Supporting information to

LETTER

Similarity-based Privacy Protection for Publishing *k*-Anonymous Trajectories

Keywords Privacy protection, Trajectory, Publishing, k-anonymous, Similarity.

1 Organization of this Supporting information

The first part introduces the organization of the Supporting information. The second part introduces the existing trajectory privacy protection methods, analyzes different strategies based on trajectory clustering and reconstruction methods, and introduces different trajectory similarity measurement methods. The third part describes the system model and the symbols, describes the problem statement. In the fourth part, a new trajectory similarity measurement is proposed. In the fifth part, a new method for privacy protection of publishing trajectories is proposed, including clustering of similar trajectories and reconstruction of trajectories within clusters. The sixth part verifies the efficiency of the privacy protection algorithm and the usefulness of the published trajectory through experiments. Finally, we summarize this paper and put forward two open questions.

2 Related work

With the development of location technology and mobile communication technology, a large number of LBS emerged. LSP continuously provides information services and accumulates a large number of spatial and temporal location data. Through these data, users can be traced directly or some sensitive personal information can be inferred, which will seriously lead to the disclosure of personal privacy [1].

2.1 Application scenarios of trajectory privacy protection

2.1.1 Trajectory privacy protection for LBS

If untrusted LSP can identify the specific owner of the spatial-temporal location information, they can construct a spatial-temporal location sequence, which may reveal the user's privacy. Before sending service request information to LSP, it is necessary to ensure that the spatial-temporal location information is anonymous, which is the trajectory privacy protection for LBS.

2.1.2 Privacy protection for publishing trajectories

Users can send a location service request to trusted LSP in plain code. Before publishing or delivering the data to untrusted third party for analysis, the data owner should process the trajectory data to ensure that the third party cannot obtain the user's sensitive information through background knowledge. This is called privacy protection for publishing trajectories.

2.2 Privacy protection methods for publishing trajectories

The privacy protection methods for publishing trajectories are divided into three categories, corresponding to three basic operations (add, delete and update) in privacy protection methods for relational database.

2.2.1 Add the dummies

The Dummy method generates dummy data according to the statistical characteristics of the original trajectory data, and adds it to the original data to improve the anonymity.

This method ensures that the specific statistical characteristics of the published data are not damaged seriously [2]. The dummy method increases the amount of data greatly [3]. Due to the spatial-temporal correlation and multi-dimensionality of trajectory data, the success rate of existing dummy-based trajectory privacy protection schemes is less than 15% [4].

2.2.2 Suppress sensitive data

The fundamental purpose of privacy protection is to hide connections between sensitive attributes and individuals. The suppression-based approach aims to remove sensitive locations in the published trajectory data to achieve privacy protection [5, 6]. Sensitive locations can be determined according to specific problems, such as the location frequently visited by users, the location with important semantic characteristics etc.

Algorithms based on suppressing sensitive data are all designed under the condition of limiting the prior knowledge of the attacker, which belongs to QID-known anonymous technology [7].The difficulty lies in obtaining the sensitive position and the opponent knowledge of the data in advance [8].The suppression of sensitive information causes a great deal of information loss and limits the availability of published data.

2.2.3 Generalize original data

The generalization method generates the published trajectory data, retains the important features of the original trajectory data, protects the sensitive information or reduces the probability of the attacker to identify the trajectory body. The published trajectory is not a new trajectory, but represents a range of possible experiences of the original trajectory, so it can be regarded as a generalization of the original trajectory. This method avoid large increase in data and information loss. The generalization methods of trajectories are divided into two categories. One is the method to cluster and generalize trajectories, which tries to realize the privacy protection of the identity of trajectory subject through the anonymization of trajectories. The other is to achieve privacy protection of sensitive attributes through generalization of trajectory positions.

2.3 Cluster-based method for generalization of trajectories

For the published trajectory data, anonymity is the main requirement of trajectory subject identity privacy protection. *k*-Anonymity is one of the main privacy protection policies, which can be interpreted as that the probability of the adversary determining the target trajectory in the trajectory data is no greater than 1/k. This method achieves *k*-anonymity of trajectories by clustering trajectories and reconstructing trajectories within the trajectories.

2.3.1 Measurement of trajectory similarity

Cluster is the collection of similar trajectories, so the measurement of trajectory similarity is essential. The metric of trajectory similarity depends on its application scenario, which can be any combination of trajectory features, such as trajectory direction, velocity, shape, position, route, lifespan etc. [9]. Trajectories have the dual characteristics of position and sequence. The following is an introduction to the trajectory similarity (distance) measurement method based on these two characteristics.

2.3.1.1 Sequence-based similarity measurement

Trajectory data is a kind of multidimensional time series data, and many of its similarity measurements are derived from one-dimensional sequence data. Weather, stock price, etc. are sequence databases/time-series databases that appeared in the database field earlier than the trajectory data. The

similarity of one-dimensional time series data has been widely studied. The methods that can be used to measure the similarity of trajectory data includes the Longest Common Subsequence (LCSS), the Dynamic Time Warping (DTW), Hausdorff Distance, Edit Distance (ED), Discrete Fourier transform (DFT) [10] etc.

- The advantage of the LCSS approach is robustness against noise, while the disadvantage is the penalty of assigning zero similarity values to points that slightly exceed the matching area [9]. Michail Vlachos et al. [11] proposed the normalized LCSS between trajectories, , which can process the translated sequence data by introducing the scaling factor.
- DTW obtained a better distance through warping on time [12], but it is more sensitive to different baselines and scaling of sequences [9], and its sensitivity to noise is inherent [13]. Donald J. Bemdt et al. [14] proposed Modified Hausdorff Distance (MHD) based on the mean, which is more robust. Shao Fei, et al. [15] put forward the Interpolated Modified Hausdorff short (IMHD), which solved the problem about inconsistence of sample interval in trajectory space. IMHD had robustness for the change of location update strategy. Bin Lin et al. [16] followed the statistical strategy of distance between points and sets in Hausdorff, introduced a new similarity distance definition based on one way distance (OWD) of trajectories, and proved that this method was better than DTW algorithm in accuracy and performance. But it's for the trajectory data without time information.
- ED was originally used to calculate the minimum number of single-character edits when needed to convert one word w_1 into another word w_2 [17]. ED avoided distance bias caused by noise through setting the distance between unmatched point pairs to 1. However, the disadvantage was that it sets *Dist* to 1 when the distance between the point pairs slightly exceeded the threshold, so it was imprecise. Yuan Hejin et al. [18] took the quotient of ED and the length of the longer trajectory in the two trajectories as the normalized editing distance, with a threshold [0, 1].
- Edit Distance on Real sequences (EDR) enhanced the robustness of ED [19], which also overcomes the gap issues of LCSS [11]. Osman Abul et al. [20] applied EDR to trajectory similarity measurement and proposed W4M trajectory privacy protection algorithm.
- Edit distance with Real Penalty (ERP) [21] set the distance of "change" as Euclidean distance, and the distance of "increase" and "delete" as the distance between the current point and the gap of the fixed value, which reduces the anti-noise ability while increasing the measurement accuracy [17]. EDR sets the distance to 1 when "add", "delete" and "change", which performs better in noise resistance. Mehmet Ercan Nergiz et al. [22] extended the ERP method and proposed optimize points matching (OPM), which mapped the trajectory points to the canonical grid points, expressed the distance between them with log cost metric (LCM) of the bounding box of two points, and then iteratively calculated the optimal points matching distance (OPM) between trajectories by dynamic programming algorithm (DPA). Christos Fallouts et al. [23] used a sliding window on the data sequence and extracts its features, resulting in a trajectory in the feature space. These trajectories were divided into sub trajectories and represented by their minimum boundary rectangles (MBRs). The difference between ERP and DTW was that ERP didn't replicate the previous elements [21].

