Fall 2014

Privacy and trustworthiness management in moving object environments

Sashi Gurung

Department: Computer Science

PRIVACY AND TRUSTWORTHINESS MANAGEMENT
IN
MOVING OBJECT ENVIRONMENTS

by

SASHI GURUNG

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2014

Approved by
Dr. Dan Lin, Advisor
Dr. Ali R Hurson
Dr. Sanjay Madria
Dr. Wei Jiang
Dr. Maciej J Zawodniok

ABSTRACT

The use of location-based services (LBS) (e.g., Intel's Thing Finder) is expanding. Besides the traditional centralized location-based services, distributed ones are also emerging due to the development of Vehicular Ad-hoc Networks (VANETs), a dynamic network which allows vehicles to communicate with one another. Due to the nature of the need of tracking users' locations, LBS have raised increasing concerns on users' location privacy. Although many research has been carried out for users to submit their locations anonymously, the collected anonymous location data may still be mapped to individuals when the adversary has related background knowledge.

To improve location privacy, in this dissertation, the problem of anonymizing the collected location datasets is addressed so that they can be published for public use without violating any privacy concerns. Specifically, a privacy-preserving trajectory publishing algorithm is proposed that preserves high data utility rate. Moreover, the scalability issue is tackled in the case the location datasets grows gigantically due to continuous data collection as well as increase of LBS users by developing a distributed version of our trajectory publishing algorithm which leveraging the MapReduce technique.

As a consequence of users being anonymous, it becomes more challenging to evaluate the trustworthiness of messages disseminated by anonymous users. Existing research efforts are mainly focused on privacy-preserving authentication of users which helps in tracing malicious vehicles only after the damage is done. However, it is still not sufficient to prevent malicious behavior from happening in the case where attackers do not care whether they are caught later on. Therefore, it would be more effective to also evaluate the content of the message. In this dissertation, a novel information-oriented trustworthiness evaluation is presented which enables each individual user to evaluate the message content and make informed decisions.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Dan Lin for her supervision and support. I would not have overcome the challenges in my research endeavors without her vision and suggestions, which guided me to the right direction. She has always been very encouraging and inspiring.

I would like to take this opportunity to express my love and gratitude for my families; baba, mummy, Sanju Dai and Nikhil Kale, and my sano manche, Arbin.

I would like to thank all the members of my dissertation advisory committee, Drs. Ali Hurson, Sanjay Madria, Wei Jiang and Maciej Zawodniok for their time, suggestions and comments in improving the quality of my research and publications.

I would also like to thank all my friends in the Pervasive Computing Laboratory who helped in creating a lively and supportive environment in the lab. I want to thank my besties; Kalpana, Chadani and Sujata, and my friends from Rolla; Shikhu dai, Shristy, Nancy, Tiny and Puja aunty for their love and support.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

1. INTRODUCTION

The use of location-based services (LBS) such as AT&T TeleNav GPS Navigator, Sprint's Family Locator, and Intel's Thing Finder is expanding. Besides these traditional centralized location-based services, distributed location-based services are also emerging attributed to the development of Vehicular Ad-hoc Networks (VANETs) which allows vehicles to communicate with one another and form a dynamic network. For example, through VANETs, a vehicle may send inquiries to vehicles around certain landmarks to obtain the up-to-date parking information, the condition of a road, or convenient lodging. According to Cisco, global mobile data traffic has reached 1.5 exabytes a month and is increasing rapidly. In 2013, 526 million mobile devices were added to cellular and wifi networks [81]. Included in this increase in demand for more data is the use of location based mobile applications. Currently, 74% of adults who own smartphones use their phone to get directions and other information based on their current location. 30% of adults with an account on social media sites say they have at least one of those accounts include their current location in their posts [82]. Even if hand held devices are ignored, as many as 96% of cars mass produced in 2013 are built with event recorders that include GPS [83]. This does not include older cars with other GPS systems or vehicles with OnStar technology. As a result, a huge amount of location information has been collected and stored for analysis. More specifically, in centralized LBS, the central server collects users' locations; in distributed LBS like VANETs, there are Road-Side Units which collects users' locations for authentication purposes.

As more and more personal location data being collected, there have been increasing concerns on users' location privacy. Although many research has been carried out to allow users to submit their locations anonymously, the collected anonymous location data may still be mapped to individuals when the adversary has related background knowledge. For example, a trajectory with an anonymous ID but starting from one's home address can be easily associated with the home owner through public information such as yellow page. In addition, as a consequence of users' need to be anonymous in LBS, it becomes extremely challenging to evaluate the trustworthiness of messages disseminated by anonymous users. Existing research efforts are mainly focused on privacy-preserving authentication of users. Such authentication would discourage most users from misbehaving by tracing of the malicious users after the damage is done. However, it is still not sufficient to prevent malicious behavior from happening in the case of attackers that do not care whether they are caught later on.

Bearing the above challenges in mind, in this dissertation, three approaches are proposed to achieve location privacy and trustworthiness management in centralized and decentralized moving object environments. An overview of the approaches are presented in the following subsections.

1.1. PRIVACY-PRESERVING LOCATION PUBLISHING UNDER ROAD-NETWORK CONSTRAINTS

To improve location privacy, in this dissertation, the problem on how to anonymize the collected location datasets is addressed first so that they can be published for public use without violating any individual's privacy concerns. It is worth noting that publishing of location data can benefit people in many fields.

- **Intelligent Transportation System [11]:** If trajectories consistent with the road network constraints are published, mining of the trajectory data enables offline extraction of interesting patterns with associated temporal factor. These extractions can help find out which routes are busy at which time of the day which further assists in estimating potential points of traffic jams. 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. It also helps answering specific query like how many vehicles entered a certain region and also calculation of effective routes from one location to another at certain time instant or in an emergency (e.g., for ambulances).

- **Infrastructure Construction :** The published trajectory data combined with the knowledge of spatial databases can help determine which regions are most busy and what times of the day. 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.
- **Traveling [79]:** Publishing trajectory data allows the mining of frequent patterns as well as popular destinations. Such extractions can be of great help if you are traveling in an unfamiliar region. Extraction of popular destinations can help you to decide on places to visit and interesting patterns extraction can assist you in efficient trip planning [71, 72]. Overall these extractions can help to make your traveling experiences better.

However, in the meantime, location privacy concerns [37, 44, 62] may hinder the development of the above attractive usage of traffic information. Therefore, a privacy-preserving trajectory publishing algorithm is proposed that protects individual's privacy while preserving high data utility rate after anonymization. Unlike previous works which typically ignore the constraints imposed by road-networks, the approach ensures that the anonymized trajectories still follow the road networks and does not have any so-called inference route problem (definition is presented in Section 3). Theoretical and experimental studies have been conducted using both real and synthetic datasets to prove the efficiency and effectiveness of the approach. Details of this work is presented in Section 3.

1.2. PRIVACY-PRESERVING LOCATION PUBLISHING IN BIG TRAJECTORY DATASETS

The second challenge tackled in this dissertation is how to achieve privacy preserving location publishing when the total number of trajectories is extremely large. As mentioned earlier, LBS users generate 1.5 EB of data every month, and this number is projected to grow to 15.9 EB per month in 2018 [81]. Last year's global mobile internet traffic, at 18 EB, was 18 times the size it was in 2000. This increase is attributed to over a half a billion mobile devices being added to mobile networks last year. [81] Much of that data has location and trajectory information that is stored for analysis. Currently, the data limit for database type storage systems is in the order of exabytes [86]. While this is impressive, the amount of information generated from several cities reporting trajectory data will very quickly exceed this limit. In order to handle data of this magnitude, companies rely on hundreds of thousands of computers working in parallel [87]. And even with these resources, processing time can be often very slow due to the need to access several machines at once and storing the data on multiple servers to allow fault tolerance and recovery. With processing times slow enough already, anonymizing the data to protect privacy will make it take even longer. None of the existing location publishing techniques have considered how to deal with big trajectory datasets.

Therefore, a novel approach is proposed that is able to efficiently anonymize a huge amount of trajectory data. Specifically, based on the previously proposed privacy-preserving location publishing algorithm, a distributed version is proposed by leveraging the MapReduce technique. In Section 4, the details of this approach will be elaborated.

1.3. TRUSTWORTHINESS EVALUATION DURING LOCATION-BASED SERVICES

As a consequence of users being anonymous (attributed to efforts of privacy preserving techniques), it introduces a new challenge in terms of evaluating the trustworthiness of messages disseminated by anonymous users. Existing research efforts are mainly focused on privacy-preserving authentication of users. Such authentication would discourage most users from misbehaving by tracing of the malicious users after the damage is done. However, it is still not sufficient to prevent malicious behavior from happening in the case of attackers that do not care whether they are caught later on. For example, terrorists may take advantage of Vehicular Ad-hoc Networks (VANETs) to send fake message and create massive car accidents.

Therefore, it would be more effective to also evaluate the content of the message. However, due to the dynamic nature of moving objects and the dynamically changing topology of VANETs, existing solutions for information validation in alternative domains such as P2P and social network environments [13,14,23,25,48,49,80], are not suitable. For example, in social network sites, users typically gain reputation if they contribute correct information. Based on one's reputation (and possibly content analysis [14]), other users can determine whether his information is trustworthy. However, reputation is established using a stable network over a relatively long period of time (a day, a week or even longer), and neither one of them exists in VANETs. In VANETs, even if an individual keeps a historical database of vehicles that he traveled along with, the database may not be useful since he may not come across the same vehicles again in the future. Moreover, compared to social networks, the mobility of vehicles imposes strict time constraints on making informed decisions. Notice that authentication protocols are also not sufficient, as they can only certify message origin but cannot guarantee that the identity holder will send truthful and accurate messages in VANETs.

In this dissertation, a novel information-oriented trustworthiness evaluation approach is presented which enables each individual user to evaluate the message content and make informed decisions. In Section 5, the details of this work are presented.

1.4. DISSERTATION OUTLINE

The rest of the dissertation is organized as follows:

- Section 2 reviews different anonymization techniques to preserve location privacy, MapReduce technology and different works in adopting this technology for processing big location data and different approaches used to evaluate trustworthiness in VANETs.

- Section 3 defines a new privacy problem, Inference route problem and attempts to solve it using the proposed clustering-based anonymization technique, an error function to control entry of trajectory to various clusters and C-tree for efficient clustering.

- Section 4 presents the adoption of MapReduce programming model to efficiently anonymize big location data.

- Section 5 presents the proposed real-time trustworthiness evaluation scheme which takes data similarity, data conflict and route similarity into consideration.

- Section 6 concludes the work and discusses directions for future work.

2. LITERATURE REVIEW

Since k-anonymity is very effective for privacy preservation, a brief background information about k-anonymity is presented in this section. Then, existing works on location privacy protection are discussed.

2.1. k-ANONYMITY

The growing demand for sharing information globally, aided by the availability of huge data warehouses, has led to the release of specific data (microdata). Unlike the release of statistical information of the data, release of microdata allows to perform analysis as required. Both computational power and active research going on in data mining are ever-increasing. This helps in effective analysis of the released data. The analysis may reveal interesting patterns which can be deployed for decision making. Neither the removal nor the encryption of explicit identifiers (e.g., social security numbers) is sufficient in ensuring anonymity for privacy protection. Therefore, certain approaches need to be adopted to preserve privacy. Of these approaches k-anonymization is one of the most dominant.

In many cases, information needs to be anonymized before it is shared with other people to ensure that privacy is preserved. To protect against linking attacks, k-anonymity can be used. An example is given in Table 2.1. In this table, explicit identifiers, (i.e., social security number and name) were removed before the table was published to preserve privacy. However, an attacker can still utilize specific information (such as the Voter's list as given in Table 2.2) to identify a particular person. The attacker does this by linking a combination of attributes in Table 2.1 with similar attributes in Table 2.2 (e.g., date of birth, sex, zip code and occupation). For example, the attacker can infer that Alice Smith has tax

Table 2.1. De-identified table (tax-return)

SSN	Name	Date of Birth	Sex	ZIP	Occupation	Tax Return($)
		82/10/12	M	65401	Professor	3000
		83/01/11	F	65402	Software Analyst	4000
		82/11/10	F	65400	Student	1000
		83/12/25	F	65401	Computer Programmer	4000
		83/12/20	F	65400	Marketing Manager	5000

Table 2.2. Public table (Voter's list)

Name	DOB	Sex	ZIP	Occupation
....	....	....	....	....
Smith Alice	83/01/11	F	65402	Software Analyst
....	....	....	....	....

return of $ 2000, a breach of Alice's privacy. These attributes, whose values are available from external sources for linking are termed as quasi-identifiers.

The k-anonymization approach ensures that each released tuple is indistinguishable from at least k other tuples [18]. The probability of identifying the tuple is, at most, $1/k$. Consider $k = 2$, the tuples in Table 2.1 can be anonymized as follows. The attribute "Date of Birth" is generalized by publishing only the birth year. The attribute "ZIP" is generalized by publishing the first four digits and the "Occupation" is generalized as related to either academics or industry. The anonymization result is presented in Table 2.3 which satisfies k-anonymity.

Two approaches are commonly employed to achieve k-anonymity: generalization and suppression. Generalization [6, 24, 60, 66] technique is most often used to achieve k-anonymity. Generalization involves substituting the attribute of published data with

Table 2.3. k- anonymized table

SSN	Name	Date of Birth	Sex	ZIP	Occupation	Tax Return($)
		82	unknown	6540*	Academics	3000
		83	F	6540*	Industry	4000
		82	unknown	6540*	Academics	1000
		83	F	6540*	Industry	4000
		83	F	6540*	Industry	5000

more general values. Certain outlier tuples with support less than k may create a high generalization. For example if Table 2.1 has a tuple $\{$83, M, 68001, Married, Professor, $3000 $\}$, the generalization of attribute “Sex” is increased to unknown and “ZIP code” is increased to 6**** in certain tuples within the table. Therefore this tuple can be considered as an outlier and suppressed accordingly. The released data becomes less accurate as the generalization increased. Generalization with suppression is proposed to increase data utility. Suppressing [1, 5, 41] the outlier tuples helps to achieve k-anonymity within an acceptable generalization. However, the data becomes more incomplete as suppression rate increases. The maximum number of tuples to be suppressed is assumed to have been given and k-Minimal Generalization with Suppression is defined such that this generalization satisfies k-anonymity, the number of tuples suppressed is less than or equal to the given value; and no other generalization exists with a higher information content [10, 38, 58, 69].

In traditional databases, tuples in a single table share the same set of quasi-identifiers. However, in trajectory databases, the quasi-identifiers may vary for each mobile object. An adversary may know the objects’s locations at different times. Therefore k-Minimal Generalization with Suppression is not directly applicable to mobile object databases. The k-anonymity approach remains the dominating approach for preserving privacy due to its practical implications.

2.2. LOCATION PRIVACY

When a mobile object wishes to use a location-based service, it needs to report its location. These locations are collected ubiquitously by location-based service providers such that queries of mobile objects over both mobile objects (for instance, ”find my friends near me”) and static objects (for instance, ”find the nearest Japanese restaurant”) can be accomplished. However, this reporting poses a risk of information misuse. Location information could be linked to real people with the help of publicly available information (e.g., the Yellow Pages). Historical trajectories can be revealed and private information no longer remains private. This privacy violation necessitates some measures for privacy protection before the location information is reported.

Existing works on location privacy protection generally fall into two categories: (1) online location or trajectory anonymization (2) offline trajectory anonymization for trajectory publishing, as shown in Figure 2.1. When a mobile device wants to use a location-based service, it has to report its location along with the service request. The online location or trajectory anonymization is implemented by anonymizing location and trajectory while the mobile device is using the service to preserve privacy. The offline anonymization of trajectories is performed to preserve location privacy while publishing location data collected by various sources (e.g., a location-based service provider). The approach presented here considers a scenario in which location privacy needs to be preserved while publishing trajectory information for mining useful knowledge.

2.2.1. Online Location and Trajectory Anonymization. A great deal of research has been conducted to better understand privacy issues in location-aware mobile devices. Three types of techniques are commonly used to achieve online location anonymization: (1) Policy-based anonymization (2) Spatial-temporal cloaking (3) Encryption-based anonymization.

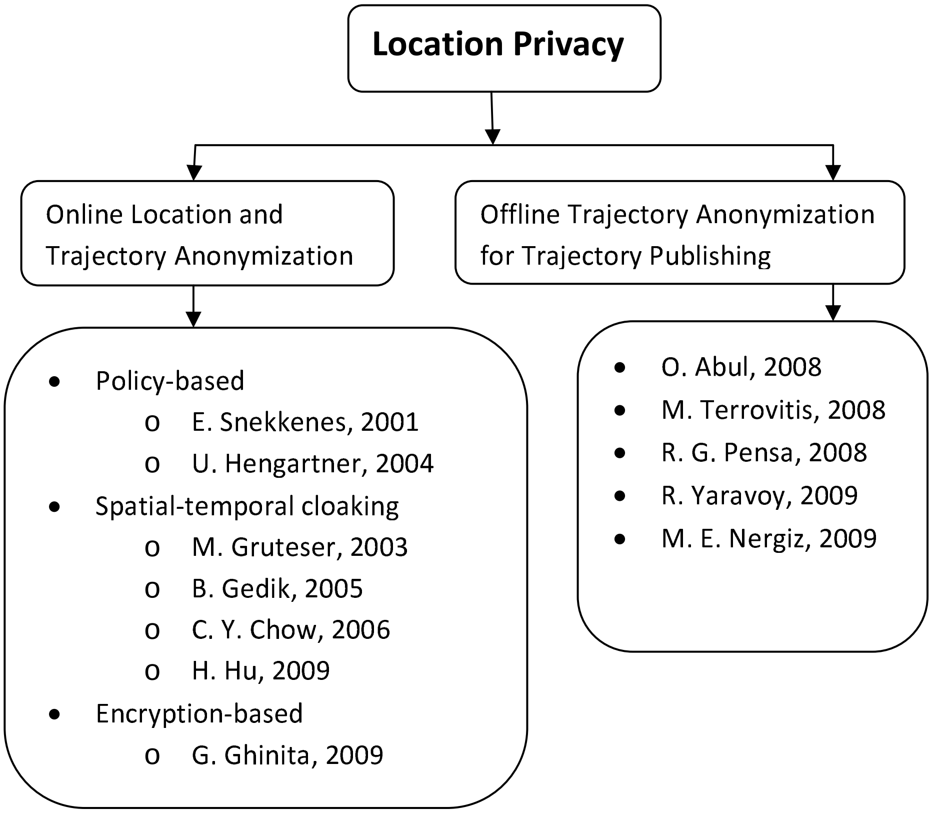


Figure 2.1. Location Privacy Division

Early works focused on maintaining policies on how a user's location could be used by the service providers [33, 59]. However it is difficult to define such policies clearly, to enforce them and also to detect the violation of these policies. Therefore a more practical approach (spatial temporal cloaking) was defined.

