Summer 2021

Optimization of transit smart card data publishing based on differential privacy

Chenxi Chen

Department:

OPTIMIZATION OF TRANSIT SMART CARD DATA PUBLISHING BASED ON DIFFERENTIAL PRIVACY

by

CHENXI CHEN

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

CIVIL ENGINEERING

2021

Approved by:

Xianbiao Hu, Advisor
Jenny Liu
Hongyan Ma

ABSTRACT

Privacy budget allocation is a key step of the differential privacy (DP)-based privacy-preserving data publishing (PPDP) algorithm development, as it directly impacts the data utility of the released dataset. This research describes the development of an optimal privacy budget allocation algorithm for transit smart card data publishing, with the goal of publishing non-interactive sanitized trajectory data under a differential privacy definition. To this end, after storing the smart card trajectory data with a prefix tree structure, a query probability model is built to quantitatively measure the probability of a trajectory location pair being queried. Next, privacy budget is calculated for each prefix tree node to minimize the query error, while satisfying the differential privacy definition. The optimal privacy budget values are derived with Lagrangian relaxation method, with several solution property proposed. Real-life metro smart card data from Shenzhen, China that includes a total of 2.8 million individual travelers and over 220 million records is used in the case study section. The developed algorithm is demonstrated to output sanitized dataset with higher utilities when compared with previous research.

ACKNOWLEDGMENTS

I cannot express enough gratitude to my research advisor, Dr. Xianbiao Hu, for leading me in the way of doing research with his support and encouragement. His advising assists me in all the time of study, research and writing this work. Beside my advisor, I would like to thank the committee for their continued support and instruction: Dr. Jenny Liu and Dr. Hongyan Ma. I offer my sincere appreciation for the learning opportunities provided by my committee.

I would like to thank my colleagues Qing, Yang and Yanqiu for the selfless assistance in study, research and life, for all the days and nights we work together and for all the laughter and depression we have had in the lab.

Finally, my completion of this program could not have been accomplished without the support of my parents – thank you for allowing your only child leaving you for the other side of the earth for several years. My sincerely thanks.

.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page

LIST OF TABLES

Page

NOMENCLATURE

Symbol	Description
$\mathcal{D}$	Trajectory dataset
$\widehat{\mathcal{D}}$	Sanitized trajectory dataset output by PPDP algorithm
$\mathcal{T}$	Timestamp domain
$\mathcal{L}$	Location domain
t	Timestamp, $t \in \mathcal{T}$
l	Location, $l \in \mathcal{L}$
$\mathcal{PT}$	Prefix tree
v_0	Root of prefix tree $\mathcal{PT}$
E	Set of edges of prefix tree $\mathcal{PT}$, each edge represents a pair of timestamp and location
V	Set of nodes of prefix tree $\mathcal{PT}$, each node stores the count of a sub-trajectory
v_i	A tree node in set V
c_i	A count number on node v_i
e_{in}	A tree edge in set E and an in-edge of node v_i
e_{out}	A tree edge in set E and an out-edge of node v_i
$t_i l_i$	A trajectory point on edge e_{in}, $t_i \in \mathcal{T},\ l_i \in \mathcal{L}$
$t_{i+1} l_{i+1}$	An adjacent trajectory point with $t_i l_i$ on edge e_{out}, $t_{i+1} \in \mathcal{T},\ l_{i+1} \in \mathcal{L}$
h	Prefix tree height
θ	Threshold to determine if a noisy prefix tree node should be deleted or not

tr	Trajectory of a trip that include pairs of timestamp and location, represented by E in a prefix tree
ϵ	Privacy budget
δ	Parameter that relaxes differential privacy requirements

1. INTRODUCTION

Privacy issues have been a major concern in transportation engineering, as transportation datasets usually capture each individual traveler's spatial-temporal movements and, as a common practice, to make them publicly available after some simple attempts at anonymity. In this manuscript, we focus on data collected by a smart card (or IC card) that record the payment history of travelers who boarded and/or alighted from transit vehicles in Shenzhen, China. The dataset being analyzed includes a total of 2.8 million different travelers and over 220 million records. One would think that, with merely two boarding/alighting records for each trip, and without including personal information (such as names, home addresses, and dates of birth), such data would not impose a privacy concern. However, our analysis shows that, if a traveler's two travel records are known, and by using subway station names and departure times (with an accuracy of 10 minutes), 30.7% of users can be uniquely identified even though their personal information has been removed from the original dataset.

Figure 1.1 presents part of June 7, 2016, Shenzhen metro smart card dataset, which includes anonymous ID and ride records. Each line includes an anonymous identifier for the passenger, part of the trajectory records, and sensitive information that can be inferred from historical trajectories (such as home and work address). For example, as shown in the second line, a user with a pseudo-identifier ID 20016755 checked into "Bu Xin" station at 09:24am, and then "Fu Tian" station at 09:52am. The red line in Figure 1.1 represents the background knowledge owned by the attacker. The green line represents the sensitive information that an attacker may obtain. If an attacker

has already known Alice has traveled to “Bu Xin station” on that day, around 7:20am-7:30am (i.e., with an accuracy of 10 minutes), and to “Long Cheng Center” station (on the same day) around 8:09am -8:19am, Alice’s unique ID can easily be found to be 20015461 as she is the only passenger with these two travel records in the dataset. With this information, the attackers can discover all historical travel records for Alice, and use them to infer sensitive personal information (such as approximate home and work addresses and other living habits).