2.3.1.2 Location-based similarity measurement

Euclidean Distance (EUD) is the most intuitive measure of trajectory Distance. For two equal length synchronous trajectories, the average distance between sample points of trajectories on each time slice is the Euclidean distance between trajectories. If the distance between the two trajectories on each sampling time slice is less than the threshold δ, then the two trajectories share location[24]. Josep Domingo-Ferrer

et al. [25, 26] proposed a trajectory distance measurement method p%-contemporary based on EUD, which took the quotient between the average distance of the overlapped part of two trajectories and the proportion of this part in the whole trajectory as the trajectory similarity measure. Chao Wang et al. [27] added shape similarity to the measure based on Euclidean distance. Sheng Gao et al. [28] added an angle to the Euclidean distance. Zhaowei Hu et al. [29] took Manhattan distance and angle between trajectories as similarity measures. Lan Sun et al. [30] took Euclidean distance as the determination of similarity between trajectories, and the similarity degree is represented by the difference of its privacy requirements. Jing Yang et al. [31] defined (s, λ) coverage based on the angle λ and distance s between trajectories, and proposed a standardized measurement of the distance between trajectories whose range is within [0, 1].

2.3.1.3 Others

N. Pelekis etc. defines Locality In-between 2D Polylines (LIP) distance function which calculate the area of the shape formed by two 2D polylines [9]. E. Tiakas et al. [32] proposed the measurement of trajectory similarity on the road network, where the distance between trajectory points was the quotient between the shortest distance on the road network and the diameter of the road network. The distance between the trajectories was the mean of the distances between all the points on the trajectories. Yue Sun et al. [33] proposed trajectory similarity based on points of interest, taking the quotient of the number of common interest points and the number of all the interest points on two trajectories as the measure of trajectory similarity. Yossi Rubner et al. [34] proposed Earth Mover's Distance on Trajectory (EMDT) based on Earth Mover's Distance (EMD), but its efficiency is proportional to the cube of the sampling points O(n^3) [35].

2.3.2 Optimal clustering of trajectories

Optimal clustering of multidimensional sequential data is a NP problem. Osman Abul et al. [24] proposed a trajectory clustering scheme based on the greedy algorithm. Mehmet Ercan Nergiz et al. [22] selected a trajectory with the minimum distance to all trajectories in the equivalence class as the central trajectory of the cluster, and clusters with other similar trajectories in the equivalence class. Jae-Gil Lee et al. [36] defined the core trajectory by measuring the trajectory density around each trajectory and completed clustering..

Other methods map the trajectories to the trajectories distance graph in the process of clustering, and cluster the trajectories by the partition method in graph theory. Zheng Huo et al. [37] proposed the method of trajectory clustering based on the minimum cut of trajectory distance graph. Sheng Gao et al. [28] determined the connection of trajectories by whether their direction are similar, and transformed the selection of the trajectory k-anonymous set into a constrained minimum spanning tree problem. Lan Sun et al. [30] took the difference of privacy requirements as the edge weight. Jing Yang et al. [31] defined (s, λ)-coverage to determine whether to connect the trajectories, where s represents the number of the trajectory points overlapped, λ represents the angular constraint between trajectories, and the weight of edges was the combination of the normalized EUD and angular distance between trajectories. Zhaowei Hu et al. [29] defined the (l, δ)-constraint as the judgement to connect the trajectories, where l represents the Euclidean distance and δ represents the difference in privacy requirements.

2.3.3 Trajectory reconstruction within the cluster

The principle for reconstructing the trajectory data within the cluster should consider the availability of generated trajectory data and the effectiveness of privacy protection. The reconstruction should not only ensure that the trajectories within the cluster are indistinguishable, but also improve the availability of the published trajectories by minimizing the distortion of trajectories.

The first method for reconstructing trajectories within

clusters is based on aggregation. First, the center trajectory is found, and then the locus within the cluster is gathered to the center with certain rules to achieve anonymity. Osman Abul et al. [24] transformed the intra-cluster trajectory into the curved cylinder with the central trajectory as the axis, which took the uncertainty threshold δ as the diameter. Osman Abul et al. [20] achieved anonymity through unilateral editing during trajectory reconstruction. This clustering scheme strengthens the clustering characteristics of trajectories within the cluster, and the distribution of trajectory data is closer to the trajectory grouping selected by the anonymous algorithm, which is more vulnerable to secondary clustering attacks [38].

In the second strategy, trajectory reconstruction is realized by sampling or exchanging trajectory points within a certain distance threshold in the distribution region of trajectory points, which does not change the clustering characteristics of trajectory. Mehmet Ercan Nergiz et al. [22] took samples in the bounding box and reconstructed the trajectory for publication with the generated points. Josep Domingo-Ferrer et al. [25] proposed SwapTriples, a trajectory reconstruction method based on position swapping, which synthesizes trajectory data by randomly swapping positions within a certain spatial-temporal threshold and replaces the original trajectory. This method preserves the location and number of locus points in the cluster. By improving SwapTriples, Josep Domingo-Ferrer et al. [26] proposed an anonymous method of SwapLocations, this method randomly swaped positions within a cluster of points, and deleted points that do not meet the conditions of clustering and swapping. Yingjie Wu et al. [38] proposed a publishing trajectory privacy protection algorithm based on clustering hybridization. This method did not change the overall distribution of trajectory points and avoided secondary clustering. However, compared with aggregation-based strategy, its data distortion is serious.

3 Problem statement

3.1 Definitions and symbols

Definition 1 (Trajectory point) The spatial-temporal sampling position of moving object is defined as **p=**(*TID, t, x, y*) [39], where TID denotes the body of the object, t denotes the sampling time, (*x, y*) denotes the spatial coordinate.

Definition 2 (Trajectory) A collection of sampling positions belongs to the same moving object is defined as $\mathbf{T}=\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{|\mathbf{T}|}\}$ [20] , where |T| represents the number of points.

Definition 3 (Trajectory Data) A collection of trajectories is defined as $\mathbf{TD}=\{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_{|\mathbf{TD}|}\}$ [9] , where |**TD**| represents the number of trajectories and $|\mathbf{TD}|_{\mathbf{p}}$ represents the number of points.

Definition 4 (Transformed Trajectory) The releasable trajectory processed by the privacy protection algorithm to meet the requirement of anonymity is defined as $\mathbf{T}^*=\{\mathbf{p}_1^*, \mathbf{p}_2^*, \ldots, \mathbf{p}_{|\mathbf{T}^*|}^*\}$.