Spatial temporal cloaking has been widely used as an anonymization approach for location privacy [17, 26, 29, 30, 32, 36, 45]. Gruteser et al. [32] first introduced the notion of spatial temporal cloaking. As part of this approach, the user's exact co-ordinate location is cloaked into a region (either a rectangle or a circle) such that the user is k-anonymous in that region. They proposed a variation to the spatial temporal cloaking by allowing users to have different values of k according to their privacy requirements. For the cloaking purpose, most approaches [26, 32, 45] used a third party anonymizer and the user reported its exact location to the anonymizer. Mokbel et al. [45] proposed a grid-based cloaking algorithm using a third party anonymizer. This algorithm focused on the granularity metric to obtain an optimal region with k-anonymity. Using a third party for cloaking require the anonymizer to be trusted. An anonymizer can be vulnerable to attacks. It can also be malicious. They only provide protection in a single snapshot and are unprotected against correlation attacks. Recent approaches [17, 29, 30, 36] have focused on cloaking in a peer to peer environment, eliminating the need for a centralized trusted anonymizer. Chow et al. [17] proposed a client form a group of k users among its peers by multi-hop communication and report the region covering the group. Ghinita et al. [30] tried to obtain an optimal cloak region that would satisfy k-anonymity. They proposed $hilbASR$, an approach that used a Hilbert space-filling curve to preserve locality and sort locations. This ordering of locations which preserves proximity was stored in a distributed annotatedB^+-tree index and k users were grouped in this order. These approaches which perform cloaking in a peer to peer environment however, still require that the exact location be revealed to the trusted peers. Hu et al. [36] does not require the exact location of the user to be exposed. It utilizes the proximity information gathered through either the received signal's strength or the time difference in the beacon signal's arrival to identify the k closest peers. A secure bounding protocol is then applied such that the cloaked region's size is reduced and the exact locations are not exposed. This approach is not suitable in a dynamic environment. No mechanism can monitor the user's locations.to keep track of the locations of the users. The cloak region may not contain k users after a certain period of time. In Gidofalvi et al. [31], segments of mobile objects' trajectories are cloaked by a rectangle. The rectangle's size and location probability handle the user's specific privacy requirements. Anonymization is done on the client's side eliminating the need for a trusted middleware. However, the cloaking rectangle can be mapped to the actual trajectory if the rectangle covers only one road in the real map.

Ghinita et al. [28] deployed encryption based techniques that used a grid based framework to preserve location privacy. In this technique, the user encrypts the cell at which he is located and he retains the ability to retrieve correct information. This process is based on Private Information Retrieval(PIR) and supports private nearest neighbor queries. Unfortunately this technique requires the entire database to be encrypted and is computationally expensive.

A comparative analysis of these three approaches is performed (see Table 2.4). Comparisons are done with respect to the privacy protection model, query accuracy, complexity and the use of a trusted agent. Policy-based approaches provide the least privacy protection as nothing is implemented to preserve privacy, they are just policies. Query accuracy depends on the accuracy of the location reported. Policy-based approaches provide 100% accurate query results. They also maintain the lowest complexity because they are simple policies and do not use third party anonymizers. Spatial-temporal cloaking provides an in-between privacy protection among the three approaches. A cloaked location is reported and the probability of identifying the location in the cloaked region fulfils the privacy requirement. It provides lower query accuracy than either of the other two approaches due to the reporting of the cloaked region. It has higher complexity than policy based. A mobile object has to contact a third-party anonymizer to cloak its location before using a location-based service. However, it has less complex than encryption-based approach as encryption is not used. The encryption-based approaches provide the highest privacy protection as the entire database and location is encrypted. They also provide 100% query accuracy. However, they have the highest complexity as encryption is computationally expensive. They do not use any third-party anonymizers.

2.2.2. Offline Trajectory Anonymization for Data Publishing. Privacy preserving location publishing is a relatively young area in which little research has been conducted. Studies conducted on privacy-preserving location publishing considered trajectories that were represented as sequences of coordinates; they utilized output

Table 2.4. Online Location Privacy Preserving Approaches Analysis

Approaches	Privacy Protection Level	Query Accuracy	Complexity	Use of Trusted Agent
Policy-based	Low	High (100%)	Low	No
Spatial-temporal cloaking	Medium	Low	Medium	Yes
Encryption-based	High (100%)	High (100%)	High	No

anonymization results in the form of either cloaking regions or centers of clusters. However, these approaches did not generate anonymized trajectories that followed the road network constraints.These anonymization results preserved the user's privacy but were not beneficial to the traffic analysis of individual roads. The goal of this study was to achieve both.

Nergiz et al. [51] represented each trajectory an ordered set of spatio-temporal 3D volumes (e.g., points). Their approach adopted a condensation based grouping algorithm for trajectory k-anonymity. Each cluster was then anonymized to ensure that the optimal point matching minimized the log cost. Finally, reconstruction was deployed to output atomic trajectories and ensure privacy. Monreale et al. [46] clustered trajectories and then transformed them into into a sequence of Voronoi cell centroids. Such anonymized trajectories are no longer real trajectories. They can be located even in the middle of two parallel roads. Domingo-Ferrer et al. [20] used a distance function to cluster trajectories. They replaced a location time triple in an anonymized trajectory with an existing triple that was in close proximity to the original trajectory , thereby, satisfying k-anonymity. Two triples, though close in proximity, may belong to two different roads. This will make make it easier for the adversary to identify fake trajectories given the road map is publicly available. Abul et al. [1] used a coarsening strategy to remove one or more spatial points in a trajectory to achieve anonymization. An anonymized trajectory may contain disconnected paths. Similarly, Mohammed et al. [43] adopted a greedy algorithm to suppress locations in the trajectories and achieve anonymity. However, using suppression alone may decrease the utility of the anonymization results. Mohammed et al. [43] did not provide any experimental results that would prove the effectiveness of their approach. Abul et al. [3] considered a trajectory to be a cylindrical volume in which the radius represents the location's imprecision. They then perturbed and clustered the trajectories with overlapping volumes to ensure that each released trajectory volume enclosed at least $k - 1$ additional trajectories. Finally they used the sum of the euclidean distance between location points at each trajectory's time points to measure the clusters' similarities. Rather than grouping trajectories according to their similarities, Yarovoy et al. [73] grouped according to so-called quasi-identifiers . Quasi-identifiers (QIDs) are identified as a set of time stamps at which the the moving object's location is assumed to be known. Each moving object has its own set of quasi-identifiers. The primary objective of grouping QIDs is to generalize the locations at the QIDs to a region. This grouping to achieve k-anonymity is done such that the induced attacker graph is symmetric. A coordinate location is converted into a one-dimensional proximity preserving, hilbert index. The top k candidates available to form a group with a moving object are computed according to their overall score. This score is defined as the sum of the absolute difference between the hilbert indices of the moving objects' locations at all time points. It proposes the following two algorithms

- Extreme union where union of all QIDs of the moving objects (MOBs) in a group is computed and then all the MOBs in the group are generalized at all QIDs in the union.

- Symmetric anonymization where the QIDs for generalization are fixed and then the group is adjusted such that the induced attacker graph is symmetric. For instance if MOB A is in group of B, then B should be included in group of A.

However, the selection of these quasi-identifiers is quite difficult in practice.

Pensa et al. [54] proposed a prefix-tree based anonymization algorithm. This algorithm guarantees k-anonymity of the published trajectories in such a way that no trajectories with support less than k will be published. Longest Common Subsequence (LCS) is used as a distance metric to measure the similarity between two trajectories. Pensa et al. [54] defined the support of a trajectory Trj as the number of trajectories containing Trj. This definition however, causes the inference route problem. Here, the manner in which the k anonymity is applied will affect the quality of the anonymization result.

Additional studies were conducted to examine trajectories that are represented by either landmarks or locations of interests. Such trajectories, however, provide primarily moving patterns. They do not provide real trajectories For example, Andrienko et al. [8] examined the various behaviors of moving objects (e.g., positions of start and end, significant turns, and significant stops) to cluster the trajectories. Monreale et al. [47] proposed a generalization approach using semantics of the trajectories. They temporally ordered sequence of important places visited by a moving object with the help of a places taxonomy. However, even though a sensitive location (e.g, an Oncology clinic) may be generalized to Clinic, there may be only one clinic at that location and hence an adversary could still infer the sensitive information. Two related works used time confusion and path confusion, respectively. The time confusion approach [35] mixes the location samples of different trajectories, and the path confusion approach crosses paths in areas in which at least two users meet. The primary issue with these two approaches is that traffic flows are no longer preserved.

Several researchers assumed that attackers have a certain amount of knowledge prior to their attack. Terrovitis and Mamoulis [63] assumed that the adversaries know the partial trajectory information of some individuals. For example, consider Octopus, a company based in HongKong [63] keeps track of the customers who use an Octopus card in day-to-day transactions. If the company publishes the customers' trajectories, it can contribute to mining movement and behavioral patterns of HongKong's residents. If, however, a customer uses the card to pay at convenient stores that belong to the same chain, the convenient store can extract its transaction history and deduce a subset of the customer's total trajectory. If this partial trajectory uniquely identifies the customer in the trajectories published by the Octopus company, then the customer's privacy is violated. The location points in the trajectories are suppressed to prevent the inference of new location points with high certainty. Similarities between the original and anonymized trajectories are used to measure the data's utility. If a point is suppressed, the distance between the point and its anonymized counterpart is equal to the maximum distance between any two points on the map. Terrovitis and Mamoulis used the partial trajectories owned by the adversaries as part of the input into their anonymization algorithm. Such usage limits not only the generality but also the feasibility of their approach. Chen et. al. [16] proposed an algorithm to publish differentially private trajectory data. This algorithm added noise to a prefix tree under Laplace transform.

Some representative related works [3, 50, 54, 63, 73] have been summarized based on their key ideas in Figure 2.2. The key ideas include the distance metric used to measure similarity between two trajectories, consideration of road network constraints, the complexity and the data's utility. None of the approaches consider the road network constraints. These approaches do, however, use a variety of distance metrics (e.g., euclidean distance, the hilbert index, the log cost metric and LCS). Data utility is measured on the basis of how much the results of a common data mining technique (e.g., clustering and range query) differ when both the original and the anonymized data sets are used. A worst case complexity analysis of these approaches is listed in Fig. 2.2. Here, n is the total number of trajectories. The complexity of the greedy clustering [3] is $O(nM)$ where M is the number of seeds used in clustering. This value is much smaller than n. Range query distortion is used to measure data utility. The same range query is applied to the original and anonymized dataset. In most instances, the distortion was below 10%.

Approaches	Key Idea	Distance Metric	Road Network Constraints	Complexity	Data Utility
O. Abul, 2008	• It exploits δ, possible location imprecision and proposes (k, δ) anonymity • Greedy clustering and space translation	Euclidean distance	No	O(nM)where M is no. of seeds in clustering	Range query distortion(below 10% in most cases)
M. Terrovitis, 2008	Suppression of points in trajectories such that adversaries cannot infer new location points not in its private database with high probability	Euclidean distance (If a point is suppressed, the distance is equal to the maximum distance between two points in the map).	No	O(nA) A is the no. of adversaries	Measured by average distance between the original and published trajectories
R.G. Pensa, 2008	• Prefix-tree based anonymization algorithm. • Guarantees k-anonymity of the published trajectories in a way that no trajectories with support less than k will be published	Longest Common Subsequence (LCS)	No	O(n^2)	Frequent patterns in original and anonymized datasets are compared and found less in anonymized one.
R. Yaravoy, 2009	• Each moving object is associated with a set of Quasi Identifiers (QIDs). • Generalization at QIDs to achieve k anonymity and symmetric attack graph	Hilbert index	No	O($mnk\alpha(nk)$) where m is no. of time points in MOD	Range query distortion (Symmetric anonymization outperforms extreme union)
M.E. Nergiz, 2009	• Clustering based generalization approach for k-anonymity(such that log cost is minimal in a cluster) • Reconstruction of representative trajectory per	Log cost metric	No	O(n^3) distance computations for clustering O(l^2) for each ERP computation	Distortion on clustering (Reported good results for a reasonable no. of clusters(e.g. up to 20)

Figure 2.2. Offline Location Privacy Preserving Approaches Analysis

Terrovitis and Mamoulis [63] used the similarity between the original and the anonymized trajectories to measure the data utility. For each trajectory, the algorithm iterates through each adversary's private databases making the complexity, $O(nA)$, where A is the total adversaries.

The complexity of prefix tree anonymization [54] is $O(n^2)$. The data utility was measured by comparing the frequent patterns in the original dataset to the frequent patterns in the anonymized dataset. Pensa et al. [54] found that the frequent patterns decreased in the anonymized dataset.

The complexity of the approach proposed by Yarovoy [73] is $O(mnk\alpha(nk))$. The complexity is the summation of disjoint sets union/find data structure with path compression's complexity, $O(nk\alpha(nk))$ and generalization's complexity $O(mn)$. Range query distortion was used to measure data utility. Yarovoy [73] found that the symmetric anonymization outperformed the extreme union.

The complexity of the approaches as discussed in [50] can be summed up from distance computation's complexity, $O(n^3)$ in hierarchical clustering and ERP computation's complexity, $O(l^2)$ using dynamic programming where l is the longest trajectory. Clustering was used to measure the data utility. The original and anonymized set of trajectories were grouped respectively using the same clustering approach. Good results were reported up to a reasonable number of clusters (e.g., 20).

None of the aforementioned approaches consider the impact of road network constraints. Hence, their anonymization results are vulnerable to attack when the malicious party either knows the road map or holds other background information. For example, if a cloaking region covers only one road, the corresponding trajectory can be easily mapped to the road. To sum up, the privacy preserving location publishing approach proposed in this dissertation is superior to existing works in terms of the following two major aspects.

- The anonymized trajectories follow road-network constraints and hence are more effective for traffic analysis.

- The anonymized trajectories prevent inference problems that have never been studied by any others before.

2.3. LOCATION PRIVACY IN BIG LOCATION DATA

The handling of big data requires a scaling up of both storage and processing power. Hadoop, an open source system which provides efficient storage and processing (using HDFS and MapReduce respectively) was employed in this study. Works on big data analysis using MapReduce are also reviewed.

2.3.1. Background in MapReduce. MapReduce is a functional programming paradigm that enables the parallel programming of large data efficiently through multiple nodes. Its programming model is built upon a distributed file system (DFS) that provides distributed storage. Programmers specify two functions: *Map* and *Reduce*. The *Map* function receives a key/value pair as input and generates intermediate key/value pairs to be processed further. The *Reduce* function merges all of the intermediate key/value pairs associated with the same (intermediate) key and then generates a final output. In a cloud computing setting, these functions are orchestrated by the Master. They are carried out by both mappers and reducers. The Master acts as the coordinator responsible for task scheduling, job management and so forth.

A Master's module (typically the data partitioner) splits the input data into a set of M blocks. These blocks will be read by M mappers through DFS I/O. The execution of map and reduce tasks is automatically distributed across all the nodes in the cluster. The *Map* function takes as input one of the M blocks (defined as a key-value pair) and produces a set of intermediate key-value pairs. The intermediate result is sorted by the keys so that all pairs with the same key will be grouped together (the shuffle phase). If the memory size is limited, an external sort can be used to handle large amounts of data at one time. The intermediate results' locations are sent to the Master. The master then notifies the reducers so they can prepare to receive the intermediate results as their input. The reducers then use Remote Procedure Call (RPC) to read the data received from the mappers. The user-defined reduce function is then applied to the sorted data; the key pairs with the same key will be reduced in some way, depending on the user-defined reduce function. Each mapper will process the data by parsing the key/value pair. It will then generate the intermediate result that is stored in the local file system. Finally, the output will be written to DFS.

2.3.2. Big Data and MapReduce. Few studies [67, 78, 91] have been focused on big location data analysis using MapReduce. Wang et al. and Zhang et al. [67, 78] represented a moving object as a point object with a location. Gedik and Liu [91] simplified a customizable k-anonymity-based solution that hides a user's identity. This method works well for both small and large datasets. This method uses databases and a group of computers to compare each piece of trajectory information with all other the rest of the data. Hence, this method becomes very slow for huge amounts of data and ends up useless in an environment that demands real time information.

Ene at al. and Zhenhua at al. [22, 40] focused applying popular clustering algorithms, such as k-means and k-median on big data using MapReduce. However, these algorithms are supervised and require multiple MapReduce jobs to accomplish which increases latency. The approach discussed in the following subsections is unsupervised and can be completed in a single MapReduce job, thus making it more efficient.

