LOCATION PRIVACY IN THE ERA OF BIG DATA AND MACHINE LEARNING

Sina Shaham

Engineering and Information Technologies

University of Sydney

A thesis submitted to fulfil requirements for the degree of

Master of Philosophy

December 2019

Statement of Originality

I hereby declare that all material presented in this thesis are the original work of the author, except where specific reference is made to the work of others. The content of this thesis has not been submitted in whole or in part consideration for any other degree or qualification in this, or any other university.

The original motivation to pursue my research was provided by my supervisor Dr. Zihuai Lin in the school of Electrical and Information Engineering, the University of Sydney, Australia.

Sina Shaham

December 2019

Acknowledgements

I would like to pay special thankfulness, warmth, and appreciation to many people who helped me. This work would not have been possible with the inspiration and encouragement from them.

I would like to show my warm thanks to my supervisor Dr. Zihuai Lin who supported me at every bit, including but not limited to the research.

I am very thankful to Dr. Ming Ding. His vast knowledge and guidance on research has been quite invaluable to me. His valuable comments have significantly improved the level of my research.

I am extremely grateful to my parents for their unwavering love and encouragement.

Publications

The following is a list of my publications finished during my Master of Philosophy.

[C1] Sina Shaham, Ming Ding, Bo Liu, Zihuai Lin, Jun Li, 'Machine Learning Aided Anonymization of Spatiotemporal Trajectory Datasets' accepted by INFOCOM WORKSHOPS, 2019.

[C2] Sina Shaham, Ming Ding, Bo Liu, Zihuai Lin, Jun Li, 'Transition-Entropy: A Novel Metric for Privacy Preservation in Location-Based Services' accepted by INFOCOM WORKSHOPS, 2019.

[J1] Sina Shaham, Ming Ding, Bo Liu, Shuping Dang, Zihuai Lin, and Jun Li, 'Privacy Preservation in Location-Based Services: A Novel Metric and Attack Model' Submitted to IEEE Transactions on Knowledge and Data Engineering, 2019.

[J2] Sina Shaham, Ming Ding, Bo Liu, Shuping Dang, Zihuai Lin, and Jun Li, 'Privacy Preserving Location Data Publishing: A Machine Learning Approach' Submitted to IEEE Transactions on Knowledge and Data Engineering, 2019.

[J3] Sina Shaham, Ming Ding, Matthew Kokshoorn, Zihuai Lin, Shuping Dang, and Rana Abbas, 'Fast Channel Estimation and Beam Tracking for Millimeter Wave Vehicular Communications' Published in IEEE Access, 2019.

Abstract

Location data of individuals is one of the most sensitive sources of information that once revealed to ill-intended individuals or service providers, can cause severe privacy concerns. In this thesis, we aim at preserving the privacy of users in telecommunication networks against untrusted service providers as well as improving their privacy in the publication of location datasets.

For improving the location privacy of users in telecommunication networks, we consider the movement of users in trajectories and investigate the threats that the query history may pose on location privacy. We develop an attack model based on the Viterbi algorithm termed as Viterbi attack, which represents a realistic privacy threat in trajectories. Next, we propose a metric called transition entropy that helps to evaluate the performance of dummy generation algorithms, followed by developing a robust dummy generation algorithm that can defend users against the Viterbi attack. We compare and evaluate our proposed algorithm and metric on a publicly available dataset published by Microsoft, i.e., Geolife dataset.

For privacy preserving data publishing, an enhanced framework for anonymization of spatio-temporal trajectory datasets termed the machine learning based anonymization (MLA) is proposed. The framework consists of a robust alignment technique and a machine learning approach for clustering datasets. The framework and all the proposed algorithms are applied to the Geolife dataset, which includes GPS logs of over 180 users in Beijing, China.

Table of Contents

List of Figures

List of Tables

Nomenclature

Acronyms / Abbreviations

DGH	Domain Generalization Hierarchy
DLS	Domain Location Selection
GPS	Global Positioning System
LBS	Location Based Service
LCA	Lowest Common Ancestor
MLA	Machine Learning based Anonymization
RAM	Random Access Memory
RDG	Robust Dummy Generation
SA	Sequence Alignment

Chapter 1

Introduction

This chapter presents a brief background knowledge of our research topics and raises research problems. Moreover, this chapter illustrates the main contributions of this thesis.

1.1 Location Privacy

The ubiquitous use of location-based mobile applications has made location data one of the primary sources of information. Users of such applications provide their location data to location-based service (LBS) providers in exchange for the services they offer. This process is referred to as querying a service from the LBS provider. An example of LBS application is Google Maps, with over 2 billion monthly users in 2018. Perhaps, it is not a surprise to know that the annual market for LBSs is expected to reach 77.84 billion US dollars by 2021, according to 'Research and Markets' report [2]. Fig. 1.1 categorizes the services LBS applications provide based on their most widely used applications. On the service provider side, location data are captured in trajectories of moving objects and stored in datasets. Each entry of the dataset indicates a path traveled by a user ordered based on the time of queries.

In spite of numerous advantages that the LBS applications provide for their users, they are associated with a number of location privacy concerns that severely compromise the

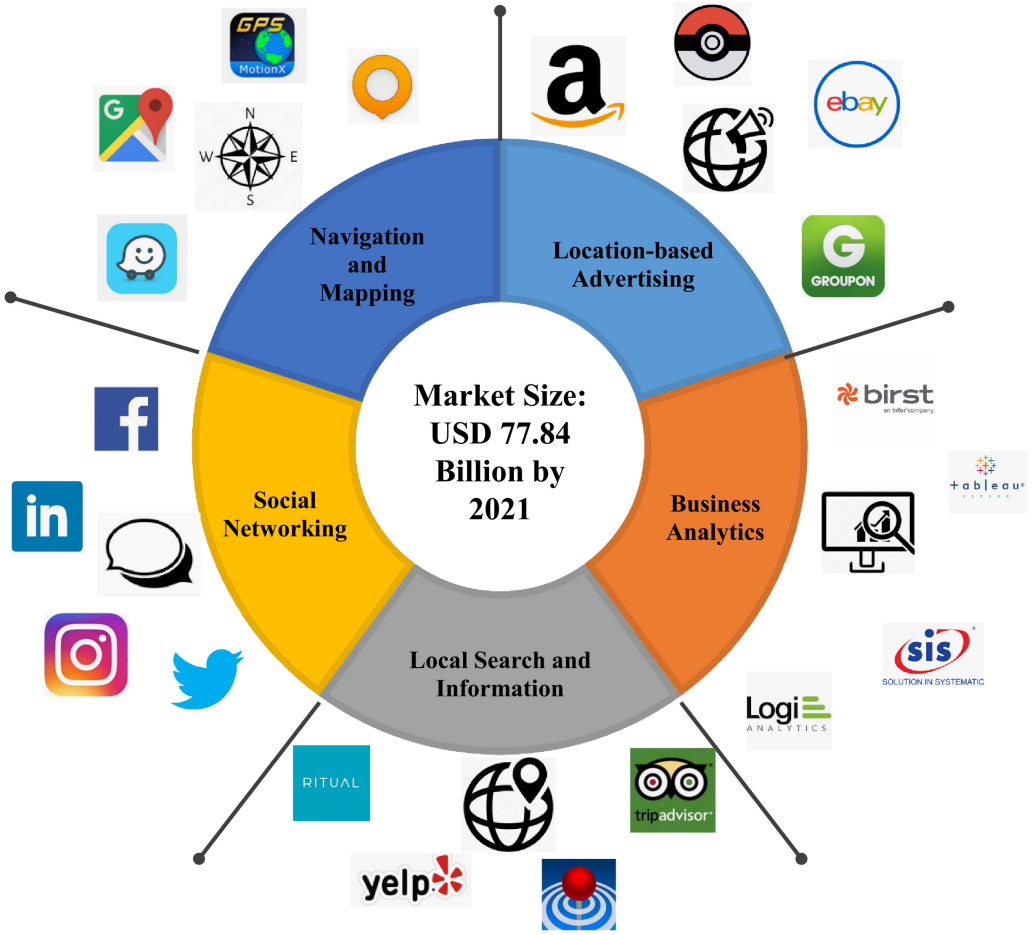


Fig. 1.1 Classification of the LBS apps based on their application.

privacy of users. If ill-intended individuals or organizations (adversaries) have access to the location data of users, they can compromise the privacy of users. Adversaries may unveil sensitive information, such as home address [3], workplace [4], health conditions [5], children's schools [6], and daily shopping habits [7].

Privacy threats become more drastic if the user locations are stored along with other sources of information. For instance, consider an LBS application that daily records the health conditions of its users and stores them along with their location coordinates [8–10]. If the LBS provider publishes the stored dataset without applying any anonymization technique, adversaries may be able to identify individuals and infer their health status. To mention a few, adversaries can know about diseases users carry, the medication they use, or health conditions

that they may have. Therefore, it is crucial to anonymize the location data of users so that they can enjoy the benefits of LBS applications without compromising their privacy.

More formally, Beresford et al. [11] define location privacy as the ability to prevent other parties from learning one's current or past locations. In simple words, it refers to having control over on how our location data is being used. Recent misconducts in the US election and Facebook scandal has exposed the importance of privacy and in particular location privacy more than any time before in history [12]. Also, the most widely used metric for preserving the location privacy of users is called k-anonymity, in which the aims is to hide the location of users among at least $k-1$ other users to prevent malicious attacks on their privacy.

1.2 Research Problems

In this thesis, we consider the location privacy of individuals in telecommunication networks in two perspectives. In the first scenario, the location privacy of users is investigated in telecommunications networks, in which the LBS provider itself can be the source of threat. In this scenario, the aim is to hide the actual location of users to protect them from untrusted LBS providers. In the second scenario, the publication of location trajectory datasets is considered, in which the aim is to anonymize the dataset so that no individual can be identified in the dataset. In the following two subsections, these two scenarios and the existing research problems are elaborated.

1.2.1 Location Privacy in Telecommunication Networks

The architecture of the first scenario is shown in Fig. 1.2. In this scenario, the LBS users are directly in contact with the LBS provider with no middle-man or a third-party service provider. If the LBS provider is untrusted, it can collect the location data of users and analyze them to learn sensitive information, such as the type of queries submitted [3], shopping habits

of users [7], and the address of users' properties or workplaces [4]. Therefore, it is of great importance to devise new ways to preserve the location privacy of LBS application users.

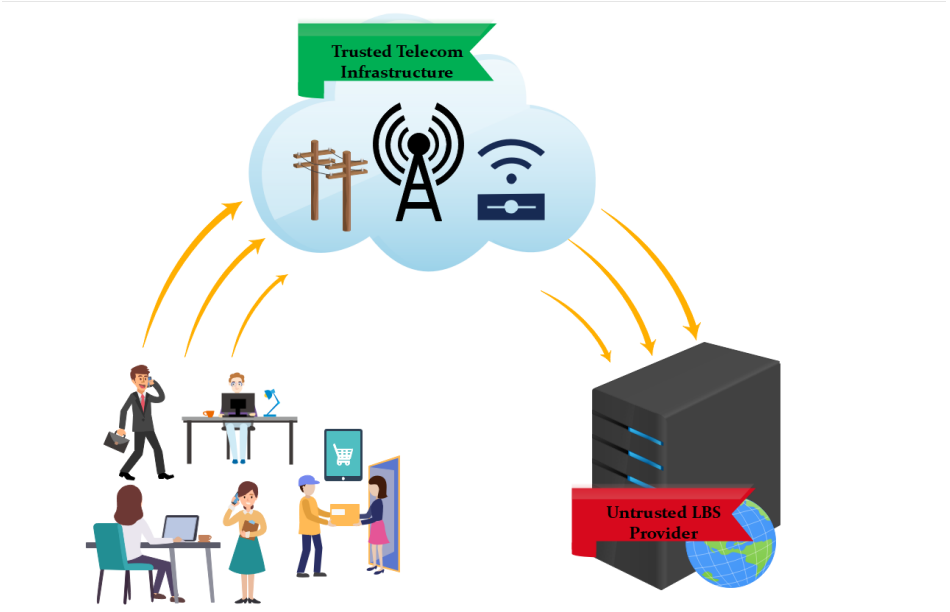


Fig. 1.2 Scenario 1: Location privacy of users in telecommunication networks.

1.2.2 Location Privacy in Publication of Location Datasets

The architecture used for the second scenario is shown in Fig. 1.3. This scenario considers the publication of location trajectory datasets to the public or third-parties. Despite numerous use cases that the publication of location data can provide to users and researchers, it poses a significant threat to users' privacy. As an example, consider a person who has been using GPS navigation to travel from home to work every morning of weekdays. If an adversary has some prior knowledge about the user, such as the home address, it may be able to identify the user. This can compromise private information about the user, such as the user's health condition and how often does the user visit his/her specialist. Therefore, it is crucial to anonymize spatiotemporal datasets before publishing them to the public.

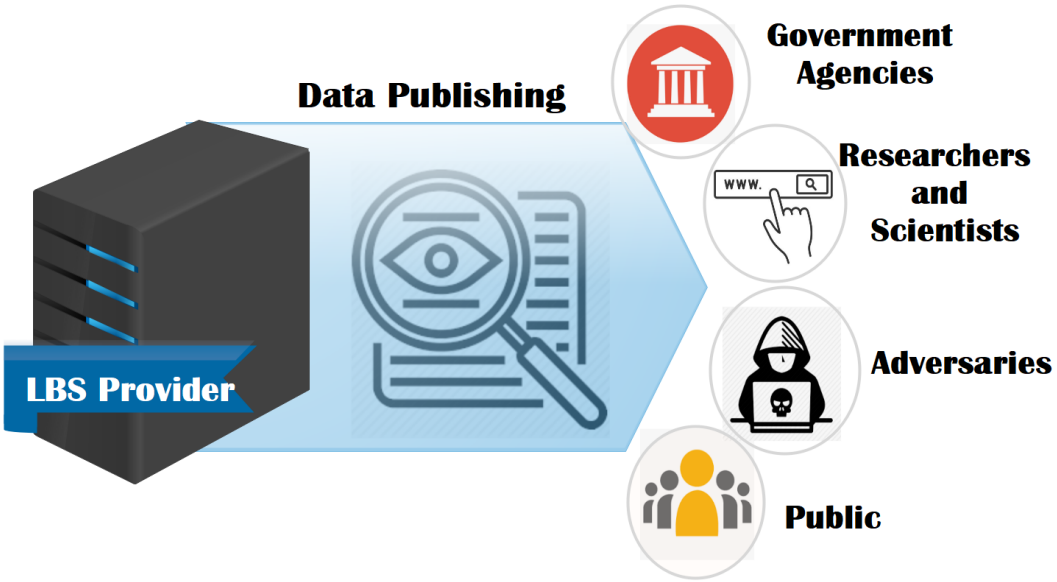


Fig. 1.3 Scenario 2: Location privacy in publication of location trajectory datasets.

1.3 Contribution of The Thesis and The Thesis Outline

In this thesis, improving the location privacy of users is considered in telecommunication networks as well as the publication of location datasets. With respect to the two major research problems, contributions are classified into the following two parts.

With regard to location privacy in telecommunication networks, we have made the following contributions:

- We propose a novel metric called the transition entropy, which considers the privacy of users in trajectories and not just the static snapshots of the queried locations. We explain the calculation of the metric for two consecutive locations and then expand it to paths with higher lengths. Moreover, we develop an exhaustive search algorithm to improve the transition entropy of the existing dummy-generation algorithms.

- We propose an attack model based on the Viterbi algorithm and term it as Viterbi attack. Based on our experiment, it is crucial to consider the Viterbi attack in the design of dummy-based algorithms as it can severely compromise the privacy of users.
- We propose an algorithm called robust dummy generation (RDG) that is resilient to the Viterbi attack while maintaining the high performance in terms of the traditional cell entropy metric in addition to having a robust performance in terms of transition entropy.
- We compare and evaluate the performance of the proposed metrics and algorithms on a publicly available dataset published by Microsoft, i.e., Geolife dataset.

For the publication of location datasets without compromising the privacy of users, our contributions are as follows:

- We propose to use k'-means algorithm for trajectory clustering and develop a technique to enable it. We also propose a variation of k'-means algorithm to preserve user privacy in overly sensitive spatio-temporal trajectory datasets.
- We propose to use a method termed the progressive sequence alignment for alignment of the trajectories in each cluster.
- We propose a privacy metric to evaluate and compare generalization algorithms based on the released area by data generalization.

Chapter 2

Background

In this chapter, comprehensive background knowledge and literature review associated with the two research problems explained in the previous chapter are provided and elaborated.

2.1 Location Data in Telecommunication Networks

Anonymity is defined as “the state of being not identifiable within a set of subjects, the anonymity set" [13]. Also, the location of a user is said to be k-anonymous if it is not distinguishable from at least $k-1$ other user locations [14]. To obtain k-anonymity for users, several approaches have been proposed, from which we have identified four broad categories: location cloaking, mixed-zones, pseudonyms, and dummy aided algorithms. The location cloaking technique is based on requesting LBSs for an area consisting of k locations via a trusted party, mixed-zones are predicated on anonymous regions for users, the pseudonyms approach takes advantage of fake IDs for users, and finally, the dummy generation algorithms query fake locations to confuse adversaries.

Gruteser and Grunwald [15] initiated the research on location cloaking. The key idea is to employ a trusted server in order to aid users become k-anonymous. Upon receiving a query from a user, the location anonymizer server computes a cloaking box including the location of

the user and $k-1$ other user locations and queries the requested service from the LBS provider for all the k locations. Therefore, making it difficult for the LBS provider to identify the user [16, 17]. Several algorithms have been proposed to implement location cloaking scheme such as ICliqueCloak [18] and MaxAccuCloak [19]. The main drawback of the location cloaking is the need for a location anonymizer, which is an additional cost overhead to the system. Also, the location anonymizer can become a data privacy threat itself.

The authors in [20] proposed the idea of mixed zones. Mixed zone is defined as the spatial zone where the identity of users is not identifiable. All users entering into a mixed zone will change their pseudonym to a new unused pseudonym making it difficult for adversaries to identify the users. The anonymization process is performed by a middle-ware mechanism before transferring the data to third-party applications. The authors further extended their work in [21] by considering irregular shapes for mixed zones. Moreover, the use of mixed zones has particularly attracted attention in vehicular communications. Applying mixed zones on road networks is considered in [22, 23], where a mixed zone construction method called MobiMix is proposed. Lu et al. [24] exploited the pseudonym changes in mixed zones at social spots, and Gao et al. [25] applied mix zones approach on trajectories for mobile crowd sensing applications. Furthermore, the use of cryptography for the generation of mixed zones in vehicular communications is considered in [26]. As it is the case for location cloaking approach, the main drawback of mixed zones is also the need for a middle-ware mechanism or a trusted party before transferring the data to an untrusted LBS provider.

Another technique to increase the location privacy of users is based on the assignment of pseudonyms to hide the identity of users. The identity of a user can be the name of the person, a unique identifier, such as IP address, or any properties that can be related to the user. The authors in [27] proposed a scenario called the intermediary scenario, in which a trusted intermediary collects the location information of users, such as GPS data and assigns a pseudonym before sending them to an untrusted third-party LBS provider. The paper claims

that the use of pseudonyms prevents the third-party LBS provider from identifying and tracking users. The work in [28] suggests that instead of delegating the generation of pseudonyms to the location intermediary, users are suggested to generate the pseudonyms themselves. The use of pseudonyms for preserving the location privacy has also been considered in vehicular communication systems, such as the work presented in [29]. There are several drawbacks associated with this approach. First of all, many of the location-based applications require users to subscribe in order to use services. Secondly, similar to the last two categories, this approach also requires a trusted intermediary, and more importantly, by analyzing the patterns in location data, an adversary can discover the identity of the users [30].

