International Journal of Geo-Information

MDPI

Article

A Trajectory Privacy Protection Method Based on Random Sampling Differential Privacy

Tinghuai Ma [1,†,‡] **and Fagen Song** [1,2,*,‡]

[1] School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China; thma@nuist.edu.cn
[2] Yancheng Institute of Technology, Yancheng 224051, China
* Correspondence: songfagen@ycit.cn
† Current address: Nanjing 210044, China.
‡ These authors contributed equally to this work.

Abstract: With the popularity of location-aware devices (e.g., smart phones), a large number of trajectory data were collected. The trajectory dataset can be used in many fields including traffic monitoring, market analysis, city management, etc. The collection and release of trajectory data will raise serious privacy concerns for users. If users' privacy is not protected enough, they will refuse to share their trajectory data. In this paper, a new trajectory privacy protection method based on random sampling differential privacy (TPRSDP), which can provide more security protection, is proposed. Compared with other methods, it takes less time to run this method. Experiments are conducted on two real world datasets to validate the proposed scheme, and the results are compared with others in terms of running time and information loss. The performance of the scheme with different parameter values is verified. The setting of the new scheme parameters is discussed in detail, and some valuable suggestions are given.

Keywords: trajectory privacy protection; differential privacy; K-anonymity; exponential mechanism; laplace mechanism

Citation: Ma, T.; Song, F. A Trajectory Privacy Protection Method Based on Random Sampling Differential Privacy. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 454. https://doi.org/10.3390/ijgi10070454

Academic Editor: Wolfgang Kainz

Received: 17 April 2021
Accepted: 24 June 2021
Published: 1 July 2021

1. Introduction

Due to the development of information technology, especially the popularization of intelligent equipment (e.g., smart phones), it is much easier to collect a user's data, including trajectory data [1,2]. This data is a valuable resource. This data, including the trajectory data, may subsequently be uploaded to various service providers after user permission is granted. On the one hand, the rational use of this data can further improve travel comfort and user satisfaction. Moreover, this data can also be applied to traffic monitoring, market analysis, urban management and other fields [3–5]. On the other hand, without proper protection methods, the trajectory dataset may reveal a user's privacy. With the help of auxiliary information, attackers can infer user identity, interests, habits, religious beliefs, political opinions and other information based on the trajectory dataset [6,7]. It is highly important to protect private information in trajectory data, and the privacy protection method should not compromise the data availability [8].

Many privacy protection models have been proposed by researchers, such as K-anonymity [9] and differential privacy [10]. K-anonymity is widely used in the privacy protection field. However, it cannot resist background knowledge attack. Differential privacy makes no assumption about the users' background knowledge and also supplies a quantitative analysis of privacy breach risk. Unfortunately, it is hard to get a good result if differential privacy is directly applied to the original trajectory dataset [11].

Two problems must be addressed before using the differential privacy to protect the trajectory privacy. The first one is how to improve the efficiency. Trajectory data is a special kind of big data, and the computational cost of differential privacy is large. If the efficiency

is low, it will take a lot of time to do differential privacy transformation. In our scheme, a random sampling process is added to improve efficiency. The second problem is how to reduce the amount of information loss. The trajectory data is a time series of location records, each combination of location and time can be used as a quasi-identifier. Any modification of the original data will result in a loss of information. In our scheme, both the Laplace mechanism and the exponential mechanism are used. We use fake location near the cluster center instead of the center of the cluster to replace the real location, which can reduce the amount of information loss.

Montjoye et al. pointed out in [12,13] that over 90% of the trajectories can be reidentified using no more than four locations. It is very urgent to design an effective method to protect the trajectory privacy. The contributions of this paper are summarized as follows.

- An efficient trajectory privacy protection method is proposed in this paper. Different from others, there is an additional random sampling process in this scheme. The random sampling process can greatly reduce the amount of records which will be used to divide the original dataset, and this will significantly improve the efficiency of this scheme.
- Our scheme can provide more privacy protection without increasing the amount of information loss. Both Laplace mechanism and Exponential mechanism are used in our method, which can provide more protection for privacy. Exponential mechanism is used to select an approximate optimal partition from all the partition results, and Laplace noises are added to the count of trajectories of each partition. Different from others, the fake location records are not replace by the cluster centers. Those records are generated randomly near the cluster centers. If the parameter setting is reasonable, the scheme can provide more protection than K-anonymous.
- Experiments are conducted on two real-world datasets, and the results show that our scheme is superior to others. Specially, the loss of information is no more than that of others, and the efficiency is much higher than that of other schemes. The setting of the system parameter is discussed in detail, and some pieces of advice are given.

The remainder of this manuscript is organized as follows. Section 2 briefly reviews the related literature. In Section 3, the preliminaries are introduced. Our method is proposed in Section 4, and the experimental results are described in Section 5. The setting of the system parameter is discussed in detail in Section 6, and finally conclude the paper in Section 7.

2. Related Work

Existing trajectory publishing mechanisms can be classified into two types [1,14,15]. The first one is used for publishing a set of trajectories, and each trajectory is regarded as one record. The second type publish one trajectory and each position in the trajectory is regard as one record [1,15,16]. In this paper, we will focus on the first type of trajectory data publishing. Privacy-preserve trajectory data release mainly divide into anonymous [6,17,18], suppression [17], data encryption [19–23], random perturbation [14,24–26] and others [27]. Cryptography can provide security protection for any type of information. However, encryption usually takes a long time, and ciphertext greatly limits the use of data. The scheme proposed in this paper can protect trajectory privacy without encrypting for data.

Researchers have done a lot of works on trajectory privacy protection and got many valuable results [28]. Most of the trajectory publishing mechanisms are partition-based privacy models. Differential privacy [14,24–26] and K-anonymity(including l-diversity and t-closeness) [6,17,18,29] are two important privacy protection methods without encrypting data. K-anonymity makes at least k trajectories indistinguishable by clustering or generalization. K-anonymity is easy to implement, and the damage to the original data is relatively small [17]. Therefore, many trajectory privacy protection schemes are designed based on K-anonymity [24,30]. In [31], Abul et al. proposed the K-anonymity model which is used for preserving location privacy of moving objects. It requires that there are at least $k-1$ other trajectories in the same uncertainty region which are indistinguishable from each other, where k represents the number of indistinguishable records. In [32],

K-anonymity is achieved by sensitive attribute generalization and trajectory local suppression. Xin et al. proposed a new K-anonymity which can be used in dynamic datasets in [6]. In [33], an efficient method for finding the desired anonymity set is proposed in the GeoSpark environment. The security of k-anonymity method is reinforced by dual transformation in [34].

However, K-anonymity cannot provide sufficient privacy protection and is vulnerable to attack [35–37]. It cannot provide protection against background knowledge attacks. Differential privacy model [1,29], which is one kind of randomization-based privacy model, is quickly applied in the field of trajectory privacy protection. This model makes no assumption about the adversary's background knowledge [15,25,26], and its security can be proved mathematically. In [14], Jiang K et al. compared three differential privacy mechanisms by adding noise to the whole trajectory, adding noise to each position and adding noise to each coordinate, respectively. The experimental results in [14] show that adding noise to each position is superior to the others. In this paper, we will focus on the first type of trajectory data publishing, and the differential privacy transformation will be done by adding noise to each position. In [38], Chen et al. first apply differential privacy model to trajectory publishing. A noisy prefix tree, which groups the sequences with the same prefixes into the same branch, is built. With the growth of the prefix tree, there will be a large number of leaf nodes, which need to add a lot of noise to achieve differential privacy. However, adding too much noise will greatly reduce the availability of trajectory dataset. A multi-level query tree is used in [39]. Chen et al. use a variable n-gram model to improve the utility in [40]. The methods proposed in [38–40] are all data dependent sanitization mechanism. In [1], Hua J et al. proposed a differentially private publishing mechanism for more general time-series trajectories, which need not have the prefix tree or n-grams structure. Adding too much noise may make the trajectory data meaningless. In order to improve the utility of the dataset, limited noise is used in [15]. Yilmaz et al. proposed a new privacy-preserving mechanism based on differential privacy and homomorphic encryption [19]. However, the homomorphic encryption greatly reduces the efficiency of the scheme.

All the mechanisms discussed above, whether it is data dependent or not, have a common drawback, which is that the computational cost is large. The method proposed in this paper can improve the efficiency significantly without increase the loss of information. In [41], Li et al. pointed out that we can benefit from the adversary's uncertainty about the data, so as to improve the security of differential privacy. Their conclusion is that random sampling is a powerful tool to improve the security of K-anonymity and differential privacy. In our scheme, the trajectory dataset is divided into sub datasets according to the timestamp of each location record, and differential privacy transformation is carried out on each sub dataset. There is an additional random sampling process in our scheme. The random sampling process is mainly used to improve the efficiency, although it also enhances the security. The details will be described in Section 4.

3. Preliminaries

If there are more than k records indistinguishable in a dataset, we say that the dataset has achieved K-anonymity. K-anonymity was proposed by P. Samarati et al. in [9]. There are mainly two kinds of method to achieve K-anonymity, one is generalization, the other is suppression. Generalization technique makes the records indistinguishable from others by generalizing the attribute values of different records to a larger range of allowable values. Suppression technique makes the records indistinguishable from others by deleting records or replacing the record value with another. While K-anonymity is used to protect the privacy of location or trajectory dataset, the cluster center is usually used to replace the other records. In this paper, the original records are replaced by the fake location records which are generated randomly in a certain range around the center of the cluster.