MapReduce research, thus far, has focused on providing a simple, yet powerful, interface for handling large amounts of data. This research also focuses on providing a dynamic way to handle divide and conquer techniques and optimizing parallelization. The goals of MapReduce research are to achieve high performance on large clusters of commodity PCs [92]. MapReduce technology, pioneered by Google®, is an excellent tool for clustering and simplifying data. However, it has never before been used to anonymize trajectories. The focus of this study was on using MapReduce to anonymize big trajectory data.

2.4. TRUSTWORTHINESS EVALUATION IN VEHICULAR AD-HOC NETWORKS

Existing works on information trustworthiness in Vehicular Ad-hoc Networks (VANETs) can be classified into three main categories [77]: (i) entity-oriented trust model (ii) data-centric trust model and (iii) combined trust model.

The trustworthiness of information in an entity-oriented trust model is estimated according to the message sender's. For example, Raya et al. [55] utilized a static infrastructure, such as a Certification Authority (CA), to evict malicious vehicles in VANETs. They made the assumption that most of the users in an attacker's neighborhood are honest. Doing so allowed the vehicles to trust their honest neighbors in order to evict attackers. Raya et al. [55] proposed two methods for misbehaving node revocation by the CA. The first method is known as Revocation of the Trust Component (RTC). This method deprives the misbehaving node of its cryptographic keys thus confirming that all of its messages are disregarded by all other legal nodes. RTC is not robust against a sophisticated adversary that controls the communication link between the CA and the TC. The other method is known as Misbehavior Detection System (MDS) with Local Eviction of Attackers by Voting Evaluators (LEAVE) protocol. The main principle of LEAVE is that the neighbors of the misbehaving vehicle temporarily evict it. In Gerlach et al. and Minhas et al. [27, 42] require a vehicle to build up a profile of each vehicle it comes in contact with. This vehicle evaluates the trustworthiness of its peers based on its past interactions. It then determines whether or not the information received is trustworthy. Despite their capabilities, however, entity-oriented trust models have a number of limitations. For example, VANET is a very dynamic environment and relationships among entities do not last very long. This short-lived interactions cause difficulties to collect enough evidences to trust an interacting entity. Additionally, even if an entity is trustworthy and honestly forwards a message it received, the receiver can not determine whether or not the message itself is correct.

To address limitations in entity-oriented trust models, a number of researchers have proposed that a message's content be evaluated directly in addition to validating a message sender's identity. Raya et al. [56] used Bayesian inference and Dempster-Shafer theory to evaluate the evidence received regarding an event's occurrence. Their approach relies on the availability of trust scores for the individual evidence (i.e., message) related to an event. However, the calculation of trust scores is presented as a black box, which is considered system dependent. The work discussed in this dissertation is distinguishable from the Raya et al. [56] study in several aspects. First, specific functions were designed to compute the trust score for each message rather than just a framework. Second, a more thorough set of factors is explored including similarity among message routing paths, rather than information received from directly interacting nodes [56].

The combined trust model [15, 21, 52] uses opinions gathered from various peer vehicles to determine a message's trustworthiness.This determination is used to suggest a vehicle that has been identified as trustworthy by a number of trusted peer vehicles. A vehicle's honesty value increases as the number of trusted opinions increases (the vehicle becomes more trusted). This process is an iterative process that is similar to the true fact discovery problem in Internet [19, 74], an approach used to evaluate Data Trustworthiness based on Data Provenance. However, this model has limitations similar to the entity-oriented model. This model also assumes the peer vehicles have specific methods they can use to evaluate the message content's trustworthiness. The work discussed in this dissertation actually develop a specific approach to evaluate a message's content and quantify the message's trustworthiness based on this evaluation.

3. PRIVACY-PRESERVING LOCATION PUBLISHING UNDER ROAD-NETWORK CONSTRAINTS

The challenges on how to wisely use the location data without violating each user's privacy concerns are addressed in this section. 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 3.1, user u_1's trajectory can be represented as $IABC$ and user u_4's trajectory is ABD. Many approaches [68] 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 [8]. After taking into account the privacy concerns, the goal becomes to prevent adversaries from mapping published locations to a specific individual.

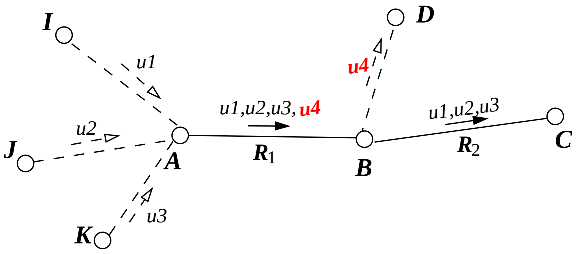


Figure 3.1. An Example of Inference-Route Problem

One may think that a trajectory resembles a conventional sequential pattern. Hence, a naturally raised question is that if it is feasible to directly employ privacy preserving data publishing approaches [7, 9, 53, 75] 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, traditional algorithms can not be used 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 [3, 8, 31, 51, 63] on anonymizing trajectories. Some work [63] considers trajectories as a sequence of landmarks, e.g., stores and museums, which ignore the paths connecting these places. Others [3, 8, 31] 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 [46]). 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 this work, the anonymization output is also trajectories on real road-network.

There are very few works that generate actual road-network-constrained trajectories as the anonymization output. The most recent one is by Pensa et al. [54], who anonymize trajectories based on k-anonymity [61]. 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 [54], 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 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 3.1 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

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

ID to Alice with probability greater than $\frac{1}{k}$ by using the above method, then there is an inference-route problem.

In this work, the problem of privacy-preserving location data publishing under the assumption that road-network data are public information is addressed. This work 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. A C-Tree (Cluster-Tree) is proposed to speed up the clustering process and develop methods to incrementally calculating error rates.

3.1. 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, specific locations and velocities are respectively replaced by road ID and moving direction; trajectories are anonymized 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.

The frequent road and inference-route problem are defined as follows.

Definition 1. Let W be a time interval, and let k be a threshold. 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. The frequency of the road is the number of moving objects on that road.

In case the trajectory dataset covers a long time frame (e.g, days, weeks or months), the time frame is divided into shorter intervals (e.g., hours) and trajectories falling into the same time interval are anonymized. 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. Two types of time dimension partitioning are supported. One is to let users define a time frame which depends on their time period of interest and the other is to divide the time frame uniformy. The unit of division chosen 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 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 3.1. Node B is an intersection of three roads. On road $\overline{AB}$, U_{AB}^+ = $\{u_1$,u_2,u_3, $u_4\}$; on road $\overline{BC}$, 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.

The methods to evaluate the quality of the anonymized dataset of trajectories are presented. Intuitively, the less difference between the anonymized dataset and the original dataset, the better quality the anonymized dataset is. Therefore, two commonly accepted metrics have been used: 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 3.1, 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.

$$E = \frac{1}{N}\sum_{i=1}^{N} E_i = \frac{1}{N}\sum_{i=1}^{N}\frac{|anonymized_{r_i} - original_{r_i}|}{original_{r_i}} \tag{3.1}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(E_i - E)^2} \tag{3.2}$$

3.2. THE APPROACH

In this section, the anonymization algorithm of this work is presented. It consists of two main steps. First, the time axis is partitioned into intervals, and records within the same interval are grouped . In each obtained sub-dataset D, the records that are associated with infrequent roads, i.e., roads with less than k objects within same time interval are removed. The obtained dataset is denoted as D'. In D', partial trajectories are constructed 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 dissertation, words “trajectory” and “partial trajectory” are interchangeable.

The second step is the core of the anonymization process. A clustering-based anonymization algorithm is proposed which guarantees that by achieving strict k-anonymity (defined in Section 3.2.1.) among partial trajectories, the anonymization result is free of the inference-route problem. Compared to traditional k-anonymization approaches, the 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.

3.2.1. 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.

First, a proper way to represent trajectories needs to be selected. Trajectories are initially represented as a sequence of timestamped locations. In the anonymized dataset, exact locations are not disclosed because detailed information increases attackers' chances to link published location to specific individuals. Instead, information about which object passing by which road is only reported. There are two options: (i) representing a trajectory by road IDs; or (ii) representing a trajectory by node IDs. As illustrated in Figure 3.2, 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, the second option is adopted and a trajectory is represented 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, the *edit distance* [64] is employed 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 into the other, where an operation is an insertion, deletion, or substitution of a node. For example, the edit distance between $Trj_1(n_5 n_2 n_3)$ and $Trj_2(n_1 n_2 n_4)$ is 4, while the distance between Trj_2 and $Trj_3(n_1 n_2 n_6)$ is 2.

The clustering-based anonymization algorithm is presented in this section. An outline is given in Figure 3.3. First, same trajectories are grouped and the trajectory’s*support* is counted. 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'. The support of trajectory Trj is 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, it is compared with existing clusters. If there exists a suitable cluster, the new trajectory is inserted into that cluster and update the cluster’s information. Otherwise, a new cluster will be created for Trj. At the end of

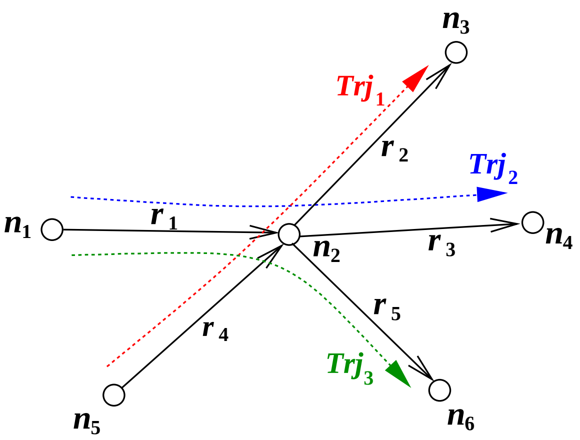


Figure 3.2. Trajectory Representation

```
Clustering-based Anonymization ($TRJ$, $k$)
Input: $TRJ$ is a set of trajectories to be $k$-anonymized
Group same trajectories and form $TRJ'$
Sort trajectories in $TRJ'$ in a descending order of supports
for each $Trj$ in $TRJ'$ do
    if $Trj.support \geq k$ then
        create a new cluster for $Trj$
    else
        check existing clusters
        if Find_Cluster($Trj$,$C$) then
            insert $Trj$ to cluster $C$
            Select_Representative_Trajectory($C$,$Trj_r$)
            update $C$'s error rate
            update $C - tree$
        else
            create a new cluster for $Trj$
/* Clustering Adjustment Phase */
for each cluster $C$
    if $C.Total\_TRJ \geq \rho_a$ then set $C.Total\_TRJ = k$
    else remove $C$
/* Data Publishing */
Translate representative trajectories into output format
```

Figure 3.3. An Outline of Clustering-based Anonymization Algorithm

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, it is directly removed. Otherwise, dummy trajectories are added 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 3.3. Finally, representative trajectories together with their supports are translated into output format, which contains object anonymous IDs, road names, and objects' moving directions. For example, the following intermediate result is obtained after anonymizing the trajectories shown in Figure 3.1: $u'_1(ABC)$, $u'_2(ABC)$, $u'_3(ABC)$ 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.

3.2.2. Finding Candidate Clusters. Figure 3.4 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, an in-memory index structure, the C-tree (Cluster-tree) is employed 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 roads are modeled 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 3.2 can be represented as $\{r_2, r_4\}$

```
Find_Cluster ($Trj$,$C$)
Input: $Trj$ is a trajectory
Output: $C$ is a cluster
$NODE \leftarrow \{$C-tree.root$\}$
while ($NODE$ is not empty) do
    for each node $N$ in $NODE$ do
        for each entry $en$ in $N$ do
            if $Sim_c(Trj, en.RID) > \rho_t$ then
                if $N$ is not a leaf node then
                    add $en$'s child node to $NODE$
                else add $en$'s cluster to candidate list $L_c$
for all clusters in $L_c$ do
    find clusters with smallest $E^c$ regarding $Trj$
    if $E^c < \rho_c$ then
        return the cluster found
```

Figure 3.4. Algorithm of Finding Clusters

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

$$Sim_c(Trj, RID) = \frac{|S(Trj) \cap RID|}{|S(Trj)|} \tag{3.3}$$

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, the child node of this entry is visited. This process is repeated until all entries in the leaf nodes with Sim_c above the threshold are found. 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 node N_3 need not be visited. The child node N_2 pointed by the first entry is visited. 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, the edit distance between their representative trajectories and the new trajectory Trj is calculated. Based on the edit distance, a local error E^c (defined in Section 3.2.4.) is then computed and the candidate cluster with the smallest E^c is selected. Only when E^c is lower than a threshold ρ_c (defined in Section 3.3.), Trj will be

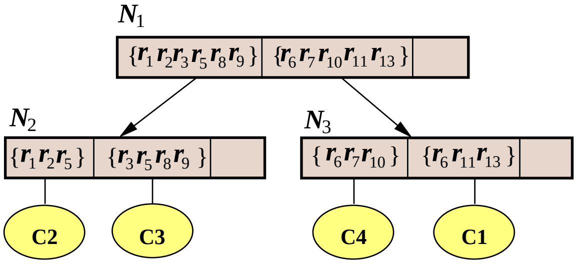


Figure 3.5. An Example C-tree

inserted into the corresponding candidate cluster. Otherwise, a new cluster will be created for Trj.

When actually adding Trj to a cluster, both the representative trajectory and the corresponding entries in the C-tree need to be updated. The algorithm for computing the representative trajectory is presented in Section 3.2.3. After the representative trajectory is determined, the node in the C-tree is checked if it needs to be updated with respect to current cluster. If current cluster contains road IDs which are not included in the road ID list of the corresponding C-tree entry, the new road IDs are appended 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 3.5 and suppose that a new trajectory that consists of roads r_2, r_8 and r_9 will be inserted into cluster C_3. A check is done to the road list of C_3's entry in the C-tree, which is $\{r_3 r_5 r_8 r_9\}$ and does not contain r_2. r_2 is then added to the road list. Now the second entry in N_2 becomes $\{r_2 r_3 r_5 r_8 r_9\}$. Next, its parent entry, the first entry in N_1 is checked. 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, it requires 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. The new entry is inserted 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 it is evenly split into two nodes. In particular, an entry is randomly selected as seed. Then Sim_c (Equation3.3) is computed 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, the entry pointing to N is updated. 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, a new entry which includes road IDs needs to be inserted 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.

3.2.3. 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, the following algorithm is designed.

In a cluster, the trajectory with the highest support is found and then trimmed from both ends to obtain the final representative trajectory. It is illustrated using example in Figure 3.6.

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, it is further looked at. The error rate E is computed by

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

Figure 3.6. An Example of Selecting Representative Trajectory

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. In this example, if Trj_1 is used as the representative trajectory, the error rate will be E = 58%.

$$E = (E_{n_1 n_2} + E_{n_2 n_4} + E_{n_4 n_7} + E_{n_7 n_8} + E_{n_8 n_9})/5$$

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

Observe that $E_{n_8 n_9}$ is higher than 100%. If the road $n_8 n_9$ 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, the nodes can not be arbitrarily removed from a trajectory. The strategy is to remove nodes starting from both ends of the selected trajectory. Also, too many nodes should not be removed, which otherwise leads to poor pattern preservation. To reach the balance, only removing the nodes with error rate above certain threshold is considered. In this case, the threshold is set 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 such a road cannot be found at either end of the trajectory. The final representative trajectory for the example case is $n_1 n_2 n_4 n_7 n_8$. The algorithm is summarized in Figure 3.7.

3.2.4. Definitions of Local Error E^c. In the following discussion, C is used 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 3.4, where clusters C_1, ..., C_m are clusters containing road r_i.

$$anonymized_{r_i} = \sum_{j=1}^{m} (anonymized^{c_j}_{r_i}) \tag{3.4}$$

```
**Select_Representative_Trajectory** ($C$,$Trj_r$)
Input: $C$ is a cluster
Output: $Trj_r$ is the representative trajectory
support($Trj_r$)← 0
for each $Trj$ in $C$ do
    if support($Trj$) >support($Trj_r$) then
        $Trj_r \leftarrow Trj$
        support($Trj_r$) ← support($Trj$)
$i \leftarrow 1$; $j \leftarrow length(Trj_r)$-1
continue ← 1
while ($i < j$ and $continue$) do
    continue ← 0
    if original($r_i$) <support($Trj_r$)-original($r_i$) then
        $i \leftarrow i + 1$; continue← 1
    if original($r_j$) <support($Trj_r$)-original($r_j$) then
        $j \leftarrow j - 1$; continue← 1
$Trj_r$ ←($r_i$...$r_j$)
return $Trj_r$
```

Figure 3.7. Algorithm of Selecting Representative Trajectory

Given a new trajectory Trj_{new}, E^c is computed by assuming that Trj_{new} has been inserted into cluster C. The new cluster with Trj_{new} is denoted as C' and is assumed 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 3.5, 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, two values, $trans_{r_i}$ and $change_{r_i}$ are calculated. 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., $change_{r_i}$= ($|anonymized^{c'}_{r_i} - anonymized^{c}_{r_i}|$).

$$E^c = \frac{1}{|R|}\sum_{r_i \in R} E^c_{r_i} = \frac{1}{|R|}\sum_{r_i \in R} (change_{r_i} - trans_{r_i})^2 \tag{3.5}$$

For better understanding of Equation 3.5, the calculation is illustrated 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. Now E^c is computed upon the insertion of a new trajectory $Trj_3(n_2n_4n_7n_8)$ with $support(Trj_3) = 6$ into the cluster C. Table 3.1 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 3.1 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 it can be observed 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.

