Hindawi
Security and Communication Networks
Volume 2018, Article ID 4248092, 14 pages
https://doi.org/10.1155/2018/4248092

Research Article

Research on Trajectory Data Releasing Method via Differential Privacy Based on Spatial Partition

Qilong Han, Zuobin Xiong, and Kejia Zhang

Department of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Kejia Zhang; kejiazhang@hrbeu.edu.cn

Received 23 August 2018; Revised 30 September 2018; Accepted 9 October 2018; Published 1 November 2018

Guest Editor: Liran Ma

A number of security and privacy challenges of cyber system are arising due to the rapidly evolving scale and complexity of modern system and networks. The cyber system is a fundamental ingredient for Internet of Things (IoT) and smart city which are driven by huge amount of data. These data carry a lot of information for mining and analysis, especially trajectory data. If unprotected trajectory data is released, it may disclose user's personal privacy, such as home, religion, and behavior mode, which will endanger their personal security. Until now, many methods for protecting trajectory information have been proposed. However, these methods have the following deficiencies: (i) they cannot defend against speculative attacks if the attacker's background knowledge is maximized; (ii) when studying the problem, they made some strong assumptions that did not match the reality; (iii) the implementation algorithm is complicated and the time complexity is high, which means that data cannot be executed quickly when the amount is large. So, in this paper, we propose a spatial partition based method to publish trajectory data via differential privacy. First, by exponential mechanism, we divide location set at the same time into different partitions fast and accurately. Then we propose another effective method to release trajectory in a differential private manner. We design experiment based on the real-life dataset and compare it with existing method. The results show that the trajectory dataset released by our algorithm has better usability while ensuring privacy.

1. Introduction

In recent years, with the development of IoT, smart city is becoming popular to us and facilitates our life. As the important foundation of IoT application, cyber-physical system collects and provides a lot of data to it from users. Usually, personal data of user include the real-time location, usage time, and biometric information [1]. Among them, trajectory information is very important to user as well as adversary. Because it carries a lot of information for data mining and scientific analysis, if the trajectory is obtained illegally or released without protection, it is easy to reveal the user's personal privacy, such as home address and behavior mode, which will endanger the personal security of the user [2–4]. Once the information is leaked into an attacker, it will cause immeasurable loss to the user, resulting in threats to personal safety and property. Therefore, in cyber system, for the security and privacy of user, it is extremely urgent to provide an effective protection method for a large amount of users' privacy data [5–11].

Privacy disclosure and protection in trajectory data publication can usually be divided into two categories [12]:

(1) Only one trajectory is included in the published trajectory dataset. Each location point on the trajectory corresponds to a record, and the user's privacy requirement is to ensure that the location at a certain point is safe [13].

(2) The published trajectory dataset contains multiple trajectories. Each of these trajectories is considered a record. Our aim is to publish a sanitized dataset so that the attacker could not know the correspondence between a trajectory and the user.

Based on the above two types of problems, many privacy protection methods based on *k-anonymity* [14] and partitioning have emerged in recent years, such as *l-diversity* [15], *t-closeness* [16], and (α, k)- *anonymity*. Although these methods can protect more details of the data, they all require special attack assumptions and background knowledge. In addition, for the above privacy protection methods, some new attack models have emerged, such as *combined attacks* [17] and

background knowledge attacks [18]. These new attack models are really serious challenges to the effectiveness of above methods.

The root cause of above situation is that (i) the background knowledge of attacker is difficult to define and (ii) these early privacy protection models did not provide an effective and rigorous way to prove their level of privacy protection. Therefore, researchers are trying to find a sufficiently usable privacy protection model that can resist various forms of attack with the attacker's maximum background knowledge. The rise of differential privacy (DP) [19] makes it possible to implement this idea.

Differential privacy is a probability-based privacy model proposed by Dwork [19] in 2006 for privacy breach of statistical database. The advantages of differential privacy are the following: (i) it is based on a powerful mathematical model that can provide quantitative analysis for privacy level; (ii) the usability can be controlled by adjusting privacy budget to add proper noise; (iii) the privacy can still be guaranteed even if the attacker's background knowledge is maximum. Because of these above advantages, differential privacy has quickly gained the attention of researchers [20].

The main content of this paper is to use differential privacy to protect the trajectory dataset generated by moving objects. For the spatiotemporal trajectory dataset, it is important to know how to use the differential privacy method to process the data, so that the published data can protect the relationship between the user and the trajectory while protecting the sensitive location of the user on the trajectory.

Contributions. The main contributions of this paper are threefold:

(1) For original dataset in which location points are strictly ordered by timestamp, we propose a Hilbert curve based spatial partition method. According to the sparsity of location distribution in the area at the same time, we leverage exponential mechanism to get the most likely accurate partition, which protects sensitive locations of individual moving object.

(2) We then propose a simple and effective differential privacy data publishing algorithm to generate trajectory. On the basis of the partitions of each timestamp, we generate similar trajectories of original trajectory. Then we design a method to reduce the Laplace noise injected into the data. The released noisy dataset protects the relationship between the user and trajectory.

(3) Through theoretical analysis and experimental evaluation on the real-life dataset, the privacy guarantee and usability of the proposed publishing method are proven. Compared with the existing algorithm, using the Hausdorff distance and spatiotemporal range query distortion as the evaluation criteria, the experimental results show that our algorithm is superior to the previous algorithm.

2. Related Work

Researchers have conducted a lot of researches on trajectory privacy protection and achieved rich results. Nicolás [21] pointed out that directly deleting the ID of the trajectory does not guarantee the user's privacy. With the advancement of the attack means, even if the location of each user is protected, the attacker can still learn the user's mode through association analysis and data mining.

In such methods, trajectory *k-anonymity* is one of the most commonly used methods [1]. Sweeney first proposed the *k-anonymity* model [14] and at least k-1 records were indistinguishable from each record. This method guarantees user privacy to some extent, but the sensitive attribute values in the same anonymous set may be identical or of few types, and the attacker can still infer the information of a record from the table. In response to this problem, Shwin Bgl et al. [15] proposed a privacy protection standard for *l-diversity*, which requires that each k-anonymous set has at least l different sensitive attribute value under the premise that the data record satisfies the *k-anonymity* model. This prevents an attacker from matching a record to a determined individual. In [22, 23], the trajectory *k-anonymity* is extended, and the (k, δ)-*anonymity* model is proposed. It is required to find at least k-1 other trajectories in the δ uncertainty region around it. In [24], by suppressing some sensitive information in the user's trajectory, the probability of the attacker acquiring the user's trajectory information through data mining is reduced, so the trajectory privacy is protected. Terrovitis [25] proposed a new suppression mechanism, which divides the trajectory into sensitive regions and nonsensitive regions. When the user enters the sensitive area, the user's location in the area is suppressed, and the information update is stopped; when entering the nonsensitive area, no suppression is performed. Kido [26] proposed a trajectory data suppression mechanism that can simultaneously suppress sensitive nodes in the trajectory and nodes that can uniquely identify users to achieve trajectory privacy protection. Mohammed [27] designed a user trajectory suppression mechanism for high-dimensional sparseness and selected appropriate suppression points combined with *k-anonymity,* so that each subsequence of the same length in the user trajectory has no less than k identical subsequences. Shokri [28] proposes a way to add random noise to the user's actual location by comprehensively considering the user's privacy requirements, the attacker's knowledge, and the maximum tolerable quality of service degradation caused by the confusion of the real location. When attacker reconstructs the actual location of the user, the error rate of speculating rises while satisfying the user's quality of service requirements. According to the predictability of human behavior patterns, Theodorakopoulos [29] proposed an algorithm for exchanging access-sensitive location points to predict the user's future trajectory while protecting the privacy of the user's location.

After the differential privacy model is proposed, it was quickly applied in the privacy protection of trajectory data release. In [30], the differential privacy model was applied for the first time to propose the prefix method. This method uses the hierarchical framework to construct the prefix tree, dividing the trajectory sequence with the same prefix into the same branch of the tree and adding the noise by counting the node's count value. However, as the tree grows, the prefix will form a large number of leaf nodes, making the added noise too large and reducing the accuracy of the published

Table 1: Notations that will be used.