Differential privacy proposed by C. Dwork in [10] has become one of de facto standard in the research field of privacy protection. It requires that modifying a single record should have a negligible effect on the query outcome. The formal definition is as follows.

Definition 1. *(ε-differential privacy). A randomized algorithm Ag is differentially private if and only if any two databases D' and D contain at most one different record (D' and D are neighbor datasets), and for any possible anonymized output $O \in Range(Ag)$*

$$Pr[Ag(D) = O] \leq e^{\varepsilon} \times Pr[Ag(D') = O] \tag{1}$$

where $Pr[*]$ is the probability that algorithm Ag outputs a certain value, and ε is the differential privacy budget. The smaller the value of ε, the stronger privacy protection can be provided by differential privacy. There are mainly two techniques for achieving differential privacy. One is Laplace mechanism [42], the other is Exponential mechanism [43].

Definition 2 (global sensitivity)**.** *For a given function $f : D \to R^d$, its global sensitivity is*

$$\Delta f = max_{D,D'} \parallel f(D) - f(D') \parallel \tag{2}$$

where D and D' are neighbor datasets (D' and D differ in a individual record).

Laplace mechanism is always used for the functions whose outputs are real. Proper Laplace noises are added to the real outputs to achieve differential privacy. The Laplace noises are generated according to a Laplace distribution $Lap(\mu, \Delta f/\varepsilon)$. The probability density function is $Pr(x|\mu, \Delta/\varepsilon) = \frac{1}{2*\Delta f/\varepsilon} e^{\frac{-|x-\mu|}{\Delta f/\varepsilon}}$ where μ is the mean of this distribution, and the value is always zero. Δf is the global sensitivity, and ε is the privacy budget. While the value of μ is zero, we use $Lap(\Delta f/\varepsilon)$ to denote $Lap(\mu, \Delta f/\varepsilon)$.

Theorem 1 ([43])**.** *For any function $f : D \to R^d$, the mechanism*

$$A(D) = f(D) + Lap(\Delta f/\varepsilon) \tag{3}$$

achieves $\varepsilon - differential\ privacy$.

Exponential mechanism proposed by Mcsherry F et al. in [43] is mainly used for the queries whose output values are non numeric. A score function $u : (D \times \tau) \to R$ is defined, and each output r is assigned a real value score. The probability of output result $r \in R$ is proportional to $e^{\frac{\varepsilon u(D,r)}{2\Delta u}}$, where $\Delta u = max_{\forall r,D,D'} |u(D, r) - u(D', r)|$ is the sensitivity of the score function. As a result, the outputs with higher scores will be more likely to be output.

Theorem 2 ([42])**.** *For any function $u : (D \times \tau) \to R$, if the mechanism chooses an output $r \in R$ (R is the output domain) with the probability proportional to $e^{\frac{\varepsilon u(D,r)}{2\Delta u}}$, the mechanism satisfies $\varepsilon - differential\ privacy$.*

There are two important properties of differential privacy. The first is named sequential composition. A sequence of differential privacy transformations are done on the same dataset independently. The whole transformation provides differential privacy, and the privacy budget is accumulated. Theorem 3 gives a formal description. The second is known as parallel composition. Several differential privacy transformations are done on disjoint sub dataset, respectively, the whole transformation also provides differential privacy, and the privacy budget is determined by the worst case. The formal description is shown in Theorem 4.

Theorem 3 ((sequential composition) [15]). *Let function f_i each provide differential privacy, and the privacy budget ε_i is, respectively. Then running in sequence all functions f_i over a database D provides $\sum_i \varepsilon_i - differential\ privacy$.*

Theorem 4 ((parallel composition) [15]). *Let function f_i each provide differential privacy, and the privacy budget is ε_i. Then applying each function over a set of disjoint databases D_i provide $max_i\{\varepsilon_i\} - differential\ privacy$.*

4. Trajectory Database and Privacy Protection Method

4.1. Method Overview

A trajectory is a trace history of one user. It is a sequence of time and location tuples. A trajectory is marked $T = (l_1, t_1) \rightarrow (l_2, t_2) \rightarrow \cdots \rightarrow (l_{|T|}, t_{|T|})$, where $|T|$ denotes the number of locations of T. l_i is a spatial point, and t_i is the timestamp of the location record (l_i, t_i). A user may arrive at the same point at different time, which means that one location in the same trajectory may appear more than one time. A trajectory database D contains many trajectories, and the size is denoted as $|D|$. Each record in D corresponds to one trajectory, and the length of different trajectory may be different.

There are three main steps in this scheme. First of all, trajectory dataset is divided into different sub datasets. In the same sub dataset, the timestamp value of each location record is the same. The second step is to divide sub dataset into different clusters. φ clustering results will be generated, and one of partition results will be selected according to the value of score function. Finally, Laplace noise is added to the count of different clusters. Specially, in the second step, there is an additional random sampling process in this scheme, and the method how to divide the original dataset is modified accordingly.

An example is shown in Figure 1. There are six trajectories marked $T_1, T_2, \cdots, T_6$ in the trajectory database, which is divided into three sub datasets according to the value of its timestamps. Random sampling is performed on the sub dataset to obtain the dataset s_{rs}. The dataset s_{rs} is clustered using K-means algorithm, and the original dataset is divided into several clusters according to the distance from the cluster centers of s_{rs}. For instance, suppose L_1, L_2, L_5 and L_6 are selected in sub dataset t_1. According to K-means clustering results, the region is divided into two partitions. L_1^1 and L_2^1 are the centers of the two partitions, respectively, and then all the records including L_3 and L_4 are classified into the two partitions according to the distance between the records and the partition centers. Finally, the original location records are generalized to random location records which are generated by adding noises to the cluster centers. The original database is shown in Table 1. After clustering and generalization, the final released database is shown in Table 2. T_2 and T_3 are generalized to NT_2. T_1, T_4, T_5 and T_6 are generalized to NT_1, NT_3, NT_4 and NT_5, respectively. The notation '*real_count*' in Table 2 represents the actual number of records in the cluster, and the '*noisy_count*' is the final released count, which is generated by adding Laplace noises to the '*real_count*'. We summarize the main symbols used in this study, as shown in Table 3.

Table 1. Original trajectory records.

ID.	Trajectories	ID	Trajectories
T_1	$(L_1, t_1) \rightarrow (L_9, t_2) \rightarrow (L_{15}, t_3)$	T_4	$(L_4, t_1) \rightarrow (L_{11}, t_2) \rightarrow (L_{16}, t_3)$
T_2	$(L_2, t_1) \rightarrow (L_7, t_2) \rightarrow (L_{14}, t_3)$	T_5	$(L_5, t_1) \rightarrow (L_{10}, t_2)$
T_3	$(L_3, t_1) \rightarrow (L_8, t_2) \rightarrow (L_{17}, t_3)$	T_6	$(L_6, t_1) \rightarrow (L_{12}, t_2) \rightarrow (L_{13}, t_3)$

Table 2. Noisy trajectory records.

ID.	Trajectories	Real_Count	Noisy_Count
NT_1	$(L_1^1, t_1) \rightarrow (L_1^2, t_2) \rightarrow (L_1^3, t_3)$	1	0.811
NT_2	$(L_1^1, t_1) \rightarrow (L_1^2, t_2) \rightarrow (L_2^3, t_3)$	2	2.50
NT_3	$(L_2^1, t_1) \rightarrow (L_2^2, t_2) \rightarrow (L_2^3, t_3)$	1	1.30
NT_4	$(L_2^1, t_1) \rightarrow (L_1^2, t_2)$	1	0.93
NT_5	$(L_2^1, t_1) \rightarrow (L_2^2, t_2) \rightarrow (L_1^3, t_3)$	1	0.97

Table 3. Summary of notations.

Notation	Description	Notation	Description
T	The original trajectory dataset	P_{opt}	The selected partition result
T'	The sanitized trajectory dataset	R	A dataset of partition result
s_i	Sub dataset of T divided according to the value of timestamp	c_i	The i-th cluster
s_i'	The noisy dataset of s_i	sc_i	The cluster center of the i-th cluster
P_k	One partition result of s_i	ε	The privacy budget

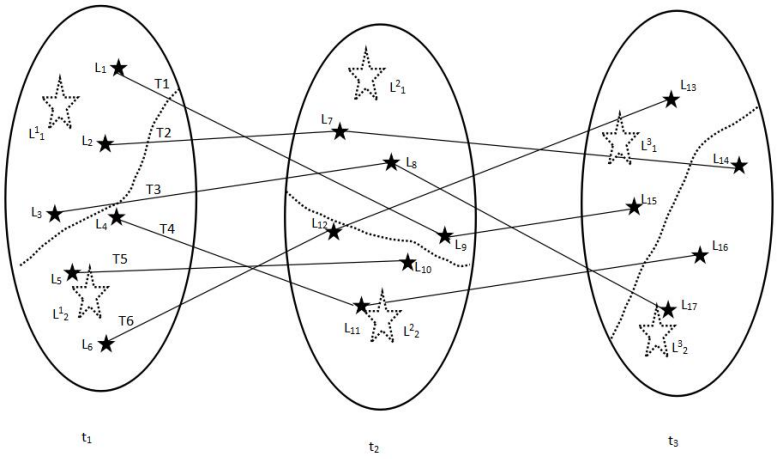

Figure 1. An example of our method.

