Interchange-based Privacy Protection for Publishing Trajectories

Shuai Wang1,2, Chunyi Chen*1, Guijie Zhang2 and Yu Xin3

1 School of Computer Science and Technology, Changchun University of Science and Technology, 7089 Weixing Road, Changchun 130022, China
2 Teacher training center of colleges and universities of Jilin province, Jilin Normal University, Siping, Jilin 136000, China
3 College of Computer Science and Technology, Ningbo University, Ningbo, 315211, China

Corresponding author: Chunyi Chen (e-mail: chenchunyi@hotmail.com).

This work was supported by the National Science Foundation of China (NSFC) under Grant No.61602133, Science and Technology Development Plan Project of Jilin Province under Grant No.20180519012JH, Scientific Items of Jilin Provincial Department of Education under Grant No.JKH20191025KJ, Science and Technology Development Plan project of Siping No.2016058, 2017098.

ABSTRACT

Information extracted from trajectory data is very useful in many practical application scenarios. Before trajectories for data mining are published, they need to be processed to protect the privacy of the trajectories' bodies. In this paper, a method for such privacy protection is proposed. Our method guarantees that the generated trajectory points satisfy the k-anonymity by interchanging the positions of the trajectory points on the k-core subnet of the relation network. The method treats the trajectory points as the privacy protection object. It overcomes the curse of dimensionality resulting from the K-anonymity of trajectories, and reduces the distortion of the generated trajectories significantly. Moreover, our proposed strategy can preserve the original positions of the trajectory points. Experiments on both real-life and synthetic data sets are carried out with different methods for comparison. The results show that our method has greater efficiency and lower distortion of the processed trajectories.

INDEX TERMS


I. INTRODUCTION

With the application of mobile communication devices and the development of positioning technology, the location-based services (LBS) have proliferated. Location-service providers (LSP) can obtain massive trajectories by content and geographical locations of service requests.

The trajectory characteristics can be applied to many fields such as intelligent transportation and supply chain management, but the trajectory owners may need to publish trajectories to untrustworthy third parties for analysis. If the third parties associate the trajectories with specific individuals, privacy may be threatened [1]. So trajectories must be processed for privacy protection before being published [2, 3]. This paper focuses on the methods for privacy protection of publishing trajectories.

K-anonymity is an important privacy protection principle for relational database. Given a set of quasi-identifiers (QID), at least K indistinguishable records should be associated in the database. It takes two steps to achieve K-anonymity of trajectories: clustering and generalization [4]. In the first step, each trajectories are clustered such that each cluster contains at least K similar trajectories. In order to make K trajectories satisfy the similarity condition, the average distance between \((nK^2-nK)/2\) pairs of points in K trajectories cannot exceed the cluster radius, in which \(n\) is the average length of the trajectories. In the generalization step, the trajectories in each cluster are reconstructed into indistinguishable K trajectories. This leads to distortion, which is positively correlated with the cluster radius. The cluster radius increases rapidly according to \(n\), which aggravates data distortion during trajectory reconstruction. This is known as the curse of dimensionality [5].

In order to reduce the data distortion, a typical approach is to desensitize the trajectory points by generalizing them into an area, which effectively reduces data distortion. However, it does not quantify the level of privacy protection [6]. Another approach is to realize desensitization of trajectory points through the swapping of positions between neighbor trajectory points, which not only reduces data distortion, but also preserves the original position of trajectory data [7]. However, the trajectories generated by this approach do not satisfy the K-anonymity of trajectory points.

This paper proposes a privacy protection method for publishing trajectories. It uses the approach of interchanging positions between neighbor trajectory points. It maps the interchangeable relations between neighbor points in the trajectories to a relation network of trajectory points, and then interchanges the positions of the neighbor
points in the K-core subnet of this network[8]. This method aims not only to reduce the data distortion, but also to preserve the original positions of the trajectory data, so that the generated trajectory points satisfy the K-anonymity.

The main contributions of our method are as follows:

First, the method discards the K-anonymity requirement to trajectories, and treats trajectory points as the privacy protection objects. The reduction in granularity of the operational objects significantly reduces data distortion.

Second, our method proposes a Shell-similar Interchangeable Region (SIR), in which the positions of trajectory points can be interchanged. The interchanges of positions in SIR make the published trajectories satisfy the requirements of desensitization and availability. This perturbation strategy based on interchange retains the home positions of trajectory points.

Third, our algorithm maps the interchangeable distance relationship between trajectory points to a network, and interchanges the positions of neighbor nodes on the K-core subnet of this relationship network. Then we get the published trajectories in which the trajectory points satisfy K-anonymity.

Fourth, the effectiveness and feasibility of our method is verified by comparing with three classical methods on real-life and synthetic data sets.

This paper is organized as follows. The first section introduces the research status and the main contributions of this paper. The second section describes the related work of privacy protection for publishing trajectories. The fourth section introduces the algorithm IPKN in detail, including the modeling of SIR, the mapping algorithm of the trajectories to the K-core subnet; and then introduces the Interchanging of Positions in K-core subnet of Trajectory point’s relation Net (IPKN). The fifth section conducts experiment and analysis, selects three typical privacy protection methods for publishing trajectories as comparing algorithms of IPKN, analyzes and compares the efficiency execution and the availability of the generated trajectories. The sixth section summarizes our work.

II. PROBLEM STATEMENT

A. DEFINITIONS AND SYMBOLS

Definition 1 Point: the spatial-temporal sampling position of moving object, which is denoted by \( p \), and represented as a \textit{tetrad} \( (TID, t, x, y) \), where \( TID \) denotes the body of the object, \( t \) denotes the sampling time, \( (x, y) \) denotes the spatial coordinate. \((t, x, y)\) is called a \textit{triple}.

Definition 2 Trajectory: A collection of sampling positions belongs to the same moving object[9], denoted by \( T = \{ p_1, p_2, ..., p_{|T|} \} \), where \(|T|\) represents the number of points in \( T \).

Definition 3 Trajectory Database: A collection of trajectories[10], denoted by \( TD = \{ T_1, T_2, ..., T_{|TD|} \} \), where \(|TD|\) represents the number of trajectories. Intuitively, \( TD \) is a set of points : \( \{ p_1, p_2, ..., p_{|TD|} \} \), where \(|TD|\) represents the number of points in \( TD \).

Definition 4 Transformed Trajectory Database: According to certain rules, the replacement of \( p_i \) with other region is called position transformation of \( p_i \), in which \( p_i \) is a trajectory point of \( T_i \), \( 0 \leq i \leq |T| \), and the region may be a point or an area, or empty, denoted by \( p^*_i \). The new trajectory is called transformed trajectory of \( T_i \), denoted by \( T^*_i \). The new trajectory database is called the transformed trajectory database of \( TD \), denoted by \( TD^* \). Typically, the published trajectory database is also represented by \( TD^* \).

Definition 5 K-Anonymity of Trajectories: For any \( T \) in \( TD \), if its \( |T| \) points can only be associated with at least \( K \) trajectories in \( TD^* \), that is, the probability of determining the corresponding unique \( T \) in \( TD^* \) is less than \( 1/K \), then we say that \( TD^* \) satisfies K-anonymity of trajectories.