3.2 Adversary model

It is assumed that the adversaries possess three types ofinformation. The first is the partial trajectory points $\mathbf{P}_a$ of the target $\mathbf{T}_{a+b}$. The second is $\mathbf{TD}^*$ containing the converted target trajectory $\mathbf{T}_{a+b}^*$. Finally, the privacy protection method and relevant parameters.

The purpose is to obtain the unknown target trajectory points $\mathbf{P}_b$ through the analysis of $\mathbf{TD}^*$, $\mathbf{P}_a$ and privacy policy.

The attack strategy of adversary for trajectory data satisfying trajectory *k*-anonymity includes three tasks. First, get $\mathbf{P}_a^*$ according to $\mathbf{TD}^*$, $\mathbf{P}_a$ and privacy policy. Then, obtained the converted target trajectory $\mathbf{T}_{a+b}^*$ by according to $\mathbf{P}_a^*$. Finally, obtained the unknown target trajectory point $\mathbf{P}_b$ according to $\mathbf{T}_{a+b}^*$.

3.3 Objectives of privacy protection

The goal of the algorithm is to ensure the *k*-anonymity of the published trajectory. In other words, in the case that the privacy policy and its parameters are disclosed, the adversary cannot obtain the published trajectory corresponding to the target trajectory from the published trajectory TD* by virtue of the background

knowledge mastered by the opponent. This privacy target corresponds to the purpose of attack in the adversary model.

4 Measurement for similarity of trajectories

4.1 The method for measuring the similarity of trajectory points

Let's expand on $Sim_i = (Dist/\beta + 1)^{-\alpha}$. When α=1 and β=1, the function is simplified to $1/(Dist+1)$, which more intuitively expresses the negative correlation between distance and similarity. $Sim_i(p_1, p_2)$=1 when Dist=0, and if one of the points is missing (Dist→∞), $Sim_i(p_1, p_2)$=0. When α=2 and β=3, the attenuation trend of Sim_i is shown in the curve I of Fig.1. The other two curves (II and III) are only two possible trends of similarity decaying with distance. We do not give specific examples of functions, and the functional relationship depends on the specific application.

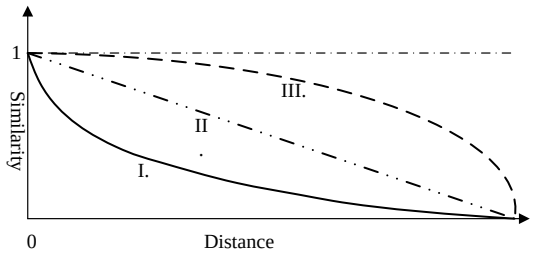


Fig.1 Three kinds of trends of trajectory point similarity changing with distance

4.2 Distance-based measurement for the similarity of trajectories (DMST)

In the location-based similarity measurement method for trajectories, we can apply the *p*%-contemporary method proposed by [25, 26] for trajectories with different start and end time.

When we measure the similarity between the trajectories with similar start and end time, if only the trajectory points with the same sampling time are compared, the comparison on a large number of asynchronous trajectory points will be discarded, and it will result in serious distortion. We adopt the following synchronization strategy for the coincidence part.

4.2.1 Synchronization of trajectory data

We adopt an interpolation method similar with reference [15] to achieve the synchronization of trajectories. It is assumed that the trajectory is a uniform linear motion between adjacent sampling points. The trajectory **T** does not sample at time *t*, but it has sampling points $\mathbf{p}_{t+}$ and $\mathbf{p}_{t-}$ at time t_+ and time t_-, respectively. The trajectory $\mathbf{T}^*$ has the sampling point $\mathbf{p}^*_t$ at time *t*. In order to synchronize **T** with $\mathbf{T}^*$ at time *t*, a sampling point $\mathbf{p}_t$ needs to be inserted into **T** at time *t*. We can get a scale parameter for the time dimension, which is $\rho = (t - t_-)/(t_+ - t_-)$. According to the assumption, **T** is uniform motion in a straight line between t_+ and t_-, so we can determine sampling position $\mathbf{p}_t = (x_- + (x_+ - x_-)\cdot\rho, y_- + (y_+ - y_-)\cdot\rho)$ at time *t*. Fig.2 illustrates the interpolation process.

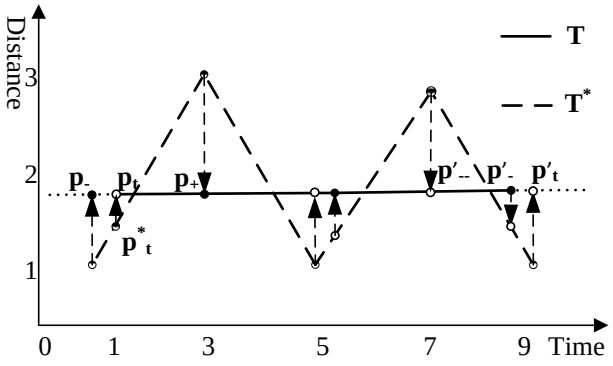


Fig.2 Trajectory synchronization through interpolation

At the endpoint of the trajectory, if there is a sampling point on one side of the interpolation point *t*, but not on the other side, the interpolation at time *t* is realized through the extension line on the sampling side. We take the interpolation at time *t* at the end of the trajectory as an example. Scale parameter is $\rho' = (t' - t'_-)/(t'_- - t'_{--})$, the sampling position at time *t* is:

$$\mathbf{p}'_t = (x'_- + (x'_- - x'_{--})\cdot\rho', y'_- + (y'_- - y'_{--})\cdot\rho'). \quad (2)$$

4.2.1 The calculation of the results in Tab.1

Table 1. The Euclidean distance from T_1

Trajectory	$Dist_{t=1}$	$Dist_{t=2}$	$Dist_{t=3}$	$Dist_{t=4}$	$Dist_{t=5}$	EUD
T_2	1.00	1.00	1.00	1.00	1.00	1.00
T_3	1.00	1.00	1.00	1.00	1.00	1.00
T_4	1.00	1.00	2.00	1.00	1.00	1.20
T_5	1.00	1.00	3.00	1.00	1.00	1.40
T_6	1.00	1.00	19.00	1.00	1.00	4.60

Table2.The Edit distance ($\alpha=1/2, \beta=1$) from T_1

Trajectory	$Dist_{t=1}$	$Dist_{t=2}$	$Dist_{t=3}$	$Dist_{t=4}$	$Dist_{t=5}$	DMST
T_2	0.71	0.71	0.71	0.71	0.71	0.71
T_3	0.71	0.71	0.71	0.71	0.71	0.71
T_4	0.71	0.71	0.58	0.71	0.71	0.68
T_5	0.71	0.71	0.50	0.71	0.71	0.67
T_6	0.71	0.71	0.22	0.71	0.71	0.61

where, the EUD between $\mathbf{T}_4$ and $\mathbf{T}_1$ at time t =3 is 2，then, $Sim_3(\mathbf{T}_1,\mathbf{T}_4)=(2/\beta+1)^{-\alpha}=(2/1+1)^{-1/2}=0.58$

5 Privacy protection algorithm based on DMST for publishing trajectories

5.1 Clustering of trajectories

Trajectory clustering consists of three steps: merge the trajectories into equivalence classes, construct trajectory relation network within an equivalence class, and construct trajectory clustering based on relation network.