The dummy-based algorithms are considered to be a more promising approach as there is no need for a trusted anonymizer. This technique was initially proposed in [31]. The key idea is to achieve k-anonymity by sending $k-1$ dummy locations aside from the real location of the user while requesting for a service. All locations use the same identifier corresponding to the user, and therefore, it would be difficult for adversaries to identify the real locations of users. Several algorithms have been proposed to help users generate dummies. The authors in [32] proposed to use a virtual circle or a virtual grid that is based on the real location of users to generate dummies. The idea was further developed in [33]. More recently, an algorithm called dummy-location selection (DLS) was proposed in [34]. The algorithm takes the number of queries made on the map into consideration and demonstrates via simulations that the previous algorithms are susceptible to probability attacks. Although the algorithm provides an excellent framework for the generation of dummies, it does not take into account the susceptibility of users in trajectories and the privacy threats associated with that. Do et al. [35] utilized conditional probabilities to generate realistic false locations, and Hara et al. [36] proposed a method based on physical constraints of the real environment.

2.2 Publication of Location Data

In the second scenario considered in this thesis, the service provider aims to publish location datasets to the public or third parties. Unfortunately, merely removing unique identifiers of users cannot protect their privacy, as databases can be linked to each other based on their quasi-identifiers. Doing so, adversaries can reveal sensitive information about the users and compromise their privacy.

2.2.1 Generalization Technique

Generalization is currently one of the mainstream approaches for the anonymization of spatiotemporal trajectory datasets. The generalization technique is predicated on two interrelated mechanisms: clustering and alignment. Clustering aims at finding the best grouping of trajectories that minimizes a predefined cost function, and the alignment process aligns trajectories in each group.

The notion of k-anonymity was adopted in [37] for anonymization of spatiotemporal datasets. The authors proved that the anonymization process is NP-hard and followed a heuristic approach to cluster the trajectories. The use of 'edit distance' metric for anonymization of spatiotemporal datasets was proposed in [38]. In this work, the authors target grouping the trajectories based on their similarity and choose a cluster head for each cluster to represent the cluster. Also, dummy trajectories were added to anonymize the datasets further. Yarovoy et al. [39] proposed to use Hilbert indexing for clustering trajectories. The authors in [40, 41] chose to avoid alignment by selecting trajectories with the highest similarity as representatives of clusters. Poulis et al. [42] investigated applying restriction on the amount of generalization that can be applied by proposing a user-defined utility metric. Takahashi et al. [43] proposed an approach termed as CMAO to anonymize the real-time publication of spatiotemporal trajectories. The proposed idea is based on generalizing each queried location point with $k-1$ other queried location by other users, and hence, achieving k-anonmity.

The current state-of-art technique for applying gerelization to spatiotemporal datasets is based on domain generalization hierarchy (DGH) trees. In essence, DGH can be seen as a coding scheme to anonymize trajectories. We have categorized types of DGHs in the literature as:

- Full-domain generalization: This technique emphasizes on the level that each value of an attribute is located in the generalization tree. If a value of an attribute is generalized to its parent node, all values of that attribute in the dataset must be generalized to the same level [44–46].
- Subtree generalization: In this method, if a value of an attribute is generalized to its parent node, all other child nodes of that parent node need to be replaced with the parent node as well [47, 48].
- Cell generalization: This generalization technique considers each cell in the table separately. One cell can be generalized to its parent node while other values of that attribute remain unchanged [49–51].

2.2.2 Other Anonymization Techniques

Aside from the generalization technique, we have categorized the existing methods for the anonymization spatiotemporal datasets into three major groups:

- **Perturbation** anonymizes location datasets by addition of noise to data;
- **ID swapping** swaps user IDs in road junctions to anonymize location datasets;
- **Splitting** divides trajectories into shorter lengths to anonymize location datasets.

The authors in [52] proposed an algorithm that swaps the IDs of users in trajectories once they reach an intersection. Doing so, the algorithm prevents adversaries from identifying a

particular user. Cicek et al. [53] made a distinction between sensitive and insensitive location nodes of trajectories. Their proposed algorithm only groups the paths around the sensitive nodes and exploits generalization to create supernodes.

Moreover, Cristina et al. [54] shifted the burden of privacy preservation in data publishing to the user side. The authors attempted to anonymize the data on the mobile phones before storage on the database as they would have more control over their privacy. Instead of clustering trajectories for anonymization, Cicek et al. in [53] focused on the obfuscation of underlying map for sensitive locations. Brito et al. [55] minimized the information loss during the data anonymization by suppressing key locations. The Local suppression and splitting techniques were considered for trajectory anonymization in [56]. Although the proposed approach is useful for a predefined number of locations, it cannot be generalized to system models in which the users can make queries from an arbitrary location on the map. Naghizadeh et al. [57] focused on the stop points along trajectories. A sensitivity measure is introduced in this work, which relies on the amount of time users spend in different locations. Sensitive locations are replaced or displaced with a less sensitive location to preserve the privacy of users. Jiang et al. [58] considered the perturbation of locations by adding noise to preserve the privacy of users. Adding noise can generate fake trajectories that do not correspond to realistic scenarios.

Chapter 3

Location Privacy in Mobile Networks

3.1 Introduction

With the ubiquitous use of smartphones and social networks, location-based services (LBSs) have become an essential part of contemporary society. The users of smart devices can download LBS applications from Google Play or Apple Store, and query for LBSs they desire. For example, users can query their locations from an LBS provider to find restaurants nearby [59], refine route planning [60], and receive location-based advertisements [61]. The annual market for LBSs is expected to reach 77.84 billion US dollars by 2021, with an annual growth rate of 38.9% [2].

Unfortunately, the privacy issues associated with the LBSs have raised many concerns. Notably, after the recent Facebook data privacy scandal occupying the headlines of major media [62]. Different from the security of data, which is mainly concerned with secure encryption and integrity, privacy indicates how in control users are to prevent the leakage of their data; Can LBS providers analyze users' locations to find out their home address? Can LBS providers take advantage of users' data to figure out their shopping habits? Can LBS providers share user data with third-parties? And these are just some of the issues that may compromise the location privacy of users.

Krumm et al. [63] warn about the current location privacy threats. The authors show that just by having the last location of a day, it is possible to estimate the home location within 60 meters of the actual site. The authors in [64] demonstrate that even when locations are queried from LBS providers as members of a community, sensitive locations associated with users can still be identified based on the distribution of queries. Beresford and Stajano [11] also warn that a system collecting users' locations may invade their location privacy. Therefore, it is crucial to devise new ways to preserve the location privacy of users formally defined as "the ability to prevent other parties from learning one's current or past locations" [65].

Researchers have proposed several approaches to preserve the location privacy of users, among which dummy-based algorithms have drawn a great deal of attention [66–69, 65, 70–73]. For a given user location, the dummy generation algorithms aim at generating $k-1$ dummy locations aside from the actual location of the user and submitting them all together to the LBS server. Thus, making it difficult for untrusted servers, or so-called adversaries, to identify the actual user location. All algorithms are executed in the application layer of mobile phones before sending queries to LBS providers. The groundwork in this field was laid by the authors of [31]. They generated dummies randomly throughout the map and evolved them as users move. Followed by this work, the authors in [32] and [33] proposed to choose the candidate dummies from a virtual circle or grid constructed around the current location of the user.

More recently, an enhanced algorithm was proposed in [34], termed as the dummy-location selection (DLS) algorithm. The algorithm considers the likelihood of locations being real or fake predicated on the history of queries on the map. The basic idea of the DLS algorithm can be explained intuitively in Fig. 3.1. Assume that a user is at location A and a dummy generation algorithm is required to generate one dummy to preserve the location privacy of user shown by A'. The DLS algorithm argues that A' cannot just be any point on the map but a location that has a similar likelihood of being queried as to the location A. Such a likelihood can be calculated from the history of queries on the map. For instance, if the location A has been queried 1000

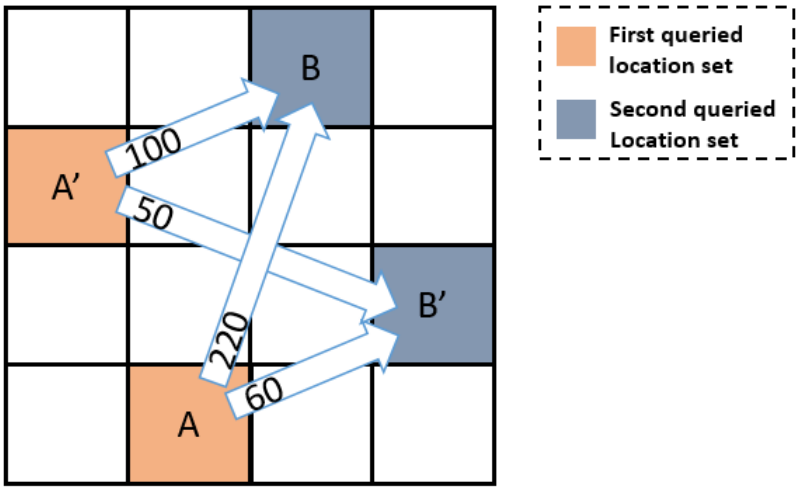


Fig. 3.1 An example of location privacy of the user being compromised by considering the introduced side information.

times, and A' has been queried only 5 times, the LBS provider can infer with a high likelihood that the location A is the real location of the user. Based on this logic, the DLS algorithm attempts to select dummies with the same likelihood as the actual locations. Information such as the query likelihood, and extra details that adversaries may know are usually referred to as 'side information'. Unfortunately, the DLS algorithm overlooks a significant piece of side information, which can severely compromise the location privacy, as explained in the following.

Suppose that the user A moves to location B and the DLS algorithm generates another dummy B' associated with the location B. Based on the history of trajectories traveled on the map, the adversary may know the likelihood of paths which have been traveled by users. For instance, the location B has been queried sequentially after the location A for 220 times. This is shown by a directed arrow connecting A to B in the map. Let us now look at the four directed edges connecting the two sets of locations and consider the number of times that each path has been traveled. It can be seen from Fig. 3.1 that in total, location B has been called 320 times after locations A and A', whereas location B' has been queried only 110 times. Therefore, the adversary can infer with a high likelihood that the real location is possibly location B, and thus, compromise the location privacy of user.

In this work, we study the impact of such side information on the location privacy of users. Compared with the existing literature, the main contributions of this study are presented as follows:

- We propose a novel metric called transition entropy. This metric investigates the privacy of users in trajectories as well as static queries. To improve the transition entropy of already existing algorithms, we have also developed an exhaustive framework that can improve transition entropy for a given algorithm.
- We show the susceptibility of user privacy in trajectories by developing an attack model based on Viterbi Algorithm.
- We propose an algorithm called robust dummy generation (RDG) that is resilient to the Viterbi attack. Moreover, this algorithm maintains high cell entropy performance as well as higher levels of transition entropy.
- We compare and evaluate the performance of the proposed metrics and algorithms on a publicly available dataset published by Microsoft, i.e., Geolife dataset.

3.2 System Model and Problem Formulation

3.2.1 System Architecture

Following the recent standards and the current system designs used in the telecommunications industry [74, 41, 75, 76], we adopt a non-cooperative system architecture as shown in Fig. 3.2. In this design, there are two main parties involved: LBS users and an LBS server. There is also the telecommunication infrastructure in between which works as a medium for communications between the two parties. The role of each party is explained in the following.

1) LBS users: The system model consists of multiple users equipped with mobile phones with embedded GPS modules. Users can benefit from numerous LBSs provided by LBS servers

via a variety of LBS applications that can be downloaded and installed. Regardless of whether applications require users to log in to the system or not, the users request for services by providing their (I) identifiers such as IP address, username, etc. (II) location information (III) type of services (IV) some dummy locations to hide their exact locations. Moreover, in this work, we focus on 'explicit' trajectory data in which queries are made at uniform time intervals. GPS data is the most representative example of explicit trajectory data which is widely used in researches of trajectory analysis [77–79].

2) LBS server: The LBS provider is responsible for providing queried services by users. The LBS server is capable of storing the queried information and may have access to other databases and side information. This configuration enables the LBS server to infer historical query probabilities of users. After each query from a user, the server stores the requested information and updates the database accordingly.

3) Intermediary infrastructure: The queried services from the LBS server are transmitted through telecommunications infrastructure. The telecommunications infrastructure is controlled by mobile operators and regulated by government agencies [80, 41]. Therefore, such infrastructure is considered to be trusted in the system model. Admittedly, this assumption might not hold for untrusted operators and governments that violate the privacy of users in the name of national security. Such a consideration is out of the scope of our work here.

3.2.2 Preliminaries

Assume that the location map is divided into an $n \times n$ grid, and a user communicates with an LBS server for service. At the time t^q, the user intends to make his/her q-th query from the service provider, preserving k^q-anonymity. Here, k^q quantifies the privacy protection requirement of the user. This metric implies that the adversary is not able to identify the real location of the user with a probability higher than $1/k^q$. Hence, such a user needs to transmit $k^q - 1$ dummy locations to hide his/her true location from the observer. Note that the

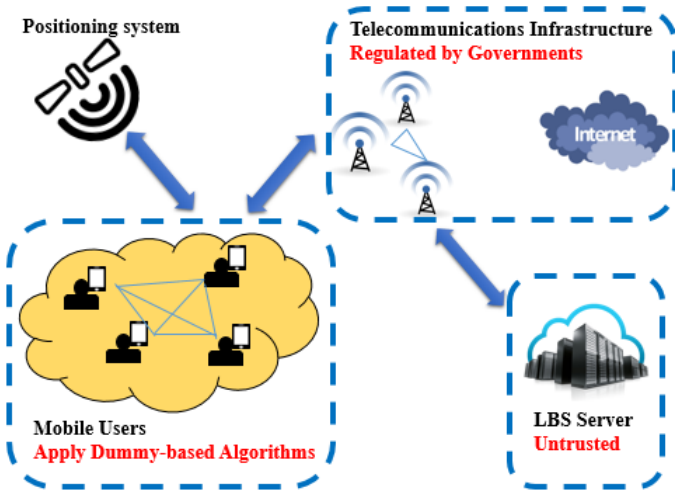


Fig. 3.2 System architecture of LBSs.

term 'location' refers to the cell in which the user is located. We denote the set of locations transmitted to the LBS provider at q-th query by

$$LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}. \tag{3.1}$$

Also, the real location is shown by r^q, where $r^q \in LS^q$. The probability of location l_x^q being the real location can be expressed as

$$\Pr(l_x^q = r^q), \ \forall x = 1, ..., k^q. \tag{3.2}$$

In the next query, the user requires k^{q+1}-anonymity and queries the location set $LS^{q+1} = \{l_1^{q+1}, l_2^{q+1}, ..., l_{k^{q+1}}^{q+1}\}$ from the LBS provider. The probability of $l_y^{q+1} \in LS^{q+1}$ being queried consecutively after $l_x^q \in LS^q$ is denoted by

$$\Pr(l_x^q \Rightarrow l_y^{q+1}). \tag{3.3}$$

3.2.3 Cell Entropy Metric

The cell entropy metric was implicitly proposed as part of the DLS algorithm in [34]. The metric is predicated on two factors: query probabilities of cells and the concept of entropy explained as follows.

For a given location set $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$ which includes the real location of a user and $k^q - 1$ dummies chosen to preserve k^q-anonymity, the set of query probabilities are shown by $B^q = \{b_1^q, b_2^q, ..., b_{k^q}^q\}$, where b_j^q is the query probability of location (cell) l_j^q for $j = 1, 2, ..., k^q$. The query probability of cell l_j^q is calculated by

$$b_j^q = \frac{\text{number of queries in } l_j^q}{\text{number of queries in the whole map}}. \tag{3.4}$$

The cell entropy borrows the concept of entropy from information theory to quantify uncertainty in query probabilities. The cell entropy metric for location set LS^q can be calculated as [34]

$$h_c = -\sum_{j=1}^{k^q} b_j^q \log_2(b_j^q). \tag{3.5}$$

3.2.4 Adversary Model

Two types of adversary models are considered in our work: active adversary, and passive adversary. A passive adversary can listen to the communications between the users and the LBS provider. The passive adversary can compromise the location privacy of users by performing an eavesdropping attack and analyzing the collected information. An active adversary, on the other hand, compromises the LBS provider and has access to the data stored on the server.

3.2.5 Side Information

There are several side information that adversaries may possess to compromise the location privacy of users. Adversaries may know about the probability of a query being made in different

locations of the map. For instance, if a location has been queried five times among the overall 1000 queries made on the map, its query probability can be calculated as $5/1000$. Exploiting query probabilities, adversaries can understand the likelihood of locations being genuine or fake. For instance, if a user queries two locations at the same time, one with a comparably higher probability, it is more likely that the real location is the one with the higher probability.

Query probability has always been a critical consideration in the generation of dummy locations. In this work, apart from the possession of traditional side information by adversaries, we consider another prominent side information that can severely compromise the privacy of users. That is, the trajectories users have traveled, which reveals how many time a location has been queried after its neighbor locations. Authorities do not specify any time limit for storing the location information of the users, as it is the case in the US [81]. This lack of legislation enables adversaries to monitor users and get access to trajectories they travel.

3.3 Transition Entropy

In this section, we propose a metric called transition entropy to quantify privacy preservation in LBSs. We first explain the metric for two consecutive queries, then, expand it to trajectories with higher lengths. This metric quantifies the privacy of users in trajectories and can be used as a benchmark to compare and evaluate the performance of dummy-based algorithms. Moreover, transition entropy necessitates the development of new algorithms, as it reveals the susceptibility of user privacy.

3.3.1 Transition entropy metric for two consecutive queries

Consider q-th and $(q+1)$-th query of a user from the LBS provider. In the q-th query, the user requests service for the location set $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$ including $k^q - 1$ dummies and the real location of the user to achieve k^q-anonymity; followed by, moving to a new location with the

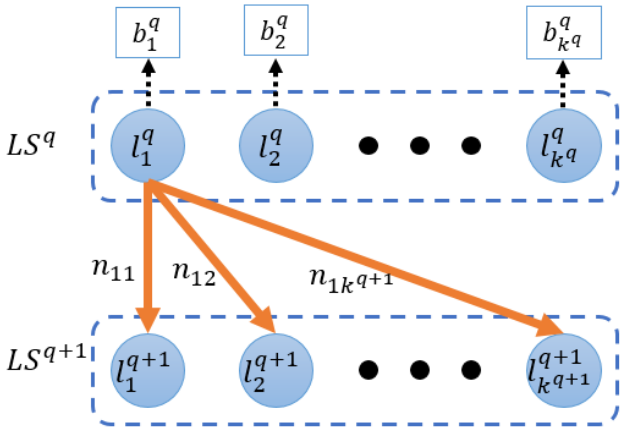


Fig. 3.3 Bipartite graph generated by two consecutive queries of a user.

anonymity constraint of k^{q+1} and making its $(q+1)$-th query. Dummies can be generated using any of the existing algorithms. Based on the sets LS^q and LS^{q+1}, we generate a bipartite graph shown in Fig. 3.3, where each set forms vertices at one side of the bipartite graph. Looking at the history of queries on the map, we denote the number of times location $l_y^{q+1} \in LS^{q+1}$ has been queried after location $l_x^q \in LS^q$ by n_{xy}, and assign it to the directed edge connecting l_x^q to l_y^{q+1}. Also, as explained in the system model section, for every location $l_x^q \in LS^q$, we denote the query probability of location l_x^q by b_x^q. Query probabilities are also calculated from the historical data stored at the LBS provider.