Notation	Explanation
T_i	a trajectory in dataset
$\widetilde{T}_i$	generalized trajectory in noisy dataset
D	dataset has multiple trajectories
r	radius of the stand deviation circle
r_w	radius of the weighted stand deviation circle
L, L_i	location universe and the location set at each timestamp i
$\widetilde{L}, \widetilde{L_i}$	location universe and location set at timestamp i after dividing
L_{ij}	center of location set after dividing in the timestamp i, jth set
C	set of trajectory count number in dataset
d	distance set in adjacent trajectory count number
$\widetilde{D}, \overline{D}$	the noisy dataset and the published dataset

dataset. Also, these above methods only consider the spatial characteristics of the trajectory data, regarding the trajectory data as a sequence of spatial location points, or make bad utility when trajectory is long.

3. Materials and Methods

In this chapter, we define the spatiotemporal trajectory dataset, review the knowledge of differential privacy, and introduce a few methods that we will use in the next part. Besides, notations we use are listed in Table 1.

3.1. Spatiotemporal Trajectory Database

Definition 1 (spatial-temporal trajectory). A spatiotemporal trajectory is a location sequence generated by ordering multiple timestamps, representing the trajectory of a moving object in space. $T = (l_1, t_1) \longrightarrow (l_2, t_2) \longrightarrow \cdots \longrightarrow (l_{|T|}, t_{|T|})$, where $|T|$ is the length of this trajectory, and, $\forall i$ $(1 \leq i \leq |T|)$, $l_i \in L_i$ is a discrete spatial point, which is represented by latitude and longitude coordinate.

L_i is the location universe of locations at time t_i. We use $Time(T)$ to represent the timestamps of a spatiotemporal trajectory.

A spatiotemporal trajectory dataset D is a dataset consisting of $|D|$ spatiotemporal trajectories, like $D = \{T_1, T_2, \cdots, T_{|D|}\}$. In general, the length of each trajectory and the sampling interval are different due to the different sources and sampling methods of the dataset. For the sake of convenience, the original dataset will be preprocessed in our experiments, making

$$Time\left(T_i\right) = Time\left(T_j\right), \quad \forall T_i, T_j \in D,\ i \neq j. \tag{1}$$

3.2. Differential Privacy. As a well-defined and provable privacy model, differential privacy has been widely used in data protection and data mining privacy protection since it was introduced [20].

Definition 2 (ε-differential privacy). A randomized algorithm is differential privacy if and only if any two databases D and D' contain at most one different record and, for any possible anonymized output $O \in Range(A)$,

$$\Pr\left(A\left(D\right) = O\right) \leq e^{\varepsilon} \times \Pr\left(A\left(D'\right) = O\right) \tag{2}$$

We say that the algorithm satisfies ε-differential privacy.

In the above formula, ε is a parameter. The smaller it is, the stronger the privacy protection provided by the differential privacy mechanism is.

In this paper, two important tools for implementing differential privacy are the Laplace mechanism [31] and the exponential mechanism [32]; both of them use the global sensitivity.

Definition 3 (global sensitivity). For a given function $f : D \longrightarrow R^d$, its global sensitivity is

$$\Delta f = \max_{D_1, D_2} \left\| f\left(D_1\right) - f\left(D_2\right) \right\|_1 \tag{3}$$

D_1 *and* D_2 are neighboring databases that differ in at most one record.

Laplace Mechanism. The mechanism is designed for functions whose query results are numerical. Differential privacy is achieved by injecting appropriate noise into the result of query. The noise generation is based on the Laplace distribution function and its probability distribution is

$$p\left(x \mid b\right) = \frac{1}{2b} e^{-|x|/b}, \quad b = \frac{\Delta f}{\varepsilon} \tag{4}$$

where b is the noise scale and it is determined by the global sensitivity of the function and the privacy budget.

Theorem 4. *For any $f : D \longrightarrow R^d$, the mechanism that adds independently generated by Laplace noise on the real result, like*

$$A\left(D\right) = f\left(D\right) + Laplace\left(b\right) \tag{5}$$

Then the mechanism satisfies ε-differential privacy.

Exponential Mechanism. For some functions whose query result is nonnumeric or has no meaning after adding noise,

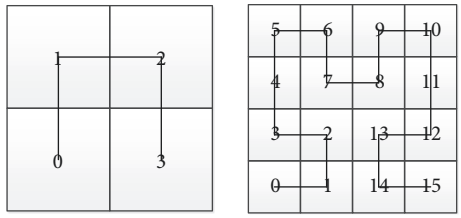


Figure 1: 1- and 2-order Hilbert curve.

such as the query result which is a certain attribute in the dataset, Mcsherry and Talwar [32] proposed a protection mechanism that satisfies differential privacy for this situation. It first defines utility function $u : (D \times r) \longrightarrow R$ that assigns a real score for every possible output r in the output domain R. Higher score means more utility. It then selects an output $r \in R$ with the probability proportional to $e^{(\varepsilon \times u(D,r))/2\Delta u}$; $\Delta u = \max_{\forall r, D_1, D_2} |u(D_1, r) - u(D_2, r)|$ is the sensitivity of utility function. As the outputs with higher score are more likely to be selected, this mechanism is close to optimal. In addition, the utility function should be insensitive to the change of a single record.

Theorem 5. *For any function $u : (D \times r) \longrightarrow R$, the mechanism chooses an output r, $r \in R$, with the probability proportional to $e^{(\varepsilon \times u(D,r))/2\Delta u}$ being able to guarantee ε-differential privacy.*

Composition Properties

Theorem 6 (sequential composition). *Suppose that each algorithm A_i satisfies ε_i-differential privacy. A sequence of A_i over database D provides $\sum \varepsilon_i$-differential privacy.*

Theorem 7 (parallel composition). *Suppose that each algorithm A_i satisfies ε_i-differential privacy. A sequence of A_i over a set of disjoint database D_i provides* $\max(\varepsilon_i)$*-differential privacy as a whole.*

3.3. Hilbert Curve. The Hilbert space filling curve is a continuous but nonconductible mathematical curve proposed by German mathematician Hilbert in 1891 [33], which is being widely applied in spatial sorting currently. Space filling curve is a method of mapping d-dimensional space into 1-dimensional space. It passes through every discrete unit of high-dimensional space only once and numbers these units in a linear order as in Figure 1. From the d-dimensional space to the linear space mapping process, the Hilbert curve can maintain location relationship between the point and its neighbors to some extent, so it has good characteristics in the spatial point division problem [34].

Using Hilbert curve to divide the space is relatively fast and capable of resisting speculative attacks [35]. However, due to the uneven distribution of spatial points and sparsity, the space of the Hilbert curve partition is slightly larger than the space based on the *KNN* method [36], which results in larger errors in the calculation of spatial point anonymity and relatively high computational time. Therefore, we need an efficient method that maintains both the spatial nature of the Hilbert curve and the ability to produce relatively small partition.

We note that points with similar Hilbert values are similar in the 2-dimensional or high-dimensional space is a sufficient unnecessary condition [37], which means that, in a Hilbert curve, there may be similar points in the two-dimensional space but the Hilbert values differ greatly. In Figure 2(a), U_2 and U_8 are such points. And because the Hilbert curve is recursively divided using the quadrant mode, this results in no Hilbert curve that makes the two points close, no matter how high order there is. This is why the space partition in Hilbert curve is slightly larger than the space based on the *KNN* method. We also noticed that, for spatial points under the same distribution, even the equal-order Hilbert curves could provide different partition after rotating. In Figure 2, the eight users are in the same locations, but the spatial regions obtained after 3-anonymous partitioning [38] through different 3-order Hilbert curves have a large difference in size.

In Figure 2(a), the partition of U_1, U_2, U_3 occupies 16 cells and U_4, U_5, U_6, U_7, U_8 have 49 cells. The average is 32.5. Meanwhile, in Figure 2(b), U_5, U_6, U_7 occupies 6 cells and U_8, U_2, U_3, U_4, U_1 20. The average is 13. The effect after the rotation is visible. Therefore, this paper considers using such method to divide the space. It is hoped that when the divided regions are constructed, the accuracy of the partitions can be ensured while obtaining a smaller divided region.

