the anchor point, without sending the user's actual location to the location-based service server. They considered that almost all sensitive locations exhibit a high frequency of visits, which is unreasonable according to our aforementioned analysis. Wang et al. [23] proposed a *DPPP* scheme to prevent attackers from obtaining individuals' sensitive attributes through attribute link, record link, or similarity attacks. In [9], Dai et al. proposed an effective, personalized, trajectory privacy protection method. The main idea was to mark the semantic attributes of all sampling points on the trajectory, construct the corresponding classification tree, and extract the sensitive stay points. However, its definition of "stay point" is not sufficiently precise, and frequently visited places are often regarded as stay points.

2.4 Sensitive location recognition

In privacy protection research for trajectory data publishing, many works have sought to distinguish the location data on the trajectory; however, they have only considered sensitivity from a single perspective when distinguishing the location. This means that the sensitive location data identified is not sufficiently accurate, which reduces the anonymous trajectory data's availability after processing. In [29], to protect respondents' places of residence (collected in a travel survey before data publication), Godwin et al. directly treated the trajectory data corresponding to a place of residence as sensitive. However, this assumption is unrealistic; for the same place, different users may have different sensitivities, and the sensitivity of a place closely depends on the specific user; In [33], Yang et al. proposed a trajectory clustering algorithm (*TAD*) based on spatial-temporal density analysis. It considered the trajectory characteristics from both time and space perspective and constructed a new density function, which can more accurately distinguish the different types of points on the trajectory. In article [8], Zhou et al. defined sensitive locations by considering their user's access frequency and peak access time. However, they only considered the location itself without intergrating other associated information. thus, their method differs from that of the present study.

3 Preliminaries

In this section, we introduce a set of relevant concepts, including knowledge graphs and trajectory privacy protection. To facilitate further explanation, the specific problem definition and terminology used in our research are listed below.

3.1 Notation

Definition 1(Trajectory) Given an ordered set of timestamp positions $T_i = \{D_i, (x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_i, y_i, t_i), ..., (x_n, y_n, t_n)\}$ with, where $1 \leq i < n$ and $t_i < t_i + 1$, n is the number of sampling points on the trajectory, D_i is the unique identifier of the trajectory relative to a single user. Here, x_n and y_n represents the latitude and longitude of the position, and t_i represents a timestamp, then (x_n, y_n, t_n) indicates that the latitude and longitude coordinates of the moving object position at the time stamp t_n. We called T_i is a trajectory of a user.

Definition 2 (Trajectory database) Let $TS = \{T_1, T_2, T_3, ..., T_i, ..., T_l\}$ be a set of trajectories, where l be the number of users and T_i be the i-th trajectory in the trajectory dataset. Then we called TS as a trajectory database.

Definition 3 (Road sequence of trajectory) Given a trajectory T_i=$\{D_i, (x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_m, y_m, t_m)\}$ with m road numbers in chronological order, and r_i is the road number corresponding to the latitude and longitude of the location, and it contains many locations. If $(x_i, y_i) \in r_i$ and $r_1 \cap r_2 ... \cap r_m = \varnothing$, then we called the road sequence of a trajectory T_i as: $R_i = \{D_i, (r_1, t_1), (r_2, t_2), ..., (r_m, t_m)\}$.

We give an example to further explain Definition 3. Assuming that the original trajectory of Alice is T_m=$\{D_m, (x_1, y_1, t_1), (x_2, y_2, t_2), (x_3, y_3, t_3), (x_4, y_4, t_4), (x_5, y_5, t_5), (x_6, y_6, t_6), (x_7, y_7, t_7)\}$, according to Fig. 1, we can transfer Alice's road sequence of trajectory $R_m = \{D_m, (r_{17}, t_1), (r_1, t_2), (r_{10}, t_3), (r_{12}, t_4), (r_6, t_5), (r_5, t_6), (r_{15}, t_7)\}$.

Definition 4 (Frequent road) A road is frequented if and

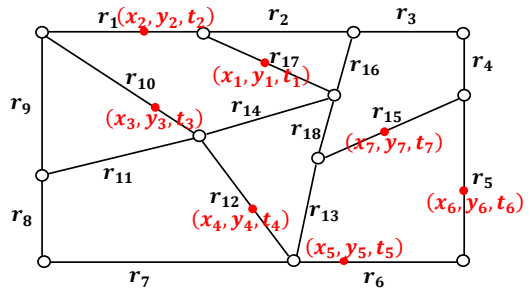


Fig. 1. Road sequence of trajectory.

only if the number of one-way moving objects on the road is not less than θ, where θ denotes a threshold and is referred to as the number of moving objects frequenting the road.

We give an example to further explain Definition 4. As shown in Fig. 2, assuming that there are four users u_1, u_2, u_3 and u_4, and their trajectory are $e \to a \to c \to d$, $f \to a \to b \to d$, $g \to a \to b \to d$ and $h \to a \to c \to d$ respectively. After the map mapping operation, their road sequences of trajectory are $r_{11} \to r_{14} \to r_{18} \to r_{15}$, $r_{10} \to r_{14} \to r_{19} \to r_{15}$, $r_{21} \to r_{14} \to r_{19} \to r_{15}$ and $r_{12} \to r_{14} \to r_{18} \to r_{15}$. And assuming the threshold value θ=2, since the number of mobile users passing through road r_{14} and r_{18} is both 4, we can get that road r_{14} and r_{15} are frequent roads according to Definition 4.

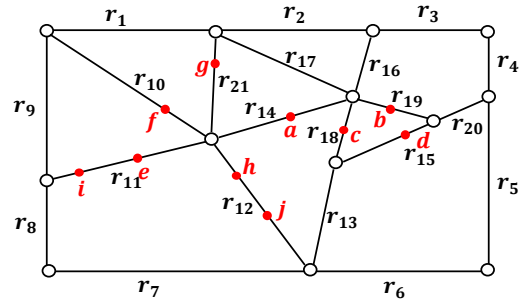


Fig. 2. An example of Frequent road.

Definition 5 (Knowledge graph)A knowledge graph is represented by a network of knowledge bases, can be formalized as a set of triples $G = \{(u, rel, l)\} \subseteq \varepsilon \times R \times \varepsilon$, where $rel \in R$ denotes a relationship between the two entities $u, l \in \varepsilon$, referred to as the *user* and *location* of the triple.

Taking an example for further explanations, we give a knowledge graph of trajectory as shown in Fig. 3, where nodes represent entities and edges represent relationships therebetween. Besides, these nodes can also contain rich information of entities and their attributes as labels. As shown in Fig. 3, the location entities have labels such as *latitude* and *longitude* coordinates, *the total number of visits*, and *the corresponding road of the location*. And the user entities can have labels such as $profession$, age and so on.

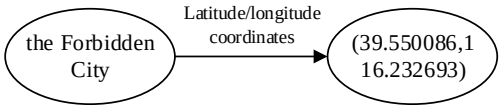


Fig. 3. Knowledge graph triple.