Our goal is to find out how probable it is for each member of the location set LS^{q+1} to be the real location, given the location set LS^q. In other words, the aim is to calculate the posterior probability of members in LS^{q+1} with respect to LS^q. This probability for each member of LS^{q+1} can be written as

$$\forall\, l_y^{q+1} \in LS^{q+1}:$$

$$\Pr(l_y^{q+1} = r^{q+1} | LS^q) = \tag{3.6}$$

$$\sum_{s=1}^{k^q} \Pr((l_s^q \Rightarrow l_y^{q+1}), (l_s^q = r^q)) = \tag{3.7}$$

$$\sum_{s=1}^{k^q} \Pr(l_s^q \Rightarrow l_y^{q+1} | l_s^q = r^q) \Pr(l_s^q = r^q), \tag{3.8}$$

where (3.7) is the joint probability of l_s^q being the real location of LS^q, and moving to the location l_y^{q+1} after l_s^q. The former probability in (3.8) can be calculated as

$$\forall\, l_y^{q+1} \in LS^{q+1},\ \forall\, l_x^q \in LS^q:$$

$$\Pr(l_x^q \Rightarrow l_y^{q+1} | l_x^q = r^q) = \frac{n_{xy}}{\sum_{y=1}^{k^{q+1}} n_{xy}}, \tag{3.9}$$

and the latter probability indicates the normalized query probability and is given by

$$\forall\, l_x^q \in LS^q:\ \Pr(l_x^q = r^q) = \frac{b_x^q}{\sum_{j=1}^{k^q} b_j^q}. \tag{3.10}$$

Note that (3.10) indicates that the posterior probabilities of cells in LS^q are set to the normalized query probability of the locations in LS^q. By calculating (3.8) for every member of LS^{q+1}, the posterior probabilities of locations in LS^{q+1} are determined based on LS^q. Having these probabilities, we exploit the concept of entropy to infer the uncertainty in identifying dummies. The entropy can be derived by

$$h_t = -\sum_{y=1}^{k^{q+1}} \Pr(l_y^{q+1} = r^{q+1} | LS^q) \log_2(\Pr(l_y^{q+1} = r^{q+1} | LS^q)). \tag{3.11}$$

Algorithm 1: Transition entropy for two consecutive queries.

Input: LS^q and LS^{q+1}
Output: h_t
Initialization: $CellSum = 0$, $h = 0$.
for $1 \leq x \leq k^q$ **do**
  $EdgeSum = 0$
  for $1 \leq y \leq k^{q+1}$ **do**
    $EdgeSum = EdgeSum + n_{xy}$
  end
  for $1 \leq y \leq k^{q+1}$ **do**
    $\Pr(l_x^q \Rightarrow l_y^{q+1} | l_x^q = r^q) = n_{xy}/EdgeSum$
  end
end
for $1 \leq x \leq k^q$ **do**
  $CellSum = CellSum + b_x^q$
end
for $1 \leq x \leq k^q$ **do**
  $\Pr(l_x^q = r^q) = b_x^q/CellSum$
end
for $1 \leq y \leq k^{q+1}$ **do**
  $\Pr(l_y^{q+1} = r^{q+1} | LS^q) = 0$
  for $1 \leq x \leq k^q$ **do**
    $\Pr(l_y^{q+1} = r^{q+1} | LS^q) = \Pr(l_y^{q+1} = r^{q+1} | LS^q)$
      $+\Pr(l_y^{q+1} = r^{q+1} | l_x^q = r^q)\Pr(l_x^q = r^q)$
  end
  $h_t = h_t -$
    $\Pr(l_y^{q+1} = r^{q+1} | LS^q) \log_2(\Pr(l_y^{q+1} = r^{q+1} | LS^q))$
end
return h_t

We define h_t as the transition entropy of the location set LS^{q+1} with respect to LS^q. The transition entropy metric reveals the uncertainty in identifying the real location by adversaries. Having a higher transition entropy reveals that for each member of LS^{q+1}, the probability of paths originating from LS^q to the destination of that member is similar to the other members of LS^{q+1}. Hence, it would be more difficult for the adversary to compromise k^{q+1}-anonymity of the user. The formal algorithm for computing the transition entropy in two consecutive queries is presented in Algorithm 1.

Algorithm 2: Transition entropy for trajectories of length $c+1$.

Input: $LS^q, LS^{q+1}, ..., LS^{q+c}$
Output: h_t
Start:
Run Algo. 1 for LS^q and LS^{q+1}
for $q+1 \leq query \leq q+c-1$ **do**
    Normalize posterior probabilities of LS^{query}
    Query probabilities of $LS^{query} \leftarrow$ posterior probabilities of LS^{query}
    Run Algo. 1 for LS^{query} and $LS^{query+1}$
end
$h_t \leftarrow$ Normalize posterior probabilities of LS^{q+c} and calculate their entropy
return h_t

The main advantages of the transition entropy metric are:

- considering the performance of the dummy based algorithms in trajectories and not just for a stationary set of locations.
- being able to investigate the performance of the dummy based algorithms for users with varying k-anonymity requirements in their trajectories.
- elimination of the need for many other previously considered factors, such as time reachability and direction similarity.

3.3.2 Transition entropy metric for trajectories

Here, we generalize the transition entropy metric for trajectories with different lengths. Consider a user requesting for its $(c+1)$-th query at time t^{q+c}. Hence, providing the LBS provider with the location set $LS^{q+c} = \{l_1^{q+c}, l_2^{q+c}, ..., l_{k^{q+c}}^{q+c}\}$ in order to preserve k^{q+c}-anonymity. The previous queried location sets of the user are shown by LS^{q+i} for $i = 0, ..., c-1$, each with the privacy requirement shown by k^{q+i}. Initially, we aim to calculate the posterior probability of each location in LS^{q+c}. The posterior probabilities indicate the likelihood of any location in LS^{q+c} being the real location based on the previous queries of the user. The posterior probability for each location in LS^{q+c} can be written as

$$\sum_{s^c=1}^{k^{q+c-1}} \sum_{s^{c-1}=1}^{k^{q+c-2}} \dots \sum_{s^1=1}^{k^q} \Big(\Pr(l_{s^1}^q = r^q)\Pr(l_{s^c}^{q+c-1} \Rightarrow l_y^{q+c} | l_{s^c}^{q+c-1} = r^{q+c-1}) \times \prod_{i=1}^{c-1} \Pr(l_{s^i}^{q+i-1} \Rightarrow l_{q+i}^{q+i+1} | l_{s^i}^{q+i-1} = r^{q+i-1})\Big) \tag{3.15}$$

$$\forall\, l_y^{q+c} \in LS^{q+c} : \Pr(l_y^{q+c} = r^{q+c} | LS^q, ..., LS^{q+c-1}) = \tag{3.12}$$

$$\sum_{s^c=1}^{k^{q+c-1}} \Pr((l_{s^c}^{q+c-1} \Rightarrow l_y^{q+c}),$$

$$(l_{s^c}^{q+c-1} = r^{q+c-1}) | LS^q, ..., LS^{q+c-2})) = \tag{3.13}$$

$$\sum_{s^c=1}^{k^{q+c-1}} \Pr(l_{s^c}^{q+c-1} \Rightarrow l_y^{q+c} | l_{s^c}^{q+c-1} = r^{q+c-1}) \times$$

$$\Pr(l_{s^c}^{q+c-1} = r^{q+c-1} | LS^q, ..., LS^{q+c-2}). \tag{3.14}$$

Following the same process of moving from (3.12) to (3.14), the probability of $\Pr(l_{s^{c-1}}^{q+c-1} = r^{q+c-1} | LS^q, ..., LS^{q+c-2})$ can be solved recursively to reach (3.15). Also, the transition probabilities in (3.12) can be calculated as (3.9). Therefore, evaluating (3.15) for each node in LS^{q+c}, we can determine the likelihood of a location being the real location of the queried set LS^{q+c}. Finally, we borrow the concept of entropy to characterize the uncertainty in probabilities of LS^{q+c}. So that:

$$h_t = -\sum_{y=1}^{k^{q+c}} \Pr(l_y^{q+c} = r^{q+c} | LS^q, ..., LS^{q+c-1})$$

$$\log_2(\Pr(l_y^{q+c} = r^{q+c} | LS^q, ..., LS^{q+c-1})). \tag{3.16}$$

We call h_t, the transition entropy of the set LS^{q+c} with respect to location sets $LS^q, ..., LS^{q+c-1}$. Our experiments will demonstrate that the proposed transition entropy metric shows the high possibility of revealing the real location of users from their previous queries made on the map. The algorithm to calculate the transition entropy metric is presented formally in Algorithm 4.

In the derivation of transition entropy, the only place in which query probabilities of locations play a role is in the first queried location set. The transitions between the queried locations determine the remaining factors. It is essential to understand why the query probabilities of the other locations on the path are not used in the calculation of transition entropy.

We explain the concept using an example. Fig. 3.4. demonstrates a case where a user requests an LBS in two consecutive queries. The numbers written on the nodes indicate the normalized query probability of locations, and the numbers printed on the edges indicate the normalized probability of that transition. Now, consider the calculation of LS^{q+1} based on the previous queried location set LS^q. The purpose of the example is to illustrate why the posterior probabilities calculated by previous queries for LS^{q+1} is more reliable than the query probabilities of locations in LS^{q+1}. First, let us calculate the posterior probabilities of LS^{q+1} and its entropy. According to (3.15), the posterior probabilities can be written as

$$\text{Posterior probability of A being the true location} = \frac{3}{5}\times\frac{1}{3}+\frac{1}{5}\times\frac{1}{4}+\frac{1}{5}\times\frac{1}{4}=\frac{6}{20} \tag{3.17}$$

$$\text{Posterior probability of B being the true location} = \frac{3}{5}\times\frac{1}{3}+\frac{1}{5}\times\frac{2}{4}+\frac{1}{5}\times\frac{3}{4}=\frac{9}{20} \tag{3.18}$$

$$\text{Posterior probability of C being the true location} = \frac{3}{5}\times\frac{1}{3}+\frac{1}{5}\times\frac{1}{4}=\frac{5}{20} \tag{3.19}$$

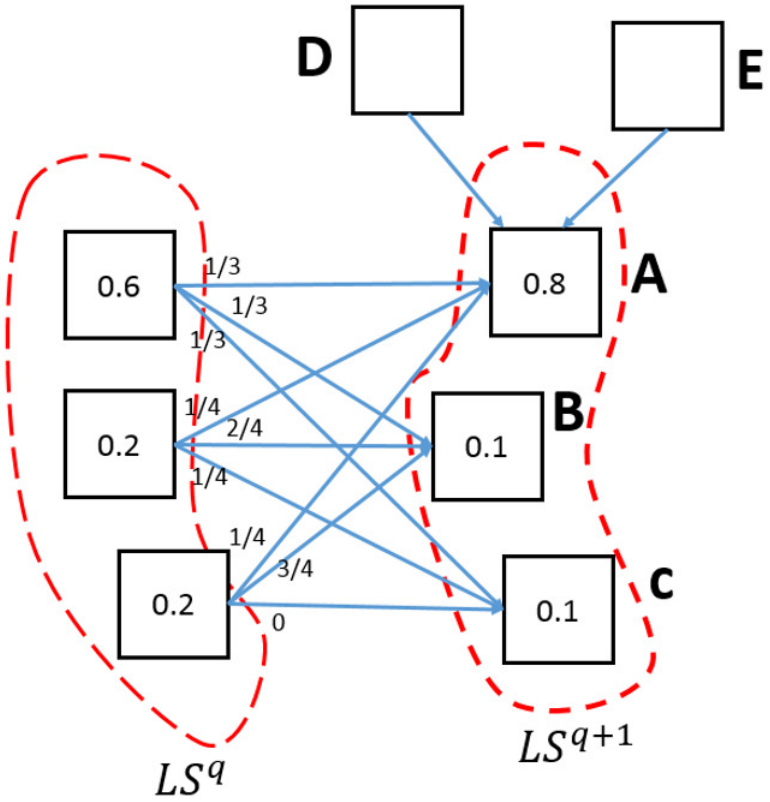


Fig. 3.4 An example of two consecutive queried location sets.

According to the query probabilities of LS^{q+1}, the location A is more likely to be the real location as it has a significantly higher query probability. However, looking at the posterior probabilities calculated for the location set, we can see that based on LS^q, location B is more probable to be the real location of the user. This discrepancy can be explained by looking at what the actual meaning of query probability is. The query probability indicates the number of times a location has been called but does not specify if it has been called after any particular location. Therefore, although the location A has been called more times than the other locations in LS^{q+1}, most of these queries have been made after locations E and D, which are not a member of the location set LS^q. Hence, it can be seen that the posterior probabilities are more credible, as they are considering the number of times queries made after the previous location set LS^q.

3.4 Viterbi Attack

The Viterbi algorithm is a well-known dynamic programming algorithm proposed in 1967 [82]. Initially, it was used for convolutional codes, but then it found numerous applications, such as exploring the most likely sequence of hidden states in Hidden Markov Models (HMMs). For a given graph, the aim of the algorithm is to find the shortest path or so-called Viterbi path. The Viterbi algorithm provides several features which distinguish this algorithm from the other existing algorithms for this purpose. The most essential characteristic of the algorithm is the low computational complexity. Here, we design an attack model based on the Viterbi algorithm and name it Viterbi attack, since the principal idea behind the attack is inspired by the Viterbi algorithm. The proposed Viterbi attack can significantly compromise the location privacy of users if it is not considered in the design of the dummy generation algorithms. As it will be demonstrated in simulations, even for short trajectories, the Viterbi attack can reveal a significant number of user locations.

Given the location sets LS^q, LS^{q+1},...,LS^{q+c}, corresponding to a trajectory of length $c+1$ of a user, an adversary seeks to find the most probable location sequence or so-called state sequence. Hence, the attacker aims to identify locations which are most likely to be the actual locations of the user and not the dummies. The desired state sequence of the adversary includes all the real locations of the user shown by $\{r^q, r^{q+1}, ..., r^{q+c}\}$.

We define $\mu(c+1,u)$ to be the maximum probability of a state sequence with the length of $c+1$, given $z^q, z^{q+1}, ..., z^{q+c}$ where $z^j \in LS^j$ and $z^{q+c} = u \in LS^{q+c}$. This function can be mathematically expressed as

$$\mu(c+1,u) = \max_{z^{q:q+m} | z^{q+m}=u} \Pr(z^{q+m} = r^{q+m}), \tag{3.20}$$

where for each $u \in LS^q$, and the initial value of the μ function is set to be

$$\mu(0,u) = \Pr(u = r^q). \tag{3.21}$$

As the most credible information for the first queried location set is the query probability, $\Pr(u = r^q)$ is calculated via equation (3.10). Starting from the second queried location set the most probable path can be calculated recursively as

$$\mu(m+1,u) = \max_{u' \in LS^{q+c-1}} \mu(c,u') \Pr(u' \to u). \tag{3.22}$$

Algorithm 3: Viterbi attack.

1 **Input:** Location sets LS^q, LS^{q+1},...,LS^{q+c} and the normalized query probability for the location set LS^q
2 **Output:** $EstState$ which is the most likely path
3 **Start:** .
4 **for** $1 \leq u \leq k^q$ **do**
5 $\quad \mu(q,u) = \Pr(l_u^q = r^q)$
6 $\quad pointer(q,u) = 0$
7 **end**
8 **for** $1 \leq j \leq c$ **do**
9 $\quad$ **for** $1 \leq u \leq k^{q+j}$ **do**
10 $\quad\quad \mu(q+j,u) = \max_{u' \in LS^{q+j-1}} \mu(q+j-1,u') \Pr(u' \to u)$
11 $\quad\quad pointer(q+j,u) \leftarrow$ state of $\max_{u' \in LS^{q+j-1}} \mu(q+j-1,u')$
12 $\quad$ **end**
13 **end**
14 $EstState[c] =$ state of $max(\mu(q+c,:))$
15 **for** $c-1 \geq j \geq 0$ **do**
16 $\quad EstState[j] = pointer(q+j+1, EstState[j+1])$
17 **end**
18 **Output:** $EstState$.

The formal presentation of Viterbi attack is given in Algorithm 3. The algorithm starts by setting the initial values of the μ array to their normalized query probability in lines $4-7$. An array called *pointer* is used to keep track of the most likely state of the previous queried

location set as the most probable path is calculated in lines 8 – 13. Finally, the most probable path is chosen and the corresponding states are returned as outputs.

3.5 Proposed Algorithms to Improve Location Privacy of Users

In this section, we start by developing an exhaustive search algorithm to improve the transition entropy metric for a given dummy-generation algorithm. We denote this hypothetical algorithm by X and aim at increasing its transition entropy in trajectories.

Next, we propose an algorithm called RDG that significantly increase the privacy of users against the Viterbi attack, while maintaining the high performance in terms of transition entropy and cell entropy. The basis of the RDG algorithm is an algorithm called DLS proposed in [34].

3.5.1 Exhaustive Search Algorithm

Algorithm 4: Exhaustive search algorithm

1 **Input:** $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, $\{l_1^{q+1}\}$, k^{q+1}
2 **Output:** LS^{q+1}
3 **Start:**
4 $D \leftarrow$ generate a pool of $4k^{q+1}$ dummies using the X algorithm
5 $\{S_1, S_2, ...S_m\} \leftarrow$ choose m distinct $(k^{q+1}-1)$-subsets of D
6 **for** $1 \leq y \leq m$ **do**
7 $\quad S_y \leftarrow S_y \cup \{l_1^{q+1}\}$
8 $\quad h_y \leftarrow$ calculate transition entropy of S_y
9 $\quad H \leftarrow H \cup \{h_y\}$
10 **end**
11 $LS^{q+1} \leftarrow S$ corresponding to the maximum h
12 **return** LS^{q+1}

Suppose that a user has made its q-th query shown by $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, which includes the real location and its associated dummies. The dummies in LS^q are generated using a given

algorithm X. In the next query, the user moves to a new location (l_1^{q+1}) and seeks to generate $k^{q+1}-1$ dummy locations. The following approach will help the user increase its transition entropy while generating LS^{q+1}.

The idea is to generate a pool of dummies based on the algorithm X instead of only $k^{q+1}-1$ dummy locations. Having the dummy pool, the exhaustive search algorithm goes through $k^{q+1}-1$ subsets of the pool to find the one that maximizes the transition entropy. The formal description of the proposed exhaustive approach is presented in Algorithm 4. The inputs of the algorithm are the location set $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, the real location of the user in $(q+1)$-th query, and the privacy requirement of the user in $(q+1)$-th query.

The exhaustive search algorithm starts by generating a pool of $4k^{q+1}$ dummies using the X algorithm and assigns them to an empty set D. Then, m distinct subsets of D with $(k^{q+1}-1)$ members are chosen and assigned to $\mathscr{S} = \{S_1, S_2, ...S_m\}$. Any of the members in $\mathscr{S}$, once attached to l_1^{q+1} will form a complete k^{q+1} set, preserving k^{q+1}-anonymity. Note that the constraint m is chosen to limit the number of subsets computed in case of a large pool size. Next, the transition entropies resulted from the members of $\mathscr{S}$ are calculated and stored in H. Finally, the member of S that results in the maximum transition entropy is returned as the output.

3.5.2 RDG Algorithm

We propose a robust algorithm called RDG to preserve the location privacy of LBS users. The RDG algorithm has three advantages compared with the existing algorithms: (I) Provides high resilience against the Viterbi attack (II) Achieves near-optimal cell entropy (III) Results in a much higher transition entropy compared with the existing approaches. The algorithm is based on the idea of posterior probabilities, and it is formally presented in Algorithm 5.