Definition 6 K-Anonymity of Trajectory Points: If the probability of getting the home position of \( p \) from \( p^*_i \) is no more than \( 1/K \), then we say that \( TD^* \) satisfies K-anonymity of trajectory points, \( K \) is the anonymity threshold of \( TD^* \).

Definition 7 K-core subnet: A network, in which the degree of any node is not less than \( K \), is called K-core network. In network \( G \), we repeatedly delete all the nodes whose degrees are less than \( K \), until there are no nodes with degree less than \( K \), then we obtain the K-core subnet of \( G \) [8].

B. ADVERSARY MODEL

The background information that an adversary may obtain includes: \( TD^* \); privacy protection strategy and its parameters; a collection \( P_i \) that contained the trajectory points of an attacked target \( T_i \).

The privacy attack oriented to the \( TD^* \) that satisfies K-anonymity of trajectories can be described as: the adversary associates trajectories in \( TD^* \) with the trajectory points in \( P_i \), then they can infer other sensitive information of \( T_i \). Suppose that adversary can find at least \( n \) associated trajectories in \( TD^* \), and each of them may be the \( T^*_i \) obtained from \( T_i \) by K-anonymity method. Then the probability of adversary to determine \( T^*_i \) is no more than \( 1/n \). If \( n < K \), the attack is considered successful.

The \( TD^* \) that satisfies K-anonymity of trajectory points may not satisfies K-anonymity of the trajectories. In extreme cases, adversary can determine the unique \( T^*_i \) in \( TD^* \) according to \( P_i \). Suppose that \( n \) points can be found in the neighborhood of \( p^*_i \) on \( T^*_i \), where each position may be the actual position of \( p \) on \( T_i \). Then the probability for adversary to determine \( p \) is no more than \( 1/n \). If \( n < K \), the attack is considered successful.

III. RELATED WORK

The privacy protection of relational database has been studied comprehensively. With the popularity of GPS and mobile communication technology, various LBS come into being[11]. LBS generates a large amount of \( TD \), which can
be used for data mining. We need to process TD before it is published. The purpose is to ensure that adversary cannot obtain sensitive information of the TD from TD*. This process is called privacy protection for publishing trajectories [12].

A. CLASSIFICATION OF PRIVACY PROTECTION FOR PUBLISHING TRAJECTORIES

Distinguished by the operation to TD, the privacy protection methods for publishing trajectories are divided into three categories, which correspond to the three basic operations to relational database (add, delete, and update).

1. The methods based on adding dummy data
   - The main idea is to add dummy data into the original data. The dummy data is generated according to the characteristics of TD. This category improve the anonymity level of mixed data on the premise of ensuring no serious data loss[13, 14]. This category increases the amount of data processing[15]. And due to the spatial-temporal correlation and multi-dimensionality of the trajectories, the existing dummy methods have a success rate of no more than 15% in protecting the privacy of users' trajectories [16]. This category is commonly used in privacy protection for LBS.

2. The methods based on suppressing sensitive data
   - This category aims to delete sensitive points before publishing TD [17-20]. The sensitive points can be specific to the problem: positions frequently visited by users, positions with important semantic features, etc. This category is based on QID-aware, which has specific limitations on the background knowledge of adversary. The difficulty of this category lies in understanding the sensitive points in TD and the background knowledge of adversary accurately [11]. The massive suppressed sensitive points lead to a large amount of information loss, which limits the availability of TD*.

3. The methods based on generalization
   - The generalization method generates TD* that represents TD. TD* prevents the sensitive information disclosure or reduces the probability of the adversary identifying the bodies of the trajectories, while preserving the important features of TD. In TD*, T* is not a new trajectory, but represents a range of T, so it can be regarded as a generalization of T. The generalization method neither forms too large redundant data like dummy methods, nor causes serious data loss such as the suppression methods. This paper focuses on the privacy protection method based on generalization.

B. CLASSIFICATION OF METHODS BASED ON GENERALIZATION

The methods based on generalization can be divided into two categories according to the object of privacy protection.

1. The methods based on generalization of trajectories takes the identities of trajectories as the privacy protection object. Anonymity is the main category for this kind of privacy protection method[21]. The notion of K-anonymity has been originally proposed for relational databases. The original proposal to achieve K-anonymity of trajectories was based on generalization and suppression of the information contained in the QID attributes[22]. It usually includes two steps: First, clustered the trajectories whose similarity[2, 4, 9, 23-25] is less than cluster radius, and ensured the number of trajectories in the cluster is not less than K[4, 23, 25-29]. Then, by reconstructed the trajectories in each cluster, we obtain the TD* which satisfies the K-anonymity of trajectories [2, 4, 7, 9, 25, 30].

   - The privacy protection methods based on clustering trajectories cause serious information loss. Under the synchronous sampling scenario, the similarity of K trajectories is that the average distance of \((nK^2-nK)/2\) point pairs does not exceed the cluster radius, in which n is the average length of trajectories. This is called the curse of dimensionality[5]. In the process of clustering, the cluster radius increases with the trajectory length n and the anonymous threshold K, so as not to suppress a large number of trajectories that do not meet the cluster radius. But the information loss caused by the reconstruction of trajectories is positively correlated with the cluster radius. The increase of cluster radius will cause serious data distortion.

   - The other kind of method is based on the generalization of trajectory points. These methods eliminated the anonymity requirement of trajectories, effectively avoided the curse of dimensionality, and improved the availability of TD*. These methods are divided into two categories according to the selection of points to be generalized.

   - 2.1 The first category only takes the sensitive trajectory points as the protected objects. The trajectory points that satisfy certain spatial-temporal characteristics were defined as the sensitive trajectory points subjectively and generalized, while the non-sensitive trajectory points were directly published[31-33]. This method needs to design privacy protection policies according to the different definitions to sensitive point.

   - 2.2 The second category takes all the trajectory points as the protected objects. Monreale et al.[6] proposed a typical generalization method for all trajectory points. First, the important points were clustered, which include the initial points, the end points, the significant inflection points and the obvious stagnation points. Then the Voronoi cells were generated based on the centers of the clusters. Finally, the sampling positions distributed in the cells were replaced by the center of the cells to form the generalization trajectories. These generalization trajectories look like a network [34], where the generalization region can be regarded as nodes in the network.

   - Yu X. et al.[34] proposed the Gibbs sampling-cluster method within the strategic framework of [6], formed the representative area RR through multiple convergences of the trajectory points. And then aggregated the RR
secondarily to form the generalization area GR. At last, the
generalized trajectory network was published as \( \text{TD}^* = \{ \text{T}_1, \text{T}_2, \ldots, \text{T}_{\text{TD}} \} \), where \( \text{T}^* = \{ (t_i, \text{GR}_1), (t_j, \text{GR}_2), \ldots, (t_r, \text{GR}_p) \} \). In order to make the trajectory segments in
\( \text{TD}^* \) satisfy \( K \)-anonymity, GR is required to contain at least \( \lceil \sqrt{K} \rceil \) RRs. The result under this requirement is similar to the
\( K \)-core subnet of \( \text{TD}^* \) [34].

