PRIVACY-PRESERVING TRAJECTORY DATA PUBLISHING VIA DIFFERENTIAL PRIVACY

by

Ishita Dwivedi

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Computer Science

Boise State University

December 2017

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Ishita Dwivedi

Thesis Title: Privacy-Preserving Trajectory Data Publishing Via Differential Privacy Date

of Final Oral Examination: 17 October 2017

The following individuals read and discussed the thesis submitted by student Ishita Dwivedi, and they evaluated the presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Gaby Dagher, Ph.D.	Chair, Supervisory Committee
Dianxiang Xu, Ph.D.	Member, Supervisory Committee
Yantian Hou, Ph.D.	Member, Supervisory Committee

The final reading approval of the thesis was granted by Gaby Dagher, Ph.D., Chair of the Supervisory Committee. The thesis was approved by the Graduate College.

dedicated to ”my mother”

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Gaby Dagher for constant guidance throughout my research. He provided my research with proper direction. I would also like to thank my committee members for their valuable feedbacks throughout the term of my research. Also, I would like to thank my family and friends for the constant support and motivation.

Abstract

Over the past decade, the collection of data by individuals, businesses and government agencies has increased tremendously. Due to the widespread of mobile computing and the advances in location-acquisition techniques, an immense amount of data concerning the mobility of moving objects have been generated. The movement data of an object (e.g. individual) might include specific information about the locations it visited, the time those locations were visited, or both. While it is beneficial to share data for the purpose of mining and analysis, data sharing might risk the privacy of the individuals involved in the data. Privacy-Preserving Data Publishing (PPDP) provides techniques that utilize several privacy models for the purpose of publishing useful information while preserving data privacy.

The objective of this thesis is to answer the following question: How can a data owner publish trajectory data while simultaneously safeguarding the privacy of the data and maintaining its usefulness? We propose an algorithm for anonymizing and publishing trajectory data that ensures the output is differentially-private while maintaining high utility and scalability. Our solution comprises a twofold approach. First, we generalize trajectories by generalizing and then partitioning the timestamps at each location in a differentially-private manner. Next, we add noise to the real count of the generalized trajectories according to the given privacy budget to enforce differential privacy. As a result, our approach achieves an overall ϵ-differential privacy on the output trajectory data. We perform experimental evaluation on real-life data, and demonstrate that our proposed approach can effectively answer count and range queries, as well as mining frequent sequential patterns. We also

show that our algorithm is efficient w.r.t. privacy budget and number of partitions, and also scalable with increasing data size.

Contents

List of Tables

List of Figures

LIST OF ABBREVIATIONS

EH – External Homogeneity

EGH – External Gap Homogeneity

EGH^{Occ} – External CountGap Homogeneity

IH – Internal Homogeneity

IGH – Internal Gap Homogeneity

IGH^{Occ} – Internal CountGap Homogeneity

LIST OF SYMBOLS

ϵ	epsilon symbol, used as the privacy parameter
θ	theta symbol, denotes the number of pivot timestamps
α	alpha symbol, denotes the size of each pivot-cluster

Chapter 1

INTRODUCTION

1.1 Motivation

Over the past decade, the collection of data by individuals, businesses and government agencies has increased tremendously. While it is beneficial to share data for the purpose of mining and analysis, data sharing might risk the privacy of the individuals involved in the data. Privacy-reserving data publishing (PPDP) [32] provides techniques that utilize several privacy models for the purpose of publishing useful information while preserving data privacy. Unlike differential privacy [25], other privacy models such as k-anonymity [94], l-diversity [63] and t-closeness [59] do not fully protect against attacks that are based on the prior knowledge of the adversary about individuals in data. Such attacks include table-linkage attacks, attribute-linkage attacks, and probabilistic attacks. On the other hand, differential privacy overcomes such attacks and makes no assumptions about the background knowledge an adversary may have, and does not reveal the participation of an individual in the published data [116]. In this thesis, our goal is to achieve differential privacy guarantee on the published data.

Due to the widespread of mobile computing and the advances in location-acquisition techniques, an immense amount of data concerning the mobility of moving objects have been generated. The movement data of an object (e.g. individual) might include specific information about the locations it visited, the time those locations were visited, or both. In

general, the origin of movement data can be the mobility of either people, vehicles, animals or natural phenomena [115]. Furthermore, movement data can be broadly classified into *sequential* and *trajectory* data. Sequential data contains a set of sequences, where each sequence lists in chronological order the locations visited by a moving object. On the other hand, the movement of an object in a trajectory data is represented as a sequence of doublets (l, t) representing the location l that was visited at timestamp t. In this thesis, we consider the problem of publishing vehicle trajectory data.

1.2 Challenges & Concerns

While trajectory data can be used to perform several mining tasks, including trajectory pattern mining, trajectory classification and trajectory outlier detection [115], publishing trajectory data imposes several concerns.

A major concern is *data privacy*. The application of typical privacy models such as [94][63][59] via privacy-preserving data publishing techniques on trajectory data does not protect the published data against privacy attacks, which include:

- Background knowledge attacks, where an adversary utilizes its background knowledge about an individual in the trajectory data to infer sensitive information about said individual.

- Probabilistic attacks, where an adversary can infer the presence or absence of an individual's trajectory in the published trajectory data.

Example 1.2.1 illustrates the privacy concern in trajectory data publishing.

Example 1.2.1. Let Table 1.1 represent a raw trajectory dataset which comprises of eight trajectories, each of which is linked to a sensitive attribute. Table 1.2 represents a 2-

Table 1.1: **Raw trajectory data**

Index	Trajectory
$\mathcal{T}_1$	c1 $\rightarrow$ b2 $\rightarrow$ c4 $\rightarrow$ d5
$\mathcal{T}_2$	b5 $\rightarrow$ d8 $\rightarrow$ a10
$\mathcal{T}_3$	a2 $\rightarrow$ b3 $\rightarrow$ e5 $\rightarrow$ b7
$\mathcal{T}_4$	e6 $\rightarrow$ a7 $\rightarrow$ c9
$\mathcal{T}_5$	b3 $\rightarrow$ e5 $\rightarrow$ c9
$\mathcal{T}_6$	e6 $\rightarrow$ a7 $\rightarrow$ c9 $\rightarrow$ d10
$\mathcal{T}_7$	a1 $\rightarrow$ b2 $\rightarrow$ d4 $\rightarrow$ e5
$\mathcal{T}_8$	a3 $\rightarrow$ b7 $\rightarrow$ d8

Table 1.2: **2-anonymous trajectory data based on Table 1.1**

Index	Anonymous Trajectory
$\hat{\mathcal{T}}_1$	b3 $\rightarrow$ e5
$\hat{\mathcal{T}}_2$	e6 $\rightarrow$ a7 $\rightarrow$ c9
$\hat{\mathcal{T}}_3$	b3 $\rightarrow$ e5
$\hat{\mathcal{T}}_4$	e6 $\rightarrow$ a7 $\rightarrow$ c9

anonymous version of the raw data. If an adversary knows that an individual visited location b at timestamp 3, the adversary can determine from Table 1.2 that either trajectory $\hat{\mathcal{T}}_1$ or trajectory $\hat{\mathcal{T}}_3$ represents that individual. As a result, the adversary can infer with 100% confidence that the individual visited location e at timestamp 5, given that doublet $e5$ exists in both trajectories. Moreover, if the sensitive attributes associated with $\hat{\mathcal{T}}_1$ and $\hat{\mathcal{T}}_3$ are the same, then the adversary can perform *homogeneity attack* and infer with 100% confidence the sensitive attribute of that individual. Otherwise, it will be able to infer the sensitive attribute with 50% confidence. ■

Another concern with respect to publishing trajectory data is *high-dimensionality*. Trajectories might consist of a long sequence of doublets, thus increasing the dimensionality of the data. As a result, publishing trajectory data will typically produce low utility output due to the curse of high dimensionality [23].

Example 1.2.2. Assume that a metro system runs between 50 stations for 22 hours every day. The number of doublets that can be possibly generated based on the possible locations and timestamps are $50 * (22 * 3600) = 3,960,000$ doublets (assuming accuracy is to the second), which also represents the total possible dimensions in the data. ■

Another concern with respect to publishing trajectory data is *sparseness*. For example, in a taxi trajectory data, a taxi can visit only a few locations over a period of time, which results in the raw data being sparsely populated as each trajectory consists of a small subset of all possible doublets. Also, a trajectory of any taxi can contain a limited number of doublets because a taxi can be at only one location at a given time. Sparseness and high dimensionality in raw data typically leads to reduction in size in the anonymized output due to the suppression of trajectories, as illustrated in Example 1.2.3.

Example 1.2.3. Due to the sparsity and high dimensionality of the doublets in Table 1.1, the 2-anonymous data in Table 1.2 contains only 4 trajectories, even though the raw data contains 8 trajectories. ■

Extensive research [79][89][74][1][17][95] has been done regarding movement data sharing while preserving the privacy of the individuals involved, mostly using differential privacy[11][30][49][46][85]. One such research [46] for applying differential privacy to publish trajectory data, aims to generalize the trajectories by generalizing the locations and adding noise to the number of occurrences of trajectories to ensure differential privacy. The approach we propose in this thesis also applies the differential privacy model and is distinct from the previous approaches because it provides high utility for count queries, range queries, and frequent sequential pattern mining. More specifically, in this thesis, we propose an algorithm for anonymizing and publishing trajectory data, such that the output is differentially-private while maintaining high utility. Our solution comprises of a

twofold approach. In the first phase (Phase 1), we generalize trajectories by generalizing and partitioning the timestamps at each location while guaranteeing differential privacy. In the next phase (Phase 2), we add noise to the real count of the generalized trajectories to ensure differential privacy. As a result, our approach achieves an overall ϵ-differential privacy on the output trajectory data.

1.3 Thesis Statement

The objective of this thesis is to answer the following question: **How can a data owner publish its trajectory data while simultaneously safeguarding the privacy of the data and maintaining its usefulness?**

More specifically, given a trajectory dataset $\mathbb{D} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 .... \mathcal{T}_{|\mathbb{D}|}\}$ and a privacy budget ϵ, the goal of this thesis is to propose an approach for generating and publishing an anonymized version of the data $\hat{\mathbb{D}}$ for the purpose of data mining and analysis such that:

1. The published data $\hat{\mathbb{D}}$ satisfies differential privacy.
2. The published data $\hat{\mathbb{D}}$ maintains high utility.
3. The proposed approach is efficient and scalable.

1.4 Organization of the Thesis

This Thesis is organized as follows:

- Chapter 2 discusses the background knowledge needed for a better understanding of the terms used throughout this thesis.

- Chapter 3 discusses the related literature over the past years in the fields related to publishing and mining movement data and other types of data, via differential privacy or other privacy models.

- Our algorithm for publishing trajectory data using differential privacy is proposed in Chapter 4. We discuss in detail the proposed algorithms and how differential privacy is achieved over different steps before the data can be published.

- Chapter 5 discusses the properties of the datasets we use to perform our experiments. We test the performance of the proposed algorithms by measuring its scalability, efficiency and utility over two trajectory datasets.

- We conclude the thesis in Chapter 6 with a discussion about future work.

Chapter 2

BACKGROUND

We begin this section by a presenting an overview of movement data (trajectory and sequential), and then we introduce differential privacy as a mechanism for data anonymization.

2.1 Movement Data

Movement data represents the actual movement of an individual over a period of time. It is essential to effectively hide the identity of the individuals in the movement data before the data is shared. This has led to a vast amount of research in the field of movement data publishing in general. Over the last few years, after the authors of [25] introduced the concept of differential privacy, research has changed focus to publishing movement data using differential privacy. Movement data is typically presented as either trajectory or sequential data. Since our approach is designed mainly for trajectory data, we will introduce the notations that we will be using through the rest of this thesis.

Definition 2.1.1. Trajectory. A trajectory $\mathcal{T}_i$ represents information about the displacements of an individual i, wherein time t is taken into account and the trajectory can be represented as a series of doublets:

$$\mathcal{T}_i = (l_1, t_1) \rightarrow (l_2, t_2) \rightarrow (l_3, t_3) \rightarrow .... \rightarrow (l_{|\mathcal{T}_i|}, t_{|\mathcal{T}_i|}) \tag{2.1}$$

where each doublet comprises of a timestamp t_j and a location l_j, and doublets in a trajectory are ordered chronologically according to their timestamps. ■

Definition 2.1.2. Trajectory data. Trajectory data $\mathbb{D}$ is a set of trajectories owned by a data owner, and represented as:

$$\mathbb{D} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3....\mathcal{T}_{|\mathbb{D}|}\} \tag{2.2}$$

where each trajectory $\mathcal{T}_i$ represents an individual. ■

2.2 Differential Privacy

We begin this section by introducing differential privacy. We discuss the concept of sensitivity and the mechanisms that exist for achieving ϵ-differential privacy, namely exponential mechanism and Laplace mechanism.

2.2.1 ϵ-Differential Privacy

Differential privacy [25], proposed by Cynthia Dwork, aims to achieve a strong guarantee that the presence or absence of an individual cannot be inferred when analyzing differentially-private published data, regardless of the background of the attacker.

ϵ-differential privacy is an extension of the general idea of differential privacy, where ϵ is the privacy budget (privacy parameter) that controls the level of privacy provided when differential privacy is applied to a raw dataset.

Definition 2.2.1. ***ϵ-Differential Privacy [25].*** A randomized function $\mathcal{K}$ provides ϵ-differential privacy if for all datasets D_1 and D_2 differing in at most one record, and all possible outputs $S \subseteq Range(\mathcal{K})$, then:

$$\Pr\left[\mathcal{K}(D_1) \in \mathrm{S}\right] \leqslant \exp\left(\epsilon\right) \times \Pr\left[\mathcal{K}(D_2) \in \mathrm{S}\right] \tag{2.3}$$

where parameter ϵ is the privacy level. ■

Note that privacy budget ϵ affects privacy and accuracy (and therefore utility) of the generated differentially-private data such that a lesser value of ϵ means stronger privacy but also poor utility.

Sensitivity

Mechanisms such as noise addition and exponential mechanism are utilized in order to achieve differential privacy. However, the effectiveness of such mechanisms depends on *data sensitivity*. Sensitivity can be informally defined as the maximum possible change in the utility function when a single record is either removed, added, or altered. More formally:

Definition 2.2.2. ***Sensitivity [25].*** Given a function f : D $\rightarrow \mathbb{R}^d$ over a domain D, the sensitivity of f is defined as:

$$\Delta(f) = \max_{D_1, D_2} \parallel \mathrm{f}(D_1) - \mathrm{f}(D_2) \parallel \tag{2.4}$$

where D_1 and D_2 are neighboring datasets that differ in a maximum of one record. ■

Sensitivity varies depending on the type of data, which affects the utility of the output since it is impacted by the level of noise added.

Exponential Mechanism

McSherry and Talwar [67] proposed a technique to achieve differential privacy using exponential mechanism. Exponential mechanism determines the outcome by taking as input the score *q* ($Utility_{each}$) generated by the utility function, the dataset *D*, an output range Γ and privacy budget ϵ. The outcome is chosen with the following probability:

$$\frac{exp\frac{\epsilon}{2\Delta u}(Utility_{each})}{\sum_{i=1}^{All}\frac{\epsilon}{2\Delta u}(Utility_i)} \tag{2.5}$$

where Δu is the sensitivity of the utility function.