Definition 6 (Sensitive location recognition) A sensitive location refers to a location at which users are reluctant to disclose their private information to the public. Our objective is to identify sensitive locations using the following four conditions: (1)Recognition based on whether the user attributes and the tags of the visited location correspond. (2)Recognition based on whether the visiting times and tags of the visited location are reasonable. (3)Recognition based on whether the frequency of visits to a location is normal. (4)Recognition based on whether the visiting time is unusual.

Definition 7 (Adaptive parameter) δ_{sr} is an adaptive parameter used to adjust the user privacy tolerance p_r; it is defined as follows:

$$\delta_{sr} = \varepsilon_{sr} * F_{sr}, F_{sr} = \frac{sup(sr)}{|N|} \tag{1}$$

Here, sr represents the road containing the sensitive location, $sup(sr)$ denotes the number of roads sr, and $|N|$ denotes the total number of roads in the trajectory dataset. $\varepsilon_{sr}(\varepsilon_{sr} > 0)$ is the so-called sensitivity factor of sr; it is used to tune the value of δ_{sr} in combination with F_{sr}. If F_{sr} is higher, a larger ε_{sr} is chosen to prevent privacy disclosure. ε_{sr} is adjusted using the sensitivity of the road associated with the sensitive location and the user's protection requirements.

Definition 8 (Adaptive user privacy tolerance) The adaptive user privacy tolerance is expressed as $p_r = \rho + \delta_{sr}(1 > p_r > 0)$, where δ_{sr} is the aforementioned adaptive parameter; ρ is the threshold and is set uniformly. For trajectory data, the anonymous dataset satisfies our privacy model only if the success probability for all trajectories containing sensitive positions to be inferred by an attacker is below p_r.

Definition 9 (Privacy risk) The likelihood that attacker v will infer the sensitive location $s\text{-}loc_j$ on the trajectory (based on his/her knowledge q) is defined as follows:

$$p(q \to s\text{-}loc_j) = \frac{n(s\text{-}loc_j, q, TS)}{|N(q, TS)|} \tag{2}$$

where $n(s\text{-}loc_j, q, TS)$ represents the number of trajectories containing both q and $s\text{-}loc_j$ in TS, and $|N(q, TS)|$ represents the number of trajectories containing q in TS; q represents

Table 1. Trajectory Dataset TS

Numbering t_{id}	*User's Profession*	*Trajectory*
t_1	doctor	$r_1 \rightarrow \boldsymbol{s_1} \rightarrow r_2$
t_2	teacher	$r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow \boldsymbol{s_3}$
t_3	student	$r_1 \rightarrow \boldsymbol{s_2} \rightarrow r_2$
t_4	doctor	$r_1 \rightarrow \boldsymbol{s_2} \rightarrow r_3$
t_5	programmer	$r_1 \rightarrow r_3 \rightarrow \boldsymbol{s_1}$
t_6	teacher	$r_2 \rightarrow r_3 \rightarrow \boldsymbol{s_1}$
t_7	student	$r_2 \rightarrow r_3 \rightarrow \boldsymbol{s_2}$
t_8	writer	$r_1 \rightarrow r_3 \rightarrow \boldsymbol{s_2} \rightarrow \boldsymbol{s_3}$

Table 2. Anonymous Dataset TS

Numbering t_{id}	*User's Profession*	*Trajectory*
t_1	doctor	$r_1 \rightarrow \boldsymbol{s_1} \rightarrow r_2$
t_2	teacher	$r_1 \rightarrow r_2 \rightarrow r_3$
t_3	student	$r_1 \rightarrow \boldsymbol{s_2} \rightarrow r_2$
t_4	doctor	$r_1 \rightarrow \boldsymbol{s_2} \rightarrow r_3$
t_5	programmer	$r_3 \rightarrow \boldsymbol{s_1}$
t_6	teacher	$r_3 \rightarrow \boldsymbol{s_1}$
t_7	student	r_3
t_8	writer	$r_3 \rightarrow \boldsymbol{s_2}$

the background knowledge of the attacker and is represented by $q = \{r_1, r_2, ..., r_j\}$, where $j \leq m$.

3.2 **Problem definition**

The existing methods of local suppression and adding false trajectory data can effectively prevent identity link attacks without considering attributes information. However, there are two problems with existing methods.

First, existing methods cannot resist attribute link attacks effectively. In practical application scenarios, it is necessary to integrate data from a variety of fields for data analysis. Taking *cross-domain recommendations* as an example, we usually need to integrate social network data for more accurate trajectory recommendation, that is to obtain the user attributes outside of the trajectory data as background knowledge. Yet, existing methods cannot provide sufficient privacy protection for users when integrating with multiple types of background knowledge, such as user attributes and location tags. To better illustrate the problem, we give a specific example below. **Table 1** includes the original trajectory data of 8 users, and **Table 2** is anonymous result by adopting the method [26] without considering users' attributes, suppose $QID = \{r_1, r_2, r_3\}$, $Set_{sensitive} = \{\boldsymbol{s_1}, \boldsymbol{s_2}\}$, that is: the bold part in the table represents the sensitive position. It satisfies the p_r-uncertainty privacy model under the condition of $p_r = \frac{1}{2}$, that is, when the attacker knows any information in the QID set, he can not infer that a user has been to the sensitive location $\boldsymbol{s_1}$ or $\boldsymbol{s_2}$ with a probability greater than $\frac{1}{2}$. However, as mentioned above, the attacker can obtain the background knowledge including the users' attributes and trajectory information from other external sources. For example, if a neighbour knows that Mary is a doctor and has visited $\{r_1, r_2\}$, then combining with the anonymous data published in **Table 2**, the attacker can be uniquely certain that Mary must have also visited sensitive location $\boldsymbol{s_1}$, thus rising a privacy breach problem called as multi-attributes background attacks.

Second, existing methods ignore the problem of adding unnecessary noise. Existing works have proposed some definitions of sensitive location, but they only consider the location itself or the frequency of location access [8], etc. For example, some works have considered hospitals and prisons as sensitive locations, or they have directly specified the high-frequency locations (i.e., residences) as sensitive [29]. According to these sensitive location definitions, the sensitive locations on all user trajectories identified are the same. We note that different people have different sensitivity to the same position. The sensitive location of a user is generally related to the location labels and the attributes of the user. For example, hospital $\boldsymbol{s_2}$ is usually regarded as a sensitive location, because most people think that people who go to hospitals usually have health problems. But for the doctor t_4, the hospital is a place to work and is not sensitive. Thus, the same location has a different sensitivity for different users. If only considering access time and frequency, all sensitive locations on the user's trajectory cannot be accurately identified. Assume that a user may need frequent business trips for work reasons, so the location information related to the airport or railway station will frequently appear in his trajectory. But in fact, these locations do not refer to users' private information. However, most of the existing methods ignore the problem, which treats equally without discrimination will incur unnecessary noise addition and reduce the utility of the data. Therefore, before anonymous processing of the trajectory data, it is necessary to accurately identify the sensitive location for different users, which is conducive to subsequent adaptive anonymous processing operations.