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, a great number of comparison can be avoided 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 3.6, where ED denote the edit distance.

$$E^c = \frac{1}{|R|} ED(Trj_r, Trj_{new}) \cdot support(Trj)^2 \tag{3.6}$$

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, E^c can be computed as follows, which yields the same result as using Equation 3.5: $E^c = \frac{1}{5}(6^2) = 7.2$

Table 3.1. An Example of E^c Calculation

Road	$anonymized^c$	$anonymized^{c'}$	$trans$	$change$
$n_1 n_2$	15	15+6=21	0	6
$n_2 n_4$	15	15+6=21	6	6
$n_4 n_7$	15	15+6=21	6	6
$n_7 n_8$	15	15+6=21	6	6
$n_8 n_9$	0	0	0	0

3.3. SELECTION OF THRESHOLD

The threshold selection is a critical task which affects clustering speed and anonymization accuracy. This subsection discusses 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, the threshold ρ_a is set as follows.

$$\rho_a = \frac{k}{2} \tag{3.7}$$

The basic idea of Equation 3.7 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, the threshold ρ_c is defined as shown in Equation 3.8.

$$\rho_c = \left(\frac{k}{2}\right)^2 \tag{3.8}$$

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 3.5. 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, it can be observed 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.

3.3.1. Strict k-anonymity.

In this section, the notion of *strict k-anonymity* is defined. 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. Trj satisfies strict k-anonymity if Support(Trj) is no less than k.

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

Theorem 1. *Trajectories that satisfy strict k-anonymity do not contain any inference-route.*

Proof. *It is proved by contradiction. It is assumed that the 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 the anonymization result which only contain trajectories with support no less than k. Therefore, it is concluded that the approach does not have any inference-route problem. □

3.3.2. Complexity Analysis. In this section, the time and space complexity of the approach are analyzed. In what follows, n is used to denote the total number of original trajectories, and l is used to denote the maximum number of roads in a trajectory in the raw dataset D.

First, the time complexity is analyzed. The 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, the road segments contained in all the trajectories need to be scanned 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, a search is done 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, obtained is the total time complexity of the clustering-based anonymization, which is $O(log(n)) + O(n)$.

Finally, the total time complexity of the 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 the approach is linear to the total number of trajectories, which is also confirmed by the following experimental results.

As for the space complexity, the 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 $O(n) + O(f^{log(n)})$.

3.4. EXPERIMENTAL STUDY

In this experimental study, the two approaches: Clustering-Based Anonymization (CBA) [39] and Improved Clustering-Based Anonymization (ICBA) are compared. CBA used E (Equation 3.1) during the clustering while ICBA used the new metric E^c (Equation 3.5). Then, the effect of the C-tree adopted by ICBA is studied. After that, ICBA is compared with the latest related work (denoted as Prefix [54]) by testing the original source code provided by the authors of [54]. Both synthetic and map-based datasets are used and a variety of parameters including the data size, data distribution, average trajectory length and value of k are varied.

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, the generator by Brinkhoff [12] is used. 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 the data set containing 50,000 moving objects is used as the default setting. The parameters used in the experiments are summarized in Table 3.2, 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 3.1; (iii) standard deviation as given by Equation 3.2; (iv) number of inference-routes in the anonymization result; (v) number of frequent

Table 3.2. Parameters and Their Settings

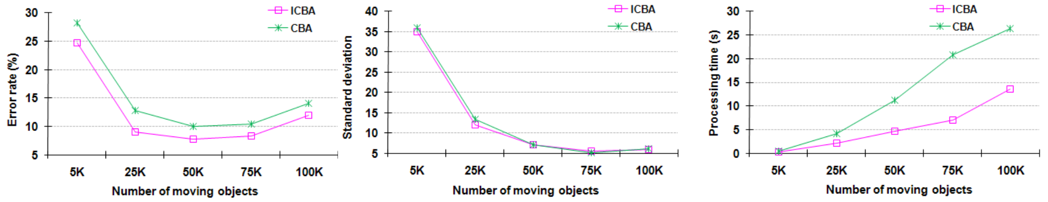
Synthetic Dataset	
Parameter	**Setting**
k	10,20,**30**,40,50
Number of moving objects	5K, 25K, **50K**, 75K, 100K
Average trajectory length (km)	20, 30, 40, 50, 60

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)

patterns after anonymization. All the experiments were run on a PC with 2.6G Pentium IV CPU and 3GB RAM.

3.4.1. Anatomy of Our Approaches. The CBA and ICBA approaches are compared and the results are reported in this section. The effect of the C-tree is also observed.

3.4.1.1 CBA vs. ICBA. The first round of experiments compares the performance of the two approaches: CBA and ICBA, by using synthetic datasets. Figure 3.8(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 cases, while ICBA benefits from the optimal threshold selection (Equation 3.8) as well as the newly defined metric E^c (Equation 3.5). Figure 3.8(b) reports the standard deviation

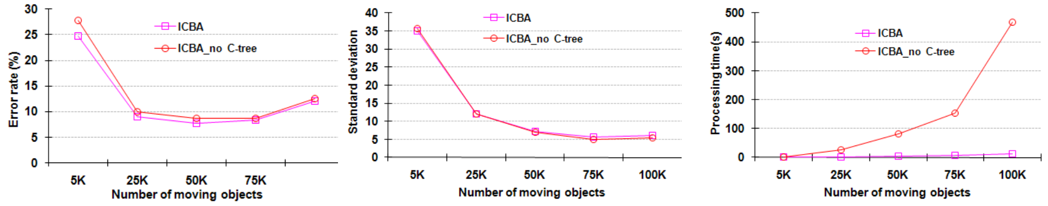


(a) Error rate (b) Standard deviation (c) Processing time

Figure 3.8. CBA vs. ICBA

where it can be seen that ICBA performs similarly to CBA. Figure 3.8(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, only ICBA will be considered.

3.4.1.2 Effect of the C-tree. In this set of experiments, the effect of the C-tree is studied 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 3.9(a) and (b) report the average error rate and standard deviation with respect to the two versions, and Figure 3.9(c) compares their processing time. It can be observed that the use of C-tree does not affect the accuracy of the anonymization result, but significantly reduces processing time (more

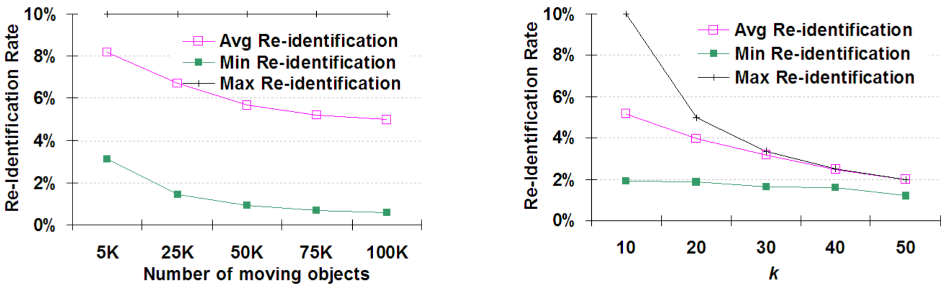


(a) Error rate (b) Standard deviation (c) Processing time

Figure 3.9. Effect of the C-tree

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.

3.4.1.3 Measuring the probability of re-identification. The probability of re-identification of a user is also analyzed in the 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 3.3.1., the approach guarantees k-anonymity which means the re-identification probability will not be higher than $\frac{1}{k}$. In the actual experiments, a much lower re-identification rate is observed as reported in Figure 3.10. In particular, the maximum, the average and minimum probability of re-identification rate of all the clusters are recorded. 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

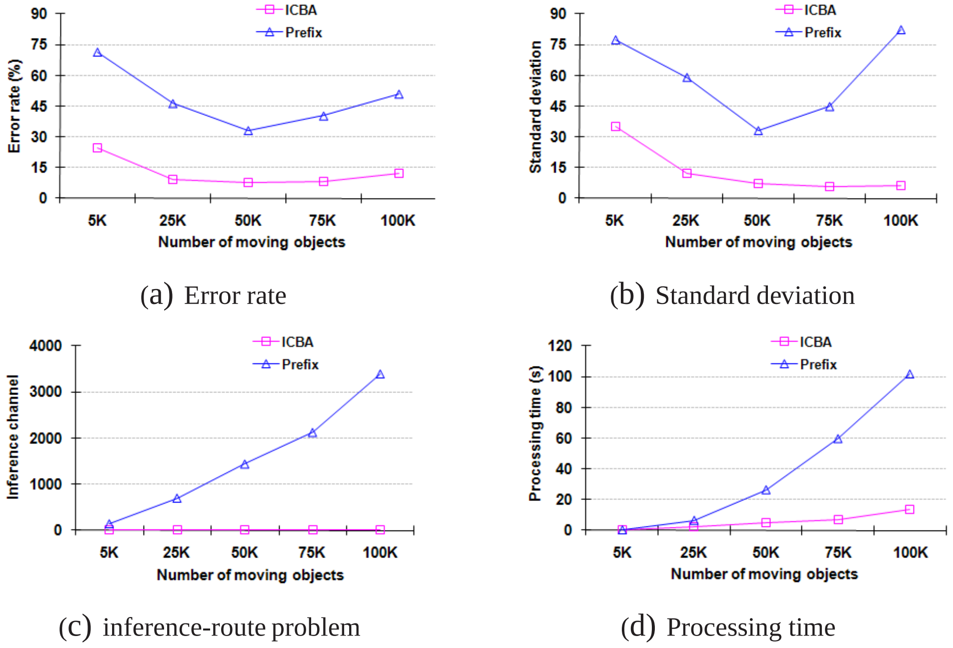
Figure 3.10. Probability of Re-identification

3.4.2. Experimental Results in Synthetic Datasets. The experiments are conducted using synthetic datasets and results are reported in this section.

3.4.2.1 Effect of data sizes. The performance of ICBA is now compared with Prefix approach by varying the number of moving objects (i.e. number of trajectories) from 5K to 100K. Figure 3.11(a) shows the average error rate of the anonymization results obtained from ICBA and Prefix. It can be observed 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. Also measured is the standard deviation of the anonymization results obtained from two approaches . As shown in Figure 3.11(b), the anonymization result generated by ICBA has much lower standard deviation than that by Prefix, which indicates that the anonymization result on each road has similarly good quality.

Figure 3.11(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 the 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.

The processing time of both approaches is compared. As shown in Figure 3.11(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



(a) Error rate

(b) Standard deviation

(c) inference-route problem

(d) Processing time

Figure 3.11. Effect of Data Size

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.

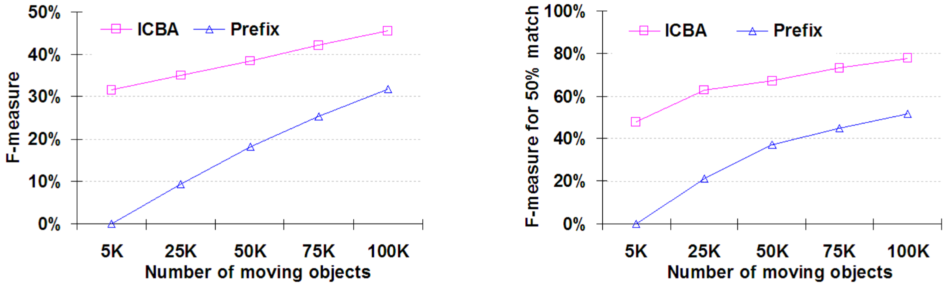
3.4.2.2 Preservation of frequent patterns. The quality of anonymization results is evaluated by comparing the anonymized trajectories obtained from ICBA and Prefix with the frequent patterns discovered from original datasets using the traditional data mining tool (i.e., PADS software [76]) as reported in Figure 3.12. When using PADS, each transaction is corresponding to an original trajectory. Each item is corresponding to a road ID in the trajectory. The anonymization parameter k is used 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, the widely adopted F-measure is used 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.

$$F(P_r, P_a) = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.9}$$

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

Figure 3.12(a) reports the F-measure values of the Prefix approach and the 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. 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, how many trajectories that partially match the data mining results is also evaluated. For this, the anonymized trajectories that have at least 50%



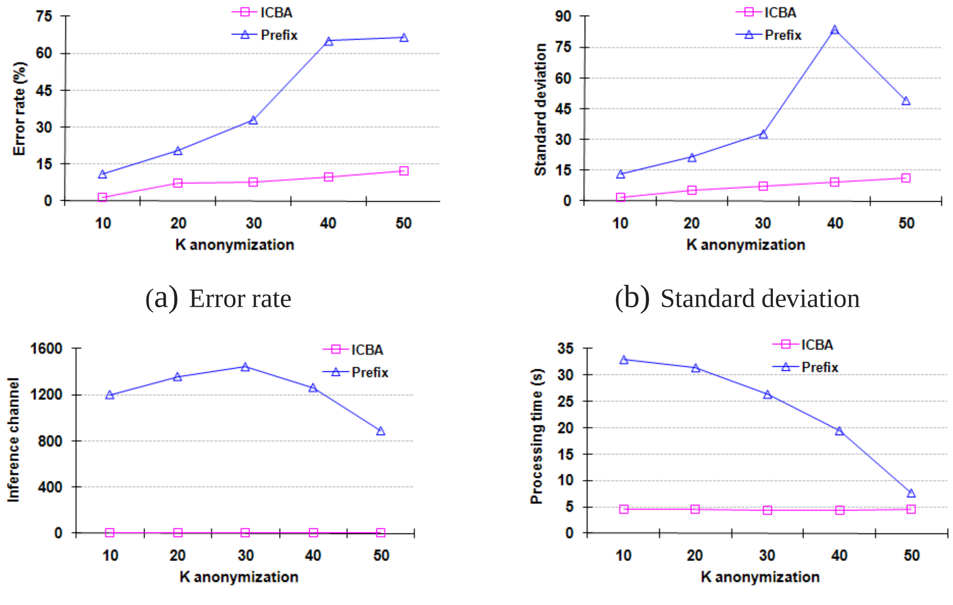
(a) Exact Match

(b) Partial Match

Figure 3.12. F-measure

road segments matching a frequent pattern in the original data mining results are recorded, and added to N_m for computing the F-measure. Figure 3.12(b) shows the results. From this figure, it can be seen that the F-measure values have been almost doubled compared to that in Figure 3.12(a). This indicates that the anonymization results preserve partial frequent pattern information very well.

3.4.2.3 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 3.13(a), the error rate increases drastically with k by using the Prefix algorithm, while k has only minor effect on the 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 3.13(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 3.13(c). Regarding processing time (in Figure 3.13(d)), ICBA has



(a) Error rate

(b) Standard deviation

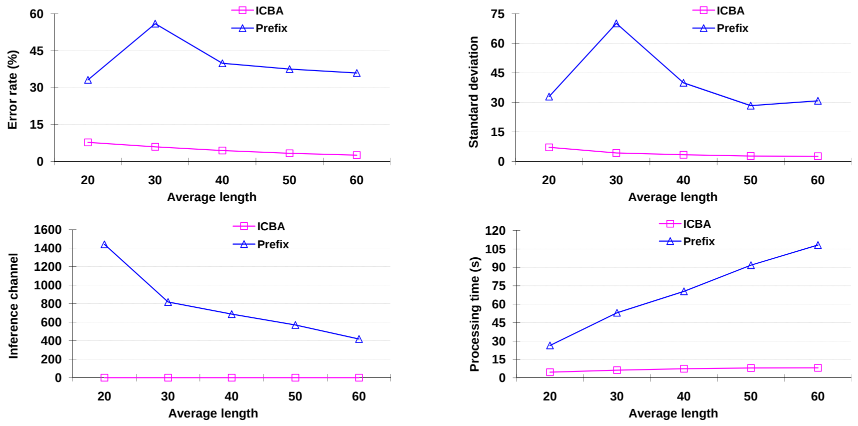
(c) inference-route problem

(d) Processing time

Figure 3.13. Varying Parameter k

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.

3.4.2.4 Effect of the average trajectory length. The effect of the average length of the trajectory in terms of number of roads is now evaluated. The length is determined by two factors: the length of time interval being considered and object moving speed. As shown in Figure 3.14(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 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.



(c) inference-route problem

(d) Processing time

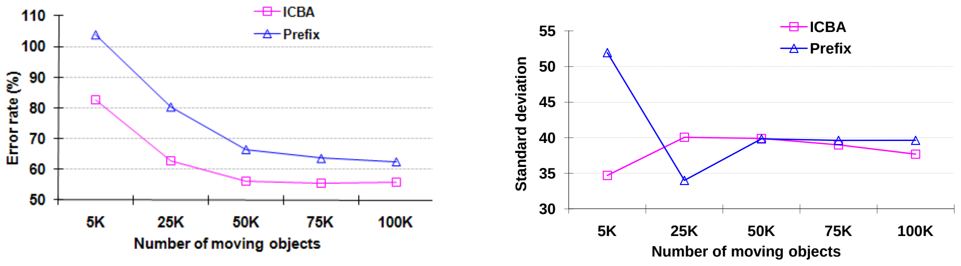
Figure 3.14. Varying Average Length of the Trajectory

As for the inference-route problem (Figure 3.14(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 3.14(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.

3.4.3. Experimental Results in Map-based Datasets. Ihe performance of ICBA and Prefix is evaluated by using datasets generated based on real road maps using the generator in [12]. The same four aspects are examined: variation of data sizes, frequent patterns, value of k and average trajectory length, as that in synthetic datasets. In addition, the effect of data distribution is also studied by using different road maps.

3.4.3.1 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 3.15 and Figure 3.16, ICBA consistently outperforms Prefix in terms of both effectiveness and efficiency.

