A

Traffic Information Publication with Privacy Preservation

Sashi Gurung, Missouri University of Science and Technology
Dan Lin, Missouri University of Science and Technology
Wei Jiang, Missouri University of Science and Technology
Ali Hurson, Missouri University of Science and Technology
Rui Zhang, University of Melbourne

We are experiencing the expanding use of location-based services such as AT&T TeleNav GPS Navigator and Intel's Thing Finder. Existing location-based services have collected a large amount of location data, which have great potential for statistical usage in applications like traffic flow analysis, infrastructure planning and advertisement dissemination. The key challenge is how to wisely use the data without violating each user's location privacy concerns. In this paper, we first identify a new privacy problem, namely *inference-route* problem, and then present our anonymization algorithms for privacy-preserving trajectory publishing. The experimental results have demonstrated that our approach outperforms the latest related work in terms of both efficiency and effectiveness.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Spatial databases and GIS

Additional Key Words and Phrases: Location privacy, Trajectory anonymization, Road-network constraint

ACM Reference Format:
Gurung, S., Lin, D., Jiang, W., Hurson, A. and Zhang, R., Traffic Information Publication with Privacy Preservation. ACM Trans. Embedd. Comput. Syst. V, N, Article A (January YYYY), 23 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

The extensive use of location-based services, such as AT&T TeleNav GPS Navigator, Sprint's Family Locator, and Intel's Thing Finder, have collected a large amount of location data. If information like vehicle IDs and moving directions on roads can be published, people in many fields will benefit from it. For example, with respect to the public sector, traffic flow information can be extracted from published IDs and moving directions. Such information will play an important role in infrastructure construction and traffic light control. With respect to the business domain, traffic information can help decide the location of company branches, and also advertisements can be customized and disseminated at the most advantageous locations. With respect to our daily lives, traffic information is certainly useful for detecting and predicting traffic jam, and calculating better routes for travelers [Xue et al. 2013a; Xue et al. 2013b] or in an emergency (e.g., for ambulances). However, in the meantime, location privacy concerns [Mokbel 2007; Tanner 2008; Huo et al. 2013] may hinder the development of such attractive usage of traffic information. It is well known that using a pseudonym is not sufficient to prevent the linkage of a published location to a real ID [Bettini1 et al. 2005]. The key challenge is how to wisely use the location data without violating each user's privacy concerns. This problem is termed as *privacy preserving historical location data publishing*.

Historical location data forms a sequence of locations in chronological order, termed as *trajectory*. In general, one's trajectory consists of roads he has visited. For instance, in Figure 1, user u_1's trajectory can be represented as $IABC$ and user u_4's trajectory is ABD. Many approaches

Author's addresses: S. Gurung, D. Lin, W. Jiang and A. Hurson, Computer Science Department, Missouri University of Science and Technology; R. Zhang, University of Melbourne, Australia.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

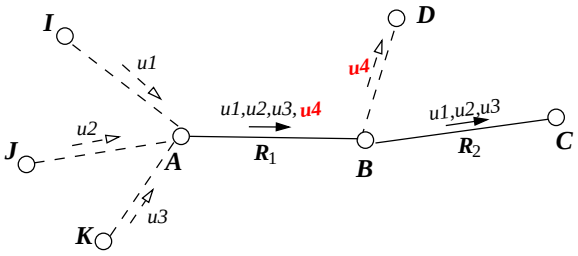


Fig. 1. An Example of Inference-Route Problem

[Wei et al. 2012] have been proposed to construct popular routes from trajectory datasets. Publishing trajectories consistent with the road network will enable the data mining algorithms to extract more precise routes patterns in comparison to representing a trajectory as a sequence of symbols [Andrienko et al. 2009]. After taking into account the privacy concerns, the goal becomes to prevent adversaries from mapping published locations to a specific individual.

One may think that a trajectory resembles a conventional sequential pattern. Hence, a naturally raised question is that if we can directly employ privacy preserving data publishing approaches [Agrawal and Srikant 2000; Atzori et al. 2008; Pei et al. 2004; Zaki 2001] developed in non-spatial-temporal databases? The answer is negative, and the main reason is that a trajectory distinguishes itself from the conventional sequential patterns due to additional constraints (e.g., road-network information) which do not exist in the traditional sequences. More specifically, elements in traditional sequences are usually independent of one another, while the relationship of elements in the trajectory sequence is fixed under a particular road-network information. Therefore, we cannot use traditional algorithms to arbitrarily remove or replace elements in the sequences because such operations will create unrealistic trajectories consisting of non-connected road segments.

There have been several recent efforts [Abul et al. 2008; Gidofalvi et al. 2007; Terrovitis and Mamoulis 2008; Nergiz et al. 2009; Andrienko et al. 2009] on anonymizing trajectories. Some work [Terrovitis and Mamoulis 2008] considers trajectories as a sequence of landmarks, e.g., stores and museums, which ignore the paths connecting these places. Others [Abul et al. 2008; Gidofalvi et al. 2007; Andrienko et al. 2009] consider trajectories as a sequence of coordinates in Euclidean space but do not fully consider the road-network constraints. Specifically, their anonymization results mainly provide movement trends (e.g., centroid of clusters of trajectories [Monreale et al. 2010]). Since the centroid of clusters could even be off road, e.g., a middle point of two parallel roads, it is hard to tell the actual roads that a group of vehicles are traveling from the anonymized results. Consequently, such anonymization results may not be as useful as real trajectories in terms of providing good insight on traffic condition analysis for individual roads, and traffic lights placement. Therefore, in our work, we ensure that the anonymization output is also trajectories on real road-network.

In the literature, there is very few works that generate actual road-network-constrained trajectories as the anonymization output. The most recent one is by Pensa et al. [Pensa et al. 2008], who anonymize trajectories based on k-anonymity [Sweeney 2002]. The notion of k-anonymity guarantees that each anonymized trajectory is a common trajectory of at least k users, and such anonymized trajectories are called frequent trajectories. However, their approach may not preserve trajectory information as much as possible. This can be demonstrated by the example given below.

In [Pensa et al. 2008], trajectories are stored and anonymized by using a prefix tree which may not be an appropriate structure to model the road-network. For instance, consider four users who leave their homes (I, J, K, A) and head for work. Let k be 3, which means a trajectory can be published if at least three users have this trajectory. Suppose that the input to their algorithm is the following four trajectories: $u_1(IABC)$, $u_2(JABC)$, $u_3(KABC)$ and $u_4(ABD)$[1], their anonymization result will

[1] u_1, u_2, u_3 and u_4 can be thought as either a trajectory ID or a person's symbolic ID.

be an empty set since the prefix tree treats trajectories with different starting points independently. Such result obviously loses too much useful information. To achieve better information utility, an alternative way is to directly take partial trajectories as input, i.e., consider only busy roads with more than k users. In this case, the input becomes $u_1(ABC)$, $u_2(ABC)$, $u_3(ABC)$ and $u_4(AB)$, and the new anonymization result is: $u'_1(ABC)$, $u'_2(ABC)$, $u'_3(ABC)$ and $u'_4(AB)$, which is more meaningful than the previous empty set.

In addition, since road maps can be found everywhere, in the domain of privacy-preserving location publishing, it is reasonable to assume road-network information is available to any adversary. Thus, cautions are very much needed when publishing anonymized trajectories. For instance, let us continue from the previous example and assume that the road-network in Figure 1 is accessible to an adversary Bob. If Bob observes that Alice passes by road $\overline{AB}$ and $\overline{BD}$ at similar time every weekday, then Bob can infer that u'_4 is Alice who is the only one with trajectory entering $\overline{BD}$ in this published dataset. Upon knowing the anonymous ID of Alice, Bob can track Alice's remaining trajectories in the published dataset. This *inference-route problem* is caused by the fact that an adversary can infer someone's unpublished trajectories from the published location dataset. Because the inferred trajectories are infrequent (i.e., not many users have such trajectories), with high probability, these trajectories, combined with certain external knowledge, can be used to identify a particular individual's trajectory information in the published dataset. In general, given a threshold k, if the attacker can link any anonymous ID to Alice with probability greater than $\frac{1}{k}$ by using the above method, then we say there is an inference-route problem.

In this paper, we address the problem of privacy-preserving location data publishing under the assumption that road-network data are public information. Our approach has three main properties: (1) it guarantees k-anonymity of published data, (2) it avoids the inference-route problem, and (3) the anonymization results follow the road-network constraints. The basic idea is to employ a clustering-based anonymization algorithm to group similar trajectories and minimize the data distortion caused by anonymization through a careful selection of representative trajectories. We propose a C-Tree (Cluster-Tree) to speed up the clustering process and develop methods to incrementally calculating error rates.

A preliminary version of this paper appears in [Lin et al. 2010], where we presented the basic idea of the trajectory anonymization. In this paper, we make the following additional contributions. First, we proved that the anonymization result of our approach satisfies strict k-anonymity. Second, we improved the anonymization approach by taking into account both global and local error rates. Third, we proposed a method to automatically determine the threshold used during clustering. Fourth, we conducted an extensive experimental study including detailed analysis of our approach and comparison to the most recent related work by using both synthetic and real datasets. In addition, we also provide more detailed description of our algorithms.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 gives the problem statements. Section 4 presents our proposed approach. Section 5 reports experimental results. Finally, Section 6 concludes the paper.