Following the same setup as the proposed exhaustive search algorithm, a user has made its q-th query shown by $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, which includes the real location and its associated

dummies. The dummies in LS^q are generated using the DLS algorithm. In the next query, the user moves to a new location (l_1^{q+1}) and seeks to generate $k^{q+1}-1$ dummy locations. If LS^q is the initial query of the user from the LBS provider, then the initial posterior probabilities are set to the normalized query probabilities of the locations in LS^q; otherwise, the posterior probabilities are calculated by (3.12). In the algorithm, posterior probabilities are assigned to an array called *weight*.

The algorithm starts with the generation of a pool of dummies using the DLS algorithm based on the real location of LS^{q+1}. Using the DLS algorithm to generate the pool of dummies will ensure high performance in terms of cell entropy. From our experiments, setting the pool size to four times of the k^{q+1} maintains the cell entropy sufficiently high, while resulting in a robust performance in terms of the transition entropy and Viterbi attack resilience. Next, the algorithm continues by employing a greedy approach to add the most suitable dummies for the location set LS^{q+1}. For choosing the i-th member of the set LS^{q+1}, each of the remaining dummies in the pool is checked one by one. A criterion chosen here is based on maximizing the entropy for the array *weight*. For each member $u \in LS^{q+1}$, the *weight* array is calculated as

$$weight(q+1,u) = \max_{u' \in LS^q} weight(q,u')\Pr(u' \to u). \tag{3.23}$$

The first index of the *weight* array is used to distinguish between weights corresponding to different location sets. For each member of the dummy pool, its weight is calculated, followed by the entropy of the weight array. After calculation of the entropy for all possible members, the member having the maximum entropy is chosen as the next member of LS^{q+1}. The process continues until all $k^{q+1}-1$ dummies of LS^{q+1} are chosen. Note that before the calculation of entropy, the weights are normalized to make the accumulation of probabilities add up to one. The algorithm is designed to provide a high cell entropy and transition entropy for users' of the LBS applications while protecting them from the Viterbi attack on trajectories.

Algorithm 5: RDG algorithm.

Input: $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, $\{l_1^{q+1}\}$, k^{q+1}
Output: LS^{q+1}
Start:
for $1 \leq u \leq k^q$ **do**
  $weight(q,u) \leftarrow$ Posterior probability of l_u^q
end
$D \leftarrow$ generate a pool of $4k^{q+1}$ dummies using the DLS algorithm
for $1 \leq member \leq k^{q+1} - 1$ **do**
  $entropy = zeros(1 \times |D|)$
  for $1 \leq d \leq |D|$ **do**
    $LS^{q+1} = LS^{q+1} \cup \{D[d]\}$
    for $1 \leq u \leq k^{q+1}$ **do**
      $weight(q+1,u) = \max_{u' \in LS^q} weight(q,u')\Pr(u' \rightarrow u)$
    end
    normalize $weight(2,:)$
    $entropy[d] \leftarrow$ entropy of $weight(q+1,:)$
    $LS^{q+1} = LS^{q+1} - \{D[d]\}$
  end
  $NewMember \leftarrow \{$member of D which maximize $entropy\}$
  $LS^{q+1} = LS^{q+1} \cup \{NewMember\}$
  $D = D - \{NewMember\}$
end
return LS^{q+1}

3.6 Performance Evaluation

3.6.1 Experimental Setup

In our experiments, we use the data collected by Geolife project [83–85]. The Geolife dataset includes the GPS trajectories of 182 users from April 2007 to August 2012 in Beijing, China. The dataset contains $17,621$ trajectories with a total distance of $1,292,951$ km. Two main advantages are distinguishing Geolife dataset for our work: Firstly, the recorded data aside from monitoring the daily routines of users, such as going to work or home, includes trajectories involving sports activities such as hiking and cycling. Secondly, many of the recorded

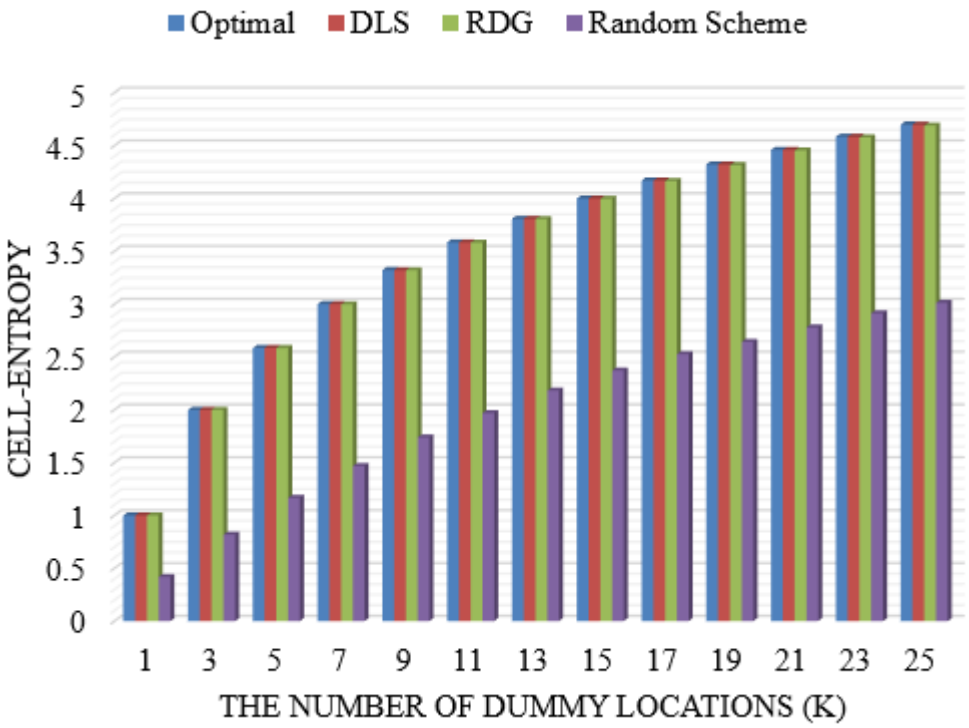


Fig. 3.5 Comparison of algorithms in terms of cell entropy for different k.

trajectories are tagged with transportation modes, which indicate the use of various means of traveling from bus and car to airplane and train.

We conducted our experiments on 1km$\times$ 1km central part on the Beijing map with the resolution of 0.01km$\times$0.01km for each grid cell. The location privacy requirements of users are investigated for values 2 to 30. For each value of k, the trial is repeated 3000 times to ensure the reliability of results. The experiments were performed on a PC with a 3.40 GHz Core-i7 Intel processor, 64-bit Windows 7 operating system, and an 8.00 GB of RAM. Python programming is used to implement algorithms.

3.6.2 Performance Analysis

We evaluate the performance of the proposed algorithms and metrics through extensive experiments. We intend to show that the proposed RDG algorithm can achieve:

- Near-optimal cell entropy;

- Robust transition entropy performance compared to prior works;
- Privacy protection against the Viterbi attack.

Therefore, in the following subsections, we start by evaluating the performance of algorithms in terms of cell entropy, followed by transition entropy analysis and investigating the resilience to the Viterbi attack.

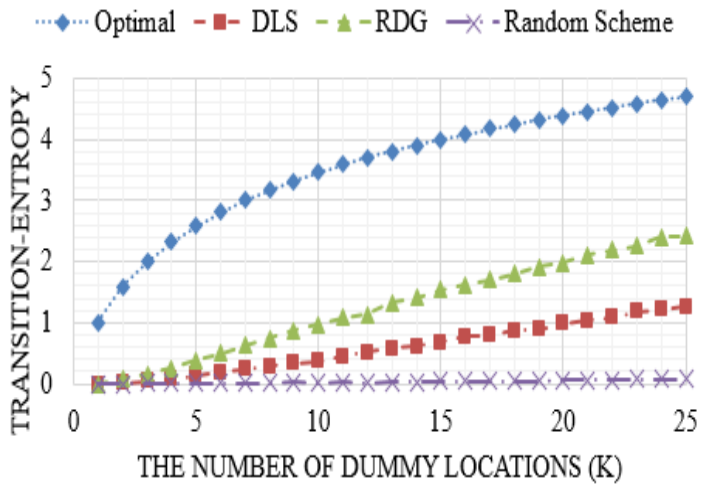
Cell entropy performance evaluation

Cell entropy indicates how different is the query probability of the actual user location from its associated dummies. A higher cell entropy is desirable, as it results in higher uncertainty of finding the real location. Fig. 3.5 presents the comparison among different algorithms in terms of cell entropy. The optimal value is achieved when all k locations queried from the LBS provider have the same probability of $1/k$, or equivalently, the location set has the cell entropy of $h = \log_2(k)$. The optimal value is the upper bound for all algorithms since it is the maximum entropy that a location set can achieve.

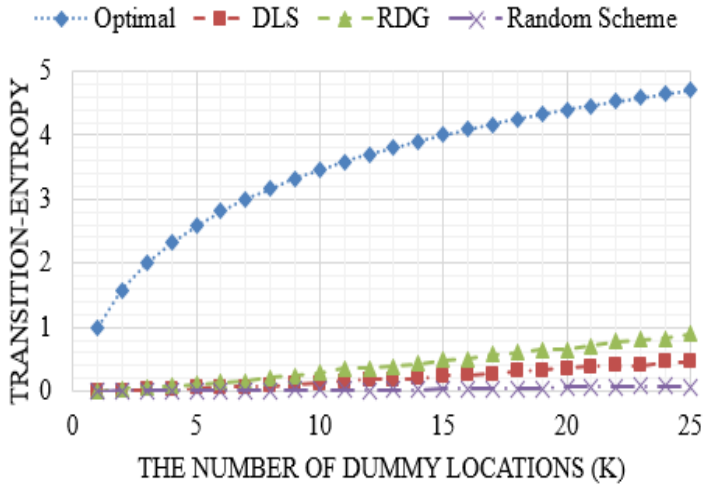
In Fig 3.5, three algorithms are compared, including the DLS algorithm which is the conventional method for generation of dummy locations, our proposed RDG algorithm, and the random scheme by which dummies are chosen randomly. Moreover, optimal cell entropy values are shown as a benchmark.

As expected, the random scheme proposed in [31] results in a lower cell entropy compared to the other algorithms due to the random generation of dummies. On the other hand, the RDG and DLS algorithms both consider query probability of cells in the generation of dummies. Therefore, the cell entropy of these two algorithms is higher than the random scheme and almost achieve near-optimal performance. Having such a high cell entropy ensures that the adversary is not able to compromise the location privacy of users from a stationery set of locations submitted to the server. Unfortunately, although the DLS algorithm has a robust

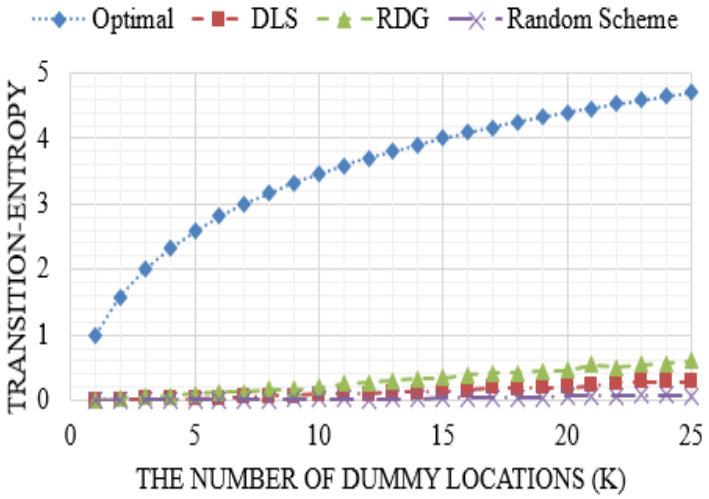
performance for a single collection of queried locations, no consideration has been given to locations queried as part of trajectories.



(a) Trajectories of length 2.



(b) Trajectories of length 3.



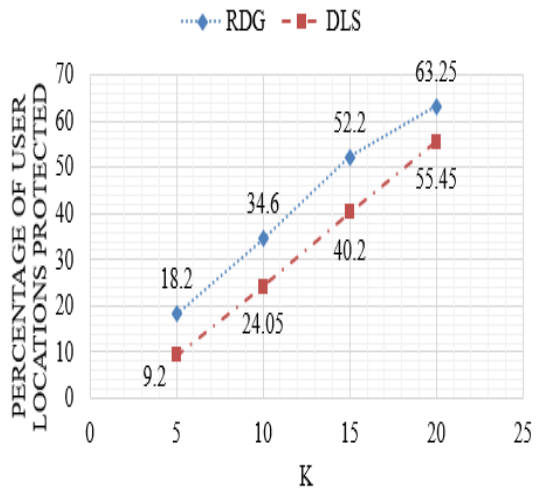
(c) Trajectories of length 4.

Fig. 3.6 Comparison of algorithms in terms of transition entropy for different k.

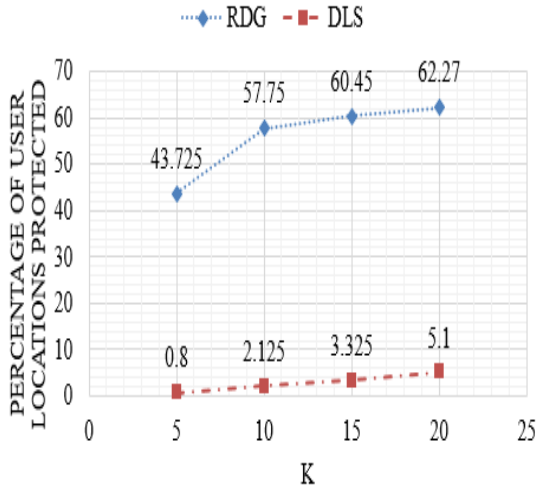
Transition entropy performance evaluation

The currently established cell entropy metric only considers the location privacy for a stationary set of queried locations submitted to the LBS server but overlooks the fact that users may ask for services successively. If users query location sets in consecutive attempts, they reveal the trajectory they are traveling. Therefore, adversaries can use the likelihood of traveling different paths between consecutive location sets to calculate the posterior probabilities and compromise the location privacy of users.

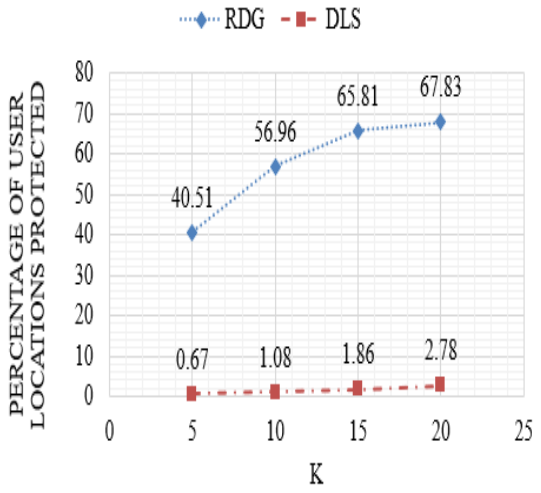
Fig. 3.6 compares the performance of different algorithms in terms of the transition entropy metric for various k. Having a lower transition entropy suggests a lower privacy level for users of the LBS applications and a higher likelihood for adversaries to find out the actual coordinates of the users. We start our evaluation by trajectories of length 2 in Fig. 3.6a, and then focus on the transition entropy for longer paths in Figs. 3.6b and 3.6c. In all the three graphs, the comparison is conducted among the optimal transition entropy values, the widely adopted algorithm DLS, the proposed RDG algorithm, and the random scheme.



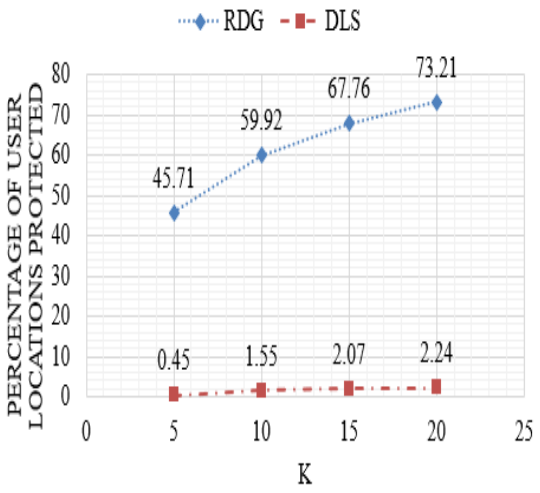
(a) Percentage of user locations protected for the trajectory length of 2.



(b) Percentage of user locations protected for the trajectory length of 4.



(c) Percentage of user locations protected for the trajectory length of 6.



(d) Percentage of user locations protected for the trajectory length of 8.

Fig. 3.7 The performance evaluation and comparison of algorithms against the Viterbi attack considering various path lengths and privacy requirement k.

In Fig. 3.6a, two consecutive location sets are generated based on the specified value of k. Each of the locations sets includes the real location of the user and its associated dummies. To make experiments as realistic as possible, the movement pattern is chosen randomly from the recorded trajectories in the dataset. The optimal value corresponds to a scenario, in which all members of the second location set are equally likely to be called consecutively after

the members of the first location set. This outcome is desirable, as it results in achieving k-anonymity for users and protecting their location privacy. The optimal values can be calculated in a similar way as the optimal number for the cell entropy. Considering Fig. 3.6a, the random scheme in which the dummy locations are chosen randomly achieves the lowest transition entropy, indicating that the adversary can easily recognize most of the dummies from the transition entropy even for the two consecutive location sets queried by the user.

Furthermore, it can be seen from the figure that although the DLS algorithm achieves near-optimal performance in terms of cell entropy, it results in significantly low privacy protection in trajectories. Even for two consecutive queries from the LBS provider, the DLS algorithm indicates a significantly low transition entropy. Such performance shows that adversaries can compromise the location privacy of users by calculating the posterior probabilities.

Fortunately, the proposed RDG algorithm can significantly improve the transition entropy, achieving almost twice as high transition entropy as the DLS algorithm. In other words, the likelihood of compromising the k-anonymity requirement is decreased by the proposed algorithm, which leads to a higher location privacy level for users of the LBSs.

Figs. 3.6b and 3.6c extend our analysis of transition entropy to trajectories with higher lengths. Both graphs indicate that as more locations are queried from the LBS provider, the transition entropy decreases. These experimental outcomes match well with the theory because having more information results in a more accurate calculation of posterior probabilities by adversaries; Hence, we expect to see less uncertainty and transition entropy.

Further investigating Figs. 3.6b and 3.6c, the DLS algorithm can be seen to have a very low transition entropy compared to the proposed RDG algorithm. Therefore, the RDG algorithm is viable in increasing the transition entropy of users while maintaining the cell entropy to a near-optimal level. However, as the adversary acquires more location points, the threat to location privacy of users gets more serious. This fact can be seen in Fig. 3.6. As the length of

trajectory increases, the transition entropy decreases, which refers to having a higher chance for adversaries to get access to the location data of users.

3.6.3 Performance of Algorithms against Viterbi Attack

In this section, we compare and evaluate the performance of our proposed RDG algorithm and the widely accepted DLS method against the proposed attack model. The Viterbi attack considers users in trajectories instead of just taking into account snapshots of the real locations and dummies. The Viterbi attack is based on the calculation of posterior probabilities of user locations revealed to the LBS provider. It will be shown in the following that applying such an attack can significantly compromise the location privacy of users. Therefore, having a robust algorithm such as RDG is crucial to protect the location privacy.