5.1.1 Merge the trajectories

Clustering should take place within an equivalence class consisting of trajectories with similar lifetimes. Clustering and reconstructing trajectories with similar lifetimes can reduce the distortion of published trajectory data. The algorithm sets an offset time (π). We select an integer multiple of π as the time endpoint to cut out the trajectories, and the points at both ends of the trajectory not in this range are suppressed. The trajectories after interception are merged into the corresponding equivalence class according to the start-end time. If the number of trajectories in an equivalence class is less than k, then the trajectories in the equivalence class are released and a new equivalence class is founded until no equivalence class with a capacity not less than k is founded. Finally, the interpolation method in 4.2.1 is used to synchronize the trajectories in each equivalence class.

5.1.2 Construct a network of relationship between trajectories

We set a similarity threshold $W_{\min}$, the similarity relationship between trajectories is measured by DMST. When the similarity is no less than $W_{\min}$, the two nodes are connected to each other and the weight of the edge is set as DMST. We create a linked list of pointers on each node to point to neighboring nodes in which pointers are arranged in descending order of weights for edges. Fig.4 shows the weight of a node in a TRN and the weights of its edges that describe their similarity to the surrounding trajectories.

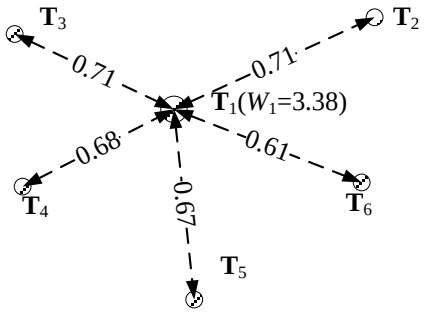


Fig. 4 The W_{edge} and W_{node} about $\mathbf{T}_1$ in TRN of Fig.3

5.1.3 Trajectory clustering based on TRN

We illustrate the process of trajectory clustering on the TRN of Fig.5. There are 6 trajectories, among which T_3 has the largest weight (W_3=4.25). Therefore, T_3 is firstly clustered as the center. When k=3, the two trajectories with the greatest similarity to T_3 are respectively T_4 and T_5, so T_3, T_4 and T_5 are clustered together. When k=4, T_6 is added to the cluster.

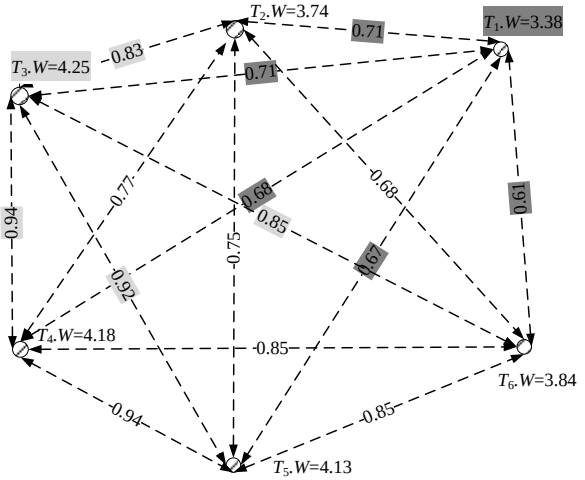


Fig.5 TRN of trajectories in Fig.3

After clustering, the trajectories without clusters were incorporated into the nearest cluster satisfying $W_{edge}>W_{min}$. The trajectories that cannot be clustered will be anonymous by deleting their *TID*s.

Algorithm 1. Trajectories Clustering

// Pseudocode for the trajectory clustering in equivalence classe EC
Input：EC, α , β , k, W_{min}
Output：The result of the trajectory clustering，The number of trajectories suppressed during trajectory clustering
Start：
Create TRN={N, NW, EW }
//N include |EC| nodes, NW is the weight set of nodes, and EW is the weight set of edges.
For i=1 to |EC|
  For j=i+1 to |EC|
    If $ew_{ij}\geq W_{min}$ then
      ew_{ij}= MDST(T_i,T_j)
      nw_i= nw_i+ MDST(T_i,T_j)
      nw_j= nw_j+ MDST(T_i,T_j)
    Endif
    $node_i$.Tag=" Active"
  Next j
Next i
Do while the node whose Tag="Active" is exist
  Select $node_i$ from TRN where nw_i=max{NW} and $node_i$.Tag="Active"
  Array="Select Top k-1 non "Clustered" nodes from $node_i$.neighbors order by *ew* desc"
  If MDST(Array[k-1], $node_i$)≥W_{min} then
    Clustering $node_i$ with the k-1 nodes in Array to TC
    Tagging the k nodes as "Clustered"
    TC.Center=$node_i$
  Else
    $node_i$.Tag="Frozen"
  Endif
Loop
Combine the obtained n trajectory clusters into {TC_1, TC_2, ..., TC_n}
For every $node_i$ whose Tag="Frozen"
  For j=1 to n
    If MDST(TC_j.Center, $node_i$)≥W_{min} then
      Add $node_i$ to TC_j
      Exit For
    Endif
  Next j
Next i
Push the nodes whose Tag="Frozen" to Trash
Output: {TC_1,TC_2,⋯, TC_n}, Trash
End

5.2. Reconstructing trajectories

To achieve *k*-anonymity, we perturb the trajectory points in the trajectory cluster.

Algorithm 2. Trajectories Reconstruction

// Pseudocode to reconstruct trajectories in cluster TC
Input： TC，*k*，W_{min}
Output：TD*
Start：
  Sort the points in TC by sampling time
  //Construct a relation network of trajectory points
  For i=1 to $|TC|_p$ //$|TC|_p$ is the number of points in TC
    j=i+1
    Do while |pi.t-pj.t|<W_{min}.t
      If $Sim_i(p_i, p_j)>W_{min}$ then Connect p_i and p_j in PRN
    Loop
  Next i
  //Reconstruction of trajectory points based on PRN

For i=1 to $|PRN|_p$ //$|PRN|_p$ is the number of points in PRN

p_i =<, t_i, x_i, y_i>

Select p* from PRN while p* connected with p_i and p*.degree$\geqslant k$

If p*.exist then p*=<TID_i, TID*, t*, x*, y*>

Next i

Store all the points in PRN to TD*

Output: TD*

End

5.3. Efficiency of the algorithm

In order to estimate the efficiency of the algorithm, we make the following assumptions:

The sampling time of all trajectories is synchronous and the number is *n*, that is to say, the length of each trajectory is *n*.

The number of trajectories in **TD** is *N*.

The number of trajectories in the equivalence class is |EC|, then the number of the equivalence class is $N/|EC|$.

The algorithm includes four steps, and the time consumption of each step is estimated as follows:

1) Merge equivalence class(EC)

In this step, the algorithm only needs to traverse all the trajectory points once to classify the trajectories of different starting and ending points into different ECs. So the time complexity is the product of the number of trajectories and the length of the trajectories $N \cdot n$.