4.2. Privacy Protection Method

The trajectory privacy protection method based on random sampling differential privacy (TPRSDP) is described in detail in Algorithm 1. The travel time is divided into time slice. In the first line, the original dataset is divided into sub dataset $\{s_1, s_2, \cdots, s_m\}$ according to the timestamp value of each record. The timestamp t of each location record in s_i has the same value. By dividing the original trajectory dataset into sub dataset, the trajectory dataset which is a sequence of positions is transformed into common dataset. This method which is very popular in the research field can be used on all kinds of trajectory dataset. In other words, this method is data independent.

In the following, the transformation will be done on different s_i $(i = 1 \cdots m)$, respectively. In the third line, the *RScluster* which will be introduced in the following, is called and the partition results P_0 is generated. Based on P_0, φ sub optimal partition results are generated by *sub_optimal* in line 4. In the fifth line, differential privacy is achieved by Exponential mechanism, and one partition result P_{opt} is selected. From line 6 to line 10, all the location records in each cluster are sanitized. The methods *sub_optimal* and *noisy_cluster* will be introduced in the following section. Finally the sanitized trajectory dataset is returned.

Algorithm 1 trajectory privacy protection method based on random sampling differential privacy (TPRSDP).

Input: The original trajectory dataset T

Output: The sanitized dataset $T' = \{s'_1, s'_2, \cdots, s'_m\}$

Procedure:

1 the original database T is divided into sub dataset $T = \{s_1, s_2, \cdots, s_m\}$, $T' = \emptyset$

2 for each s_i in $\{s_1, s_2, \cdots, s_m\}$ do

3 $P_0 = RScluster(s_i)$

4 $R = \{P_0, P_1, \cdots, P_{\varphi-1}\} = sub\_optimal(P_0)$

5 use exponential mechanism to select a partition result $P_{opt} = \{c_1, c_2, \cdots, c_k\}$ from R according to the value of score function

6 $s'_i = \emptyset$

7 for c_i in P_{opt}

8 Add Laplace noise to the count of c_j

9 $c'_j = noisy\_cluster(c_j)$

10 $s'_i = s'_i \cup c'_j$

11 endfor

12 add s'_i to T'

13 endfor

14 returnT' and noisy counts

4.2.1. Clustering the Sub Dataset

Clustering a large database is a time-consuming process. In this paper, there is an additional process of random sampling. The clustering is performed on the subset which is obtained by random sampling. We use random sampling to select records from the original dataset. The probability that each record in the original data is selected is equal, which means that the distribution of the subset is nearly the same as the distribution of the original dataset. We assume that there are n records in the dataset. According to the value of each record, all the records are assigned to k buckets. There are n_k records in the k-th bucket. We select one record from the original dataset. The probability that a record from the k-th bucket is $\frac{n_k}{n}$. If there are m records in the subset, there will be $m * \frac{n_k}{n}$ records selected from the k-th bucket. In the subset, the ratio of the number of records in each bucket is the same as that in the original dataset, which means that the data in the original dataset and the subset have the same distribution. It is reasonable to divide the original dataset according to the cluster center of the subset.

The number of records in the subset is much smaller than that in the original dataset, and the clustering efficiency will be significantly improved. The original dataset will be divided into different cluster according to the distance between the record to the subset cluster center. Herein, different from others, we use the center of the subset to divide the original dataset. The details are shown in Algorithm 2.

In line 1 of Algorithm 2, l records are selected from s_i, and how to set the value of l will be discussed in following section. In [1,15], the K-means is run on the original dataset, and one initial clustering result is got. However, In this paper, K-means is run on the random sampling subset, and the subset is divided into k clusters in line 2. The original dataset is divided into k clusters according to the distance between the location records and the k cluster centers of random sampling subset in line 3. Finally the initial partition result is returned.

Algorithm 2 RScluster.

Input: The original dataset s_i

Output: $P_0 = \{c_1, c_2, \cdots, c_k\}$

Procedure:

1 random select l locations records from s_i, $l << |s_i|$

2 run K-means on the l records, and k cluster centers $\{sc_1, sc_2, \cdots, sc_k\}$ are generated

3 redivided s_i into $\{c_1, c_2, \cdots, c_k\}$ according to the distance between the record and the k cluster centers $\{sc_1, sc_2, \cdots, sc_k\}$

4 return $P_0 = \{c_1, c_2, \cdots, c_k\}$

If there are n records in the dataset s_i, the total number of partition results will be k^n. It is very large, and it is infeasible to find out all the partition results. Inspired by the method in [1] (INFOCOM15) and [15] (INFOSCI17), we proposed a new method to reduce the partition number from k^n to φ. The other $\varphi - 1$ partition results are generated based on the initial partition result $P_0 = \{c_1, c_2, \cdots, c_k\}$. First of all, one cluster is selected randomly from the K-means partition result, then modify the cluster to generate new partition results. Herein, modifying the cluster is to delete a trajectory records from the cluster randomly or to move one trajectory record from one cluster to another. Do this until the other $\varphi - 1$ partition results are generated. Different from the method proposed in [1,15], when we modify different clusters, different trajectory records are selected for each cluster. The details are shown in Algorithm 3. One new partition result is generated by the inner loop of Algorithm 3, and the partition result is added to R in line 7. Finally, R containing φ partition results is returned.

Algorithm 3 *sub_optimal*.

Input: $P_0 = \{c_1, c_2, \cdots, c_k\}$

Output: $R_0 = \{P_1, P_2, \cdots, P_{\varphi-1}\}$

Procedure:

1 $R = \{P_0\}$

2 for $i = 1$ to $\varphi - 1$

3 copy $P_0 = \{c_1, c_2, \cdots, c_k\}$ to $temp$

4 for $j = 1$ to k

5 Select one trajectory in c_j(c_j is one element of $temp$), move the location records to c_t, where t is a random integer, $1 \leq t \leq k$.

6 endfor

7 add $temp$ to R

8 endfor

9 return R

4.2.2. Score Function

The score function plays a very important role in our scheme. As described in Line 5 of Algorithm 1, one partition result will be selected according to the value of the score function. The score function should output higher value for the reasonable partition result, and vice versa. Herein, a reasonable partition result means that the average distance between the records in the same cluster must be as small as possible. The score function we used is based on Euclidean distance. The score function is defined as Formulas (4)–(6).

The location records in each sub dataset s_i are divided into k cluster. $AvgDist^k_{c^i_j}$ denote the average inner distance of the j-th cluster, and the definition is as following:

$$AvgDist^k_{c^i_j} = \frac{2}{(|c^i_j| \cdot (|c^i_j| - 1))} \cdot \sum_{\forall l_1, l_2 \in c^i_j} Distance(l_1, l_2) \tag{4}$$

where $|c^i_j|$ represents the number of locations in cluster c^i_j. $distance(l_1, l_2)$ is the Euclidean distance between l_1 and l_2. We further define the average inner distance of all the k clusters as:

$$AvgDist_{p_i} = \frac{1}{k} \sum_{j=1}^{j=k} AvgDist_{c^i_j} \tag{5}$$

The smaller value of $AvgDist_{p_i}$, the better of the partition result. Smaller value of $AvgDist_{p_i}$ indicates that the closer points are divided into the same cluster, which means that the partition result is reasonable.

Based on $AvgDist_{p_i}$, the score function of each partition result p_i is defined as:

$$u(p_i) = 1 - \frac{AvgDist_{p_i}}{sum_{k=0,\cdots,k=\phi-1} AvgDist_{p_k}} \tag{6}$$

The global sensitivity of u is : $\Delta u = \max |(u(p_i) - u(p_j)|$. Suppose the differential privacy is ε, the probability that partition result $p_i (i = 1, 2, \cdots, m)$ is selected is:

$$Pr(q(R) = p_i) = \frac{\exp\left(\frac{\varepsilon_1}{2\Delta u} u(p_i)\right)}{\sum_{j=1,\cdots,m} \exp\left(\frac{\varepsilon_1}{2\Delta u} u(p_j)\right)} \tag{7}$$

4.2.3. Generating Noisy Clusters

One of the partition results will be selected according to the value of score function. Each cluster of the selected partition result will be generalized. Algorithm 4 describes the details of the generalized method. There are mainly two steps in Algorithm 4. The first step is to generalize the location records in the same cluster into the center of the cluster. The second step is to generate noisy location records according to the noisy count.

$|C|$ represents the records number of cluster C. $noisy\_count$ is the noisy count of cluster C, which is obtained from line 8 of algorithm 1. $\lfloor |C| - noisy\_count \rfloor$ represents the nearest integer to $|C| - noisy\_count$. $0 \leq \alpha \leq 1$. While $\alpha = 0$, it means using the cluster center to replace all the location record in the cluster, and while $\alpha = 1$, it means random select one location record to replace the original location record.

4.2.4. Privacy Analysis

In this section, we prove that our scheme satisfies differential privacy. Two differential privacy transformation is performed in Algorithm 1. In line 5, Exponential mechanism transformation is conducted and one partition result is selected. In line 8, Laplace noise is added to achieve differential privacy transformation. First of all, we prove that select one partition result from R satisfies differential privacy.