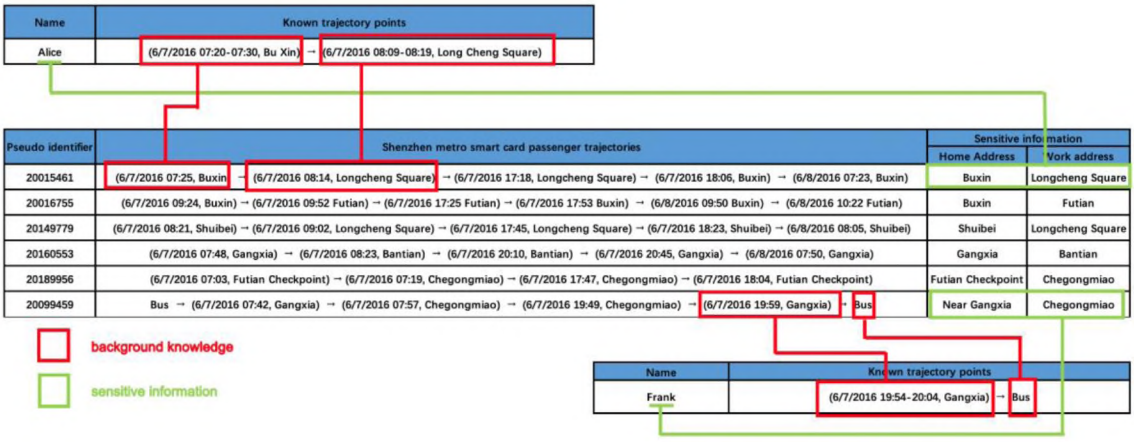

Figure 1.1 Sample smart card data.

The degree for a privacy breach increases when more background knowledge of the trip or traveler becomes available. For example, if an attacker already knows that Frank has traveled to “Gangxia station” on June 7, 2016, at around 19:00-21:00 (i.e., with an accuracy of 2 hours), they are not sure if his identification number is 20160553 or 20099459. However, if they know that Frank rode on a bus right after the subway, then Frank’s unique ID can easily be identified as 20099459. From our experiment, if we have a passenger’s background information on a bus transfer, then the likelihood of him/her

being uniquely identified in the dataset will increase to 41.4%. In other words, almost half of the people using smart cards are identifiable and an attacker can use such information to view an individual's complete travel history in the dataset.

One of the previous research (Li et al. 2020) focuses on enhancing the transit smart card sanitized data utility as well as on improving runtime efficiency. To achieve these goals, a new prefix tree structure, an incremental privacy budget allocation model, and a spatial-temporal dimensionality reduction model are proposed. It argues that previous research allocates privacy budget equally on each layer of the tree, which is problematic due to the nature of the tree structure, when the depth increases, the number of nodes in each layer decreases and the random noise generated by the same amount of privacy budget becomes more significant. The proposed incremental privacy budget allocation model is shown in Equation (1), in which the privacy budget $\epsilon_{\tilde{l}}$ of each level is the results of a function of tree level $\tilde{l}$, and σ is an adjustable parameter. As such privacy budget function $\epsilon_{\tilde{l}}$ is increasing by level, from the top down, $\epsilon_i < \epsilon_{i+1}$, $1 \leq i < h$. The reason is that, under the same privacy budget, the higher the count value is, the smaller the impact would be due to the added noise.

$$\epsilon_{\tilde{l}} = \frac{lg(\tilde{l}+\sigma)}{\sum_{i=1}^{h} lg(\tilde{l}+\sigma)} \times \epsilon, \sigma > 0 \quad (1)$$

While such incremental privacy budget allocation model is demonstrated to improve data utility along the right direction, the proposed model lacks theoretical support to reach optimality and thus, may not lead to the optimal privacy budget allocation solution. In other words, if we can formulate the privacy budget allocation problem as an optimization model, the optimal budget allocation model would lead to not only a theoretical sound but also a verifiable improved solution.

To optimally allocate the privacy budget, we use the same prefix tree structure to store the trajectory data as Li et al. (2020), but develop a query probability model to quantitatively measure the probability of a trajectory location pair being queried. The rational is that the sensitivity of outcomes caused by adding noises depends on the trajectory location pair frequencies if it is queried more frequently, the impact of adding the same amount of noise would become more significant. As such, in the next step we then allocate privacy budget for each prefix tree level based on the query probability of all nodes at that tree level, with the problem formulated with the Lagrangian relaxation method to minimize the query error.

This work is organized as follow. Work related to various privacy protection models is reviewed in Section 2. Some applications of privacy protection methods in transportation engineering are also summarized. Section 3 introduces some preliminary concepts that are relevant with this work, including the definition and properties of differential privacy, prefix tree structure, and the definition of query error. Section 3 designs the query probability model to quantitatively measure the probability of a trajectory location pair being queried. Section 4 formulates the problem with an optimization model for privacy budget allocation for each prefix tree level, and a Lagrangian relaxation method is used to solve the problem. The developed algorithm is implemented and compared with existing models in Section 5, using the real-life metro smart card data of 2.8 million individual travelers and over 220 million records from Shenzhen, China. Section 6 concludes this research along with some discussion of future work.

2. LITERATURE REVIEW

With the ubiquitous applications of intelligent transportation system and the monitoring of trajectories databases in various traffic system, the trajectory data has been widely used in various transportation research. For example, in travel time estimation (Li et al. 2018, Tang and Hu 2020), driving risk analysis (Hu et al. 2015, Zhu et al. 2017, Ma et al. 2018, Ma et al. 2019), congestion mitigation (Cheng et al. 2020, Hu et al. 2020), transit operation improvement (Deng et al. 2020a, Deng et al. 2020c, Tang et al. 2021), taxi behavior modeling (Yu et al. 2019, Deng et al. 2020b, Tang et al. 2020), and many others (An et al. 2017, Chen et al. 2020, Qi and Hu 2020). A common practice is to share the data with researchers, or, even the public. However, privacy preserving is becoming increasingly important when making these data public. Many researchers focused on privacy preserving trajectory data publishing with different privacy-preserving models. In this section, we review those works based on their privacy models.