The exponential mechanism results in the probability distribution over the output range Γ. Sampling is done over this probability distribution in order to obtain an output. As seen in equation 2.5, the probability of any output is directly proportional to $exp\frac{\epsilon}{2\Delta u}(Utility_{each})$, which means it is directly proportional to the score value of the utility. This leads us to the theorem 2.2.1 from [67].

Theorem 2.2.1. For any function having utility score $Utility$, an algorithm that chooses an output with probability directly proportional to $exp\frac{\epsilon}{2\Delta u}(Utility)$ satisfies ϵ-differential privacy. ■

Adding Laplace Noise

Dwork et al. [25] proposed *Laplace mechanism* to obtain differential privacy by adding noise. It begins by first computing the true solution to a given function over a dataset *D*. The value obtained is distorted by adding to it a noise from the Laplace distribution. Given a function $f(D)$ where *D* is original data, after applying Laplace mechanism to add a noise, the noisy value for the function is represented as follows:

$$f(\boldsymbol{D}') = f(\boldsymbol{D}) + Lap(\lambda) \tag{2.6}$$

where $Lap(\lambda)$ represents the Laplace noise sampled from Laplace distribution having probability density function (PDF):

$$Pr(x\|\lambda) = \frac{1}{2\lambda} exp(-|x|/\lambda) \tag{2.7}$$

, where variance is $2\lambda^2$ centered at 0.

Theorem 2.2.2. [25] Given a function $f : D \rightarrow \mathbb{R}^d$, the algorithm that adds Laplace noise with probability distribution $Lap(\Delta f/\epsilon)$ to each *d* output will always satisfy ϵ-differential privacy. ■

2.2.2 (ϵ, δ)-Differential Privacy

Data can be said to be differentially-private when a participant's data is altered (added/removed) in the considered dataset which leads to a minute change in the generated differentially-private data.

For input data *D* to a randomized algorithm $\mathcal{K}$, the random variable corresponding to *D* is $\mathcal{K}(D)$. The probability of the event is not similar as compared to more probable events, under the distribution $\mathcal{K}(D_1)$ and $\mathcal{K}(D_2)$ because the metric in differential privacy is multiplicative.

This necessary condition for differential privacy was later relaxed in following research and (ϵ,δ)-differential privacy is a more relaxed differential privacy model as compared to the ϵ-differential privacy that is a stronger privacy guarantee nonetheless. (ϵ, δ)-differential privacy is defined as :

Definition 2.2.3. *(ϵ, δ)-Differential Privacy: A randomized algorithm $\mathcal{K}$ is (ϵ,δ)-differentially-private if for all databases D_1, $D_2 \in (D)^n$ that varies in one individual's records:*

$$Pr\,[\mathcal{K}(D_1) \in S] \leqslant exp(\epsilon) \times Pr\,[\mathcal{K}(D_2) \in S] + \delta \tag{2.8}$$

, where S represents all subsets of outputs. [58]. ■

Note that when $\delta = 0$, Equation 2.8 represents ϵ-differential privacy.

2.2.3 ϵ_i-Differential Privacy

The authors in [68] discuss two techniques for guaranteeing privacy in the case of multiple data releases.

($\sum_i \epsilon_i$)-Differential Privacy: When there is a sequential series of analysis (release), each of which satisfies ϵ-differential privacy, then the sum of ϵ values can be added to generate ($\sum_i \epsilon_i$)-Differential Privacy. ($\sum_i \epsilon_i$)-Differential Privacy is also referred to as ***sequential composition***.

($max_i\ \epsilon_i$)-Differential Privacy: In the case of $max_i\ \epsilon_i$)-differential privacy, unlike sequential composition, structurally disjoint subsets of the data are analyzed. This disjoint (parallel) subset's sequence of analysis provides ($max_i\ \epsilon_i$)-differential privacy, which is also referred to as ***parallel composition***.

2.3 Differential Privacy Guarantee

Differential privacy, introduced in [26] is different from previous privacy definitions which attempt to prevent data leakage and disclosure, as well as other privacy violations. Differ-

ential privacy aims at preventing attackers from obtaining knowledge about the presence of an individual's records in a published data. It provides a strong guarantee that the presence or absence of of an individual record will have no effect on the result of analysis on the published data. In other words, the result to a query on a differentially-private dataset will remain almost the same with the presence or absence of an individual. Essentially, differential privacy provides ϵ privacy guarantee such that for an appropriate ϵ value, a mechanism $\mathcal{K}$ (see Equation 2.3) satisfies the definition of differential privacy. If the input datasets are almost identical in various randomized computations, the outcome distribution will also be nearly identical.

The authors in [34] state that unlike previous ad-hoc guarantees that provided security against only certain attacks, differential privacy provides an ad-omnia guarantee. Also, since differential privacy is able to provide a rigid because it is independent of the computational power of the adversary and their knowledge of any background information [26]. This suggests that differential privacy achieves privacy over data through *uncertainty*, *i.e.* via randomization. Therefore, it isn't possible for any output to reveal a single individual's data with *certainty*.

Table linkage attack is possible if the data recipient is able to confidently determine whether the individual's record exists in the released data table. Since differential privacy guarantees that presence or absence of an individual's record in the original data will have no significant effect on the generated output, this privacy model guarantees against the possibility of a table linkage attack.

In [56], however, the authors suggest that without any knowledge or assumption about the data, when differential privacy is applied to social networks, or when deterministic statistics have been previously released, the privacy guarantee could possibly degrade. This led to the development of new techniques for maintaining differential privacy guarantee in

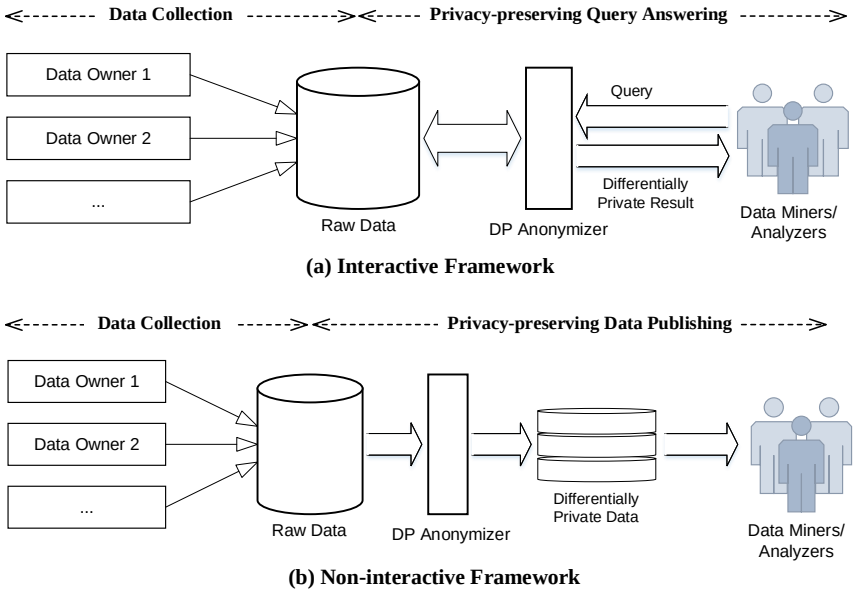


Figure 2.1: Interactive vs non-interactive frameworks of differential privacy.

the case of complex data publishing scenarios (see Section 3.2 for more details).

2.4 Interactive Differential Privacy

Instead of the data owner publishing its data in a differentially-private manner (non-interactive), the data miner/analyst could pose queries directly to the data owner. Figure 2.1 illustrates the interactive and non-interactive frameworks of differential privacy. The system could be already aware of all the queries that will be posed by the data miner in advance and it could take the appropriate measures to make the data private. However, in most cases (interactive queries), the system would respond to the posed ad-hoc queries without any knowledge of the queries or any insight into the future. *Privacy-preserving query processing* is the task wherein queries posed over statistical data are answered by injecting random noise to each of the responses to guarantee the privacy of an individual by making their presence or absence in the data unclear. A number of techniques have been proposed for predetermined

query processing [112][52], as well interactive query processing [70][31].

Privacy-preserving data mining is the task of mining information from a data wherein the data owner is responsible for maintaining the privacy guarantee in the answers sent to the data miners. A vast amount of research has proposed privacy-preserving data mining techniques under differential privacy [31][44][2][44][82] and non-differential privacy [29]. Privacy-preserving data analysis is the set of tasks where the published or mined data is analyzed by a set of analysis algorithms performed by data analysts, while maintaining efficient privacy guarantees with reference to the privacy of individual's records in the data. A number of papers have proposed privacy-preserving data analysis techniques under differential privacy [80][28][40][41][53][65][103][70] [69].

Chapter 3

LITERATURE REVIEW

Below, we review the most relevant research work in the literature.

3.1 Privacy-Preserving Data Publishing

Privacy-preserving data publishing (PPDP) provides the tools and methods for publishing data while preserving the privacy of the entities stored in the data, as well as maintaining utility of the anonymized published data. Distributed privacy-preserving data publishing (DPPDP) is a decentralized version of PPDP where multiple parties are involved in the process of data publishing. In the literature, several privacy mechanisms including k-anonymity [94], l-diversity [63], t-closeness [59] and differential privacy [25] have been suggested to publish various types of data. However, k-anonymity and differential privacy have been the most widely utilized mechanisms for publishing movement data while maintaining the privacy of the moving objects stored in the data.

3.1.1 *via* Differential Privacy

Differential privacy was introduced by Dwork *et al.* [25] for privacy-preserving data publication. Differential Privacy Preserving Data Publishing (PPDP) can be implemented using a number of techniques. In this section we categorize the proposed techniques based on the type of the input data.

Publishing Movement Data

Over the past years, there has been an extensive research for publishing movement data, which includes trajectory and sequential data. Lately, there have also been several works in the fields of trajectory data publishing [11][30][49][46][85], sequential data publishing [16][13][43], and trajectory and sequential data publishing [15].

An approach to publish differentially-private time series data for traffic monitoring was proposed in [30]. The authors introduced two estimation algorithms: the first applies posterior estimation, and for the cells onto which it was applied, a time-series quadtree model is generated. Based on this quadtree model, the second algorithm groups similar cells to generate spacial indexing structure and hence reduces the impact of data sparsity on the approach. Jiang *et al.* [49] proposed another approach for time-series data which applied sphere sampling with the addition of noise. They introduce an approach called *sampling distance and direction* (SDD) which applies exponential mechanism for sampling the next location to be published in a trajectory.

Another approach for publishing trajectory data was introduced in [46], which is an improvement to the techniques proposed in [15][13]. While the previous approaches assume the trajectories have a number of common prefixes or *n-grams* which might not be true for all data, the approach in [46] does not make this assumption. The authors introduced a differentially-private location generalization algorithm which generalizes all trajectories to merge any locations that have the same timestamps. This algorithm applies exponential mechanism to recursively choose from the partitions of the location universe at each timestamp, where the clustering approach that generated the partition replaces the locations in the same cluster by their centroid. Next, they introduce another algorithm to publish these generalized trajectories in a differentially-private manner, by generating

new trajectories based on the generalized locations, and finally publishing the noisy counts after the addition of Laplace noise. Riboni *et al.* [85] proposed a technique that integrated differential privacy and pre-filtering process, explicitly for protecting check-in data so an untrusted adversary is unable to infer check-in details shared by other individuals. Their approach primarily publishes a single version of the differentially-private data by enforcing *(L, j)*-density. They further extend this approach for incremental release by extracting *(L, j)*-private statistics from the dataset that had previously enforced *(L, j)*-density. They further apply Laplace mechanism to add Laplace noise depending on the locations visited that were pruned. Another approach proposed by [11] introduces *l*-trajectory privacy where only certain trajectories published are differentially-private, which is determined by the desired length of the trajectory.

Chen *et al.* [13] acknowledged the drawback of [15] that the number sequences represented in each branch of the prefix tree reduces considerably, thus resulting in poor utility overall. Therefore, they introduce a technique for probabilistic prediction which represents sequential data as variable length *n-grams*; which is similar to *(n-1)*-order Markov Chain Model. Their technique incorporates the addition of Laplace noise to achieve differential privacy. To limit the noise added, they employ an exploration tree that performs adaptive budget allocation and also enforces consistency constraints based on Markov assumption. The technique suggested in [43] proposes an algorithm to synthesize GPS trajectories (without timestamps) using *hierarchal reference systems* (HRS) model. The HRS captures correlations between adjacent locations in regular trajectories and is designed for realistic data which has large spatial domains. *DPT* inputs a uniformly distributed sample of the sequence of locations (latitude-longitude pairs) and applies Laplace mechanism to add Laplace noise and then outputs differentially-private synthetic trajectories.

The authors in [15][16] proposed a differentially-private data sanitization approach for

trajectory data and sequential data. The proposed algorithm inputs raw trajectory data, privacy budget ϵ and the height of the prefix tree, and publishes the sanitized data that satisfies ϵ-differential privacy. The noisy prefix tree PT is constructed by a function in the sanitization algorithm by recursively grouping trajectories in the data into disjoint subsets based on their prefix and employs a set of count queries. Another function of the sanitization algorithm applies a utility boosting technique on PT and generates the sanitized data in a differentially-private manner. The approach supports both count queries and frequent sequential pattern mining.

Publishing Non-Movement Data

The authors in [72] present an ϵ-differentially-private anonymization algorithm called *Diff-Gen* that relies on *generalization* and *specialization*. Given privacy budget ϵ, ϵ/2 is assigned to the generalization and specialization process, while the other ϵ/2 is utilized to add Laplace noise before publishing. Taxonomy trees are used for specialization, where the taxonomy tree represents the predefined hierarchy of the categorical attributes, whereas for numerical attributes split points are adaptively determined. *DiffGen* algorithm begins by generalizing each quasi-identifier attribute in the raw data to its topmost value in the corresponding taxonomy tree, where a root node is created and all data records are assigned to this node. Next, a sequence of *specializations* is performed where the records at the parent node split into disjoint child nodes. Two utility functions, InfoGain and Max, are used for determining the score of specialization of each attribute in the parent node. Exponential mechanism utilizes the heuristic functions and a part of the privacy budget to determine the attribute for specialization in a differentially-private manner. At each level *i*, a part of the generalization and specialization budget $\epsilon/2$ is consumed, where each iteration at the same level in the hierarchy consumes the same privacy budget due to the

parallel composition property. Finally, after 'h' number of specializations, $\epsilon/2$ is added to the remaining budget from the specialization steps for computing the noisy count of each leaf node. If the noisy count is greater than or equal to the given threshold value, then those records will be published. The authors in [71] extended the *DiffGen* algorithm from [72] by designing a securely two-party protocol for publishing vertically-partitioned data while satisfying differential privacy.

Barak *et. al.* [7] combined the privacy mechanism in [28] with the technique they propose to obtain strong privacy, accuracy and consistency in the published data and hence publish a set of marginals from the contingency table. They proposed an algorithm to achieve privacy by applying Laplace noise to the data, which is converted to privacy-protective intermediate data. If the publisher releases this intermediate data, the privacy is preserved. To maintain accuracy and consistency, the approach adds noise by transforming the data to *Fourier domain* which encodes the data as the marginals in a non-redundant way. *Linear programming* is then applied to generate a non-negative contingency table having Fourier coefficient values and later, for integrality, the results are rounded up. Also, the approach suggests that Fourier domain is not necessarily employed, instead, the marginals can directly be perturbed and then linear programming can be used; but in this scenario, the published data might not be as accurate as when Fourier domain is used. Although their focus lies on generating and maintaining a balance between privacy, accuracy and consistency, the proposed algorithm for obtaining differential privacy validates a balance between obtaining privacy, accuracy and consistency.