To solve the above two problems, we propose a trajectory data publishing method based on knowledge graph and privacy

protection aiming at precise sensitivity recognizing for more fine-grained noise additions in this paper. Considering other auxiliary information (user's attributes and location tags, for instance), this method uses association rules of knowledge graph to automatically identify sensitive locations and achieves the best compromise between data utility and privacy security.

3.3 **Privacy goal**

Given the original trajectory dataset TS, $TS^{'}$ represents an anonymized version of TS. If $\forall t^{'} \in TS^{'}$, the attacker v cannot infer the user's sensitive location $s\text{-}loc_j$ with a probability exceeding p_r, where p_r is the adaptive user privacy tolerance and $s\text{-}loc_j \in t^{'} \wedge s\text{-}loc_j \notin q$, then $TS^{'}$ is considered safe and can be published; otherwise, $TS^{'}$ is unsafe and cannot be published. Therefore, our privacy target is this: even if the attacker knows part of the user's real location data, they will be unable to infer other sensitive location information with a probability higher than p_r. With this, we ensure that the anonymized data has a high utility value, whilst also keeping the user's sensitive information safe.

4 Our solution

4.1 Overall framework of the scheme

In daily life, the trajectory data of users often contain certain location data associated with the sensitive information of users. Once these location data are known by an attacker, the user's privacy will be threatened. Our solution is to construct a trajectory knowledge graph, by considering various user attributes and location tags; then, we design an intelligent identification method to accurately identify the sensitive locations in each user's trajectory data; finally, for sensitive locations identified by the intelligent recognition algorithm, we propose a personalized adaptive anonymity model, which not only improves the data's utility but also protects the privacy of each user's sensitive locations. The scheme is divided into three major parts:

I. We preprocess the trajectory data by mapping the locations of the original trajectory dataset onto the map, to obtain the actual locations and their tags; then, we extract the users, locations, and the relationships between them to construct a trajectory knowledge graph, which can visualize a complicated network clearly.

II. Based on the trajectory knowledge graph, we design a recognition algorithm to intelligently identify users' sensitive locations, by constructing a sensitive rule library as described in Definition 6.

III. To strike a balance between the published data's availability and information security, we propose an adaptive anonymity model to manage the sensitive locations identified in Step II, and we calculate the adaptive anonymous factor for different users; this is because the one location may have different sensitivities for different users. For convenience, we summarize the symbols used in the algorithm in Table 3.

4.2 Construction of trajectory knowledge graph

We perform corresponding preprocessing on the original trajectory dataset. Using the open platform of Google Maps, and according to the application programming interface it provides, the position data of each trajectory in the original trajectory dataset are mapped. Thus, we obtain a collection of trajectory road sequences; using the definition for frequent roads, we delete the infrequent roads in the set, to obtain the pre-processed trajectory dataset $TS^{'}_1$. Then, we extract the node data and relational data between users and locations from dataset $TS^{'}_1$. We select three attributes like the location characteristics: the latitude and longitude coordinates, the total number of visits, and the road that the location is on. And for illustration purposes, we choose profession attributes as the user characteristics; the relationship between the user and the position on the trajectory is marked as an access relationship. This access relationship features two attributes; namely, the time at which the user visits the location and the number of times they visit it.

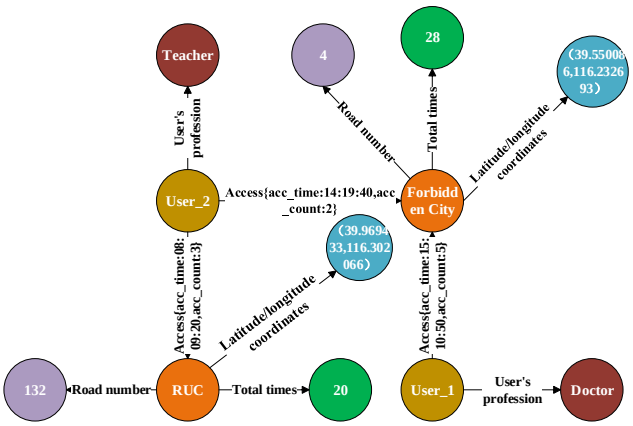


Fig. 4. Pattern diagram example

The pre-processed data is used to construct a pattern diagram of the trajectory knowledge graph, including the location name, user, location-related attributes, user-related attributes, and access relationship between user and location,

Table 3. Notations used in the algorithm

Symbol	*Description*
$TS_1^{'}$	Trajectory dataset after preprocessing
$TS^{'}$	Trajectory dataset after anonymous processing
sp_time	Special time period [24:00-6:00]
cm_time	Regular time period
$User\_Profession$	A list of user's profession
$Sensitive\_Location$	A list of user's sensitive locations
$Location\_data$	A list of user's locations
$Relation\_data$	A list of the time and number of times the user visited the location
loc	The location on the trajectory
tra_i	tra_i represents a trajectory in $TS_1^{'}$
q	Attacker's knowledge
p_r	Adaptive privacy tolerance

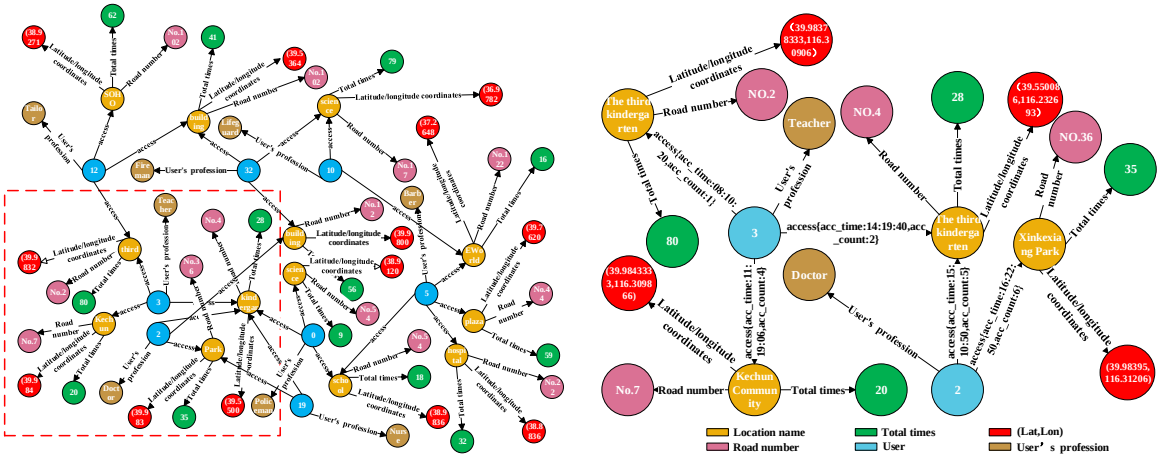
as shown in Fig. 4. Algorithm 1, which performs the detailed process of trajectory knowledge graph construction, is described as follows: Line 1 of the algorithm extracts user, location, and relationship data from $TS_1^{'}$; Lines 2–6 delete the infrequent roads from *Location_data*; and Lines 7–13 are used to create nodes with different labels for the extracted data and the relationships between nodes. We create a user node marked "user" a location node marked "location" and a property relationship between them. Finally, the corresponding network structure is constructed according to the *relation*. The pre-processed data and designed pattern diagram are used to construct a trajectory knowledge graph. Fig. 5 shows an example of a well-constructed trajectory knowledge graph, in which Fig. 5(a) is part of the network graph of the well-constructed trajectory knowledge graph, and Fig. 5(b) is an enlarged view of the red dotted frame in Fig. 5(a). We have marked the meaning of different colours in Fig. 5(b). The yellow node represents the name of the location, the green node represents the total number of times the location has been visited, the red node represents the latitude and longitude coordinates of the location, and the pink node represents the road number where the location is located, the blue node represents the user, and the yellow-grey node represents the user's profession.