Syntactic privacy models are widely utilized in trajectory data privacy preserving, such as k-anonymity (Sweeney and Systems 2002) and ϵ- diversity (Machanavajjhala et al. 2007). They stipulated that the output dataset of an anonymization algorithm must adhere to some syntactic conditions in order to protect data records and sensitive items. Nergiz et al. (2008) applied k-anonymity to a trajectory dataset, whereby every trajectory in its entirety must be indistinguishable from at least $k-1$ other trajectories. Abul et al. (2008) proposed (k, δ)-anonymity that enforced space translation, resulting in having every trajectory coexisting with a minimum of $k-1$ other trajectories within a proximity of δ. Monreale et al. (2010) achieved k-anonymity by using spatial generalization. The

novelty of their method lied in dynamically generating geographical areas based on the input dataset, as opposed to generating a fixed grid. Hu et al. (2010) applied k-anonymity to a trajectory dataset with respect to a reference dataset containing sensitive events. Moreover, Barak et al. (2007) developed local enlargement that transforms the trajectory dataset such that every sensitive event was shared by at least k users. In addition to generalization and space translation, suppression-based techniques have been proposed to achieve k-anonymity-based privacy models. Terrovitis and Mamoulis (2008) developed a privacy model that assumed different adversaries possess different background knowledge, and consequently they modeled such knowledge as a set of projections over a sequential trajectory dataset. Their anonymization method limits the inference confidence of locations to a predefined threshold. Fung et al. (2009) and (2010) proposed an LKC-privacy definition that could avoid attacking identity linkages and attribute linkages. Similarly, Cicek et al. (2014) ensured location diversity by proposing p-confidentiality, which limits the probability of visiting a sensitive location to p. Ghasemzadeh et al. (2014) proposed to preserve flow analysis in published trajectories under the LK-anonymity model. We argue that it is possible to achieve comparable analysis results without employing syntactic privacy models, which have been proven to be prone to privacy attacks.

To apply differential privacy in mobility dataset, many works discarded temporal dimension and generate trajectories as sequences. Due to the inherent sparsity and high dimensionality, it is challenging to publish differentially private sequential data. In Chen et al. (2012a), a synthetic dataset based on the Markov assumption was generated from the variable-length n-gram model to ensure the published data was differentially private.

Mir et al. (2013) proposed a differentially private algorithm, namely DP-WHERE, by adding controlled noise to the set of empirical probability distributions that is used to aggregate collections of cellphone Call Detail Records (CDRs) and form a mobility model. He et al. (2015) firstly introduced an end-to-end solution of generating ϵ-differentially private GPS mobility data, called DPT, by constructing prefix trees based on the hierarchical reference system. Xiao and Xiong (2015) protected the true GPS trajectory within a set of probable locations based differential privacy, which is named "δ-location set", to account for the temporal correlations in location data. Gursoy et al. (2018) proposed DP-Star, which added noise under a density-aware grid so that spatial densities can be preserved. Liu et al. (2019) introduced VTDP, which sanitize the fine-grained vehicle trajectories including properties like IDs, positions, speeds, accelerations, and timestamps with differential privacy.

Jiang et al. (2013) sampled distance and angle between true locations within a trajectory in order to publish an ϵ-differentially private version of that trajectory. However, their method publishes a single trajectory only, i.e., the entire privacy budget is spent on sanitizing a single trajectory. Primault et al. (2015) proposed to hide moving individuals' points of interest, such as home or work. While their method protects against inference attacks, we argue that hiding points of interest is harmful for applications that rely on such information, e.g., traffic analysis and probabilistic flow graph analysis.

Chen et al. (2012b) is the first to introduce a differentially private algorithm to publish a large sequential special dataset. The sequential locations were organized as a prefix tree from root node to leaf node and each node recorded the sub-sequence frequent pattern, in which the Laplacian noises were added. Though the authors claimed that it can

be extended to trajectory data, Li et al. (2020) shows this work is not suitable for large trajectory data. Compared with sequential data, trajectory data also contains the time dimension besides the spatial dimension, which is highly valuable in many research areas on one hand, nevertheless, on the other hand, leads to an exponentially increased dimension. Thus, directly adding noise to nodes is extremely challenging (McSherry and Talwar 2007).

Al-Hussaeni et al. (2018) implemented Chen et al.'s extension (Chen et al. 2012a, Chen et al. 2012b), called it SeqPT and generated a new noisy prefix tree which is ϵ-differntially private namely SafePath by introducing a variable height and degree tree with location and timestamps categorized. Real-life transit data experiment suggests that SafePath has significantly higher efficiency and scalability with respect to large and sparse data scenario than SeqPT. As the privacy budget is partly wasted by the taxonomy tree, the utility of published data was decreased under the same private budget. Li et al. (2020)proposed a new prefix tree structure without a taxonomy tree at each level to improve SafePath. An incremental privacy budget allocation mechanism and a spatial-temporal dimensionality reduction model by filtering unreachable nodes are also proposed to enhance the sanitized data utility as well as to improve runtime efficiency. The developed algorithm shows higher utilities and efficiency in the application of real-life metro smart card data.

3. PRELIMINARIES

3.1. TRAJECTORY DATA

Trajectory data is very common in transportation, for example, smart card data, GPS data, camera data. These data are collected by different equipment and organizations. But in essence, they describe the vehicle movements in the spatial and temporal dimension. As such, conceptually we can extract two important dimensions from these very detailed data, with a number representing the time dimension, and a letter representing the space dimension, i.e., 1Y→ 2X, means the trajectory from space Y at time 1 to space X at time 2.

Table 3.1 Trajectory dataset.

ID	Trajectory
tr_1	1Y→ 4X
tr_2	2X→ 3Z
tr_3	2X→ 3Z→ 4Y
tr_4	2Y→ 4X
tr_5	2Y→ 3Z
tr_6	3X→ 4Y
tr_7	1Z→ 2X→ 3Z
tr_8	1Z→ 4X

Table 3.1 is an example of a trajectory dataset which includes a total of eight trajectory data. Among them, the first trajectory data tr_1 travels from location Y at time slot 1 to location X at time slot 4. Note that t_i is strictly increasing in the sequence. $|tr|$

denotes the trajectory length which is the number of timestamp and location pairs in tr, for example we have $|tr_1| = 2$ in Table 3.1.