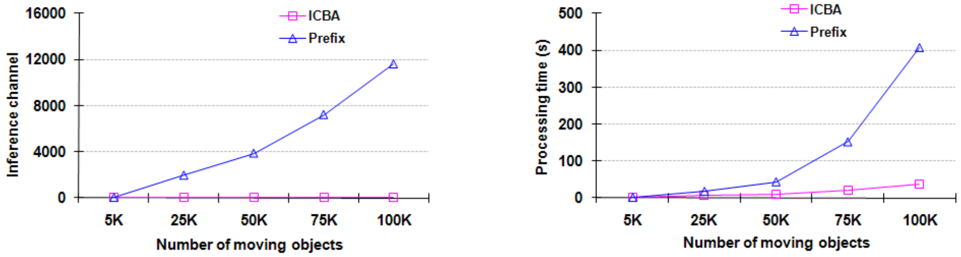


(a) Error rate

(b) Standard deviation

Figure 3.15. Effect of Data Sizes (Real Road-network)

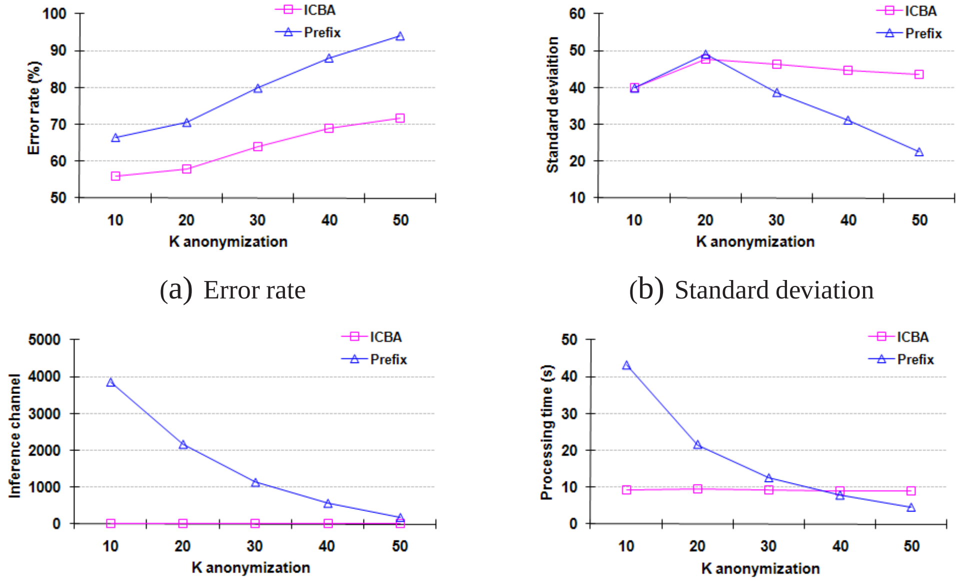
The reason is similar to that explained when evaluating synthetic datasets. In addition, both approaches have high error rates when the number of objects (i.e., trajectories) is small and the error rates go down with the increase of objects. This is because in the same road map, fewer objects result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



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

Figure 3.16. Effect of Data Sizes (Real Road-network)

3.4.3.2 Effect of parameter k. Figure 3.17 shows the performance of ICBA and Prefix when varying k from 10 to 50. From the figure, following observations can be made. 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 is lowered, the total number of nodes with inference-route problems is reduced and processing time is shorten.



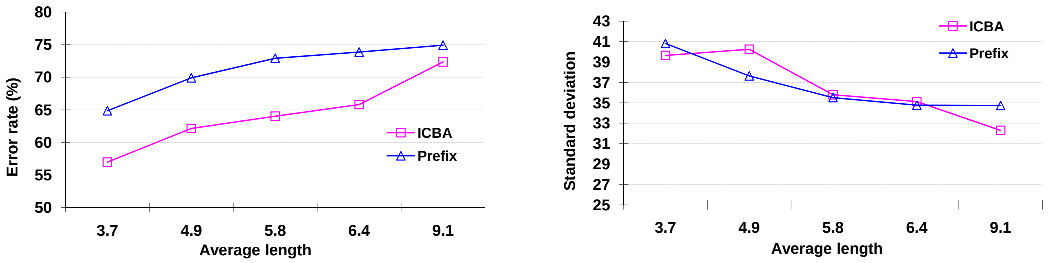
(a) Error rate (b) Standard deviation

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

Figure 3.17. Effect of Parameter k (Real Road-network)

3.4.3.3 Effect of average trajectory length. This set of experiments evaluates the effect of average trajectory length. As shown in Figures 3.18 and 3.19, ICBA again outperforms Prefix in general. It is also observed that the error rate increases for both approaches when the length of trajectory becomes longer.

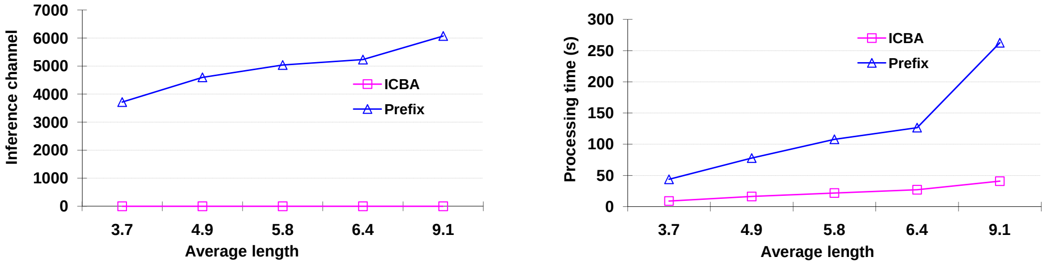
The reason is similar to that for the case with a larger k in the previous experiments. That is that the reduced number of frequent trajectory patterns with the growth of trajectory length, in turn increases the impact of trajectory modification during the anonymization



(a) Error rate (b) Standard deviation

Figure 3.18. Effect of Average Length of Trajectory (Real Road-network)

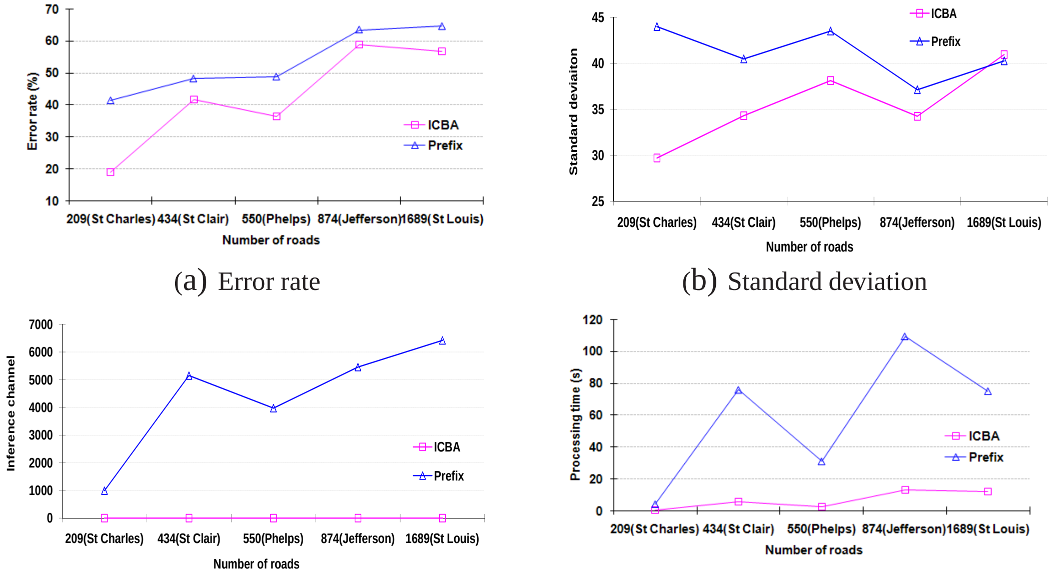
process. Moreover, with the increase of trajectory length, Prefix suffers more from the inference-route problem. The possible reason is that in the real road-network, the number of roads connected by an intersection is usually small (e.g., two to four). This increases the chance of having nodes with 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.



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

Figure 3.19. Effect of Average Length of Trajectory (Real Road-network)

3.4.3.4 Effect of data distribution. At the end, the effect of the data distribution is studied 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 3.20. 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 ICBA has relatively stable and much faster processing speed. In a summary, the result demonstrates that ICBA has better topography independency compared to Prefix.



(a) Error rate

(b) Standard deviation

(c) inference-route problem

(d) Processing time

Figure 3.20. Effect of Data Distribution

3.5. SUMMARY

Privacy preserving location data publishing has received increasing interest nowadays. In this section, this newly emerging problem is addressed by taking into account an important factor, the road network constraint, which has been overlooked by many existing works. A new privacy problem (i.e. the inference-route problem) was identified and defined. An efficient and effective clustering-based anonymization algorithm was proposed. It was proved that the 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, the following major aspects were taken into account: calculation of representative trajectories, definition and employment of local error rates, and selection of threshold used at different stages of anonymization. An extensive experimental study was conducted on both synthetic datasets and real datasets. The results demonstrated the superiority of the approach compared to other works.

4. PRIVACY-PRESERVING LOCATION PUBLISHING IN BIG TRAJECTORY DATASETS

As aforementioned, the number of LBS users is increasing fast and the amount of location data collected by the LBS service providers is also growing rapidly. In this section, the scalability issue is tackled in publishing location data with privacy preservation.

4.1. THE APPROACH

The privacy-preserving location publishing technique (in Section 3) is extended to a distributed version by leveraging MapReduce technology. For easy understanding, the key ideas are illustrated using a simple example as follows.

Suppose that a map and trajectory data from St. Louis, MO are used. For simplicity, the map is divided into four areas, *NW, SW, SE, NE* denoted as *A1, A2, A3, A4* respectively. Let k = 2 and the trajectories in the data set be u_1, u_2, ..., u_9. These trajectories come from a database controlled data center to the Master machine for the Map Reduce environment. Suppose these three trajectories were included in part of the data:

T_1 = $\{u_1$, u_3, u_4, $u_7\}$

T_2 = $\{u_1$, u_3, $u_7\}$

T_3 = $\{u_2$, u_5, u_6, $u_8\}$

The trajectories would be sent to one or more mappers, with these trajectories included, and the mappers would output key,value pairs that map each trajectory to an area.

(A1, T_1), (A1, T_2), (A2, T_3)

This output is given to one or more reducers which will cluster trajectories according to area.

(A1, T_1 T_2), (A2,T_3)

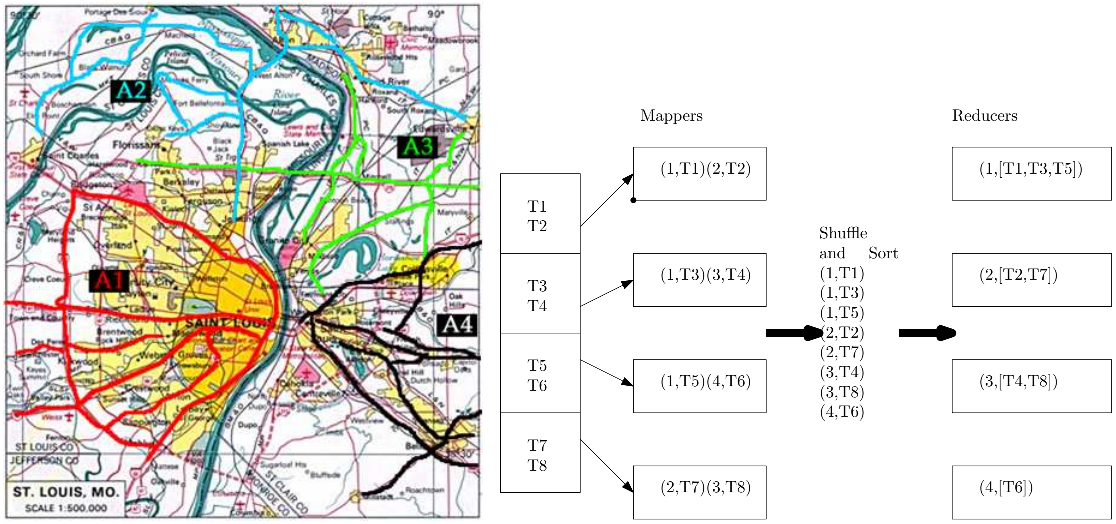
Additionally, the clusters are now anonymized at the reducer at the same time. For A2, it can be seen that there is only one trajectory in the cluster. The threshold to guarantee k-anonymity is 2. The algorithm will attempt to add the trajectory to another cluster, and if it cannot find one, then it will be removed from the published data. In area one, the threshold for the number of trajectories are met. However, if it is left as it is, then there will be an inference problem. While both trajectories are very similar, A1 includes u_4 while A2 does not. Again, the algorithm would try to find a better match for one or both trajectories. However, if they are this similar, then u_4 is removed from T_1 for publishing and now there are k exact same trajectories that meets the anonymization requirements.

Definition 5. Let (V, E) represent a road-network where V is a set of nodes or intersections and E represents the edges or roads. A road division, RD is a part of the road-network and can be represented as (V_{rd}, E_{rd}) where V_{rd} represents the vertices and E_{rd} represents the edges in road division RD. A road division also has a unique identification number, ID.

The MapReduce programming model is adopted for publication of big location data with privacy preservation. This model efficiently parallelizes the computations for such publication. A computation is divided into a map and reduce function. Each mapper gets a chunk of input object trajectories. It maps each trajectory to a suitable reducer. Each reducer gets its share of the object trajectories, decided by the mappers for clustering and anonymization. For a given trajectory in a reducer, a suitable cluster is found among the clusters in the reducer. The reducers do not share the cluster information. Therefore it is paramount for the mapper to group similar trajectories to the same reducer. Otherwise many roads will end up getting trimmed as infrequent in each reducer, increasing anonymization error.

The road map is divided into road divisions, defined in Definition 5 such that trajectories in one road division are similar as explained in Subsection 4.1.1. Mappers share the road divisions data. A mapper maps a trajectory to the road division it closely matches. Trajectories in one road division goes to the same reducer. Therefore two similar trajectories in different mappers gets mapped to the same reducer. This reduces the probability of a trajectory being removed as infrequent in each reducer though it may be frequent. This further assists in efficient clustering for the anonymization algorithm.

The MapReduce architecture in this approach is illustrated in Figure 4.1. There are eight input trajectories $T1,\ T2...,\ T8$. Each mapper has the road divisions data $R1,\ R2,\ R3,\ R4$ as in Figure 4.1(a).



(a) Road Divisions

(b) MapReduce

Figure 4.1. MapReduce Architecture

It maps each trajectory in its input data, a portion of the total input data to a $< key,\ value >$ pair. In the $< key,\ value >$ pair, key is the suitable road division and $value$ is the trajectory. In the shuffle and sort phase, all the $< key,\ value >$ pairs belonging to a reducer gets grouped together and sent to the corresponding reducers. The reducer performs anonymization on its input trajectories. The road network division algorithm, the map phase and the reduce phase are further explained in the following subsections.

4.1.1. Road-Network Division Using Hot Spots, Depth-First Traversal. In this subsection, the approach for road network division is described. It uses hot spots and depth-first traversal of the road network. Hot spots, HS, the frequent nodes or intersections given a sample trajectory data are extracted. The frequency of the nodes is counted in the sample trajectory data and sorted in descending order. The top $|HS| >> RNO$ nodes comprises the hot spots where RNO is total number of reducers. The trajectories tend to populate around hot spots. Therefore the idea is to expand the hot spots using the road network in depth first manner to form a road division. The expansion around hot spots and depth first traversal ensure that popular routes are covered in road division formation. It also divides the trajectories fairly among road divisions.

The road division formation is further explained. The hot spot, hs_{max} with maximum frequency is used as the starting node and a depth first traversal of the road network is performed. The following approaches are used as a stopping criteria of the traversal.

- The depth of each traversal path from the hot spot exceeds the average number of nodes per the sample trajectory data.

- The total distance of each traversal from the hot spot exceeds the average road distance per the sample trajectory data.

When each traversal path from the hot spot satisfies the stopping criteria, a road division is formed. The road division is represented as a network of all the traversed nodes in the region. Then the next unvisited hot spot is used as the new starting node for road division formation. This process is repeated until the total number of road divisions formed will be equal to the number of reducers.

The depth first traversal using hot spots outputs a total NO_{rd} number of road divisions. However there may be unvisited nodes which are not yet included in any of the formed road divisions. For such unvisited nodes, its neighboring nodes are checked. The neighboring node with the smallest depth from the hot spot of the road division is found. The unvisited node is placed in the road division of this neighboring node as described in Figure 4.2. This process is continued until there remain no unvisited nodes. Figure 4.3 outlines the detail procedure for road-network division.