Fig. 3.7 illustrates the performance of the RDG and DLS algorithms once the Viterbi attack is applied to the dataset. The figure consists of four subfigures to show the performance with various length of trajectories. In each subfigure, the percentage of real locations of users which have been protected are exhibited for different privacy requirements k. For instance, in Fig. 3.7b, when $k = 5$, the graph indicates that the DLS algorithm can only protect 0.8 percent of the queried locations, and therefore, adversaries can almost distinguish all true locations of users from their associated dummies. This indicates how dangerous and powerful the Viterbi attack can be in compromising the privacy of users.

Considering the performance of the DLS algorithm, it can be seen in the figure that for path lengths greater than 2, the Viterbi attack can almost find out all real locations of the users despite the existence of dummy locations. Therefore, although in a single query user locations are protected using the existing dummy generation algorithms, when users are considered in trajectories, due to the extra side information that adversaries may hold, they are able to identify user locations.

Furthermore, another mainstream observation is that increasing the number of dummies can improve location privacy. Such an effect is expected as having a larger k indicates the generation of more dummies to protect user privacy. Unfortunately, the boost in privacy by increasing the value of k is not sufficient even when the trajectory length is two.

From Fig 3.7, our proposed RDG algorithm can help users to protect their privacy significantly better. The RDG algorithm takes into account the posterior probabilities that adversaries may hold and aims at making the likelihood of different paths equal. Doing so, the algorithm confuses adversaries in identifying exact locations of users. In contrast to the DLS algorithm, the performance of RDG algorithm improves as the path length increases. It means that for longer trajectories, the adversary has a less chance of compromising user privacy. Also, expectedly, increasing the value of k improves the privacy of users for the RDG algorithm as well.

3.7 Conclusion

In this study, we investigated the location privacy of users in trajectories and considered the threats that their previous queries could pose on their location privacy. We developed an attack model based on the Viterbi algorithm that demonstrates how susceptible the location privacy of users is. Therefore, we proposed a metric called transition entropy, which enables us to compare and assess the performance of different algorithms as the users move in trajectories.

Furthermore, to improve the transition entropy metric, an exhaustive search approach was proposed, which can increase the transition entropy for a given dummy generation algorithm. We also proposed an algorithm called RDG that results in a robust performance in terms of both transition entropy and cell entropy, while protecting users against the Viterbi attack.

Chapter 4

Location Privacy in Publication of Location Datasets

Publishing data by different organizations and institutes is crucial for open research and transparency of government agencies. In Australia, since 2013, over 7000 additional datasets have been published on 'data.gov.au', a dedicated website for publication of data by the Australian government. Moreover, the new Australian government data sharing and legislation encourages government agencies to publish their data, and as early as 2018 many of them will have to do so [86]. The process of data publication can be highly risky as it may disclose individuals' sensitive information. Therefore, an essential step before publishing data is to remove any uniquely identifiable information from the dataset. However, such an operation is not sufficient for privacy preservation. Adversaries can re-identify individuals in the datasets using common attributes called quasi-identifiers or may have prior knowledge about the trajectories traveled by the users, which enables them to reveal sensitive information that can cause physical, financial and reputational harms to people.

One of the most sensitive sources of data is location trajectories or spatio-temporal trajectories. Despite numerous use cases that the publication of spatio-temporal data can provide to users and researchers, it poses a significant threat to users' privacy. As an example, consider

a person who has been using GPS navigation to travel from home to work every morning on weekdays. If an adversary has some prior knowledge about a user, such as the home address, it is possible to identify the user. Such an inference attack can compromise user privacy, such as revealing the user's health condition and how often the user visit his/her medical specialist [87], [88], [89]. Therefore, it is crucial to anonymize spatio-temporal datasets before publishing them to the public. The privacy issue gets even more severe if the adversary links identified users to other databases, such as the database of medical records. That is the very reason why nowadays most companies are reluctant to publish any spatio-temporal trajectory datasets without applying effective privacy preserving techniques.

A widely accepted privacy metric for data publishing is k-anonymity. The metric can be summarized as ensuring that every trajectory in the published dataset is at least indistinguishable from $k-1$ other trajectories. For spatio-temporal trajectories, it is particularly challenging to achieve k-anonymity since data are dependent on each other. The authors in [90], adopted the notion of k-anonymity for trajectories and proposed an anonymization algorithm based on generalization. Xu et al. [91] investigated the factors such as spatio-temporal resolution and the number of users released on privacy preservation. The authors in [40] focused on improving the clustering approach in the anonymization process. The proposed anonymization scheme is based on achieving k-anonymity by grouping similar trajectories and removing the ones that are highly dissimilar. More recently, the authors in [1] developed an algorithm called k-merge to anonymize the trajectory datasets while preserving the privacy of users from probabilistic attacks. Local suppression and splitting techniques were also considered to preserve privacy in [56].

However, there are three major problems with the aforementioned approaches.

- Lack of a well-defined method to cluster trajectories as there is not an easy way to measure the cost of clustering when considering the distances among trajectories rather than simply the locations.

- The existing literature focuses on pairwise sequence alignment, which results in a high amount of information loss.
- There is no unified metric to evaluate and compare the existing anonymization methods.

In this work, we address the mentioned problems by proposing an enhanced framework termed the machine learning based anonymization (MLA) for anonymization of spatio-temporal trajectory datasets and a metric to compare the algorithms. MLA consists of two interworking algorithms: clustering and alignment. Our main contributions are summarized in the following bullet points.

- We propose to use k'-means algorithm for trajectory clustering and develop a technique to enable it. We also propose a variation of k'-means algorithm to preserve user privacy in overly sensitive spatio-temporal trajectory datasets.
- We propose to use a method termed the progressive sequence alignment for alignment of the trajectories in each cluster.
- We propose a privacy metric to evaluate and compare generalization algorithms based on the released area by data generalization.

MLA and all algorithms associated with it are applied on Geolife dataset that contains GPS logs of users in Beijing, China. The results are compared to one of the recent work presented in [1] and the state of art algorithm introduced in [90].

4.1 System Model

We assume that a map has been discretized into an $\varepsilon \times \varepsilon$ grid and the time is discretized into bins with length ε_t. Therefore, each point in the dataset represents a snapshot of a real-world location query including x-coordinate, y-coordinate, and time. The datasets with continuous time or

space data can fit into our model using interpolation. The level of spatial-temporal granularity in discretization does not affect the effectiveness of the proposed model. In our model, we consider a spatio-temporal trajectory datasets denoted by T. The dataset consists of trajectories $tr_1, ..., tr_n$ where n represents the number of trajectories in the dataset ($T = \{tr_1, ..., tr_n\}$, $|T| = n$). The i-th trajectory tr_i is an ordered set of l_i spatio-temporal 3D points (i.e., $tr_i = \{p_1, ..., p_{l_i}\}$, $|tr_i| = l_i$). Each point p_j is defined by a triplet $< x_j, y_j, t_j >$, where x_j, y_j, t_j indicate the x-coordinate, y-coordinate, and the time of query, respectively.

4.1.1 Privacy Model

As the adversary is considered to have information regarding the trajectories in the dataset, the coordinates of queries and their corresponding times are quasi-identifiers, which can endanger the privacy of users. In this work, we use a well-known metric called k-anonymity [92] to ensure the privacy of users. The k-anonymity in our dataset implies that a given trajectory in the original dataset can at best be linked to $k-1$ other trajectories in the anonymized dataset. Definition 1 formally defines the k-anonymity in the context of dataset.

Definition 1 *k-anonymous dataset: A trajectory dataset $\overline{T}$ is a k-anonymization of a trajectory dataset T if for every trajectory in the anonymized dataset $\overline{T}$, there are at least $k-1$ other trajectories with exactly the same set of points, and there is a one to one mapping relation between the trajectories in $\overline{T}$ and T.*

We assume that no uniquely identifiable information is released while publishing the dataset. However, the adversary may:

- already know about part of the released trajectory for an individual and attempt to identify the rest of the trajectory. For instance, the adversary is aware of the workplace of an individual and attempts to identify his or her home address.

- already know the whole trajectory that an individual has traveled, but try to access other information released while publishing the dataset by identifying the user in the dataset. For instance, the published dataset may also include the type of services provided to users and if the adversary can identify the user by its trajectory, it can also access the services provided to the user.

To this end, our aim is to protect users against the adversary's attempt to access sensitive information that may endanger the privacy of users.

4.1.2 Hierarchical Tree Transformation

In this work, generalization and suppression techniques are used to anonymize the dataset. These techniques are implemented using domain generalization hierarchy (DGH) defined in Definition 2. To clarify the construction of DGH, an example of DGH for x-coordinate is demonstrated in Example 1.

Definition 2 *A DGH for attribute, referred to as $H_{\mathscr{A}}$, is a partially ordered tree structure, which maps specific and generalized values of the attribute $\mathscr{A}$. The root of the tree is the most generalized value and is returned by function RT.*

Example 1 *Consider an 8×8 map. The x-coordinate attribute can have 8 possible values $(0,1,...,7)$. The DGH divides the largest possible interval for x-coordinate ($[0-7]$), which is the root of the tree, to two, four, and eight x-coordinate intervals as the DGH increases in depth. Fig. 4.1 shows the structure of the x-coordinate DGH. For the generation of the y-coordinate and time DGHs, a similar approach can be taken, which is not repeated here for succinctness.*

Each node on a DGH can be generalized by moving up one or multiple levels of the DGH. The process of generalizing node_i to one of its parent nodes node_j is denoted using $\text{node}_i \rightarrow \text{node}_j$. A special case of generalization, in which the node is generalized to the root of the DGH, is referred to as suppression. These two techniques are used as tools to anonymize the dataset

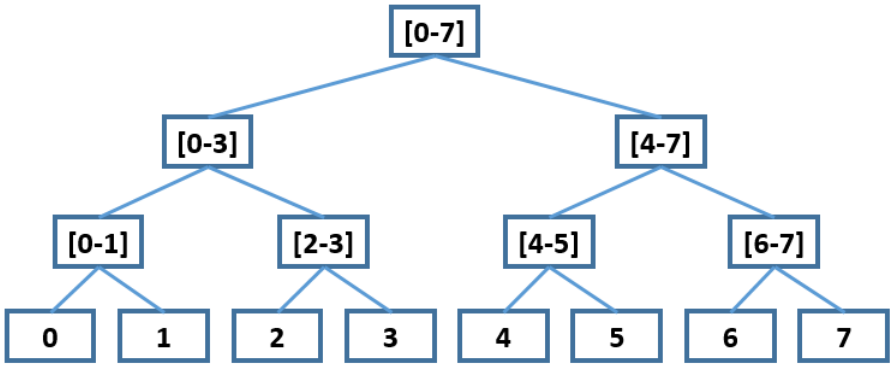


Fig. 4.1 An example of DGH for x-coordinate.

in the following sections. It must be noted that although quasi-identifiers in this work are x-coordinate, y-coordinate, and the time of query, the algorithms developed in our work can be extended to include other attributes as well.

4.1.3 Evaluation Metrics

Loss Metric

In order to quantify the loss incurred by the generalization and suppression, it is necessary to quantify the amount of loss happened while conducting the anonymization process. Here, we quantify the loss using the metric proposed in [93]. Definition 3 formally represents the metric and it is further elaborated in Example 2

Definition 3 *The information loss incurred by the generalization and suppression while replacing* $node_i$ *with* $node_j$ *in DGH* $H_{\mathscr{A}}$ *is defined in bits as*

$$LS(node_i, node_j) = \log_2 LF(node_j) - \log_2 LF(node_i) \; bits, \tag{4.1}$$

where $LF(.)$ *function returns the number of leaves in the subtree generated by a node and* $LS(.)$ *function returns the loss incurred by generalization of the nodes.*

Example 2 *Consider the DGH given in Fig. 4.1 the loss incurred for generalizing node* $[4-5]$ *to* $[4-7]$ *can be calculated as* $\log_2 4 - \log_2 2 = 2$ *bits.*

For generalizing two nodes, it is necessary to find the lowest common ancestor (LCA). The LCA is a critical point in the generalization process due to its corresponding subtree that entails both the nodes and achieves the lowest loss for any generalization. The definition of LCA is given in Definition 4. Moreover, Lemma 1 can be used to find the total loss incurred by the generalization of the two nodes to their LCA.

Definition 4 *The LCA of* $node_i$ *and* $node_j$ *in* $H_{\mathscr{A}}$ *is defined as the lowest common parent root of the two nodes. Function LCA returns the LCA.*

Lemma 1 *The total loss incurred by generalizing* $node_i$ *and* $node_j$ *in* $H_{\mathscr{A}}$ *with their LCA* $node_p$ *can be calculated as*

$$LS(node_i + node_j, node_p) = LS(node_i, node_p) + LS(node_j, node_p). \tag{4.2}$$

The total loss incurred during anonymization of a trajectory and a dataset are defined in Definitions 5 and 6, respectively.

Definition 5 *The total loss rendered by the generalization of trajectory* tr *to achieve the anonymized trajectory* $\overline{tr}$ *with respect to attribute* $\mathscr{A}$ *can be calculated as*

$$LS(\overline{tr}, \mathscr{A}) = \sum_{i=1}^{|\overline{tr}|} LS(tr_i.\mathscr{A}, \overline{tr}_i.\mathscr{A}). \tag{4.3}$$

where $tr_i.\mathscr{A}$ *indicates the i-th location of the trajectory* tr *with respect to the attribute* $\mathscr{A}$*. Here,* $\mathscr{A}$ *could denote x-coordinate, y-coordinate, or time.*

Definition 6 *The total loss with respect to an attribute $\mathscr{A}$ in an anonymized dataset $\overline{T}$ can be computed as*

$$LS(\overline{T},\mathscr{A}) = \sum_{\overline{tr}\in\overline{T}}^{|\overline{T}|} LS(\overline{tr},\mathscr{A}) \tag{4.4}$$

Privacy Metric

As different anonymization techniques utilize different generalization schemes in the existing works, it is not possible to apply one single and unified metric to compare these methods. Therefore, it is necessary to develop a metric to evaluate and compare the performance of anonymization schemes. We propose to use average released area per location as a new metric to evaluate and compare various schemes. In this subsection, the calculation of average released area per location is explained.

Any anonymization approach aims to maximize utility while preserving the privacy of users. Utility in generalization techniques refers to the area released for locations in the dataset. Consider a location in the dataset T with coordinates $< x_1, y_1, t_1 >$ and an arbitrary generalization function $\mathscr{F} : T \rightarrow \overline{T}$. After anonymization process, $< x_1, y_1, t_1 >$ is generalized with a number of other locations $< x_1, y_1, t_1 >$,..., $< x_a, y_a, t_a >$ in the dataset and an area S would be released representing these locations. For instance, if generalization returns the minimum rectangle surrounding the locations. The generalized area is given by:

$$S = (\max_i\{x_i\} - \min_i\{x_i\}) \times (\max_i\{y_i\} - \min_i\{y_i\}). \tag{4.5}$$

Once the anonymization is conducted, assume that n_1 locations are generalized to area S_1, n_2 locations are generalized to area S_2,..., n_b locations are generalized to area S_b. In this case,

the average released area per location can be calculated as

$$(\sum_{i=1}^{b} n_i \times S_i)/(\sum_{i=1}^{b} n_i), \tag{4.6}$$

in which no location belongs to more than one area. Average released area per location helps to understand how efficiently the data has been generalized and how much loss of utility has occurred by the generalization. Having the privacy requirement k-anonymity for all locations, a smaller released area per location indicates a higher utility of data while preserving the privacy of users.

4.1.4 Problem Formulation

The problem we seek to answer in this work is formally presented in Problem 1 as follows.

Problem 1 *Given a trajectory dataset T, a privacy requirement k, quasi-identifiers x-coordinate, y-coordinate, and time, how to generate an anonymized dataset $\overline{T}$ which achieves the k-anonymity privacy metric and minimizes the total loss with respect to all quasi-identifiers, which can be explicitly formulated as*

$$Minimize\{LS(\overline{T},x) + LS(\overline{T},y) + LS(\overline{T},t)\}. \tag{4.7}$$

4.2 MLA

Our proposed anonymization framework, MLA, consists of a robust alignment technique and a machine learning approach for clustering the trajectory datasets which are presented in this section.

4.2.1 Alignment

The process of alignment is defined as finding the best match between two trajectories in order to minimize the overall cost of generalization and suppression. The process of alignment between two trajectories has been studied in different domains mostly referred to as sequence alignment (SA). In this work, we adopt a multiple SA technique called progressive SA [94] for anonymization of spatio-temporal trajectories.

Progressive Sequence Alignment

The progressive SA is commonly used for SA of a set of protein sequences. Progressive SA is a heuristic approach for multiple SA. As a part of the algorithm, pairwise alignment of the trajectories is required. We use dynamic SA for this purpose. Dynamic SA is based on dynamic programming and commonly used in DNA SA [95, 96]. Fig. 4.2 illustrates an example of how the progressive SA works for four hypothetical sequences $tr_a = \{a_1, a_2, a_3, a_4\}$, $tr_b = \{b_1, b_2\}$, $tr_c = \{c_1, c_2, c_3\}$ and $tr_d = \{d_1, d_2\}$ to generate the resultant aligned trajectory $tr_r = \{r_1, r_2, r_3, r_4\}$. The longest path tr_a is chosen as the basis and it is aligned with a randomly chosen trajectory tr_b. The pairwise alignment process is implemented using dynamic SA. Then, the resultant trajectory is aligned with a third trajectory. The process continues until all trajectories are aligned. Instead of choosing the trajectories randomly during the progressive SA, the algorithm can choose the trajectory resulting in the lowest loss during the alignment. In Fig. 4.2, the way trajectory elements are located with respect to the longest path is referred to as the structure of the shorter path, and also, the spaces indicate the suppression operation during the alignment.

The dynamic SA algorithm is formally represented in Algorithm 6. Dynamic SA is based on dividing the problem of finding the best SA to subproblems and storing the solutions of subproblems in a table or matrix referred to as *SAmatrix* in the pseudocode. The objective is to achieve the minimal cost for SA. As before, the cost of alignment refers to the loss

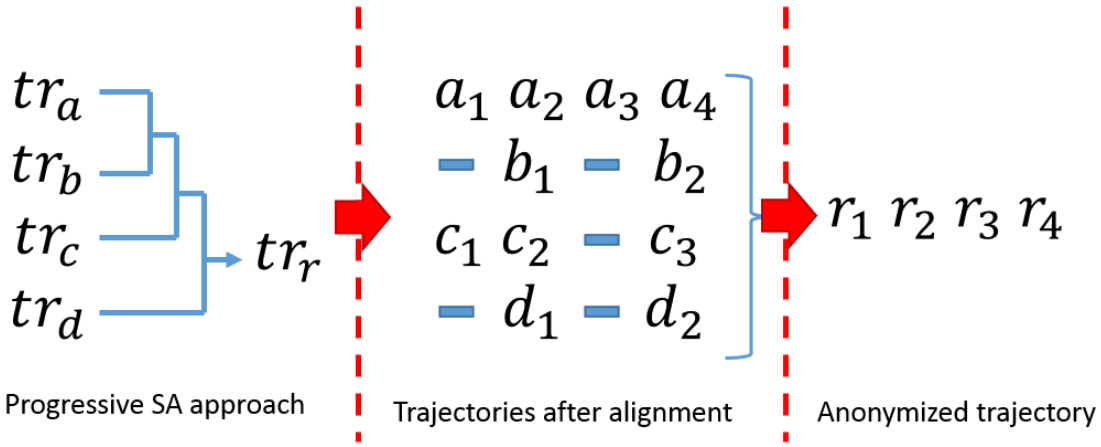


Fig. 4.2 An overview of progressive SA for alignment of four trajectories and generating the anonymized trajectory.

incurred during the alignment for different attributes of the sequence, which are x-coordinate, y-coordinate, and the time of the query.