2) Construct TRN in EC

The edge weights in TRN correspond to the similarities between trajectories. The complexity to compare the similarity between all trajectories in EC is $C_{|\mathrm{EC}|}^2 = |\mathrm{EC}| \cdot (|\mathrm{EC}| - 1)/2$. In order to calculate the similarity between two trajectories, it is necessary to calculate the similarity between trajectory points at different sampling time, and the complexity is n. Therefore, the time complexity of constructing the TRN is $C_{|\mathrm{EC}|}^2 \cdot n = \mathrm{O}(|\mathrm{EC}|^2 \cdot n)$.

3) TRN-based trajectory clustering

First, the nodes in TRN are arranged in descending order of nw, and their time complexity is O ($|EC|^2$). Then, the largest k-1 connected edges is intercepted in the linked list of adjacency pointer for the "Active" node with the largest weight. The time complexity of trajectory clustering in the EC is the product of trajectory capacity $|EC|^2$ and number of clusters |EC|/k in the EC: $|\mathrm{EC}|^2 \cdot |\mathrm{EC}|/k = |\mathrm{EC}|^3/k$

4) Trajectory reconstruction in the cluster

When the time window length is 1, the time complexity of calculating the distance between all locus points on the time window is the square of the number of locus in the cluster. In view of the uncertainty of cluster size, we take the number of trajectories in the cluster as the minimum value k, and then the time complexity of trajectory reconstruction is $O(k^2)$.

Therefore, for trajectory data **TD**, the time complexity of implementing *k*-anonymity can be expressed as follows:

$$\begin{aligned} & N \cdot n + \left(|\mathrm{EC}|^2 \cdot n + |\mathrm{EC}|^3/k + k^2\right) \cdot N/|\mathrm{EC}| \\ & = N^2 \cdot n^2 \cdot |EC|^2 + N^2 \cdot n \cdot |EC|^2/k + N^2 \cdot n \cdot k^2 \end{aligned} \quad (4)$$

It can be seen from equation (4) that the efficiency of the algorithm is proportional to the number of trajectories in **TD**, the length of trajectories, the number of equivalence classes, and the anonymous parameters. We carry out detailed experiments in the sixth part.

5.4. Analysis about the anonymity and availability

5.4.1. The anonymity analysis of the published trajectories

In the trajectory privacy protection algorithm based on clustering and reconstruction, the purpose of clustering is to reduce information distortion during reconstruction, while the purpose of reconstruction is to anonymize the trajectories within the cluster. Below, we analyze the effectiveness of the reconstruction process in terms of privacy protection.

The reconstruction of trajectory can be regarded as disturbance of trajectory point on time slice. According to the perturbation method proposed by us, each trajectory point $\mathbf{p}_i$ is translocated to an adjacent location $\mathbf{p}^*$ and a new point $\mathbf{p}_i$* is formed. The degree of $\mathbf{p}^*$ is not less than k, so the number of the trajectory points that can be translocated to $\mathbf{p}^*$ is not less than k. The probability of extrapolating the position of $\mathbf{p}_i$ from $\mathbf{p}_i^*$ is no greater than $1/k$, that is, $\mathbf{p}_i$ satisfies the k-anonymity requirement after being translocated to $\mathbf{p}^*$. If the degree of each adjacent trajectory point of $\mathbf{p}_j$ is less than k, $\mathbf{p}_j$ cannot be translocated to a position satisfying k-anonymity, then we simply remove its TID_j. This causes that there is no $\mathbf{p}_j$ in the trajectory of TID_j. $\mathbf{p}_j$ becomes an outlier in the trajectory data, so it is impossible to reveal the privacy of the trajectory subject TID_j though $\mathbf{p}_j$.

5.4.2. The availability analysis of the published trajectories

The distortion of published trajectory data mainly comes from two aspects: one is the suppression of trajectory points that cannot be anonymized, and the other is the distortion of position generated when trajectory points are anonymized. In our algorithm, we did not delete the trajectory point that could not be anonymized, but only it's *TID*. This is conducive to analysis based on trajectory points, such as the density of trajectory points in a particular space-time. In the transformation of trajectory point $\mathbf{p}_i$, we did not translocate it to a new spatial-temporal position, but chose an existing trajectory point $\mathbf{p}^*$ as its target position. The conversion based on the original location of trajectory points not only avoids the conflict between dummy trajectory points and the road network, but also ensures the authenticity of the location of trajectory points in the published data, which is conducive to obtaining more real information through the published data. Even so, the translocation causes the distortion of the trajectories, and we carried out detailed experiments on the distortion of the trajectories in the sixth part.

6 Experiments and analysis

Our strategy enables $\mathbf{TD}^*$ meet k-anonymous privacy requirements, which has been discussed in **5.4.1**. This section verifies the execution efficiency of the algorithm and the information loss of the generated trajectory data $\mathbf{TD}^*$.

6.1. Baseline algorithms for comparison

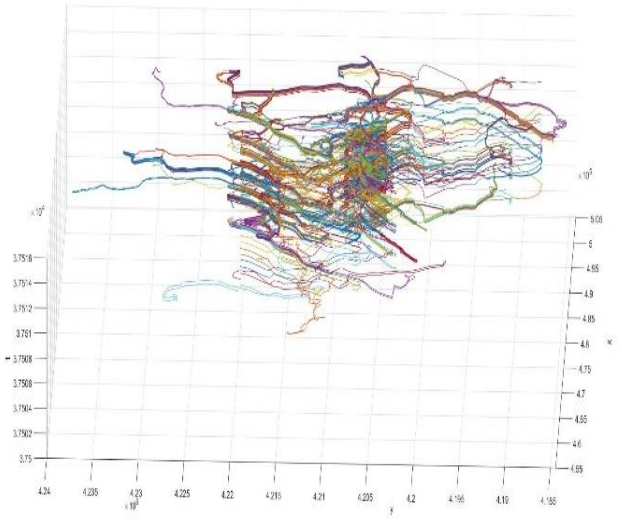
We chose three typical privacy protection methods for publishing trajectories as the baseline algorithms. Never walk alone (NWA) [24] is a typical algorithm based on clustering and reconstruction that takes trajectory k-anonymity as the privacy protection target. Dynamic Location Based Generalization (DLBG) [40] based on aggregation and generalization of trajectory points, which takes trajectory segment k-anonymity as the privacy protection target. IPKN [39] based on interchanging of positions and takes trajectory point k-anonymity as the privacy protection target. Our algorithm (DPPT) takes trajectory clustering and reconstruction as the framework, and introduces DMST as similarity measurement. Trajectory clustering method based on weighted network, and trajectory reconstruction method based on translocation of trajectory points. DPPT is closely related to but significantly different from the three typical baseline algorithms.