4.1.2. Map Phase.

Each mapper has the road divisions data. The mapper decides on the road division for mapping each incoming trajectory. The mapper computes score for each road division based on the number of nodes of the trajectory that the road division contains. The mapper then finds the road division with the highest score, RD_{hscore}. If the highest score exceeds 80% of the total nodes in the trajectory, the mapper outputs a $< key, value >$ pair as $< id\ of\ RD_{hscore}, trajectory >$. If not, the mapper outputs $< residue, trajectory >$, $residue$ is the reducer reserved for trajectories which do not fit in any of the road divisions. Another approach used to find the best matched road division is to divide the trajectory into partial trajectories. All the points in one partial trajectory

```
FindAreaForUnvisitedNodes($V$, $Regions$)
Input: List of Road Divisions, $Regions$; road-network ($V$, $E$)
$loop \leftarrow true$
while $loop = true$ do
    $loop \leftarrow false$
    for each $node$ in $V$ do
        if $node.areaData$ is empty
            for each $node_{neigh}$ in $node.Nbrs$ do
                if $node_{neigh}.areaData$ is not empty
                    for each key in $node_{neigh}.areaData$ do
                        if key in $node.areaData$
                            update depth of $node.areaData.get(key).area$ with the smaller depth
                        else add $node_{neigh}.areaData.get(key)$ to $node.areaData.get(key)$
            if $node.areaData$ is not empty
                find area with the smallest depth in $node.areaData$, $area$
                add $node$ to $area$
            else $loop \leftarrow true$
```

Figure 4.2. Algorithm for Finding Area of Unvisited Nodes

```
**RoadMapDivision**($HS$, ($V$, $E$), $RNO$ )
Input: List of hot spots, $HS$; road-network ($V$, $E$); number of reducers, $RNO$
Output: List of road map divisions, $Regions$
$index_{HS} \leftarrow 0$
$totalArea \leftarrow RNO$
$startFlag \leftarrow true$
while $startFlag = true$ do
    $node \leftarrow HS[index_{HS}]$
    if $node.areaData$ is not empty
        define list of nodes $DS$
        define region $r$
        add $node$ to region $r$
        add $node$ to $DS$ at index $0$
        $node.depth \leftarrow 1$
        add $r$, $node.depth$ to $node.areaData$ with key $r.id$
        while $DS$ is not empty do
            $firstnode \leftarrow DS[0]$
            remove element of DS at index $0$
            for each $node_{neigh}$ in $firstnode.Nbrs$
                if $node_{neigh}.areaData$ does not have key $r.id$
                    if $node_{neigh}$ does not voilate the stopping criteria
                        $node_{neigh}.depth \leftarrow firstnode.depth$ + 1
                        add $node_{neigh}$ to region $r$
                        add $r$, $node_{neigh}.depth$ to $node_{neigh}.areaData$ with key $r.id$
                        add $node_{neigh}$ to $DS$ at index $0$
        add $r$ to $Regions$
    else $totalArea \leftarrow totalArea$ + 1
    if $index_{HS} >= HS.size() \,||\, index_{HS} >= totalArea$ do
        $startFlag \leftarrow false$
FindAreaForUnvisitedNodes($V$, $Regions$)
return $Regions$
```

Figure 4.3. Road Map Division Using Depth-first Traversal

belong to a single road division and the partial trajectories are the longest that can be mapped in that road division. The map phase is outlined in detail in Figure 4.4.

4.1.3. Reduce Phase.

Figure 4.5 explains the reduce phase in detail. All the trajectories mapped to the same map division are processed in the same reducer.

Map (key, $value$)
Input: Object trajectory, $Traj_{map}$, List of road divisions, $Regions$
Output: (key, $value$)

```
$NODE \leftarrow Traj_{map}.path$
for each $node$ in $NODE$ do
    for each $region$ in $Regions$ do
        if $node$ is in $region$ then
            score($region$) $\leftarrow$ score($region$) + 1
find the region with highest score, $region_{high}$
if score($region_{high}$)$> 0.8\%$ of total nodes in $Traj_{map}$ then
    return ($region_{high}.id$, $Traj_{map}$)
else return ($residue$, $Traj_{map}$)
```

Figure 4.4. Map $(key, value)$

Reduce (key, $value$)
Input: Object trajectory list, $Traj_{red}$; ID of road division,$region_{id}$; k
Output: (key, $value$)

```
$AnonymizedTraj_{red} =$ Clustering-based Anonymization($Traj_{red}$, $k$)
return ($region_{id}$, $AnonymizedTraj_{red}$)
```

Figure 4.5. Reduce $(key, value)$

The anonymization algorithm is performed on the trajectories in each reducer and k-anonymized trajectories are obtained as described in Figure 3.3. Here, the anonymization algorithm is the same as that presented in Section 3.

4.2. EXPERIMENTAL STUDY

In this section, the experimental settings are presented. A comparative study of the MapReduce-based trajectory anonymization and the centralized approach is also reported.

4.2.1. Experimental Settings. Two MapReduce-based anonymization approaches were implemented using different stopping criteria: (1) an approach with average number of nodes per trajectory as the stopping criteria (MRAN) and (2) an approach with average road distance per trajectory (MRARD) as the stopping criteria. The MapReduce-based anonymization approaches are compared with the centralized ICBA algorithm in terms of the following two error metrics:

$$Precision = \frac{Pairs\_of\_Matching\_Traj}{Traj\_Output\_By\_MapReduce} \tag{4.1}$$

$$Recall = \frac{Pairs\_Matching\_Traj}{Pairs\_Matching\_Traj + Missing\_Traj} \tag{4.2}$$

In the above two equations, the matching trajectories are computed by comparing the anonymized trajectories obtained from the MapReduce approach and that from the centralized approach. Specifically, for each anonymized trajectory obtained by the MapReduce approach, the most similar trajectory in the centralized approach is obtained, i.e., the trajectory with the largest number of common nodes. If the identified pair of similar trajectories share more than $w\%$ of common nodes, these two trajectories are considered as a pair of matching trajectories. Then, each pair of identified matching trajectories will be removed from their datasets when searching for the next pair of matching trajectories. In the following experiments, w is set to 80. In Equation 4.2, the "$Missing\_Trj$" refers to the number of anonymized trajectories in the centralized approach that cannot find a matching trajectory in the results of the MapReduce. In a summary, both precision and recall has a value ranging from 0 and 1. The precision metric measures the amount of the false positives in the MapReduce approach while the recall metric measures the amount of the false negatives in the MapReduce approach. The higher the precision and the recall, the better the accuracy of the MapReduce approach.

The implementation was performed in Amazon Elastic MapReduce (Amazon EMR) using Hadoop, an open source framework, across a cluster of 5 Amazon EC2 m3.2xlarge instances. Each m3.2xlarge instance is configured to have High Frequency Intel Xeon E5-2670 processor and 30GB of memory.

The used test dataset consisted of 5000 real trajectories; 900 square kilometers area; 2350 roads; and average 17 nodes per trajectory. A synthetic dataset of size i.e., number of trajectories ($50k$, $100k$, $1000k$, $100000k$) was generated using the same real road map as that of real dataset. The datasets and their equivalent file size in bytes are as in Table 4.1.

Table 4.1. Experimental Settings

Trajectories in Dataset	File Size in Bytes
5k	1.3M
50k	14M
100k	31.3M
1000k	313.2M
100000k	27.4GB

4.2.2. Experimental Results.

The accuracy of MapReduce-based algorithms, MRAN and MRARD are compared in Figure 4.6 for data size $5k$. It can be observed that they have almost the same precision. However MRAN has higher recall than MRARD. This can be attributed to using average nodes as the stopping criteria allows greater expansion than average road distance. Therefore average nodes per trajectory is used as the stopping criteria for comparing accuracy and processing time for bigger data size. In Figure 4.7, the accuracy and processing time of the MapReduce-based approach are reported as the data size vary from $5k$ to $100k$. The accuracy is reported in terms of both precision and recall. It can be observed that the MapReduce-based approach parallelizes

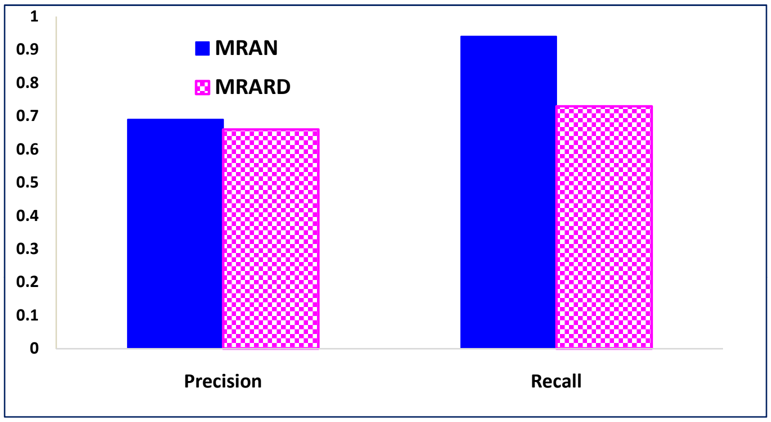
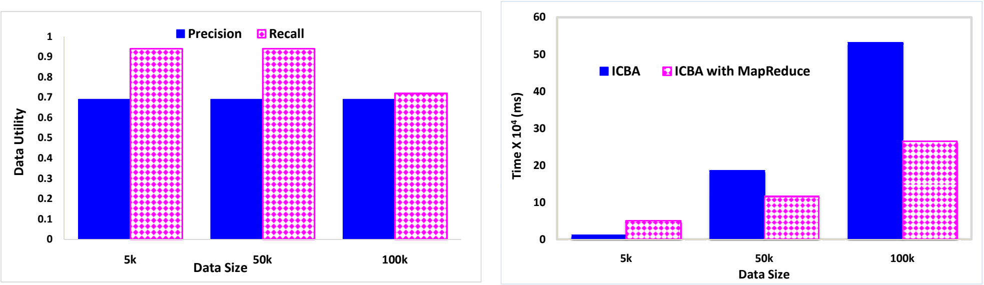


Figure 4.6. MapReduce-Based Approaches

the anonymization with less error as both the precision and recall are high irrespective of the data sizes. This can be attributed to the road network division algorithm which efficiently groups similar trajectories. The MapReduce-based anonymization algorithm was also tested for bigger datasets, $1000k$ and $100000k$. The centralized approach failed at $1000k$ given the available resources. The processing times of the centralized and MapReduce-based distributed anonymization algorithm are also compared. It is observed that the change in processing times between the two approaches increases with increase in data size. The MapReduce-based approach is more efficient when the data size is huge. Adopting MapReduce programming model efficiently parallelizes



(a) Data Utility

(b) Processing Time

Figure 4.7. Effect of Varying Data Size

the anonymization algorithm. The results show that using MapReduce model is very promising in anonymizing huge amounts of trajectory data.

4.3. SUMMARY

By using Map Reduce to efficiently parallelize the computations needed to simplify data, the amount of data that can be processed was increased greatly. The increase was enough to confidently claim that the method could handle the exabytes of data being produced per month globally and scale to handle even more data in the future. Additionally, the trajectory data was efficiently anonymized and protected from direct knowledge or inference attacks.

5. TRUSTWORTHINESS EVALUATION DURING LOCATION-BASED SERVICES

This section presents the approach on evaluating the trustworthiness of messages disseminated during location-based services. The Vehicular Ad-Hoc Networks (VANETs) are used as the background platform to elaborate the approach.

5.1. SYSTEM OVERVIEW

An overview of the proposed Real-time Message Content Validation (RMCV) scheme is given first. Each step of the scheme is then elaborated including the associated trust model.

The core of the RMCV is an information-oriented trust model which estimates the trustworthiness of message content by taking into account a variety of VANET-specific dimensions, such as who handled the message at what location and what time. The RMCV scheme consists of two main components: (i) Message Classification; and (ii) Information-oriented Trust Model. The outcome of the scheme is a "trustworthiness" value associated to each received message.

The model applies to information inquiry or information sharing applications, for which the following format of messages was adopted:

Definition 1. *Let Msg(${loc}_q, {loc}_{int}$, etype, info, t_e, mpath) be a message transmitted in VANETs for information inquiry or sharing:*

- ${loc}_q$*: The location of the query issuer or the entity to receive the shared information.*
- ${loc}_{int}$*: The querying location that the query issuer would like to know about the information, or the location of the shared information.*
- *etype: The event type which could be "traffic condition", "road condition", "coupon", etc.*

- *info: The information about the location loc_{int}, which could be the query results or shared information.*

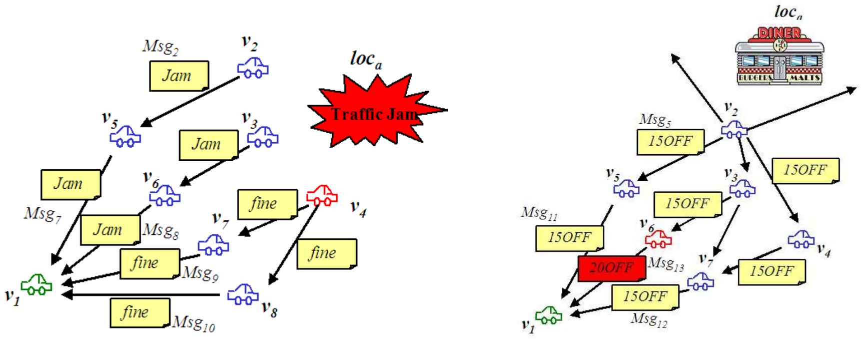
- *t_e: The time the query results or the shared information is available.*

- *mpath: This records the message propagation path. It is in the form of [(loc_{s_1}, t_{s_1}),(loc_{s_2}, t_{s_2}), ...), which means a vehicle at loc_{s_1} generated the message Msg at t_{s_1} and then the message was forwarded by the vehicle at loc_{s_2} at t_{s_2}, and so on. The locations of senders and message sending time are assumed to be stamped by a tamper-proof device installed in the vehicle.*

Figure 5.1(a) illustrates an example scenario of information inquiry. Vehicle V_1 at location loc_1 initiates a query on traffic condition at location loc_a. The query message is in the form of $Msg_1(loc_1, loc_a$, “traffic”, NULL, NULL, [(loc_1, t_1)]), where two fields *info* and t_e are waiting to be answered. The query was propagated to vehicles (V_2, V_3, V_4) close to the querying location loc_a. V_2 and V_3 honestly reported that there was a traffic jam by sending back the messages Msg_2 and Msg_3 respectively:

Msg_2(loc_1, loc_a, “traffic”, “traffic jam”, t_2, [(loc_2, t_2)])

Msg_3(loc_1, loc_a, “traffic”, “traffic jam”, t_3, [(loc_3, t_3)])



(a) Information Query (b) Information Sharing

Figure 5.1. Example Scenarios

However, a malicious node V_4 who lied that the traffic was fine and sent the following message: Msg_4(loc_1, loc_a, “traffic”, “traffic fine”, t_4, [(loc_4, t_4)]). Further, in order to make the message appear trustworthy, V_4 forwarded the message to multiple vehicles (V_7 and V_8) instead of the one close to V_1. A malicious vehicle may not know how many other malicious vehicles out there. Thus vehicle V_4 has to spread his messages to more vehicles otherwise his false messages can be easily ruled out based on a simple majority vote by V_1.

Upon receiving the messages initially sent by V_2, V_3 and V_4, the querying vehicle V_1 needs to analyze the conflicting information carried by the messages. It needs to figure out which one to trust. The proposed RMCV scheme can be executed by V_1 to conduct the trust evaluation, and it is expected that the true messages provided by V_2 and V_3 will receive higher trust scores.

The RMCV scheme also works for scenarios wherein one would like to share information with others. As shown in Figure 5.1(b), the owner of vehicle V_2 would like to share a coupon from a restaurant that he/she just visited. Thus, V_2 broadcasts the coupon code to other vehicles using message Msg_5, where loc_q is set to NULL as this is a broadcasting message: Msg_5(NULL, loc_a, “coupon”, “15% off code of TJ Restaurant 15OFF”, t_5, [(loc_2, t_5)]).

During the message propagation, some malicious nodes may purposely modify the coupon code to be invalid such as given by Msg_{13}. However, the malicious node would not be able to fake location and time information (i.e., $mpath$) which is directly generated by vehicle’s tamper proof device by using techniques such as [57]. For a vehicle which receives multiple coupon messages, it will again utilize the RMCV scheme to help identify the more trustworthy version: Msg_{11}(NULL, loc_a, “coupon”, “15% off code of TJ Restaurant 15OFF”, t_6, [(loc_2, t_5),(loc_5, t_6]).

5.2. MESSAGE CLASSIFICATION

In VANETs, one vehicle may receive multiple messages with different and possibly contrasting information from different vehicles during a short period of time. These messages may be related to different events (or different queries) occurring at same or different places. Therefore, the first step is to identify the messages describing the same event from the potentially large amount of received messages so that the analysis can be conducted separately for each event.

One may think of using clustering algorithms to cluster these messages. Messages corresponding to the same event may be similar or conflicting, if spurious or inaccurate messages are included. Direct adoption of conventional clustering algorithms is likely to put these related but conflicting messages in different groups, and hence affect the construction of the trust model. For example, applying a conventional K-means clustering algorithm to messages received by the vehicle V_1 illustrated in Figure 5.1, three clusters may be obtained: cluster C_1 (containing messages of “traffic jam”), cluster C_2 of messages about “traffic fine”, and cluster C_3 for the coupon code. Such clustering did not provide any hint that information in C_1 and C_2 is in fact responding to the same query and they are conflicting. Moreover, the cluster of C_3 did not identify the false coupon code either since the messages are very much similar in terms of content and other values of other components (e.g., location, event type) in the messages.

Thus, in order to better classify messages disseminated in VANETs, a two-level clustering algorithm is proposed. The first level clustering groups messages describing the same event regardless the message content. To achieve this, messages are clustered based on their similarity on the three components: loc_{int}, t_e, and $etype$. Specifically, two messages (Msg_i and Msg_j) would be placed in the same cluster if they satisfy all the following conditions:

- $D_l(loc_{int_i}, loc_{int_j}) \leq \rho_d$: D_l is the Euclidean distance of two locations. This condition requires that the two messages are reporting events not further than distance ρ_d so that it can be inferred that the two messages are likely to be about the same event. In this work, ρ_d is selected to be the width of a road which is about 20 meters for a three-lane road.
- $|t_{e_i} - t_{e_j}| \leq \rho_t$: Messages sent from the same locations may not refer to the same event. For example, messages responding to different queries may be sent from the same location at different timestamps. Therefore, the time threshold ρ_t is used to constraint the consideration within messages sent during nearby timestamps. In the experiments, ρ_t is set to be 30s within which most query results would not have big changes. For example, traffic condition would not change a lot within 30s.
- $etype_i = etype_j$: Two messages about the same event obviously need to have the same event type.