A subproblem generation for matching the first to j-th element of tr_1 ($tr_1 = \{p_1, , p_2, ..., p_j\}$) with the first to i-th element of tr_2 ($tr_2 = \{q_1, , q_2, ..., q_i\}$) can be given as 1) match p_j and q_i; find the optimal alignment for $tr_1 = \{p_1, , p_2, ..., p_{j-1}\}$ and $tr_2 = \{q_1, , q_2, ..., q_{i-1}\}$ 2) suppress p_j; find the optimal alignment for $tr_1 = \{p_1, , p_2, ..., p_{j-1}\}$ and $tr_2 = \{q_1, , q_2, ..., q_i\}$ 3) suppress q_i; find the optimal alignment for $tr_1 = \{p_1, , p_2, ..., p_j\}$ and $tr_2 = \{q_1, , q_2, ..., q_{i-1}\}$.

The algorithm starts by creating a $(m+1) \times (n+1)$ matrix (*SAmatrix*), where m and n denote the length of the trajectories. The matrix will be used to store the minimum cost of each cell of the grid. Moreover, a list called *code* stores how the cells have been reached. Cell $[j+1, i+1]$ can be reached from three cells $[j, i+1]$, $[j+1, i]$, $[j, i]$. Each path corresponds to one of the subproblems explained. After finding all values of the matrix and tracing back the list *code*, the outputs of the algorithm are the value of cell $[m, n]$ indicating the minimum value of the total loss (*TotLoss*) required for the dynamic SA, the aligned trajectory (*GenTraj*), and the structure of the shorter path compared to the longer path as *ShoTrajStr*.

Algorithm 6: DynamicSA(tr_1, tr_2, H_x, H_y, H_t).

Required variables: $tr_1 = \{p_1, , p_2, ..., p_m\}$, $tr_2 = \{q_1, , q_2, ..., q_n\}$, H_x, H_y, H_t

$SAmatrix \leftarrow$ np.zeros($[m+1, n+1]$)
for i *in range*(m) **do**
    $Loss \leftarrow LS(p_i.x, rt(H_x)) + LS(p_i.y, rt(H_x))$
        $+LS(p_i.t, rt(H_t))$
    $SAmatrix[i+1, 0] \leftarrow SAmatrix[i, 0] + Loss$
end
for i *in range*(n) **do**
    $Loss \leftarrow LS(q_i.x, rt(H_x)) + LS(q_i.y, rt(H_x))$
        $+LS(q_i.t, rt(H_t))$
    $SAmatrix[0, i+1] \leftarrow SAmatrix[0, i] + Loss$
end
$options \leftarrow$ np.zeros(3)
$code \leftarrow$ list()
for i *in range*(m) **do**
    for j *in range*(n) **do**
        $Loss \leftarrow$ loss incurred by generalizing p_i and q_j
        $options[0] \leftarrow SAmatrix[i, j] + Loss$
        $Loss \leftarrow$ loss incurred by suppressing q_j
        $options[1] \leftarrow SAmatrix[i+1, j] + Loss$
        $Loss \leftarrow$ loss incurred by suppressing p_i
        $options[2] \leftarrow SAmatrix[i, j+1] + Loss$
        $BestOption \leftarrow$ np.argmin($options$)
        $code$.append(index of option with minimum value)
    end
end
$TotLoss \leftarrow SAmatrix[m, n]$
$GenTraj \leftarrow$ trace back the $code$ to generate the aligned trajectory
$ShoTrajStr \leftarrow$ trace back the $code$ to find out structure of shorter trajectory while alignment
Return $GenTraj, ShoTrajStr, TotLoss$

4.2.2 Clustering

Clustering can be seen as a search for hidden patterns that may exist in datasets. In simple words, it refers to grouping data entries in disjointed clusters so that the members of each cluster are very similar to each other. Clustering techniques are applied in many application areas such as data analysis and pattern recognition. In this subsection, first, we propose a heuristic approach for clustering spatio-temporal datasets. Due to high complexity of the algorithm, we come up with a technique to apply machine learning algorithms such as k'-means algorithm for clustering spatio-temporal trajectories. Then, we propose a variation of k'-means that can help to ensure k-anonymity for all the trajectories.

Heuristic Approach

Our proposed heuristic approach for clustering spatio-temporal trajectory datasets is detailed in Algorithm 7 and its helper function in Algorithm 8. The intuition behind the heuristic algorithm is to form the clusters by sequentially adding the most suitable trajectory that minimizes the total loss incurred by generalization and suppression for x-coordinate, y-coordinate, and the time of query, given their DGHs H_x, H_y, H_t.

The algorithm starts by calculating the number of clusters that needs to be generated and making a duplicate of the dataset called T. Moreover, a two-dimensional list is created, which will hold the trajectory IDs for each cluster. In lines [4-21], for each cluster (i.e., cluster c), the algorithm assigns the first trajectory in T to $Traj1$ as well as its duplicate $AlignedTraj$ and removes it from the database. This trajectory would be the first member of the cluster c. Then, given the privacy requirement k, $k-1$ other members of the cluster are chosen in lines [10-20]. Two memory lists $LossMemory$ and $TrajMemory$ are generated to hold the outcome of each execution of StaticSA function. In lines [12-15], for each remaining trajectory in the dataset, total loss and the aligned trajectory are calculated and assigned to the memory lists. Then, in lines [17-19], the trajectory ID which has resulted in the minimum total loss is attached to the

Algorithm 7: HeuristicClustering($OriginalDataset$, k, H_x, H_y, H_t).

$NumOfClus \leftarrow \lceil \frac{|T|}{k} \rceil$
$Clusters \leftarrow$ list()
$T \leftarrow OriginalDataset$
for c *in range*$(0, NumOfClus)$ **do**
    $Traj1 \leftarrow$ first trajectory in T
    $AlignedTraj \leftarrow Traj1$
    $cluster[c]$.append($NewMember$)
    T.remove($Traj1$)
    for i *in range*$(1, k)$ **do**
        $LossMemory =$ zeros$(|T|)$
        $TrajMemory =$ zeros$(|T|)$
        for j *in range*$(1, |T|)$ **do**
            $Traj2 \leftarrow j$-th trajectory in T
            $(NewTraj[j], LossMemory[j]) \leftarrow$ StaticSA($AlignedTraj$, $Traj2$, H_x, H_y, H_t)
        end
        $NewMember \leftarrow$ The trajectory ID with minimum loss memory
        $cluster[c]$.append($NewMember$)
        T.remove($NewMember$)
        $AlignedTraj \leftarrow$ update based on $NewMember$
    end
end
$(\overline{T}, Loss) \leftarrow$GenerateAnonymizedDataset($cluster$, $OriginalDataset$, H_x, H_y, H_t)
Return $(\overline{T}, Loss)$

cluster and removed from the database. Having calculated the trajectory IDs for each cluster, the helper function GenerateAnonymizedDataset is called in order to return the anonymized dataset ($\overline{T}$) and its loss.

The helper function (GenerateAnonymizedDataset) takes the original dataset and the two-dimensional list of IDs (*cluster*) which indicates the trajectory IDs that need to be in each cluster as inputs. The target of the algorithm is to return the total loss and an anonymized trajectory. The algorithm starts by initializing the total loss to zero in Line 1 and creating an empty list ($\overline{T}$) to hold the new anonymized dataset. In lines [3-16], for each cluster, the IDs are fetched into *CluTraIDs*, and then, the total generalized trajectory is calculated in lines [6-10]. Finally, in lines [11-16], the total loss for each cluster is calculated and the cluster head is attached to the anonymized dataset.

Algorithm 8: GenerateAnonymizedDataset($cluster$, $OriginalDataset$, H_x, H_y, H_t).

```
1  TotalLoss ← ∅
2  T̄ ← list()
3  for i in range(0, len(cluster)) do
4      CluTraIDs ← cluster[i]
5      Traj1 ← trajectory corresponding to CluTraIDs[0]
6      for j in range(0, len(CluTraIDs)) do
7          Traj2 ← trajectory corresponding to CluTraIDs[j]
8          (NewTraj, Loss) ← StaticSA(traj1, traj2, H_x, H_y, H_t)
9          Traj1 ← NewTraj
10     end
11     for j in range(0, len(CluTraIDs)) do
12         TempTraj ←trajectory corresponding to CluTraIDs[j]
13         (CacheTraj, Loss) ← StaticSA(NewTraj, TempTraj, H_x, H_y, H_t)
14         T̄.append(NewTraj) TotalLoss ← TotalLoss + Loss
15     end
16 end
17 Return (T̄, TotalLoss)
```

$$\text{Total loss} = \underbrace{\sum_{i=1}^{|T|}(LS(tr_i.x, RT(H_x)) + LS(tr_i.y, RT(H_y)) + LS(tr_i.t, RT(H_t)))}_{\text{A}} - \underbrace{(\sum_{i=1}^{|cluster|}\sum_{j=1}^{|cluster[i]|}(LS(h_j.x, RT(H_x)) + LS(h_j.y, RT(H_y)) + LS(h_j.t, RT(H_t))))}_{\text{B}}. \tag{4.8}$$

k'-means Clustering Approach

k'-means algorithm [97] is an attractive clustering algorithm currently used in many applications, especially in data analysis and pattern recognition [98]. The main advantage of k'-means algorithm is simplicity and fast execution. The reason behind using a prime notation on top of the variable k is to avoid confusion between the "k" in the clustering algorithm and the k used in the definition of k-anonymity addressed before.

The algorithm aims to partition the input dataset into k' clusters. The only inputs to the algorithm are the number of clusters k' and the dataset. Clusters are represented by adaptively-changing cluster centres. The initial values of the cluster centres are chosen randomly. In each stage, the algorithm computes the Euclidean distance of data from the centroids and partition them based on the nearest centroid to each data. More formally, representing the set of all centroids by $C = \{c_1, c_,..., c_{k'}\}$, each point in the dataset, denoted by x, is assigned to a centroid that has the shortest Euclidean distance to the point. This can be written as

$$\underset{c_i \in C}{\operatorname{argmin}}\, dist(x, c_i)^2, \tag{4.9}$$

where the function $dist(.)$ returns the Euclidean distance between two points. Denoting the set of assigned data to the i-th cluster by S_i, new centroids are calculated in the second stage via

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i. \tag{4.10}$$

The algorithm continues the same process until the values of centroids no longer change. The k'-means algorithm is guaranteed to converge [99].

In the rest of this section, we first present a Lemma followed by explaining how the k'-means algorithm can be applied to trajectory datasets to reinforce the privacy preservation of users.

Lemma 2 *The total loss incurred by generalizing $node_i$ and $node_j$ with respect to $H_{\mathscr{A}}$ can be calculated as*

$$\begin{aligned} &LS(node_i, node_j) = \\ &\quad |LS(node_i, RT(H_{\mathscr{A}})) - LS(node_j, RT(H_{\mathscr{A}}))|. \end{aligned} \tag{4.11}$$

Lemma 2 indicates that the loss incurred by generalizing two nodes is equal to the difference between losses incurred by their suppression. As before, for any clustering outcome of data, assume that *cluster* is a two-dimensional list, in which the j-th element of the list returns the IDs of the trajectories in the j-th cluster. Moreover, we denote the j-th cluster head after generalization and suppression for all trajectories as h_j. Therefore, the total loss can be written as

$$\begin{aligned}\text{Total loss} &= LS(\overline{T},x)+LS(\overline{T},y)+LS(\overline{T},t)\\ &=\sum_{j=0}^{k-1}\sum_{tr\in cluster[j]}(LS(h_j.x,tr.x)\\ &\quad +LS(h_j.y,tr.y)+LS(h_j.t,tr.t)).\end{aligned} \tag{4.12}$$

As explained in (4.7), the objective of clustering algorithms is to minimize this equation. Therefore, using Lemma 2 the equation (4.12) can be written as

$$\text{Total loss} = \tag{4.13}$$

$$\begin{aligned}&\sum_{j=0}^{k-1}\sum_{tr\in cluster[j]}(|LS(h_j.x,RT(H_x))-LS(tr.x,RT(H_x)|\\ &\quad +|LS(h_j.y,RT(H_y))-LS(tr.y,RT(H_y)|\\ &\quad +|LS(h_j.t,RT(H_t))-LS(tr.t,RT(H_t))|.\end{aligned} \tag{4.14}$$

Rearranging (4.13), the objective equation can be found by minimizing total loss formulated in (4.8). This can be done by maximizing part B and minimizing part A. Since the cluster heads are generated based on the clustering algorithm, they cannot be used as part of the optimization process. Therefore, we aim at minimizing part A in (4.8).

Part A in the equation (4.8) refers to finding the total distance of each trajectory from DGH root of the attributes. Therefore, for each trajectory, a three-dimensional vector $< d_x, d_y, d_t >$ is constructed, where d_x, d_y, d_t store the loss incurred by generalizing the x-coordinate, y-coordinate, and time, respectively. Having distances of all points from the roots, we cluster the trajectories using the k'-means algorithm. The algorithm clusters trajectories with a similar loss

Algorithm 9: Pseudocode of iterative k'-means algorithm.

```
while true do
    run k'-means algorithm on dataset (#clusters = ⌊#data trajectories / k⌋)
    remove trajectories that belong to clusters with at least k members from the dataset
    if #len(dataset) < 2 * k then
        cluster the remaining trajectories together
        break;
    end
end
```

from the root in the same group. This process is particularly important as trajectory datasets usually include trajectories as short as one query to trajectories with hundreds of queries.

A major drawback of the k'-means algorithm is clustering the trajectories without any constraint on the minimum number of trajectories that needs to be in each cluster. Therefore, the algorithm might result in some of the clusters including less than k trajectories that violate the k-anonymity of the trajectories. If the data is not extremely sensitive such as the data used in military, it is usually acceptable to have a few trajectories below the k-anonymity criterion. As it will be demonstrated in Section 4.3 experiments, as a general rule the number of trajectories not achieving k-anonymity is close to or below 20% of the trajectories based on the value of k chosen for the privacy. To amend the naive k'-means algorithm for sensitive applications, we propose to use a variation of k'-means algorithm, which we call it iterative k'-means. The idea relies on running the k'-means algorithm iteratively to ensure that all clusters will achieve k-anonymity. Therefore, after each iteration of the k'-means algorithm, the clusters including at least k trajectories are disbanded and the trajectories are put back into the pool for the next iteration of the k'-means algorithm. This process continues until all clusters have at least k members. Algorithm 9 represents the pseudocode of the iterative k'-means.

4.3 Experiments

In our experiment, we use the data collected by Geolife project [83–85]. We have conducted our experiments on a $1\,km \times 1\,km$ central part of the Beijing map with the resolution of $0.01km \times 0.01km$ for each grid cell. The various location privacy requirements (k) of the users are investigated for the values 2, 5, 10, and 15. The experiments were performed on a PC with a 3.40 GHz Core-i7 Intel processor, 64-bit Windows 7 operating system, and an 8.00 GB of RAM. Python programming is used to implement the algorithms.

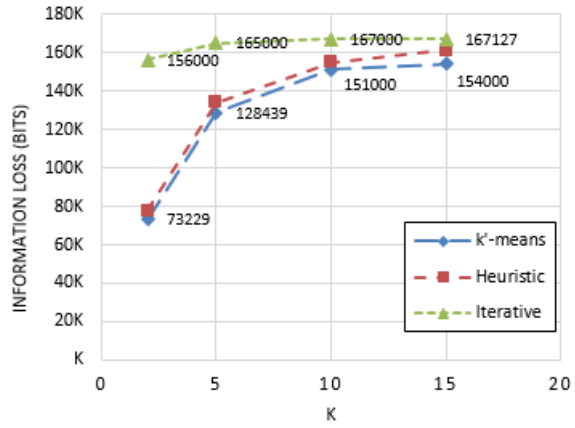
4.3.1 Performance Evaluation and Comparison

Fig. 4.3 presents the performance evaluation of MLA predicated on three clustering approaches developed in this work. The algorithms have been investigated from three aspects: information loss, increase in trajectory length, and execution time. In all graphs, x-axis indicates k-anonymity requirement for the dataset. The total information loss and average information loss per cluster of algorithms are considered in Figs. 4.3a and 4.3b, respectively. Information loss, shown in the y-axis, indicates the total loss incurred while applying generalization and suppression on x-coordinate, y-coordinate, and the time of the query. The maximum possible incurred information loss for the whole dataset by suppressing all trajectories is 474572 bits. This value is the upper bound on all anonymization algorithms. Note that this constant changes for different datasets. The main existing trend in Figs. 4.3a and 4.3b is that by increasing the value of k, the total incurred loss increases. This outcome meets our expectation as increasing the value of k indicates having larger cluster sets, which results in the alignment of a higher number of trajectories in each cluster, and thereby, a higher total loss by the alignment. Among our proposed algorithms, k'-means algorithm provides the best performance as it corresponds to minimum lost bits incurred by the generalization and suppression.

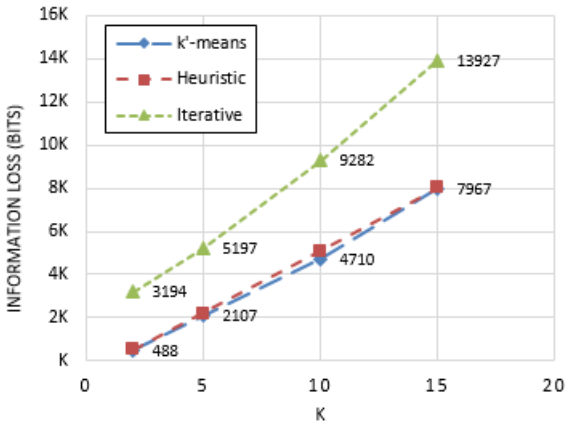
The amount of information that k'-means algorithm preserves is even higher than that of the heuristic approach, in which the most suitable trajectories are chosen to minimize the

information. This trend can be seen for both of the total information loss of the dataset and the average information loss of dataset per cluster for different k values. Such a trade-off exists, because some clusters contain a small number of trajectories not satisfying the k-anonymity requirement. The loss of privacy by k'-means algorithm is further analyzed in Fig. 4.4 which will be explained later in this section. The iterative k'-means algorithm is constructed on top of the k'-means algorithm to ensure that all the trajectories satisfy the required privacy requirement. This is particularly important for sensitive applications, in which there are strict requirements for privacy preservation. The cost of having higher privacy for the iterative k'-means algorithm is a larger loss of information.

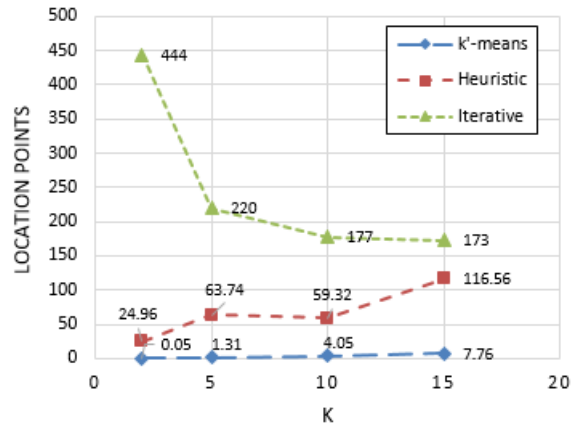
Figs. 4.3c and 4.3d present the average increase in the length of trajectories for the whole dataset and per cluster. Due to the alignment process, shorter trajectories may need to be aligned with longer trajectories, which result in an increase in the length of trajectories in the anonymized released dataset. The best performance among the algorithms is yielded by the k'-means algorithm with the lowest increase in the lengths of trajectories. Compared to other two approaches, the heuristic strategy performs better than the iterative k'-means with a smaller k, but as the k value increases, the average increase in trajectory length converges due to a large cluster size. Figs. 4.3e and 4.3f compare the total and average per cluster execution time of the different algorithms. Note that since the heuristic algorithm requires a significantly higher amount of time to run, it is shown on top of the graphs as a flat line with the corresponding values shown below it. The execution time of the k'-means and iterative k'-means algorithms are significantly lower than that of the heuristic algorithm and as expected the iterative k'-means consumes slightly more execution time as it has additional steps to ensure the k-anonymity of all trajectories.