2. RELATED WORK

Most existing works on privacy-preserving location publishing consider trajectories represented as sequences of coordinates and output anonymization results in the form of cloaking regions or centers of clusters. However, these approaches do not generate anonymized trajectories that follow the road network constraints. Their anonymization results preserve the user privacy but are not beneficial for traffic analysis on individual roads, while our goal is to achieve both. In what follows, we give an overview of these approaches. In [Gidofalvi et al. 2007], the spatial-temporal cloaking technique is applied to generate cloaking regions covering segments of trajectories. In [Abul et al. 2008; 2010b], Abul et al. consider a trajectory as a cylindrical volume where the radius represents the location imprecision. Then they perturb and cluster trajectories with overlapping volumes to ensure that each released trajectory volume encloses at least $k-1$ other trajectories. In [Nergiz et al. 2009], each trajectory is an ordered set of spatio-temporal 3D volumes(e.g. points). It adopts a con-

densation based grouping algorithm for trajectory k-anonymity. Each cluster is then anonymized which ensures optimal point matching that will minimize the log cost. Finally reconstruction is deployed to output atomic trajectories to ensure privacy. In [Monreale et al. 2010], Monreale et al. cluster trajectories and transform them into into a sequence of centroids of Voronoi cells. Such anonymized trajectories are no longer real trajectories. They can be located even in the middle of two parallel roads. In [Domingo-Ferrer and Trujillo-Rasua 2012], trajectories are clustered based on a distance function and then a location time triple in an anonymized trajectory is replaced by an existing triple with close proximity in original trajectory to satisfy k-anonymity. However, two triples though close in proximity may belong to two different roads and hence making it easy for the adversary to identify fake trajectories given the road map is publicly known. In [Abul et al. 2007], Abul et al. used a coarsening strategy which removes one or more spatial points in a trajectory to achieve anonymization. An anonymized trajectory may contain disconnected paths. This is different from our approach which preserves continuous trajectories based on road-network information. Similarly, in [Mohammed et al. 2009], Mohammed et al. adopt a greedy algorithm to suppress locations in the trajectories to achieve anonymity. However, using suppression alone may decrease the utility of the anonymization results. They did not provide any experimental study to prove the effectiveness of the approach. Unlike the previous works which are based on the similarity of trajectories, Yarovoy et al. [Yarovoy et al. 2009] group trajectories based on so-called quasi-identifiers which is hard to be selected in practice.

Some other works consider trajectories that are represented by landmarks or locations of interests. However, such kind of trajectories provide mainly moving patterns rather than real trajectories as considered in our work. For example, [Andrienko et al. 2009] considers various behaviors of moving objects like positions of start and end, significant turns and significant stops to cluster the trajectories. [Monreale et al. 2011] proposes a generalization approach of semantic trajectories, temporally ordered sequence of important places visited by a moving object with the help of a places taxonomy. However, even though a sensitive location for instance an Oncology clinic may be generalized to Clinic, there may be only one clinic at that location and hence an adversary could still reveal the sensitive information. Two other related works used time confusion and path confusion respectively. The time confusion approach [Hoh et al. 2007] mixes location samples of different trajectories, and the path confusion approach [Hoh and Gruteser 2005] crosses paths in areas where at least two users meet. The main problem of the two approaches is that traffic flows are no longer preserved.

In addition, a few works make the assumption that attackers have certain prior knowledge and take such prior knowledge as input for anonymization, while our anonymization is more general and does not need such assumptions. For example, in [Terrovitis and Mamoulis 2008], Terrovitis and Mamoulis assume that the adversaries know partial trajectory information of some individuals. They use it as part of input to their anonymization algorithm. Such usage limits the generality and feasibility of their approach. In [Abul et al. 2010a], trajectories are consistent with the road network. However, the work serves a completely different purpose compared to our work. They hide the sensitive patterns from spatio-temporal patterns by taking a given set of sensitive patterns as input parameters whereas our approach does not assume such kind of prior knowledge. In [Chen et al. 2011], Chen et. al. propose algorithm to publish trajectory data which is differentially private by adding noise to a prefix tree under Laplace transform. Their approach has a totally different privacy preserving goal compared to our work.

The most related work is by Pensa et al. [Pensa et al. 2008]. They proposed a prefix-tree based anonymization algorithm which guarantees k-anonymity of the published trajectories in a way that no trajectories with support less than k will be published. They defined the support of a trajectory Trj as the number of trajectories containing Trj, which however causes the inference-route problem. Here, we can see that how the concept of k-anonymity is applied will affect the quality of the anonymization result.

To sum up, our work is superior to existing works in terms of the following two major aspects: (1) our anonymized trajectories follow road-network constraints and hence are more effective for traffic

analysis; (2) our anonymized trajectories prevent inference problems that have never been studied by any others before.

3. PROBLEM STATEMENT

In general, raw data collected by location-based applications contains user (object) information as a four-tuple $\langle ID, loc, vel, t\rangle$, where ID is the object ID, loc and vel are object location and velocity at timestamp t respectively. The anonymized dataset contains object information in the form of $\langle aid, rid, dir, t_{int}\rangle$, where aid is an anonymized object ID, rid is a road ID, dir is the object's moving direction, and t_{int} is a time interval that includes the object actual traveling time t. Here, for privacy concerns, we replace specific locations and velocities by road ID and moving direction; we anonymize trajectories in the same time interval t_{int} to preserve the time relationship among trajectories. Such representation is sufficient to derive trajectories or traffic flow information.

The road network is modeled as a directed graph, where each edge corresponds to a road with objects moving at one direction, and each node represents an intersection. Specifically, an edge is represented as $\overline{n_i n_j}$, which means objects move from node n_i to node n_j. Each directed edge is given a road ID r_i.

We next define the frequent road and inference-route problem.

DEFINITION 1. Let W be a time interval, and let k be a threshold. We say a road is a frequent road if the number of moving objects moving along one direction on this road is no less than k within time W. We call the number of moving objects the frequency of the road.

In case the trajectory dataset covers a long time frame (e.g, days, weeks or months), we will divide the time frame into shorter intervals (e.g., hours) and anonymize trajectories falling into the same time interval. The motivation is that trajectories sharing roads may not have enough impact on each other if they are far apart temporally. The unit of division of time frame should be selected such that trajectories sharing roads may influence each other on various conditions like increase in traffic or accidents. We support two types of time dimension partitioning. One is to let users define a time frame which depends on their time period of interest and the other is that we uniformly divide the time frame. The unit of division we chose is one to five hours.

DEFINITION 2. Let Υ be an intersection of roads r_1, ..., r_m, and let U_i^+, U_i^- be the sets of objects moving toward and outward Υ on road r_i ($1 \leq i \leq m$) during W, respectively. If $\exists\ U_i^+$, U_j^-, $|U_i^+| \geq k$, $|U_j^-| \geq k$, and ($0< |U_i^+ - U_j^-| < k$ or $0< |U_j^- - U_i^+| < k$), then we say Υ has an **inference-route** problem.

In the above definition, the constraints $|U_i^+| \geq k$, $|U_j^-| \geq k$ ensure that only frequent road segments are considered, and ($0< |U_i^+ - U_j^-| < k$ or $0< |U_j^- - U_i^+| < k$) check if there is an inference-route problem. To have a better understanding, let us revisit the example in Figure 1. Node B is an intersection of three roads. On road $\overrightarrow{AB}$, $U_{AB}^+ = \{u_1, u_2, u_3, u_4\}$; on road $\overrightarrow{BC}$, $U_{BC}^- = \{u_1, u_2, u_3\}$. Since $U_{AB}^+ - U_{BC}^- = \{u_4\}$, $|U_{AB}^+ - U_{BC}^-| = 1 < k$, node B has an inference-route problem.

Next, we present how to evaluate the quality of the anonymized dataset of trajectories. Intuitively, the less difference between the anonymized dataset and the original dataset, the better quality the anonymized dataset is. Therefore, we use two commonly accepted metrics: average error rate and standard deviation. Suppose there are N roads (or edges in a road-network graph) and r_i represents road i. Let $original_{r_i}$ and $anonymized_{r_i}$ denote r_i's original frequency and frequency after the trajectories have been anonymized. Then in Equation 1, the error function E is defined as the average difference between $original_{r_i}$ and $anonymized_{r_i}$ (i.e., E_i), and σ is the standard deviation of the error rates. A low standard deviation indicates that the anonymization quality of each road is similar and close to the average error rate.

$$E = \frac{1}{N}\sum_{i=1}^{N} E_i = \frac{1}{N}\sum_{i=1}^{N} \frac{|anonymized_{r_i} - original_{r_i}|}{original_{r_i}} \quad (1)$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(E_i - E)^2} \quad (2)$$

4. OUR APPROACH

In this section, we present our anonymization algorithm. It consists of two main steps. First, we partition the time axis into intervals, and group records within the same interval. In each obtained sub-dataset D, we remove records associated with infrequent roads, i.e., roads with less than k objects within same time interval. We denote the obtained dataset as D'. In D', we construct partial trajectories for the remaining objects based on moving directions. Note that one user may have several disconnected partial trajectories because he may visit some infrequent roads. Each partial trajectory will be assigned an anonymous ID. For the rest of the paper, words "trajectory" and "partial trajectory" are interchangeable.

The second step is the core of the anonymization process. We propose a clustering-based anonymization algorithm which guarantees that by achieving strict k-anonymity (defined in Section 4.1) among partial trajectories, our anonymization result is free of the inference-route problem. Compared to traditional k-anonymization approaches, our approach not only needs to minimize errors caused by anonymization but also needs to satisfy some unique requirements. Road-network constraints should be enforced during the entire anonymization process, especially when computing the representative trajectories. The first step is relatively straightforward. Therefore, the following discussion focuses on the anonymization step.

4.1. An Overview of Clustering-based Anonymization

The essential idea of clustering-based anonymization algorithm is to find clusters of similar trajectories and anonymize them by using a representative trajectory. The details are the following.

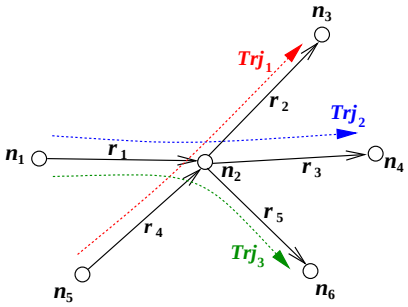


Fig. 2. Trajectory Representation

First, we need to select a proper way to represent trajectories. Trajectories are initially represented as a sequence of timestamped locations. In our anonymized dataset, we do not disclose exact locations because detailed information increases attackers' chances to link published location to specific individuals. Instead, we report only information about which object passing by which road. There are two options: (i) representing a trajectory by road IDs; or (ii) representing a trajectory by node IDs. As illustrated in Figure 2, trajectories Trj_1, Trj_2 and Trj_3 can be represented as r_4r_2, r_1r_3, and r_1r_5 respectively following the first option. Using the second option, trajectories Trj_1, Trj_2 and Trj_3 can be represented as $n_5n_2n_3$, $n_1n_2n_4$, and $n_1n_2n_6$ respectively. Both types of representations well capture the similarity between trajectories Trj_2 and Trj_3 which share one common road. However, the first option treats Trj_1 and Trj_2 as two irrelevant trajectories even though they intersect. To better reflect relationships among trajectories, we adopt the second option and represent a trajectory by a sequence of node IDs.