Algorithm 1 *Constructing a trajectory knowledge graph*

Input: $TS_1^{'}$
Output: *trajectory knowledge graph*
1: $User\_data, Location\_data, Relation\_data \leftarrow TS_1^{'}$ ▷Extract user, location, and relational data from $TS_1^{'}$
2: **for** each $loc \in Location\_data$ **do** ▷Remove infrequent roads from *Location_data*
3: **if** $loc < \theta$ **then** ▷θ is the threshold value used to determine frequent road
4: delete *loc*
5: **end if**
6: **end for**
7: **for** each $user \in User\_data$ and $loc \in Location\_data$ **do** ▷Create entity nodes and relationships
8: $User\_node \leftarrow Node('user', name = user)$
9: $Location\_node \leftarrow Node('location', name = location)$
10: $properties \leftarrow Relation\_data$ ▷Extract attribute information from *Relation_data*
11: $relation \leftarrow Relationship(User\_node, 'access', Location\_node, **properties)$
12: $graph.create(relation)$
13: **end for**
14: **return** *trajectory knowledge graph*

4.3 Recognition algorithm based on knowledge graphs

We construct the trajectory knowledge graph in Section 4.2. To accurately identify the sensitive locations on each user's trajectory, we design an intelligent recognition algorithm according to Definition 6, to query the constructed trajectory knowledge graph and recognize sensitive relationships. Here, we divide the period into "special" and "regular" periods. The special time covers the interval from 24:00 to 6:00 am; other times are regarded as the regular time. These rules are primar-



(a) Part of the network graph of the trajectory knowledge graph (b) A partially enlarged view

Fig. 5. Examples of trajectory knowledge graph

ily employed whenever a user visits a location during the special time and that location is not related to his/her profession. These locations are regarded as sensitive. For example, if $u1$ is a doctor or hospital worker and visits the hospital during the period [24:00–6:00], it indicates that he/she is working the night shift. Therefore, the location of the hospital is not a sensitive location for $u1$. However, if $u2$'s profession is not that of a doctor or hospital staff, and he/she visits the hospital at any time, it indicates that the user may have health problems. Thus, the hospital is a sensitive location for $u2$. Besides, the number of times each user visits each location within the regular time is calculated, along with the average number of times that location is visited by the user during this period. If the number of visits exceeds the average number of visits, it indicates that most people do not visit the place, though the user visits more frequently. This place may have some special significance for the user; therefore, its location is sensitive to them. For example, a user may have a chronic disease and be required to take a certain drug over a long time; therefore, the pharmacy location may appear more frequently than others on his/her trajectory and is a sensitive location for them. In Algorithm 2, line 1 looks for all locations occurring in special and regular periods in the trajectory knowledge graph according to *sp_time* and *cm_time*, which are represented by the lists *Special_Location* and *Common_Location*, respectively. Lines 2–6 remove locations relating to the user's profession from *Special_Location* and *Common_Location*. Lines 7–13 traverse each location in *Common_Location* and calculate the average number of times each location is visited; then, the algorithm compares this with the number of times the user visits each location, and it adds locations with visit numbers exceeding the average to *Sensitive_Location*; Lines 14–15 return a list of each user's sensitive locations.

Algorithm 2 *Recognition algorithm based on knowledge graph*

Input: *trajectory graph*, *sp_time*, *cm_time*, *User_Profession*
Output: *Sensitive_Loction*

1: *Special_Location*, *Commom_Location*←*trajectory graph*
2: **for** each *profession* ∈ *User_Profession* **do**
3: **if** profession∈*Special_Location*||*Common_Location* **then**
4: delete *profession* ▷Delete locations related to user professions
5: **end if**
6: **end for**
7: **for** each *loc* ∈ *Common_Location* **do**
8: *number* ← *count*(*loc*)
9: *ave_number* ← *average*(*loc*) ▷Calculate the average number of visits to a location
10: **if** *number* > *ave_number* **then**
11: *Sensitive_Location* ← *loc*
12: **end if**
13: **end for**
14: *Sensitive_Location* ← *Special_Location*
15: **return** *Sensitive_Location*

4.4 Privacy protection for sensitive locations

According to the previous two subsections, we can obtain the sensitive location of each user's trajectory. We first calculate the probability that the attacker can infer the user's sensitive location based on the knowledge q possessed by the attacker, and then we compare this probability with the user's adaptive privacy threshold p_r. If the probability is greater than or equal

Algorithm 3 *Adaptive anonymization algorithm*

Input: $TS^{'}_1, q$
Output: $TS^{'}$

for each $s\text{-}loc_j \in TS^{'}_1$ **do**
  $p(q \rightarrow s\text{-}loc_j) = \frac{n(s\text{-}loc_j, q, TS^{'}_1)}{|N(q, TS^{'}_1)|}$ ▷Calculate the probability of the attacker guessing the position $s\text{-}loc_j$
  $F_r = \frac{sup(s\text{-}loc_j)}{|N(TS^{'}_1)|}$
  according to $p_r = \rho + \delta_r$ compute value p_r
  if $p(q \rightarrow s\text{-}loc_j) \geq p_r$ **then**
    $location\_list \leftarrow s\text{-}loc_j$
  end if
end for
for each $s\text{-}loc_j \in location\_list$ **do**
  if $s\text{-}loc_j \in tra_i$ and $tra_i \in TS^{'}_1$
    delete $s\text{-}loc_j$
  else
    $N(q \rightarrow s\text{-}loc_j) = |N(q \cup s\text{-}loc_j, TS^{'}_1)| - \lfloor |N(q, TS^{'}_1)| * p_r \rfloor$
    randomly select $N(q \rightarrow s\text{-}loc_j)$ trajectories to delete $s\text{-}loc_j$
  end if then
end for
return $TS^{'}$

to p_r, we control the identification accuracy of sensitive locations within the range allowed by our privacy model through local elimination, to protect the privacy security of the user's sensitive location. We assume that the attacker knows some of the true location data in $TS^{'}_1$. According to the definition of the knowledge possessed by the attacker, we can obtain the attacker's knowledge on each trajectory in the dataset. We use Formula 2 to calculate the probability that the knowledge possessed by the attacker can be used to infer the other sensitive locations on the trajectory. Then, we compare the probability $p(q \rightarrow s\text{-}loc_j)$ with the size of the adaptive privacy tolerance parameter p_r. If $p(q \rightarrow s\text{-}loc_j)$ is greater than p_r, then the location $s\text{-}loc_j$ is placed into a list. For each sensitive location $s\text{-}loc_j$ in the list, if only one trajectory in $TS^{'}_1$ contains the location $s\text{-}loc_j$, then location $s\text{-}loc_j$ is deleted from the trajectory. Otherwise, we calculate the number of sensitive location $s\text{-}loc_j$ to be deleted by Formula 3, randomly select $N(q \rightarrow s\text{-}loc_j)$ trajectories containing $s\text{-}loc_j$ from $TS^{'}_1$, and delete location $s\text{-}loc_j$ from these trajectories. Formula 3 is expressed as follows:

$$N(q \rightarrow s\text{-}loc_j) = |N(q \cup s\text{-}loc_j, TS^{'}_1)| - \lfloor |N(q, TS^{'}_1)| * p_r \rfloor \quad (3)$$