Domingo F. et al.[7] proposed ReachLocations method, based on the swapping of trajectory points, realized privacy protection of sensitive points. First, the method finds the neighbor set for each trajectory point successively. Then, for each point \( p \), if the degree of \( p \) is not less than \( K \), the position of \( p \) will be swapped with the position of any neighbor.

We propose IPKN algorithm (Interchanging of Positions on \( K \)-core subnet of trajectory points relation Net). It is inspired by the algorithm based on swapping positions of trajectory points [7] and the algorithm for \( K \)-anonymity of trajectory segments [34]. Fig.1 shows the location of
IPKN in the classification of privacy protection methods for
\( \text{TD} \).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{The location of the IPKN method in the classification of trajectories privacy protection methods.}
\end{figure}

\section*{IV. TRAJECTORY PRIVACY PROTECTION ALGORITHM (IPKN)}

Intuitively, the higher the individual density is, the easier it is to hide the individual. On the contrary, if the individual density is too low, it is difficult to hide. The hiding ability of the individual is different when passing through the spatial-temporal regions with different individual density. That is, different points of a trajectory have different hiding abilities. Based on this fact, we propose the IPKN method for privacy protection of \( \text{TD} \).

Our method first defines an interchangeable distance relation between trajectory points, which satisfies the desensitization and effectiveness requirement. Then, our method connects the trajectory points that conform to this relation, and establishes the trajectory points relation network \( \text{TN} \) and its \( K \)-core subnet \( \text{KTN} \). Finally, \( K \)-anonymity of trajectory points can be achieved by interchanging spatial-temporal positions between neighbor nodes on \( \text{KTN} \). Fig.2 describes the steps of the method: \( a \). original trajectories, \( b \). trajectory points set, \( c \). trajectory points relation network \( \text{TN} \), \( d \). \( \text{TN} \)'s \( K \)-core subnet \( \text{KTN} \), \( e \). interchange positions of the trajectory points, \( f \). the reconstructed trajectories. In Fig.2, the spatial dimension is symbolized by only one dimension in the horizontal coordinate, \( K=3 \).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.png}
\caption{The method flow}
\end{figure}

\subsection*{A. INTERCHANGEABLE DISTANCE RELATION BETWEEN TRAJECTORY POINTS}

The determination of the interchangeable distance relation between trajectory points is the basis of trajectory privacy protection based on interchanging positions. \( \text{TD}^* \) must maintain a certain degree of availability, therefore, the position interchanging of trajectory points should be limited to a certain threshold to restrain the distortion of \( \text{TD}^* \). In order to desensitize the sensitive trajectory points, it is necessary to ensure a certain degree of discrimination between \( p \) and \( p^* \). Therefore, we propose the concept of a Shell-similar Interchange Region (SIR) which describes the interchangeable distance relation between the trajectory points.

1. Spatial-temporal distance relation between trajectory points

In the spatial-temporal region where \( \text{TD} \) is formed, each trajectory point has a unique identity, time, and space attribute. It is expressed as \( \langle \text{TD}, t, s \rangle \). The distance relation between \( p_i \) and \( p_j \) is reflected in spatial dimension \( s_{ij} \) and temporal dimension \( t_{ij} \). Therefore, a particular spatial-temporal region around \( p_i \) can be taken as the range of \( p_i \) that corresponds to particular distance relationship with \( p_i \). We define the threshold of the spatial dimension as \( S_a \) and the threshold of the temporal dimension as \( T_s \). Based on these two dimensions, a distance relationship between \( p_i \) and \( p_j \) can be established as formula (1):
When \( \mathbf{p} \) is used as a reference point, the position of \( \mathbf{p}_i \) conforms to the distance relationship of inequality (1) is limited to an elliptical area centered on \( \mathbf{p} \). If the spatial distance \( s_{ij} \) between the trajectory points is measured in Euclidean distance, after substituting \( s_{ij} = ( (x_i-x_j)^2 + (y_i-y_j)^2 )^{1/2} \), this distance relationship appears as an ellipsoid centered on \( \mathbf{p} \). We use \( \text{RM}_1 \) to represent this distance region model.

According to specific application requirements, we can choose other spatial-temporal relation as the distance relationship between the trajectory points. Such as:

\[
T_s x_i + T_r t_i \leq S_s T_r
\]

In this distance relationship, the position of \( \mathbf{p}_i \) is limited to the diamond-shaped area centered on \( \mathbf{p} \). If the spatial distance \( s_{ij} \) between the trajectory points is defined as Manhattan distance, \( s = |x| + |y| \), then in the three-dimensional space formed by \( t, x, y \), the distance relationship between trajectory points is expressed as a \( \mathbf{p}_i \)-centered octahedron. We use \( \text{RM}_2 \) to represent this distance region model.

Or else, we can strip the relationship between the two dimensions, and simply defined it as following:

\[
s_{ij} \leq T_s - t_i \leq T_r
\]

At this time, \( \mathbf{p}_i \) is limited to a rectangular area centered on \( \mathbf{p} \). When the Euclidean distance \( s_{ij} = ( (x_i-x_j)^2 + (y_i-y_j)^2 )^{1/2} \) is used as the spatial distance between the trajectory points, the distance relationship is expressed as a cylinder. We use \( \text{RM}_3 \) to represent this distance region model.

2. Shell-similar interchangeable region (SIR)

There are two types of subjects that should be considered in the privacy protection of \( \text{TD} \): one is the body that generates \( \text{TD} \), and the other is the user that mines \( \text{TD}^* \). In \( \text{TD} \), the trajectory points may represent the privacy information of the body, that is, the spatial-temporal position of the point exposes some sensitive information of the body. For example, \((\text{TID}, 8:00\text{am}, \text{AIDS test center})\) indicates that the \( \text{TID} \) may be related to AIDS: infected, likely or close contact with the patient. The bodies of the trajectories require \( \text{TD}^* \) not to disclose sensitive information. So there must be a minimum distance between \( \mathbf{p} \) and \( \mathbf{p}^* \), then the probability of adversary inferring sensitive information of \( \mathbf{p} \) from \( \mathbf{p}^* \) will not exceed a certain limit. This is called desensitization. We define a spatial-temporal sensitive threshold pair \( <S_s, T_r> \) and a distance region model \( \text{RM} \) to form a sensitive area (\( A_s \)). The points in \( A_s \) are not suitable for \( \mathbf{p} \) to interchange the position. Desensitization of \( \mathbf{p}^* \) can only be achieved when separated from the sensitive area of \( \mathbf{p} \). Fig.3 shows the effect when \( \text{RM}_1 \) is selected as the distance region model for the sensitive region.

To ensure the availability of \( \text{TD}^* \), we need to limit the distortion of points to a certain extent. So we can define an effective area (\( A_e \)), the same as \( A_s \) for a sensitive area. The distortion of \( \mathbf{p} \) will not affect the availability of \( \text{TD}^* \) unless \( \mathbf{p} \) was interchanged with the trajectory points outside \( A_e \).