Yang *et. al.* [108] extends the research of [7] through application of the technique on a number of examples; if there is a possibility of making sensible inferences based on the published data, the extent to which these inferences can be made. Based on the results obtained, they conclude that the proposed technique is unsuitable for publishing large,

sparse contingency tables. On the other hand, [104] considers application of probabilistic inference to the measurements and measurement process of obtaining differentially-private data. They conclude by stating that probabilistic interference and differential privacy are complementary lines of research and application of probabilistic interference improves accuracy, integrates multiple observations and measures uncertainty.

Hardt *et. al.* [42] [40] propose an algorithm – *MWEM* for publishing differentially-private data. *MWEM* combines the concept of exponential mechanism with Multiplicative Weights update rule to achieve ϵ-differential privacy by posing queries non-interactively using exponential mechanism for selecting the best scoring result from the distribution and Laplace mechanism for reporting measurements as approximate sums of bounded functions and addition of Laplace noise; where sum is the result of a linear query on the dataset. Next, multiplicative weights update rule is applied as used in [41] and [39] by continuously improving the approximate distribution. It suggests that, if a query's result on true data is much greater than on approximate data, the approximating weights on the records that are contributing positively should be increased and the approximating weights on the records that are contributing negatively should be reduced, and vice versa if a query's response on true data is much less than on approximate data.

The authors of [18] present an algorithm *F-BCQT* (Filter-Build Consistency Quadtree) for two-dimensional sparse data publication which boosts the accuracy of range queries on the published data. The algorithm consists of two parts: First, two side filter algorithm is used to compress the dataset and obtain the sampling dataset from the original two-dimensional sparse dataset. Next, the incomplete quadtree is built based on the sampling dataset, where a quadtree is such a tree data structure whose each internal node has exactly four child nodes. The Filter algorithm hides the true location of the original dataset based the incomplete quadtree. The second part of the algorithm compresses the data, and then

adjust the noise under consistency between tree nodes. They employ BLUE (Best Linear Unbiased Estimator) algorithm to adjust the values of nodes for adjusting any inconsistencies existing between father and child nodes. They further experimentally analyze the algorithm by comparing it to the previously proposed algorithms for checking the accuracy of range queries on the published data.

A fairly new approach to achieve differential privacy on tabular data was introduced by [91], where they combine a technique to achieve *k*-anonymity privacy model with differential privacy, for enhancement of utility of published data. To reduce the Laplace noise that is added for achieving ϵ-differential privacy, noise must be added to the *k*-anonymous version of the dataset, which is achieved by micro-aggregation of all attributes. While we consider *D* to be the dataset input in the algorithm, D_ϵ represents the differentially-private version of dataset *D*. To improve the utility of D_ϵ, the algorithm comprises of two steps: First, *k*-anonymous data set $\bar{D}$ is generated from *D* by using micro-aggregation like MDAV[22] with the assumption that all attributes are QI attributes (quasi-identifier). Next, *k*-anonymous dataset $\bar{D}$ generates ϵ-differentially-private dataset D_ϵ based on the ϵ-differentially-private response to the posed queries. The idea behind possibility of improvement in utility by using *k*-anonymity is that unlike when differential privacy is directly applied for a number of individuals, this technique applies for groups. Also, the sensitivity is considerably low when this technique is applied since each record in published dataset $\bar{D}$ depends on at least *k* or more records in original dataset *D*.

The authors of [71] proposed distributed differentially-private anonymization algorithm *DistDiffGen* to publish vertically partitioned data, where two publishers possess different attributes of the same participants. The proposed approach is an extension of *DiffGen* algorithm proposed by [72], with an addition of distributed exponential mechanism for analyzing the candidate score pairs and generating the winner based on the definition of

exponential mechanism. This winner candidate is later used for specialization. Additionally, this approach uses addition of Gaussian noise for privacy protection against the other publisher and Laplace noise for achieving differential privacy.

In [6], the authors propose an approach to publish horizontally partitioned data, where a part of the data is held by two publishers. They present an algorithm for obtaining the winner candidate by applying exponential mechanism in a two-party scenario. While this algorithm can be used as a sub-algorithm in any algorithm, they apply it on their two-party algorithm similar to [71]. Finally, Laplace noise is added by each party for maintaining differential privacy while exchanging data.

The paper [14] introduces an algorithm based on probabilistic top-down specialization approach to obtain ϵ-differential privacy on set-valued data, by applying differentially-private sanitization algorithm *DiffPart*. *DiffPart* performs the first step of its top-down partitioning algorithm by performing generalization of all records iteratively till a single root partition is reached. Further, the sub-partitions are generated recursively until leaf partition is reached, based on the taxonomy tree representation and non-empty sub-partitions are further re-partitioned, for which either exponential or Laplace mechanisms can be employed. However, they claim that exponential mechanism leads to a smaller privacy budget allocation for each operation since it does not consider the composition property even under the circumstances when all sub-partitions contain disjoint datasets and this leads to less accurate results; Therefore, the proposed mechanism uses Laplace mechanism. Finally, for each leaf partition that is greater than a predetermined size, *DiffPart* adds the noisy number of records in that partition based on its noisy size, to publish the differentially-private data.

The authors of [114] propose an algorithm–*IncTDPart* for incrementally publishing a series of differentially-private datasets. The proposed technique employs Top-down partitioning/top-down specialization (TDS) by means of taxonomy tree and update-bounded

sanitization mechanism. The approach is similar to the TDS technique and employs *Diff-Part* algorithm proposed by [14] for publishing static set-valued data, differing in the fact that there exists an incremental release mechanism that has prior knowledge about the maximum number of updates that will be permitted, making the mechanism U-bounded. Such a scenario where data is published incrementally is further discussed in Section 3.2.

The authors of [20] discuss the problems of big data research in terms of analysis, archiving and reusing the data and generation of results. The problems discussed include, the fact that big data analysis needs to be performed on cloud therefore unusual expertise is essential, also the large datasets pose an increased risk of revealing personally identifiable information. They then discuss solutions to contain these challenges for publishing and analysis of big data by application of differential privacy.

The paper [96] discusses the issues with publication of two-dimensional datasets using differential privacy by use of methods like construction of a hierarchy of partitions which cannot be implemented for high dimensional datasets or by using a one or two level equi-width grid over data domain which is not suitable for skewed datasets. They also propose a technique as their solution to the discussed problems which uses private h-tree which makes use of a two level tree and a data dependent method. h-tree requires less budget for node counts since its height is deliberately kept low, which leads to more budget being assigned for median partitioning. The paper proposes a recursive budget strategy for minimizing the noise added, by reducing the number of median splits from linear to logarithmic, since the splitting points are selected in a differentially-private manner. The experimental evaluation of this approach on real-world and synthetic datasets demonstrated that the proposed approach is better than existing approaches, especially in the case of skewed datasets having outliers.

The authors of [107] introduce *DPCube* which is the component in Health Informa-

tion DE-identification (HIDE) framework, and makes use of differentially-private access mechanisms and two-phase multidimensional partitioning strategy for publishing multidimensional data cubes or multidimensional histograms for sensitive data. HIDE framework is used for integrating heterogeneous structured and unstructured health information and includes techniques for PPDP. The multidimensional data cubes and histograms published by DPCube achieve good utility in a differentially-private manner. The paper demonstrates that the data cubes published using DPCube is differentially-private version of the raw dataset, and the published data can be used for On-line Analytical Processing (OLAP) queries and learning mechanisms.

The paper [73] discusses that the existing solutions for publishing relational and set-valued data and propose an algorithm to publish it for health-care data in a differentially-private manner. The proposed method differs from the existing methods by adding noise after generalizing the records, instead of generating a contingency table for addition of noise. The flow that the proposed technique follows is that raw data is generalized first and next, optimal noise is added to guarantee differential privacy. Additionally, they build a decision tree classifier from the differentially-private published dataset to demonstrate the utility of the published data. The experimental evaluation of the technique showed that it is scalable and has efficient performance, although the utility might be affected when the domain size of the output is very large.

3.1.2 *via* Other Privacy Models

Prior to introduction of differential privacy, data publishing research primarily encompassed a number of other privacy-preserving models including k-anonymity [87][88], l-diversity [64] and t-closeness [61]. Each of these privacy models provide one or more

privacy guarantees for the published data as detailed in [34] where each mentioned technique is explored in detail and shows each of their respective limitations.

Publishing data with *k*-anonymous privacy guarantee, the record of an individual is hidden by grouping records that have the same Quasi-Identifiers, with *k* number of records. There has been a lot of research on *k*-anonymity and can be explored in [36][35][45][77][93]. Since *k*-anonymity assumes that each record represents a distinct individual, it provides little privacy to a group of *k* records being owned by fewer than *k* owners. To overcome this issue in *k*-anonymity, (*X,Y*)-anonymity [98] was introduced. *k*-anonymity based privacy models rely on formation of group, but if the records in the assigned group consist of sensitive attributes having similar values, the adversary could perform attribute linkage attack *i.e.* infer an individual's sensitive value based on the values received from the entire group and singling out the individual thereby eliminating privacy guarantees. To avoid this, further research was done which gave rise to privacy-preserving models that could potentially defend against attribute linkage.

One of the most noted of these contributions was *l*-diversity that guarantees every quasi-identifier group will have at lease *l* sensitive attributes. In [64][54][57][102], techniques were proposed to achieve *l*-diversity and recursive *(c,l)*-diversity (an improvement over *l*-diversity). *l*-diversity, while a significant improvement over *k*-anonymity, has its own drawbacks–each sensitive attribute taking values uniformly being one of them. To avoid this, other privacy models were proposed to prevent attribute linkage that could be achieved even in *l*-diversity. These include confidence bounding [100][99], *(X,Y)*-privacy [98], *(a,k)*-anonymity [105], *(k,e)*-anonymity [112] and *t*-closeness. One of the most noteworthy of them being *t*-closeness, implemented by [61][92][10][84][60] which provides great privacy guarantee in the published data. In addition to these models, other set of privacy models were being researched on that would provide privacy guarantees in cases where

previous models would fail. One such case would be when an attacker might not know an individual's record in dataset, but might confidently be able to infer the presence or absence of an individual's record in published data. To overcome this table linkage, techniques that implemented privacy guarantees: δ-presence [78][76], *(d,*γ*)*-privacy [83], distributional privacy [8] and ϵ-differential privacy [26][24][27] were employed. Additionally, to reduce an attacker's probabilistic belief about an individual changes once they have received access to the published data, a number of other privacy models were implemented. [12] introduced *(c,t)*-isolation privacy model, *(d,*γ*)*-privacy by [83], distributional privacy introduced by [8] and ϵ-differential privacy by [26][24][27] were a few among said models.

[34] then discusses publishing complex data in multiple scenarios and the ongoing research in each field. The primary research for these complex publishing scenarios can be found in–multiple release [109][55][7], sequential release [98][88][93], continuous release [9][106][33] and collaborative release [101][50][51]. [34] then explores the privacy guarantees in each scenario prior to introduction of differential privacy by Cynthia Dwork [26][27].

Other than these, trajectory and sequential data publication using k-anonymity can be broadly classified as follows: generalization [79][89][74], spacial translation [1] and suppression [17][95]. Research on publishing trajectory data using k-anonymity started when Nergiz *et al.* [79] proposed a generalization based approach combined with random reconstruction of the original dataset from anonymization. Spacial translation was applied by Abul *et al.*. [1] to publish trajectory data with (k,δ)-anonymity guarantee, which suggests that since trajectories are uncertain and can be represented in 3D space, each of the trajectories may have $(k-1)$ other trajectories with δ nearness. Another approach was proposed by Chen *et al.* [17] where they incorporate local and global suppression to trajectory data. They conclude that local suppression which removes certain instances

from the dataset provides better utility as compared to global suppression that removes all the occurrences of the item from the dataset. Other research [37] focuses on publishing movement data by ensuring LK-privacy. They ensure utility of published data using probabilistic flow-graph for anonymization.

The distinction between the approaches for publishing movement data are specified in Table 5.1.

Table 3.1: **Comparative evaluation of main features in related privacy-preserving data publishing approaches (properties in columns are positioned as beneficial with fulfillment denoted by • and non applicability by N/A)**

Research Work	**Movement Data**		**Privacy Model**		**Mechanism**		**Supported Queries**			
	Trajectory	**Sequential**	**Differential Privacy**	**Others**	**Laplace**	**Exponential**	**Count Query**	**Range Query**	**Frequent Sequential pattern Mining**	**Other**
Cao *et al.* [11]	•		•		•	•				•
Fan *et al.* [30]	•		•		•			•		
Hua *et al.* [46]	•		•		•	•		•		
Riboni *et al.* [85]	•		•		•					•
Chen *et al.*[16]		•	•		•		•		•	
Chen *et al.* [13]		•	•		•	•	•		•	
He *et al.* [43]		•	•		•		•			
Chen *et al.* [15]	•	•	•		•		•		•	
Nergiz *et al.* [79]	•			•	N/A	N/A				•
Sherkat *et al.* [89]	•			•	N/A	N/A	•			•
Abul *et al.* [1]	•			•	N/A	N/A		•		
Ghasemzadeh *et al.* [37]	•			•	N/A	N/A				•
Proposed Approach: Chapter 4	•		•		•	•	•	•	•	

3.2 Complex Data-Publishing

The discussion so far in all the previous sections has been on reporting data anonymization and publishing for a single release. While this is crucial, in real-world applications, data publishing and release are not facile to achieve. That is, the data could be made up sections that were anonymized separately; and every record of this data may therefore each have its own ϵ values and different values of noise added. Additionally, the real-world differentially-private data could be released multiple times with minor changes in the its records. When publishing such complex data under practical and real-world applications of data publishing, special attention is needed in the implementation phase based on the category of data. The following subsections consider such extended publishing scenarios for complex data publishing [34]. Table 3.2 summarizes the various specifics of these extended data release scenarios. The three publishing scenarios are discussed in the sub-sections below. In order to further elaborate, we discuss the techniques used to achieve those publishing scenarios.