For each cluster obtained from the first level clustering, the second level clustering is conducted. The second level clustering aims to identify conflicting information regarding the same event. This clustering is conducted mainly by examining the message content, i.e., the similarity between the value of component (*info*) in the message. To compute the similarity of message content, first the keywords are extracted from *info* of a message by excluding articles (“a”, “an”, “the”) and connection words that do not carry important information. For example, given a message “there is no traffic jam”, it is converted to a set of keywords {“no”, “traffic”, “jam”}. Then, the keywords in the set are sorted in the alphabetical order. After that, the edit distance [65] and WordNet [70] are applied to compute the distance between keywords belonging to two messages. The distance calculation of two keyword sets KW_1 and KW_2 consists of three steps:

1. Firstly, the pairs of keywords that fully match each other are identified and removed from further consideration.

2. Next considered are the remaining keywords in the two sets that are pairs of synonyms based on WordNet. All such pairs are removed.

3. For remaining keywords, the keywords in KW_1 and KW_2 which have small edit distance are paired. These edit distances are summed up to obtain the edit distance (denoted as D_{ed}).

4. If there is any keyword left unpaired, such as when the two keyword sets have different number of keywords, the total characters of the unpaired keywords are summed up and added to D_{ed}.

If the distance (D_{ed}) between two message content is smaller than ρ_{info}, the two messages will be put in the same cluster. To ensure that conflicting information would have a high probability to be placed in different clusters, a strict threshold ρ_{info} is adopted which is set to 2 (the length of an important keyword "no"). For example, suppose that KW_1={"no", "traffic", "jam"} and KW_2={"traffic", "congestion"}. After sorting the keywords in each set, step 1 removes the matching keyword "traffic". Step 2 removes the synonyms "jam" and "congestion". Step 3 is skipped since there is no more pair left. Step 4 returns the final distance $D_{ed} = 2$ which is the length of the remaining keyword "no". It is worth noting that due to variety of the ways to express the same information, the distance here is just an estimation and may not be always accurate in some cases when messages have same meaning but are expressed in very different ways. The discussion on advanced natural language processing is out of the scope of this work.

To obtain a better understanding of the whole process of the message classification, the example scenarios are studied given in Figure 5.1. Vehicle V_1 received 7 messages which are Msg_7, Msg_8, ..., Msg_{13}. Suppose that t_e in all the messages are fairly close to one another, i.e., the difference less than ρ_t. Applying the three conditions on loc_{int}, t_e and $etype$, the following two clusters are obtained after the first-level clustering:

$C_1 = \{Msg_7, Msg_8, Msg_9, Msg_{10}\}$, $C_2 = \{Msg_{11}, Msg_{12}, Msg_{13}\}$.

This is because messages in C_1 report the same type of event "traffic" at the same location loc_a almost at same time, while messages in C_2 are about coupon information at loc_a.

Next, second-level clustering is conducted for C_1 and C_2 respectively. The cluster C_1 is further divided into two clusters based on the message content:

$C_{11} = \{Msg_7, Msg_8\}$, $C_{12} = \{Msg_9, Msg_{10}\}$.

Similarly, the cluster C_2 is also divided into two clusters based on the content:

$C_{21} = \{Msg_{11}, Msg_{12}\}$, $C_{22} = \{Msg_{13}\}$.

5.3. INFORMATION-ORIENTED TRUST MODEL

After the message classification, the next task is to determine which group of messages are truth-telling. To achieve this, an information-oriented trust model is designed. The overall process is to identify the factors that may be indicative of message trustworthiness, and then quantify their impact and integrate their effects to generate an overall trustworthiness score that can be easily understood by end users for making decisions. Three important factors are identified that affect message trustworthiness, which are *content similarity*, *content conflict* and *routing path similarity*. In what follows, an explanation of why they are important, how they affect the trust score is provided. The trust model is finally derived based on these factors.

5.3.1. Effect of Content Similarity. Given a group of messages associated to a same event, similar messages are generally considered to be supportive to one another. Moreover, similar to daily life conversations, the more people supporting the same fact, the more likely the fact would have some true ground. Though this observation may not always hold as discussed later in Section 5.3.2., it is certainly an important factor to be considered when judging the trustworthiness of a message. To model these two effects, two parameters are used. The first parameter is the maximum distance ($maxD_c$) of content between two messages in the same cluster. It quantifies the similarity of information in the same cluster. The smaller the distance, the higher support level of the information given by each other. The second parameter is the number of messages (N_c) in the cluster which models the second effect: the more messages in the cluster, the higher support the message received. The two parameters are then integrated to compute the support value by using Equation 5.1.

$$Support(c) = \frac{e^{\frac{N_c}{N_e}}(\frac{3}{2} - \frac{maxD_c}{\rho_{ed}})}{\frac{2}{3}e} \tag{5.1}$$

The rationale behind Equation 5.1 is explained as follows.

- In the first part of the formula, N_e is the total number of messages regarding the event. Dividing N_c by N_e is for the purpose of obtaining a normalized value ranging in 0 and 1, since $0 \leq N_c \leq N_e$. Such normalization helps make values obtained from different clusters of messages comparable. The effect of N_c is then modeled by an exponential function $e^{\frac{N_c}{N_e}}$. The reason to choose the exponential function is that the resulting value grows faster when the effect becomes more dominant. This maps the following scenario. For groups of few number of messages (e.g., two or three messages), it is hard to say one group is more trustworthy than the other just because of it has one more supportive message. Therefore, such groups will have very close trust scores. When the number of messages in a group is much bigger, the trust score will grow much faster using the exponential function, and this represents that the probability of the message being true is higher.

- In $\frac{maxD_c}{\rho_{ed}}$, $maxD_c$ is normalized to the range of 0 to 1 by using the possible maximum distance ρ_{ed}. Recall that ρ_{ed} is the threshold used to determine whether two messages can be placed in the same cluster. The value $\frac{3}{2}$ is used for two purposes. First, it reverses the effect of $\frac{maxD_c}{\rho_{ed}}$ so that when the difference of messages is greater, the trust score would be lower. Second, it ensures that the second part will have certain effect on the overall trust score even if it reaches the maximum distance. In particular, when messages in the cluster are the same, i.e., $maxD_c = 0$, the second part returns a value 1.5. In contrast, when $maxD_c = 1$, the second part returns value 0.5.
- The value obtained from the product of the previous two components ranges from $\frac{1}{2}$ to $\frac{3}{2}e$. By dividing the product by $\frac{3}{2}e$, the final similarity score is normalized to be less than 1. It is always greater than 0 since messages in the same cluster are expected to have at least some similarity.

5.3.2. Effect of Routing Path Similarity. It is likely for one to trust a message which has a large number of other similar messages as the support. However, considering content similarity may not be sufficient to determine the trustworthiness of the message since in some cases a large number of messages may also cause illusion. An extreme case is that if all messages have the same origin and the origin is a malicious vehicle, these messages should not be trusted. From the example shown in Figure 5.1, the vehicle V_1 received two groups of conflicting messages about the traffic condition. These two groups of messages have equal content similarity scores according to Equation 5.1 in Section 5.5., making it difficult to tell which is more trustworthy. However, if observed closely, one may notice that the group of false messages (Msg_9 and Msg_{10} are actually provided by the same source vehicle, while the group of true messages (Msg_7 and Msg_8) have different source providers. Following a general assumption that majority of people are honest, it is less likely that the majority of people purposely provide wrong information. Therefore, the probability of multiple source providers reporting the same wrong information is expected to be lower than that of a single source provider in most cases. More generally speaking, if similar messages share more common nodes during their routing paths, the risk of messages being tampered increases.

Based on the above discussion, the effect of routing path similarity is modeled by using three parameters: the number of messages (N_c) in the cluster, the number of the origins of the messages (N_{src}), and the number of distinct vehicles (N_{dif}) in the routing paths of messages in the same cluster. Then, the path similarity function is designed based on the following guidelines:

- If there are a large number of source providers (N_{src}), the message routing paths are less likely to be similar.
- If there are common vehicles in multiple paths and the common vehicle is malicious, all messages forwarded by the malicious vehicle may be tampered. To model this, the more distinct vehicles (N_{dif}) involved in the same cluster of messages, the lower path similarity should be.

The following equation sums up the above effects:

$$Path_c = 1 - \big(0.5\frac{N_{src}}{N_c} + 0.5\frac{N_{dif}}{N_{all}}\big) \tag{5.2}$$

In Equation 5.2, N_{all} denotes the total number of vehicle nodes involved in forwarding the messages in the cluster C. If the same vehicle occurs in different paths, each of its occurrence would be counted to N_{all}. Then, $\frac{N_{dif}}{N_{all}}$ yields the percentage of the distinct vehicles in the routing paths. Though this percentage also reflects the difference of source providers to certain degree, an equal weight (0.5) is still assigned to the number of source providers due to its importance.

The steps of computing the path similarity are illustrated using the example in Figure 5.1 . In cluster $C_{11} = \{Msg_7, Msg_8\}$, the routing paths are the following:

$Msg_7 : V_2 – V_5$; $Msg_8 : V_3 – V_6$.

Observe that in the above two ($N_c = 2$) messages, there are two different sources ($N_{src} = 2$), four different nodes ($N_{dif} = 4$), and total four nodes ($N_{all} = 4$). Therefore, the $Path_c = 1 - (0.5 \cdot \frac{2}{2} + 0.5 \cdot \frac{4}{4})$=0, which means the paths are totally different.

In cluster $C_{12} = \{Msg_9, Msg_{10}\}$, the routing paths are the following:

Msg_9:$V_4 – V_7$; $Msg_{10} : V_4 – V_8$.

Accordingly, $N_c = 2$, $N_{src} = 1$, $N_{dif} = 3$, $N_{all} = 4$ are obtained. Then, the numbers are plugged into Equation 5.2 and path similarity is obtained as $Path_c = 1 - (0.5 \cdot \frac{1}{2} + 0.5 \cdot \frac{3}{4})$ = 0.375 which has a higher path similarity score compared to cluster C_{11}.

The path similarity serves as a penalty value to the support value of a cluster of messages. The more similar the routing paths of messages in the same cluster, the less support to each other will be considered. In other words, the more independent of routing paths, the less probability of messages being tampered. The Equation 5.1 is revised as follows:

$$Support'(c) = (1 - Path_c) \cdot Support(c) \tag{5.3}$$

5.3.3. Effect of Content Conflict. The analysis of messages referring to a same event, may result in more than one cluster of messages. Messages in different clusters indicate the inconsistency of the information of the event. As shown in the example of Figure 5.1, one cluster of messages claim there is traffic jam while the other claim the traffic is fine. It is obvious that content conflict has a negative impact on the trustworthiness of messages, and the more conflicting messages the heavier impact. Specifically, let C_1, ..., C_k be the clusters of messages regarding the same event. For each cluster of messages, a conflicting value Con_{c_i} is computed given by Equation 5.4.

$$Con_{c_i} = \frac{e^{\frac{\sum_{j=1}^{k} Support'_{c_j} - Support'_{c_i}}{\sum_{j=1}^{k} Support'_{c_j}}}}{e} \tag{5.4}$$

A higher conflicting value will be obtained if there are more messages against current cluster C_i. The conflicting value is 0 if there is not any conflicting clusters. Here, the exponential function is adopted for the same purpose of amplifying the effect.

5.3.4. Final Trust Score. To obtain the final trust score $trust(c)$, the conflicting value is integrated to the support score Support'(c). In particular, the conflicting value is used to further penalize the support value as given by the following equation.

$$trust(c) = \frac{(e^{\xi} - e^{\xi \cdot Con_c})Support'(c)}{e^{\xi} - 1} \tag{5.5}$$

It is modeled based on the following rationale. When the conflicting value is small, its effect should not be very dominant. In this way, if there exist few false messages, these false messages would not affect the overall trustworthiness of the true messages. When the conflicting value is big, its effect grows faster as it is more likely that the information in the cluster being affected is not true regarding the existence of a large number of opponents. Therefore, as can be seen from Equation 5.5, $e^{\xi \cdot Con_c}$ models the impact of the conflicting value whereby the exponential function along with a parameter ξ make the resulting value grow faster with the increase of Con_c. Here, ξ is a positive value that helps adjust the importance of the conflicting value, and it is set to e in the experiments. Finally, the score is normalized to range 0 to 1 by multiplying $\frac{1}{e^{\xi}-1}$. The higher the trust score, the more trustworthy the message may be.

Finally, the overall process of estimating the trustworthiness of a message is summarized. Given a bunch of messages received by vehicle V within a short time interval ρ_t, the RMCV scheme first clusters messages according to the events, and then further clusters messages based on their content. After that, trust scores are computed for all the clusters of messages. For clusters of the same event, the one which received the highest trust score is selected. If its trust score is above an experience threshold (e.g., 0.5), the system would report that the content of this cluster may be trustworthy. Otherwise, the system would report that none of the received messages are trustworthy.

In addition, one more interesting scenario is introduced that can also be handled by this approach. Suppose that a vehicle V_x sends the following two messages:

- Msg_{x1}: At time t_1, there is a traffic jam between exits 25 and 30 in HWY 65.
- Msg_{x2}: At time t_2, there is no traffic jam between exits 25 and 30 in HWY 65.

It may be the case that between t_1 and t_2 things have changed, or it could be the case that a vehicle can only observe some partial view and later on may see a complete view and send a different message for correction.

For the given scenario, the RCMV scheme will deal with it as follows:

- Case 1: Suppose that t_2 is far from t_1 (e.g., 30 minutes later). All messages (including the one from vehicle V_x and others) about traffic jam sent around time t_1 would be considered as message for one event. These messages are compared to see if there was a real traffic jam at t_1. Messages sent around t_2 will be considered as another event (no jam) which could be true if the traffic was clear at t_2.
- Case 2: Suppose that t_2 is close to t_1 (e.g., only a couple of minutes different), and there is in fact no traffic jam but vehicle V_x made a wrong observation at t_1. In this case, the message of “traffic jam” will be considered as a conflicting message. Assuming that majority is honest, more messages of “no traffic jam” is expected around timestamp t_1, so that the receiver would not be confused.

5.4. EXPERIMENTAL STUDY

In this section, the experimental settings are presented and a comparative study of the approach against the existing work is also reported.

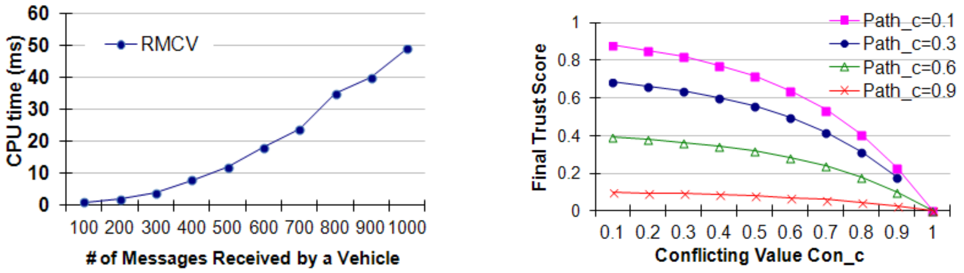
The implementation is written in JAVA and conducted in a desktop of 64-bit Intel(R) Xeon(R) E5630 2.53GHz machine. The message disseminated in VANETs is simulated as follows. A parameter is adopted that controls the number of hops N_{hop} between the source provider and the query issuer (or the last message receiver) being considered. In the experiments, N_{hop} is varied from 1 to 5. At each hop, 100 vehicles are generated. For each event, on a randomly selected hop, δ percent of malicious vehicles is selected. For the vehicles at the first hop, true messages are generated about several events for honest vehicles, and conflicting messages for malicious vehicles. Honest vehicles will honestly forward whatever messages they receive to one vehicle at the next hop, while malicious vehicles will modify the received messages and forward them to multiple vehicles (ranging from 1 to N_f) at the next hop.

The approach is compared with the work by Raya et al. [56] which is the latest representative work on data-centric trust establishment in VANETs. As their work is based on Bayesian Inference, it is denoted as BI in the experiment figures. Since the BI work only considers a single event, the messages are limited to one event when comparing to them. Also, the BI work assumes the existence of trust scores (probability of trustworthiness) of each message for computing the final trust score of the event. In the simulation in their work, they assume the probability of trustworthiness of individual messages follows a Beta distribution with the mean equals to 0.6 and 0.8. The same parameters as in their work are adopted in the experiments.

5.4.1. Experimental Results. In the first two rounds of experiments, the properties of the RMCV are examined. In the last round of experiments, the RMCV approach is compared with the BI work in terms of the ability of preventing attack.

5.4.1.1 Efficiency. In the first round of experiments, the objective is to evaluate the efficiency of the RMCV scheme. Unlike the BI work which assumes the existence of scores of individual messages and just computes one equation for the final trust score, the RMCV scheme offers detailed steps to obtain the trust scores of individual messages. These steps include message classification and routing path similarity analysis. The Figure 5.2(a) reports the total time taken by the RMCV scheme from messages being received till the trust score being computed. The total number of messages that a vehicle received during ρ_t are varied from 100 to 1000. There are five hops along each routing path. It is not surprising to see that the processing time increases with the number of messages to be handled. This is because the more messages, the more time needed for message classification and path analysis. It is also observed that the time for processing 1000 messages is really short (less than 50ms), which indicates that the scheme is feasible and efficient to meet the strict time constraint in real-time applications.

5.4.1.2 Effect of conflicting value and path similarity on trustworthiness score. In this experiment, it is presented how conflicting values and path similarity values affect the overall trustworthiness score. From Figure 5.2(b), it can be observed that the trustworthiness score decreases with the increase of conflicting values or path similarity values. More importantly, the trust score drops faster when the conflicting value and path similarity value become larger. Thus, the model is tolerant to cases when there are few false reports (i.e., conflicting information), and becomes more sensitive when the number of false reports increases.