We define a spatial-temporal effective threshold pair \( <S_e, T_e> \) and select \( \text{RM}_2 \) as the distance region model of the effective region. Fig.3 shows the effect of the \( A_e \) for \( \mathbf{p} \).

In order to satisfy the desensitization requirements, the transition target point of \( \mathbf{p} \) is outside \( A_s \), in order to satisfy the data availability, the transition target point of \( \mathbf{p} \) is within \( A_s \). When \( A_s \cap A_s \neq \emptyset \), there is no transition region that satisfies both desensitization requirements and data availability. When \( A_s \cap A_s \cap A_s \) forms the interchangeable region of \( \mathbf{p} \). This region is equivalent to removing a core \( A_e \) from \( A_e \) and forms a \( A_e \cap A_e \cap A_s \) shell, so we call it a shell region. During the trajectory position transition process, \( \mathbf{p} \) can only be interchanged with other trajectory points distributed in its shell region. We call this region a Shell-similar interchangeable region (SIR) as shown in Fig.3.

**B. THE RELATIONSHIP NETWORK OF TRAJECTORY POINTS BASED ON SIR**

IPKN is the privacy protection method based on the interchange between positions of trajectory points. In order to record the interchangeable distance relationship between trajectory points, we map \( \text{TD} \) to a network (\( \text{TN} \)), where the network node corresponds to the trajectory points in \( \text{TD} \), and the edges between the nodes indicate the relationship that the two trajectory points satisfy SIR.

1. The data structure of \( \text{TN} \)

We store the trajectory points in the chained list, which is used as the storage structure of \( \text{TN} \). The network structure can usually be represented as a matrix, where the elements on \( i \)-row and \( j \)-column are 1 or 0, indicating whether or not there is an edge. However, in a sparse high-dimensional network, too many "0" in the matrix increase the invalid storage burden, and the calculation of high-dimensional vectors is a bottleneck in matrix operations. In this method, we choose the chained list as the main storage structure of \( \text{TN} \), as shown in Fig.4. \( \mathbf{p}_{\text{previous}} \) points to the previous node of the current node, \( \text{NodeNum} \) is the unique serial number assigned to each node. \( \text{TID} \) is the trajectory identifier which represent the body of the node, \((t, x, y)\) is...
the spatial-temporal position of the trajectory point, \( P_{next} \) points to the next node of the current node. \( NeighborNum \) is the degree of the node in the network, and \( P_{neighbor} \) points to the neighbor set of the node.

The neighbor set of each node is saved in the form of a binary search tree, which can reduce the time complexity of the algorithm from \( K^2 \) to \( \log(K) \). \( P_{father} \) points to the parent node of current node, \( NodeNum \) is the unique sequence number of the neighbor node, \( P_{left} \) and \( P_{right} \) point to the left and right child nodes of current node, and \( P_{neighbor} \) points to the node of current neighbor in the main storage chained list.

The time window is shown in Fig.5. In \( TD \), when we search the neighbor points in \( SIR \) of \( p_i \) only the points whose sampling time \( t \) satisfies the condition \( |t_t-t|\leq T_e \) may become the neighbor points of \( p_i \). Then we need to compare the spatial-temporal distance between \( p_i \) and the candidate neighbor points, while other points must not satisfy \( SIR \) with \( p_i \). We define the time window of the trajectory point \( p_i \) as \([t_i, t_i+T_e]\) for temporal distance detection. The correlation detection for \( p_x \) and \( p_i \) \((t_s<t_i)\) has been completed when detecting the neighbor points of \( p_x \).

When the operation pointer (OP) points to \( p_i \), our algorithm calculate the \( SIR(p_i,p_j) \), in which \( p_i \) is the node within the time window \([t_i, t_i+T_e]\) and does not belong to the same trajectory with \( p_i \). If \( SIR(p_i,p_j) \) is true, then connect \( p_i \) and \( p_j \). That is, add each other to the neighbor node sets of the two trajectory points. After all nodes in the time window complete detection with \( p_i \), the neighbor set of \( p_i \) is obtained. Then move \( OP \) to the next point and repeat the process. After all points on the chained list complete the searching for neighbors, we get a \( SIR \) relational network about trajectory points in \( TD \). In future statements, node in \( TN \) is equivalent to point in \( TD \). The construction process of \( TN \) is described as Algorithm 1.

The method realizes the privacy protection by interchanging the positions of points. For example, in Fig.6, there are five points from different trajectories in \( TD \): \( p_1=[TID_1,t_1,x_1,y_1] \), \( p_2=[TID_2,t_2,x_2,y_2] \), \( p_3=[TID_3,t_3,x_3,y_3] \), \( p_4=[TID_4,t_4,x_4,y_4] \), and \( p_5=[TID_5,t_5,x_5,y_5] \). In \( TD^* \), \( p_1 \) and \( p_2 \) are interchanged to each other’s spatial-temporal positions. Then \( p_1 \) and \( p_2 \) map to \( p_1^*=[TID_i,t_2,x_2,y_2] \) and \( p_2^*=[TID_4,t_4,x_4,y_4] \) respectively. It can be seen from the example above that the positions interchange of trajectory points can also be regarded as the interchange of \( TIDs \).

\( TD^* \) is obtained by interchanging the positions of the trajectory points in \( TD \). Each trajectory point in \( TD^* \) corresponds to a certain position in \( TD \). The interchanging of positions between trajectory points does not change the spatial-temporal distribution of trajectory points. The set of
trajectory points in TD* is a subset of the trajectory points in TD without considering TID.

Next, we discuss the privacy level of 1/3 of the interchanged nodes being restored is no more than randomly interchanged in a TD*-anonymity, then the relation network of trajectory points on neighbors. In fact, if all the points in TD* satisfy t, then adversary can infer the home position of interchanged to the position (t, x, y), which has only one neighbor, so adversary can infer the home position of TD immediately, even if the position (t, x, y) have four neighbors. In fact, if all the points in TD* satisfy K-anonymity, then the relation network of trajectory points on TD* is the K-core subnet of TN.

Theorem 1: If and only if neighbor nodes are randomly interchanged in a K-core network, the probability of the interchanged nodes being restored is no more than 1/K.

Proof of sufficiency: Suppose p1 and p2 are two neighbor nodes in the K-core network G, the degree of p1 is g1 (g1< K), and the degree of p2 is g2 (g2< K). Interchange the positions of p1 and p2 to form new positions p1* and p2*. So we can get that g1* = g2 ≤ K, g2* = g1 ≤ K. p1* may be obtained by interchanging with any of the g1* nodes in the new neighborhood, and the probability of reducing p1* is 1/g1* (1/g1* ≤ 1/K). Similarly, the probability of reducing p2* is 1/g2* (1/g2* ≤ 1/K).