6.2. Experimental data

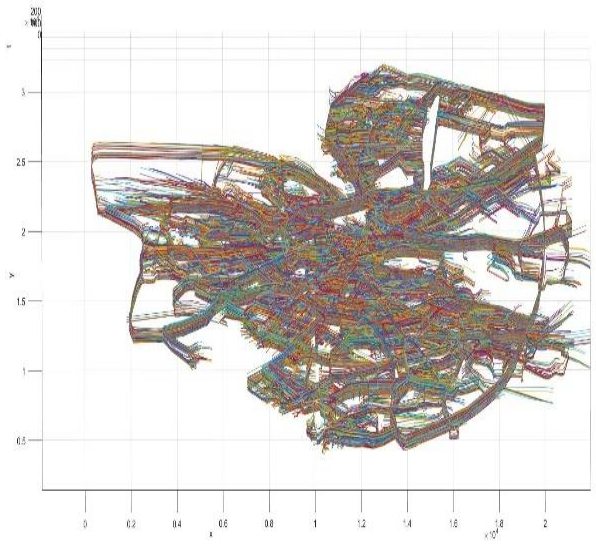
To test the effectiveness of the algorithm, we carried out experiments on both real-life data and synthetic data.

Trajectories of the construction trucks is real-life trajectory data. It contains 112,203 trajectory points, which make up 273 trajectories. The trajectory data spans 24 hours, reflects the case that 50 trucks to deliver concrete to several construction sites around Athens in 33 days. We ignored the date difference between trajectories in the experiment. This set of data is called Trucks and is distributed as shown in Fig.6 (a).

Oldenburg is a set of synthetic data based on roads in Oldenburg. It contains 47, 80,954 trajectory points, which make up 100,000 trajectories. The trajectories are generated by the Brinkhoff data generator. This set of data is called Oldenburg, and its distribution is shown in Fig.6 (b).



(a)



(b)

Fig.6 Two graphical representations of the trajectory data used in experiment (a) Real-life trajectory data (Trucks); (b) Synthetic trajectory data (Oldenburg)

In Tab.3, we describe the basic characteristics of the test data, including trajectory number |**TD**|, coverage area, and time span (T_{span}), average speed (AS), and trajectory point density.

Tab.4 shows the suppression of trajectory points and equivalence class of the two groups of experimental data after trimming.

Table 3 Characteristic parameters of the two **TD**s

TD	**Trucks**	**Oldenburg**
$\|\mathbf{TD}\|_p$	112203	4780954
|TD|	273	100000
Area(km^2)	2403.3	634.49
T_{span}(s)	79208	4350
AS(m/30s)	91.04	93.21
Density	5.89E-04	1.73E+00

Table 4 Sorting results of the two **TD**s

TD	**Trucks**	**Oldenburg**
$\boldsymbol{\pi}$	415	5
Number of EC	12	435
Number of suppressed trajectory points	36365	491928
Ratio of suppressed trajectory points	32.41%	10.29%

6.3. Setting of experimental parameters

In **TD**, information loss includes data suppression and data distortion. Data suppression refers to the removal of trajectories and trajectory points, while data distortion refers to the deviation of trajectory points after conversion. We set the same anonymous parameters k and data suppression rate for the three baseline algorithms and our DPPT algorithm.

Table 5 The parameters in four algorithms

Algorithm	**Parameters**
NWA	R_{max0}=1000;ΔR_{max}=100;δ=0
DLBG(Trucks**)**	r=500;R=500;ΔR=200
DLBG(Oldenburg **)**	r =50;R =100;ΔR =50
IPKN	S_{e0}=50; ΔS_e=10; S_s= 0; T_e= S_e/v; T_s= 0
DPPT	α=1; β=10;W_{min}=0.8; ΔW=0.1

In NWA, the data suppression comes from merging equivalence classes, trajectories clustering and trajectory points reconstruction. DLBG suppressed noise data when it used the aggregation method to obtain RR and GR. When IPKN iteratively constructs k-core subnet of TN, it suppressed trajectory points whose degree are less than k. DPPT suppressed data for the same reason as NWA. In NWA, we limit the

suppression rate caused by trajectory clustering to less than 10%, and the total suppression rate on Trucks and Oldenburg was 42.41% and 20.29%, respectively. We take them as the maximum of the suppression rate in the other three algorithms. The parameters corresponding to the four algorithms are shown in Tab.5.

6.4. Results and analysis

6.4.1. Information loss of published data

To analyze the availability of published data, we use the following formula to calculate the information loss of published data. The less the information loss, the better the availability.

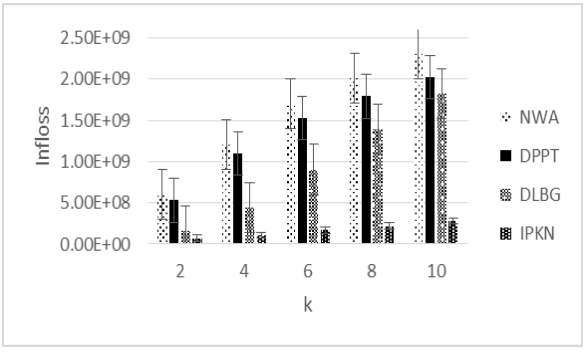
$$\text{Infloss} = \sum_{i=0}^{|TD^*|_p} Dist(p_i, p_i^*) \qquad (5)$$

where *Dist* is the Euclidean distance between $\mathbf{p}_i$ and $\mathbf{p}_i^*$:

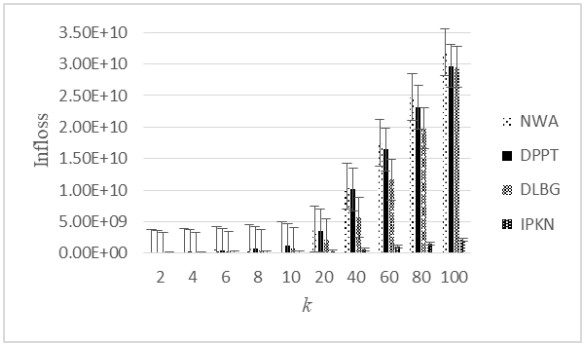
$$Dist = \left[\left(\Delta x^2 + \Delta y^2 + \Delta t^2 v^2 \right)^{\frac{1}{2}} \right] \qquad (6)$$

We conduct two sets of experiments. The first set of experiments compared different algorithms under the parameters in Tab.6. Another set of experiments compared DPPT algorithms with different parameters. Each privacy protection algorithm is executed 50 times in each data set, and The experimental results are analyzed statistically as follows

6.4.1.1 Information loss of the first set of experiments



(a)



(b)

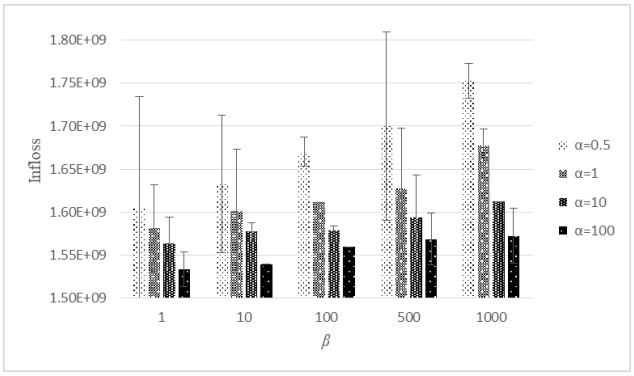
Fig.7 Information loss caused by different algorithms to trajectory data (a) Information loss in Trucks; (b) Information loss in Oldenburg

DPPT keeps the ideal level of privacy protection while reducing the loss of information. Due to the reduction of the dimension of data anonymity, the ability of privacy protection is gradually (4) reduced. NWA and DPPT can guarantee *k*-anonymity of trajectories, but DPPT generates less information loss. By contrast, DLBG can only guarantee anonymity of trajectory segments, and IPKN can only guarantee anonymity of trajectory points.