3.2.1 Multiple Data Publishing

During the release of a dataset, said dataset might not be completely required by all the data miners. Multiple releases can be scheduled by dividing the complete dataset into a number of smaller dataset views based on what part of the dataset a particular data miner might be interested in. Such a release can be scheduled by taking the repercussions of partial data release into consideration. One of the major concerns with such an approach will be that the attacker could combine these views to obtain more specific data which might have had not been available previously. Since, in this case, the data publisher will be unable to foresee or prevent when an attacker obtains more than one view of the dataset. [34]

Definition 3.2.1. ***Multiple Data Publishing.*** *Let D be a dataset owned by data owner A^D. For each data release request R by a data miner U, A^D publishes a subset dataset D^R satisfying R.* ■

Fung *et. al.*[34] suggested that privacy is preserved in multiple data releases by anonymizing using *k*-anonymity and *l*-diversity approaches. In [26], [7] differential privacy is used for anonymizing in the case of Multiple data publishing. Barak *et al.* [7] summates multiple release issue in differential privacy and the threats the marginals (subsets of the original dataset) would potentially lead to. Their study provides a formal guarantee for preserving privacy, accuracy and consistency in the published marginals/subsets. They propose an approach for maintaining Differential Privacy is a 3 step process where the dataset is transformed into the Fourier domain. Differential privacy is then applied on the transformed data, to which linear programming is then applied, to finally obtain a result in a non-negative contingency table. The application of differential privacy on transformed data is obtained by perturbation while from the Fourier coefficient values, the resulting contingency table that consists of no negative values is obtained. The proposed technique affects neither the accuracy nor the consistency of the dataset during the entire process. The research in [28] further substantiates the use of contingency tables and marginals and compares it to histogram data, to determine the noise that will be added to prove the efficiency of the proposed mechanism.

3.2.2 Incremental Data Publishing

We comprise sequential and continuous data publishing as Incremental data publishing. In the sequential data-publishing scenario, the data publisher knows the datasets D_1 to D_{n-1} that were published previously and would like to publish D_n, where D_i is an updated version of D_{i-1}. While sequential anonymization presumes that the dataset does not update

dynamically and remains static throughout, the continuous data publishing on the other hand presumes that the data updates dynamically with time; this is the major difference between otherwise similar sequential and continuous anonymization. Another distinction between sequential and continuous data publishing is that in continuous data publishing all the data that is published belongs to the same database schema, while all the data that is released is a projection of the same database in the case of sequential data publishing [34]. Incremental update is similar in meaning to sequential data-publishing, and can be defined as any update to an existing dataset where each of these updates may differ either on a wide range or on a very minute scale.

Definition 3.2.2. ***Incremental Data Publishing*** *Given a dataset D that is continuously updated, the data owner A^D publishes D_1, D_2, ..., D_n such that each published dataset is an updated version of D.* ■

The paper [34] cites research for achieving sequential and continuous data-publishing by techniques like *k*-anonymity, *(X,Y)*-anonymity, *l*-diversity and *m*-invariance. These techniques, however, are not primarily aimed at differentially-private data. The techniques for incremental release are published in [86] and [113] with each technique relating to a different type of data like set-valued data[113], check-in data[86]. [86] presents the technique for providing differential privacy over check-in data to protect spatio-temporal data from the untrusted third parties who have access to the data, and from other users who might infer locations visited by other users, with the help of pre-filtering process. Their research is further extended for incremental release of this check-in data. [113] proposed an algorithm *IncTDPart* for publishing incrementally updating scenario on set-valued data. This algorithm generates a series of differentially-private releases like in [86]. The algorithm uses Top-down partitioning based on the generated item-free taxonomy tree.

Table 3.2: **Properties of various Complex Data Publishing Scenarios**

	Data Publishers		**Status of the data**			**Published Data**				**Data Recipients**	
	Single	Multiple	Static	Dynamic		Type		Count			
				Update	Add/Delete	Subset	Whole	Single	Multiple	Single	Multiple
Multiple Data Publishing	•		•			•		•			•
Incremental Data Publishing	•			•	•		•		•	•	•
Collaborative Data Publishing		•		•	•	•	•	•	•	•	•

3.2.3 Collaborative Data Publishing

The publishing scenarios discussed above considered the data to originate from a single publisher. Although collaborative data publishing realizes that in real-world scenarios, there might be cases where multiple publishers are present. In such scenarios, the release mechanism fails to identify how these multiple publishers interact with each other and the system on the whole. The interactions can be classified mainly into: multiple data publishing organizations that share data or subsets of data with other organizations in exchange for their data, multiple publishers that release the data to a Third-party organization and multiple individuals who individually publish their own record to an organization that collects data from such varied sources.

Definition 3.2.3. ***Collaborative Data Publishing*** *Given datasets D_1, D_2, ..., D_n owned by, the data owners ${A_1}^{D_1}$, ${A_2}^{D_2}$, ..., ${A_n}^{D_n}$, such that the data owners collaboratively publish their datasets.* ■

This is discussed further in [90][81][66], where the techniques for collaborative release are portrayed. The paper [90] considers untrusted aggregator to learn over multiple participant's data in a differentially-private manner. The proposed technique that lets a group of participants to upload a sequence of encrypted data values to the aggregator. Which in turn, permits the aggregator to summate all the values uploaded in a time period, but does not provide the aggregator permission to be able to learn anything else from the data

values. Although this might not directly be a data publishing scenario, [90] demonstrates how a collaborative scenario works. [81] presents a technique that can be used by a third party organization as per the scenarios listed above. They propose a combination of differential privacy and secret sharing in the same system for protecting the privacy of data publishers with the privacy of individuals whose records might be in the data. Differing in the number of publishers and data miners, [66] presented a technique for secure data exchange between two parties where both parties act as a data publishers and the joint data is made differentially-private while preserving the privacy of both datasets.

3.3 Privacy Preserving Data Mining

Privacy-preserving data mining (PPDM) is another related field for data sharing which focuses on mining information from a shared data. While mining, the privacy of the entities involved needs to be maintained and not be disclosed to the data miner who may also be an adversary. The goal of PPDM is to preserve the privacy while mining from the dataset. There has been vast literature [3][4][5][48] for data mining while preserving the privacy of data, which has been discussed in detail in the survey [2][62] by Agarwal *et al.* and [97] by Verykios *et al.*.

Distributed privacy-preserving data mining (DPPDM) is a decentralized version of PPDM where multiple parties are involved in the process of data mining on distributed data. Over the past years, there has been vast research [19][62] on distributed data mining. We refer the reader to [2] for in depth discussion about the literature on PPDM and DPPDM.

Chapter 4

PROPOSED ALGORITHM

We propose a solution for publishing trajectory data in a differentially-private manner. We apply partitioning to generalize the timestamp occurrences in the original doublets in order to generate differentially-private doublets and therefore differentially-private trajectories. Partitioning is applied over timestamps that occurred at the location.

Distinct timestamps that exist for each location in the trajectory data is initially represented as a cluster. Next, each of these clusters containing timestamps are subdivided into $2^{len(clus)-1}$ partition-cases to further generalize the existing timestamps based on the obtained partitioning results. The score of each of these partitioning-cases are generated based on the utility function we introduce. Based on the utility score, each partitioning-case is sampled using exponential mechanism, and the newly generated timestamp becomes the representative of the previously existing timestamps in the partitioning-case. Next, for all locations, we generate new trajectories by replacing the actual timestamps in the original data. To achieve differential privacy before publishing the data, Laplace noise is added to the count of newly generated trajectories.

In Figure 4.2.(a), we portray an example in which a location cluster consisting of 3 unique timestamps, where timestamps 1 and 7 are repeated 2 times each. We demonstrate all the possible partitioning-cases (2^{n-1}) for this cluster. In the figure, Internal (<- ->) and External ($\leftrightarrow$) number of Gaps and number of occurrences of Gaps are represented.

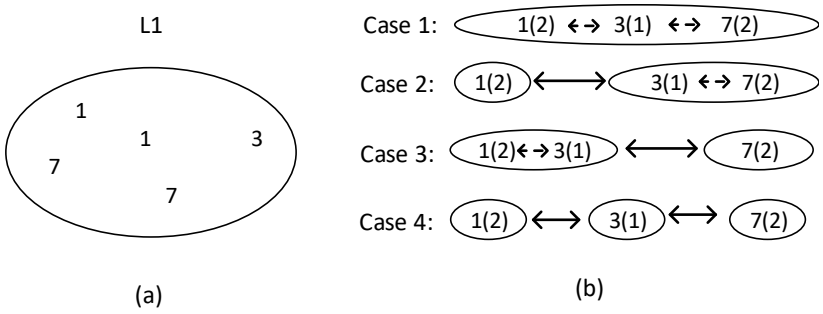


Figure 4.1: (a) original cluster for 3 unique timestamps with different counts and (b) possible partitioning-cases ($2^{n-1} = 4$) for this cluster, and Internal (<- ->) and External ($\leftrightarrow$) number of Gaps and number of occurrences of Gaps

Our initial approach for partitioning the timestamps was highly dependent on the number of timestamps that exist in the cluster. This led to an infeasible number of partitioning cases.

Example 4.0.1. If we have a cluster of 20 timestamps at a given location, our approach will partition these timestamps and generate 2^{20-1} partitioning cases. And if we have 50 different locations each having 20 timestamps, we need to generate $50 * 2^{19} = 26214400$ partitioning cases over which we need to calculate utility and input into exponential mechanism. ■

To overcome this problem, we introduce a differentially-private sub-algorithm which incorporates differential privacy on existing timestamps at each location in order to choose the best θ pivot ranges, where θ is the number of pivots and is equal to $\frac{size\_of\_cluster}{\alpha}$ (α varies from 2 to 10). We begin by generating disjoint ranges for these timestamps, such that each range is represented by the count of timestamps within that range. By applying exponential mechanism, we choose the top θ pivot points based on the counts of these ranges, where a higher count represents better utility. Once these pivot points are determined, we apply a greedy approach to generate disjoint partitions including all timestamps before the pivot.

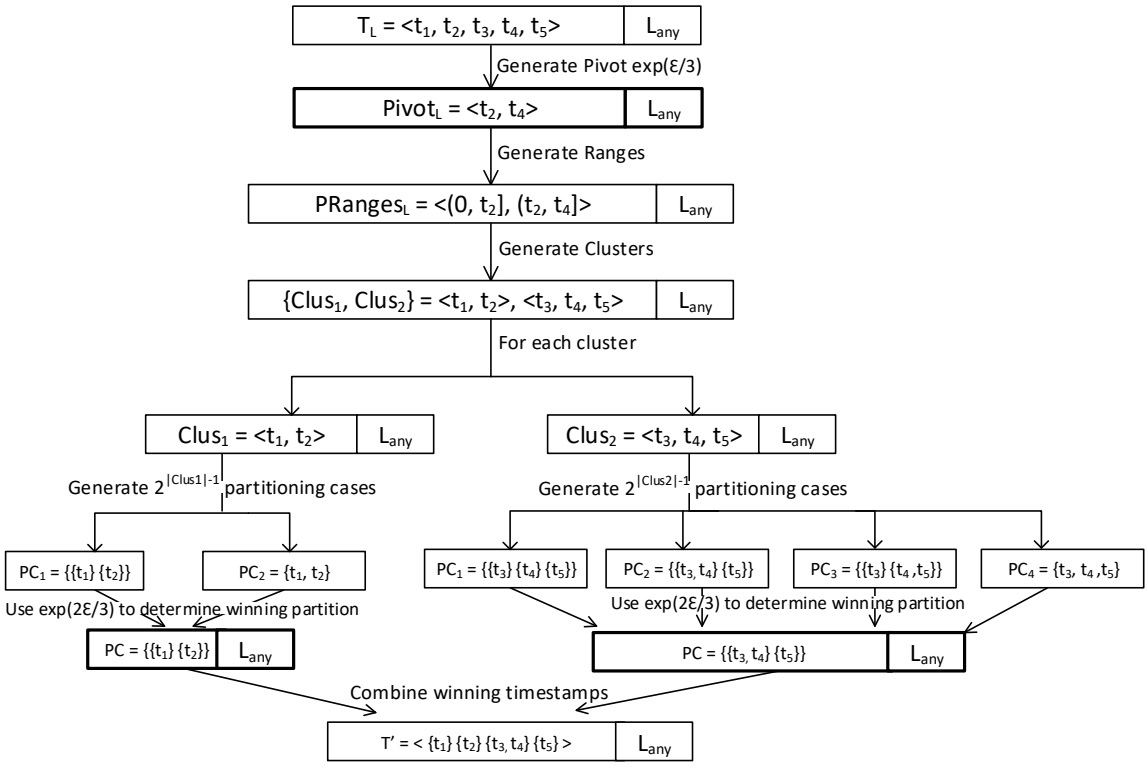


Figure 4.2: The process of generating pivot timestamps and partitioning

For the last pivot, all timestamps after will be included. Our algorithm is then incorporated on these partition cases for each pivot individually, thus drastically reducing the number of partitioning cases generated.

Example 4.0.2. Given a cluster of 20 timestamps, at a certain location our approach will generate 10 pivot elements, which if distributed evenly in the cluster would mean that approximately $2^{2-1} * 10 = 20$ partition cases are generated. If we have 50 locations, each of which having 20 timestamps, and they all have pivot elements distributed evenly through the cluster, the we need to generate $20 * 50 = 1000$ partitioning cases throughout. ■

This proves that our sub algorithm makes the partitioning process more feasible while

maintaining the differential privacy guarantee.

4.1 Solution Overview

The flow of our proposed approach is depicted in the pipeline diagram in Figure 4.3. The first step is Data Collection (not shown in pipeline diagram), during which the data owners contribute their data to build the raw trajectory dataset. The privacy-preserving data publishing process begins by preprocessing (cleaning, proper formatting) the raw trajectory data. Next, exponential mechanism is applied in Phase 1, which includes generation of pivot timestamps and partitioning. Next, in Phase 2, Laplace mechanism is used to add noisy values to the real count of each trajectory in the raw data. From the differentially-private anonymizer, the differentially-private trajectory data is published. The trajectory data published using our approach is suitable for count queries, range queries, and mining frequent sequential patterns.

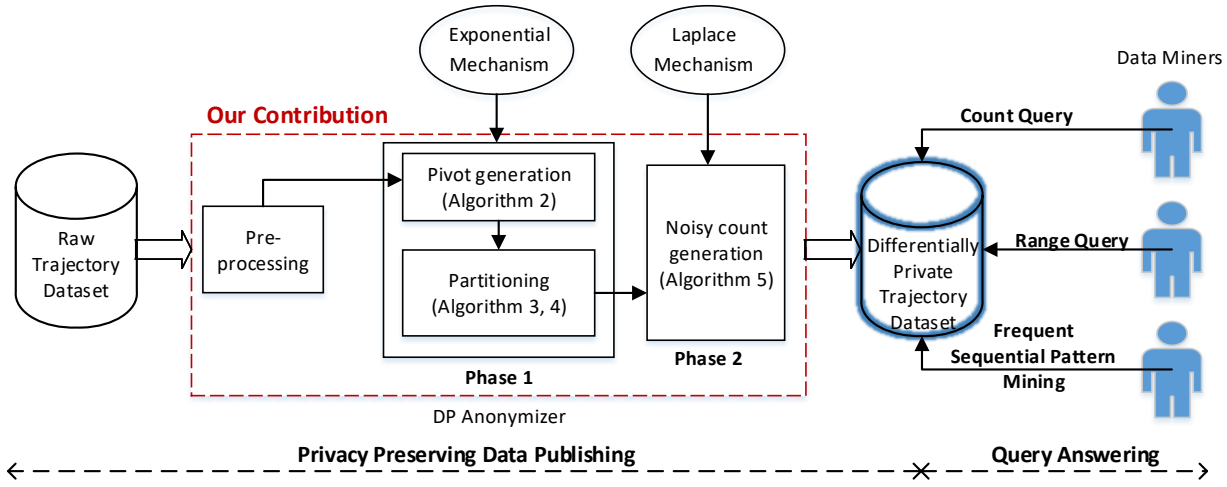


Figure 4.3: Pipeline diagram of our proposed approach

Next, we provide an overview of the proposed algorithms. The main algorithm 1 begins by generating clusters of timestamps at each unique location, and then executes algorithm 2 to determine the winning partitioning case out of numerous possibilities. A representative value is then sampled from each partition within winning partitioning cases at each location. Next, after regenerating the trajectories, we add Laplace noise to the real counts of these trajectories before publishing the differentially-private trajectory data. The main algorithm calls upon the cluster generation algorithm 2 in step 2 for each cluster T_{l_j} at location l_j using half of the privacy budget. The cluster generation algorithm uses exponential mechanism for generating the initial partitions of the cluster and further generates a set of ordered clusters, based on the pivot ranges generated using threshold θ. This algorithm calls algorithm 3 for determining the winning partition for each cluster of timestamps, to return the winning partitioning case to algorithm 1.