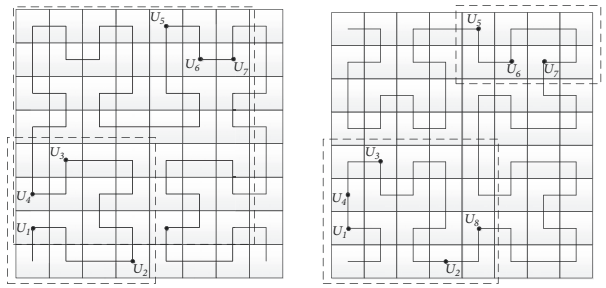
3.4. Location Entropy. The definition of location entropy is derived from Shannon entropy [39] and is a measure of uncertainty. In the field of location privacy, many methods [40, 41] have adopted the concept of location entropy to measure the popularity of points of interest (POI) or to use location entropy to design privacy protection mechanisms.

For a given location l, let V_l be the set of visits to that location. Thus, $c_l = |V_l|$ is the total number of visits to l. Also, let U_l be the set of distinct users that visited l, and let $V_{l,u}$ be the set of visits that user u has made to the location l. Thus, $c_{l.u} = |V_{l,u}|$ denotes the number of visits of user u to location l. The proportion of visits l by user u to the total number of visits l is $p_{l,u} = c_{l,u}/c_l$. According to Shannon entropy [42], the location entropy of l is

$$LE(l) = -\sum_{u \in U_l} p_{l,u} \log_2 p_{l.u} \tag{6}$$

The greater the location entropy, the higher the uncertainty of the location and the level of privacy protection. Location entropy reaches a maximum $\log_2 k$, when all users have the same number of visits, where $k = |U_l|$.

3.5. Weighted Standard Deviation Circle. Due to the unevenness of the spatial point distribution pattern and its sparsity, the parameter of the subspace size generated by clustering or partitioning does not fully reflect the quality of the partitioning result, which in turn affects the availability of data. So we introduce a standard deviation circle to solve this problem. In spatial point mode analysis, standard deviation



Node	U_1,U_2,U_3	U_4,U_5,U_6,U_7,U_8
Size	16	49

(a)

Node	U_5,U_6,U_7	U_8,U_2,U_3,U_4,U_1
Size	6	20

(b)

Figure 2: 3-anonymous partition by Hilbert curve.

circle, weighted standard deviation circle, of coordinate x/y is used to describe the discrete trend of spatial distribution [43].

The radius of the standard deviation circle is similar to the standard deviation in classical statistics, describing the spatial deviation of the observed points.

The standard deviation circle radius calculation method is

$$r = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \overline{x})^2 + (y_i - \overline{y})^2}{n-2}} \tag{7}$$

where $\overline{x}, \overline{y}$ is the center point of the space, x_i, y_i are the coordinates of the point, and n is the number of points in the space. For two areas with equal size, if the total number of points is the same, then the area with a large standard deviation circle radius has a large spatial dispersion [43]. However, when some attributes of the spatial point itself can affect the degree of spatial dispersion, the result of the standard deviation circle will be biased. In order to correct this offset, some attributes of the spatial point can be used as weights to generate a weighted standard deviation circle.

Then the radius is calculated as follows:

$$r_w = \sqrt{\frac{\sum_{i=1}^{n} w_i (x_i - \overline{x})^2 + \sum_{i=1}^{n} w_i (y_i - \overline{y})^2}{\sum_{i=1}^{n} w_i}} \tag{8}$$

where w_i is the attribute weight; other parameters are defined as above (7).

4. Our Proposal

In this chapter, we describe how to design a differential privacy publishing method for a trajectory dataset. First, we introduce our method and then decompose the algorithm into two subalgorithms and finally prove that the whole algorithm satisfies ε-differential privacy.

4.1. Preview. We solve processing the original data by dividing the location set of the trajectory database at each timestamp. For example, the similar points in the original dataset are divided into the same area, and the divided areas are guaranteed to be as accurate as possible. In this way, we can greatly reduce the set of location points within the timestamp, thereby reducing the output domain of the trajectory dataset. Further, similar trajectories are also merged, which greatly increases their counts, reduces the injected noise, and improves data availability.

4.1.1. Differential Privacy Spatial Division Algorithm. This algorithm uses the Hilbert curve to divide the set of location points L_i of each timestamp t_i and divides the original set into multiple subsets, which is regarded as a kind of partition. When using multiple Hilbert curves for dividing, many partitions are produced. The size of the area produced by each partition is different, and the distribution of internal location points is not same. According to the location entropy of each point in an area, the weighted standard deviation circle of all points in the area is calculated, and then the exponential mechanism is used to output the best partition with higher availability. Thus, the location set L_i in the original dataset becomes $\widetilde{L_i}$, as an input to the next algorithm.

4.1.2. Differential Private Data Publishing Algorithm for Trajectory Generated. This algorithm generates generalized spatiotemporal trajectories based on the output $\widetilde{L_i}$ of last step at each timestamp t_i. Then we use the Laplace mechanism to publish the noise count of the generalized spatiotemporal trajectory. To improve efficiency and utility, we only focus on the trajectories that exist in the original dataset and design a method to reduce the noise injected.

As in Figure 3, there are 8 trajectories in the original database and the length is 3. We divide the location set at each timestamp t_i in Algorithm 1 and generalize noisy trajectories by Algorithm 4. Then we release the processed trajectory dataset.

The details of two core algorithms are elaborated separately below.

4.2. Differential Privacy Spatial Division Algorithm. In order to reduce the size of the spatiotemporal location set and

Input: original dataset D, privacy budget ε
node amount k of each area in partition
Output: $\widetilde{L} = \{\widetilde{L_1}, \widetilde{L}_2, \cdots \widetilde{L_{|T|}}\}$ after dividing
(1) Initialize L_i. Assign locations at each time t_i of the trajectory dataset to L_i
(2) For each L_i, computing $|L_i|$ and location entropy for l_i in L_i, determining order of Hilbert curve
$$order = \left\lfloor \log_2^{\sqrt{|L_i|}} \right\rfloor$$
(3) Generate four Hilbert curves, for each curve
(4) computing $aver(r_w) = (1/|G|) \sum_{i=1}^{|G|} r_w$ according to $order$ and k
(5) select r_w with probability P
P is proportional to $e^{(\varepsilon \times u(D,r))/2\Delta u}$
(6) Output the optimal partition of L_i at t_i
(7) Output the set $\widetilde{L} = \{\widetilde{L_1}, \widetilde{L}_2, \cdots \widetilde{L_{|T|}}\}$

Algorithm 1: DPSD(D, ε).

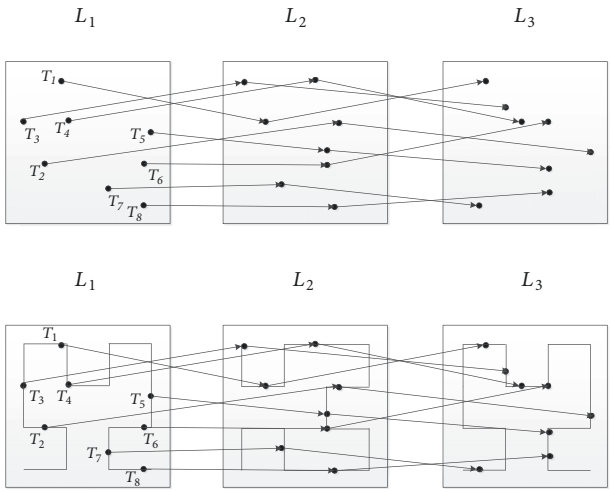


Figure 3: Original trajectory dataset.

ensure the availability of the trajectory data after the differential privacy mechanism, we try to process the location set in the original dataset by dividing the regions and merging them. Hua J [44] pointed out that the k-means method can be used for clustering in the location set, but the traditional k-means method needs to determine the number of clusters k in advance. And the selection of the initial center point of k-means cluster has a great influence on clustering effect and time. For the problem of uneven distribution of spatial points, although density-based clustering method [45] can perform adaptive clustering without determining special the number of clusters, in density-based clustering, the distance between cluster nodes and cluster center is uncontrollable [46], which is not suitable for the application scenario of this paper. So, to solve the above problems, we use the idea of grid partitioning aggregation in [46] and propose a differential privacy spatial division algorithm.