The second issue is to define the distance between trajectories. Since a trajectory can be seen as a string of road-segment IDs, we employ the *edit distance* [Wagner and Fischer 1974] to compute the amount of different road-segment IDs in the two trajectories. Specifically, the edit distance between two trajectories is given by the minimum number of operations needed to transform one trajectory

Clustering-based Anonymization (TRJ, k)
Input: TRJ is a set of trajectories to be k-anonymized

1. Group same trajectories and form TRJ'
2. Sort trajectories in TRJ' in a descending order of supports
3. **for** each Trj in TRJ' **do**
4. **if** $Trj.support \geq k$ **then**
5. create a new cluster for Trj
6. **else**
7. check existing clusters
8. **if** Find_Cluster(Trj,C) **then**
9. insert Trj to cluster C
10. Select_Representative_Trajectory(C,Trj_r)
11. update C's error rate
12. update $C - tree$
13. **else**
14. create a new cluster for Trj

/* Clustering Adjustment Phase */

15. **for** each cluster C
16. **if** $C.Total\_TRJ \geq \rho_a$ **then** set $C.Total\_TRJ = k$
17. **else** remove C

/* Data Publishing */

18. Translate representative trajectories into output format

Fig. 3. An Outline of Clustering-based Anonymization Algorithm

into the other, where an operation is an insertion, deletion, or substitution of a node. For example, the edit distance between $Trj_1(n_5n_2n_3)$ and $Trj_2(n_1n_2n_4)$ is 4, while the distance between Trj_2 and $Trj_3(n_1n_2n_6)$ is 2.

Now we are ready to present our clustering-based anonymization algorithm. An outline is given in Figure 3. First, we group same trajectories and count its *support*. Support is defined as the number of users who have the same trajectories (Definition 3).

DEFINITION 3. Let u be a user's anonymous ID and Trj_u denote his trajectory in D'. We have the support of trajectory Trj as follows: Support(Trj) = $|\{u|Trj_u = Trj, \forall$ u$\}|$.

Distinct trajectories are arranged in a descending order of their supports. If a trajectory's support is more than the anonymization threshold k, the trajectory itself forms a cluster. For the remaining trajectories, say Trj, we compare it with existing clusters. If there exists a suitable cluster, we insert the new trajectory into that cluster and update the cluster's information. Otherwise, a new cluster will be created for Trj. At the end of clustering, there is a *clustering adjustment phase* which deals with clusters containing less than k trajectories. In particular, if a cluster contains less than ρ_a ($\rho_a < k$) trajectories, we directly remove it. Otherwise, we add dummy trajectories to the cluster by increasing the support of the representative trajectory to k. The selection of a proper ρ_a will be discussed in Section 4.5.

Finally, we translate representative trajectories together with their supports into output format, which contains object anonymous IDs, road names, and objects' moving directions. For example, we obtain the following intermediate result after anonymizing the trajectories shown in Figure 1: $u'_1(ABC)$, $u'_2(ABC)$, $u'_3(ABC)$ and $u'_4(ABC)$, where $k = 3$. The published dataset will look like this: $(u'_1, R_1, \overline{AB})$, $(u'_1, R_2, \overline{BC})$, $(u'_2, R_1, \overline{AB})$, $(u'_2, R_2, \overline{BC})$, ..., $(u'_4, R_2, \overline{BC})$, where R_i is the name of a road.

The algorithms for finding candidate clusters and selecting representative trajectories along with definitions of local error rates and threshold will be elaborated in the following subsections.

4.2. Finding Candidate Clusters

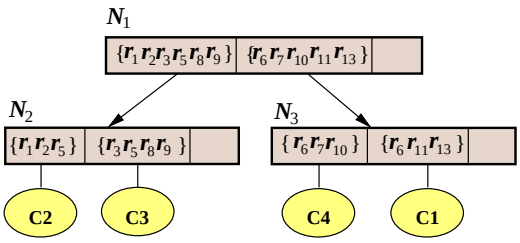


Fig. 4. An Example C-tree

Figure 5 outlines the procedure to find a candidate cluster for a new trajectory. The first step is to check whether a new trajectory can be absorbed by an existing cluster. As the number of clusters increases, comparing Trj with all clusters becomes very costly. Therefore, we employ an in-memory index structure, the C-tree (Cluster-tree), to prune unnecessary comparisons. In particular, each node in the C-tree contains multiple entries and each entry in a node has two fields: a pointer ptr and a set of road IDs (denoted as RID). In leaf nodes, each entry has a pointer to a cluster and the IDs of roads occurring in that cluster. In internal nodes, each entry has a pointer to a child node and the union of roads IDs in its child node. It is worth noting that since we model roads as directed edges, a trajectory can be represented as a set of road IDs without confusion. For example, the trajectory r_4r_2 in Figure 2 can be represented as $\{r_2, r_4\}$ since there does not exist a trajectory r_2r_4 that is against the moving direction. The use of road IDs for representing trajectories here facilitates easy comparison of supports on each road as presented below. Such representation is only used for locating candidate clusters, thus it does not affect the final selection of the most similar trajectory.

Figure 4 illustrates an example C-tree. Given a new trajectory Trj, starting from the root of the C-tree, we calculate the similarity between Trj and RID in every entry of the node by using the following similarity function.

$$Sim_c(Trj, RID) = \frac{|S(Trj) \cap RID|}{|S(Trj)|} \tag{3}$$

Sim_c computes the percentage of common roads included in Trj and RID, where $S(Trj)$ denotes the set of road IDs in trajectory Trj. If Sim_c is above a threshold ρ_t, we continue to visit the child node of this entry. This process is repeated until we find all entries in the leaf nodes with Sim_c above the threshold. All the clusters belonging to these entries will be considered as candidate clusters. For example, suppose that a new trajectory contains roads r_2, r_8 and r_9, and the threshold ρ_t is 60%. The similarity Sim_c between the new trajectory and the first and second entries in the root node N_1 are 100% and 0% respectively. The tree below the second entry is pruned and thus we do not need to visit node N_3. We continue to visit the child node N_2 pointed by the first entry. The Sim_c between the trajectory and the first and second entries in N_2 are 33% and 67% respectively. Since the second entry has the similarity score above the threshold, its corresponding cluster C_3 becomes the candidate cluster for further consideration.

Among candidate clusters, we calculate the edit distance between their representative trajectories and the new trajectory Trj. Based on the edit distance, we then compute a local error E^c (defined in Section 4.4) and select the candidate cluster with the smallest E^c. Only when E^c is lower than a threshold ρ_c (defined in Section 4.5), Trj will be inserted into the corresponding candidate cluster. Otherwise, a new cluster will be created for Trj.

When actually adding Trj to a cluster, we need to update both the representative trajectory and the corresponding entries in the C-tree. The algorithm for computing the representative trajectory is presented in Section 4.3. After the representative trajectory is determined, we check whether the node in the C-tree with respect to current cluster needs to be updated. If current cluster contains road IDs which are not included in the road ID list of the corresponding C-tree entry, we will append the new road IDs to the road ID list. This change will be propagated to higher levels of the C-tree until an entry containing all road IDs in current cluster is reached. Consider the C-tree in Figure 4 and suppose that a new trajectory that consists of roads r_2, r_8 and r_9 will be inserted into cluster C_3. We check the road list of C_3's entry in the C-tree, which is $\{r_3r_5r_8r_9\}$ and does not contain r_2. We then add r_2 to the road list. Now the second entry in N_2 becomes $\{r_2r_3r_5r_8r_9\}$. Next, we check

Find_Cluster (Trj**,**C**)**
Input: Trj is a trajectory
Output: C is a cluster

```
1.  NODE ← {C-tree.root}
2.  while (NODE is not empty) do
3.      for each node N in NODE do
4.          for each entry en in N do
5.              if Sim_c(Trj, en.RID) > ρ_t then
6.                  if N is not a leaf node then
7.                      add en's child node to NODE
8.                  else add en's cluster to candidate list L_c
9.  for all clusters in L_c do
10.     find clusters with smallest E^c regarding Trj
11.     if E^c < ρ_c then
12.         return the cluster found
```

Fig. 5. Algorithm of Finding Clusters

its parent entry, the first entry in N_1. Since r_2 is included in the first entry in N_1, the tree update operation completes.

In the other case when a new cluster is created for Trj, we need to insert a new entry for this new cluster to the C-tree. Recall that each entry in the node of the C-tree has two fields: (i) a set of road IDs and (ii) a pointer. The maximum number of entries in each node is the same. All insertions start at a leaf node which is identified during the process of finding candidate clusters. We insert the new entry into that node (denoted as N) with the following steps:

(1) If the node N contains fewer than the maximum legal number of entries, then there is room for the new entry. Insert the new entry in the node.
(2) Otherwise N is full, and we evenly split it into two nodes. In particular, we randomly select an entry as seed. Then we compute Sim_c (Equation3) between other entries and the seed. The average of all Sim_c serves as a separation value. Entries with Sim_c above the average are put in the node N, and the remaining entries are put in the new right node N'.
(3) Next, we update the entry pointing to N. The road ID set in the parent is updated to include all roads occur in N. The update may be propagated to the upper levels of the tree. Moreover, if there is a split in the previous step, we need to insert a new entry which includes road IDs in the new node N' to the parent level. This may cause the tree to be split, and so on. If current node has no parent (i.e., the node is the root), a new root will be created above this one.

4.3. Selecting Representative Trajectory

There are two key requirements when selecting a representative trajectory. First, the global error rate E should be minimized. Second, the representative trajectory must satisfy the road-network constraint. By keeping these in mind, we design the following algorithm.

In a cluster, we find the trajectory with the highest support and then trim the trajectory from both ends to obtain the final representative trajectory. To illustrate it, we use the example in Figure 6. The cluster contains three types of trajectories: Trj_1, Trj_2 and Trj_3. Each trajectory is associated with a number of support, e.g., $support(Trj_1) = 10$. Numbers on the last line indicates the original numbers of users on each road, e.g., $original(n_1n_2)$=15. Since Trj_1 has the highest support, let us have a further look at it. We compute the error rate E by treating Trj_1 as the representative trajectory. The support of the representative trajectory is the sum of the supports of all the trajectories in the cluster. The reason behind is to maintain the same amount of trajectories after anonymization.