When algorithm 3 is called by the cluster generation algorithm, each cluster $clus_i$ in CL is input with the privacy budget $2\epsilon/3$. The algorithm 3 generates all possible partitioning cases for the cluster $clus_i$ and computes utility score of each partitioning case using the utility metrics we propose in algorithm 4. Based on these utility scores, exponential mechanism is then used to determine the winning partitioning case pc_w, which is returned to algorithm 4.

Algorithm 4 is called by algorithm 3, where the partitioning case pc is the input, and the algorithm returns the computed utility for pc based on the properties of the timestamps and partitions in the partition case. These properties include: (i) the gaps between timestamps which is computed by the difference between adjacent timestamps, (ii) the CountGaps between timestamps which is the difference between the number of occurrences of adjacent timestamps, (iii) the gaps between partitions which is the difference between the last and first timestamp of two adjacent partitions respectively. The algorithm computes

the External Homogeneity which is the homogeneity between partitions and the Internal Homogeneity, which is the homogeneity between timestamps within partitions for all partitions using Standard Deviation. Finally, the algorithm computes the utility score U_{pc} which is returned to algorithm 3.

For adding Laplace noise, the main Algorithm 1 calls Algorithm 5 over the generalized dataset $\mathbb{D}'$ using privacy budget ϵ. This will add noise to the number of occurrences of the generalized trajectories $\mathcal{T}'$ in $\mathbb{D}'$. Then Algorithm 5 adds $Lap(\epsilon)$ to the real count of each trajectory $\mathcal{T}' \in \mathbb{D}'$. The differentially-private data $\hat{\mathbb{D}}$ must be then generated as per the noisy counts of the trajectories, and $\hat{\mathbb{D}}$ is returned to the main algorithm 1, which publishes the differentially-private data $\hat{\mathbb{D}}$.

4.2 Algorithms

In this section, we discuss our algorithms in detail for publishing trajectory data while guaranteeing differential privacy. Algorithm 1 is the main algorithm that inputs the raw trajectory dataset $\mathbb{D}$ and returns differentially-private dataset $\hat{\mathbb{D}}$. This algorithm calls upon other algorithms 2 and 5 for cluster generation and addition of Laplace noise to the real count of trajectories, respectively. The algorithms 3 and 4 are called by algorithm 2 for determining the winning partitioning case and computing the utility score for determining the winning partitioning cases, respectively.

The main algorithm 1 comprises of all major steps for publishing differentially-private trajectory data. This algorithm calls upon other algorithms 2 and 5 at different steps to accomplish their necessary functions. It begins with the generation of all clusters T_l for each location $l \in L$ currently present in the raw data $\mathbb{D}$. Within these clusters, there might be one or more occurrences of a timestamp $t \in T_l$. The second step in this algorithm

Algorithm 1 Main algorithm to generate differentially-private trajectories

Differentially-Private Trajectory Data Generation

Input: Trajectory dataset $\mathbb{D} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$, privacy budget B
Output: Differentially-private trajectory dataset $\hat{\mathbb{D}}$

1. Given raw dataset $\mathbb{D}$, determine the *set* of unique locations: $L = \{l_1, \ldots, l_m\}$, and for each location $l \in L$, determine the cluster T_l of all timestamps corresponding to that location. Note that any timestamp $t \in T_l$ might have multiple occurrences.
2. For each location $l_j \in L$:
   (a) Execute algorithm 2 on T_{l_j} using privacy budget $B/2$ to determine the best partitioning case of T_{l_j} in a differentially-private manner. The result is a set of timestamp partitions (intervals) $P_{l_j} = \{P_{l_j,1}, \ldots, P_{l_j,|P_{l_j}|}\}$.
   (b) From each partition $P_{l_j,i} \in P_{l_j} : 1 \leq i \leq |P_{l_j}|$, we uniformly sample a value v_i to be the representative timestamp of $P_{l_j,i}$.
3. Generate $\mathbb{D}'$ from $\mathbb{D}$ such that for each trajectory $\mathcal{T} \in \mathbb{D}$ there is a corresponding trajectory $\mathcal{T}' \in \mathbb{D}'$, and each timestamp t in every doublet in $\mathcal{T}$ is generalized. That is, for each doublet $d(l,t) \in \mathcal{T}$, add doublet $d'(l,v)$ to $\mathcal{T}'$, where v is the representative timestamp of t at location l.
4. Execute algorithm 5 over generalized dataset $\mathbb{D}'$ using the remaining privacy budget $B/2$ in order to add Laplace noise to real counts of trajectories and therefore generate differentially-private trajectory data $\hat{\mathbb{D}}$.
5. Return differentially-private trajectory dataset $\hat{\mathbb{D}}$.

calls Algorithm 2 for each location $l_j \in L$ on the cluster T_{l_j} using half of the assigned privacy budget B, in order to determine the best partitioning case for T_{l_j} while guaranteeing differential privacy. This returns a set of timestamp partitions P_{l_j}. For each location $l_j \in L$, a set of representative timestamps v_i are uniformly sampled from each timestamp partition $P_{l_j,i} \in P_{l_j}$. The next step of this algorithm generates generalized trajectories

$\mathcal{T}' \in \mathbb{D}'$ for each trajectory $\mathcal{T} \in \mathbb{D}$, such that, for each doublet $d(l,t) \in \mathcal{T}$, a generalized doublet $d'(l,v) \in \mathcal{T}'$ is added where v is the representative timestamp of t at the location l.

The fourth step in algorithm 1 calls upon algorithm 5 over the generalized dataset $\mathbb{D}'$ using the remaining half of the privacy budget B for addition of Laplace noise to the real count of each trajectory $\mathcal{T}' \in \mathbb{D}'$ to generate differentially-private trajectory data $\hat{\mathbb{D}}$. This differentially-private data $\hat{\mathbb{D}}$ can then be published for data mining and analysis.

Algorithm 2 Choosing differentially-private partitions by exponential mechanism

Partition Generation
Input: Cluster of timestamps $T = \{t_1, \ldots, t_{|T|}\}$, privacy budget ϵ
Output: Differentially-private partitioning multiset P

1. Compute the score of each timestamp $t_i \in T$ by applying exponential mechanism on $(t_i, Occ(t_i))$ using $\epsilon/3$ of the privacy budget, where $t_i \in T$ and $Occ(t_i)$ is the occurrence of t_i in T:

$$EM(t_i, Occ_{t_i}) = \frac{exp_{\frac{\epsilon/3}{2\Delta Occ}}(Occ(t_i))}{\sum_{t_j \in T} exp_{\frac{\epsilon/3}{2\Delta Occ}}(Occ(t_j))} \tag{4.1}$$

2. Choose top θ timestamps from previous step as the pivot timestamps: $PT = \langle pt_1, \ldots, pt_\theta \rangle$, where $\theta = |T|/\alpha$, and $pt_i < pt_{i+1} : 1 \leq i < \theta$ and α can range from 2 to 10.

3. Generate a set of ordered clusters $CL = \langle clus_1, \ldots, clus_\theta \rangle$ over PT, where each cluster $clus_i \subseteq CL$ represents the range $(pt_{i-1}, pt_i]$ and $pt_0 = 0$.

4. For each cluster $clus_i \in CL$:

   (a) Assign each timestamp $t_j \in T$ to $clus_i$ if t_j is within the range $(pt_{i-1}, pt_i]$.

   (b) Using the remaining privacy budget $2\epsilon/3$, partition $clus_i$ in a differentially-private manner using algorithm 3, and add the winning partitioning case to P.

5. Return P.

The main algorithm calls upon the partition generation algorithm in step 2, for each cluster of timestamps T at location $l_j \in L$ to result in a differentially-private partitioning multi-set P. The fist step computes the score of each timestamp $t_i \in T$ having an occurrence of Occ_{t_i} via exponential mechanism EM(t_i,Occ_{t_i}) using one-thirds of the alloted privacy budget (see Equation 4.2). This results in timestamps that resulted the best score (while maintaining differential privacy). The second step stores such timestamps as pivot timestamps $pt_x \in PT$, where x can range from 1 to θ. Here the value of θ is $|T|/\alpha$, but in the Chapter 5 we experiment on different values of α between 2 to 10. In the next step, the algorithm generates a set of ordered clusters $clus_x \in CL$ for each $pt_x \in PT$, such that that cluster $clus_i$ represents all timestamps that may exist in the range $(pt_{i-1}, pt_i]$ with an exception of pt_0 = 0.

Now for each cluster $clus_i \in CL$, each timestamp $t_j \in T$ is assigned to cluster $clus_i$ if the timestamp t_j is within the range of $(pt_{i-1}, pt_i]$. Algorithm 3 is then called upon each partition $clus_i \in CL$ using remaining two-thirds of the assigned privacy budget ϵ. This algorithm returns the winning partitioning case pc_w which is added to the differentially-private partitioning multi-set P. The algorithm returns P to main algorithm 1.

To determine the winning partition for each $clus_i \in CL$, Algorithm 3 is called upon by Algorithm 2 using the assigned privacy budget ϵ for this task. The first step generates all possible partitioning cases $PC = pc_1, \ldots, pc_m$ for the cluster $clus_i \in CL$, where $m = 2^{|clus_i|-1}$. The generation of these partitioning cases are discussed with an example in Figure 4. The second step calls upon Algorithm 4 for computing the utility score u_i for each partitioning case $pc_i \in PC$. Exponential mechanism EM(pc_i, u_i) is then applied, using the privacy budget ϵ for determining the partitioning case having high scores (see Equation 4.2). Such partitioning case sampled from the scores generated from the $EM(p_c, u_i)$ is the winning partitioning case pc_w. This winning partitioning case pc_w is

Algorithm 3 Choosing differentially-private partitions by exponential mechanism

Determining the winning partitioning case

Input: Cluster of timestamps $clus$, privacy budget ϵ
Output: Winning partitioning case pc^w

1. Generate all possible partitioning cases $PC = \{pc_1, \ldots, pc_m\}$ from cluster $clus$, where $m = 2^{|clus|-1}$.
2. For each partitioning case $pc_i \in PC$, compute the utility score u_i of pc_i using algorithm 4.
3. Using privacy budget ϵ, apply exponential mechanism on $(pc_i, u_i) : 1 \leq i \leq m$ to determine the winning partitioning case pc^w by sampling from the $EM(pc_i, u_i)$ scores:
$$EM(pc_i, u_i) = \frac{exp\frac{\epsilon}{2\Delta u}(u_i)}{\sum_{pc_j \in clus} exp\frac{\epsilon}{2\Delta u}(u_j)} \tag{4.2}$$
4. Return pc^w.

returned to Algorithm 2.

Algorithm 4 is used to compute the utility score U_pc for each partitioning case $pc = \{p_1, \ldots, p_k\}$. The concept of external gaps and internal gaps are described in Figure 4. The first step computes the External Gap Homogeneity (EGH) using Standard Deviation (SD) on external gaps, between adjacent partitions in pc using Equation 4.3. In the next step, compute External CountGap Homogeneity (EGH^{Occ}) using Standard Deviation (SD) on the difference between size of each partition $|p_i|$ using Equation 4.4. Using EGH and EGH^{Occ}, we compute the External Homogeneity (EH) as in Equation 4.5.

Next, the algorithm computes the Internal Gap Homogeneity (IGH) using Standard Deviation on the internal gaps between the distinct and adjacent timestamps for each partition $p_i \in pc$ using Equation 4.6. We next compute the Internal CountGap Homogeneity

Algorithm 4 Equations to compute utility of a partitioning case

Utility Score Computation
Input: Partitioning case: $pc = \{p_1, \ldots, p_k\}$
Output: Utility score U_{pc}

1. Compute External Homogeneity (EH) using External Gap Homogeneity (EGH) and External CountGap Homogeneity (EGH^{Occ}):

   (a) Calculate *External Gap Homogeneity (EGH)* using the Standard Deviation on external gaps between adjacent partitions in pc to determine the overall variance:

   $$EGH = \underset{1 \leq i \leq k-1}{SD}(p_{i+1}[t_{first}] - p_i[t_{last}]) \tag{4.3}$$

   where t_{first} and t_{last} are the first and last timestamps in a partition p_x.

   (b) Next, calculate the *External CountGap Homogeneity (EGH^{Occ})* using Standard Deviation on the difference between size of each partition $p_i \in pc$:

   $$EGH^{Occ} = \underset{1 \leq i \leq k-1}{SD}(|h_{i+1} - h_i|) \tag{4.4}$$

   where h_i is the number of timestamps in partition p_i.

   (c) Compute the *External Homogeneity* (EH) as:

   $$EH = EGH \times EGH^{Occ} \tag{4.5}$$

2. Compute Internal Homogeneity (EH) using Internal Gap Homogeneity (EGH) and Internal CountGap Homogeneity (EGH^{Occ}):

   (a) Calculate the *Internal Gap Homogeneity (IGH)* using the Standard Deviation on internal gaps between distinct adjacent timestamps for each partition $p_i \in pc$ to determine the amount of variation across it:

   $$IGH = \sum_{i=1}^{i=k} \underset{1 \leq j \leq h_i - 1}{SD}(|p_i[t_{j+1}] - p_i[t_j]|) \tag{4.6}$$

   where h_i is the number of timestamps in partition p_i.

   (b) Next, calculate the *Internal CountGap Homogeneity (IGH^{Occ})* using Standard Deviation on the difference between number of occurrences of distinct adjacent timestamps in each partition $p_i \in pc$:

   $$IGH^{Occ} = \sum_{i=1}^{i=k} \underset{1 \leq j \leq h_i - 1}{SD} |Occ(p_i[t_{j+1}]) - Occ(p_i[t_j])| : p_i[t_{j+1}] \neq p_i[t_j] \tag{4.7}$$

   where p'_i is the modified partition p_i which contains *distinct* timestamps from p_i and has length l_i, and $Occ(p'_i[t_j])$ is the number of occurrences of timestamp $p'_i[t_j]$ in p_i.