In the original dataset of Figure 3, each trajectory is recorded as T_i; we divide each spatiotemporal point on the trajectory into different location set L_i by time t_i and perform the same dividing operation for each L_i separately. At first, we want to divide the location set as what we introduced in Section 3.3 by average size of cells. However, in the subsequent research, we found the shortcomings of this approach. Consider the division of the following two location distribution.

In Figure 4, a and b represent two kinds of distribution. There is one point in 0-3 cell in a, but there is one point in cells 0 and 1 and two points in cell 3 in b. When dividing, these two distributions will get the same dividing area with same anonymity effect but different service quality because of its uneven density.

For the same reason, in the distribution of c and d, after dividing, the size of the cell area in c is 4, while in d it is 6. But, in fact, the division in d should be better. These two comparative examples in Figure 4 illustrate that Hoa Ngo's dividing method based on the number of grids proposed in [36] is prone to be inaccurate for some distributions, so we recommend using a more accurate method to divide location set, which can indicate the difference in distribution.

We use the standard deviation circle in Section 3.5 to solve this problem. According to formula (7), we can calculate the standard deviation circle radius of the two comparative sample pieces of data in Figure 4. The results are a=1>b=0.935 and c=1.581>d=1.17, which is in line with our observation and actual condition. So it can be used as an evaluation criterion for area dividing. However, we cannot just simply use multiple Hilbert curves for spatial dividing and then choose the best partition with the smallest average standard deviation circle radius, even if this method can produce the most accurate division. Because the optimal partitioning produced by each Hilbert curve is different, this simple way of choosing the minimum output is susceptible to speculative attacks. To solve this problem, we introduce the

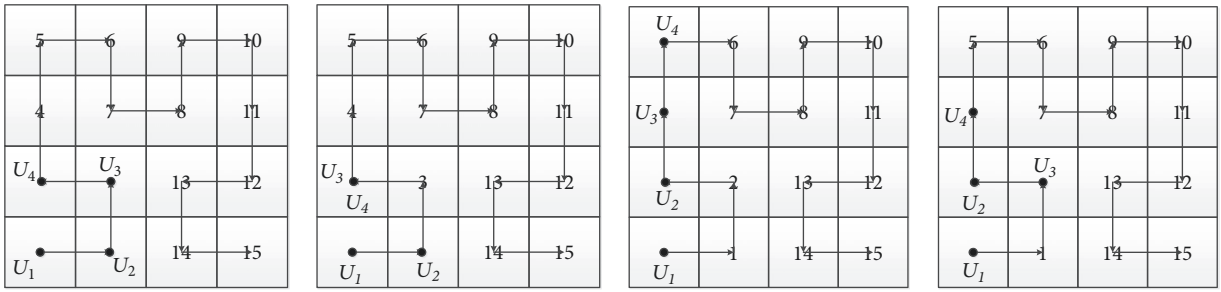


Figure 4: 2-order Hilbert curve partition.

exponential mechanism of differential privacy to ensure a relatively accurate division to be output. The utility function $u(D, r) \longrightarrow R$ is designed as

$$u(D, r) = 1 - \frac{Size(r)}{MaxSize(R)} \quad r \in R \tag{9}$$

where r is the average radius of the standard deviation circle of each area in the partition and R is the collection of all r.

For example, we use three Hilbert curves to divide a location set, and the results are r_1=5, r_2=8, and r_3=10, respectively; then their corresponding utility scores are $u(D, r_1) = 0.5$, $u(D, r_2) = 0.2$, and $u(D, r_3) = 0$. In this way, we get the highest score for the optimal result of dividing the region. Then we can randomly select an output r with the probability $P(D, r)$ proportional to $e^{(\varepsilon \times u(D,r))/2\Delta u}$, where

$$P(D, r) = \frac{e^{(\varepsilon \cdot u(D,r))/2\Delta u}}{\sum_{r \in R} e^{(\varepsilon \cdot u(D,r))/2\Delta u}} \tag{10}$$

ε is privacy budget and Δu is sensitivity of $u(D, r) \longrightarrow R$. Differential privacy requires global sensitivity to be as small as possible to minimize the noise injected. The sensitivity of our utility function is $\Delta u = \max_{r \in R} |u(D, r) - u(D', r)| = 1$, so it can be used. In addition, we noticed that when using data-relative publishing method, the properties of the data itself will also affect the publishing results. Based on this idea, we can replace the standard deviation circle used in the utility function with the weighted standard deviation circle. The attribute of the location entropy in dataset is taken into account, which can be calculated by formula (8) to get weighted standard deviation circle radius. At same time, the utility function should be modified to

$$u(D, r_w) = 1 - \frac{Size(r_w)}{MaxSize(R)} \quad r_w \in R \tag{11}$$

After the process of Algorithm 1, L_i in the original dataset can be divided into different area, as shown in Figure 5, and the pseudocode of the algorithm is shown in Algorithm 1.

The order we use to generate the Hilbert curve is based on the number of location points in the target area. Our aim is that, in the best case, each location can get a Hilbert value, so we need at least n cells, which require the order of Hilbert curve to be $order = \lfloor \log_2^{\sqrt{|L_i|}} \rfloor$.

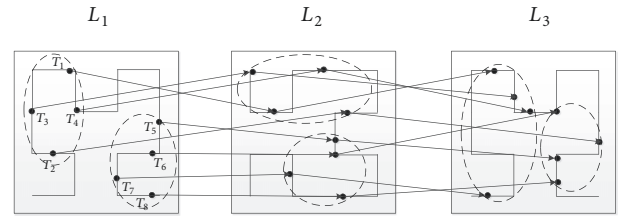


Figure 5: Divided original dataset.

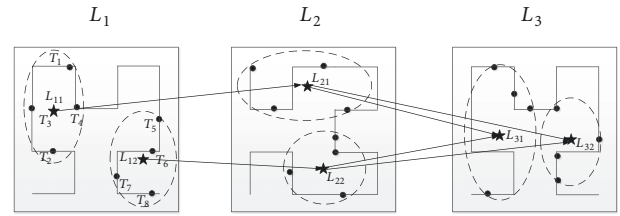


Figure 6: Replaced trajectory dataset.

4.3. Differential Private Data Publishing Algorithm for Trajectory Generated. Through Algorithm 1 in previous section, location set L_i at each timestamp t_i of the original spatiotemporal trajectory dataset is divided into $\widetilde{L_i}$. After this, in order to construct generalized trajectories that are similar to the trajectories in the original dataset, we also need to perform the Algorithm 4. It is to generate a trajectory dataset $\overline{D}$ to be published by divided location sets $\widetilde{L_i}$. It should be noted that, in Algorithm 1, once k is fixed, the number of partitions in $\widetilde{L_i}$ is determined to be $[|L_i|/k]$. If the trajectory's length is l, the number of all possible trajectories is $[|L_i|/k]^l$. Suppose that the number of partitions is 80 in each $\widetilde{L_i}$; the total number of possible trajectories with the length 36, like our setting in experiment, which we have to consider is 80^{36}. It is obviously unacceptable if we add Laplace noise on all of these trajectories. Therefore, we need to process the output of Algorithm 1 to reintegrate the different sets of location partitions into trajectories that are similar to original dataset.

We adopt the following strategy: first, for each $\widetilde{L_i}$, location points that belong to the same area are replaced by the center of this area, and other points in the area are deleted, so that each area corresponds to only one point. Then, according to

Table 2: Generalized trajectories.

Generalized Tra	Original Tra	Count
$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$	T_1, T_3, T_4	3
$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{32}$	T_2	1
$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{31}$	NULL	0
$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$	NULL	0
$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$	NULL	0
$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{32}$	NULL	0
$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{31}$	T_7	1
$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$	T_5, T_6, T_8	3

every trajectory in the original dataset, we find the area it passed and replace this original location point with center of this area. Finally, we get generalized $\widetilde{D}$ trajectory dataset after dividing, as shown in Figure 6.

After the replacement, the count of some generalized trajectories also increases because each divided subset contains multiple location points, as can be seen from Table 2.