Algorithm 4 generate sanitized cluster

Input: A cluster C and a parameter α

Output: Noisy cluster C'

Procedure:

1 find out the cluster center c and radius r

2 $|C|$ points are generated randomly, and the distance between these points and point c does not exceed $\alpha \cdot r$

3 add all the generated location records to C'

4 if $noisy\_count > |C|$

5 $\lfloor noisy\_count - |C|$ points are generated randomly, and the distance between those points and the cluster center c does not exceed $\alpha \cdot r$

6 if $noisy\_count < |C|$

7 Randomly delete $\lfloor |C| - noisy\_count \rfloor$ records from C'

8 return C'

Lemma 1. *The process of selecting one partition result from R satisfies differential privacy.*

Proof. Proof. Suppose that q is the query function, and R' is the neighbor dataset of R. $Pr(*)$ is the output probability.

$$\begin{aligned} \frac{Pr(q(R) = p_i)}{Pr(q(R') = p_i)} &= \frac{\frac{\exp(\frac{\varepsilon_1}{2\Delta u} u(R,p_i))}{\sum_{j=1,\cdots,m} \exp(\frac{\varepsilon_1}{2\Delta u} u(R,p_j))}}{\frac{\exp(\frac{\varepsilon_1}{2\Delta u} u(R',p_i))}{\sum_{j=1,\cdots,m} \exp(\frac{\varepsilon_1}{2\Delta u} u(R',p_j))}} \\ &= \frac{\exp(\frac{\varepsilon_1}{2\Delta u} u(R,p_i))}{\exp(\frac{\varepsilon_1}{2\Delta u} u(R',p_i))} * \frac{\sum_{j=1,\cdots,m} \exp(\frac{\varepsilon_1}{2\Delta u} u(R',p_j))}{\sum_{j=1,\cdots,m} \exp(\frac{\varepsilon_1}{2\Delta u} u(R,p_j))} \end{aligned} \tag{8}$$

the first multiplication factor of formula (8) is :

$$\begin{aligned} \frac{exp(\frac{\varepsilon_1}{2\delta u} u(R,p_i))}{exp(\frac{\varepsilon_1}{2\delta u} u(R',p_i))} &= \exp(\frac{\varepsilon_1}{2\Delta u}(u(R,p_i) - u(R',p_i))) \\ &<= \exp(\frac{\varepsilon_1}{2\Delta u} * \Delta u) = \exp(\frac{\varepsilon_1}{2}) \end{aligned} \tag{9}$$

the second multiplication factor of formula (8) is :

$$\begin{aligned} \frac{\sum_{j=1,\cdots,m} \exp(\frac{\varepsilon_1}{2\Delta u} u(R',p_j))}{\sum_{j=1,\cdots,m} \exp(\frac{\varepsilon_1}{2\Delta u} u(R,p_j))} &= \frac{\sum_{j=1,\cdots,m} (\exp(\frac{\varepsilon_1}{2\Delta u} u(R,p_j)) * \exp(\frac{\varepsilon_1}{2\Delta u} u(R',p_j) - (\frac{\varepsilon_1}{2\Delta u} u(R,p_j))))}{\sum_{j=1,\cdots,m} \exp(\frac{\varepsilon_1}{2\Delta u} u(R,p_j))} \\ &<= \frac{\sum_{j=1,\cdots,m} (\exp(\frac{\varepsilon_1}{2\Delta u} u(R,p_j)) * \exp(\frac{\varepsilon_1}{2}))}{\sum_{j=1,\cdots,m} \exp(\frac{\varepsilon_1}{2\Delta u} u(R,p_j))} = \exp(\frac{\varepsilon_1}{2}) \end{aligned} \tag{10}$$

from Formulas (8)–(10), we can get :

$$\frac{Pr(q(R) = p_i)}{Pr(q(R') = p_i)} <= \exp(\varepsilon_1) \tag{11}$$

which means that the process of selecting one partition from R satisfies differential privacy. □

Lemma 2. *The transformation of line 8 of Algorithm 1 satisfies differential privacy.*

Proof. We prove that adding Laplace noise to the number of location record in c_j satisfies differential privacy. $|c_j|$ is the number of location record in c_j. q is the query funciton. Let

X be the noise injected to $q(c_j)$. X follows the Laplace distribution $Lap(\frac{\Delta q}{\varepsilon_2})$. $Pr(*)$ is the output probability.

$$Pr[q(c_j) = t] = Pr[|c_j| + X = t] = Pr[X = t - |c_j|] = \frac{\varepsilon_2}{2\Delta q} * \exp(\frac{-\varepsilon_2|t - q(c_j)|}{\Delta q}) \quad (12)$$

Suppose c'_j is the neighbor of c_j. Similarly, we have

$$Pr[q(c'_j) = t] = \frac{\varepsilon_2}{2\Delta q} * \exp(\frac{-\varepsilon_2|t - q(c'_j)|}{\Delta q}) \quad (13)$$

Thus,

$$\begin{aligned} \frac{Pr[q(c'_j) = t]}{Pr[q(c_j) = t]} = \frac{\exp(\frac{-\varepsilon_2|t-q(c_j)|}{\Delta q})}{\exp(\frac{-\varepsilon_2|t-q(c'_j)|}{\Delta q})} &= \exp(\frac{\varepsilon_2(|t - q(c'_j)| - |t - q(c_j)|)}{\Delta q}) \\ &<= \exp(\frac{\varepsilon_2|q(c_j) - q(c'_j)|}{\Delta q}) <= \exp(\varepsilon_2) \end{aligned} \quad (14)$$

□

Theorem 5. *The Algorithm 1 satisfies differential privacy.*

Proof. According to Lemma 1, the transformation of line 5 in Algorithm 1 satisfies differential privacy and the differential privacy budget is ε_1. According to Lemma 2, the transformation of line 8 in Algorithm 1 satisfies differential privacy. The differential privacy budget is ε_2. According to Theorem 3, the transformation for each time slice s_i satisfies differential privacy, and the total privacy budget is $\varepsilon_1 + \varepsilon_2$. This transformation is done on different time slice, respectively. According to Theorem 4, Algorithm 1 satisfies differential privacy, and the privacy budget is $max_{s_i}\{\varepsilon_1 + \varepsilon_2\}$. □

5. Experiments and Analysis

Experiments are conducted on two different real-world trajectory databases. The GPS trajectory dataset is collected in Geolife (Microsoft Research Asia) project by 182 users in a period of over five years, and this dataset is used in [44–46]. It contains 17,621 trajectories with total distance of 1,292,951 km. Here, transportation mode is ignored. Four of seven attributes, which are latitude, longitude, date and time are used in our experiments. Two new attributes trajectory ID (IDentity) and location record ID (IDentity) are added. There are too many records in the original dataset. We only use the trajectory data from 6:00 a.m. to 7:00 a.m. on 1–2 January and 2009. There are 38 trajectories with 106,535 location records in the dataset we use.

The second dataset is T-drive Taxi Trajectories dataset collected by Microsoft Research, which has been used in [1,15,47,48]. The dataset contains the GPS trajectories of 10,357 taxis during the period of 2–8 February 2008 within Beijing. Each taxi's GPS trajectory record is saved in a single file, respectively, and every record has four attributes named 'taxi id', 'date time', 'longitude', 'latitude'. The trajectory data of 1000 taxis are selected randomly from 6:00 a.m. to 7:00 a.m. on 4 February. The proposed method is compared with the existing methods on the above two datasets in terms of execution efficiency and data utility. In the following, our experimental results are mainly compared with those in [1] (INFOCOM15) and [15] (INFOSCI17).

5.1. Information Loss

Any modification of the original dataset will cause some damage to it, which will reduces the utility of the original dataset. In this paper, the distance between the sanitized

dataset and the original dataset is used to measure the loss of information. The farther the distance is, the more information is lost and the vice versa.

Similar to [1,15], Hausdorff distance is used to measure the infmormation loss, and the definition is as follows:

$$infor\_loss(D', D) = max\{h(D, D'), h(D', D)\} \quad (15)$$

where $h(D, D') = max_{T \in D}\{min_{T \in D'}\{distance(T, T')\}\}$. The smaller the value of $infor\_loss(D', D)$ is, the less information is lost.

5.2. Compared with Other Method

Comparative experiments are conducted on the two real world datasets. The experimental results are shown in the following figures. From Figures 2 and 3, we can see that the efficiency of INFOSCI17 proposed in [15] is significantly better than INFOCOM15 proposed in [1] on the both datasets. The results of our method proposed in this paper are marked as 'TPRSDP_0.3', 'TPRSDP_0.6' and 'TPRSDP_0.9', which represent our method with different sampling rates 0.3, 0.6 and 0.9, respectively. On the Geolife dataset, the time cost of our scheme with different sampling rate is 50% less than that of INFOCOM15. On the T-drive dataset, the time cost of our scheme is obviously better than existing work. As can be seen in Figure 3, while the sampling rate is 0.9, the time cost is be slightly higher than INFOSCI17. This because, compared with the exiting work, our scheme has an additional random sampling process, and the sampling rate is relatively high. The experimental results show that when the sampling rate is 0.6, the time cost of this scheme is obviously better than that of INFOCOM15 and INFOSCI17.

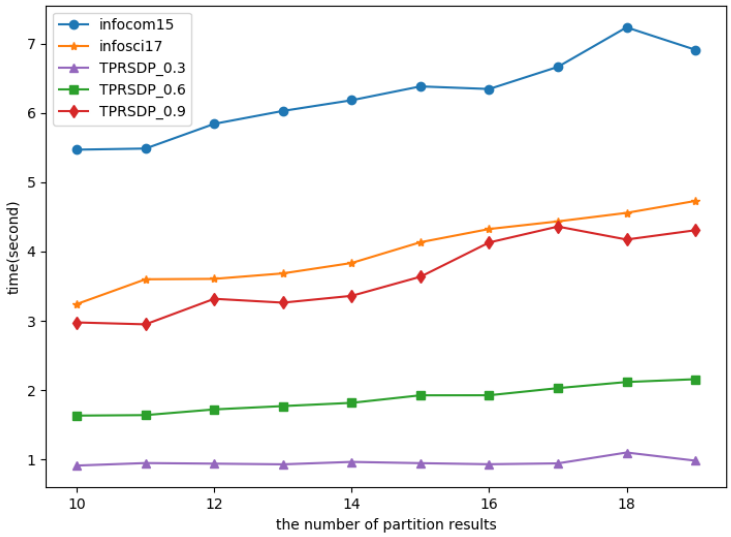

Figure 2. Time cost (Geolife dataset).