Proof of necessity: Suppose p1 and p2 are two neighbor nodes in network G, p1 is the only node whose degree is less than K. Then the degree of p1 is g1 (g1< K), and the degree of p2 is g2 (g2< K). Interchange the positions of p1 and p2 to form p1* and p2*, so we have g1* = g2 ≤ K, g2* = g1 ≤ K. p2* may be obtained by interchanging with any of the g2* nodes in the new neighborhood, and the probability of reducing p2* is 1/g2* (1/g2* > 1/K).

We can get the following inference from theorem 1.

Inference 1: In order to achieve K-anonymity of trajectory points in TD* by the privacy protection algorithm based on interchanging positions, the interchanges must be confined to the K-core subnet of TN.

In TN, the K-core subnet can be obtained by iteratively delete all the nodes with degrees less than K [35]. The process is shown in algorithm 2.

Fig.7 shows the process of obtaining 3-core subnet of TN. State a is the original relational network of trajectory point (TN). State b-d is the intermediate state of successive deletion of points with degrees less than 2. State e is the 2-core subnet of TN. State f is the 3-core subnet of TN. The TD* that satisfies 3-anonymity of trajectory points can be obtained by interchanging the spatial-temporal positions of neighbor trajectory points on f.

**Algorithm 2.** KTN(TNoutput)\(//create the K-core subnet of TNoutput\)

**Input:** TNoutput (Relationship network between the points in TD), K (core parameter)

**Output:** KTNoutput (K-core subnet of NToutput)

Start
flag=1
Do while flag=1
  For i=1 to |TNoutput|
    If pi,NeighborNum< K then
      Delete pi from its neighbors
      Delete pi
    endif
  Next i
  If there is any node whose degree is less than K, then flag=1,
  Else flag=0
  Loop
End

**D. INTERCHANGE ALGORITHM BASED ON MINIMUM DEGREE PRIORITY AND ITS IMPROVED ALGORITHM IPKN**

In KTN, by interchanging positions between neighbor nodes, we can obtain the TD* that satisfies the K-anonymity of trajectory points. The optimal choice of node pairs for interchanging in KTN is a problem with high complexity. In a 2-core network shown in Fig.8, interchanging with the first plan will left a single node that cannot be interchanged. With the second plan, we can get the TD* that satisfies 2-anonymity of trajectory points.
We design an interchange algorithm based on minimum degree priority, in order to maximize the probability of interchange all nodes in a network.

a. In order to increase the interchangeability of nodes with small degree, we establish a degree-based binary search tree of the nodes and mark all nodes as active.

b. Then the active node \( p \) with the smallest degree in KTN is selected, and interchanged the positions of \( p \) with its neighbor whose degree is the smallest, and then mark them as interchanged.

c. After the interchange, we delete all the edges of the two interchanged nodes except the edges with their co-neighbors, with the purpose of increasing the interchangeable opportunities for their co-neighbors.

d. Then we revise the degree of the two interchanged nodes and their original neighbor nodes, and update the degree-based binary search tree. If an active minimum degree node has no neighbor node, it is marked as frozen.

e. Repeat the process b to d until there are no active nodes.

f. At last, we insert every frozen node into its neighbor trajectory randomly to ensure that the \( K \)-anonymity of trajectory points in TD* is not destroyed.

This algorithm processes low degree nodes preferentially in order to implement pairing and interchanging all nodes as much as possible. However, the frequent searching and modification in the degree-based binary reach tree reduces the operational efficiency. In order to improve the efficiency, we simplified the algorithm. In step b, we try to interchange the position of every nodes with its random neighbor. This reduces the comparison and sorting of trajectory points greatly and improves the speed of the algorithm.

Suppose the number of nodes suppressed when construct KTN from TN is \( m_1 \), the number of remaining frozen nodes after location swapping in KTN is \( m_2 \). Then the ratio \( m_2/m_1 \) reflects the success rate of the interchange. Experiments showed that: in the TD* generated by the original interchange algorithm, this ratio is usually less than 0.06, and when we use the simplified algorithm, the ratio is no more than 0.1. But the time complexity reduced from \( K^2 \) to \( \log(K) \), so this trade-off is reasonable. The simplified algorithm 3 is described as following:

```
Input: The \( K \)-core subnet of TN output (KTN_output)
Output: TD* that satisfies \( K \)-anonymity of trajectory points
Start
For i=1 to |KTN_output|
    If \( p_i \) is active and \( \text{NeighborNum}=0 \) then
        select a neighbor \( p_j \) randomly
        interchange the triples of \( p_i \) and \( p_j \)
        mark \( p_i \) and \( p_j \) as interchanged
        delete the link between \( p_i \) and \( p_j \)
        delete all the links with the non-co-neighbors (\( p_x \)) of \( p_i \) and \( p_j \)
        \( p_i \).TID=\( p_j \).TID
        \( p_j \), mark as frozen
    Endif
Next i
Output TD*=KTN_output
End
```

Algorithm 3. IPKN (KTN_output//Interchanging of Positions on \( K \)-core subnet of TN)

When applies IPKN in reality, we take TD, SIR and \( K \) as inputs. By sequentially processing the three algorithms TN, KTN, and IPKN, we obtain TD* that satisfies \( K \)-anonymity of trajectory points. The desensitization requirements (\( A_s, K \)) of the trajectory bodies and effectiveness requirements (\( A_s \)) of the trajectory user are fixed respectively in TD*. The trajectory points that do not meet these requirements will be suppressed, so the number of the suppressed trajectory points is part of the result, which is denoted by Trash. After interchanging the positions of trajectory points, the time series of each trajectory should be rearranged to form the final TD* for publishing.

E. ANALYSIS OF ALGORITHM

1. Privacy analysis of TD*

Suppose that the adversary knowledge includes two parts: a. TD* and a set \( P \) with \( m \) trajectory points of the target trajectory \( T \); b. the trajectory privacy protection method and its parameters. The purpose of adversary is the position information of \( p_m \) in \( T \). The attack strategies depend on the privacy protection methods.

The attack strategy for positions of the TD* satisfied \( K \)-anonymity of trajectories can be expressed as follows:

a. Adversary firstly query the \( R_{\text{max}} \) neighborhood area of \( p_1, p_2, ..., p_l, ..., p_m \) in TD*, where \( R_{\text{max}} \) is the cluster radius. Suppose the trajectory set contained in the neighborhood of \( p_1 \) is TS1, then \( m \) trajectory sets \{TS1, TS2, ..., TS_m\} can be found according to the \( m \) known trajectory points.

b. The intersection of the \( m \) trajectory sets includes at least \( K \) trajectories, which necessarily contain the transformed trajectory \( T_i \), and the probability of determining \( T_i \) is no more than \( 1/K \).

c. The trajectory cluster determined by this intersection is the target desired by the adversary, and the adversary can
further obtain the relevant information of the trajectory cluster, that is, the macro trajectory of \( T_i \).