Compared to the unprocessed trajectories, increasing the count of trajectories reduces the disturbance of adding noise to the smaller count, which allows us to publish trajectory data more accurately than before. For example, if the original dataset is not processed, its published dataset is $D = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8\}$; the count set of trajectories is $C = \{1, 1, 1, 1, 1, 1, 1, 1\}$. We will have very low availability after adding Laplace noise on such a sequence. The generalized dataset after replacing is $\widetilde{D} = \{\widetilde{T_1}, \widetilde{T_2}, \widetilde{T_3}, \widetilde{T_4}\}$, where each $\widetilde{T_i}$ is the generalized trajectory in Table 2 and $\widetilde{D}$ is the noisy dataset that contains all generalized trajectories.

The count set is $C = \{3, 1, 1, 3\}$. Such a trajectory dataset with differentiated count can withstand the influence of Laplace noise and achieve better usability.

We have noticed that when the number of generalized trajectories is large, the total amount of Laplace noise can also be large. Let us take Table 2 as an example; when we add Laplace noise to $\widetilde{D} = \{\widetilde{T_1}, \widetilde{T_2}, \widetilde{T_3}, \widetilde{T_4}\}$, we need to add independent noise N, $N \sim Lap(\Delta f/\varepsilon)$, where $\Delta f = 1$, because any adjacent trajectory dataset's count values differ by up to 1. For these four trajectories in dataset, the noise amount we need is $4N$. But when we sort and group the count set C, $\{3, 1, 1, 3\} \Longrightarrow \{\{1, 1\}, \{3, 3\}\}$.

For the dataset after group, we just need to add $2N$ Laplace noise to satisfy differential privacy, which means that we just add $1/2N$ noise to each trajectory. While when reducing Laplace noise, new errors arise, which we call mean error, because we need to use the average value to restore the count of each trajectory in each group when publishing data. The difference between average count and the true count results in this error. We still use the example $\widetilde{D}$: the count set of dataset $\widetilde{D}$ is $C = \{3, 1, 1, 3\}$. The count set of neighbor dataset of $\widetilde{D}$ may be $C' = \{3, 1, 1, 4\}$. When we add Laplace noise directly, the noise error is $4N$ and mean error is 0. After grouping this count set, C becomes $C' = \{\{1, 1\}, \{3, 4\}\}$; then the Laplace noise is $2N$ and mean error is $|3 - 3.5| + |4 - 3.5| = 1$. If $4N$ is greater than $2N+1$, then we believe that the error generated by the dividing is acceptable and this release method is preferable. From a global perspective, as the number of group increases, the Laplace error increases gradually, and the mean error decreases. They are mutually constrained. However, from the perspective of data availability, we only need to minimize the total error of data release.

Definition 8 (total error). For count set of the trajectory dataset used in this paper, the total published error is the sum of Laplace error and the mean error.

$$Error\,(C) = Error\,(Lap) + Error\,(Mean) \tag{12}$$

$Error(Lap) = \sum_k N$, where k is group size after dividing and N is noise amount.

$$Error\,(Mean) = \sum_{i=1}^{|G|} \left| c\,(T_i) - aver\,(G) \right|, \tag{13}$$

where $c(T_i)$ is count value of every trajectory in group and $aver(G)$ is the average count in group G.

There are two main factors that affect the performance of this release method: (i) the sorting process may not guarantee differential privacy and (ii) the existence of statistical distance outlier may affect usability. For problem (i), when sorting the trajectory count value, sorted partition is sensitive to adding or deleting a single trajectory. We use the same example to illustrate it: $C = \{3, 1, 1, 3\}$ is the count set of $\widetilde{D}$, which is divided into $C = \{\{1, 1\}, \{3, 3\}\}$. The count set of neighbor dataset of $\widetilde{D}$ may be $C' = \{3, 1, 0, 3\}$ and produce $C' = \{\{1\}, \{3, 3\}\}$. It is apparent that the probability that $C' = \{\{1\}, \{3, 3\}\}$ is equal to $C = \{\{1, 1\}, \{3, 3\}\}$ is 0, which could not satisfy differential privacy. Therefore, in order to satisfy the differential privacy, when sorting and dividing the count set, we add Laplace noise to real count and then sort count. By doing this, Laplace mechanism can still make $c(T) + Lap(1/\varepsilon)$ output the same result as $c(T)' + Lap(1/\varepsilon)$ with a high probability when the count value of some trajectory turns from $c(T)$ to $c(T)'$. The sort set not only can provide differential privacy but also is close to real order to a large extent.

The algorithm is designed as shown in Algorithm 2.

In Algorithm 2, ε is the privacy budget required for sorting, but since line (5) replaces the noisy count with the real count and outputs an approximate order of the true values, there is no impact on the published results, so the algorithm does not actually consume the privacy budget [47].

For problem (ii), the influence of the statistical distance outlier trajectory is important. In this paper, the statistical distance outlier trajectory is defined as follows.

Definition 9 (statistical distance outlier trajectory). Given real number δ, T is a trajectory; if the difference between count value $c(T)$ of T and count value of trajectories T_i is d and if the difference $d = |c(T) - c(T_i)| > \delta$, trajectory T is the statistical distance outlier trajectory.

It can be known from the definition that if the outlier trajectory is divided into the same group with nonoutlier

Input: generalized dataset $\widetilde{D}$, privacy budget ε
Output: approximate sorting set of trajectory count *SortSet*
(1) Initialize *SortSet*=null;
(2) For each trajectory in $\widetilde{D}$
(3) Count real count to get C
(4) For each in C
Adding Laplace noise $Lap(1/\varepsilon)$ to real count and get $\widetilde{C}$
(4) Sort the counts in $\widetilde{C}$
(5) Keep the order of sort in $\widetilde{C}$, replace the noise count with original count
(6) return *SortSet*=$\widetilde{C}$

Algorithm 2: CountSort.

trajectories, the mean error will be very large, which will seriously affect the release result. So, we need to first find the outlier trajectory and try to separate the outlier trajectories into a single group to avoid the influence of other mean errors of group.

Generally, we can take the following methods to divide. First, the output of Algorithm 2 is an approximate correct trajectory count ordered set. Supposing that the generalized count set $C = \{c(T_1), c(T_2), c(T_3), \cdots c(T_n)\}$, we calculate the difference between adjacent elements of this approximate order set $d_i = |c(T_{i+1}) - c(T_i)|$ and get the set of difference $d = \{d_1, d_2, d_3, \cdots d_{n-1}\}$. Then we choose the max $d_{\max} = d_i$; by doing this, we can find the maximum difference trajectory in C, which is the relative outlier trajectory. We divide the count set C into $C_1 = \{c(T_1), c(T_2) \cdots c(T_i)\}$ and $C_2 = \{c(T_{i+1}), c(T_{i+2}) \cdots c(T_n)\}$. If the error after dividing $Error(C_1) + Error(C_2) < Error(C)$, we accept this division and then recursively perform the above process on $C = \{\{C_1\}, \{C_2\}\}$ until all subdivisions do not meet the requirements.

However, the process above does not guarantee privacy. When an attacker has strong background knowledge, it is possible to obtain user privacy by speculating attacks at each division. Therefore, in order to protect privacy from the strongest attacker, we use differential privacy to defend. When we choose $d_{\max}$ from $d = \{d_1, d_2, d_3, \cdots d_{n-1}\}$, exponential mechanism can be used to output $d_{\max}$ with a high probability. We design a simple and effective utility function: $u(D, d_i) = d_i$; the bigger d_i is, the higher its score and probability are, which conform the rule. The sensitivity $\Delta u = 1$ because the max difference is 1 when there is only one different record on neighbor dataset. The probability of output d_i is

$$P(D, d_i) = \frac{e^{(\varepsilon \cdot u(D, d_i))/2\Delta u}}{\sum_{d_i} e^{(\varepsilon \cdot u(D, d_i))/2\Delta u}}, \quad d_i \in d \tag{14}$$

We can get correct group division after above steps.

In Algorithm 3, $Error(C)$ is the initial min error and $Error(C) = N + \sum_{i=1}^{n} |c(T_i) - aver(C)|$, where N is Laplace noise and C is the whole count set. The algorithm continually iterates until the global error is no longer reduced, so the number of iterations and partitions cannot be determined.