Here, $|N(q \cup s\text{-}loc_j, TS^{'}_1)|$ represents the number of trajectories in $TS^{'}_1$ that contain both the current record q and the current sensitive location $s\text{-}loc_j$, $|N(q, TS^{'}_1)|$ represents the number of trajectories containing the current record q, and p_r represents the adaptive privacy tolerance. The detailed process of the adaptive anonymization algorithm is shown in Algorithm 3.

4.5 Complexity analysis

Our algorithm consists of three parts: (1) Construction of trajectory knowledge graph, (2) Recognition algorithm based on knowledge graphs, (3) Privacy protection for sensitive locations. For the first part, the main overhead is to delete infrequent roads and build nodes. To delete infrequent roads from $TS^{'}_1$, we must traverse the roads included all trajectories, then we perform the same operation when constructing nodes. Assuming that $TS^{'}_1$ contains n trajectories and the maximum number of roads in the trajectory is m, given a constant m ($m \ll n$), the total number of roads is $n * m$, then the complexity of deleting infrequent roads is $O(n * m)$, and the time complexity of building nodes is $O(n+n*m)$, so the total of the first part is $O(n)$. The main overhead of the second part is to calculate the average number of each location visited, which is $O(n)$. The main overhead of the third part is to calculate the probability of the attacker guessing the sensitive locations with the background knowledge q. Assuming that the total number of sensitive locations is num_s, the time complexity is $O(num_s)$. As the number of trajectories increasing, the number of sensitive locations increases, the time complexity of the third part is $O(n)$. In summary, the total time complexity of our scheme is $O(n) + O(n) + O(n)$, which is $O(n)$.

4.6 Privacy analysis

In this paper, we consider two attack models and analyze how our privacy preserving model satisfies the privacy objective. (1) Identity linkage attack: because the attacker possesses part of the user's information as well as their identity information, they can launch an identity linkage attack using this local information to infer the user's identity. (2) Attribute linkage attack: using the local information of the user, the attacker can initiate an attribute linkage attack on the user's quasi-identifier, thereby deducing other user attribute information.

Theorem Our method can satisfy ρ-uncertainty and withstand both types of attacks.

Proof. Our work adopts ρ-uncertainty conditions for the anonymized data of multiple sensitive location types; this provides a similar privacy protection intensity to k-anonymity conditions. When the attacker has certain background knowledge, the ρ-uncertainty conditions can ensure that the prob-

ability of a successful attack is lower than ρ. Using k-anonymity conditions, the same privacy protection effects can be achieved when $k = \frac{1}{\rho}$. For the example in Section 3.2, the process of anonymization is as follows. Assuming that Mary's privacy threshold is $p_r = \frac{1}{3}$ and her sensitive position is s_2, we know that the probability of the attacker inferring the sensitive location s_2 based on its knowledge $q = \{r_1, r_2\}$ is $\frac{1}{2}$ greater than p_r. Therefore, Formula 3 is adopted to calculate the number of sensitive positions s_2 to be deleted; hence, $N(\{r_1, r_2\} \rightarrow s_2) = |N(\{r_1, r_2\} \cup s_2, TS)| - \lfloor |N(\{r_1, r_2\}, TS)| * p_r \rfloor = 2 - \lfloor 4 * \frac{1}{3} \rfloor = 1$. However, of the four trajectories t_1, t_2, t_3, and t_4, the only two trajectories including s_2 are t_3 and t_4, so one of these trajectories is randomly selected and the sensitive position s_2 is deleted. At this point, the probability of the attacker inferring the sensitive location based on knowledge q is $\frac{1}{4}$ for s_1, $\frac{1}{4}$ for s_2, and $\frac{1}{4}$ for s_3; these are all below the customized privacy threshold p_r. After our anonymization processing, the attacker cannot accurately determine which trajectory is Mary's using the background knowledge, nor can he/she infer the sensitive locations on the user trajectory with a probability higher than p_r. Thus, the anonymized data generated by our method can resist the two types of attacks whilst protecting user privacy.□

5 Experiments

To verify the performance of the proposed algorithm, this study primarily evaluates the availability of the published data. We compare our *PSR&PPM_KG* algorithm with three methods: (1) *TOPF* described in [28]; (2) *Prefix*, proposed by [31]; (3) *ICBA*, proposed by [32]. In the experiment, we calculated the average error rate and standard deviation between the anonymized dataset and the original, comparing the results with the three algorithms to illustrate the availability of the data after anonymization. We also measured the number of frequent-visit patterns remaining after anonymization, by using F-measure.

Generally, experimental data can be associated with other data to obtain relevant background information, such as social network data, business data, etc. Therefore, in our experiments, we use the dataset generating by the Thomas Brinkhoff moving object generator [27] and obtain users attributes from in the classic Movielens dataset, which includes 21 types of users' professions. The object generator generated the location data of 5000, 25000, 50000, 75000, and 100, 000 mobile objects. Then we mapped the longitude and latitude coordinates of the locations on the trajectory to the map to obtain the meaning of each position through the API interface of Google map. All algorithms in the experiment were implemented in the Python language. The experimental environment was a Windows 7 operating system, the memory space was 8GB, and the CPU processor consisted of two 2.80 GHz Intel (R) Core(TM)i7-7700HQ. To ensure accurate measurements, all experiments took the average of the results over ten iterations. In the experiment, we used three commonly used metrics to evaluate the trajectory data, namely Average error rate, Standard deviation and F-measure. Among them, Average error rate and Standard deviation are used to evaluate the difference between anonymous data and original data. The smaller the values are, the smaller the difference between the original trajectory dataset and the anonymous trajectory dataset, the less information loss of the trajectory data, and the higher the availability of the released trajectory data. F-measure is used to calculate the number of frequent patterns in the anonymous trajectory dataset. The larger F-measure value is, the less frequent roads are deleted, the less information loss of trajectory data is, and the higher availability of released trajectory data is. The metrics of our experiment are defined as follows:

(1) Average error rate

The average error rate was used to evaluate the difference between the anonymous dataset and the original one. Suppose N represents the number of roads, and r_j represents the j_th road. Let ori_{r_j} denote the number of roads in the original dataset and ano_{r_j} denote the number of roads in the anonymized one. The average error rate is expressed as follows:

$$E = \frac{1}{N}\sum_{j=1}^{N} E_j = \frac{1}{N}\sum_{j=1}^{N} \frac{|ano_{r_j} - ori_{r_j}|}{ori_{r_j}} \tag{4}$$