To the TD* generated by IPKN, the attack strategy can be expressed as follows:

a. Adversary firstly find the \( m \) trajectory sets corresponding to \( m \) known trajectory points as to the TD* satisfies \( K \)-anonymity of trajectories;

b. In TD* generated by IPKN don’t satisfies \( K \)-anonymity of trajectories. In an extreme case, the intersection of the \( m \) trajectory sets contains only one trajectory, which is the transformed trajectory \( T_i^* \) from \( T_i \);

c. \( p^* \) is a point in \( T_i^* \), the home position of \( p^* \) is included in its SIR, which contains at least \( K \) points. So what the adversary gets is just a region with \( T_i^* \) as the axis, such as the result obtained from the TD* satisfies \( K \)-anonymity of trajectories.

Fig.9 shows the privacy protection effect of the 6-anonymity of trajectory points. Assuming that the adversary grasps \( p_{i1} \) and \( p_{i2} \) of the trajectory \( T_i \), then at least 6 interchanged positions can be identified in the SIR of \( p_{i1} \), so at least 6 correlated trajectories can be associated. We use TS\(_i\) to represent the set of trajectories correlated with \( p_{i1} \). Similarly, we can get the TS\(_i\) associated with \( p_{i2} \). \( T_i^* \) must be in the intersection of these two sets, so the number of intersection trajectories is not less than one.

In extreme case, the intersection contains only one trajectory \( T^* \), then \( T^* \) is the transition trajectory \( T_i^* \) of \( T_i \). The positions of the trajectory \( T_i \) on the arbitrary time \( t_j \) must be within the SIR of \( p_{i0}^* \), and its neighborhood contains at least 6 trajectory points, so the probability to determine the home position of \( p_{i3} \) is no more than 1/6.

![FIGURE 9. TD* satisfies 6-anonymity of trajectory points of IPKN](image)

2. The efficiency execution of IPKN

Suppose the number of trajectories in TD is \( N \), and the average length of trajectories is \( n \). Elapsed time of IPKN is consumed in three sub-processes:

a. Constructing TN: The algorithm compares the distance between each point and its neighbors within the time window, which costs \( O(\Delta T \cdot n \cdot N^2) \);

b. Construction of KTN: In TN, the algorithm recursively traverses the trajectory points that do not meet \( K \)-core requirement, and deletes them from its neighbors, which cost \( O(n \cdot N \cdot Trash \cdot \log(K)) \), where Trash refers to the suppression ratio of trajectory points;

c. During the interchanging process, the algorithm needs to traverse and interchange the trajectory positions of the neighbor nodes, which cost \( O(n \cdot N) \).

Actually, the process of constructing NT is only executed once. The main elapsed time is caused by the process of repeatedly detecting and deleting positions that do not meet \( K \)-core requirement. The time complexity of the method is \( O(n \cdot N \cdot Trash \cdot \log(K)) \), where \( n \cdot N \) is the number of trajectory points in TD and \( K \) is the anonymity threshold.

V. EXPERIMENTAL VERIFICATION

The strategy of the method determines the privacy level of TD*, which has been discussed in section IV part E. This section verifies the efficiency execution and the information loss of TD* generated by IPKN.

We select three typical trajectory privacy protection methods as comparing algorithms, among which NWA[4] is the typical \( K \)-anonymity of trajectories algorithm based on clustering and generalizing of trajectories. DLBG[34] is a \( K \)-anonymity of trajectory segments algorithm based on the method proposed by Monreale A et al.[6], and ReachLocations[7] is a privacy protection algorithm for trajectory positions based on position swapping. We compare the three algorithms with our method on both real-life and synthetic datasets respectively to verify IPKN on efficiency execution and information loss.

A. THE DATA FOR EXPERIMENT

In order to verify the effectiveness of the algorithm on different data, we select three TDs, including a real-life TD and two synthetic TDs, the synthetic TDs contains a sparse small-scale dataset and a dense large-scale TD.

The first TD is a real-world dataset, which come from 50 trucks delivering concrete to several construction places around Athens metropolitan area in Greece for 33 distinct days. It consists of 112203 points, which constitute 273 trajectories. It is called Trucks1. The distribution of Trucks is shown in Fig.10.1.

The second one is a synthetic small-scale TD, which comes from KAIST. It consists of 135055 points, which constitute 92 trajectories. It is referred as KAIST2. We can see that the trajectories on KAIST are longer. It is shown in Fig.10.2.

The third one is a synthetic large-scale TD, generated by Brinkhoff’s data generator. It consists of 4780954 positions, which constitutes 100000 trajectories and is referred as Oldenburg3. It is shown in Fig.10.3.

The TDs have different sampling characteristics. The sampling frequency and continuous time are the most important features, which directly affect the compatibility

---

1 http://chorochronos.datastorries.org/sites/default/files/uploaded_dataset/trucks.zip
3 https://www.kdd.isti.cnr.it/NWA/Oldenburg_100k_traj.zip
between different sampling data. In this paper, the data is normalized: when the sampling frequency is constant, the sampling time is unified to 0:00-24:00, and the trajectories in different dates are regarded as independent trajectories. For the trajectories that spans 24:00, this may cause some trouble, but we ignore it.

In table 1, we reported the characteristics of the three TDs: |TD| is the number of trajectories, Area is the spatial span covered by TD, and Tspan is the time span. AS is the average speed. Density is the spatial-temporal density of trajectory points, which is the quotient of the trajectory point’s number and spatial-temporal span.

### B. THE METHOD OF EXPERIMENT

Information loss in TD* includes data suppression and data distortion. Data suppression refers to the removal of trajectories and trajectory points, while data distortion refers to the offset of converted trajectory position from the home position.

Under the premise of setting the same parameter \( K \) and data suppression ratio, three typical \( \text{TD} \) privacy protection algorithms are compared with IPKN. We test the four algorithms on three \( \text{TDs} \), the advantages of IPKN in terms of elapsed time and the information loss generated by data distortion are verified.

1. **Basic theory of experiment**

In NWA algorithm, if the points at the beginning or end of the trajectories are not in the projection range, they will be suppressed when mapping the trajectories to normal time period. The trajectory suppression occurs when merging equivalence class and clustering trajectories. These two types of suppression above constitute the total data suppression of NWA algorithm. The data distortion is caused by reconstruction of trajectories in the trajectory clusters, and the data distortion is positively correlated with the cluster radius \( R_{\text{max}} \).

In DLBG algorithm, suppressed data includes noise points generated when RR is aggregated, and noise regions generated when GR is aggregated. The data distortion depends on the radius \( R \) of GR.

In ReachLocation algorithm, data suppression results from those trajectory points that cannot be swapped, which have no \( K \) neighbor trajectory points. Data distortion depends on the maximum neighborhood radius \( r \) of the trajectory points.

In IPKN, data suppression comes from the suppression of points during the construction of KTN from TN. The data distortion is caused by the interchange of trajectory positions and positively correlated with \( S_e \).