Input: *SortSet* output by Algorithm 2, privacy budget ε
Output: divided trajectory count set C
(1) Initialize $Error(C)$
(2) Computing difference set d according to *SortSet*
(3) $\varepsilon = \varepsilon/2$, select d_i with the probability proportional to $e^{(\varepsilon \cdot u(D, d_i))/2\Delta u}$, and group C into $C = \{C_1, C_2\}$
(4) if $Error(C_1) + Error(C_2) < Error(C)$
save C_1, C_2, d=$d - \{d_i\}$,
repeat step (3) and (4) on C
(5) else (stop dividing)
(6) return C

Algorithm 3: TraDivision.

Table 3: Published trajectory dataset.

No.	Tra ID	Generalized Tra
1	$\widetilde{T}_1$	$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$
2	$\widetilde{T}_1$	$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$
3	$\widetilde{T}_1$	$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$
4	$\widetilde{T}_2$	$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{32}$
5	$\widetilde{T}_3$	$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{31}$
6	$\widetilde{T}_4$	$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$
7	$\widetilde{T}_4$	$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$
8	$\widetilde{T}_4$	$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$

In this algorithm's line (3), $\varepsilon = \varepsilon/2$ means that, in each iteration, the differential privacy mechanism consumes half of the remaining privacy budget in previous step. Because we do not know the exact running times of the process, we can limit the whole budget privacy less than ε.

Both Algorithms 2 and 3 are just part of the differential privacy data publishing algorithm for trajectory generated. The complete algorithm for trajectory generated is as shown in Algorithm 4.

After processing by Algorithm 4, the format of the published trajectory dataset is as Table 3.

4.4. Privacy Analysis. In this section, we analyze the privacy attribute of the algorithm proposed in this paper. The main algorithm of this paper consists of two parts. The first part

Input: $\widetilde{L} = \{\widetilde{L_1}, \widetilde{L}_2, \cdots \widetilde{L_{|T|}}\}$, privacy budget ε
Output: published dataset $\overline{D}$
(1) Initialize $\varepsilon = \varepsilon_1 + \varepsilon_2$, $\overline{D} = \Phi$
(2) Based on $\widetilde{L}$, use the centers of each area to replace original trajectories, making generalized $\widetilde{D}$
(3) *SortSet*=CountSort($\widetilde{D}, \varepsilon$)
(4) C=TraDivision($Sortset, \varepsilon_1$)
(5) For C_i in C, the noisy count c of trajectories in C_i
$c = aver(C_i) + Lap(1/|C_i|\varepsilon_2)$;
add trajectories in C_i into $\overline{D}$
(6) return published dataset $\overline{D}$

Algorithm 4: 2DPA-TG($\widetilde{L}, \varepsilon$).

is the differential privacy spatial partition algorithm. When designing the algorithm, we use the exponential mechanism to guarantee differential privacy. The privacy budget we assign to this algorithm is ε_d for each of the location sets of each timestamp. Then we execute algorithm sequentially. According to Theorem 6 (sequential composition), we have the following theorem.

Theorem 10. *The differential privacy spatial partition algorithm guarantees $|T| \cdot \varepsilon_d$-differential privacy across the entire trajectory dataset.*

The second part differential privacy data publishing algorithm for trajectory generated consists of two subalgorithms, in which there are two parts consuming privacy budget: (i) Algorithm 3 TraDivision uses the exponential mechanism to select the maximum adjacent count difference and (ii) Algorithm 4's line (4) uses the Laplace mechanism to add noise on trajectory counts. Algorithm 3 consumes half of the remaining privacy budget ε each time, so the total privacy budget is $\sum_{i=1}^{k}(1/2^i) \cdot \varepsilon_1 < \varepsilon_1$, where k is the total number of iterations when the algorithm stops.

Line (4) of Algorithm 4 consumes a privacy budget ε_2 for one-time consumption. Then we have the following theorem.

Theorem 11. *Differential privacy data publishing algorithm for trajectory generated satisfies ε_p-differential privacy, $\varepsilon_p = \varepsilon_1 + \varepsilon_2$.*

So, as the sequential property of differential privacy, we have the theorem as follows.

Theorem 12. *The protection mechanism we propose in this paper satisfies ε-differential privacy, and $\varepsilon = |T| \cdot \varepsilon_d + \varepsilon_p$.*

5. Evaluation

5.1. Dataset and Environment. The experimental dataset of this paper is published by Microsoft Research Asia, which contains the trajectories of 10,357 taxis in Beijing in a week. Each record in the dataset is like (taxi_id, time, longitude, latitude) and the GPS sample frequency of each taxi ranges from 1s to 10min. In order to fit the algorithm design and facilitate the experiment, we extracted the data from 6:00 to 12:00 in one day, and the sampling frequency of each data is 10min, which means that every spatiotemporal point on the trajectory has 10min interval. After processing the raw dataset, we obtained original experimental data containing 7,369 taxi trajectories.

The experimental hardware environment is Windows 7, Intel Core i5 6500 CPU, 3.2 GHz, and 8G memory, and the experiment is programmed by Java.

5.2. Utility Metric. The algorithm proposed in this paper causes the loss of usability between the published dataset and the original dataset in two aspects. The first is the loss caused by division and replacement of the location points at each timestamp and the second is the error caused by Laplace noise and exponential perturbation. Therefore, in this section, we evaluate availability of the published trajectory dataset by Hausdorff distance [48] and spatiotemporal range query [49].

Hausdorff distance is a way to measure the degree of similarity between two point sets. The spatiotemporal trajectory can also be seen as a form of point set, so the Hausdorff distance measurement can be used.

The Hausdorff distance between T_i and T_j is calculated by the following formula:

$$HD\left(T_i, T_j\right) = \max\left(hd\left(T_i, T_j\right), hd\left(T_j, T_i\right)\right)$$

$$hd\left(T_i, T_j\right) = \max\left(\left(\min\left\|p_i^m, p_j^n\right\|, \ p_j^n \in T_j\right), \ p_i^m \in T_i\right) \tag{15}$$

$hd(T_j, T_i)$ is the same as above. Obviously, the smaller the Hausdorff distance, the higher the usability.

We use the DPR algorithm in [44] as a comparison experiment. DPR is similar to the algorithm framework of our proposal. The DPR algorithm uses k-means clustering for each timestamp and finally publishes the dataset with Laplace noise and generated fake trajectories. It should be noted that the DPR algorithm needs to specify k as the number of cluster clusters. In our paper, the required k is the minimum number of location points in each partition area.

We calculate the Hausdorff distance between the original trajectory dataset and the published trajectory dataset. In Figure 7, at first, as ε increases, the Hausdorff distance tends

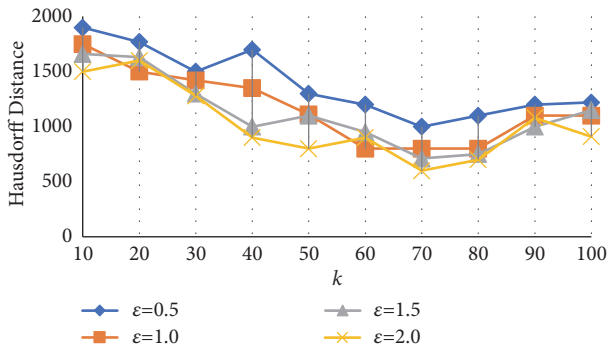


Figure 7: Hausdorff distance varies with k.

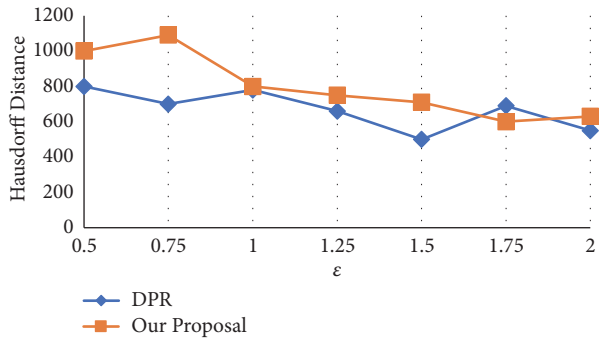


Figure 8: DPR versus our proposal in Hausdorff distance.

to decrease. The reason is that when ε increases, Laplace noise and exponential noise will decrease, which make the difference between published dataset and the original dataset smaller. With the increase of k, Hausdorff distance decreases first and then increases. This is because k has an influence on the location set L_i in Algorithm 1 and number of generalized trajectories in Algorithm 4. After the value of k is too large, the partition area is increased, and the distance between center of L_i and other points in L_i is increased, thereby affecting the availability of the entire dataset. When k=70, the Hausdorff distance gets its minimum, so we fix k=70 to perform the subsequent experiments.