3.2. PREFIX TREE

Prefix tree is a commonly seen tree structure to organize structural data. A prefix tree is a kind of tree data structure that is often used to store a dictionary table or some sequence of characters. The trajectory data concerned in this manuscript is a kind of spatial-temporal sequence data, which makes a prefix tree a good match.

For example, corresponding to the example shown in Figure 3.1. The maximum tree height is 4, the longest tree is 4 like $A \rightarrow C \rightarrow H \rightarrow N$. The shortest tree length is 3, like $A \rightarrow B \rightarrow G$.

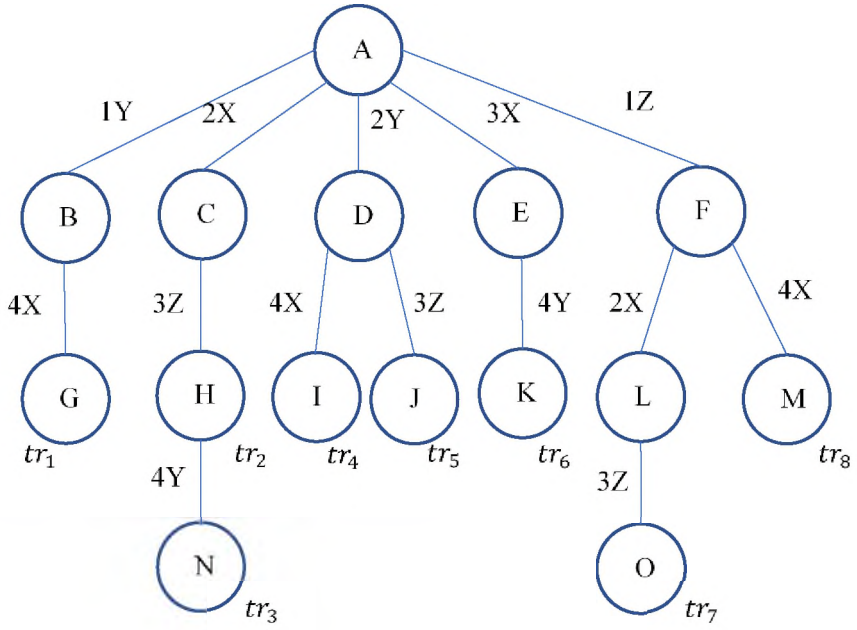

Figure 3.1 Prefix tree of sample trajectory.

In the resulted prefix tree, each user location from the smart card dataset becomes an arc, and two arcs form a trajectory location pair. For example, Arc AB represents 1Y, and arcs AB and BG form a trajectory location pair 1Y->4X, i.e., trajectory tr_1 is now represented by arcs AB and BG, or nodes $A \rightarrow B \rightarrow G$. The node count equals to the number of incoming trajectory data, for example, for node B, the number of incoming trajectory data 1Y is 1, so we have count(B)=1, whereas for node C we have count(C)=2.

3.3. DIFFERENTIAL PRIVACY

Differential Privacy is different from K-anomality which we need to assume the attackers' knowledge, capabilities, and goals. In contrast, no matter what the attackers know and want to do, differential privacy protects the any kind of individual information, such as sensitive attributes and if the dataset contains a specific individual. That means there is no need to assume the attacking pattern.

Definition 3 (Differential Privacy). A randomized mechanism M gives ϵ-differential privacy if for any neighboring datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ differing by at most one record and for any possible sanitized dataset $\widehat{\mathcal{D}} \in Range(M)$, the following Equation (2) is always satisfied:

$$\Pr[M(\mathcal{D}_1) = \widehat{\mathcal{D}}] \leq exp(\epsilon) \times \Pr\left[M(\mathcal{D}_2) = \widehat{\mathcal{D}}\right] \quad (2)$$

The parameter ϵ refers to the privacy budget, which controls the level of privacy guarantee achieved by mechanism M. A smaller ϵ represents a stronger privacy level and can cause more noise to be added to the true answer. ϵ typically ranges $0 < \epsilon \leq 1$.

Sequential composition properties: Suppose we output via K_1 and K_2 with ϵ_1, ϵ_2 differential privacy, result of (K_1, K_2) is $\epsilon_1 + \epsilon_2$ differentially private.

Parallel composition properties: If the inputs K_1 and K_2 are disjoint, then result of (K_1, K_2) is max (ϵ_1, ϵ_2) differentially private.

3.4. RELATIVE ERROR

Definition 4 (Relative Error) Relative error of count queries on synthetic dataset $\widehat{\mathcal{D}}$ is defined as Equation (3). It represents how different the synthetic dataset $\widehat{\mathcal{D}}$ is from the real trajectory dataset D.

$$relative_{error} = \left(\frac{\left| q(\widehat{\mathcal{D}}) - q(\mathcal{D}) \right|}{max\{q(\mathcal{D}), s\}} \right) \tag{3}$$

where s is a sanity bound and is suggested to take a value of 0.1% of the dataset size.

4. METHODOLOGY

In the section below, we discuss a new algorithm of privacy-preserving trajectory data publishing based on differential privacy. Section 4.1 is a brief outline of the algorithm, Section 4.2 illustrates each step of the algorithm, and Section 4.3 presents a theoretical analysis of privacy and algorithm complexity.

4.1. METHODOLOGY OVERVIEW

This proposed algorithm generates sanitized trajectories by differential privacy with optimized budget. Compared with former algorithms, one major advantage of the methodology is that it considers the probability of different trajectories being queried which improves the utility of the sanitized trajectory when satisfies the same level of privacy security. Another improvement is that our algorithm optimizes the budget allocation of all nodes over prefix tree which contributes to a lower relative error rate when doing the trajectory query as well.