The loss of information of different methods on the Geolife dataset and the T-drive dataset is shown in Figures 4 and 5. The amount of information lost generated by our scheme is significantly less than that caused by INFOCOM15 on the Geolife dataset. While the sampling rate is 0.6 or 0.9, the loss of information by our scheme is nearly the same with INFOSCI17. However, while the sampling rate is 0.3, the loss of information by our scheme is more than that of INFOSCI17. The reason for this is that the sampling rate is too low, and it is not reasonable to divide the original dataset according to the cluster center of the random sampling sub-dataset. On the T-drive dataset, the performance of our scheme is nearly similar to INFOCOM15 and INFOSCI17. The performances of our scheme are different on the two datasets in terms of information loss. It is mainly because the two datasets have different characteristics. There are only 182 users' trajectory records

in Geolife dataset, however, there are more than one thousand taxis'trajectory records in T-drive dataset. While the sampling rate is only 0.3, there will be many records extracted from T-drive dataset. These records can well represent the characteristics of the original dataset.

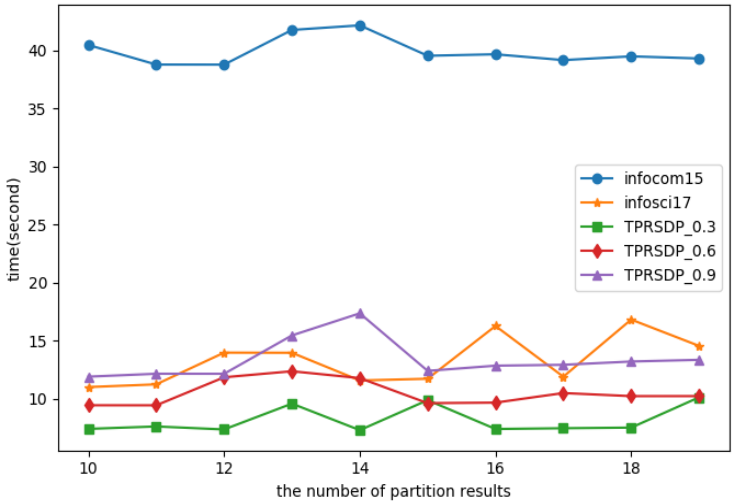

Figure 3. Time cost (T-drive dataset).

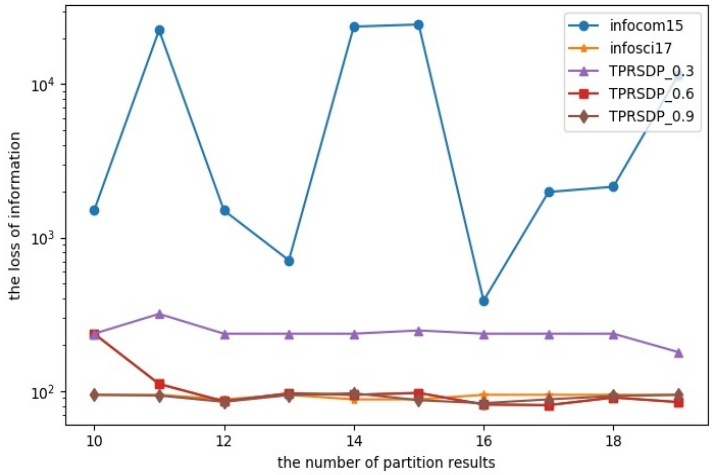

Figure 4. The loss of information(Geolife dataset).

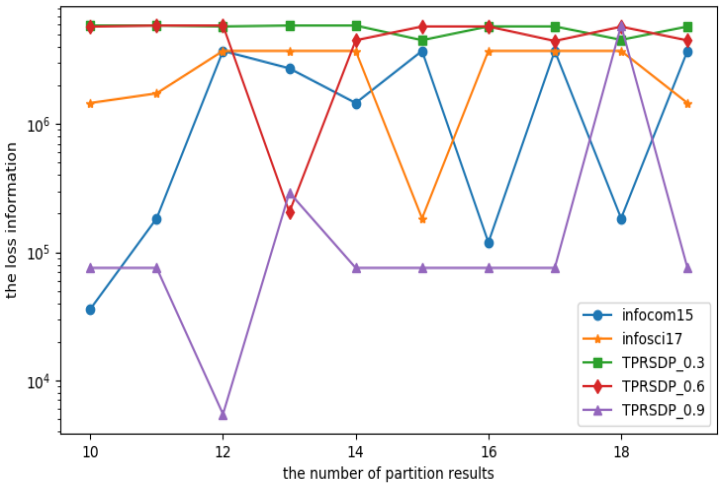

Figure 5. The loss of information(T-drive dataset).

As discussed above, the efficiency of the proposed method is obviously better than that of other methods, and the loss of information of the sanitized dataset is affected by the dataset itself and system parameter settings. In the following, we will discuss the setting of system parameters in detail.

6. Parameter Setting

The value of parameters have a great influence on the system performance. Experiments are conducted on the two real-world datasets to verify the influence of the parameters with different values on the system performance. The number of partition result is φ in Algorithm 1. In the following experiments, we set φ to 10, 15 or 20, respectively, to verify the performance of our scheme.

6.1. The Number of Cluster

Experimental results are shown in Figures 6–9 with different cluster number. In Figures 6 and 7, the time costs of our scheme on Geolife dataset and T-drive dataset are shown, respectively. As can be seen from Figures 6 and 7, when the number of partition results is larger, the experiment will take more time. The reason for this is obvious. It will take more time to generate more partition results. On the two datasets, the time cost decreases rapidly with the increase of the number of clusters in each time slice. However, when the number of clusters exceeds 10, the change becomes insignificant. This is because there are fewer records in each cluster, and it will take less time to generate fake location records.

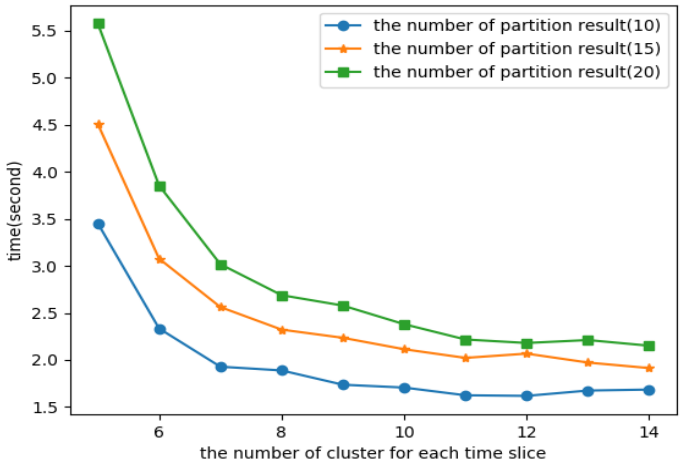

Figure 6. Time cost (Geolife dataset).

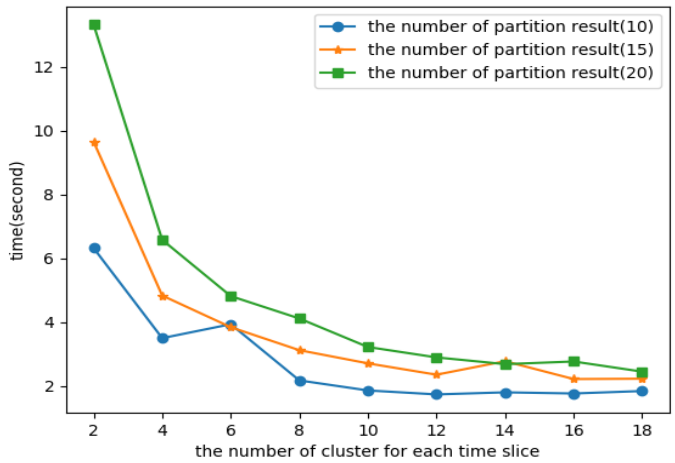

Figure 7. Time cost (Tdrive dataset).

The losses of information of our scheme on the two datasets are shown in Figures 8 and 9. while the number of cluster is less than 7, the loss of information is decrease rapidly on Geolife dataset. However, while the number of cluster is more than 7, the change of information loss is relative small. On the T-drive dataset, while the number of partition results is less than 6, the loss of information varies greatly, and the trend of change is inconsistent. While the number of partition result is more than 6, the losses of information are nearly the same corresponding to different partition number. From the above discussion, we know that in the same sub dataset, when the number of clusters is greater than 6, the time cost is lower and the amount of information loss is relatively smaller. The reason for this is that our method is based on K-means. One of the popular ways to choose the number of cluster in K-means is elbow method [40,49]. The main idea is as follows: When the value of K is much smaller than the actual number of categories, increasing the value of K will significantly increase the performance of K-means algorithm. On the contrary, when the value of K is greater than or equal to the actual number of categories, increasing the value of K will not significantly improve the algorithm. As can be seen from Figures 6 and 7, when the number of cluster on the Geolif dataset is 7, it is relatively reasonable. The reason may be that most of the location records are concentrated in seven areas. From Figures 8 and 9, we know that the reasonable cluster number may be 6 on T-drive dataset. From the above discussion, we know that the number of clusters should be set according to the distribution of different datasets. In other words, the number of clusters depends on the dataset itself. However, it will be better to be larger than the number of relatively dense areas in the dataset.