(2) Standard deviation

The average difference between ano_{r_j} and ori_{r_j} is determined by the error function E. The smaller the standard deviation, the better the anonymity of each road, and the closer E is to the average error rate. The standard deviation is expressed as follows:

$$\sigma = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(E_j - E)^2} \tag{5}$$

(3) F-measure

We use F-measure to evaluate the number of frequent-visit patterns remaining after anonymous processing. Let *po* be the trajectory set in the original trajectory dataset, *pa* be the trajectory set in the anonymous dataset, and *Nm* be the number of trajectories matching the original dataset in the anonymous

dataset. No and Na represent the total number of trajectories in the original and anonymous trajectory datasets, respectively. Simultaneously, $Precision = \frac{Nm}{No}, Recall = \frac{Nm}{Na}$. The F-measure is calculated as follows:

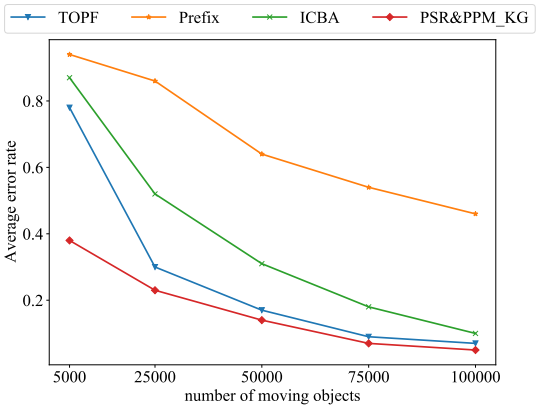
$$F(po, pa) = 2 * (\frac{Precision * Recall}{Precision + Recall}) \quad (6)$$

The parameters used in the experiment are as follows:

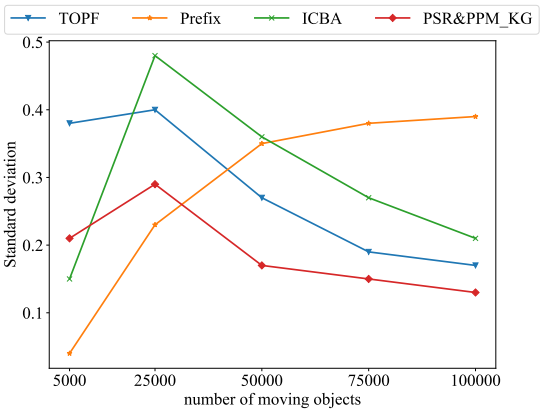
(1) The selection of threshold θ is a critical task that affects the accuracy of anonymization. Threshold θ determines whether a road sequence of trajectory is the most frequent in the trajectory dataset. If a low threshold is used, the road sequence of trajectory may be determined by only a few roads in the sequence. And the low threshold also leads to the low diversity of the road sequence of trajectory, which increases the average error rate. However, if a high threshold is selected, a large amount of data in the trajectory dataset needs to be deleted, which will reduce the utility of the data. Based on the results of multiple experiments, we set the threshold for judging infrequently visited roads to two, because this achieved the optimal results.

(2) In the above three methods, the larger the value of k, the better the user's privacy. Here, we use the user's privacy tolerance, for which smaller values indicate superior user privacy. To compare with the three algorithms mentioned above, we here set the privacy tolerance p_r as the reciprocal of the k values featured there: $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{6}$, $\frac{1}{8}$, and $\frac{1}{10}$.

(3) The adaptive anonymous factor in the experiment, we set the range from 0 to 1 without including 0.



(a) Average error rate



(b) Standard deviation

Fig. 6. Effect of dataset sizes

5.1 Effect of dataset sizes

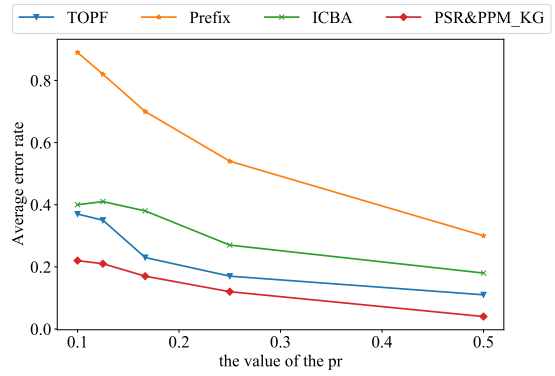
We first compared the performance of our $PSR\&PPM\_KG$ method with $TOPF$, $Prefix$ and $ICBA$, changing the number of moving objects from 5000 to 100,000 by varying the size of the dataset. Fig. 6(a) shows the average error rate of the overall anonymization results obtained from the two methods. Notably, given a different number of moving objects, our $PSR\&PPM\_KG$ achieves a lower average error rate than $TOPF$, $Prefix$ and $ICBA$. When the number of moving objects is small (e.g., 5000), the average error rate of our $PSR\&PPM\_KG$ method is significantly smaller than that of $TOPF$; this is because each road appears infrequently and there are fewer roads upon which the attacker's knowledge can be used to predict sensitive locations. Thus, the number of roads that must be deleted to meet the user's privacy tolerance is lower, and the average error rate is relatively low. In the $TOPF$ method, when the number of moving objects is small, the anonymization process has a significant impact on the average error rate; this is because the number of objects on each road is small, even if the trajectories of the moving objects vary only minimally. As the number of moving objects increases from 5000 to 100,000, the average error rate of $PSR\&PPM\_KG$ decreases continuously and is consistently lower than that of $TOPF$, $Prefix$ and $ICBA$. This is because the $Prefix$ and $ICBA$ methods do not differentiate the sensitivity of the location and are handled uniformly during the anonymous processing.

Fig. 6(b) shows the standard deviations of the four methods. A very low standard deviation indicates that the anonymity quality of each road is high and approximately equal to the average error rate. As can be seen from the figure, the standard deviation of the $PSR\&PPM\_KG$ method is low in most cases, because the probability that the attacker guesses other roads that include the user's sensitive locations (through the knowledge that he/she already controls) is less than the user's privacy tolerance. Therefore, the number of roads that contain the user's sensitive locations and must be deleted during

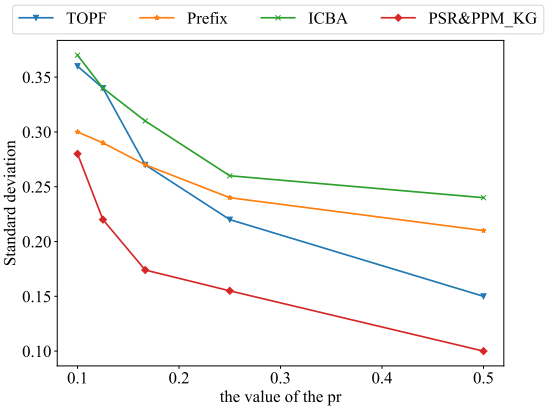
the anonymization process is small, resulting in a decrease in the standard deviation as the number of moving objects increases. This indicates not only that the anonymous results of our $PSR\&PPM\_KG$ method have lower error rates, but that roads containing the user's sensitive locations are well protected.