In case of the same data suppression ratio (Trash), TD* generated by these four algorithms will generate some information loss, which is caused by data distortion during the trajectories transition. We examine the availability of the four algorithms by comparing the information loss by data distortion. The loss of information due to distortion of TD* is represented by the following formula:

\[
\text{Infloss} = \sum_{i=0}^{[\text{TD}^*]} \text{Dis}(p_i, p_i^*)
\]

Where \( |\text{TD}^*| \) represents the number of trajectory points in \( \text{TD}^* \). \( p_i \) is the transition position of \( p_i \). \( \text{Dis} \) represents the distance between two spatial-temporal positions, which is defined as following:

\[
\text{Dis} = \left( \Delta x^2 + \Delta y^2 + \Delta t^2 v^2 \right)^{1/2}
\]

Where \( v \) is the average speed of \( \text{TD} \).
2. Data preprocessing

DLBG, ReachLocations and IPKN do not need to preprocess the experiment data. The data preprocessing for NWA is as follows:

In Trucks, \( p_t (t_i, x_i, y_i) \) is synchronized to a time point \( t'_i \), which is the nearest time point of 29 second or 59 second that greater than the current time. The spatial coordinates \( (x'_i, y'_i) \) of the synchronization position are obtained according to the spatial-temporal position of current point \( p_i \) and the next point \( p_{i+1}(t_{i+1}, x_{i+1}, y_{i+1}) \) on the same trajectory:

\[
\begin{align*}
    t'_i &= int(t_i/30) \times 30 + 29 \\
    x'_i &= x_i + (x_{i+1} - x_i) \frac{(t_{i+1} - t_i)}{(t_{i+1} - t_i)} \\
    y'_i &= y_i + (y_{i+1} - y_i) \frac{(t_{i+1} - t_i)}{(t_{i+1} - t_i)} 
\end{align*}
\]  

In the process of merging equivalence classes, we choose an integer multiple of \( 415 \times 30 \) seconds as the time endpoint of the intercepted trajectory. The points at both ends of the trajectory that are not included in this range will be suppressed. Among the equivalence classes, there are three equivalence classes with only one trajectory, so they are also suppressed. The remaining 270 trajectories are distributed in 12 equivalence classes. The suppressed ratio of the trajectory points during this reprocessing reached 32.41%.

<table>
<thead>
<tr>
<th>Equivalence class order</th>
<th>Start time</th>
<th>End time</th>
<th>Trajectories number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12450</td>
<td>37350</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>12450</td>
<td>49800</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>12450</td>
<td>62250</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>12450</td>
<td>74700</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>24900</td>
<td>37350</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>24900</td>
<td>49800</td>
<td>37</td>
</tr>
<tr>
<td>7</td>
<td>24900</td>
<td>62250</td>
<td>87</td>
</tr>
<tr>
<td>8</td>
<td>24900</td>
<td>74700</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>37350</td>
<td>47350</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>37350</td>
<td>49800</td>
<td>33</td>
</tr>
<tr>
<td>11</td>
<td>37350</td>
<td>62250</td>
<td>33</td>
</tr>
<tr>
<td>12</td>
<td>37350</td>
<td>74700</td>
<td>9</td>
</tr>
</tbody>
</table>

In KAIST, the trajectories are continuously synchronized, and we select \( \pi = 500 \) to merge equivalence classes. The four equivalence classes in Table 3 are obtained, and the amount of suppressed trajectory points is 23555, accounting for 17.44% of the total trajectory points.

<table>
<thead>
<tr>
<th>Class order</th>
<th>Trajectory end time</th>
<th>Trajectory cut length</th>
<th>Number of trajectories</th>
<th>Number of trajectory points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500-999</td>
<td>500</td>
<td>17</td>
<td>8500</td>
</tr>
<tr>
<td>2</td>
<td>1000-1499</td>
<td>1000</td>
<td>34</td>
<td>34000</td>
</tr>
<tr>
<td>3</td>
<td>1500-1999</td>
<td>1500</td>
<td>26</td>
<td>39000</td>
</tr>
<tr>
<td>4</td>
<td>( \geq 2000 )</td>
<td>2000</td>
<td>15</td>
<td>30000</td>
</tr>
</tbody>
</table>

In Oldenburg, we also refer to the classification criteria in the NWA algorithm, and select the parameter \( \pi = 5 \) to merge the equivalence classes. We obtain 435 equivalence classes, the amount of suppressed trajectory points is 491928, accounting for 10.29% of the total number of trajectory points.

<table>
<thead>
<tr>
<th>Class order</th>
<th>Trajectory end time</th>
<th>Trajectory cut length</th>
<th>Number of trajectories</th>
<th>Number of trajectory points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>415</td>
<td>12</td>
<td>36365</td>
<td>32.41%</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>4</td>
<td>23555</td>
<td>17.44%</td>
</tr>
<tr>
<td>3</td>
<td>435</td>
<td>491928</td>
<td>10.29%</td>
<td></td>
</tr>
</tbody>
</table>

3. Parameter setting

In NWA algorithm, we limit the TD suppression ratio caused by trajectory clustering no more than 10%, then the total data suppression ratio in Trucks, KAIST and Oldenburg are 42.41%, 27.44% and 20.29% respectively. We take it as the upper limit of the data suppression ratio in the other three algorithms.

The initial values of the parameters corresponding to the four algorithms are shown in Table 5. In NWA, \( R_{max0} \) is the initial value of the trajectory cluster radius; \( \Delta R_{max} \) is the increase of \( R_{max} \) each time. In the original algorithm, this parameter is designed as a ratio, but the ratio leads to the growing of \( R_{max} \) too fast; in order to take NWA as a baseline to compare with IPKN, the uncertainty threshold \( \delta \) is fixed to 0.

In DLBLG, the aggregate parameter \( r \) of RR was set according to the \( Density \) in different data. \( R \) is the initial value of aggregation parameter for GR, \( \Delta R \) is the increase of \( R \) \((T = R/t)\) is the neighborhood radius in time dimension.

In ReachLocations, \( S \) is the neighborhood radius in space dimension, \( \Delta S \) is the increase of \( S \) \((T = S/s)\) is the neighborhood radius in time dimension.

In IPKN, \( S_{ref} \) is the initial value of the effective threshold in the SIR; \( \Delta S_e \) is the increase of \( S_e \) each time; \( S_e \) is the initial value of the effective threshold in the SIR.
is the sensitive threshold, \( v \) is the average speed in TD, \( S_e/S_k \) is denoted by \( \alpha \), with values of 0, 0.25, 0.5 and 0.75 respectively. Practically, there is no proportional relationship between \( S_e \) and \( S_k \), so the assumption of such proportion is only used for analysis in the experiment. TABLE 5. Experimental parameter settings in the four Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWA</td>
<td>( r_{max} )=1000;  ( \Delta R_{max} )=100; ( \delta )=0</td>
</tr>
<tr>
<td>Trucks</td>
<td>( r)=500;  ( R)=500; ( \Delta R )=200</td>
</tr>
<tr>
<td>KAIST</td>
<td>( r)=100, ( R)=200; ( \Delta R )=100</td>
</tr>
<tr>
<td>Oldenburg</td>
<td>( r)=50, ( R)=100; ( \Delta R )=50</td>
</tr>
<tr>
<td>ReachLocations</td>
<td>( S_{max}=50; \Delta S_{max}=10; \alpha=S_e/\alpha; T_e=S_e/v; T_k=S_k/v )</td>
</tr>
<tr>
<td>IPKN</td>
<td>( S_{max}=50; \Delta S_{max}=10; S_e=S_k/\alpha; T_e=S_e/v; T_k=S_k/v )</td>
</tr>
</tbody>
</table>