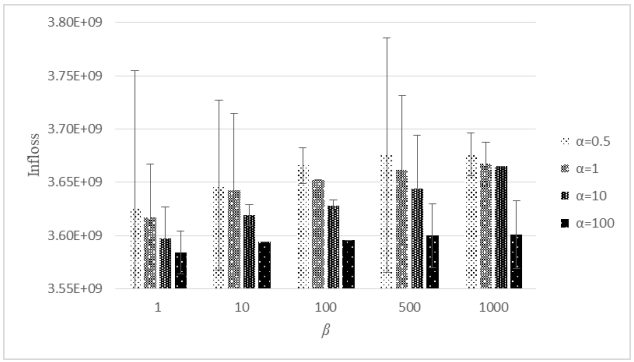
Trajectory density is another important factor in information loss. Oldenburg data volume is 42.61 times of the Trucks, data density is 2937 times of the Trucks, and the ratio of information loss of the two data in the same parameter is about 0.32, so information loss and track density is negatively correlated.

6.4.1.2 The Information loss of the second set of experiments

It can be seen from Fig.8 that the experiments reflect the commonality on two aspects. First, the information loss caused by DPPT is negatively correlated with α. As α increases, similarity becomes more sensitive to distance, resulting in more accurate results and less information loss. Second, the information loss is negatively correlated with β that reflects the granularity of similarity comparison. The larger the granularity, the broader the criterion of similarity, and the greater the information loss.



(a)



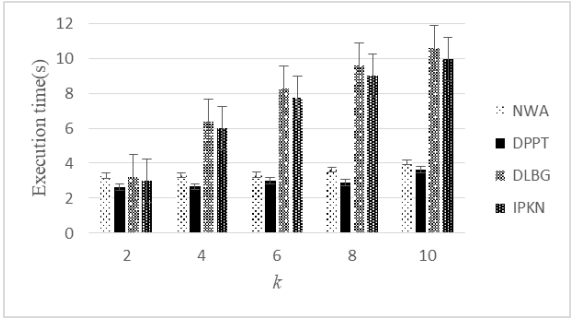
(b)

Fig.8 Information loss caused by DPPT on trajectory data under different parameters; (a) Information loss in Trucks; (b) Information loss in Oldenburg

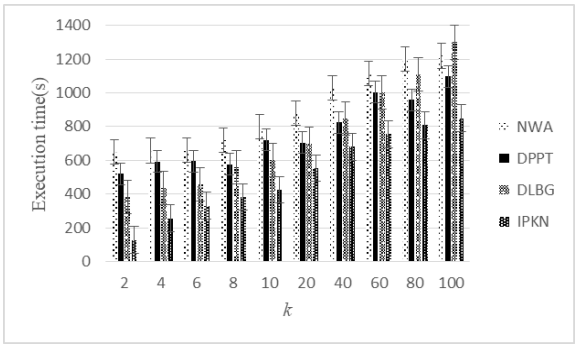
6.4.2. The algorithm efficiency

During the execution of the two experiments, we recorded the execution time. By comparing the execution time, we can compare the efficiency of different algorithms and the efficiency of DPPT under different parameters.

6.4.2.1 The execution time of the first set of experiments



(a)



(b)

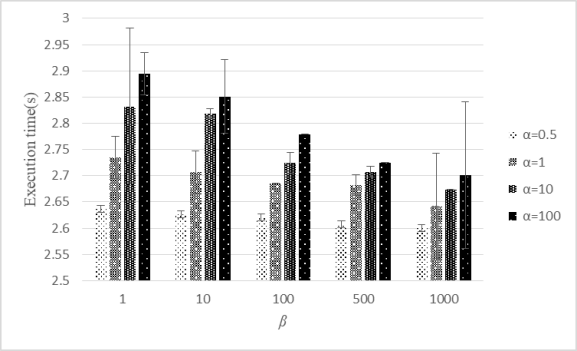
Fig.9 Execution time of different algorithms on two sets of data (a) Execution time on Trucks; (b) Execution time on Oldenburg

In Trucks, the algorithms based on trajectory anonymity (NWA and DPPT) consume less execution time significantly than algorithms based on trajectory segment anonymity (DLBG) and trajectory point anonymity (IPKN).

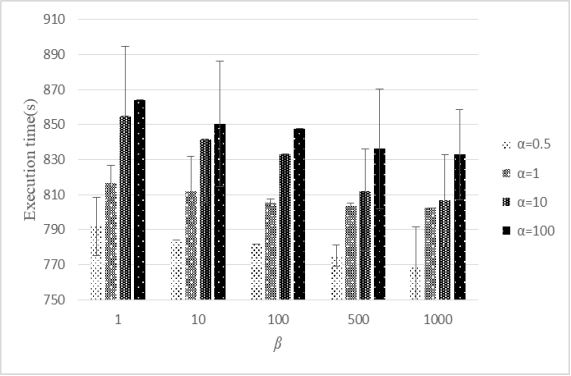
In Oldenburg, the opposite effect is found. The algorithms based on trajectory anonymity consume more time, which is determined by the characteristics of trajectories. The length of trajectories in Oldenburg is significantly small than the Trucks, which results in a significant increase in the operate objects of NWA and DPPT for the same amount of data. In addition, in the process of constructing a larger trajectory graph, the trajectory relations that need to be calculated also increase to the power with the number of trajectories $O(N^2)$.

6.4.2.2 The execution time of the second set of experiments

Fig.10 shows that the execution time increases with α increasing. This is due to the fact that the influence of distance on similarity between trajectory points decreases rapidly with increasing of α.



(a)



(b)

Fig.10 DPPT execution time under different parameters (b) Execution time on Trucks; (b) Execution time on Oldenburg

In order to reach the required *k* value, the similarity threshold $W_{\min}$ should be reduced. When the similarity threshold decreases ΔW, the number of trajectory pairs that require similarity comparison increases with α. This is the main reason why execution time increases with α. From the experiments on the two sets of data, it can be found that the execution time of the algorithm decrease with the increase of β.

7 Conclusion and future work

The excellent noise resistance and precision of DMST are derived from the application of a similarity measurement method (Sim_i). Sim_i has excellent scalability and can be combined with the similarity determination method of sequence data to obtain a more robust and sensitive sequence data query application.

We analyzed the effectiveness of DPPT algorithm for trajectory *k*-anonymous privacy requirements. Experimental results show that the algorithm is superior to the classical NWA algorithm in information loss and execution efficiency. In terms of privacy level, it is better than both the privacy protection algorithm based on trajectory segment anonymity and the privacy protection method based on trajectory point anonymity.

Compared with the privacy protection method based on trajectory point anonymity, our method has a higher level of privacy protection, but at the expense of huge information loss. How to improve the availability of published data based on a higher level of privacy is a problem that needs further research. In the process of continuous production of trajectory data, how to avoid the revelation of privacy during the combination of successively published trajectory data will not reveal privacy is also a problem to be studied in the future.