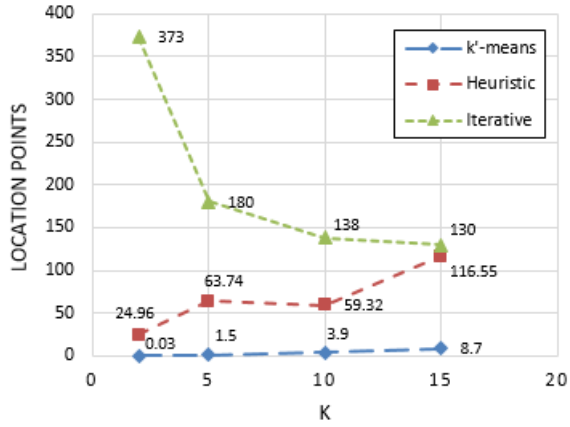
(a) Total information loss



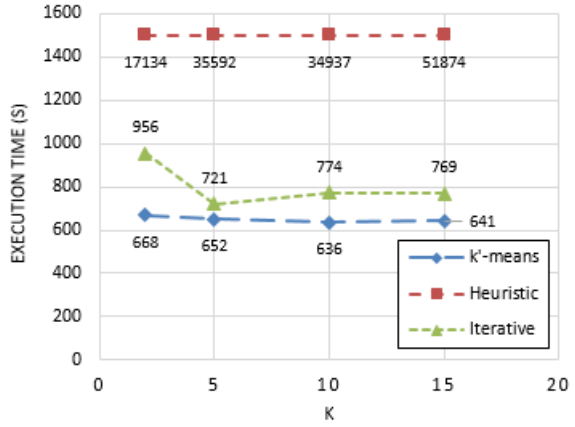
(b) Average information loss per cluster



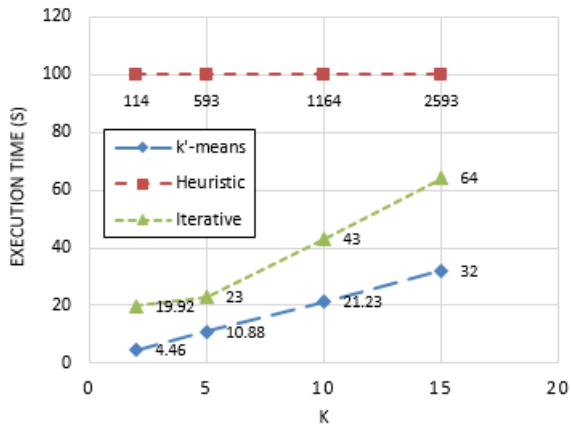
(c) Average increase in length of trajectories



(d) Average increase in length of trajectories per cluster



(e) Total execution time (the heuristic algorithm's results are shown as a flat line with the values written below the line)



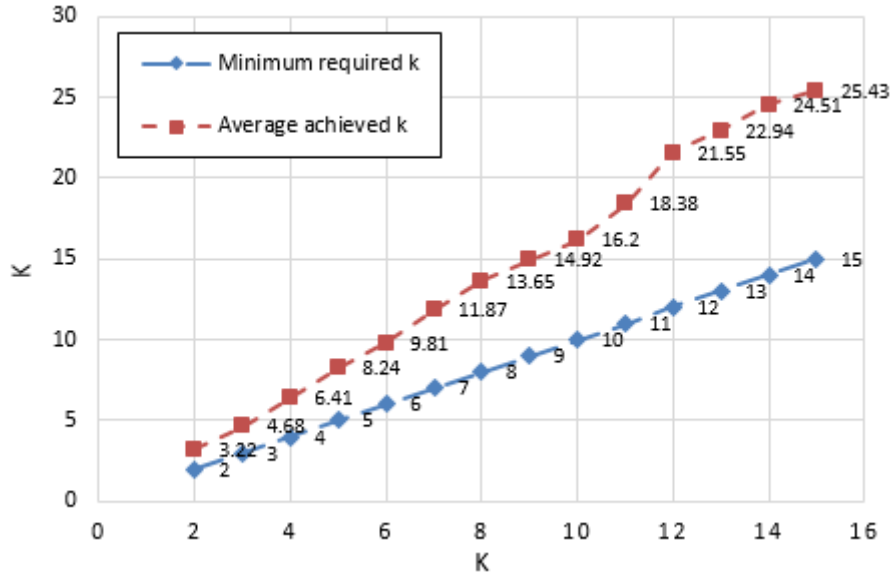
(f) Average execution time per cluster (the heuristic algorithm's results are shown as a flat line with the values written below the line)

Fig. 4.3 Performance evaluation of MLA with different values of k.

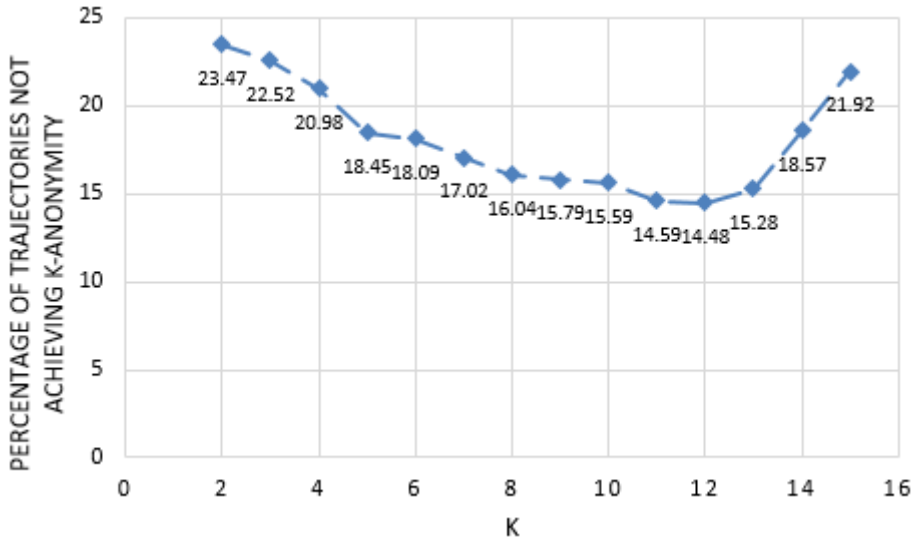
4.3.2 Detailed Analysis of k'-means Algorithm

Overall, the detailed k'-means algorithm's results in satisfactory performance in terms of information loss, execution time, and the average increase in the length of trajectories. Moreover, the complexity of the k'-means algorithm is of an order of the number of data entries for large datasets, whereas the order of the heuristic algorithm is proportional to the square of this number. Therefore, the k'-means algorithm has several significant advantages compared to the heuristic approach. Hence, if it is acceptable for the datasets to have a few trajectories below the k-anonymity requirement, then, it is more beneficial to use the k'-means algorithm instead of the heuristic or the iterative k'-means algorithm. This is usually true for datasets not entailing classified information. Therefore, we further analyze the performance of this algorithm in the remaining of this section and compare it to the state-of-art algorithms recently proposed. Also, note that in the rest of this paper when MLA is mentioned, the k'-means algorithms is adopted for clustering by default.

Fig. 4.4 provides two graphs showing the details of the performance yielded by the k'-means algorithm. The first graph indicates the average value of k achieved while applying the k'-means algorithm, and the second graph shows the percentage of trajectories that did not achieve the k-anonymity in the anonymization process with different values of k. In Fig. 4.4(a), it is evident that despite some of the trajectories losing their k-anonymity during the anonymization, the average value of anonymity achieved is above the minimum requirement. The value of the average gets even better as the value of k increases. Fig. 4.4(b) shows the percentage of the trajectories not achieving the minimum required k-anonymity. This value is below 20% on average, which means that over 80% of the trajectories are guaranteed to at least have k-anonymity. The reason causing the uneven curves in the figure is because the number of clusters is divisible by k, which results in an additional cluster distorting the curves.



(a) Average value of k achieved by applying the k'-means algorithm



(b) Percentage of users not satisfying k-anonymity requirement by applying the k'-means algorithm

Fig. 4.4 Detailed performance evaluation of the k'-means algorithm.

4.3.3 Comparison

We compare MLA with the static algorithm proposed in [90], and recently published anonymization approach in [1]. The idea behind the static alignment algorithm in [90] is that two tra-

jectories are matched element by element without any shifts or spaces. In more details, the static algorithm attempts to match two sequences based on the same index. Therefore, each element of the first sequence tr_1 is aligned with an element having the same index in the other input trajectory tr_2. Based on our evaluation, the total incurred information loss is reduced by 7.2% by using the proposed progressive SA algorithm. It must be noted that the dataset includes trajectories as large as hundreds of queries and as small as a single query from the location-based service provider. Therefore, matching these length-variant trajectories would impose a substantial information loss even for the best possible match of the sequences.

Fig. 4.5 indicates the comparison result between our proposed anonymization technique and the recent generalization method proposed in [1]. The authors in [1] attempted to minimize the incurred loss of the anonymization by sorting out the spatio-temporal locations in the time domain and applying a heuristic approach for generalization. They also used a heuristic approach for clustering trajectories. Note that any anonymization approach aims to maximize utility while preserving the privacy of users. Utility in generalization techniques refers to the area released for locations in the dataset. Therefore, to have a fair comparison, we compare our work with the approach proposed in [1] based on the average released area for locations. The metric is thoroughly explained in Section 4.1. It can be seen from the figure that our proposed algorithm can significantly increase the utility of the generalization approach. In other words, the anonymized dataset has on average smaller released area per location while preserving the privacy of users. To further compare alignment approaches, in Fig. 4.5, we applied random clustering to group the trajectories, and then, used the alignment approach in our proposed work and the previous work to generate anonymized trajectories. As can be seen in the figure, our alignment approach outperforms the previous work by a higher utility of anonymized dataset.

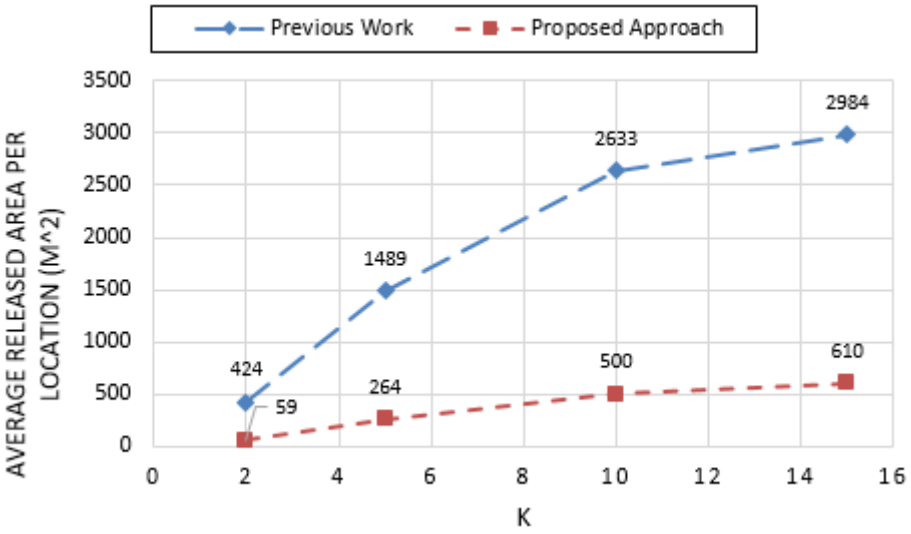


Fig. 4.5 Comparison of MLA with the previous work proposed in [1].

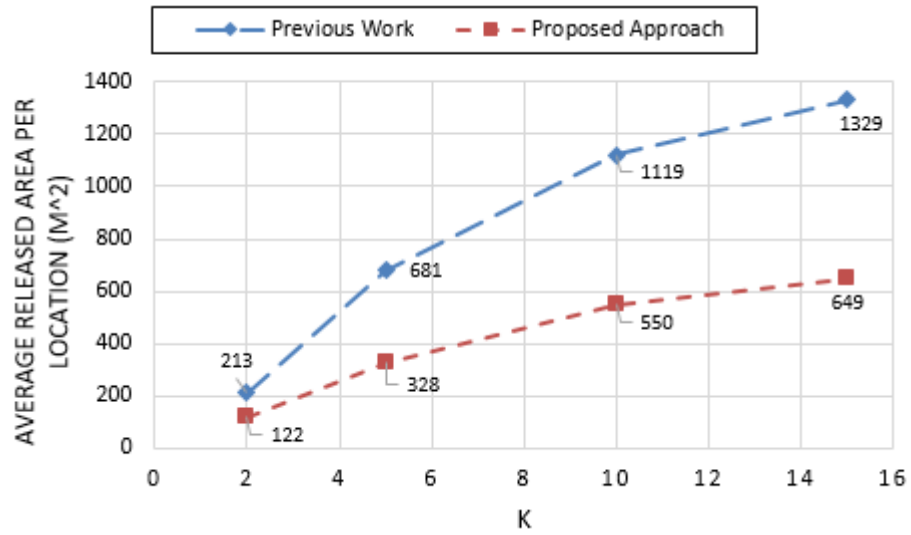


Fig. 4.6 Comparison of MLA with the previous work proposed in [1] when applying random clustering to both.

4.4 Applications

In this section, we introduce several applications that we believe our work has the most impact on.

4.4.1 Location-Based Data

As the framework for anonymization presented in this work considers location trajectories, one of the main applications of the framework is the privacy of location-based data. The use of location-based applications is more prevalent than any time before. Governments attempt to analyze the infrastructure using the location data and researchers use these data to investigate human behavior. Research has verified that even simple analytics on these published trajectory data would yield serious risk of users' privacy and even be capable of identifying users of location-based applications. [100]. Therefore, applying anonymization techniques such as the one we have developed in this work is necessary to preserve the privacy of the users.

4.4.2 Medical Records

The recent advances in medical information technology have enabled the collection of a detailed description of patients and their medical status [101]. Such data is usually stored in electronic medical record systems [102–104]. Similar to spatio-temporal trajectories, many of the medical records need to be published by agencies and organizations. Unfortunately, research has shown that solely relying on de-identification is insufficient to protect users' privacy, as the medical records from multiple databases can be linked together to identify individual patients [90]. Therefore, there is an urgent need for viable algorithms to anonymize the medical data. The problem of anonymization in spatio-temporal trajectories is very similar to anonymization in longitudinal electronic medical records. This can be easily justified by the similar way, in which these data are stored. Assume a patient who has referred to medics several times in his or her lifetime. Each time the records of the patient are stored in a longitudinal dataset, in which the age and the diagnosed disease record are registered. These longitudinal records can be seen as a trajectory for the patient, and our proposed algorithms in this work can be applied to anonymize a dataset of such longitudinal electronic medical records.

4.4.3 Web Analytics

Another important application of the framework developed in this work is web analytics. Web analytics refers to analyzing online traces of users. Web analytics has become a competitive advantage for many companies due to the amount of detailed information that can be extracted from the data. Therefore, protecting the trajectories that the users explored on the Internet has become a major challenge for researchers. The similarity between spatio-temporal trajectories and web analytics can be well explained by the following example. For instance, Geoscience Australia is constantly recording and publishing the site logs users make on their website. The site log filename is composed of a four-digit station identifier, followed by a two-digit month and a two-digit year, e.g., ALIC0414 is the site log for the Alice Springs GNSS site that was updated in April 2014 [86]. Such a trajectory of logins to the website is analogous to a spatio-temporal trajectory with three attributes. Therefore, the framework developed in this work can be used to anonymize the online traces of users before publishing web browsing data.

4.5 Conclusion

In this work, we have proposed a framework to preserve the privacy of users while publishing the spatio-temporal trajectories. The proposed approach is based on a efficient alignment technique termed progressive sequence alignment in addition to a machine learning clustering approach that aims to minimize the incurred loss by the anonymization process. For clustering, several techniques have been proposed: a heuristic approach to minimize the incurred loss, an approach based on the k'-means algorithm, and finally a variation of the k'-means algorithm for guaranteeing the k-anonymity in sensitive datasets. Our results indicate the superior performance of our proposed framework compared with the previous works.

Chapter 5

Conclusion

In the broader view of the preserving location privacy of the users, we have considered two commons scenarios in this thesis: (I) privacy preservation of users in telecommunications networks against untrusted LBS servers, and (II) publication of spatio-temporal trajectories by trusted service providers while preserving the privacy of users.

For the first category, we have proposed a metric called transition entropy. This metric enables us to evaluate and compare the performance of existing algorithms. We have also developed an attack model based on the Viterbi algorithm to identify the susceptibility of the user location privacy against malicious attacks. Moreover, to improve the location privacy of users, we have developed a robust algorithm called RDG, which, according to our simulations, has resulted in significant advancements in terms of transition entropy. The RDG algorithm also preserves the performance in terms of the traditional accepted metric cell entropy. There are several potential future directions associated with our work:

- Extend our approach to 'implicit' datasets, in which the time intervals between queries are not equal. Our work has been focused on 'explicit' datasets with an equal time interval between queries. However, it is significantly important to extend the approach for 'implicit' datasets as well.

- Improve the comprehensiveness of posterior probabilities in the calculation of transition entropy to consider the temporal information of users. There could other factors not considered in the calculation of our proposed posterior probabilities. This can also depend on the considered system model. Therefore, it could be advantageous to experiment with other significant factors in the calculation of posterior probabilities.

- Improve the RDG algorithm to achieve higher transition entropy levels. Although our algorithms can improve the transition entropy performance, it is still far away from the optimal value. This particularly becomes evident for large trajectories.

To improve privacy in the second scenario, we proposed a robust anonymization framework termed as MLA, preserving the k-anonymity of the users. The MLA framework also significantly reduces the information loss compared with the previous approaches. The framework is based on the k'-means algorithm and develops a cost-effective methodology for the anonymization of location datasets. Some future directions worth consideration would be:

- Developing methods to reduce information loss further while achieving k-anonymity. Although our proposed framework can significantly improve information loss performance, future work can focus on further minimizing the information loss to achieve higher levels of privacy.

- Considering other domain generalization hierarchy trees, which can help to reduce the complexity while decreasing information loss. In this work, we only investigated binary search trees as the coding scheme; however, other approaches could be replaced in the MLA framework to result in higher performance levels.

- Investigating theoretical limits of the MLA algorithm.

References

[1] Marco Gramaglia, Marco Fiore, Alberto Tarable, and Albert Banchs. Towards privacy-preserving publishing of spatiotemporal trajectory data. *arXiv preprint arXiv:1701.02243*, 2017.

[2] Location-based services (lbs) and real time location systems (rtls) market by location (indoor and outdoor), technology (context aware, uwb, bt/ble, beacons, a-gps), software, hardware, service and application area - global forecast to 2021. https://www.marketsandmarkets.com/Market-Reports/location-based-service-market-96994431.html.