Figure 4.1 illustrates the algorithm framework, which includes four main components. Firstly, a prefix tree with the original time and location trajectory is built. Secondly, we optimize the budget allocation strategy and generate customized budget for each trajectory node in the prefix tree. After that, we add Laplace distributed noise according to the generated budgets to the prefix tree and introduce the sanitized tree, and finally, we traverse and output sanitized trajectories from the sanitized tree.

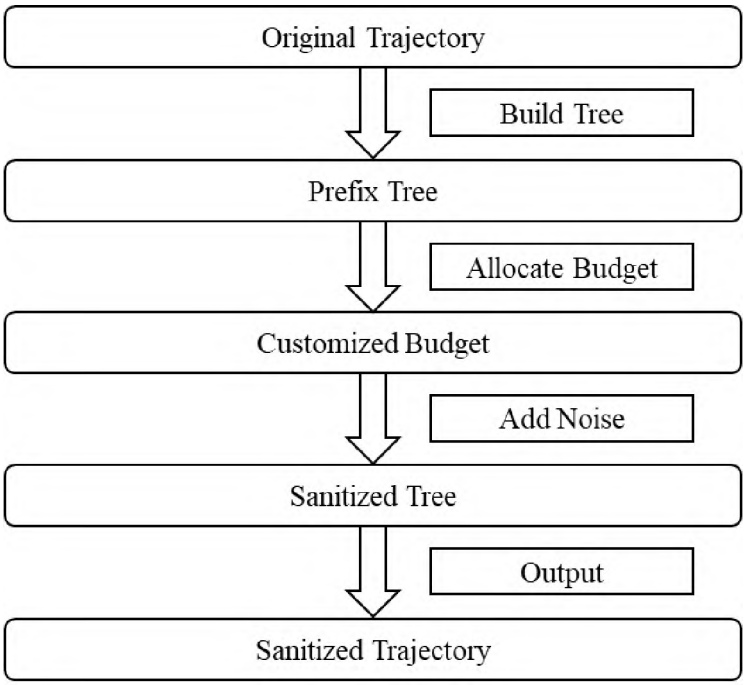

Figure 4.1 Algorithm framework.

4.1.1. Query Probabilistic Model. This section describes the assumed query probability model. Assumptions in trajectory query generation are as following:

1. Queries are generated randomly until all possible trajectory patterns are enumerated.
2. Each location in the trajectory data is sampled from the entire location set with replacement. However, following typical trip patterns, two adjacent locations in a trajectory set cannot be the same.
3. Each trajectory query is generated independently.

For example, if we have three locations, X, Y and Z from a trajectory dataset with $l_{min} = 1$ and $l_{max} = 3$, the set of trajectory data that might be queried by a hacker will include [X, Y, Z] for trajectory = 1, [XY, XZ, YX, YZ, ZX, ZY] for trajectory length = 2,

and [XY, XYZ, XYX, XZ, XZX, XZY, YX, YXY, YXZ, YZ, YZX, YZY, ZX, ZXY, ZXZ, ZY, ZYX, ZYZ] for trajectory length = 3.

In general, if we have a total of n locations, for trajectory length=l, the total number of possible trajectory data can be calculated as Equation (4)

$$N_l = n * (n-1)^{l-1} \tag{4}$$

So, for trajectory length from l_{min}~l_{max}, the total number of possible trajectory data is expressed as Equation (5).

$$\begin{aligned} N = \sum_{l=l_{min}}^{l_{max}} N_l &= \sum_{l=l_{min}}^{l_{max}} n * (n-1)^{l-1} \\ &= \frac{n * (n-1)^{l_{min}-1}[(n-1)^{l_{max}-l_{min}+1} - 1]}{n-2} \end{aligned} \tag{5}$$

A bottom-up approach is used to examine the probability of each arc in the prefix tree. Firstly, we consider from the bottom level (with $l = l_{max}$) and examine the arcs associated with the leaf nodes. As there is only one possible trajectory from a leaf to the root, the probability of a leaf node being queried equals the probability of that the single trajectory being queried, thus we have:

$$p_{leaf} = \frac{1}{N} = \frac{n-2}{n * (n-1)^{l_{min}-1}[(n-1)^{l_{max}-l_{min}+1} - 1]} \tag{6}$$

Then move on to one level up to $l = l_{max} - 1$, if the leaf node has $k - 1$ sibling nodes, i.e., the parent node i has a total of k children nodes, the probability of the parent node being queried equals to the probability of itself being queried, plus that of any of its children node being queried. For example, XY is queried by queries of not only XY but also XYZ and XYX.

So, we have

$$p_i = p_{itself} + k * p_{child} = \frac{1}{N} + k * \frac{1}{N} \tag{7}$$

In general, for each node in the prefix tree, the probability of it being queried equals the probability of itself being queried, plus that of any of its children being queried. So iteratively, the following Equation (8) can be generated for any node i in the prefix tree:

$$p_i = p_{itself} + \sum_{j \in child(i)} p_j = \frac{1}{N} + \sum_{j \in child(i)} p_j \tag{8}$$

4.1.2. Optimal Privacy Budget Allocation Algorithm. We assume the noise follows Laplace distribution. If $X \sim Laplace(\mu, b)$, the probability density function is shown as following:

$$f(x|\mu, b) = \frac{1}{2b} e^{\frac{-|x-\mu|}{b}} = \begin{cases} \frac{1}{2b} e^{\frac{x-\mu}{b}} & if\ x < \mu \\ \frac{1}{2b} e^{\frac{\mu-x}{b}} & if\ x \geq \mu \end{cases} \tag{9}$$

The expectation value of Laplace distribution $E(X) = \mu$, and the variance $D(X) = 2b^2$.