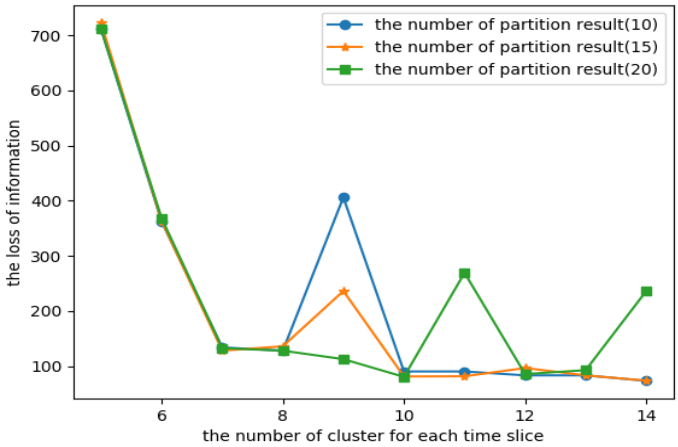

Figure 8. The loss of information (Geolif dataset).

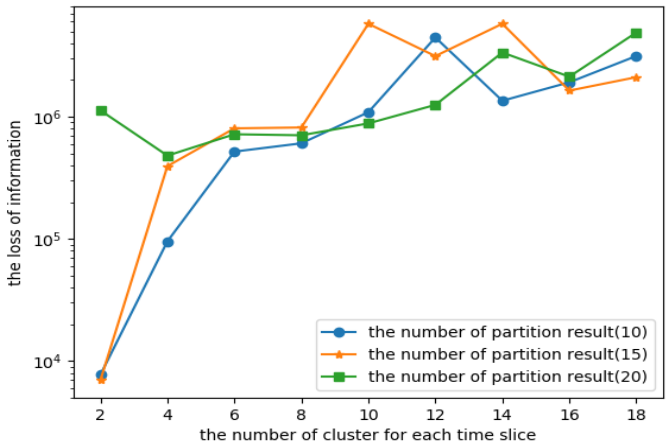

Figure 9. The loss of information (T-drive dataset).

6.2. The length of Time Slice

The travel time is divided into time slices, and the dataset is divided into sub dataset according to the time slice. The length of the time slice has a great influence on the performance of the system. On the two datasets, the time costs of the experiments with different length of time slice are shown in Figures 10 and 11. While the length of time slice increases, the cost of time will increase gradually. The reason for this is obvious. If the length of time slice is longer, there will be more location records in each cluster, and the time cost of running K-means will increase. When the length of time slice is constant, the greater the number of partition results is, the more time cost is. This means that if the number of partition results is large, the length of the time slice should not be too long.

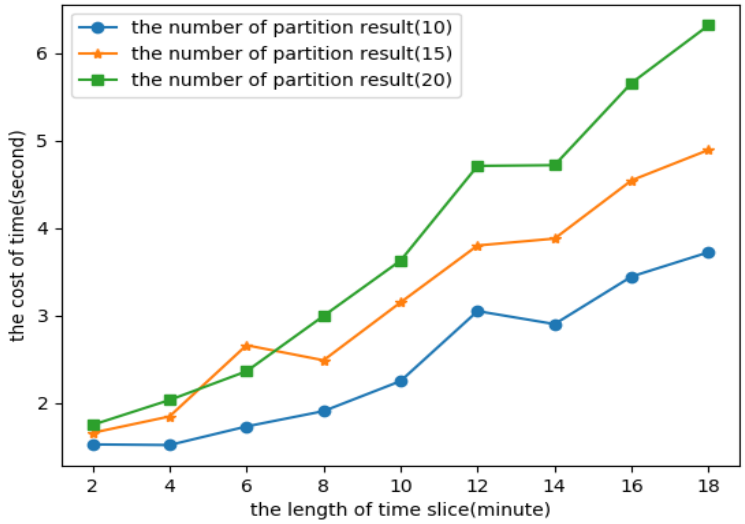

Figure 10. The cost of time (Geolife dataset).

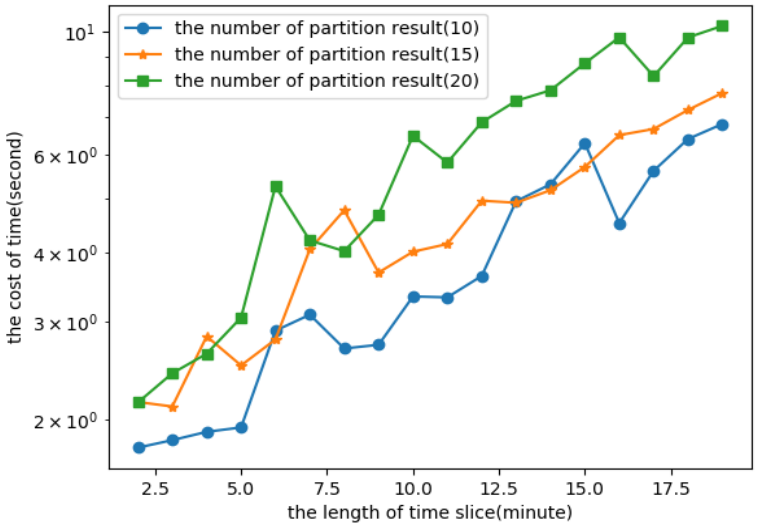

Figure 11. The cost of time (T-drive dataset).

The losses of information on Geolife dataset and T-drive dataset are shown in Figures 12 and 13, respectively. On the Geolife dataset, while the length of time slice is less than 8 min, the amount of information loss is relatively small. However, while the length of time slice is more than 8 min, the amount of information lost increases significantly. On the T-drive dataset, the amount of information lost does not change too much with different length of time slice, and the reason for this is that there much more trajectories in the T-drive dataset. From Figures 12 and 13, we can see that when the number

of partitions are 10, 15 and 20, respectively, the amount of information loss is almost the same, which means that the number of partitions has only a little influence on the amount of information loss. When the length of time slice increases, the fluctuation of information loss will increase. Generally speaking, when the length of time slice is shorter than 8 min, the performance of our scheme is better.

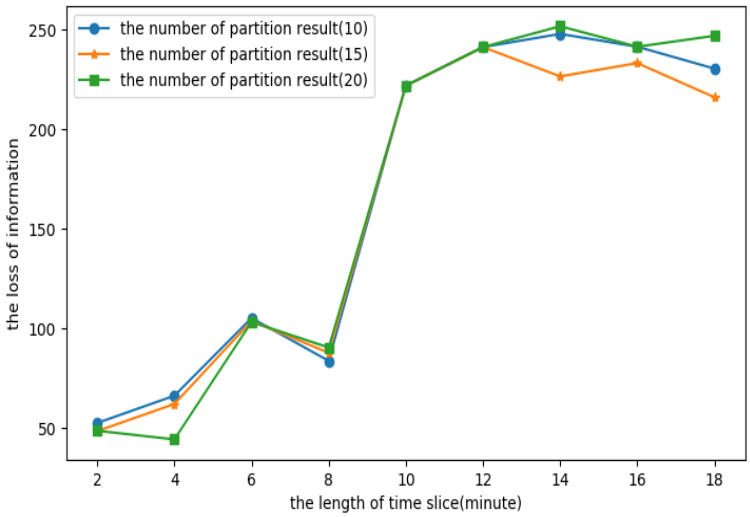

Figure 12. The loss of information (Geolife dataset).

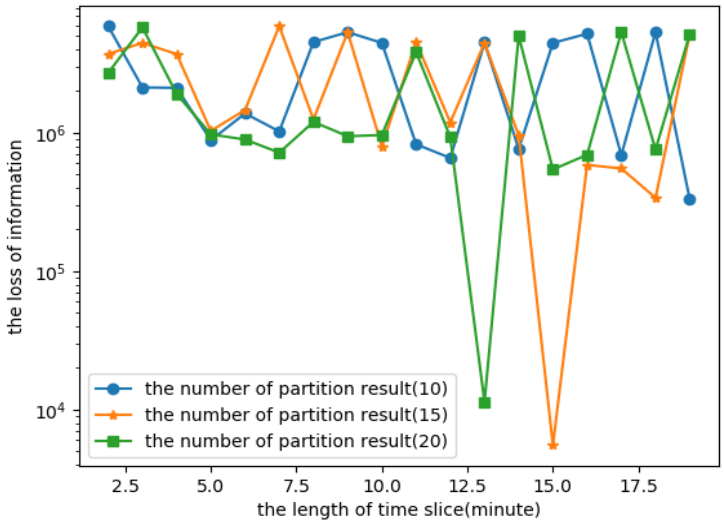

Figure 13. The loss of information (T-drive dataset).

6.3. *The Budget of Differential Privacy*

The impact of privacy budget is validated in the following experiments. The time cost with different budget on the two dataset is shown in Figures 14 and 15, respectively. It is can be seen that there is no significant differences between the time costs with different budgets, and the reason for this is that the effect of adding noise is offset by the process of random sampling. From the above comparison, we can see that the time cost of our scheme is mainly depends on the number of partition results, the number of clusters, the length of time slices and the dataset itself.