   (c) Compute the *Internal Homogeneity* (IH) as:

   $$IH = IGH \times IGH^{Occ} \tag{4.8}$$

3. Calculate the final *Utility Score* U_{pc} using the external and internal homogeneities EH and IH computed in Steps 1 and 2:

   $$U_{pc} = \frac{EH}{IH} \tag{4.9}$$

4. Return U_{pc}.

Algorithm 5 Applying Laplace noise to the original counts of generalized trajectory data

Addition of Laplace Noise

Input: Dataset $\mathbb{D}'$, privacy budget ϵ
Output: Differentially-private trajectory data $\hat{\mathbb{D}}$

1. For each trajectory $\mathcal{T}_i \in \mathbb{D}'$, add Laplace noise Lap($\epsilon$) to the real count of $\mathcal{T}_i$: $Occ_{\mathcal{T}_i}$, using the privacy budget ϵ.
2. Append each trajectory $\mathcal{T}_i$ based on its noisy count $Occ'_{\mathcal{T}_i}$ to dataset $\hat{\mathbb{D}}$.
3. Return differentially-private trajectory data $\hat{\mathbb{D}}$

(IGH^{Occ}) between the number of occurrences of distinct and adjacent timestamps for each partition $p_i \in pc$, using 4.7. Next the algorithm computes the Internal Homogeneity (IH) using Equation 4.8.

In the final step, this algorithm computes the Utility Score (U_{pc}) using the computed EH and IH in the previous steps. Algorithm 4 returns the computed U_{pc} to algorithm 3 for each partitioning case pc.

The main algorithm 1 calls upon the algorithm 5 over each trajectory T' in the generalized dataset $\mathbb{D}'$, using the remaining half of the total privacy budget $\epsilon = B/2$. For each trajectory $T_i \in \mathbb{D}'$, Laplace noise Lap(ϵ) is added to its real count Occ_{T_i} to generate noisy count Occ'_{T_i}. After the noise is added to each trajectory in the generalized dataset $\mathbb{D}'$ to generate differentially-private dataset $\hat{\mathbb{D}}'$.

4.3 Complexity Analysis

We denote by n the number of trajectories and by d the number of doublets in a dataset. We can determine the time complexity of the proposed approach in terms of these two notations. Our proposed approach broadly comprises of two main phases: (Phase 1) generating pivot timestamps and partitioning via exponential mechanism and (Phase 2) addition of Laplace noise to counts.

The computation for Phase 1 without generating pivot timestamps originally had a time complexity of $\mathcal{O}(2^{(t-1)})$ where *t* is the number of timestamps originally in the cluster. This led to the introduction of generation of pivot timestamps which reduced the time complexity of by subdividing the timestamps prior to partitioning. We compute the number of pivot timestamps $\theta = \frac{|T|}{\alpha}$, where $|T|$ is the number of timestamps originally in the cluster. The generation of this pivots led to the generation of sub-clusters each of which is of size $\simeq \alpha$. This reduces the time needed for generation of partitions to $\mathcal{O}(\theta * 2^{(\alpha-1)})$ where θ is the number of pivot timestamps and α is the size of each pivot. Therefore, for all locations in the dataset, the time complexity for Phase 1 is $\mathcal{O}(|L| * \theta * 2^{(\alpha-1)})$, where $|L|$ represents the number of distinct locations in the dataset. This can be simplified by the knowledge that the value of α is between 2 to 10, thus $2^{(\alpha-1)}$ is bounded by a constant. Also, we know that $|L| * \theta \leq d$. Therefore the total time complexity for Phase 1 is $\mathcal{O}(d)$.

The next step before Phase 2 is the regeneration of generalized trajectories which has the time complexity of $\mathcal{O}(d)$.

Phase 2 comprises of the addition of Laplace noise to the real counts of each raw trajectory in the generalized data. This phase therefore has a total time complexity of $\mathcal{O}(n * k)$, where $n = |\mathcal{T}|$ represents the number of trajectories and k is the time needed for adding noise to one trajectory (k is constant).

Thus the total time complexity for our proposed approach is $\mathcal{O}(n + d)$, where n represents the number of trajectories and d represents the number of doublets in the data.

Chapter 5

EXPERIMENTAL EVALUATION

In this chapter, we discuss the implementaion of our algorithms, including the datasets used and the experiemental evaluation over the generated differentially-private data. More specifically, we discuss scalability, efficiency and utility, as well as the complexity of the approach. The implementation for our solution has been done in Python 2.7 on a Linux machine with Intel Xeon(R) CPU E5-1620 v4 @ 3.50GHz Processor.

5.1 Datasets

We implement our approach on two datasets of taxi trajectories, whose features are listed in Table 5.1.

Table 5.1: **Properties of Datasets we performed experiments**

Dataset	# of doublets	# of locations	# of trajectories
goTrack	18,107	8,394	163
TDrive	10,158,088	97,822	8,890

The first dataset on which we evaluate the performance of our approach is *GPS Trajectories Dataset* [21]. This dataset is composed of two tables: *go_track_tracks.csv* and *go_track_trackspoints.csv* where *go_track_tracks.csv* has general attributes and each instance owns trajectory in the *go_track_trackspoints.csv* dataset. This dataset contains about 18,000 doublets. Another dataset that we use to evaluate the scalability and performance of our

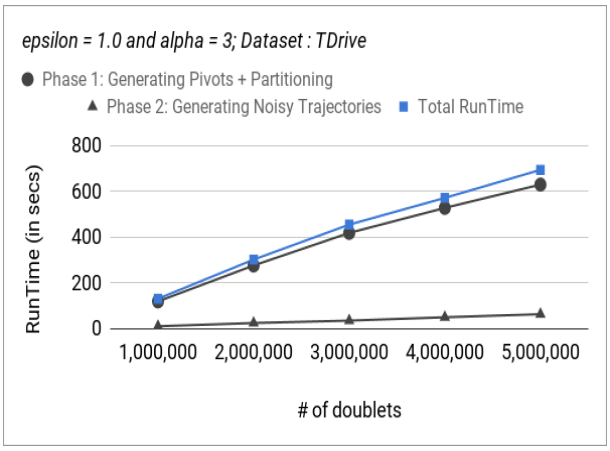
proposed approach is *TDrive* dataset [110] [111]. It contains more than 10 million doublets of (location, timestamp) for the trajectories of taxis in Beijing. The original format of this dataset is "$taxi\_id, datetime, longitude, latitude$". Through our experiments over both datasets, we consider each unique "$longitude, latitude$" pair as unique locations and "$datetime$" pair as timestamp. A pair of such ($location, timestamp$) is called a doublet.

5.2 Experimental Results

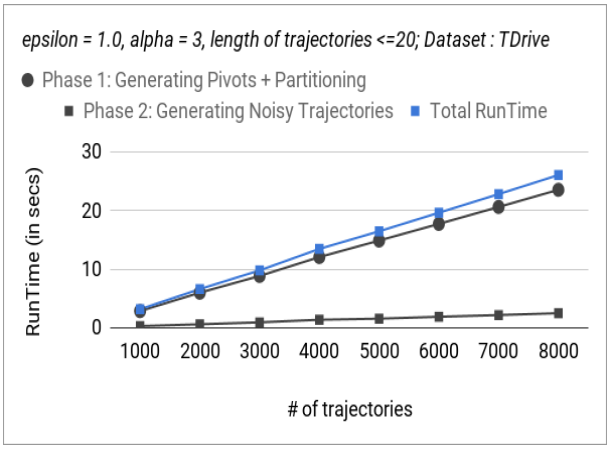
We evaluate the performance of our approach by implementing the proposed solution over the datasets for the following metrics : *scalability*, *efficiency* and *utility*.

5.2.1 Scalability

We set up our experiments to test the scalability of our approach on the *TDrive* dataset. We measure the runtime (RT) with respect to linear increase in # of doublets and # of trajectories, while setting the value of privacy budget ϵ to 1. We also set α, the parameter determining the number of pivots, to 3. Figure 5.1.a illustrates the when the $\#$ of doublets increases linearly from $1,000,000$ to $5,000,000$, while Figure 5.1.b illustrates the runtime when $\#$ of trajectories has a linear growth from $1,000$ to $8,000$, assuming that the maximum trajectory length is 20. We observe from Figures 5.1.a and 5.1.b that the total runtime of our algorithm grows linearly when the data size (# of doublets or # of trajectories) increases linearly. We also observe that Phase1 (generating pivots and partitioning) is the dominating phase compared to Phase2 (generating noisy trajectories); however, they both scales linearly w.r.t data size.



(a) Growth of data (number of doublets)

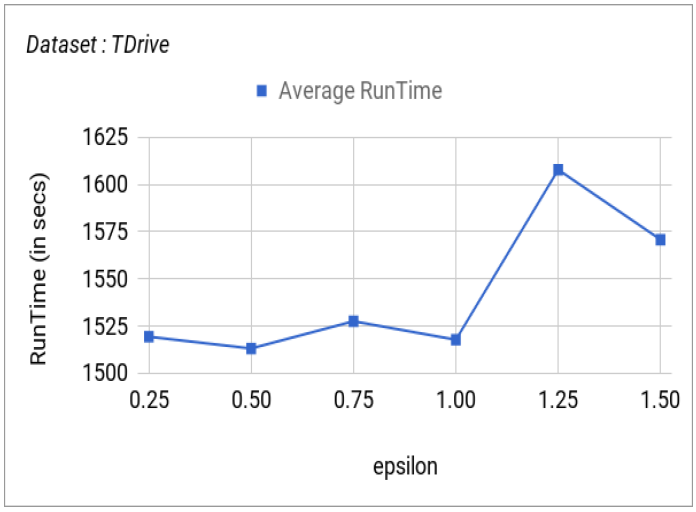


(b) Growth of data (number of trajectories)

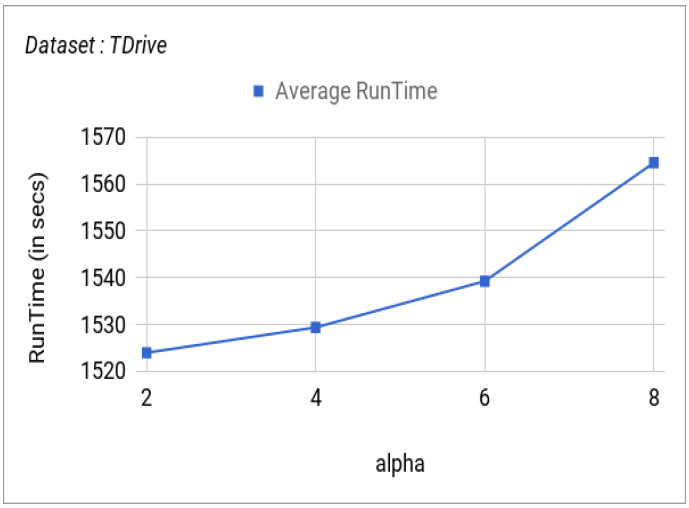
Figure 5.1: Scalability w.r.t. (a) linear growth of number of doublets, and (b) linear growth of number of trajectories.

5.2.2 Efficiency

We evaluate the efficiency of our proposed approach on the TDrive dataset with respect to privacy budget-ϵ and number of pivot timestamps α. Figure 5.2.a illustrates the average runtime obtained over 10 cycles when the value of ϵ is increased between 0.25 to 1.5 at an



(a) Privacy budget ϵ



(b) Pivot timestamps α

Figure 5.2: Efficiency w.r.t. (a) privacy budget ϵ, and (b) size of pivot timestamps α, averaged over 10 cycles.

interval of 0.25. We observe that the runtime remains consistent (around 1525 sec) until ϵ is equal to 1, and then starts increasing with a sudden spike to 1610 when the value of ϵ reaches 1.25. This anomaly is due to the fact that generated pivots are typically not evenly spaced out within the cluster, resulting in more than usual runtime needed for Phase-1, which includes partitioning. Given that the level of noise reduces when the the value of

ϵ grows, we conclude from Figure 5.2.a that the optimal value of ϵ to balance between efficiency and privacy is 1. Figure 5.2.b illustrates the the average runtime obtained over 10 cycles when the value of α increases from 2 to 8 with an interval value of 2. We observe that as α increases, the runtime increases accordingly (from 1,524 seconds for $\alpha = 2$ to 1,565 seconds for $\alpha = 8$). As a result, we conclude that the lower α is, the better our algorithm performs.

5.2.3 Utility

Utility is the usability of the output data compared to the original one. The higher the privacy of the published data is, the lower its utility. In this section, we measure the utility of the anonymized (differentially-private) published data with respect to *count queries*, *range queries*, and *frequent sequential pattern mining*. We measure the utility of the output data for count queries by counting the number of trajectories in which certain randomly-chosen locations exist in raw data $\mathbb{D}$ and output differentially-private data $\hat{\mathbb{D}}$. The error rate (query distortion) is then computed as follows [47]:

$$\textit{relative error / error rate} = \frac{|Q(\hat{\mathbb{D}}) - Q(\mathbb{D})|}{max(Q(\hat{\mathbb{D}}), Q(\mathbb{D}))} \tag{5.1}$$

where $Q(\mathbb{D})$ and $Q(\hat{\mathbb{D}})$ represent the output of count queries on raw data $\mathbb{D}$ and anonymous data $\hat{\mathbb{D}}$, respectively.

Count Queries

The count queries we set up for our experiments on the differentially-private dataset are queried over a predetermined number of locations. The number of locations on which the

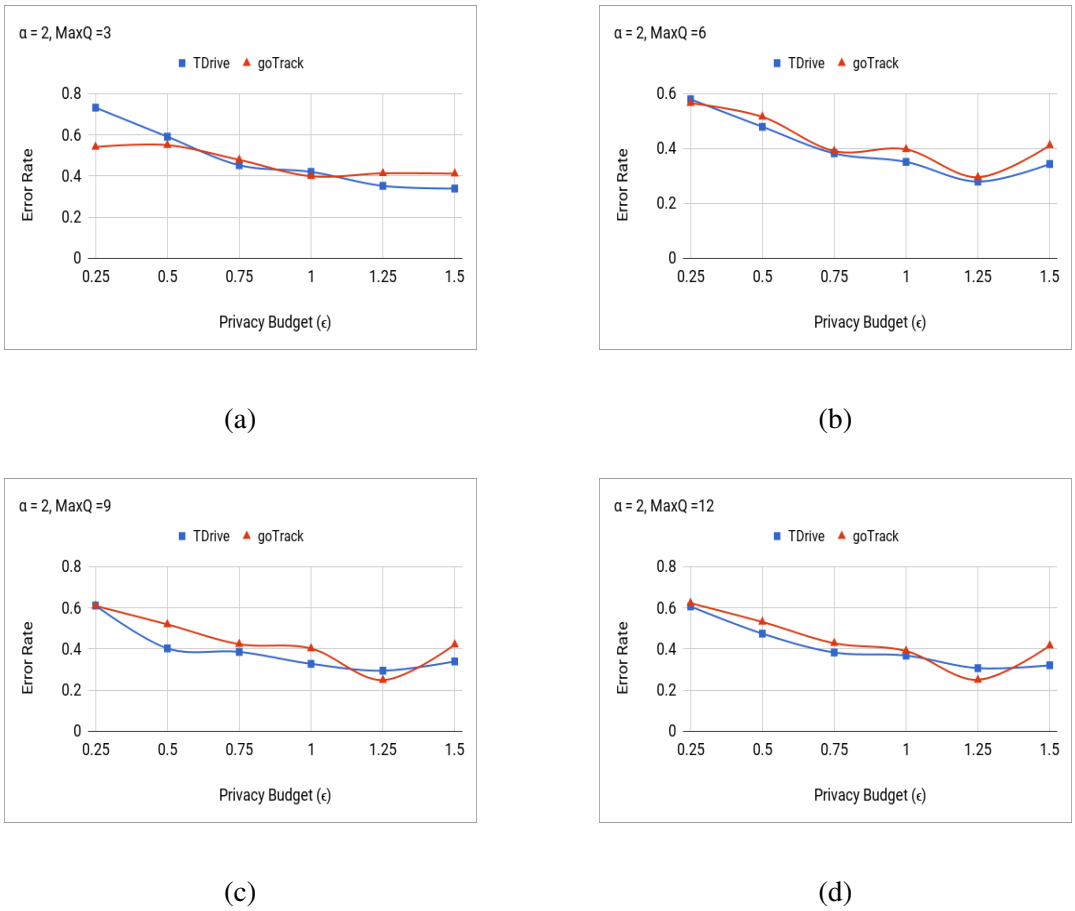


Figure 5.3: Error rate of anonymized data, where $\alpha = 2$, and *MaxQ* is set to 3, 6, 9 and 12.

count query Q_1 asks queries, is a random that ranges between the previous value of $MaxQ$ and current $MaxQ$.