Trj_1 (10):	n_1—	n_2—	n_4—	n_7—	n_8— n_9
Trj_2 (5):	n_1—	n_2—	n_4—	n_7	
Trj_3 (6):		n_2—	n_4—	n_7—	n_8
original:	15	21	21	16	10

Fig. 6. An Example of Selecting Representative Trajectory

In this example, if we use Trj_1 as the representative trajectory, we will have E = 58%.

$$E = (E_{n_1n_2} + E_{n_2n_4} + E_{n_4n_7} + E_{n_7n_8} + E_{n_8n_9})/5$$

$$= \frac{(\frac{21-15}{15} + \frac{21-21}{21} + \frac{21-21}{21} + \frac{21-16}{16} + \frac{21-10}{10})}{5} = 58\%$$

Observe that $E_{n_8n_9}$ is higher than 100%. If the road n_8n_9 is excluded from the representative trajectory Trj_1, the overall error can be reduced to 34%. Based on this observation, the second step is to trim the roads in the trajectory that can help reduce the overall error rate. Due to the road-network constraint, we cannot arbitrarily remove nodes from a trajectory. Our strategy is to remove nodes starting from both ends of the selected trajectory. Also, we should not remove too many nodes, which otherwise leads to poor pattern preservation. To reach the balance, we only consider removing the nodes with error rate above certain threshold. In our case, we set the threshold to be 100% in order to ensure that the overall error rate does not exceed 100%. Specifically, if a road r which is located at the end of the trajectory and has an error rate larger than 100% (i.e., $original_r < support(Trj_1) - original_r$), this road will be removed from the representative trajectory. The process continues until we cannot find such a road at either end of the trajectory. The final representative trajectory for the example case is $n_1n_2n_4n_7n_8$. The algorithm is summarized in Figure 7.

Select_Representative_Trajectory (C**,**Trj_r**)**
Input: C is a cluster
Output: Trj_r is the representative trajectory

1. support(Trj_r)← 0
2. **for** each Trj in C **do**
3. **if** support(Trj) >support(Trj_r) **then**
4. $Trj_r \leftarrow Trj$
5. support(Trj_r) ← support(Trj)
6. $i \leftarrow 1$; $j \leftarrow length(Trj_r)$-1
7. continue ← 1
8. **while** ($i < j$ and $continue$) **do**
9. continue ← 0
10. **if** original(r_i) <support(Trj_r)-original(r_i) **then**
11. $i \leftarrow i + 1$; continue← 1
12. **if** original(r_j) <support(Trj_r)-original(r_j) **then**
13. $j \leftarrow j - 1$; continue← 1
14. $Trj_r \leftarrow (r_i ... r_j)$
15. return Trj_r

Fig. 7. Algorithm of Selecting Representative Trajectory

4.4. Definitions of Local Error E^c

In the following discussion, we use C to denote a cluster and Trj_r to denote its representative trajectory. Let r_i and $anonymize^c_{r_i}$ denote the road r_i and r_i's frequency after anonymization within cluster C, respectively. Note that here $anonymized^c_{r_i}$ is specific to a cluster and it is different from (just a portion of) global $anonymized_{r_i}$. Formally, the relationship between $anonymized^c_{r_i}$ and $anonymized_{r_i}$ is given in Equation 4, where clusters C_1, ..., C_m are clusters containing road r_i.

$$anonymized_{r_i} = \sum_{j=1}^{m} (anonymized^{c_j}_{r_i}) \tag{4}$$

Given a new trajectory Trj_{new}, E^c is computed by assuming that Trj_{new} has been inserted into cluster C. Let us denote the new cluster with Trj_{new} as C' and assume that the representative trajectory of C' is still the same as C but with an increased support by $support(Trj_{new})$. The definition of E^c is shown in Equation 5, where R is the set of roads appearing in the new cluster C', and $|R|$ denotes the total number of roads in R. For each road r_i in R, we calculate two values, $trans_{r_i}$ and $change_{r_i}$. The value $trans_{r_i}$ is the difference of frequency of r_i in C and C'. The value $change_{r_i}$ is the change of frequency of r_i in the anonymized results of cluster C', i.e., $change_{r_i}$= ($|anonymized^{c'}_{r_i} - anonymized^c_{r_i}|$).

$$E^c = \frac{1}{|R|} \sum_{r_i \in R} E^c_{r_i} = \frac{1}{|R|} \sum_{r_i \in R} (change_{r_i} - trans_{r_i})^2 \tag{5}$$

For better understanding of Equation 5, we illustrate the calculation through the following example. Consider the cluster C containing two types of trajectories: $Trj_1(n_1n_2n_4n_7n_8n_9)$ and $Trj_2(n_1n_2n_4n_7)$, where $support(Trj_1)$=10, $support(Trj_2)$=5. Suppose that the representative trajectory is $Trj_r(n_1n_2n_4n_7n_8)$ and $support(Trj_r)$= 15. We now compute E^c upon the insertion of a new trajectory $Trj_3(n_2n_4n_7n_8)$ with $support(Trj_3) = 6$ into the cluster C. Table I summarizes the changes for each road after the insertion of the new trajectory, where roads are listed in the first column of the table, followed by its original anonymization value ($anonymized^c$), the anonymized value in the new cluster ($anonymized^{c'}$), and corresponding values of *trans* and *change*. Specifically, after the insertion, the anonymized values of the roads in Trj_r will be increased by $support(Trj_3) = 6$ as shown in the second column in Table I and the last column $change$ denote the value of this change. The difference between road frequency in cluster C and C' is shown in the third column of the table, from which we can see that the insertion of the new trajectory does not change the overall frequency of roads n_1n_2 and n_8n_9 since the new trajectory does not contain the two roads.

Table I. An Example of E^c Calculation

Road	$anonymized^c$	$anonymized^{c'}$	$trans$	$change$
n_1n_2	15	15+6=21	0	6
n_2n_4	15	15+6=21	6	6
n_4n_7	15	15+6=21	6	6
n_7n_8	15	15+6=21	6	6
n_8n_9	0	0	0	0

Accordingly, E^c can be computed as follows.

$$E^c = (E^c_{n_1n_2} + E^c_{n_2n_4} + E^c_{n_4n_7} + E^c_{n_7n_8} + E^c_{n_8n_9})$$

$$= \frac{(6-0)^2 + (6-6)^2 + (6-6)^2 + (6-6)^2 + (6-6)^2}{5} = 7.2$$

Compared to the approach using merely E during clustering, E^c is more effective since it captures the effect of error change after inserting a new trajectory. More specifically, the value of E is dominated by $original_{ri}$. If a cluster contains many roads which have a large value of $original_{ri}$, the insertion of even a dissimilar trajectory into the cluster will result in a low E. In other words, global $original_{ri}$ does not truly reflect the situation in a cluster. As more dissimilar trajectories are accumulated in the same cluster, the global error E also increases. Unlike E, E^c is defined with respect to each individual cluster, and hence conquers the aforementioned problem.

E^c has another advantage in that it can be quickly computed based on edit distance. In this way, we avoid a great number of comparison between original number of objects and anonymized number of objects during error calculation. Specifically, E^c can be expressed in terms of the edit distance between the representative trajectory Trj_r and the new trajectory Trj_3 as shown in Equation 6, where ED denote the edit distance.

$$E^c = \frac{1}{|R|} ED(Trj_r, Trj_{new}) \cdot support(Trj)^2 \tag{6}$$

Considering the same example discussed in this subsection, R contains five roads and the edit distance between Trj_r and Trj_3 is 1. Therefore, we can compute E^c as follows, which yields the same result as using Equation 5: $E^c = \frac{1}{5}(6^2) = 7.2$

4.5. Selection of Threshold

The threshold selection is a critical task which affects clustering speed and anonymization accuracy. In this subsection, we discuss how to determine the threshold ρ_a for the clustering adjustment phase and the threshold ρ_c for the clustering process.

After clustering all the trajectories, some clusters may contain less than k trajectories. For these clusters, the threshold ρ_a is used to determine whether to remove the clusters or add dummy trajectories to them. To minimize error after the adjustment, we set the threshold ρ_a as follows.

$$\rho_a = \frac{k}{2} \tag{7}$$

The basic idea of Equation 7 is that insertion or deletion of fewer trajectories induces less error. Specifically, if the total number of trajectories in a cluster is less than or equal to $k/2$, removing the cluster will introduce less error by adding more than $k/2$ dummy trajectories. In the other case, if a cluster has more than $k/2$ trajectories, adding less than $k/2$ trajectories will introduce less error than removing the entire cluster.

The threshold ρ_c determines whether a new trajectory can be inserted into an existing cluster or not. If a low threshold is used, fewer trajectories will be inserted into a cluster as only highly similar trajectories will be selected. This may result in having more clusters with less than k trajectories at the end of the clustering. Such clusters will either be removed or include dummy trajectories, which in turn can increase the error rate. If a high error threshold is chosen, even the trajectories which are less similar may be inserted into the same cluster which also introduces more errors. To reach a balance, we define the threshold ρ_c as shown in Equation 8.

$$\rho_c = \left(\frac{k}{2}\right)^2 \tag{8}$$

This threshold is derived according to the clustering adjustment algorithm. As aforementioned, if a cluster needs to be adjusted, the maximum number of trajectories inserted into or deleted from the cluster is equal to $k/2$. The value of ρ_c is equivalent to the error E^c induced when $k/2$ trajectories are inserted into or deleted from the cluster computed using Equation 5. Given a new trajectory, if the corresponding E^c exceeds ρ_c, this trajectory will not be inserted into the cluster being considered. Therefore, even if the cluster needs to be removed during the adjustment phase, it will not introduce

an error more than ρ_c. Moreover, we can observe that the value of ρ_c depends on the value of k. That is, a larger k yields a higher threshold ρ_c. This is beneficial for the clustering due to the following reason. A larger k may increases the risk of letting more clusters go to the adjustment phase and hence may increase the global error. A higher threshold will counteract this effect as it will group more trajectories into a cluster and reduce the number of clusters with trajectories less than k.

4.6. Strict k-anonymity

In this section, we define the notion of *strict k-anonymity*. It is called "strict" because the calculation of trajectory supports is based on an exact match of entire trajectories.