References

1. Zhang, S., Q. Liu, and G. Wang, *Trajectory privacy protection method based on location obfuscation.* Journal on Communications, 2018. **39**(7): p. 11.
2. Luper D , C.D., Miller J. *Spatial and Temporal Target Association through Semantic Analysis and GPS Data Mining*. in *Proceedings of the 2007 International Conference on Information & Knowledge Engineering*. 2007. Las Vegas, Nevada, USA. DBLP.
3. You, T.H.P., Wen Chih， Lee, Wang Chien. *Protecting Moving Trajectories with Dummies*. in *International Conference on Mobile Data Management*. 2008.
4. Lei, K., X. Li, and H. Liu, *Dummy trajectory privacy protection scheme for trajectory publishing based on the spatiotemporal correlation.* Journal on Communications, 2016. **12**: p. 9.
5. Gruteser, M. and L. Xuan, *Protecting privacy in continuous location-tracking applications.* IEEE Security & Privacy, 2004. **2**(2): p. 28-34.
6. Abul, O., et al. *Hiding Sensitive Trajectory Patterns*. in *IEEE International Conference on*

Data Mining Workshops. 2007.

7. Bonchi, F., L.V.S. Lakshmanan, and H. (Wendy)Wang, *Trajectory anonymity in publishing personal mobility data.* ACM SIGKDD Explorations Newsletter, 13(1):30., 2011: p. 13.
8. Mohammed N , F.B.C.M., Debbabi M . . *Walking in the crowd_ Anonymizing trajectory data for pattern analysis*. in *ACM Conference on Information & Knowledge Management. ACM.* 2009.
9. Pelekis, N., et al. *Similarity Search in Trajectory Databases*. in *International Symposium on Temporal Representation & Reasoning*. 2007.
10. Agrawal, R., C. Faloutsos, and A.N. Swami. *Efficient Similarity Search In Sequence Databases*. in *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*. 1993.
11. Vlachos, M., G. Kollios, and D. Gunopulos. *Discovering similar multidimensional trajectories*. in *International Conference on Data Engineering*. 2002.
12. Keogh, et al., *Exact indexing of dynamic time warping*. Knowledge & Information Systems, 2005. **7**(3): p. 358-386.
13. Byoung-KeeYi, H.V. Jagadish, and C. Faloutsos. *Efficient retrieval of similar time sequences under time warping*. in *Fourteenth International Conference on Data Engineering*. 1998.
14. Berndt, D. and J. Clifford, *Using dynamic time warping to find patterns in time series.* AAAI-94 Workshop on Knowledge Discovery in Databases, 1994: p. 12.
15. Fei, S., S. Cai, and J. Gu. *A modified Hausdorff distance based algorithm for 2-dimensional spatial trajectory matching*. 2010.
16. Lin, B. and J. Su, *One Way Distance: For Shape Based Similarity Search of Moving Object Trajectories.* Geoinformatica, 2008. **12**(2): p. 117-142.
17. Zhang, Y., *Objects recognition and unusual events modelling&analysing in intelligent video surveillance*, in *Institute of image processing and pattern recognition*. 2009, Shanghai Jiao Tong University.
18. Yuan, H., et al., *A Trajectory Pattern Learning Approach Based on the Normailzed Edit Distance and Spectral Clustering Algorithm.* Journal of computer aided design & computer graphics, 2008(6): p. 73-78.
19. Chen, L. and V. Oria. *Robust and fast similarity search for moving object trajectories*. in *ACM Sigmod International Conference on Management of Data*. 2005.
20. Abul, O., F. Bonchi, and M. Nanni, *Anonymization of moving objects databases by clustering and perturbation.* Information Systems, 2010: p. 28.
21. Chen, L. and R. Ng, *On The Marriage of Lp-norms and Edit Distance*. VLDB, 2004.
22. ErcanNergiz, M., M. Atzori, and e. al. *Towards trajectory anonymization: a generalization-based approach*. in *Sigspatial ACM GIS International Workshop on Security & Privacy in GIS & Lbs. IIIA-CSIC.* 2008.
23. Faloutsos, C., M. Ranganathan, and Y. Manolopoulos, *Fast Subsequence Matching in Time-Series Databases.* ACM SIGMOD Record, 1994. **23**(2).
24. Abul, O., F. Bonchi, and M. Nanni. *Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases*. in *IEEE International Conference on Data Engineering. IEEE Computer Society.* 2008.
25. Domingo-Ferrer, J., M. Sramka, and R. Trujillo-Rasúa. *Privacy-preserving publication of trajectories using microaggregation*. in *ACM Sigspatial International Workshop on Security & Privacy in GIS & Lbs. ACM.* 2010.
26. Domingo-Ferrer, J. and R. Trujillo-Rasua, *Microaggregation- and permutation-based anonymization of movement data.* Information Sciences, 2012. **208**: p. 26.
27. Wang, C., J. Yang, and J. Zhang, *Privacy preserving algorithm based on trajectory location and shape similarity.* Journal on Communications, 2015. **36**(2): p. 14.

28. Gao, S., et al., *Balancing trajectory privacy and data utility using a personalized anonymization model*. Journal of Network and Computer Applications, 2014: p. 10.
29. Hu, Z., J. Yang, and J. Zhang, *Trajectory privacy protection method based on the time interval divided*. computers & security, 2018: p. 12.
30. Lan, S., et al., *Personalized privacy preserving algorithm for trajectory data publishing.* Systems Engineering and Electronics, 2014: p. 6.
31. Yang, J., et al., *Personalized trajectory privacy preserving method based on graph partition.* Journal on Communications, 2015. **36**.
32. Tiakas, E., et al., *Searching for similar trajectories in spatial networks.* Journal of Systems & Software, 2009. **82**(5): p. 772-788.
33. Sun, Y., *Research on Privacy Preservation in the Publication of Trajectories*. 2014, Harbin Engineering University.
34. Rubner, Y., C. Tomasi, and L.J. Guibas, *The Earth Mover's Distance as a Metric for Image Retrieval*. International Journal of Computer Vision, 2000. **40**(2): p. 99-121.
35. He, D., et al., *Efficient and robust data augmentation for trajectory analytics: a similarity-based approach.* World Wide Web, 2019(12).
36. Jae-Gil Lee, J.H., Kyu-Young Whang, *Trajectory clustering: a partition-and-group framework.* ACM XXX, 2007.
37. Huo Z, H.Y., Meng X. *History trajectory privacy-preserving through graph partition*. in *Proceedings of the first international workshop on mobile location-based service. ACM.* 2011.
38. Wu, Y., et al., *A clustering hybrid based algorithm for privacy preserving trajectory data publishing.* Journal of Computer Research and Development, 2013. **50**(3): p. 16.
39. Wang, S., et al., *Interchange-based Privacy Protection for Publishing Trajectories.* IEEE Access, 2019. **7**(1): p. 16.
40. Xin, Y., Z. Xie, and J. Yang, *The privacy preserving method for dynamic trajectory releasing based on adaptive clustering*. Information Sciences, 2017: p. 13.

1