4. Algorithm design for experiments

In the experiment, we need to compare the data distortion in TD* generated by different algorithms under the same Trash. Because the data distortion in the IPKN algorithm is positively related to \( S_e \), we designed another experimental algorithm. Take TD, K, Trash, \( S_{\alpha}, \Delta S_e, \alpha \) as input, get TD* and its total amount of data distortion with the same Trash. The pseudo code of the overall algorithm is as follows:

\[
\begin{align*}
\text{Input: } &TD, S_{\alpha}, \Delta S_e, \text{Trash}, K \\
\text{Output: } &TD^*, \text{Distortion} \\
\text{Start} & \\
\text{Sort} &\text{output}=\text{Sort}(TD)// \text{Sort trajectory points in ascending order of sampling time} \\
\text{TN}_{\text{output}} &= \text{TN}(\text{Sort} \text{output}, S_e=S_{\alpha}) \\
\text{Do while the number of trajectory points with degree less than } &K \text{ is greater than Trash in } \text{TN}_{\text{output}} \\
\text{TN}_{\text{output}} &= \text{TN}(\text{Sort} \text{ output}, S_e=S_e+\Delta S_e) \\
\text{Loop} & \\
\text{KTN}_{\text{output}} &= \text{KTN}(\text{TN}_{\text{output}}, K) \\
\text{Do while the number of trajectory points with degree less than } &K \text{ is greater than Trash in } \text{KTN}_{\text{output}} \\
\text{KTN}_{\text{output}} &= \text{KTN}(\text{Sort} \text{ output}, S_e=S_e+\Delta S_e) \\
\text{KTN}_{\text{output}} &= \text{KTN}(\text{TN}_{\text{output}}, K) \\
\text{Loop} & \\
\text{TD^*}=\text{IPKN}(\text{LTN}_{\text{output}}) & \\
\text{Do while } [TD^*-TD^*>\text{Trash}]& \\
\text{TN}_{\text{output}} &= \text{TN}(\text{Sort} \text{ output}, S_e=S_e+\Delta S_e) \\
\text{KTN}_{\text{output}} &= \text{KTN}(\text{TN}_{\text{output}}, K) \\
\text{TD^*}=\text{IPKN}(\text{KTN}_{\text{output}}) & \\
\text{Loop} & \\
\text{Find the distortion distance between all the corresponding trajectory points in } &TD \text{ and } TD^* \text{ and add them to Distortion} \\
\text{Output } &TD^*, \text{Distortion} \\
\text{End} &
\end{align*}
\]

Algorithm 4. Experiment (TD, K, \( S_{\alpha}, \Delta S_e, \text{Trash} \))

In order to ensure that the obtained result has the same Trash with IPKN, the experiment about DLBG and ReachLocations also apply the experimental algorithms similar to Algorithm 4.

C. RESULTS AND ANALYSIS OF EXPERIMENT

1. Data distortion

Among the three TDs, \( R_{max} \) in NWA, R in DLBG, S in ReachLocations, and \( S_e \) in IPKN can all be regarded as representational values of data distortion. Their differences are obvious, when we select different clustering parameter K. It can be seen from Fig. 11 that for IPKN algorithm, when \( \alpha \leq 0.5 \), a significantly smaller \( S_e \) is shown in all three data sets. \( S_e \) increases with \( \alpha \), and when \( \alpha \) reaches 0.75, \( S_e \) even closes to \( R_{max} \) in NWA.
trajectories and only considers the privacy protection of trajectory points, so the cluster radius and information loss are the minimum. Based on ReachLocations, IPKN introduces the theorem 1 and effectivity-sensitivity SIR, and realizes strict $K$-anonymity of trajectory points, which not only improves data privacy level when compared with ReachLocations, but also decreases the information loss when compared with NWA.

In IPKN, with the increase of $S_s/S_e$, the information loss increases and the availability decreases. Fig.13 shows the information loss with different $\alpha$ under different $K$.

2. Efficiency of the algorithm

In the data distortion experiment, we select different TDs, algorithms and parameters. We also measure the elapsed time of those experiments. The efficiency of the algorithms in the three TDs are shown in Fig.14.
It can be seen from Fig.14 that the algorithm presents different elapsed times in different TD.

In KAIST, the trajectories are the fewest. Because the elapsed time of NWA depends on the number of trajectories, so the elapsed time of NWA is the fastest. The elapsed time of IPKN is significantly higher than NWA.

In Trucks, the number of trajectories is increased, while the number of trajectory points was almost the same as KAIST. The elapsed time of IPKN is almost unchanged, while that of DLBLG is longer slightly, and ReachLocations remains the same elapsed time as IPKN.

In Oldenburg, the number of trajectories and trajectory positions increases significantly, but the length of the trajectories decreases significantly, so the ratio between the number of trajectories and the number of trajectory points increases. Such data feature significantly extends the elapsed time of NWA, while the elapsed time of IPKN increases relatively slowly.

It can be seen from Fig.15 that the elapsed time of IPKN algorithm mainly depends on the number of trajectory points in TD. In the same data, the elapsed time is positively correlated with log(K) and α, but with the increase of K and α, the elapsed time changes slightly. Due to the unevenness of the trajectory distribution, the statistical data have some deviation from the theoretical value, but the overall trend is consistent.

VI. CONCLUSION

We propose an interchange-based trajectory privacy protection method. First, we define an interchangeable distance relation between trajectory points, and then propose a Shell-similar Interchangeable Region (SIR). By taking SIR as the metric of the relationship between trajectory points, we map trajectories to the relationship network of trajectory points. The trajectory database that satisfies K-anonymity of trajectory points can be achieved by interchanging the position between neighbor nodes on the K-core subnet of the relationship network, which is the main innovation of this paper.

The Shell-similar interchangeable region strikes a balance between privacy and usability of the published trajectories. The strategy based on position interchange of trajectory points preserves the home position of the publishing trajectories. The K-anonymity strategy based on trajectory points reduces the suppression and distortion of published trajectories.

However, these advantages are obtained on the premise of abandoning the K-anonymity of trajectories. The K-anonymity of trajectory points reduces the restriction on privacy requirements. Therefore, IPKN is suitable for the trajectory database that do not need to hide the trajectory subject identity but do need to hide the sensitive positions of the trajectory points.

In this paper, we set the same sensitive threshold, effective threshold and anonymity threshold for all the trajectory points. For a specific purpose of data analysis, the unified effective threshold is reasonable. But for different individuals, each trajectory may have different privacy requirements, corresponding to different sensitive thresholds and anonymity thresholds. Even in the same trajectory, particular trajectory points may also have different privacy protection needs. Our future research...
direction is how to maximize the availability of published trajectories under the premise of personalized privacy protection.

REFERENCES


[24] C. C. M. W. J. Y. J. Zhang, "Privacy preserving algorithm based on trajectory location and shape