The loss of information with different budget on the two datasets are shown on Figures 16 and 17. On T-drive dataset, the amount of information loss is significantly larger than that on the Geolife dataset. This is mainly because there are much more location records in T-drive dataset than in Geolife dataset. There are only 182 users' GPS trajectories

in Geolife dataset, however, more than ten thousand taxis' GPS trajectories are collected in T-drive dataset. The amount of information loss with different privacy budget is nearly the same, and with the increase of differential privacy budget, the amount of information loss does not decrease significantly. The amount of information loss is not stable. There are two main reasons. The first is that the suboptimal partition results may be generated by different K-means clustering results. The second is that the results of random sampling may be quite different.

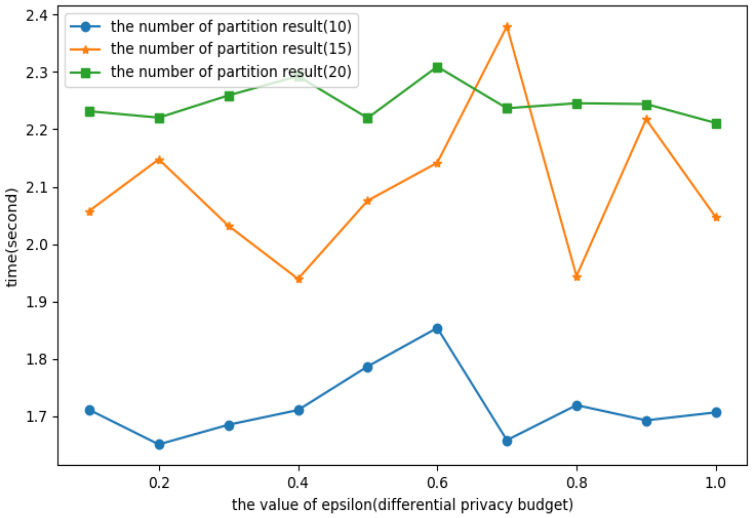

Figure 14. Time cost with different epsilon (Geolife dataset).

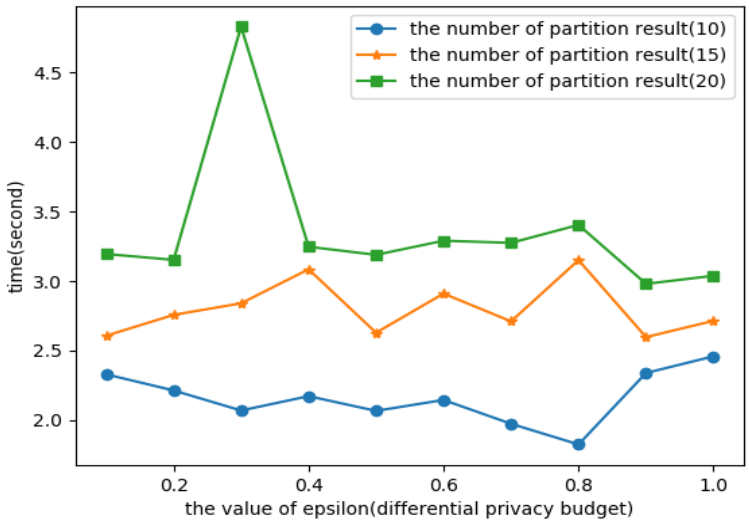

Figure 15. The time cost with different epsilon (T-drive dataset).

As discussed above, while the time slice is shorter and the number of cluster of each time slice is bigger, the performance of our scheme will be better. However, the scheme proposed in this paper must be provide more privacy protection than K-anonymous, which means that there must be more than k location records in each time slice. T is the total length of time, l is the length of time slice, c is the number of cluster, and there are n location records. Suppose that the location records follow uniform distribution, then $\frac{n}{T/t} \cdot \frac{1}{c} \geq k$. In order to provide enough protection for the trajectory datastet, the parameter of the scheme must satisfy $\frac{t}{c} \geq \frac{k \cdot T}{n}$.

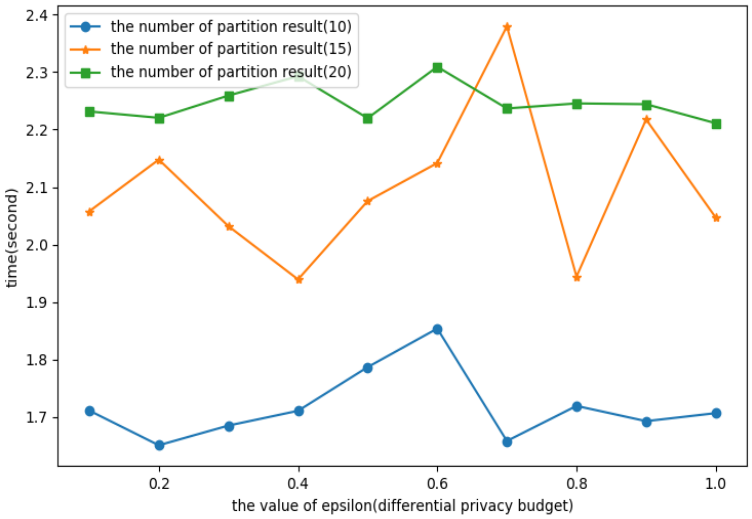

Figure 16. Time cost with different epsilon (Geolife dataset).

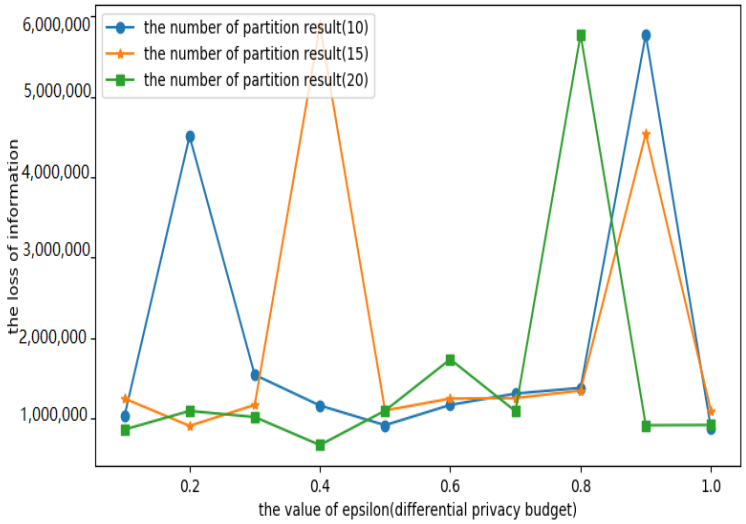

Figure 17. The information loss with different epsilon (T-drive dataset).

7. Conclusions

The scheme proposed in this paper is superior to the existing scheme; in particular, the efficiency is much higher than others. The information loss of this method is no more than that of others. A random sampling process is added, which can greatly reduce the number of data processed by the K-means algorithm. Both the Laplace mechanism and the Exponential mechanism are used in our scheme, and we proved that our scheme satisfies differential privacy.

The influence of parameter setting on the performance of the system is verified by experiments. The experimental results show that if the setting of the parameters is reasonable, the loss of information of our scheme is less than that of INFOCOM15. While the number of clusters is not too large, and the length of time slice is not too long, our scheme has a good performance. Unfortunately, while the parameter values of our scheme are the same, the performance of our scheme on different datasets may be different, which means that the parameter setting for specific dataset needs further study.