If we assign a privacy budget of ε_i to a prefix tree node i, the potential error brought by adding Laplace noise equals to

$$Err = 2/{\varepsilon_i}^2 \tag{10}$$

For a node i, if Laplace noise with privacy budget ε_i is added, since it is queried for a total of p_i times, the expectation of introduced error by noise can be derived as below.

$$E\big(Err(i)\big) = p_i * 2/{\varepsilon_i}^2 \tag{11}$$

As such, the problem can be formulated as the following equation:

$$Minimize\ f(\varepsilon_i) = \sum_i \frac{1}{2} E\big(Err(i)\big) = \sum_i \frac{p_i}{{\varepsilon_i}^2} \tag{12}$$

Start from any leaf node j to the root, the summation of privacy budget along the path should be equal to ε as shown in Equation (13).

$$\sum_{a \in path(root,j)} \varepsilon_a = \varepsilon \qquad \forall j \epsilon leaf \tag{13}$$

For example, in Figure 3.1, we have a total of 7 leaf nodes, so we have a total of 7 constraints as listed below.

$$\begin{gathered} \varepsilon_G + \varepsilon_B + \varepsilon_A = \varepsilon \\ \varepsilon_N + \varepsilon_H + \varepsilon_C + \varepsilon_A = \varepsilon \\ \dots \\ \varepsilon_M + \varepsilon_F + \varepsilon_A = \varepsilon \end{gathered} \tag{14}$$

The model formulation can be expressed by equation (15).

$$\begin{gathered} Minimize\ f(\vec{\varepsilon}) = \sum_i E\big(Err(i)\big) = \sum_i \frac{p_i}{{\varepsilon_i}^2} \\ s.t. \qquad \sum_{a \in path(root,j)} \varepsilon_a = \varepsilon \quad \forall j \epsilon leaf \end{gathered} \tag{15}$$

Converting the above problem with Lagrangian relaxation method and Equation (16) can be generated. In this equation, λ_a is the dual parameter.

$$Minimize\ L(\varepsilon_i) = \sum_i \frac{p_i}{{\varepsilon_i}^2} + \sum_{j \epsilon leaf} \lambda_a \Big(\sum_{a \in path(root,j)} \varepsilon_a - \varepsilon \Big) \tag{16}$$

For any leaf node i, if we calculate the first derivative and make it equal to 0, Equation (17) is obtained for each of the leaf node.

$$\frac{dL(\varepsilon_i)}{d\varepsilon_i} = -2 * \frac{p_i}{{\varepsilon_i}^3} + \lambda_i = 0 \tag{17}$$

For example, in Figure 1.1, we have a total of 7 equations derived as below.

$$\begin{gathered} -2 * \frac{p_G}{{\varepsilon_G}^3} + \lambda_1 = 0 \\ -2 * \frac{p_N}{{\varepsilon_N}^3} + \lambda_2 = 0 \\ \dots \\ -2 * \frac{p_M}{{\varepsilon_M}^3} + \lambda_7 = 0 \end{gathered} \tag{18}$$

For any non-leaf node i, let calculate the first derivative and make it equal to 0, we will obtain Equation (19) for each of the non-leaf node.

$$\frac{dL(\varepsilon_i)}{d\varepsilon_i} = -2 * \frac{p_i}{{\varepsilon_i}^3} + \sum_{a \in children(i)} \lambda_a = 0 \tag{19}$$

For example, in Figure 1.1, we have a total of 7 equations derived as below.

$$\begin{gathered} -2 * \frac{p_B}{{\varepsilon_B}^3} + \lambda_1 = 0 \\ -2 * \frac{p_D}{{\varepsilon_D}^3} + (\lambda_3 + \lambda_4) = 0 \\ \dots \\ -2 * \frac{p_A}{{\varepsilon_A}^3} + (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7) = 0 \end{gathered} \tag{20}$$

Before we propose the solution algorithms, examine Equations (17) and (19) for a few useful properties in the solutions.

Solution property 1: for a non-leaf node, the summation of privacy budget in any of its child tree is equal. It can be proofed as following:

Suppose a node i has two children, i_1 and i_2, and they are associated with trajectory tr_1 and tr_2 respectively.

Since for any trajectory, starting from its leaf node to the root, the summation of privacy budget added to the nodes along the path should be equal to ε.

Assume tr_1 goes through nodes $tr_{1,1}$, $tr_{1,2}$,.... i_1 before reaching node i, and tr_2 goes through nodes $tr_{2,1}$, $tr_{2,2}$,.... i_2 before reaching node i. We will have the following equations:

$$\sum_{a \in path(tr_{1,1}, i_1)} \varepsilon_a + \varepsilon_i + \sum_{a \in path(i, root)} \varepsilon_a = \varepsilon \tag{21}$$

$$\sum_{a \in path(tr_{2,1}, i_1)} \varepsilon_a + \varepsilon_i + \sum_{a \in path(i, root)} \varepsilon_a = \varepsilon \tag{22}$$

By comparing Equations (21) and (22), we can generate Equation (23)

$$\sum_{a \in path(tr_{1,1}, i_1)} \varepsilon_a = \sum_{a \in path(tr_{2,1}, i_1)} \varepsilon_a \tag{23}$$

The proof is now complete.

Solution property 2: As a special case of solution property 1: the privacy budgets of all leaf nodes with the same parent are equal as shown in Equation (24).

$$\varepsilon_i = \varepsilon_j \; if \; i \; and \; j \; are \; sibling \; leaf \; nodes \tag{24}$$

In Figure 1.1 example, $\varepsilon_I = \varepsilon_J$. Note this property applies only to leaf node. For example, as node L is not a leaf node, $\varepsilon_L \neq \varepsilon_M$.