We experimentally compared the trajectory dataset released by the DPR algorithm with the trajectory dataset published by our algorithm. It can be seen from Figure 8 that the Hausdorff distance of the DPR algorithm is a little smaller than our algorithm in most cases. This is because the DPR algorithm uses k-means based clustering algorithm when dealing with location point sets at each timestamp. Such algorithms tend to choose nearest neighbors as the same category, with higher accuracy in Euclidean distance based comparison method. But considering that the time complexity of k-means algorithm is $O(knmt)$ and the DPR is $O(n^2)$, while our proposal is $O(n \log n)$, it is perfectly acceptable to sacrifice a small amount of Hausdorff distance in exchange for time complexity.

Spatiotemporal range query is a method for measuring trajectory data quality proposed in [49]. In the experiment, we, respectively, use the algorithm proposed by this paper and DPR algorithm to publish dataset $\overline{D}$ and then perform two kinds of spatiotemporal range query on D and $\overline{D}$. We calculate the distortion of the query results of the two algorithms as follows:

$$loss\left(D, \overline{D}\right) = \frac{\left|Q\left(D\right) - Q\left(\overline{D}\right)\right|}{\max\left(Q\left(D\right), Q\left(\overline{D}\right)\right)} \tag{16}$$

For better comparison with the existing work, we choose two types of spatiotemporal range queries, namely, *PSI* query and *DAI* query.

$PSI(Possibly\_Sometimes\_Inside(T, R, t_s, t_e))$ query is what might happen in a certain period of time, which means count trajectory T that might appear in the area R during time $[t_s, t_e]$.

$DAI(Definitely\_Always\_Inside(T, R, t_s, t_e))$ query is what must happen in a certain period of time, which means count trajectory T that definitely appears in the area R during time $[t_s, t_e]$.

We generate two sets of queries Q_1 and Q_2:

$$Q_1 = selectcount(*)\ from\ D\ where\ PSI(T, R, t_s, t_e)$$
$$Q_2 = selectcount(*)\ from\ D\ where\ PSI(T, R, t_s, t_e)$$

According to the parameter settings in [23], we set R to a circular area of 1000 m and 500 m and the time interval $[t_s, t_e]$ is two hours. We generate 1000 queries and the experimental results are shown in Figure 9.

These experiments verified the degree of distortion of *PSI* and *DAI* query when the radius of the area is 500 m or 1000 m. As can be seen from Figure 9, when the privacy budget ε increases, the distortion of both algorithms will gradually decrease, which is in line with our intuitive speculation and actual situation, because, with the increase of privacy budget ε, the degree of privacy is reduced and the injection noise is also reduced. Then data will be closer to the real data; thus the usability is improved. Take Figure 9(a) as an example, when we perform this kind of query $select\ count(*)\ from\ D\ where\ PSI(T, R, t_s, t_e)$ on $\overline{D}$ and D, with the increasing of privacy budget ε, the query distortion of our algorithm is gradually reduced and is always smaller than the distortion of the DPR algorithm.

Through the results in Figure 9, we can see that the data publishing algorithm proposed in this paper is stronger than the DPR algorithm in spatiotemporal range query, because, in order to ensure sufficient privacy, the DPR algorithm uses a fake trajectory-based method to add fake trajectories into published dataset. Even though the DPR's k-means clustering stage is better than our algorithm, there are not only the Laplace noise but also the disturbance error caused by fake trajectory, while our algorithm has no such drawbacks. In Figure 10, we compare the time efficiency between DPR and our method. We implement the experiment on parameter k, which is the location number in each partition. As k increases, the time consumed in process decreases simultaneously, because, with k growing, the partition in each timestamp will decrease, which make generalized trajectories less. So, the time based on sort and group these trajectories will be less than before. The result shows that our method is better

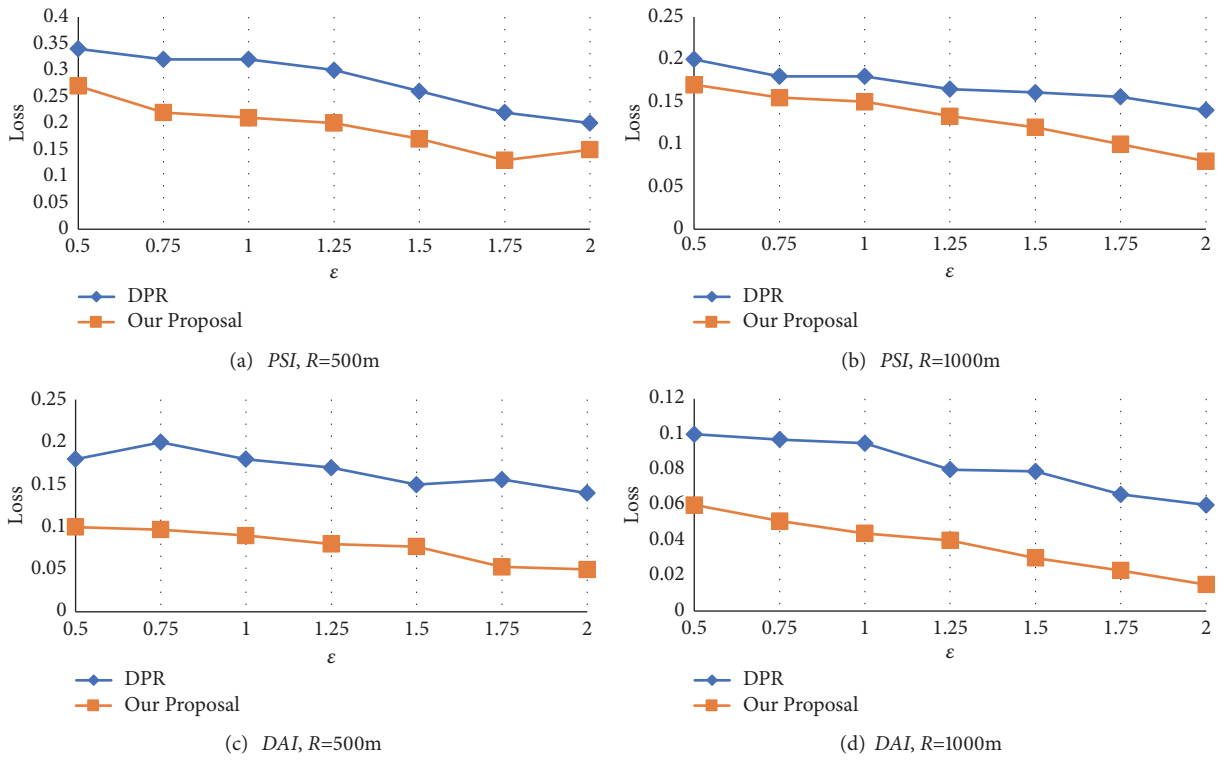


Figure 9: Query results of *PSI* and *DAI* with R=500m and R=1000m.

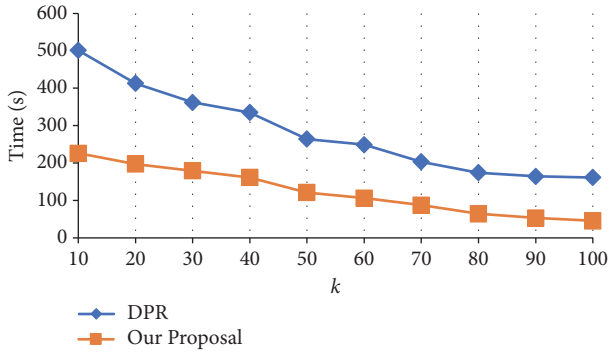


Figure 10: DPR versus our proposal in time efficiency.

than DPR method because our time complexity is $O(n \log n)$ as we mentioned before. Therefore, from comprehensive effect considerations, performance of our method on these experiments proves that our proposal is more practical and better than the DPR algorithm.