DEFINITION 4. (Strict k-anonymity over trajectories): Let Trj be a trajectory. We say Trj satisfies strict k-anonymity if Support(Trj) is no less than k.

Our anonymization results guarantees strict k-anonymity over all trajectories in dataset D'. In this way, we ensure that the anonymization result will not contain any inference-route which is given in the following theorem.

THEOREM 4.1. *Trajectories that satisfy strict k-anonymity do not contain any inference-route.*

PROOF. *We prove it by contradiction. Let us assume that our anonymization result contains at least one intersection (denoted as Υ) of roads r_1, ..., r_m, which has the inference-route problem. Then by definition 2, among roads r_1, ..., r_m, there exist at least two roads r_i and r_j such that $|U_i^+| \geq k$, $|U_j^-| \geq k$, but (0< $|U_i^+ - U_j^-| < k$ or 0< $|U_j^- - U_i^+| < k$) (where U_i^+ and U_i^- denote the sets of objects moving towards and outwards Υ, respectively).*

If 0< $|U_i^+ - U_j^-| < k$, that means less than k objects enter Υ from roads other than r_i. It implies that the trajectories of objects in $(U_i^+ - U_j^-)$ have support less than k. Similarly, if 0< $|U_j^- - U_i^+| < k$, that means less than k objects leave Υ and enter roads other than r_j. It implies that the trajectories of objects in $(U_j^- - U_i^+)$ have support less than k. Both cases contradict with the property of our anonymization result which only contain trajectories with support no less than k. Therefore, we conclude that our approach does not have any inference-route problem. □

4.7. Complexity Analysis

In this section, we analyze the time and space complexity of our approach. In what follows, we use n to denote the total number of original trajectories, and use l to denote the maximum number of roads in a trajectory in the raw dataset D.

First, we analyze the time complexity. Our approach consists of two main phases: (1) removal of infrequent roads; and (2) the clustering-based anonymization. To remove infrequent roads from the raw dataset, we need to scan the road segments contained in all the trajectories just once. The total number of such road segments is $n \times l$. Given l being a small and constant number, the complexity of the first step is $O(n)$.

For the clustering-based anonymization, the major cost is the search of the C-tree. Let f denote the average number of entries in a node of the C-tree, and let k_c denote the average number of trajectories per cluster. The height of the C-tree can be estimated as $log_f(n/k_c)$. For each identified candidate cluster, we search from the root down the leaf nodes in the C-tree. The total number of entries to be checked can be estimated by the height of the tree multiplied by the number of entries per node, i.e., $log_f(n/k_c) \times f$. If multiple candidate clusters are identified, the cost is only increased by a small constant number of additional entries being checked. Therefore, the time complexity of finding candidate clusters is still $O(log(n))$. The remaining step is to check each trajectory in the candidate clusters to select a representative trajectory, the cost of which is about $k_c \times l$. Since k_c is proportional to n and l is a small constant number, the time complexity of selecting representative trajectory is $O(n)$. Summing up the time complexity of the two steps, we obtain the total time complexity of the clustering-based anonymization, which is $O(log(n)) + O(n)$.

Finally, the total time complexity of our approach is the sum of the two phases: $O(n)$+ $(O(log(n)) + O(n))$, which is $O(n)$. This indicates that the time complexity of our approach is linear to the total number of trajectories, which is also confirmed by the following experimental results.

As for the space complexity, our approach stores all the trajectories and the C-tree. The total number of road segments in the trajectories are $n \times l$. The total number of nodes in the C-tree is $\sum_{i=0}^{h-1} f^i$, where h is the height of the tree and equals to $log_f(n/k_c)$ as previously discussed. Recall that f is the average entries per node and is a constant number. The total space complexity is $n \times l$ + $\sum_{i=0}^{h-1} f^i$, which is $O(n) + O(f^{log(n)})$.

5. EXPERIMENTAL STUDY

In this experimental study, we first compare our two approaches: Clustering-Based Anonymization (CBA) [Lin et al. 2010] and Improved Clustering-Based Anonymization (ICBA). CBA used E (Equation 1) during the clustering while ICBA used the new metric E^c (Equation 5). Then, we study the effect of the C-tree adopted by ICBA. After that, we compare ICBA with the latest related work (denoted as Prefix [Pensa et al. 2008]) by testing the original source code provided by the authors of [Pensa et al. 2008]. We use both synthetic and map-based datasets and varying a variety of parameters including the data size, data distribution, average trajectory length and value of k.

In the synthetic datasets, objects are moving on a randomly generated road map which has about 700 roads. The roads are generated by randomly selecting points (which serve as intersections) in the space and then connecting nearby points to create the roads. The average degree of an intersection is 4. Objects can have different speeds which are controlled by the parameter "average trajectory length". As for the map-based datasets, we use the generator by Brinkhoff [Brinkhoff 2004]. Objects are moving on real road networks. A road consists of multiple segments and each segment is a straight line. An object is initially placed on a randomly selected road segment and then moves along this segment in a randomly selected direction. When the object reaches the end of the segment, an update is issued and a connected segment is selected. Object speeds are varied within a given speed range which controls the "average trajectory length". Unless noted otherwise we use the data set containing 50,000 moving objects as the default setting. The parameters used in the experiments are summarized in Table II, where values in bold denote the default values.

The performance is evaluated based on five criteria: (i) anonymization time; (ii) average error rate as given by Equation 1; (iii) standard deviation as given by Equation 2; (iv) number of inference-routes in the anonymization result; (v) number of frequent patterns after anonymization. All the experiments were run on a PC with 2.6G Pentium IV CPU and 3GB RAM.

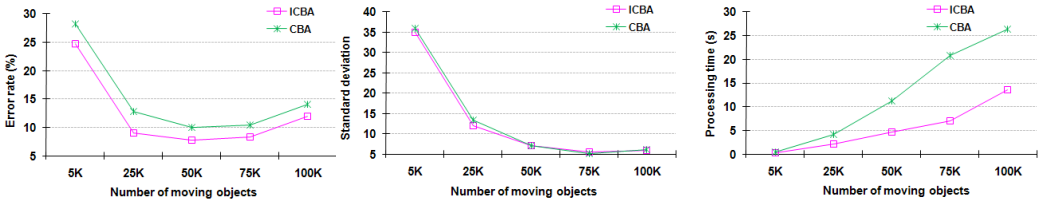
Table II. Parameters and Their Settings

Synthetic Dataset	
Parameter	**Setting**
k	10,20,**30**,40,50
Number of moving objects	5K, 25K, **50K**, 75K, 100K
Average trajectory length (km)	20, 30, 40, 50, 60

Map-based Dataset	
Parameter	**Setting**
k	**10**,20,30,40,50
Number of moving objects	5K, 25K, **50K**, 75K, 100K
Average trajectory length (km)	3.8, 5.0, 5.8, 6.4, 9.2
Number of roads (Map)	209(St Charles), 434(St Clair), **550(Phelps)**, 874(Jefferson), 1689(St Louis)

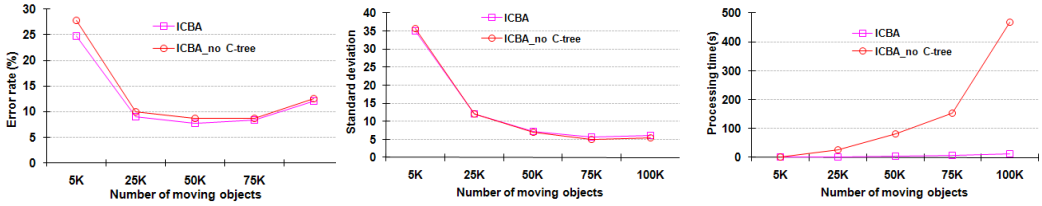
5.1. Anatomy of Our Approaches

5.1.1. CBA vs. ICBA. The first round of experiments compares the performance of our two approaches: CBA and ICBA, by using synthetic datasets. Figure 8(a) shows the average error rate of the anonymization results obtained from CBA and ICBA when varying the number of moving objects from 5K to 100K. Observe that the error rate of ICBA is lower than that of CBA for all cases. This is because CBA adopts a fixed threshold which is set to an experienced value (60%) for all



(a) Error rate (b) Standard deviation (c) Processing time

Fig. 8. CBA vs. ICBA



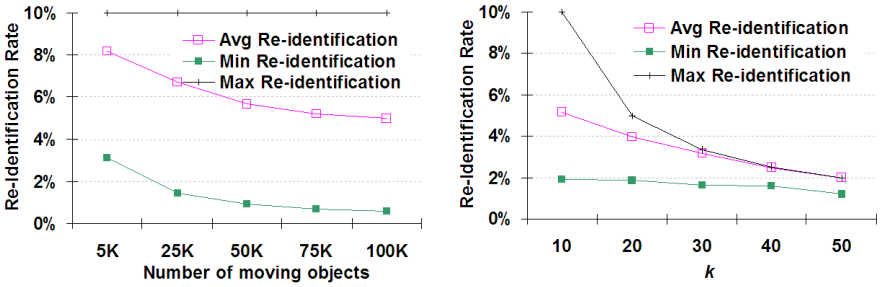
(a) Error rate (b) Standard deviation (c) Processing time

Fig. 9. Effect of the C-tree

cases, while ICBA benefits from the optimal threshold selection (Equation 8) as well as the newly defined metric E^c (Equation 5). Figure 8(b) reports the standard deviation where we can see that ICBA performs similarly to CBA. Figure 8(c) compares the processing time. As shown, ICBA is much faster than CBA. This is because that ICBA uses E^c to measure the intermediate error and E^c can also be expressed in terms of the edit distance which has already been calculated in other steps during the anonymization. In other words, ICBA requires less computation than CBA and hence ICBA is more efficient. In summary, the above observations prove that ICBA improves CBA. Therefore, in the remaining experiments, we will only consider ICBA.

5.1.2. Effect of the C-tree. In this set of experiments, we study the effect of the C-tree by comparing two versions of the ICBA approach: one with the C-tree and one without using the C-tree (denoted as "ICBA_no_C-tree"). Figure 9(a) and (b) report the average error rate and standard deviation with respect to the two versions, and Figure 9(c) compares their processing time. We can observe that the use of C-tree does not affect the accuracy of the anonymization result, but significantly reduces processing time (more than an order of magnitude for 100K datasets), which demonstrates the effectiveness of the C-tree. More specifically, when the C-tree is not used, a new trajectory needs to be compared against all existing clusters, which is time consuming. When the C-tree is used, the new trajectory just needs to be compared with a fewer number of candidate clusters.