[3] Jonathan Malkin and Csaba Kecskemeti. Home location identification using grouped location data, 2018. US Patent App. 15/331,083.

[4] Filipa Pajevi and Richard G Shearmur. Catch me if you can: Workplace mobility and big data. *Journal of Urban Technology*, 24(3):99–115, 2017.

[5] Chen-Yi Lin, Yuan-Chen Wang, Wan-Tian Fu, Yun-Sheng Chen, Kuan-Chen Chien, and Bing-Yi Lin. Efficiently preserving privacy on large trajectory datasets. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pages 358–364. IEEE, 2018.

[6] Shan Chang, Chao Li, Hongzi Zhu, Ting Lu, and Qiang Li. Revealing privacy vulnerabilities of anonymous trajectories. *IEEE Transactions on Vehicular Technology*, 67(12):12061–12071, 2018.

[7] Girish M Babu and Valasala NVA Sai Sankar. Case study: Mobile apps usage vis-à-vis shopping habits and preferences. *Advances in Management*, 11(2):19–25, 2018.

[8] Jean Hardy, Tiffany C Veinot, Xiang Yan, Veronica J Berrocal, Philippa Clarke, Robert Goodspeed, Iris N Gomez-Lopez, Daniel Romero, and VG Vinod Vydiswaran. User acceptance of locationtracking technologies in health research: implications for study design and data quality. *Journal of biomedical informatics*, 79:7–19, 2018.

[9] Dimiter V Dimitrov. Medical internet of things and big data in healthcare. *Healthcare informatics research*, 22(3):156–163, 2016.

[10] Kit Huckvale, José Tomás Prieto, Myra Tilney, Pierre-Jean Benghozi, and Josip Car. Unaddressed privacy risks in accredited health and wellness apps: a cross-sectional systematic assessment. *BMC medicine*, 13(1):214, 2015.

[11] Alastair R Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, (1):46–55, 2003.

[12] Michael Fuller. Big data and the facebook scandal: Issues and responses. *Theology*, 122(1):14–21, 2019.

[13] Andreas Pfitzmann and Marit Kohntopp. Anonymity unobservability and pseudonymity a proposal for terminology. In *Designing privacy enhancing technologies*, pages 1–9. Springer, 2001.

[14] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, Technical report, SRI International, 1998.

[15] Marco Gruteser and Dirk 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.

[16] Chi-Yin Chow and Mohamed F Mokbel. Enabling private continuous queries for revealed user locations. In *International Symposium on Spatial and Temporal Databases*, pages 258–275. Springer, 2007.

[17] Toby Xu and Ying Cai. Exploring historical location data for anonymity preservation in location-based services. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 547–555. IEEE, 2008.

[18] Xiao Pan, Jianliang Xu, and Xiaofeng Meng. Protecting location privacy against location-dependent attacks in mobile services. *IEEE Transactions on Knowledge and Data Engineering*, 24(8):1506–1519, 2012.

[19] Jianliang Xu, Xueyan Tang, Haibo Hu, and Jing Du. Privacy-conscious location-based queries in mobile environments. *IEEE Transactions on Parallel and Distributed Systems*, 21(3):313–326, 2010.

[20] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, Jan 2003.

[21] Alastair R Beresford and Frank Stajano. Mix zones: User privacy in location-aware services. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 127–131. IEEE, 2004.

[22] Balaji Palanisamy and Ling Liu. Mobimix: Protecting location privacy with mix-zones over road networks. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 494–505. IEEE, 2011.

[23] Balaji Palanisamy and Ling Liu. Attack-resilient mix-zones over road networks: architecture and algorithms. *IEEE Transactions on Mobile Computing*, 14(3):495–508, 2015.

[24] Rongxing Lu, Xiaodong Lin, Tom H Luan, Xiaohui Liang, and Xuemin Shen. Pseudonym changing at social spots: An effective strategy for location privacy in vanets. *IEEE Transactions on Vehicular Technology*, 61(1):86–96, 2012.

[25] Sheng Gao, Jianfeng Ma, Weisong Shi, Guoxing Zhan, and Cong Sun. Trpf: A trajectory privacy-preserving framework for participatory sensing. *IEEE Transactions on Information Forensics and Security*, 8(6):874–887, 2013.

[26] Julien Freudiger, Maxim Raya, Márk Félegyházi, Panos Papadimitratos, and Jean-Pierre Hubaux. Mix-zones for location privacy in vehicular networks. In *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, number LCA-CONF-2007-016, 2007.

[27] Tobias Kölsch, Lothar Fritsch, Markulf Kohlweiss, and Dogan Kesdogan. Privacy for profitable location based services. In *International Conference on Security in Pervasive Computing*, pages 164–178. Springer, 2005.

[28] Tom Rodden, Adrian Friday, Henk Muller, Alan Dix, et al. A lightweight approach to managing privacy in location-based services. 2002.

[29] Rongxing Lu, Xiaodong Lin, Tom H Luan, Xiaohui Liang, and Xuemin Shen. Pseudonym changing at social spots: An effective strategy for location privacy in vanets. *IEEE Transactions on Vehicular Technology*, 61(1):86–96, 2012.

[30] Marius Wernke, Pavel Skvortsov, Frank Dürr, and Kurt Rothermel. A classification of location privacy attacks and approaches. *Personal and ubiquitous computing*, 18(1):163–175, 2014.

[31] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. In *Pervasive Services, 2005. ICPS'05. Proceedings. International Conference on*, pages 88–97. IEEE, 2005.

[32] Ben Niu, Zhengyan Zhang, Xiaoqing Li, and Hui Li. Privacy-area aware dummy generation algorithms for location-based services. In *Communications (ICC), 2014 IEEE International Conference on*, pages 957–962. IEEE, 2014.

[33] Hua Lu, Christian S Jensen, and Man Lung Yiu. Pad: privacy-area aware, dummy-based location privacy in mobile services. In *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 16–23. ACM, 2008.

[34] Ben Niu, Qinghua Li, Xiaoyan Zhu, Guohong Cao, and Hui Li. Achieving k-anonymity in privacy-aware location-based services. In *INFOCOM, 2014 Proceedings IEEE*, pages 754–762. IEEE, 2014.

[35] Hyo Jin Do, Young-Seob Jeong, Ho-Jin Choi, and Kwangjo Kim. Another dummy generation technique in location-based services. In *Big Data and Smart Computing (BigComp), 2016 International Conference on*, pages 532–538. IEEE, 2016.

[36] Takahiro Hara, Akiyoshi Suzuki, Mayu Iwata, Yuki Arase, and Xing Xie. Dummy-based user location anonymization under real-world constraints. *IEEE Access*, 4:673–687, 2016.

[37] Mehmet Ercan Nergiz, Maurizio Atzori, and Yucel Saygin. Towards trajectory anonymization: a generalization-based approach. In *Proc. of the SIGSPATIAL ACM GIS*, pages 52–61. ACM, 2008.

[38] 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)*, 5(3):44, 2014.

[39] Roman Yarovoy, Francesco Bonchi, Laks VS Lakshmanan, and Wendy Hui Wang. Anonymizing moving objects: How to hide a mob in a crowd? In *Proc. of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 72–83. ACM, 2009.

[40] Yulan Dong and Dechang Pi. Novel privacy-preserving algorithm based on frequent path for trajectory data publishing. *Knowledge-Based Systems*, 148:55–65, 2018.

[41] Bo Liu, Wanlei Zhou, Tianqing Zhu, Longxiang Gao, and Yong Xiang. Location privacy and its applications: a systematic study. *IEEE access*, 6:17606–17624, 2018.

[42] Giorgos Poulis, Grigorios Loukides, Spiros Skiadopoulos, and Aris Gkoulalas-Divanis. Anonymizing datasets with demographics and diagnosis codes in the presence of utility constraints. *Journal of biomedical informatics*, 65:76–96, 2017.

[43] Tsubasa Takahashi and Shinya Miyakawa. Cmoa: Continuous moving object anonymization. In *Proceedings of the 16th International Database Engineering & Applications Sysmposium*, pages 81–90. ACM, 2012.

[44] Xuyang Zhou and Meikang Qiu. A k-anonymous full domain generalization algorithm based on heap sort. In *International Conference on Smart Computing and Communication*, pages 446–459. Springer, 2018.

[45] Kristen LeFevre, David J DeWitt, and Raghu 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.

[46] Saba Yaseen, Syed M Ali Abbas, Adeel Anjum, Tanzila Saba, Abid Khan, Saif Ur Rehman Malik, Naveed Ahmad, Basit Shahzad, and Ali Kashif Bashir. Improved generalization for secure data publishing. *IEEE Access*, 6:27156–27165, 2018.

[47] Giorgos Poulis, Aris Gkoulalas-Divanis, Grigorios Loukides, Spiros Skiadopoulos, and Christos Tryfonopoulos. Secreta: A tool for anonymizing relational, transaction and rt-datasets. In *Medical Data Privacy Handbook*, pages 83–109. Springer, 2015.

[48] KC Sreedhar, MN Faruk, and B Venkateswarlu. A genetic tds and bug with pseudo-identifier for privacy preservation over incremental data sets. *Journal of Intelligent & Fuzzy Systems*, 32(4):2863–2873, 2017.

[49] Muhammad Ehsan Rana, Manoj Jayabalan, and Mohung Abdoolah Aasif. Privacy preserving anonymization techniques for patient data: An overview. In *Third International Congress on Technology, Communication and Knowledge (ICTCK 2016)*, 2016.

[50] Manoj Jayabalan and Muhammad Ehsan Rana. Anonymizing healthcare records: A study of privacy preserving data publishing techniques. *Advanced Science Letters*, 24(3):1694–1697, 2018.

[51] Deepak Narula, Pardeep Kumar, and Shuchita Upadhyaya. Privacy preservation using various anonymity models. In *Cyber Security: Proceedings of CSI 2015*, pages 119–130. Springer, 2018.

[52] Jiaxin Ding. Trajectory mining, representation and privacy protection. In *Proceedings of the 2nd ACM SIGSPATIAL PhD Workshop*, page 2. ACM, 2015.

[53] A Ercument Cicek, Mehmet Ercan Nergiz, and Yucel Saygin. Ensuring location diversity in privacy-preserving spatio-temporal data publishing. *The VLDB JournalThe International Journal on Very Large Data Bases*, 23(4):609–625, 2014.

[54] Cristina Romero-Tris and David Megías. Protecting privacy in trajectories with a user-centric approach. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(6):67, 2018.

[55] Felipe T Brito, Antônio C Araújo Neto, Camila F Costa, André LC Mendonça, and Javam C Machado. A distributed approach for privacy preservation in the publication of trajectory data. In *Proceedings of the 2nd Workshop on Privacy in Geographic Information Collection and Analysis*, page 5. ACM, 2015.

[56] Manolis Terrovitis, Giorgos Poulis, Nikos Mamoulis, and Spiros Skiadopoulos. Local suppression and splitting techniques for privacy preserving publication of trajectories. *IEEE Trans. Knowl. Data Eng*, 29(7):1466–1479, 2017.

[57] Elham Naghizade, Lars Kulik, and Egemen Tanin. Protection of sensitive trajectory datasets through spatial and temporal exchange. In *Proc. of the 26th International Conference on Scientific and Statistical Database Management*, page 40. ACM, 2014.

[58] Kaifeng Jiang, Dongxu Shao, Stéphane Bressan, Thomas Kister, and Kian-Lee Tan. Publishing trajectories with differential privacy guarantees. In *Proc. of the 25th International Conference on Scientific and Statistical Database Management*, page 12. ACM, 2013.

[59] Daniel Arribas-Bel and Jessie Bakens. Use and validation of location-based services in urban research: An example with dutch restaurants. *Urban Studies*, page 0042098018779554, 2018.

[60] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. Tour recommendation and trip planning using location-based social media: a survey. *Knowledge and Information Systems*, pages 1–29, 2018.

[61] Xin Chen, Fangcao Xu, Weili Wang, Yikang Du, and Miaoyi Li. Geographic big data's applications in retailing business market. In *Big data support of urban planning and management*, pages 157–176. Springer, 2018.

[62] Facebook privacy breach. https://www.ft.com/content/87184c40-2cfe-11e8-9b4b-bc4b9f08f381, Mar 2018.

[63] John Krumm. Realistic driving trips for location privacy. In *International Conference on Pervasive Computing*, pages 25–41. Springer, 2009.

[64] Bo Liu, Wanlei Zhou, Shui Yu, Kun Wang, Yu Wang, Yong Xiang, and Jin Li. Home location protection in mobile social networks: a community based method (short paper). In *International Conference on Information Security Practice and Experience*, pages 694–704. Springer, 2017.

[65] Alastair R Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1):46–55, 2003.

[66] Tao Jiang, Helen J Wang, and Yih-Chun Hu. Preserving location privacy in wireless lans. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 246–257. ACM, 2007.

[67] Chi-Yin Chow, Mohamed F Mokbel, and Xuan 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.

[68] Man Lung Yiu, Christian S Jensen, Jesper Møller, and Hua Lu. Design and analysis of a ranking approach to private location-based services. *ACM Transactions on Database Systems (TODS)*, 36(2):10, 2011.

[69] Roman Schlegel, Chi-Yin Chow, Qiong Huang, and Duncan S Wong. User-defined privacy grid system for continuous location-based services. *IEEE Transactions on Mobile Computing*, 14(10):2158–2172, 2015.

[70] Sina Shaham, Matthew Kokshoorn, Ming Ding, Zihuai Lin, and Mahyar Shirvanimoghaddam. Extended kalman filter beam tracking for millimeter wave vehicular communications. *arXiv preprint arXiv:1911.01638*, 2019.

[71] Sina Shaham, Saba Rafieian, Ming Ding, Mahyar Shirvanimoghaddam, and Zihuai Lin. On the importance of location privacy for users of location based applications. *arXiv preprint arXiv:1911.01633*, 2019.

[72] Zheng Xiang Ma, Min Zhang, Sina Shaham, Shu Ping Dang, and Jessica Hart. Literature review of the communication technology and signal processing methodology based on the smart grid. In *Applied Mechanics and Materials*, volume 719, pages 436–442. Trans Tech Publ, 2015.

[73] Sina Shaham, Matthew Kokshoorn, Zihuai Lin, Ming Ding, and Yi Wu. Raf: Robust adaptive multi-feedback channel estimation for millimeter wave mimo systems. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2018.

[74] Reynold Cheng, Yu Zhang, Elisa Bertino, and Sunil Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *International Workshop on Privacy Enhancing Technologies*, pages 393–412. Springer, 2006.

[75] Rigzin Angmo, Veenu Mangat, and Naveen Aggarwal. Preserving user location privacy in era of location-based services: Challenges, techniques and framework. In *Proceedings of 2nd International Conference on Communication, Computing and Networking*, pages 43–52. Springer, 2019.

[76] Ayoob Salari and Wan Choi. Maximizing received energy in magnetic resonance wireless power transfer using feedback. *IEEE Transactions on Green Communications and Networking*, 2019.

[77] Xiangjie Kong, Feng Xia, Zhaolong Ning, Azizur Rahim, Yinqiong Cai, Zhiqiang Gao, and Jianhua Ma. Mobility dataset generation for vehicular social networks based on floating car data. *IEEE Transactions on Vehicular Technology*, 67(5):3874–3886, 2018.

[78] Xiangjie Kong, Menglin Li, Kai Ma, Kaiqi Tian, Mengyuan Wang, Zhaolong Ning, and Feng Xia. Big trajectory data: A survey of applications and services. *IEEE Access*, 6:58295–58306, 2018.

[79] Junchuan Fan, Cheng Fu, Kathleen Stewart, and Lei Zhang. Using big gps trajectory data analytics for vehicle miles traveled estimation. *Transportation Research Part C: Emerging Technologies*, 103:298–307, 2019.

[80] Hassan Talat, Tuaha Nomani, Mujahid Mohsin, and Saira Sattar. A survey on location privacy techniques deployed in vehicular networks. In *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 604–613. IEEE, 2019.

[81] Al Franken. Text - s.1223 - 112th congress (2011-2012): Location privacy protection act of 2012, Dec 2012.

[82] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[83] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying 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.

[84] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 312–321. ACM, 2008.

[85] Yu Zheng, Xing Xie, and Wei-Ying Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33(2):32–39, 2010.

[86] Lyria Bennett Moses, Genna Churches, Emily Watson, and Monika Zalnieriute. Submission to the data sharing and release legislative reforms discussion paper. *UNSW Law Research Paper*, (19-79), 2019.

[87] Bassam S Ali, Osman Nuri Ucan, and Oguz Bayat. A novel approach for ensuring location privacy using sentiment analysis and analysis for health-care and its effects on humans health. *Journal of Medical Imaging and Health Informatics*, 10(1):178–184, 2020.

[88] Achilleas Papageorgiou, Michael Strigkos, Eugenia Politou, Efthimios Alepis, Agusti Solanas, and Constantinos Patsakis. Security and privacy analysis of mobile health applications: the alarming state of practice. *IEEE Access*, 6:9390–9403, 2018.

[89] Zhongli Wang, Shuping Dang, Sina Shaham, Zhenrong Zhang, and Zhihan Lv. Basic research methodology in wireless communications: The first course for research-based graduate students. *IEEE Access*, 7:86678–86696, 2019.

[90] Acar Tamersoy, Grigorios Loukides, Mehmet Ercan Nergiz, Yucel Saygin, and Bradley Malin. Anonymization of longitudinal electronic medical records. *IEEE Transactions on Information Technology in Biomedicine*, 16(3):413–423, 2012.

[91] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1241–1250. International World Wide Web Conferences Steering Committee, 2017.

[92] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[93] Vijay S Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 279–288. ACM, 2002.

[94] Biswanath Chowdhury and Gautam Garai. A review on multiple sequence alignment from the perspective of genetic algorithm. *Genomics*, 2017.

[95] Xi Chen, Chen Wang, Shanjiang Tang, Ce Yu, and Quan Zou. Cmsa: a heterogeneous cpu/gpu computing system for multiple similar rna/dna sequence alignment. *BMC bioinformatics*, 18(1):315, 2017.

[96] Quan Le, Fabian Sievers, and Desmond G Higgins. Protein multiple sequence alignment benchmarking through secondary structure prediction. *Bioinformatics*, 33(9):1331–1337, 2017.

[97] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[98] Sankar K Pal and Paul P Wang. *Genetic algorithms for pattern recognition*. CRC press, 2017.

[99] Aurélie Fischer and Dominique Picard. Convergence rates for smooth k-means change-point detection. *arXiv preprint arXiv:1802.07617*, 2018.

[100] Yanming Sun, Min Chen, Long Hu, Yongfeng Qian, and Mohammad Mehedi Hassan. Asa: Against statistical attacks for privacy-aware users in location based service. *Future Generation Computer Systems*, 70:48–58, 2017.

[101] Onur Asan, Farion Cooper II, Sneha Nagavally, Rebekah J Walker, Joni S Williams, Mukoso N Ozieh, and Leonard E Egede. Preferences for health information technologies among us adults: Analysis of the health information national trends survey. *Journal of medical Internet research*, 20(10):e277, 2018.

[102] Carol Gates. Electronic medical record reminder to improve human papillomavirus vaccination rates among adolescents. 2018.

[103] Fuad Rahman, Panagiotes Karanikas, and Thomas D Giles. Systems and methods for creating contextualized summaries of patient notes from electronic medical record systems, August 17 2017. US Patent App. 15/430,401.

[104] José Colleti Junior, Alice Barone de Andrade, and Werther Brunow de Carvalho. Evaluation of the use of electronic medical record systems in brazilian intensive care units. *Revista Brasileira de terapia intensiva*, 30(3):338–346, 2018.