(a) Processing Time (b) Trustworthiness Score

Figure 5.2. The RMCV Approach

5.4.1.3 Impact of false messages on vehicles accepting true messages. The RCMV scheme is now compared with the BI work. The effect of increase in the percentage of false messages per vehicle to the percentage of good vehicles accepting true messages is examined. A simulation of 1000 rounds was run for a group of 100 vehicles. The results are reported in Figure 5.3. From the figure, it can be observed that when the amount of false messages is less than 50%, both the BI work and the RCMV approach can very well identify false reports, yielding close to 100% acceptance rate of true messages. However, once there are more than 50% false messages, the BI work results in very low (close to 0%) acceptance rate of true messages. In fact, the BI work almost downgrades to a majority vote. In contrast, the RCMV approach yields much better performance even if there are many false messages. This is attributed to the way the conflicting information and path similarity are modeled. Specifically, since false messages tend to have higher path similarity scores, the penalty score from path similarity decreases the impact of the large amount of false messages on making the final decision.

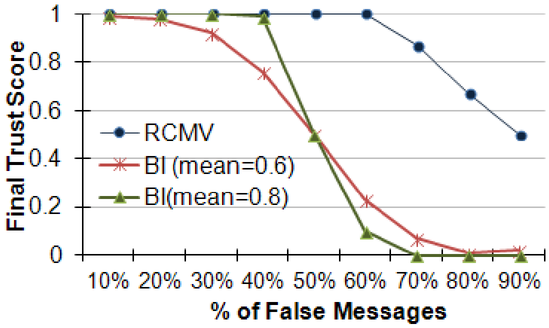


Figure 5.3. RCMV vs. BI

5.5. SUMMARY

This section presents a novel information-oriented scheme for evaluating trustworthiness of messages disseminated in VANETs, which incorporates content similarity, content conflict and route similarity into the trust model to best suit the dynamics of VANET environment.

6. CONCLUSION AND FUTURE WORK

In this dissertation, three works are presented with respect to privacy management and trustworthiness evaluation in location-based services. Specifically, the first work addresses the problem on publishing location data with privacy preservation while maintaining high data utility rate. The second work extends the centralized location data publishing approach to a distributed version by leveraging MapReduce technology, and is capable of processing a huge amount of location data in an efficient manner. Finally, the third work addresses an important issue correlated to privacy preservation, which is the trustworthiness evaluation of messages disseminated by anonymous users in location-based service. For all the proposed approaches, extensive experiments have been conducted using both synthetic and real datasets to verify the ideas.

Regarding future research directions, the following are envisioned. First, fine-grained temporal parameters may be integrated into the trajectory anoymization algorithm to generate more insight of the traffic flow. Second, a few other options of map partitioning may be explored to further reduce the information loss caused by the distributed processing in MapReduce. Third, existing natural language processing techniques may be integrated to the content evaluation in our proposed trust model to improve usability of the system.

BIBLIOGRAPHY

[1] O. Abul, M. Atzori, F. Bonchi, and F. Giannotti. Hiding sensitive trajectory patterns. In *Proc. of ICDM Workshop*, pages 693 – 698, 2007.

[2] O. Abul, F. Bonchi, and F. Giannotti. Hiding sequential and spatio-temporal patterns. *IEEE Transactions on Knowledge and Data Engineering*, 22(12):1709–1723, 2010.

[3] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proc. of the International Conference on Data Engineering*, pages 376–385, 2008.

[4] O. Abul, F. Bonchi, and M. Nanni. Anonymization of moving objects databases by clustering and perturbation. *Information Systems*, 35(8):884–910, 2010.

[5] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. *Database Theory-ICDT 2005*, pages 246–258, 2005.

[6] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonymity. *Journal of Privacy Technology*, 2005112001, 2005.

[7] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 439–450, 2000.

[8] G. Andrienko, N. Andrienko, F. Giannotti, A. Monreale, and D. Pedreschi. Movement data anonymity through generalization. In *Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*, pages 27–31. ACM, 2009.

[9] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Anonymity preserving pattern discovery. *The VLDB journal*, 17(4):703–727, 2008.

[10] R.J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. 2005.

[11] S. Brakatsoulas, D. Pfoser, and N. Tryfona. Modeling, storing and mining moving object databases. In *Database Engineering and Applications Symposium, 2004. IDEAS'04. Proceedings. International*, pages 68–77. IEEE, 2004.

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

[13] S. Buchegger and J.-Y. Le Boudec. A robust reputation system for peer-to-peer and mobile ad-hoc networks. In *Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[14] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *ACM World Wide Web Conference*, pages 675–684, 2011.

[15] C. Chen, J. Zhang, R. Cohen, and PH Ho. A trust-based message propagation and evaluation framework in vanets. In *Proceedings of the Int. Conf. on Information Technology Convergence and Services*, 2010.

[16] R. Chen, B. Fung, and B. Desai. Differentially private trajectory data publication. *arXiv preprint arXiv:1112.2020*, 2011.

[17] C.Y. Chow, M.F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 171–178. ACM, 2006.

[18] V. Ciriani, S.D.C. di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. *Secure Data Management in Decentralized Systems. Springer-Verlag*, 2007.

[19] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu. An approach to evaluate data trustworthiness based on data provenance. In *Secure Data Management*, pages 82–98, 2008.

[20] J. Domingo-Ferrer and R. Trujillo-Rasua. Microaggregation-and permutation-based anonymization of mobility data. *Information Sciences*, 2012.

[21] F. Dotzer, L. Fischer, and P. Magiera. Vars: A vehicle ad-hoc network reputation system. In *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, pages 454–456. IEEE, 2005.

[22] Alina Ene, Sungjin Im, and Benjamin Moseley. Fast clustering using mapreduce. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 681–689. ACM, 2011.

[23] L. Eschenauer, V. D. Gligor, and J. Baras. On Trust Establishment in Mobile Ad-Hoc Networks. In *Security Protocols Workshop*, pages 47–66, 2002.

[24] B.C.M. Fung, K. Wang, and P.S. Yu. Top-down specialization for information and privacy preservation. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 205–216. IEEE, 2005.

[25] S. Ganeriwal and M. Srivastava. Reputation-based framework for high integrity sensor networks. In *ACM workshop on Security of ad hoc and sensor networks*, pages 66–77, 2004.

[26] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. In *Proceedings of the IEEE International conference on Distributed Computing Systems (ICDS05)*, pages 620–629. Citeseer, 2005.

[27] Matthias Gerlach. Trust for vehicular applications. In *Proceedings of the Eighth International Symposium on Autonomous Decentralized Systems*, pages 295–304, Washington, DC, USA, 2007. IEEE Computer Society.

[28] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.L. Tan. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM, 2008.

[29] G. Ghinita, P. Kalnis, and S. Skiadopoulos. Mobihide: A mobilea peer-to-peer system for anonymous location-based queries. *Advances in Spatial and Temporal Databases*, pages 221–238, 2007.

[30] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVE: anonymous location-based queries in distributed mobile systems. In *Proceedings of the 16th international conference on World Wide Web*, pages 371–380. ACM, 2007.

[31] G. Gidofalvi, X. Huang, and T. B. Pedersen. Privacy-preserving data mining on moving object trajectories. In *Proc. of the International Conference on Data Engineering*, pages 60–68, 2007.

[32] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

[33] U. Hengartner and P. Steenkiste. Protecting access to people location information. *Security in Pervasive Computing*, pages 222–231, 2004.

[34] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *Proc. of SecureComm*, pages 194–205, 2005.

[35] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proc. of the ACM conference on Computer and Communications Security*, pages 161–171, 2007.

[36] H. Hu and J. Xu. Non-exposure location anonymity. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 1120–1131. IEEE, 2009.

[37] Z. Huo, X. Meng, and R. Zhang. Feel free to check-in: Privacy alert against hidden location inference attacks in geosns. In *Database Systems for Advanced Applications*, pages 377–391, 2013.

[38] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM, 2005.

[39] D. Lin, S. Gurung, W. Jiang, and A. Hurson. Privacy-preserving location publishing under road-network constraints. In *Proceedings of International Conference on Database Systems for Advanced Applications*, 2010.

[40] Zhenhua Lv, Yingjie Hu, Haidong Zhong, Jianping Wu, Bo Li, and Hui Zhao. Parallel k-means clustering of remote sensing images based on mapreduce. In *Web Information Systems and Mining*, pages 162–170. Springer, 2010.

[41] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 223–228. ACM, 2004.

[42] U.F. Minhas, J. Zhang, T. Tran, and R. Cohen. Towards expanded trust management for agents in vehicular ad-hoc networks. *International Journal of Computational Intelligence Theory and Practice (IJCITP)*, 5(1), 2010.

[43] N. Mohammed, B. Fung, and M. Debbabi. Walking in the crowd: anonymizing trajectory data for pattern analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1441–1444. ACM, 2009.

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

[45] M.F. Mokbel, C.Y. Chow, and W.G. Aref. The new casper: Query processing for location services without compromising privacy. In *Proceedings of the 32nd international conference on Very large data bases*, pages 763–774. VLDB Endowment, 2006.

[46] A. Monreale, G. Andrienko, N. Andrienko, F. Giannotti, D. Pedreschi, S. Rinzivillo, and S. Wrobel. Movement data anonymity through generalization. *Transactions on Data Privacy*, 3(2):91–121, 2010.

[47] A. Monreale, R. Trasarti, D. Pedreschi, C. Renso, and V. Bogorny. C-safety: a framework for the anonymization of semantic trajectories. *Transactions on Data Privacy*, 4(2):73–101, 2011.

[48] J. Mundinger and J.-Y. Le Boudec. Reputation in self-organized communication systems and beyond. In *Workshop on Interdisciplinary systems approach in performance evaluation and design of computer & communications systems*, page 3, 2006.

[49] Y. Nakajima, K. Watanabe, N. Hayashibara, T. Enokido, M. Takizawa, and S. M. Deen. Trustworthiness in peer-to-peer overlay networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, page 8, 2006.

[50] M. E. Nergiz, M. Atzori, Y. Saygin, and B. Guc. Towards trajectory anonymization: a generalization-based approach. *Transactions on Data Privacy*, 2(1):47–75, 2009.

[51] M. E. Nergiz, M. Atzori, Y. Saygin, and B. Guc. Towards trajectory anonymization: a generalization-based approach. *Transactions on Data Privacy*, 2(1):47–75, 2009.

[52] A. Patwardhan, A. Joshi, T. Finin, and Y. Yesha. A data intensive reputation management scheme for vehicular ad hoc networks. In *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*, pages 1–8. IEEE, 2006.

[53] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge & Data Engineering*, 16(1):1424–1440, 2004.

[54] R.G. Pensa, A. Monreale, F. Pinelli, and D. Pedreschi. 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*, 2008.

[55] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux. Eviction of misbehaving and faulty nodes in vehicular networks. *Selected Areas in Communications, IEEE Journal on*, 25(8):1557 –1568, oct. 2007.

[56] M. Raya, P. Papadimitratos, V. D. Gligor, and J. p. Hubaux. On datacentric trust establishment in ephemeral ad hoc networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2008.

[57] J. Richter, N. Kuntze, and C. Rudolph. Security digital evidence. In *IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*, pages 119–130, 2010.

[58] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, pages 1010–1027, 2001.

[59] E. Snekkenes. Concepts for personal location privacy policies. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 48–57. ACM, 2001.

[60] L. Sweeney. Datafly: A system for providing anonymity in medical data. In *Proceedings of the IFIP TC11 WG11. 3 Eleventh International Conference on Database Securty XI: Status and Prospects*, page 381. Chapman & Hall, Ltd., 1997.

[61] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[62] J.C. Tanner. In search of lbs accountability. In *Telecom Asia*, 2008.

[63] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *Proc. of the International Conference on Mobile Data Management*, pages 65–72, 2008.

[64] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.

[65] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *J. ACM*, 21(1):168–173, 1974.

[66] K. Wang, P.S. Yu, and S. Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 249–256. IEEE, 2005.

[67] Kai Wang, Jizhong Han, Bibo Tu, Jiao Dai, Wei Zhou, and Xuan Song. Accelerating spatial data processing with mapreduce. In *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*, pages 229–236. IEEE, 2010.

[68] L.-Y. Wei, Y. Zheng, and W.-C. Peng. Constructing popular routes from uncertain trajectories. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 195–203, 2012.

[69] W. Winkler. Using simulated annealing for k-anonymity. Washington, DC: US Census Bureau Statistical Research Division. Technical report, Technical Report 2002-07, 2002.

[70] WordNet. http://wordnet.princeton.edu/.

[71] A. Y. Xue, R. Zhang, Y. Zheng, X. Xie, J. Huang, and Z. Xu. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In *International Conference on Data Engineering*, 2013.

[72] A. Y. Xue, R. Zhang, Y. Zheng, X. Xie, J. Yu, and Y. Tang. Desteller: A system for destination prediction based on trajectories with privacy protection. 2013.

[73] R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, and W. H. Wang. Anonymizing moving objects: how to hide a mob in a crowd? In *Proc. of the International Conference on Extending Database Technology*, pages 72–83, 2009.

[74] X. Yin, J. Han, and P. S. Yu. Truth Discovery with Multiple Conflicting Information Providers on the Web. In *Proc. of ACM SIGKDD*, pages 1048–1052, 2007.

[75] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

[76] X. Zeng, J. Pei, K. Wang, and J. Li. Pads: a simple yet effective pattern-aware dynamic search method for fast maximal frequent pattern mining. *Knowledge and Information Systems*, 20(3):375–391, 2009.

[77] Jie Zhang. A survey on trust management for vanets. In *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*, pages 105 –112, march 2011.

[78] Shubin Zhang, Jizhong Han, Zhiyong Liu, Kai Wang, and Shengzhong Feng. Spatial queries evaluation with mapreduce. In *Grid and Cooperative Computing, 2009. GCC'09. Eighth International Conference on*, pages 287–292. IEEE, 2009.

[79] Y. Zheng, L. Zhang, X. Xie, and W.Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.

[80] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas. Robust Cooperative Trust Establishment for MANETs. In *ACM workshop on Security of ad hoc and sensor networks*, pages 23–34, 2006.

[81] Executive Summary (2014-02-04). http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018*, Retrieved 2014-03-05.

[82] Zixkhur, Kathryn (2013-09-12). http://www.pewinternet.org/2013/09/12/location-based-services/. *Location-Based Services*, Retrieved 2014-03-05.

[83] RT (2014-01-10). http://rt.com/usa/ford-vp-auto-surveillance-382/ *Ford VP: "We have GPS in your car, so we know what you're doing."*, Retrieved 2014-03-05.

[84] Gedawy, Hend Kamal. Dynamic Path Planning and Traffic Light Coordination for Emergency Vehicle Routing. *Thesis*, 2009.

[85] Bonchi, Francesco. Lakshmanan, Laks. Wang, Wui. Trajectory Anonymity in Publishing Mobility Data. *SIGKDD Explorations*, Vol. 13, Iss. 1.

[86] Francis, Matthew. *Future Telescope Array drives development of exabyte processing*, Retrieved 2014-10-24.

[87] Jacobs, A. The Pathologies of Big Data. *ACMQueue*, 2009.

[88] Samarati, P., and Sweeney, L. Generalizing data to provide anonymity when disclosing information. In *Proc. of the 17th ACM Symp. on Principles of Database Systems* (PODS98).

[89] Abul, O., Bonchi, F., and Nanni, M. Never Walk Alone: Uncertainty for anonymity in moving objects databases. In *Proc. of the 24nd IEEE Int. Conf. on Data Engineering (ICDE08)*.

[90] Grutesaer, Marco. Grunwald, Dirk. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. *University of Colorado at Boulder*.

[91] Gedik, Bugra. Liu, Ling. A Customizable k-Anonymity Model for Protecting Location Privacy. *Georgia Institute of Technology*.

[92] Dean, Jeffrey. Ghemawat, Sanjay. MapReduce: Simplified Data Processing on Large Clusters. *OSDI*, 2004

[93] Papadias, Dimitris. Theodoridis, Yannis. Spatial Relations, Minimum Bounding Rectangles, and Spatial Data Structures. *Technical Reports KDBSLAB-TR-94-04*.

VITA

Sashi Gurung is from Nepal, a beautiful himalayan country in Asia. She received her Bachelor of Engineering degree in Computer Engineering from Institute of Engineering, IOE in 2007 and Doctor of Philosophy degree in Computer Science from Missouri University of Science and Technology, Rolla, MO, USA, in 2014. Her research interests include Location Privacy, Spatial Databases, Data Mining and Big Data.

She was a recipient of Grace Hopper Scholarship 2013. Her research publications are as follows:

1. Sashi Gurung, Dan Lin, Wei Jiang, Ali Hurson, and Rui Zhang, “Traffic Information Publication with Privacy Preservation,” ACM Transactions on Intelligent Systems and Technology (TIST), 2014.

2. Sashi Gurung, Dan Lin, Anna Squicciarini, and Elisa Bertino, “Information -oriented Trustworthiness Evaluation in Vehicular Ad-hoc Networks,” The 7th International Conference on Network and System Security (NSS), 2013.

3. Sashi Gurung, Dan Lin, Anna Squicciarini, and Ozan Tonguz, “A Moving Zone Based Architecture for Message Dissemination in VANETs,” The 8th International Conference on Network and Service Management (CNSM), 2012.

4. Dan Lin, Sashi Gurung, Wei Jiang, and Ali Hurson, “Privacy-Preserving Location Publishing under Road-Network Constraints,” In Proceedings of 15th International Conference on Database Systems for Advanced Applications (DASFAA), pages 17–31, 2010.