5.1.3. Measuring the Probability of Re-identification. We also analyze the probability of re-identification of a user in our anonymized dataset. Note that, all the users in the same anonymization cluster will be represented by the same representative trajectory, and hence they are indistinguishable from one another regardless the amount of prior knowledge that an attacker may have. Thus, the re-identification rate of each user in the same cluster is the same and computed as $\frac{1}{k_c}$, where k_c is the number of trajectories in the cluster. As discussed in Section 4.6, our approach guarantees k-anonymity which means the re-identification probability will not be higher than $\frac{1}{k}$. In the actual experiments, we observe a much lower re-identification rate as reported in Figure 10. In particular, we record the maximum, the average and minimum probability of re-identification rate of all the clusters. The minimum re-identification rate can be as good as $\frac{1}{10}^{th}$ of the theoretical bound when the dataset is 100K. This is because the number of trajectories in each anonymization cluster is usually more than k, and hence it provides better privacy protection than the theoretical guarantee.

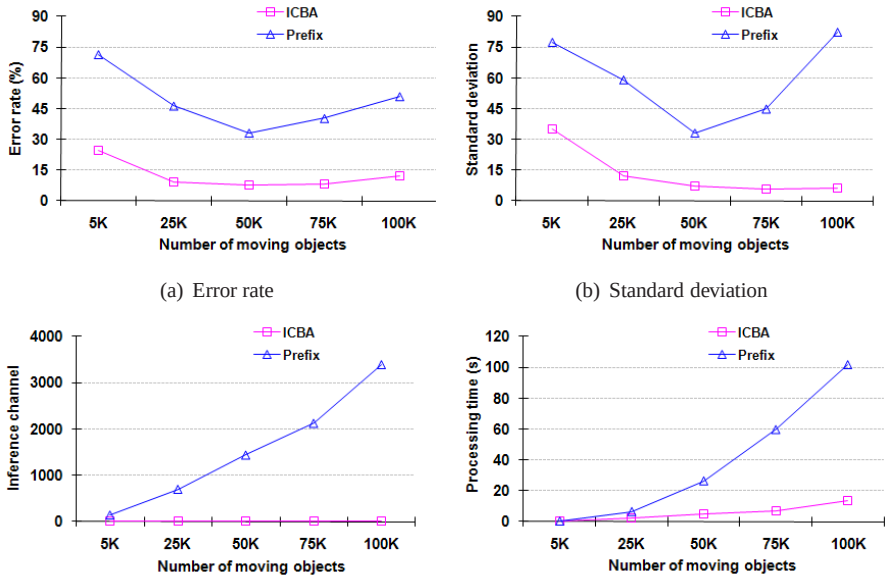


(a) Varying Dataset Size (b) Varying Parameter k

Fig. 10. Probability of Re-identification

5.2. Experimental Results in Synthetic Datasets

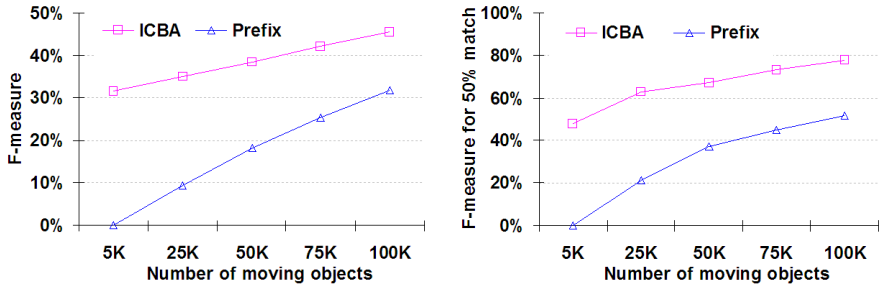
5.2.1. Effect of Data Sizes. We now compare the performance of our ICBA with Prefix approach by varying the number of moving objects (i.e. number of trajectories) from 5K to 100K. Figure 11(a) shows the average error rate of the anonymization results obtained from ICBA and Prefix. We can observe that ICBA yields much less error than Prefix in all cases. When the dataset is small (e.g., 5K), the anonymization results obtained from both algorithms have relatively high error rates. This is because the number of objects on each road is few and even a small change of an object trajectory by the anonymization process will have a big impact on the error rate. With the increase of the data sizes, the error rate caused by ICBA keeps decreasing and it is more than 5 times less compared to that of Prefix for 100K dataset. The reason of such behavior is that ICBA effectively groups similar trajectories and carefully selects representative trajectories, which minimizes the overall error rate. We also measure the standard deviation of the anonymization results obtained from two approaches. As shown in Figure 11(b), the anonymization result generated by ICBA has much lower



(a) Error rate (b) Standard deviation

(c) inference-route problem (d) Processing time

Fig. 11. Effect of Data Size



(a) Exact Match (b) Partial Match

Fig. 12. F-measure

standard deviation than that by Prefix, which indicates that our anonymization result on each road has similarly good quality.

Figure 11(c) shows the number of nodes (i.e., road intersection) having the inference-route problem. It is not surprising to see that the anonymization result produced by our ICBA algorithm contains 0 inference-route. However, the anonymization result obtained from Prefix contains a large number of nodes with the inference problems and the problem becomes more and more severe with the increase of the data sizes, which is caused by their definition of trajectory support.

We also compare the processing time of both approaches. As shown in Figure 11(d), ICBA is up to 5 times faster than Prefix. This can be attributed to the C-tree that helps prune the clusters to be compared with each new trajectory and hence avoids unnecessary calculation. The total time is inclusive of the construction and update cost of the C-tree which is almost negligible compared to the benefits brought by the C-tree.

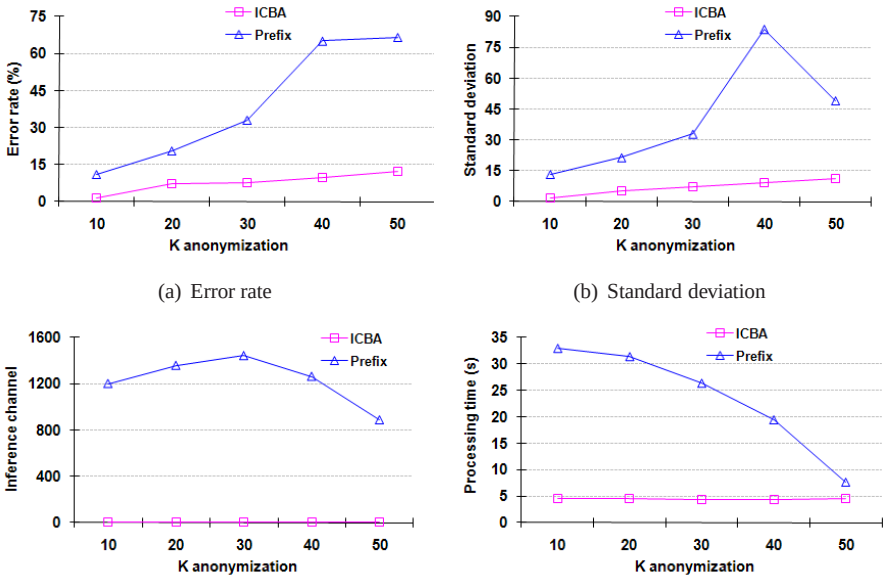
5.2.2. Preservation of Frequent Patterns. We also evaluate the quality of anonymization results by comparing the anonymized trajectories obtained from ICBA and Prefix with the frequent patterns discovered from original datasets using the traditional data mining tool (i.e., PADS software [Zeng et al. 2009]). When using PADS, each transaction is corresponding to an original trajectory. Each item is corresponding to a road ID in the trajectory. We use the anonymization parameter k as the minimum support threshold in PADS. The mining results contain sets of sub-trajectories, each of which is represented as sets of road IDs.

In general, the more frequent patterns are preserved, the better anonymization result is. To measure this, we use the widely adopted F-measure as defined below, where P_r and P_a denote the sets of trajectories in the data mining results and anonymization results respectively, N_m denotes the number of trajectories in the anonymization results that match those in the data mining results, and N_r and N_a denote the total number of trajectories in the data mining results and anonymization results respectively.

$$F(P_r, P_a) = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{9}$$

$$Precision = \frac{N_m}{N_r}, \quad Recall = \frac{N_m}{N_a}$$

Figure 12(a) reports the F-measure values of the Prefix approach and our ICBA approach. Observe that the ICBA approach yields much higher F-measure values than the Prefix approach in all cases, which indicates that ICBA preserves more frequent patterns. This is because the Prefix algorithm directly removes infrequent trajectories which do not share the prefix of a frequent trajectory, while ICBA attempts to preserve the best possible patterns of the infrequent trajectories within the error threshold.



(a) Error rate (b) Standard deviation

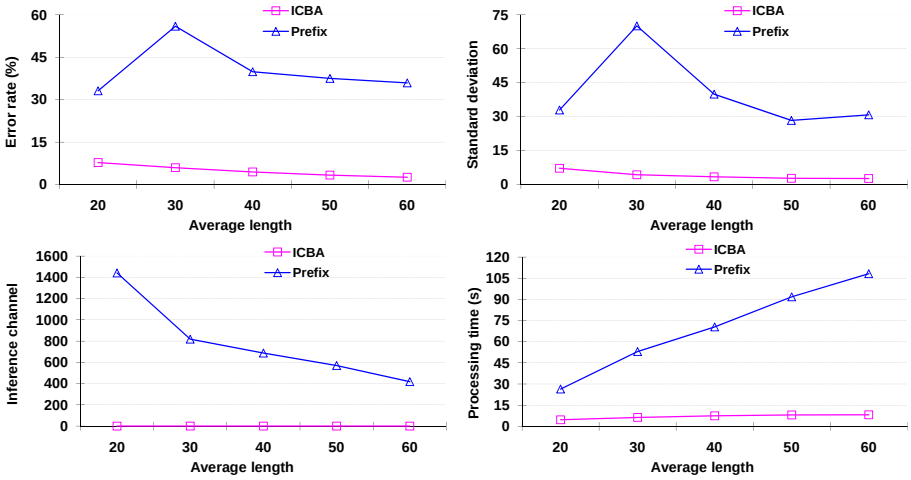
(c) inference-route problem (d) Processing time

Fig. 13. Varying Parameter k

Since trajectory anonymization always needs to distort trajectories in the output, it is unrealistic to expect to receive a perfect F-measure value which means all anonymized trajectories fully match the original frequent trajectories. Therefore, we also evaluate how many trajectories that partially match the data mining results. For this, we record the anonymized trajectories that have at least 50% road segments matching a frequent pattern in the original data mining results, and add them to N_m for computing the F-measure. Figure 12(b) shows the results. From this figure, we can see that the F-measure values have been almost doubled compared to that in Figure 12(a). This indicates that the anonymization results preserve partial frequent pattern information very well.