Figure 5.3 illustrates the change in error rate as the value of the privacy budget ϵ grows from 0.25 to 1.5 at an interval 0.25, where the number of pivots α is set to 2 and *MaxQ* is set to 3, 6, 9 and 12. The experimental results are generated using both datasets: *goTrack* and *TDrive*. We observe that error rate ranges between 25% to 75%, but decreases with the increase of ϵ with respect to both datasets. We also observe that as the value of *MaxQ* increases, indicating an increase in the number of locations for which the dataset is queried,

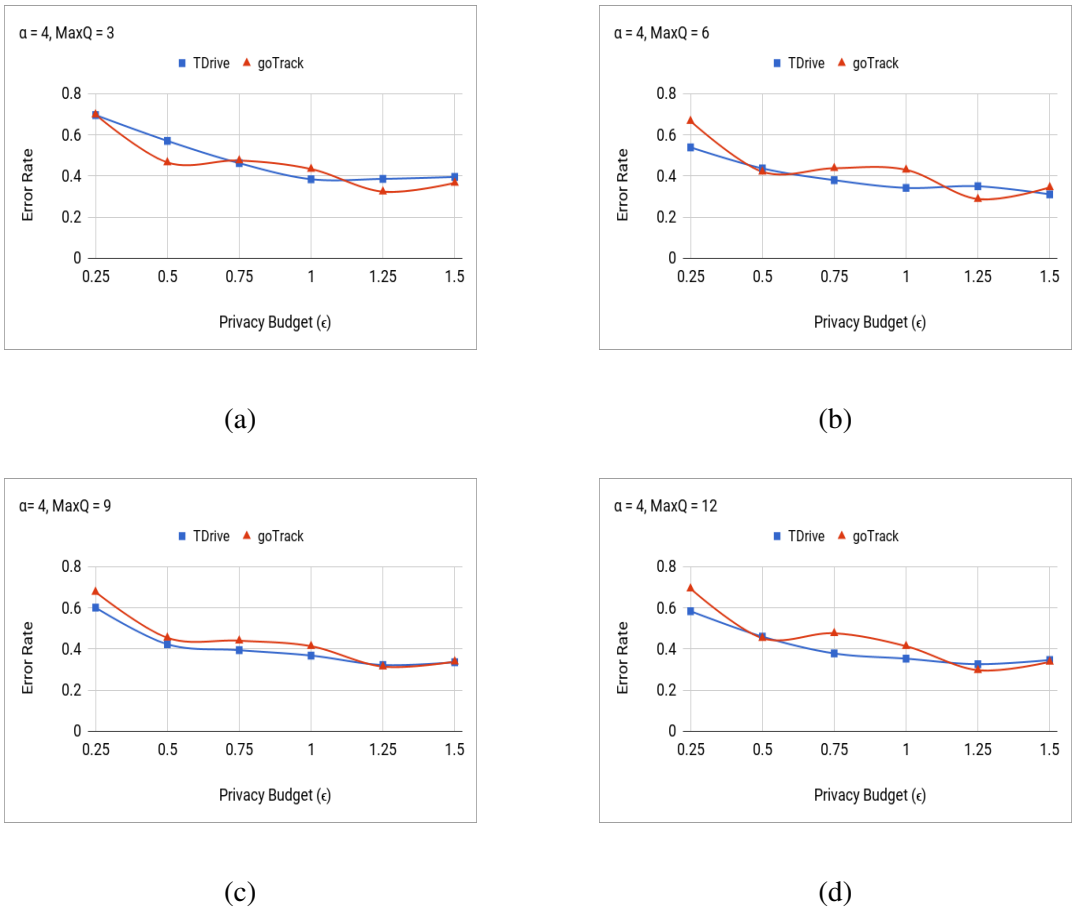


Figure 5.4: Error rate of anonymized data, where $\alpha = 4$, and *MaxQ* is set to 3, 6, 9 and 12.

the overall error rate goes down. The worst case error in figure-(a) when *MaxQ* = 3 is almost 75% whereas, the worst case error in figure-(b) when *MaxQ* = 2 is almost 60%. The highest utility (lowest error rate), i.e. 25% for goTrack dataset and 30% for TDrive dataset, is reached when $\epsilon = 1.25$ and *MaxQ* is 6, 9 or 12. Overall, We notice that the pattern of changes in error rate rate is consistent for both the datasets w.r.t. privacy budget ϵ.

Almost similar conclusions about error rate rate can be obtained from Figure 5.4, where privacy budget ϵ grows from 0.25 to 1.5 at an interval 0.25, the number of pivots α is set to 4 and *MaxQ* is set to 3, 6, 9 and 12. This indicates that unlike ϵ and *MaxQ*, changes in the

number of pivots α do not have a direct impact on the utility of the output data with respect to count queries.

Range Queries

Each range query includes a range of locations within the predetermined radius from the randomly selected location from the dataset. This query results in the count of trajectories having one or more of the locations from this range, based on the type of range query. We have queried using 2 types of range queries: *possibly sometime inside* (PSI) and *definitely always inside* (DAI) [46]. A PSI range query represents the count of trajectories that exist when a doublet from the range of (location, timestamp) doublets exists within the radius of a random location. On the other hand, a DAI Range Query represents the count of trajectories such that all the doublets in the trajectory exist from the range of doublets in the query.

We use Hausdroff distance [75] to compute the distance between doublets and determine the locations that are within the said radius of given location. We utilize the algorithm to compute Hausdroff distance from [38]. The result of a range query is the count of trajectories. We therefore query over the raw data $\mathbb{D}$ and the anonymized data $\hat{\mathbb{D}}$ to generate resulting $Q_2(\mathbb{D})$ and $Q_2(\hat{\mathbb{D}})$, respectively. These results are then used to compute the error rate according to Equation 5.1.

Figure 5.5 demonstrates the results obtained in terms of the change in error rate for TDrive dataset when the value of ϵ and α linearly grow, and the radius is set to 0.5. The range query used is PSI, which means there should be at least one doublet in the trajectory is the same as the specified range in the query, which means that the locations are within the Radius = 0.5. In the chart, we notice that the error rate from the range query reduces as the value of ϵ grows. Also, the error rate slightly flickers but generally reduces as the value

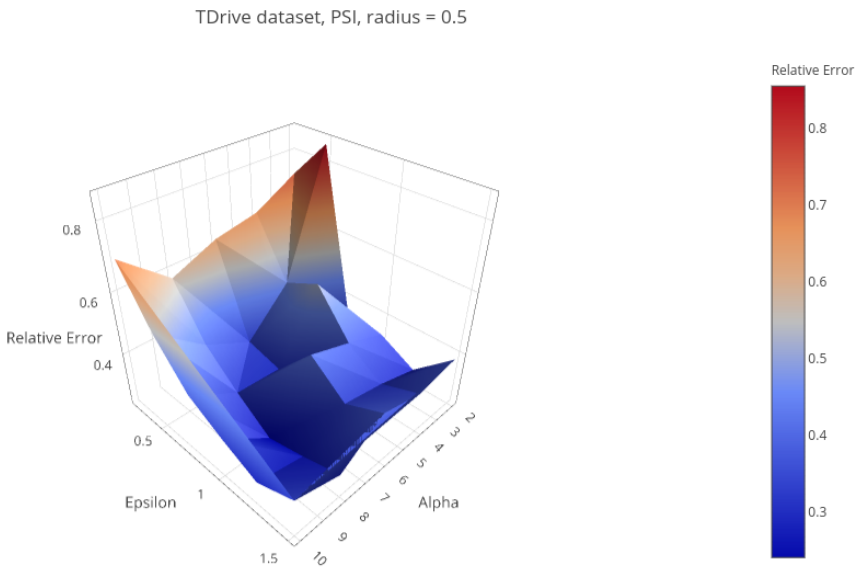


Figure 5.5: Error rate across different α and ϵ values for PSI range queries for TDrive Dataset where radius = 0.5.

of α increases. Since α is used for determining the number of pivot timestamps to reduce the complexity of partitioning, the conclusion that utility is not adversely affected by the change of the value of α is good. The worst case error occurs when both α and ϵ have the lowest values. The best case (about 30%) occurs when $\epsilon \geq 1$ with not much change in α.

Figure 5.6 demonstrates the result obtained in terms of change in error rate for TDrive dataset for growing values of α and ϵ while the radius here is 1.0 using PSI range queries. We observe that the error rate reduces when the value of ϵ grows. Also, the error rate reduces slightly as α increases with some flicker in the values.

Frequent Sequential Pattern Mining

Another set of experiments we have performed to test the utility of the differentially-private published dataset using our proposed approach is Frequent Sequential Pattern Mining. Since our approach is specifically for publishing trajectory data, determining the utility

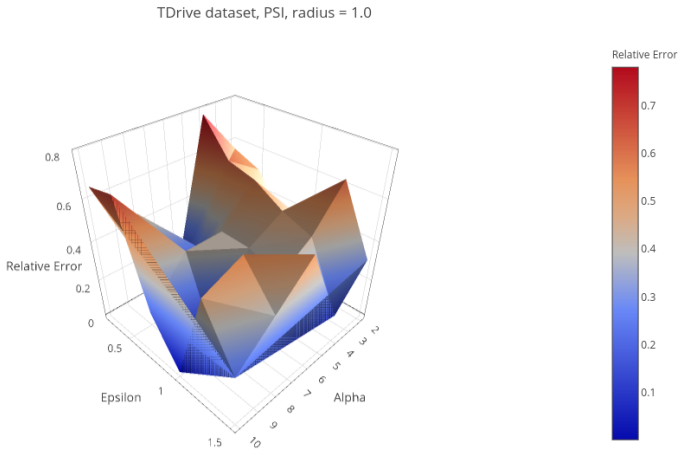


Figure 5.6: Error rate across different α and ϵ values for PSI range queries for *TDrive* Dataset where radius = 1.0.

of the differentially-private sequences by computing the error generated relative to the occurrences of those sequences in the raw dataset. The error rate is computed on the counts of frequent sequences that are beyond a threshold of $MinimumSupport$ on the raw dataset-$\mathbb{D}$ and differentially-private dataset-$\hat{\mathbb{D}}$. This results in the counts of frequently occurring sequences-$Q_3(\mathbb{D})$ and $Q_3(\hat{\mathbb{D}})$ for raw and differentially-private dataset. We then compute the error rate between $Q_3(\mathbb{D})$ and $Q_3(\hat{\mathbb{D}})$ using the formula-5.1.

In figure 5.7, we demonstrate the results obtained from mining the frequent sequential patterns on real and noisy goTrack trajectories dataset to generate error rate. For our experiments, we maintain a constant value for the privacy budget-ϵ at the optimum value of 1.25 and a *MinimumSupport* of 20%. The value of α grows from 2 to 10 with an interval of 2. We can see that the error rate reduces between 60% to 63% when α grows between 2 to 8. The error rate reduces to about 40% when α is 10. We expected this behavior since α determines the number of pivot timestamps, thus determining the the randomness added due to exponential mechanism while partitioning. That is, the greater the value of α, the less is the number of pivot timestamps. This increases the utility but affects the computation

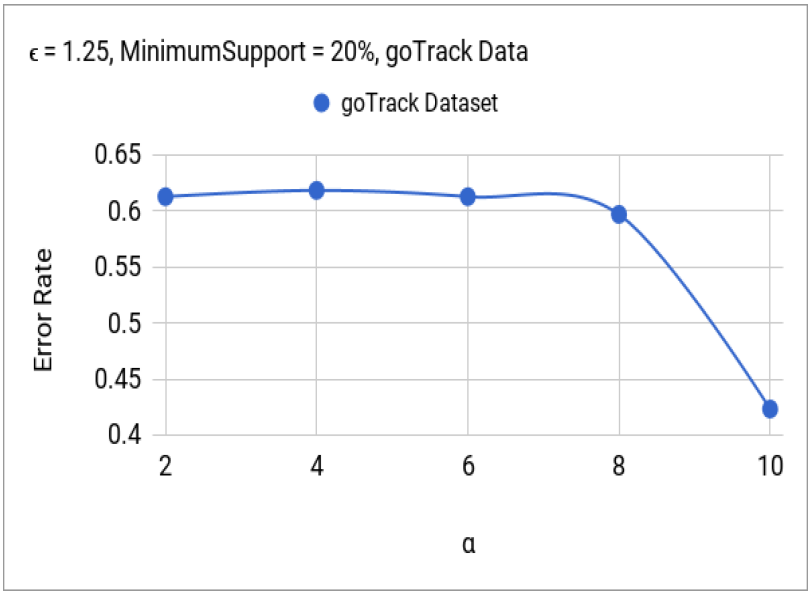


Figure 5.7: Error rate across different α values for frequent sequences over TDrive Dataset.

time adversely.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Summary

As information exchange is becoming an integral component for communication among individuals, companies and government organizations, it has become essential to maintain a safe framework for data exchange. In this thesis, we thoroughly review recent research pertaining to privacy-preserving data publishing (PPDP) via differential privacy, and propose a robust algorithm for publishing trajectory data in a differentially-private manner that is suitable for count queries, range queries, and frequent sequential pattern mining.

In Chapter 2 we first discuss the preliminaries for movement data. We define and discuss differential privacy and its variations, and the mechanisms to achieve differential privacy. We then present the difference between privacy-preserving data publishing and other interactive privacy-preserving frameworks.

In Chapter 3, we review, discuss and compare existing techniques for achieving differential privacy. Most of these techniques consider single publishing by a single trusted data publisher and therefore apply differential privacy for the first release to the first recipient. We have also assessed and discussed other possible data publishing scenarios and works that have proposed techniques for publishing data for such complex scenarios, including multiple publishing, incremental publishing and collaborative data publishing.

In Chapter 4, we proposed a novice technique for publishing trajectory movement data

that utilizes differential privacy to guarantee the privacy of the individuals involved in the data, while maintaining the utility of the published data. We propose several algorithms to compute various steps of achieving differential privacy on the trajectory data. Our approach maintains high utility by adjusting the value of the privacy budget ϵ, calibrating a co-efficient α for determining the number pivot timestamps generated.

In Chapter 5, we examined the performance of our approach in terms of scalability, efficiency and utility. From the experimental results, we concluded that our approach is scalable, efficient and provides good utility for count queries, range queries and mining frequent sequential patterns.

In a nutshell, the main contribution of this thesis is to propose an algorithm for publishing trajectory data in a differentially-private manner, while guaranteeing scalability, efficiency and utility of the published data.