5.2 Effect of user privacy tolerance p_r

This set of experiments reflects the performance of the four algorithms for different p_r values. Fig. 7(a) shows the average error rates of the anonymous datasets retrieved from both methods. From the figure, we can see that the average error rates of both methods decrease with an increase of p_r; however, the average error rate of our $PSR\&PPM\_KG$ method is lower than that of $TOPF$, $Prefix$ and $ICBA$. This is because the higher the p_r value, the more locations the attacker knows. The probability of other sensitive locations' roads being inferred from the attacker's knowledge is generally lower than the user's privacy tolerance p_r; therefore, the number of sensitive positions to be deleted during anonymization will decrease, and the average error rate decreases with the increase of p_r. We also measured the standard deviations of the anonymized results obtained from the two methods. In Fig. 7(b), the anonymized results produced by our $PSR\&PPM\_KG$ method can be seen to have a lower standard deviation than those of $TOPF$, $Prefix$ and $ICBA$, which indicates that our method reduces the data loss after anonymizing the roads containing sensitive locations. And the standard deviation of these anonymized results is significantly lower than that of our method.



(a) Average error rate



(b) Standard deviation

Fig. 7. Effect of user privacy tolerance pr

5.3 Effect of knowledge q possessed by attacker

In this section, we evaluate the impact of the amount of knowledge q that an attacker possesses regarding the experimental results. However, we do not compare our results against those in the three methods, because that study did not consider the amount of knowledge the attacker has; thus, only the results of our experiments are given here. When the quantity of the attacker's knowledge q is 2 and 3, and the user's privacy tolerance p_r = 1/2, 1/4, 1/6, 1/8, and 1/10, the experimental results in Fig. 8(a) and Fig. 8(b) show the performance of our algorithm, respectively. In Fig. 8(a), the x-axis represents the size of the attacker's knowledge q, and the y-axis represents the average error rate; we can see that for a constant privacy tolerance p_r, as the size of the attacker's knowledge q increases, the average error rate also increases; this is because the larger the value of q, the more location data the attacker possesses of the user trajectory; thus, to prevent the attacker from using his/her background knowledge to infer other locations on the trajectory, we must delete more location data, increasing the average error rate. The average standard deviation increases under a decrease of p_r, because the probability of the attacker inferring other locations using their background knowledge is much greater than the user's privacy tolerance p_r. Therefore, the number of sensitive locations that must be deleted is large, increasing the average error rate. The standard deviation in Fig. 8(b) also shows the same trend.

5.4 Frequent-visit pattern protection

By comparing our $PSR\&PPM\_KG$ method with the $TOPF$, $Prefix$ and $ICBA$ methods, we evaluate the quality of the anonymized results by using the widely adopted F-measure in Fig.9. The x-axis represents the number of moving objects, and the y-axis represents the value of the F-measure. From the figure, we can see that our method's F-measure value exceeds that of the $TOPF$ method for different moving objects,

indicating that our method protects more moving modes.

As can be seen from the above experimental results, when the size of the moving object changes, the trajectory data utility of $PSR\&PPM\_KG$ method is 0.4302 higher than that of $TOPF$ method, 0.5623 higher than that of $Prefix$ method, and 0.4917 higher than that of $ICBA$ method, respectively. When the adaptive privacy threshold p_r is changed, the trajectory data utility of $PSR\&PPM\_KG$ method is 0.1212 higher than that of $TOPF$ method, and 0.3262 higher than that of $Prefix$ method, and 0.2915 higher than that of $ICBA$ method, respectively. The experimental results show that our approach provides a higher level of data utility than existing methods for trajectory data publication.

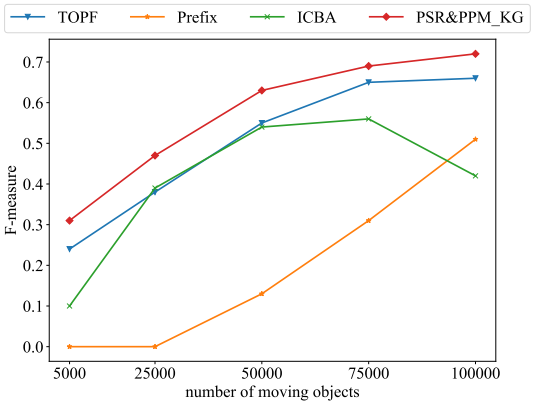
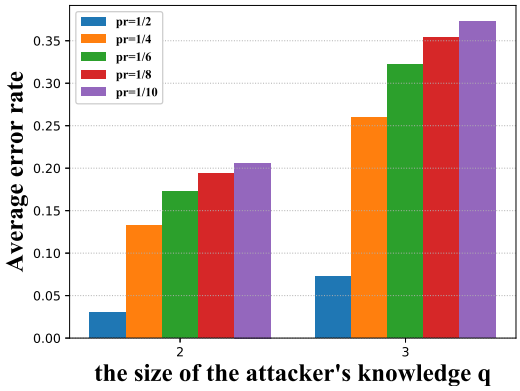
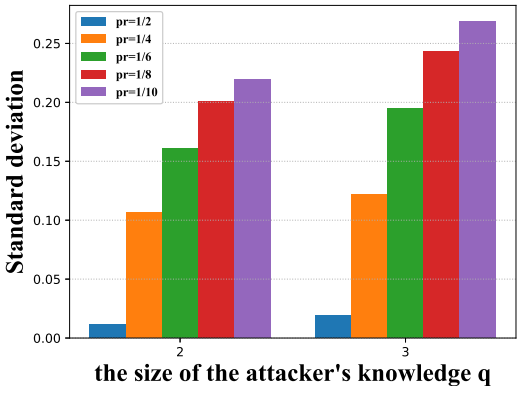


Fig. 9. F-measure



(a) Average error rate



(b) Standard deviation

Fig. 8. Effect of knowledge q possessed by attacker

6 CONCLUSION

Many anonymization approaches have been proposed for trajectory data publication; however, most of these result in a serious loss of information; the subsequent anonymized data are of low utility. Notably, previous studies have been unable to accurately identify users' sensitive locations, owing to the complex relationships between multiple types of entities; thus, these works have often applied a single unified anonymization mechanism.

In this study, we focused on the problem of how to balance data availability and privacy in trajectory data publication. And we proposed a precise sensitivity recognition and privacy protection method based on knowledge graphs for trajectory data publication, to automatically distinguish user's sensitive locations by integrating with multiple attributes knowledge, to enhance data utility whilst guaranteeing privacy. In the experiments, data utility was measured in terms of the average error rate, standard deviation, and F-measure under different parameters. The experimental results obtained for a synthetic trajectory dataset show that the proposed approach provides a higher level of data utility than existing methods for trajectory data publication.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Nos.61662008, 61672176, 61502111, and 61941201), the Guangxi "Bagui Schola" Teams for Innovation and Research Project, the Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, the Guangxi Talent Highland Project of Big Data Intelligence and Application, the Guangxi Natural Science Foundation (Nos.2020GXNSFAA297075, 2018JJA170082), and the Research Fund of the Guangxi Key Lab of Multi-source Information Mining & Security (No. 19-A-02-02), the PostGraduate Education Innovation Project of the Guangxi Normal University under grant JXXYYJSCXXM-006.