5.2.3. Effect of Parameter k. This set of experiments aims to evaluate the performance of both algorithms regarding different values of k. As shown in Figure 13(a), the error rate increases drastically with k by using the Prefix algorithm, while k has only minor effect on our ICBA approach. Such behavior can be explained as follows. Prefix removes all infrequent trajectories and adds their supports to most similar frequent trajectories. When k is large, there are more infrequent trajectories, which thus causes more errors. The standard deviation (Figure 13(b)) also demonstrats the similar pattern as the error rate. Moreover, Prefix again suffers from the inference-route problem as can be observed from Figure 13(c). Regarding processing time (in Figure 13(d)), ICBA has a consistent performance and is much faster than Prefix when k is small. When k grows bigger, the processing time of Prefix decreases. This is because Prefix needs to handle less number of frequent trajectories for a larger k, which in turn results in higher error rates.

5.2.4. Effect of the Average Trajectory Length. We now evaluate the effect of the average length of the trajectory in terms of number of roads. The length is determined by two factors: the length of time interval being considered and object moving speed. As shown in Figure 14(a) and (b), Prefix incurs much higher error rate and standard deviation than ICBA does for various lengths of trajectories. This behavior can be attributed to the fact that longer trajectories increase the possibility of getting more trajectory pattern with support less than k. Using the Prefix algorithm, the support of a trajectory pattern will be added only to the common prefix between the trajectories. Therefore, if the starting node of trajectories differ, the support will not be added even though these trajectories



(c) inference-route problem (d) Processing time

Fig. 14. Varying Average Length of the Trajectory

may share the suffix or an infix. On the other hand, ICBA attempts to capture similarity between trajectories either as prefix or suffix or an infix. This leads to less error in ICBA than the Prefix algorithm.

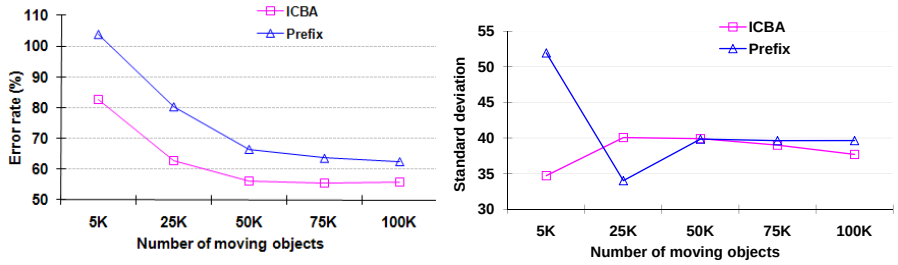
As for the inference-route problem (Figure 14(c)), the total number problematic nodes generated by Prefix decreases as the trajectory length becomes longer. This is possibly because that the increase of trajectory length results in less frequent trajectories and reduces the chance of having inference-route problems.

As shown in Figure 14(d), there is a drastic increase in anonymization time with the increase of average length of the trajectory when using the Prefix algorithm. The reason is that longer trajectory increases the depth of the prefix tree, and hence more time is needed for the anonymization process.

5.3. Experimental Results in Map-based Datasets

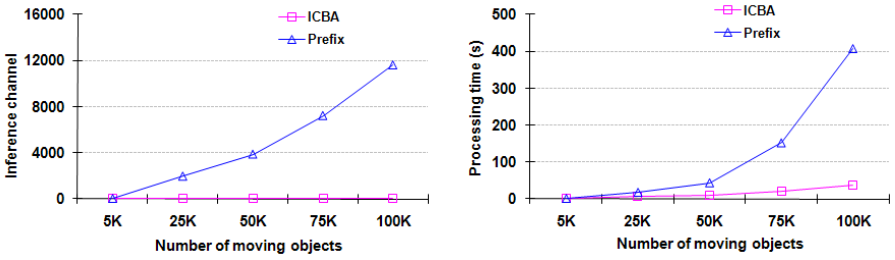
We proceed to evaluate the performance of ICBA and Prefix by using datasets generated based on real road maps using the generator in [Brinkhoff 2004]. We examine the same four aspects: variation of data sizes, frequent patterns, value of k and average trajectory length, as that in synthetic datasets. In addition, we also study the effect data distribution by using different road maps.

5.3.1. Effect of Data Sizes. In this set of experiments, the datasets are generated based on the road map of Phelps County (Missouri, USA) which contains about 550 roads. As shown in Figure 15 and



(a) Error rate (b) Standard deviation

Fig. 15. Effect of Data Sizes (Real Road-network)

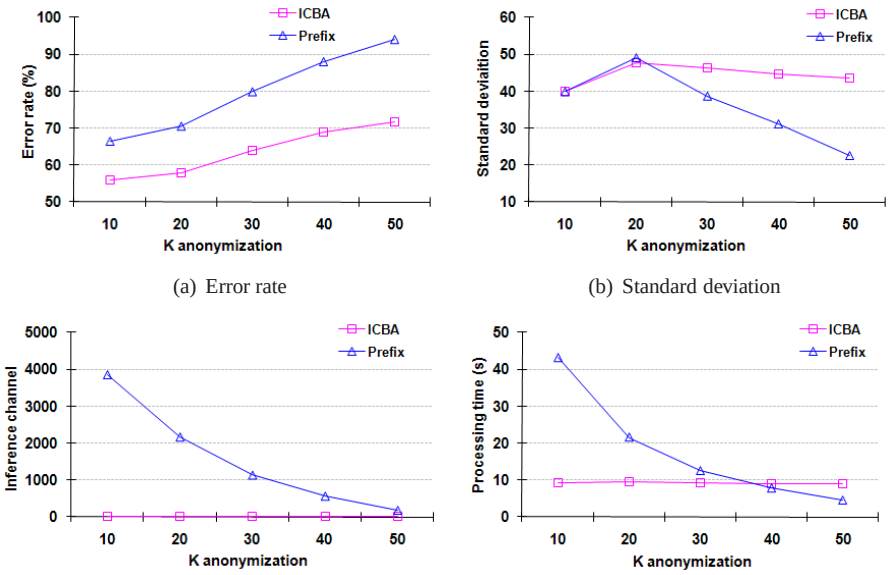


(a) inference-route problem (b) Processing time

Fig. 16. Effect of Data Sizes (Real Road-network)

Figure 16, ICBA consistently outperforms Prefix in terms of both effectiveness and efficiency. The reason is similar to that explained when evaluating synthetic datasets. In addition, both approaches have high error rates when the number of objects (i.e., trajectories) is small and the error rates go down with the increase of objects. This is because in the same road map, fewer objects result in fewer frequent trajectories, and hence the impact of trajectory modification during anonymization is more severe.

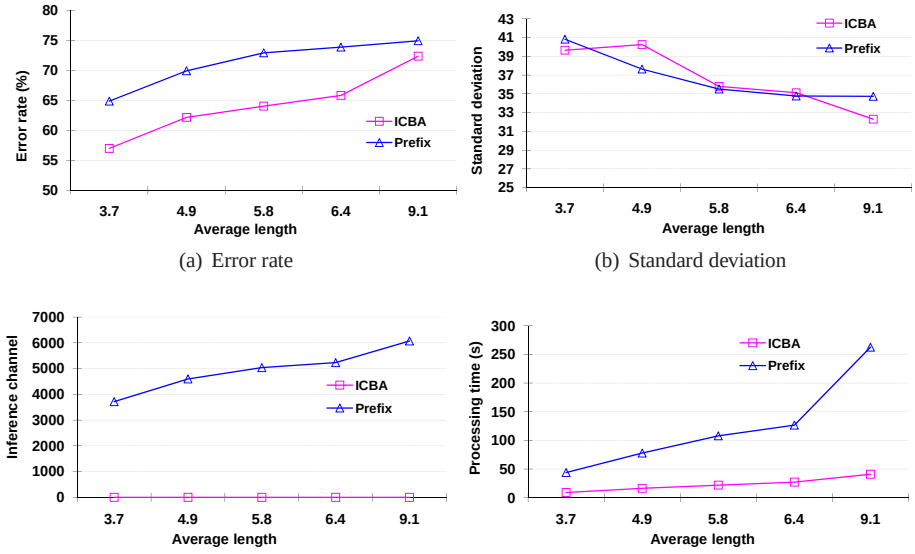
5.3.2. Effect of Parameter k. Figure 17 shows the performance of ICBA and Prefix when varying k from 10 to 50. From the figure, we have the following observations. First, both approaches yield more errors when k increases. The possible reason is that larger k results in less frequent trajectories, and hence any change to trajectories for the anonymization purpose has bigger impact on the final result. Second, it is also interesting to see that Prefix has lower standard deviation, less inference channels and even faster processing speed with a larger k. This is because that Prefix removes more infrequent trajectories for larger k, which means Prefix needs to handle much fewer number of frequent trajectories. Consequently, the standard deviation regarding each frequent trajectory pattern



(a) Error rate (b) Standard deviation

(c) inference-route problem (d) Processing time

Fig. 17. Effect of Parameter k (Real Road-network)



(a) Error rate (b) Standard deviation

(c) inference-route problem (d) Processing time

Fig. 18. Effect of Average Length of Trajectory (Real Road-network)