6. Conclusion

In this paper, we propose a spatial partition based trajectory dataset publishing algorithm. This algorithm satisfies differential privacy with high utility and can run in less time than existing method. The algorithm uses exponential mechanism to output more accurate location point partitions and trajectory count group to ensure privacy and then release data after adding Laplace noise into trajectory count. According to our knowledge, this is the first paper to use Hilbert curve to divide amount of trajectories in a noninteractive way. At last, we perform spatiotemporal range queries on real trajectory dataset, and the results are better than existing algorithm. Besides, the published data can achieve smaller Hausdorff distance within the tolerance. The two experiments show that our method can be used in practice effectively. Our method deserves continuous study in the future: before running the algorithm, we preprocess the raw dataset to get the original dataset with same time interval. In fact, the collected GPS information will not be completely regular, and the sampling interval may change randomly or even have an interruption. So how to publish such irregular dirty datasets is a hard problem we need to study in our future work.

Data Availability

The trajectory data used to support the findings of this study can be downloaded from https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/ and the detailed instructions can be found in https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/User_guide_T-drive.pdf.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper is a fundamental research and is supported by Funds for the Central Universities (Grant no. HEUCF180603) and Harbin Application Technology Research and Development Project (Grant no. 2016RAQXJ063 and Grant no. 2016RAXXJ013).

References

[1] H. Zheng and M. Xiaofeng, "Research on trajectory privacy protection technology," *Journal of computer science*, vol. 34, no. 10, pp. 1820–1830, 2011.

[2] Y. Huo, C. Yong, and Y. Lu, "Re-ADP: real-time data aggregation with adaptive w-event differential privacy for fog computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6285719, 13 pages, 2018.

[3] K. Zhang, Q. Han, Z. Cai, and G. Yin, "Rippas: A ring-based privacy-preserving aggregation scheme in wireless sensor networks," *Sensors*, vol. 17, no. 2, pp. 1–19, 2017.

[4] Q. Han, S. Liang, and H. Zhang, "Mobile cloud sensing, big data, and 5G networks make an intelligent and smart world," *IEEE Network*, vol. 29, no. 2, pp. 40–45, 2015.

[5] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science & Engineering*, vol. 99, 2018.

[6] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Communications Magazine*, vol. 5, no. 8, pp. 33–43, 2018.

[7] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart IoT systems: a consideration from privacy perspective," *IEEE Communications Magazine*, vol. 10, no. 2, pp. 12–20, 2018.

[8] Z. Cai, Z. He, and X. Guan, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable & Secure Computing*, p. 99, 2016.

[9] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, p. 99, 2017.

[10] X. Zheng, Z. Cai, G. Luo, L. Tian, and X. Bai, "Privacy-preserved community discovery in online social networks," *Future Generation Computer Systems*, 2018.

[11] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS '15)*, pp. 205–214, July 2015.

[12] Q. Han, D. Lu, K. Zhang, X. Du, and M. Guizani, "Lclean: a plausible approach to individual trajectory data sanitization," *IEEE Access*, vol. 6, pp. 30110–30116, 2018.

[13] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM '17)*, pp. 1–9, Atlanta, GA, USA, May 2017.

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

[15] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "*L*-diversity: privacy beyond *k*-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, 2007.

[16] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: privacy beyond k-anonymity and *l*-diversity," in *Proceedings of the 23rd International Conference on Data Engineering*, pp. 106–115, IEEE, Istanbul, Turkey, April 2007.

[17] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith, "Composition attacks and auxiliary information in data privacy," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 265–273, August 2008.

[18] R. C. Wong, A. W. Fu, K. Wang, P. S. Yu, and J. Pei, "Can the utility of anonymized data be used for privacy breaches?" *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 3, pp. 1–24, 2011.

[19] C. Dwork, "Differential privacy," in *Proceedings of the International Colloquium on Automata, Languages, and Programming*, vol. 4052, pp. 1–12, Springer, Berlin, Heidelberg, 2006.

[20] X. Ping, Z. Tianqing, and W. Xiaofeng, "Differential privacy protection and its application," *Journal of Computer Science*, vol. 37, no. 1, pp. 101–122, 2014.

[21] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS '14)*, pp. 251–262, November 2014.

[22] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Trajectory anonymity via clustering," ISTI-CNR, Tech. Rep. ISTI, 2007.

[23] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE '08)*, pp. 376–385, April 2008.

[24] Z. Jing and Z. Yuan, "Qinghua LTrajectory privacy preserving method based on trajectory frequency suppression," *Journal of Computer Science*, vol. 37, no. 10, pp. 2096–2106, 2014.

[25] M. Terrovitis and N. Mamoulis, "Privacy preservation in the publication of trajectories," in *Proceedings of the 9th International Conference on Mobile Data Management (MDM '08)*, pp. 65–72, April 2008.

[26] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2nd International Conference on Pervasive Services (ICPS '05)*, pp. 88–97, IEEE Press, July 2005.

[27] N. Mohammed, B. C. M. Fung, and M. Debbabi, "Walking in the crowd: Anonymizing trajectory data for pattern analysis," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*, pp. 1441–1444, Hong Kong, China, November 2009.

[28] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: Optimal strategy against localization attacks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '12)*, pp. 617–626, October 2012.

[29] G. Theodorakopoulos, R. Shokri, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Prolonging the hide-and-seek game: optimal trajectory privacy for location-based services," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES '14)*, pp. 73–82, 2014.

[30] R. Chen, B. C. M. Fung, and C. B. Desai, "Differentially private trajectory data publication," *Computer Science*, 2011.

[31] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Theory of Cryptography Conference (TCC '06)*, vol. 3876, pp. 363–2385, New York, NY, USA, March 2006.

[32] F. Mcsherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pp. 94–103, IEEE Computer Society, 2007.

[33] X. Liu and G. F. Schrack, *An Algorithm for Encoding and Decoding The 3-D Hilbert Order*, IEEE Press, 1997.

[34] L. U. Feng, "A GIS spatial indexing approach based on Hilbert ordering code," *Journal of Computer Aided Design & Computer Graphics*, 2001.

[35] J. Liu, Y. Pan, M. Li et al., "Applications of deep learning to MRI images: A survey," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 1–18, 2018.

[36] G. S. Yadav and A. Ojha, "A scalable data hiding scheme using hilbert space curve and chaos," in *Proceedings of the IEEE Trustcom/BigDataSE/ISPA*, pp. 905–909, Helsinki, Finland, August 2015.

[37] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, 2007.

[38] H. Ngo and J. Kim, "Location Privacy via Differential Private Perturbation of Cloaking Area," in *Proceedings of the 28th IEEE Computer Security Foundations Symposium (CSF '15)*, pp. 63–74, July 2015.

[39] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.

[40] B. Zhou, J. Li, X. Wang et al., "Online Internet traffic monitoring system using spark streaming," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 47–56, 2018.

[41] C. G. Peng, H. F. Ding, Y. J. Zhu, Y. L. Tian, and Z. F. Fu, "Information entropy models and privacy metrics methods for privacy protection," *Journal of Software*, vol. 27, no. 8, pp. 1891–1903, 2016.

[42] C. E. Shannon and W. Weaver, "The mathematical theory of communication," *Physics Today*, vol. 3, no. 9, pp. 31-32, 1950.

[43] Z. Zhijie, P. Wenxiang, and Z. Yibiao, "Description and application of discrete trend in spatial point pattern analysis," *Chinese Health Statistics*, vol. 25, no. 5, pp. 470–473, 2008.

[44] J. Hua, Y. Gao, and S. Zhong, "Differentially private publication of general time-serial trajectory data," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks (IEEE INFOCOM '15)*, pp. 549–557, May 2015.

[45] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[46] N. Adrienko and G. Andrienko, "Spatial generalization and aggregation of massive movement data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 205–219, 2011.

[47] Q. Han, B. Shao, L. Li, Z. Ma, H. Zhang, and X. Du, "Publishing histograms with outliers under data differential privacy," *Security and Communication Networks*, vol. 9, no. 14, pp. 2313–2322, 2016.

[48] C. Yanjun, *Research on Clustering Algorithm for Mass Trajectory Data*, Beijing Jiaotong University, 2015.

[49] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain, "Managing uncertainty in moving objects databases," *ACM Transactions on Database Systems*, vol. 29, no. 3, pp. 463–507, 2004.