6.2 Looking Ahead

One future work is to utilize the distance between locations and incorporate that as a consistency constraint when generalizing trajectories. This will limit possibility of occurrence of locations in trajectories that might never occur at that sequence. In turn, it would further enhance the accuracy of the differentially-private published data.

We know that the real count of trajectories in the raw dataset is not equal to the generated noisy count in the differentially-private dataset using our proposed approach. This affects the utility of the published dataset; The count queries and range queries do not seem to face adverse effect due to this. However, when frequent sequential patterns are mined, the noise added increases the size of the differentially-private dataset as compared to the raw dataset. These increased numbers of trajectories in the published dataset seem

to adversely affect the utility of the published data when frequent sequential patterns are mined. Therefore, another future work is investigate how to ensure that the size of the anonymized output data is at the same order of the raw data.

Bibliography

[1] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Data Engineering, IEEE International Conference on*. IEEE, 2008.

[2] Charu C. Aggarwal and Philip S. Yu. *A General Survey of Privacy-Preserving Data Mining Models and Algorithms*. Springer US, 2008.

[3] Dakshi Agrawal and Charu C Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2001.

[4] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, 2000.

[5] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 2000.

[6] Dima Alhadidi, Noman Mohammed, Benjamin CM Fung, and Mourad Debbabi. Secure distributed framework for achieving ε-differential privacy. In *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2012.

[7] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: A holistic solution to contingency table release. In *Proceedings of ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2007.

[8] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of ACM Symposium on Theory of Computing*, 2008.

[9] Ji-Won Byun, Yonglak Sohn, Elisa Bertino, and Ninghui Li. Secure anonymization for incremental datasets. In *Workshop on Secure Data Management*, 2006.

[10] Jianneng Cao, Panagiotis Karras, Panos Kalnis, and Kian-Lee Tan. Sabre: A sensitive attribute bucketization and redistribution framework for t-closeness. *The VLDB Journal*, 2011.

[11] Y. Cao and M. Yoshikawa. Differentially private real-time data release over infinite trajectory streams. In *IEEE International Conference on Mobile Data Management*, 2015.

[12] Shuchi Chawla, Cynthia Dwork, Frank McSherry, and Kunal Talwar. On privacy-preserving histograms. *CoRR*, 2012.

[13] Rui Chen, Gergely Acs, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the ACM conference on Computer and communications security*. ACM, 2012.

[14] Rui Chen, Bipin C. Desai, Noman Mohammed, Li Xiong, and Benjamin C. M. Fung. Publishing setvalued data via differential privacy, 2011.

[15] Rui Chen, Benjamin Fung, and Bipin C Desai. Differentially private trajectory data publication. *arXiv preprint arXiv:1112.2020*, 2011.

[16] Rui Chen, Benjamin Fung, Bipin C Desai, and Nériah M Sossou. Differentially private transit data publication: a case study on the montreal transportation system. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

[17] Rui Chen, Benjamin C.M. Fung, Noman Mohammed, Bipin C. Desai, and Ke Wang. Privacy-preserving trajectory data publishing by local suppression. *Information Sciences*, 2013.

[18] Tingting Chen, Siyong Huang, Hong Chen, Yingjie Wu, and Xiaodong Wang. Differentially private two-dimension sparse data publication under consistency. In *Cloud Computing and Big Data (CloudCom-Asia), International Conference on*, 2013.

[19] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y Zhu. Tools for privacy preserving distributed data mining. *ACM Sigkdd Explorations Newsletter*, 2002.

[20] Merce Crosas, Gary King, James Honaker, and Latanya Sweeney. Automating open science for big data. *The ANNALS of the American Academy of Political and Social Science*, 2015.

[21] Michael O. Cruz, Hendrik Macedo, and Adolfo Guimaraes. Grouping similar trajectories for carpooling purposes. In *Proceedings of the 2015 Brazilian Conference on Intelligent Systems*, 2015.

[22] Josep Domingo-Ferrer and Vicenç Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 2005.

[23] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. 2000.

[24] Cynthia Dwork. Ask a better question, get a better answer a new approach to private data analysis. In *Proceedings of International Conference on Database Theory*, 2006.

[25] Cynthia Dwork. *Differential Privacy*. Springer Berlin Heidelberg, 2006.

[26] Cynthia Dwork. Differential privacy. In *Proceedings of International Conference on Automata, Languages and Programming*, 2006.

[27] Cynthia Dwork. Differential privacy: A survey of results. In *Proceedings of International Conference on Theory and Applications of Models of Computation*, 2008.

[28] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of Conference on Theory of Cryptography (TCC)*, 2006.

[29] Alexandre Evfimievski. Randomization in privacy preserving data mining. *SIGKDD Explorer Newsletter*, pages 43–48.

[30] Liyue Fan, Li Xiong, and Vaidy Sunderam. Differentially private multi-dimensional time series release for traffic monitoring. In *Proceedings of the Annual Conference on Data and Applications Security and Privacy*. Springer-Verlag New York, Inc.

[31] Arik Friedman and Assaf Schuster. Data mining with differential privacy. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[32] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys*, 2010.

[33] Benjamin Fung, Ke Wang, Ada Wai-Chee Fu, and Jian Pei. Anonymity for continuous data publishing. In *Proceedings of international conference on Extending database technology: Advances in database technology*, 2008.

[34] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys*, 2010.

[35] Benjamin C. M. Fung, Ke Wang, and Philip S. Yu. Top-down specialization for information and privacy preservation. In *Proceedings of International Conference on Data Engineering(ICDE)*, 2005.

[36] Benjamin CM Fung, Ke Wang, and S Yu Philip. Anonymizing classification data for privacy preservation. *IEEE transactions on knowledge and data engineering*, 2007.

[37] Moein Ghasemzadeh, Benjamin CM Fung, Rui Chen, and Anjali Awasthi. Anonymizing trajectory data for passenger flow analysis. *Transportation research part C: emerging technologies*, 2014.

[38] Normand Grgoire and Mikael Bouillot. Hausdroff distance between complex polygons, 1988.

[39] Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, 2011.

[40] M. Hardt. *A Study of Privacy and Fairness in Sensitive Data Analysis*. PhD thesis, Princeton University, 2011.

[41] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Foundations of Computer Science*, 2010.

[42] Moritz Hardt, Katrina Ligett, and Frank Mcsherry. A simple and practical algorithm for differentially private data release. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*. 2012.

[43] Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia M. Procopiuc, and Divesh Srivastava. Dpt: Differentially private trajectory synthesis using hierarchical reference systems. *Proc. VLDB Endow.*

[44] Shen-Shyang Ho and Shuhua Ruan. Preserving privacy for interesting location pattern mining from trajectory data. *Transactions on Data Privacy*, 2013.

[45] Bijit Hore, Ravi Chandra Jammalamadaka, and Sharad Mehrotra. Flexible anonymization for privacy preserving data publishing: A systematic search based approach. In *SDM*, 2007.

[46] Jingyu Hua, Yue Gao, and Sheng Zhong. Differentially private publication of general time-serial trajectory data. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015.

[47] Jingyu Hua, Yue Gao, and Sheng Zhong. Differentially private publication of general time-serial trajectory data. In *IEEE Conference on Computer Communications*, 2015.

[48] Zhengli Huang, Wenliang Du, and Biao Chen. Deriving private information from randomized data. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. ACM, 2005.

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

[50] Wei Jiang and Chris Clifton. Privacy-preserving distributed k-anonymity. In *IFIP Annual Conference on Data and Applications Security and Privacy*, 2005.

[51] Wei Jiang and Chris Clifton. A secure distributed framework for achieving k-anonymity. *The VLDB JournalThe International Journal on Very Large Data Bases*, 2006.

[52] Aaron Johnson and Vitaly Shmatikov. Privacy-preserving data exploration in genome-wide association studies. In *Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013.

[53] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *Proceedings of the VLDB Endowment*, 2011.

[54] Michael Kern. Anonymity: A formalization of privacy-l-diversity. In *Proceeding zum Seminar Future Internet (FI), Innovative Internet Technologien und Mobilkommunikation and Autonomous Communication Networks*, 2013.

[55] Daniel Kifer and Johannes Gehrke. Injecting utility into anonymized datasets. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2006.

[56] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2011.

[57] P Mayil Vel Kumar and M Karthikeyan. L-diversity on k-anonymity with external database for improving privacy preserving data publishing. *International Journal of Computer Applications*, 2012.

[58] David Leoni. Non-interactive differential privacy: A survey. In *Proceedings of the First International Workshop on Open Data*, 2012.

[59] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE International Conference on Data Engineering*, April 2007.

[60] N. Li, T. Li, and S. Venkatasubramanian. Closeness: A new privacy measure for data publishing. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

[61] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE International Conference on Data Engineering*, 2007.

[62] Kun Liu, H. Kargupta, and J. Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering*, 2006.

[63] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 2007.

[64] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 2007.

[65] Alexandra Marin and Barry Wellman. Social network analysis: An introduction. *The SAGE handbook of social network analysis*, 2011.

[66] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil Vadhan. The limits of two-party differential privacy. In *Proceedings of Symposium on Foundations of Computer Science*, 2010.

[67] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, Annual IEEE Symposium on*, 2007.

[68] Frank McSherry. Privacy integrated queries. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2009.

[69] Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 627–636, June.

[70] Frank D. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2009.

[71] Noman Mohammed, Dima Alhadidi, Benjamin C. M. Fung, and Mourad Debbabi. Secure two-party differentially private data release for vertically partitioned data. *IEEE Transactions on Dependable and Secure Computing*, 2014.

[72] Noman Mohammed and title = Differentially Private Data Release for Data Mining booktitle = Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining series = year = 2011 Chen, Rui an@inproceedingsMohammedd Fung, Benjamin C.M. and Yu, Philip S.

[73] Noman Mohammed, Xiaoqian Jiang, Rui Chen, Benjamin CM Fung, and Lucila Ohno-Machado. Privacy-preserving heterogeneous health data sharing. *Journal of the American Medical Informatics Association*, 2013.

[74] Anna Monreale, Gennady L Andrienko, Natalia V Andrienko, Fosca Giannotti, Dino Pedreschi, Salvatore Rinzivillo, and Stefan Wrobel. Movement data anonymity through generalization. *Trans. Data Privacy*, 2010.

[75] James Munkres. Topology, ed. *Massachusetts Institute of Technology*, 1999.

[76] M. E. Nergiz and C. Clifton. δ-presence without complete world knowledge. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

[77] M. Ercan Nergiz and Chris Clifton. Thoughts on k-anonymization. *Data Knowl. Eng.*, 2007.

[78] Mehmet Ercan Nergiz, Maurizio Atzori, and Chris Clifton. Hiding the presence of individuals from shared databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2007.

[79] Mehmet Ercan Nergiz, Maurizio Atzori, and Yucel Saygin. Towards trajectory anonymization: a generalization-based approach. In *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*. ACM, 2008.

[80] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of ACM Symposium on Theory of Computing*, 2007.

[81] Martin Pettai and Peeter Laud. Combining differential privacy and secure multiparty computation. In *Proceedings of Computer Security Applications Conference*, 2015.

[82] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2010.

[83] Vibhor Rastogi, Dan Suciu, and Sungho Hong. The boundary between privacy and utility in data publishing. In *Proceedings of International Conference on Very Large Data Bases(VLDB)*, 2007.

[84] D. Rebollo-Monedero, J. Forne, and J. Domingo-Ferrer. From t-closeness-like privacy to postrandomization via information theory. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

[85] Daniele Riboni and Claudio Bettini. Incremental release of differentially-private check-in data. *Pervasive Mob. Comput.*

[86] Daniele Riboni and Claudio Bettini. Incremental release of differentially-private check-in data. *Pervasive Mobile Computing*, 2015.

[87] Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *Proceedings of ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1998.

[88] 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.

[89] Reza Sherkat, Jing Li, and Nikos Mamoulis. Efficient timestamped event sequence anonymization, 2011.

[90] Rieffel Chow Song Shi, Chan. Privacy-preserving aggregation of time-series data. In *Proceedings of NDSS*, 2011.

[91] Jordi Soria-Comas, Josep Domingo-Ferrer, David Sánchez, and Sergio Martínez. Improving the utility of differentially private data releases via k-anonymity. In *IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013.

[92] Jordi Soria-Comas, Josep Domingo-Ferrer, David Sanchez, and Sergio Martinez. t-closeness through microaggregation: Strict privacy with enhanced utility preservation. *IEEE Transactions on Knowledge and Data Engineering*, 2015.

[93] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowledge-Based Systems*, 2002.

[94] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002.

[95] Manolis Terrovitis and Nikos Mamoulis. Privacy preservation in the publication of trajectories. In *Mobile Data Management*. IEEE, 2008.

[96] Hien To, Liyue Fan, and Cyrus Shahabi. Differentially private h-tree.

[97] Vassilios S. Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Rec.*, 2004.

[98] Ke Wang and Benjamin C. M. Fung. Anonymizing sequential releases. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[99] Ke Wang, Benjamin C. M. Fung, and Philip S. Yu. Template-based privacy preservation in classification problems. In *Proceedings of IEEE International Conference on Data Mining(ICDM)*, 2005.

[100] Ke Wang, Benjamin C. M. Fung, and Philip S. Yu. Handicapping attacker's confidence: An alternative to k-anonymization. *Knowledge Information Systems*, pages 345–368, 2007.

[101] Ke Wang, Benjamin CM Fung, and Guozhu Dong. Integrating private databases for data analysis. In *International Conference on Intelligence and Security Informatics*, 2005.

[102] Pingshui Wang and Jiandong Wang. L-diversity algorithm for incremental data release. *Applied Mathematics & Information Sciences*, 2013.

[103] Yue Wang, Xintao Wu, and Leting Wu. Differential privacy preserving spectral graph analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2013.

[104] Oliver Williams and Frank McSherry. Probabilistic inference and differential privacy. In *NIPS*, 2010.

[105] Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu, and Ke Wang. (α, k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[106] Xiaokui Xiao and Yufei Tao. M-invariance: towards privacy preserving republication of dynamic datasets. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007.

[107] Yonghui Xiao, James Gardner, and Li Xiong. Dpcube: Releasing differentially private data cubes for health information. In *IEEE International Conference on Data Engineering*, 2012.

[108] Xiaolin Yang, Stephen E Fienberg, and Alessandro Rinaldo. Differential privacy for protecting multi-dimensional contingency table data: Extensions and applications. 2011.

[109] Chao Yao, X. Sean Wang, and Sushil Jajodia. Checking for k-anonymity violation by views. In *Proceedings of International Conference on Very Large Data Bases(VLDB)*, 2005.

[110] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011.

[111] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: Driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010.

[112] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In *IEEE International Conference on Data Engineering*, 2007.

[113] Xiaojian Zhang, Xiaofeng Meng, and Rui Chen. *Database Systems for Advanced Applications Proceedings*, chapter Differentially Private Set-Valued Data Release against Incremental Updates. 2013.

[114] Xiaojian Zhang, Xiaofeng Meng, and Rui Chen. Differentially private set-valued data release against incremental updates. In *International Conference on Database Systems for Advanced Applications*. Springer, 2013.

[115] Yu Zheng. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.*, 2015.

[116] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM Sigkdd Explorations Newsletter*, 2008.