Author Contributions: Conceptualization, Tinghuai Ma and Fagen Song; methodology, Tinghuai Ma; software, Fagen Song; validation, Fagen Song; writing—original draft preparation, Fagen Song; writing—review and editing, Tinghuai Ma. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by National Key Research and Development Program of China (International Technology Cooperation Project No.2021YFE014400).This work was supported in part by National Science Foundation of China (No. U1736105, No. 61572259). This work was supported in part by the Natural Science Foundation of the Colleges and Universities in Anhui Province of China under Grant(No.KJ2020A0035).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hua, J.; Gao, Y.; Zhong, S. Differentially private publication of general time-serial trajectory data. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 549–557.
2. Wang, R.; Zhou, F. Physical layer security for land mobile satellite communication networks with user cooperation. *IEEE Access* **2019**, *7*, 29495–29505. [CrossRef]
3. Ma, T.; Shao, W.; Hao, Y.; Cao, J. Graph classification based on graph set reconstruction and graph kernel feature reduction. *Neurocomputing* **2018**, *296*, 33–45. [CrossRef]
4. Ma, T; Zhou, H;;Tian, Y; Al-Nabhan, N. A novel rumor detection algorithm based on entity recognition, sentence reconfiguration, and ordinary differential equation network. *Neurocomputing* **2021**, *447*, 224–234. [CrossRef]
5. Ma, T.; Rong, H.; Hao, Y.; Cao, J.; Al-Rodhaan, M.A. A Novel Sentiment Polarity Detection Framework for Chinese. *IEEE Trans. Affect. Comput.* **2019**. [CrossRef]
6. Xin, Y.; Xie, Z.Q.; Yang, J. The privacy preserving method for dynamic trajectory releasing based on adaptive clustering. *Inf. Sci.* **2017**, *378*, 131–143. [CrossRef]
7. Ma, T.; Wang, H.; Zhang, L.; Tian, Y.; Al-Nabhan, N. Graph classification based on structural features of significant nodes and spatial convolutional neural networks. *Neurocomputing* **2021**, *423*, 639–650. [CrossRef]
8. Zhao, X.; Pi, D.; Chen, J. Novel trajectory privacy-preserving method based on prefix tree using differential privacy. *Knowl. Based Syst.* **2020**, *198*, 105940. [CrossRef]
9. Samarati, P.; Sweeney, L. *Protecting Privacy When Disclosing Information: K-Anonymity and Its Enforcement through Generalization and Suppression*; Technical Report, SRI-CSL-98-04; SRI Computer Science Laboratory. 1998. Available online: :https://www.epic.org/ (accessed on 27 June 2021).
10. Dwork, C. Differential Privacy. In Proceedings of the 33rd International Conference on Automata, Languages and Programming-Volume Part II, Berlin, Germany, 29 June 2006.
11. Chaudhuri, K.; Monteleoni, C.; Sarwate, A.D. Differentially private empirical risk minimization. *J. Mach. Learn. Res.* **2011**, *12*, 1069–1109.
12. Wang, S.; Sinnott, R.O. Protecting personal trajectories of social media users through differential privacy. *Comput. Secur.* **2017**, *67*, 142–163. [CrossRef]
13. Yuan, C.; Xia, Z.; Xingming, S. Coverless Image Steganography Based on SIFT and BOF. *J. Internet Technol.* **2017**, *18*, 435–442.
14. Jiang, K.; Shao, D.; Bressan, S.; Kister, T.; Tan, K.L. Publishing trajectories with differential privacy guarantees. In Proceedings of the 25th International Conference on Scientific and Statistical Database Management, Baltimore, Maryland, USA, 29 July 2013; pp. 1–12.
15. Li, M.; Zhu, L.; Zhang, Z.; Xu, R. Achieving differential privacy of trajectory data publishing in participatory sensing. *Inf. Sci.* **2017**, *400*, 1–13. [CrossRef]
16. He, X.; Cormode, G.; Machanavajjhala, A.; Procopiuc, C.M.; Srivastava, D. DPT: Differentially private trajectory synthesis using hierarchical reference systems. *Proc. VLDB Endow.* **2015**, *8*, 1154–1165. [CrossRef]
17. Song, F.; Ma, T.; Tian, Y.; Al-Rodhaan, M. A New Method of Privacy Protection: Random k-Anonymous. *IEEE Access* **2019**, *7*, 75434–75445. [CrossRef]
18. Komishani, E.G.; Abadi, M.; Deldar, F. PPTD: Preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression. *Knowl. Based Syst.* **2016**, *94*, 43–59. [CrossRef]
19. Yilmaz, E.; Ferhatosmanoglu, H.; Ayday, E.; Aksoy, R.C. Privacy-preserving aggregate queries for optimal location selection. *IEEE Trans. Dependable Secur. Comput.* **2017**, *16*, 329–343. [CrossRef]
20. Xie, K.; Ning, X.; Wang, X.; He, S.; Ning, Z.; Liu, X.; Wen, J.; Qin, Z. An efficient privacy-preserving compressive data gathering scheme in WSNs. *Inf. Sci.* **2017**, *390*, 82–94. [CrossRef]
21. He, S.; Zeng, W.; Xie, K.; Yang, H.; Su, X. PPNC: Privacy Preserving Scheme for Random Linear Network Coding in Smart Grid. *Ksii Trans. Internet Inf. Syst.* **2017**, *11*, 1510–1532.
22. Zhang, S.; Lin, Y.; Liu, Q.; Jiang, J.; Yin, B.; Choo, K.K.R. Secure hitch in location based social networks. *Comput. Commun.* **2017**, *100*, 65–77. [CrossRef]
23. Zeng, W.; Chen, P.; Chen, H.; He, S. PAPG: Private Aggregation Scheme based on Privacy-preserving Gene in Wireless Sensor Networks. *KSII Trans. Internet Inf. Syst.* **2016**, *10*, 4442–4466.

24. Xiong, P.; Zhu, T.; Niu, W.; Li, G. A differentially private algorithm for location data release. *Knowl. Inf. Syst.* **2016**, *47*, 647–669. [CrossRef]
25. Wang, J.; Zhu, R.; Liu, S.; Cai, Z. Node location privacy protection based on differentially private grids in industrial wireless sensor networks. *Sensors* **2018**, *18*, 410. [CrossRef] [PubMed]
26. Wang, S.; Sinnott, R.; Nepal, S. Privacy-protected statistics publication over social media user trajectory streams. *Future Gener. Comput. Syst.* **2018**, *87*, 792–802. [CrossRef]
27. Ma, Z.; Zhang, T.; Liu, X.; Li, X.; Ren, K. Real-time privacy-preserving data release over vehicle trajectory. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8091–8102. [CrossRef]
28. Ma, T.; Jia, J.; Xue, Y.; Tian, Y.; Al-Dhelaan, A.; Al-Rodhaan, M. Protection of location privacy for moving kNN queries in social networks. *Appl. Soft Comput.* **2018**, *66*, 525–532. [CrossRef]
29. Han, Q.; Xiong, Z.; Zhang, K. Research on Trajectory Data Releasing Method via Differential Privacy Based on Spatial Partition. *Secur. Commun. Networks* **2018**, *2018*, 1–14. [CrossRef]
30. Aggarwal, C.C. On k-anonymity and the curse of dimensionality. In Proceedings of the 31st International Conference on Very Large Data Bases, Toronto, ON, Canada, 31 August 2005; pp. 901–909.
31. Abul, O.; Bonchi, F.; Nanni, M. Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases. In Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, Cancun, Mexico, 7–12 April 2008; Volume 8, pp. 376–385.
32. Domingoferrer, J.; Trujillorasua, R. Microaggregation- and permutation-based anonymization of movement data. *Inf. Sci.* **2012**, *208*, 55–80. [CrossRef]
33. Dritsas, E.; Kanavos, A.; Trigka, M.; Vonitsanos, G.; Sioutas, S.; Tsakalidis, A. Trajectory Clustering and k-NN for Robust Privacy Preserving k-NN Query Processing in GeoSpark. *Algorithms* **2020**, *13*, 182. [CrossRef]
34. Dritsas, E.; Trigka, M.; Gerolymatos, P.; Sioutas, S. Trajectory Clustering and k-NN for Robust Privacy Preserving Spatiotemporal Databases. *Algorithms* **2018**, *11*, 207. [CrossRef]
35. Kifer, D. Attacks on privacy and deFinetti's theorem. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, Providence, RI, USA, 29 June 2009*; ACM: New York, NY, USA, 2019; pp. 127–138.
36. Wong, R.C.W.; Fu, A.W.C.; Wang, K.; Pei, J. Minimality attack in privacy preserving data publishing. In Proceedings of the 33rd International Conference on Very Large Data Bases, Viena, Austria, 23 September 2007; pp. 543–554.
37. Ganta, S.R.; Kasiviswanathan, S.P.; Smith, A. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, USA, 24 August 2008*; ACM: New York, NY, USA, 2008; pp. 265–273.
38. Chen, R.; Fung, B.; Desai, B.C. Differentially private trajectory data publication. *arXiv* **2011**, arXiv:1112.2020.
39. Yin, C.; Xi, J.; Sun, R.; Wang, J. *Location Privacy Protection Based on Differential Privacy Strategy for Big Data in Industrial Internet of Things*; IEEE Transactions on Industrial Informatics: New York, NY, USA, 2018; Volume 14, pp. 3628–3636. Available online: https://ieeexplore.ieee.org/document/8110700 (accessed on 27 June 2021).
40. Chen, R.; Acs, G.; Castelluccia, C. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security, New York, NY, USA, 7 November 2012*; ACM: New York, NY, USA, 2012; pp. 638–649.
41. Li, N.; Qardaji, W.; Su, D. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, New York, NY, USA, 2 May 2012; pp. 32–33.
42. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. *J. Priv. Confidentiality* **2016**, *7*, 17–51. [CrossRef]
43. McSherry, F.; Talwar, K. Mechanism Design via Differential Privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), Providence, RI, USA, 21–23 October 2007; Volume 7, pp. 94–103.
44. Zheng, Y.; Zhang, L.; Xie, X.; Ma, W.Y. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web*; ACM: New York, NY, USA, 2009; pp. 791–800. Available online: https://www.microsoft.com/en-us/research/publication/mining-interesting-locations-and-travel-sequences-from-gps-trajectories/ (accessed on 27 June 2021).
45. Zheng, Y.; Li, Q.; Chen, Y.; Xie, X.; Ma, W.Y. Understanding mobility based on GPS data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*; ACM: New York, NY, USA, 2008; pp. 312–321. Available online: https://www.microsoft.com/en-us/research/publication/mining-interesting-locations-and-travel-sequences-from-gps-trajectories/ (accessed on 27 June 2021).
46. Zheng, Y.; Xie, X.; Ma, W.Y. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* **2010**; *33*, 32–39.
47. Yuan, J.; Zheng, Y.; Xie, X.; Sun, G. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2011; pp. 316–324. Available online: https://www.microsoft.com/en-us/research/publication/driving-with-knowledge-from-the-physical-world/ (accessed on 27 June 2021).

48. Yuan, J.; Zheng, Y.; Zhang, C.; Xie, W.; Xie, X.; Sun, G.; Huang, Y. T-drive: Driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*; ACM: New York, NY, USA, 2010; pp. 99–108. Available online: https://www.microsoft.com/en-us/research/publication/t-drive-driving-directions-based-on-taxi-trajectories/ (accessed on 27 June 2021).
49. Fukuoka, Y.; Zhou, M.; Vittinghoff, E.; Haskell, W.; Goldberg, K.; Aswani, A. Objectively measured baseline physical activity patterns in women in the mPED trial: Cluster analysis. *JMIR Public Health Surveill.* **2018**, *4*, e10. [CrossRef]