Solution property 3: For any non-leaf node i, if it covers a total of k children leaf nodes denoted as $c_1, c_2 \ldots c_k$, Equation (25) holds.

$$\frac{p_i}{{\varepsilon_i}^3} = \sum_{a\epsilon[1,k]} \frac{p_{c_a}}{{\varepsilon_{c_a}}^3} \tag{25}$$

In Figure 3.1, I and J are two children leaf nodes of D, N is a child leaf node of H and C, and all leaf nodes of [G, N, I, J, K, O, M] are leaf nodes of A.

From Equation (17) we can derive Equation (26) for any leaf node.

$$\lambda_i = 2 * \frac{p_i}{{\varepsilon_i}^3} \tag{26}$$

Plug in Equation (26) to Equation (19), we will have Equation. (27) for any non-leaf node:

$$\frac{p_i}{{\varepsilon_i}^3} = \sum_{a\epsilon[1,k]} \frac{1}{2}\lambda_a = \sum_{a\epsilon[1,k]} \frac{p_{c_a}}{{\varepsilon_{c_a}}^3} \tag{27}$$

The proof is now complete.

For example, in Figure 1.1, we have

$$\frac{p_B}{{\varepsilon_B}^3} = \frac{p_G}{{\varepsilon_G}^3}$$

$$\frac{p_D}{{\varepsilon_D}^3} = \frac{p_I}{{\varepsilon_I}^3} + \frac{p_J}{{\varepsilon_J}^3} \tag{28}$$

$$\frac{p_A}{{\varepsilon_A}^3} = \frac{p_G}{{\varepsilon_G}^3} + \frac{p_N}{{\varepsilon_N}^3} + \frac{p_I}{{\varepsilon_I}^3} + \frac{p_J}{{\varepsilon_J}^3} + \frac{p_K}{{\varepsilon_K}^3} + \frac{p_O}{{\varepsilon_O}^3} + \frac{p_M}{{\varepsilon_M}^3}$$

And thus, we can calculate ε_i for any non-leaf node i as expressed in Equation (29).

$$\varepsilon_i = \left(\frac{p_i}{\sum_{a\epsilon[1,k]} \frac{p_{c_a}}{{\varepsilon_{c_a}}^3}}\right)^{\frac{1}{3}} \tag{29}$$

With these two solution properties, the optimization problem can be solved in an easier way with linear formulation.

To solve the optimization problem, we can start from the bottom level. For example, node G in Figure 3.1.

4.2. ALGORITHM STEPS

Algorithm 1 is the main algorithm including four parts: build a tree, allocate budgets, add noise and output, as illustrated in Figure 4.1. It inputs a trajectory dataset $\mathcal{D}$ and outputs the sanitized trajectory dataset $\widehat{\mathcal{D}}$.

In Algorithm 1, a raw trajectory dataset $\mathcal{D}$ is scanned once to build a trajectory prefix tree $\mathcal{PT}$, with a given height h (Algorithm 1, Line 1), and then the budgets are allocated to each nodes by Procedure 1 (Algorithm 1, Line 2). After that, noise is added to $\mathcal{PT}$, layer by layer, iteratively, to build a differential private prefix tree in a top-down fashion (Algorithm 1, Line 3-11). In the last, the sanitized trajectory dataset $\widehat{\mathcal{D}}$ is outputted (Algorithm 1, Line 12).

The most important part of the algorithm is to optimize the privacy budget allocation which is implemented in Procedure 1. The input is a prefix tree and the output is the tree with allocated budget ϵ_i of each node. Two recursion functions are included in the procedure. The first one is to allocate budgets for the leaf nodes (Procedure 1, Line 1-4) and the nodes whose every children node has been allocated with an optimized budget utilizing Equation (29) (Procedure 1, Line 5-16). And the second function *Normalize* $(v_i, coef)$ is to make sure that the sum of allocated budgets in each path is always the same, i.e. Equation (23) is always satisfied (Procedure 1, Line 17-24). That ensures for

any trajectory, starting from its leaf node to the root, the summation of privacy budget added to the nodes along the path should be equal to ε.

Algorithm 1. MainFunc

Input: Raw trajectory dataset $\mathcal{D}$, Timestamp domain T, Location domain L

Input: Height of the prefix tree h

Input: Privacy budget ϵ

Input: Threshold parameter k, b

Output: Differentially private trajectory dataset $\widehat{\mathcal{D}}$

1. Scan dataset $\mathcal{D}$ once to build a Prefix tree $\mathcal{PT}$ with height of h;

2. BudgetAllocation($\mathcal{PT}$)

3. $i = 1$;

4. while $i \leq h$ do

5. $\theta_l = k \times l^{-1} + b$;

6. **for** each node v_i in level i **of** $\mathcal{PT}$ **do**

7. add noise to the count value stored in node v_i;

8. BuildChildTree ($v_i, \epsilon_l, \theta_l$);

9. **end for**

10. $i++$;

11. **end while**

12. $\widehat{\mathcal{D}} \leftarrow$ Output ($\mathcal{PT}$);

return $\widehat{\mathcal{D}}$;

Procedure 1. BudgetAllocation

Input: Root node v_o of prefix Tree $\mathcal{PT}$

Input: Total budget ϵ

Output: Prefix Tree with optimized budget ϵ_i for each node $\mathcal{PT}'$

1. **function** Allocate(v_i)
2. **if** v_i is leaf node **then**
3. $v_i.epsilon = \epsilon$
4. **break**
5. **else**
6. **for** each children node v_{i+1} of v_i **do**
7. Allocate(v_{i+1})
8. **end for**
9. $sum = 0$
10. **for** each leaf node v_h of v_i **do**
11. $sum += \epsilon/(v_h.epsilon)^3$
12. **end for**
13. $v_i.epsilon = v_i.children\_count/sum^{1/3}$
14. Normalize $(v_i, v_i.epsilon/\epsilon + 1)$
15. **end if**
16. **end function**
17. **function** Normalize $(v_i, coef)$
18. $v_i.epsilon /= coef$