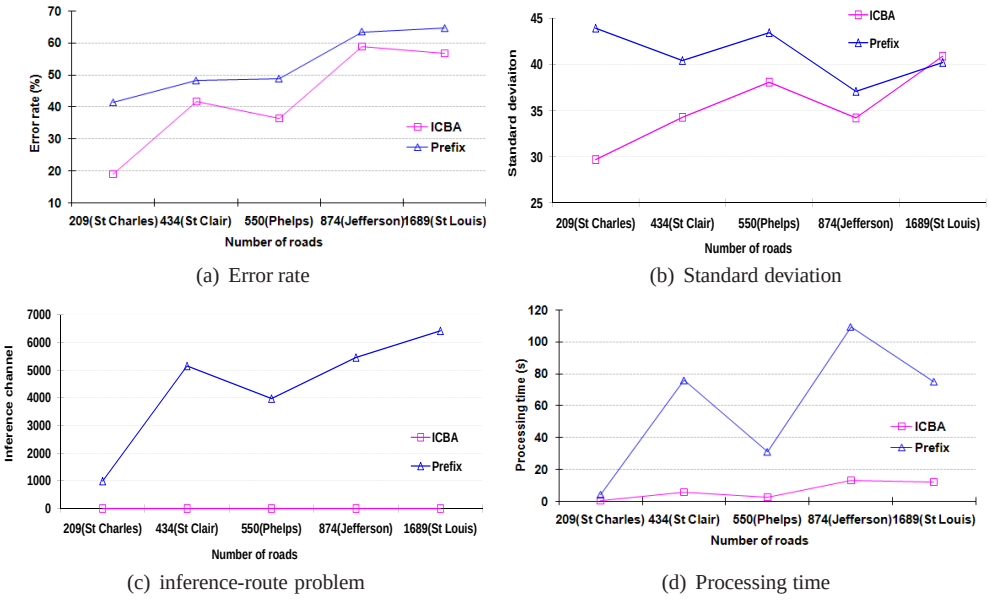
is lowered, the total number of nodes with inference-route problems is reduced and processing time is shorten.

5.3.3. Effect of Average Trajectory Length. This set of experiments evaluates the effect of average trajectory length. As shown in Figure 18, ICBA again outperforms Prefix in general. We also observe that the error rate increases for both approaches when the length of trajectory becomes longer. The reason is similar to that for the case with a larger k in the previous experiments. That is that the reduced number of frequent trajectory patterns with the growth of trajectory length, in turn increases the impact of trajectory modification during the anonymization process. Moreover, with the increase of trajectory length, Prefix suffers more from the inference-route problem. The possible reason is that in the real road-network, the number of roads connected by an intersection is usually small (e.g., two to four). This increases the chance of having nodes with inference-route problems especially in long trajectories. In addition, the trend of the processing time of two approaches resembles the case in synthetic datasets and the reason is also similar.

5.3.4. Effect of Data Distribution. At the end, we study the effect of the data distribution by using various road maps. The total number of objects (or trajectories) is the same, 50K, in all cases. The result is shown in Figure 19. Given different maps, the ratio of frequent to infrequent trajectories is different. This explains the different behavior of error rates for each map. In general, when there are more roads, the number of frequent trajectories becomes less, which may increase the error rate in the anonymized datasets obtained from both approaches. As for the inference-route problem, the more complex the map is (e.g., St. Louis), the higher chance that Prefix generates more inference-route problems in its anonymization result. Moreover, it also takes more time for Prefix to handle larger and complex maps, while our ICBA has relatively stable and much faster processing speed. In a summary, the result demonstrates that ICBA has better topography independency compared to Prefix.

6. CONCLUSION

Privacy preserving location data publishing has received increasing interest nowadays. In this paper, we address this newly emerging problem by taking into account an important factor, the road network constraint, which has been overlooked by many existing works. We identified and de-



(a) Error rate (b) Standard deviation

(c) inference-route problem (d) Processing time

Fig. 19. Effect of Data Distribution

fined a new privacy problem (i.e. the inference-route problem), and proposed an efficient and effective clustering-based anonymization algorithm. We have proved that our clustering-based algorithm guarantees strict k-anonymity of the published dataset and avoids the inference-route problem. To minimize the global error rate after anonymization, we have taken into account the following major aspects: calculation of representative trajectories, definition and employment of local error rates, and selection of threshold used at different stages of anonymization. We conducted an extensive experimental study on both synthetic datasets and real datasets. The results demonstrated the superiority of our approach compared to the latest related work and an earlier version of our work.

Acknowledgement

We thank authors of "Pattern-Preserving k-Anonymization of Sequences and its Application to Mobility Data Mining" who share their source code with us. Dan Lin is supported by University of Missouri Research Board. Rui Zhang is supported by Australian Research Council (ARC) Future Fellowships Project FT120100832.

REFERENCES

ABUL, O., ATZORI, M., BONCHI, F., AND GIANNOTTI, F. 2007. Hiding sensitive trajectory patterns. In *Proc. of ICDM Workshop*. 693 – 698.

ABUL, O., BONCHI, F., AND GIANNOTTI, F. 2010a. Hiding sequential and spatio-temporal patterns. *IEEE Transactions on Knowledge and Data Engineering 22*, 12, 1709–1723.

ABUL, O., BONCHI, F., AND NANNI, M. 2008. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proc. of the International Conference on Data Engineering*. 376–385.

ABUL, O., BONCHI, F., AND NANNI, M. 2010b. Anonymization of moving objects databases by clustering and perturbation. *Information Systems 35*, 8, 884–910.

AGRAWAL, R. AND SRIKANT, R. 2000. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD International Conference on Management of Data*. 439–450.

ANDRIENKO, G., ANDRIENKO, N., GIANNOTTI, F., MONREALE, A., AND PEDRESCHI, D. 2009. Movement data anonymity through generalization. In *Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*. ACM, 27–31.

ATZORI, M., BONCHI, F., GIANNOTTI, F., AND PEDRESCHI, D. 2008. Anonymity preserving pattern discovery. *The VLDB journal 17*, 4, 703–727.

BETTINI1, C., WANG, X. S., AND JAJODIA, S. 2005. Protecting privacy against location-based personal identification. In *Proc. of Workshop on Secure Data Management*. 185–199.

BRINKHOFF, T. 2004. A framework for generating network-based moving objects. `http://www.fh-oow.de/institute/iapg/personen/brinkhoff/generator`.

CHEN, R., FUNG, B., AND DESAI, B. 2011. Differentially private trajectory data publication. *arXiv preprint arXiv:1112.2020*.

DOMINGO-FERRER, J. AND TRUJILLO-RASUA, R. 2012. Microaggregation-and permutation-based anonymization of mobility data. *Information Sciences*.

GIDOFALVI, G., HUANG, X., AND PEDERSEN, T. B. 2007. Privacy-preserving data mining on moving object trajectories. In *Proc. of the International Conference on Data Engineering*. 60–68.

HOH, B. AND GRUTESER, M. 2005. Protecting location privacy through path confusion. In *Proc. of SecureComm*. 194–205.

HOH, B., GRUTESER, M., XIONG, H., AND ALRABADY, A. 2007. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proc. of the ACM conference on Computer and Communications Security*. 161–171.

HUO, Z., MENG, X., AND ZHANG, R. 2013. Feel free to check-in: Privacy alert against hidden location inference attacks in geosns. In *Database Systems for Advanced Applications*. 377–391.

LIN, D., GURUNG, S., JIANG, W., AND HURSON, A. 2010. Privacy-preserving location publishing under road-network constraints. In *Proceedings of International Conference on Database Systems for Advanced Applications*.

MOHAMMED, N., FUNG, B., AND DEBBABI, M. 2009. Walking in the crowd: anonymizing trajectory data for pattern analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 1441–1444.

MOKBEL, M. 2007. Privacy in location-based services: State-of-the-art and research directions. In *Proc. of the International Conference on Mobile Data Management*. 228.

MONREALE, A., ANDRIENKO, G., ANDRIENKO, N., GIANNOTTI, F., PEDRESCHI, D., RINZIVILLO, S., AND WROBEL, S. 2010. Movement data anonymity through generalization. *Transactions on Data Privacy 3*, 2, 91–121.

MONREALE, A., TRASARTI, R., PEDRESCHI, D., RENSO, C., AND BOGORNY, V. 2011. C-safety: a framework for the anonymization of semantic trajectories. *Transactions on Data Privacy 4*, 2, 73–101.

NERGIZ, M. E., ATZORI, M., SAYGIN, Y., AND GUC, B. 2009. Towards trajectory anonymization: a generalization-based approach. *Transactions on Data Privacy 2*, 1, 47–75.

PEI, J., HAN, J., MORTAZAVI-ASL, B., WANG, J., PINTO, H., CHEN, Q., DAYAL, U., AND HSU, M. 2004. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge & Data Engineering 16*, 1, 1424–1440.

PENSA, R., MONREALE, A., PINELLI, F., AND PEDRESCHI, D. 2008. Pattern-preserving k-anonymization of sequences and its application to mobility data mining. In *Proc. of the International Workshop on Privacy in Location-Based Applications*.

SWEENEY, L. 2002. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10*, 5, 571–588.

TANNER, J. 2008. In search of lbs accountability. In *Telecom Asia*.

TERROVITIS, M. AND MAMOULIS, N. 2008. Privacy preservation in the publication of trajectories. In *Proc. of the International Conference on Mobile Data Management*. 65–72.

WAGNER, R. A. AND FISCHER, M. J. 1974. The string-to-string correction problem. *Journal of the ACM 21*, 1, 168–173.

WEI, L.-Y., ZHENG, Y., AND PENG, W.-C. 2012. Constructing popular routes from uncertain trajectories. In *ACM SIGKDD international conference on Knowledge discovery and data mining*. 195–203.

XUE, A. Y., ZHANG, R., ZHENG, Y., XIE, X., HUANG, J., AND XU, Z. 2013a. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In *International Conference on Data Engineering*.

XUE, A. Y., ZHANG, R., ZHENG, Y., XIE, X., YU, J., AND TANG, Y. 2013b. Desteller: A system for destination prediction based on trajectories with privacy protection.

YAROVOY, R., BONCHI, F., LAKSHMANAN, L. V. S., AND WANG, W. H. 2009. Anonymizing moving objects: how to hide a mob in a crowd? In *Proc. of the International Conference on Extending Database Technology*. 72–83.

ZAKI, M. J. 2001. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning 42*, 1/2, 31–60.

ZENG, X., PEI, J., WANG, K., AND LI, J. 2009. Pads: a simple yet effective pattern-aware dynamic search method for fast maximal frequent pattern mining. *Knowledge and Information Systems 20*, 3, 375–391.