References

1. Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel "Unique in the crowd: The privacy bounds of human mobility"

Sci.Rep., 2013, 3, 1376.

2. Y.-A. De Montjoye, L. Radaelli, and V. K. Singh, "Unique in the shopping mall: On the reidentifiifiability of credit card metadata," Science, 2015, 347(6221), 536−539.
3. J. Tang, J. Liang, S. Zhang, H. Huang, F. Liu, Inferring driving trajectories based on probabilistic model from large scale taxi gps data,Physica A, 2018, 506, 566−577.
4. R. Wu, G. Luo, J. Shao, L. Tian, C. Peng, Location prediction on trajectory data: A review, Big Data Min. Anal. 2018, 1(2), 108−127.
5. A. Karatzoglou, N. Schnell, M. Beigl, A convolutional neural network approach for modeling semantic trajectories and predicting future locations, in: International Conference on Artificial Neural Networks,Springer, 2018, 61−72.
6. T. Hu, X. Zhu, W. Guo, S. Wang, J. Zhu, Human action recognition based on scene semantics, Multimedia Tools Appl. 2018, 1−22.
7. E. Toch, B. Lerner, E. Ben-Zion, I. Ben-Gal, Analyzing large scale human mobility data: a survey of machine learning methods and applications, Knowl. Inf. Syst. 2019, 58(3), 501−523.
8. C. Zhou, C. Ma, S. Yang, P. Wu, L. Liu:A Location Privacy Preserving Method Based on Sensitive Diversity for LBS. NPC, 2014, 409−422.
9. Y. Dai, J. Shao, C. Wei, D. Zhang, H. Tao Shen:Personalized semantic trajectory privacy preservation through trajectory reconstruction. World Wide Web, 2018, 21(4), 875−914.
10. X. Wang, Z. Zhang, Y. Luo, Q. Yu:Hierarchical interpolation point anonymity for trajectory privacy protection. Intell. Data Anal. 2019, 23(6), 1397−1419.
11. L. Sweeney, k−anonymity: a model for protecting privacy, Int. J. Uncertain.Fuzz. Knowl.-Based Syst. 2012, 10(5), 557−570.
12. M. Gruteser, D. Grunwald, Anonymous usage of location−based services through spatial and temporal cloaking, in: Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys'03), San Francisco, California, 2003, 31−42.
13. O. Abul, F. Bonchi, M. Nanni, Never walk alone: uncertainty for anonymity in moving objects databases, in: Proceedings of the IEEE Twenty−Fourth International Conference on Data Engineering (ICDE'08), Cancun, Mexico, 2008, 376−385.
14. N. Harnsamut, J. Natwichai, S. Riyana:Privacy Preservation for Trajectory Data Publishing by Look−Up Table Generalization. ADC. 2018, 15−27.
15. R. Chen, B.C.M. Fung, N. Mohammed, B.C. Desai, K. Wang, Privacy−preserving trajectory data publishing by local suppression, Inf. Sci. 2013, 231(1), 83−97.
16. S. Gao, J. Ma, C. Sun, X. Li, Balancing trajectory privacy and data utility using a personalized anonymization model, J. Netw. Comput. Appl. 2014, 38, 125−134.
17. Z. Huo, X. Meng, H. Hu, Y. Huang, You can walk alone: trajectory privacy−preserving through significant stays protection, in: Proceedings of the Seventeenth International Conference on Database Systems for Advanced Applications−Volume Part I (DASFAA'12), Springer−Verlag, Busan, South Korea, 2012, 351−366.
18. Y. Wu, Q. Tang, W. Ni, Z. Sun, S. Liao, A clustering hybrid based algorithm for privacy preserving trajectory data publishing, J. Comput. Res. Dev. 2013, 50(3), 578−593.
19. F. Li, F. Gao, L. Yao, Y. Pan, Privacy preserving in the publication of large−scale trajectory databases, in: Proceedings of the Second International Conference on Big Data Computing and Communications, Springer International Publishing,Shenyang, China, 2016, 367−376.
20. S. Zhang, K.−K.R. Ch, Q. Liu, G. Wang, Enhancing privacy through uniform grid and caching in location-based services, Future Generation Computer Systems, 2017.
21. E. Ghasemi Komishani, M. Abadi, F. Deldar: PPTD: Preserving personalized privacy in trajectory data publishing by sensitive attribute generalization andtrajectory local suppression. Knowl.BasedSyst. 2016, 94, 43−59.
22. X. Zhao, Y. Dong, D. Pi:Novel trajectory data publishing method under differential privacy. Expert Syst. Appl. 2019, 138.
23. L. Yao, X. Wang, X. Wang, H. Hu, G. Wu,Publishing Sensitive Trajectory Data Under Enhanced l−Diversity Model. MDM, 2019, 160−169.
24. C. Chen, Y. Luo, Q. Yu, G. Hu:TPPG: Privacy−preserving trajectory data publication based on 3D−Grid partition. Intell. Data Anal. 2019, 23(3), 503−533.
25. P. Sui, X. Li, Y. Bai:A Study of Enhancing Privacy for Intelligent Transportation Systems: k−Correlation Privacy Model Against Moving Preference Attacks for Location Trajectory Data. IEEE Access, 2017, 5, 24555−24567.
26. J. Zhao, Y. Zhang, X. Li,et al.A trajectory privacy protection approach via trajectory frequency suppression[J].Chinese Journal of Computers, 2014, 37(10), 2096−2106.
27. T. Brinkhoff:A Framework for Generating Network−Based Moving Objects. GeoInformatica, 2002, 6(2), 153−180.
28. Y. Dong, D. Pi:Novel Privacy−preserving algorithm based on frequent path for trajectory data publishing. Knowl. Based Syst. 2018, 148, 55−65.
29. G. Badu-Marfo, B. Farooq, Z. Patterson:Perturbation Privacy for Sensitive Locations in Transit Data Publication: A Case Study of Montreal Trajet Surveys. CoRR abs, 2019.
30. P. Sui, X. Li:A privacy−preserving approach for multimodal transaction data integrated analysis. Neurocomputing, 2017, 253, 56−64.
31. R.G. Pensa, A. Monreale, F. Pinelli, D. Pedreschi, Pattern-preserving k-anonymization of sequences and its application to mobility data mining,in: Proceedings of the International Workshop on Privacy in Location-Based Applications, DBLP, Malaga, Spain, 2009.
32. S. Gurung, D. Lin, W. Jiang, A. Hurson, R. Zhang, Traffic information publication with privacy preservation, ACM Trans. Intel. Syst. Technol, 2014, 5(03), 1−26.
33. Y. Q. Yang, J. H. Cai, H. F. Yang, J. F. Zhang, X. J. Zhao, TAD: A trajectory clustering algorithm based on spatial-temporal density analysis. Expert Syst. Appl, 2020, 139.