---

19. **if** v_i is not leaf node **then**
20. **for** each children node v_{i+1} of v_i **do**
21. Normalize $(v_{i+1}, coef)$
22. **end for**
23. **end if**
24. **end function**
25. Start allocation from the root node v_0 of prefix Tree $\mathcal{PT}$: Allocate(v_0)

One of the most important steps in Algorithm 1 is to grow a subtree of each parent node vi by selecting out-edges of e_{out}. This is implemented in Procedure 1. When handling a sub-level, noise is added first to the count on each existing node, according to privacy budget l (Procedure 2, Line 3). If the noise count on a node is greater than, or equal to, threshold l, the node is retained (Procedure 2, Line 4–7). After handling all existing nodes (Procedure 2, Line 2–11), if the summation of the noise on all existing children nodes is less than the noise count c_i on the parent node (Procedure 2, Line 12), then more timestamp and location pairs (that did not exist in the current edges) are randomly selected from a reasonable timestamp and location domain, according to the restricted location domain Lr (Procedure 2, Line 12–27). The count value on the newly selected nodes equals to 0 plus noise (Procedure 1, Line 17), and if the result is greater than, or equal to, l , the node is added to the child node set (Procedure 2, Line 17–19). The newly selected nodes, with an initial count of 0, are called "empty node". If an empty node is selected, the noise count is added to the summation output sum (Procedure 2,

Line 20–21). The summation sum is used to determine when to stop growing out-edges through the accumulation of counts on the child nodes. When the value of summation is greater than, or equal to, the count value c_i, the loop ended (Procedure 2, Line 23–25).

Take the trajectories in Table 3.1 as an example and set the total budget equals to 1. The budget allocation starts from tr_1, and calculated follows Procedure 1. Finally, the optimized result showed in Figure 4.2 can be calculated.

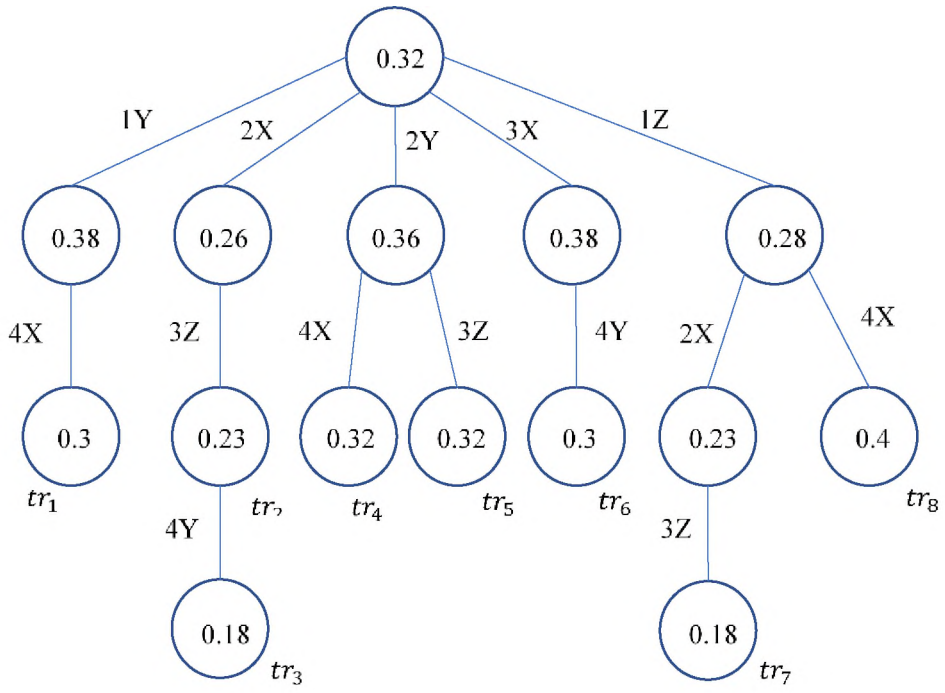

Figure 4.2 Prefix tree with optimized budgets.

Procedure 2. BuildChildTree

Input: Parent node v_i, noisy count c_i, location l_i, time t_i

Input: Privacy budget l , Threshold l

Output: Noisy child nodes set N

1: $sum = 0$;

2: **for** each child node v_{i+1} of v_i **do**

3. $v_{i+1}.count = v_{i+1}.count + v_{i+1}. + \text{Lap}(1/\epsilon_{i+1})$;

4. **if** $v_{i+1}.count > \theta_l$ **then**

5. $N \leftarrow v_{i+1}$;

6. $sum \mathrel{+}= v_{i+1}.count$;

7. **end if**

8. **if** $sum \geq c_i$ then

9. **break**;

10. **end if**

11. **end for**

12. **while** $sum < c_i$ **do**

13. Randomly select a time $t_{i+1} \in T$ **and** $t_{i+1} > t_i$

14. $L_r \leftarrow$ RestrictedLocDom $(l_i, t_{i+1} - t_i)$;

15. for each location $l_{i+1} \in L_r$ do

16. $count = \text{Lap}(1/\epsilon_{i+1}) \in \mathcal{T}_r$ **do**;

17. **if** $count = \text{Lap}(1/\epsilon_{i+1}) + 0$ **then**

18. add e_{out} as a new out-edge with $t_{i+1}l_{i+1}$;

19. add v_{i+1} as a new child node;

20. $v_{i+1}.count = coun$;

21. $sum \mathrel{+}= count$;

22. **end if**
23. **if** $sum \geq c_i$ **then**
24. **break**;
25. **end** if
26. **end** for
27. **end** while
28. **return** $\mathcal{N}$;

4.3. THEORETICAL ANALYSIS

This section includes theoretical analysis of three aspects: algorithm improvement and privacy guarantee.

4.3.1. Algorithm Improvement. We compare the proposed model with SeqPT model (Chen et al. 2012a, Chen et al. 2012b), SafePath model (Al-Hussaeni et al. 2018) and Li et al's model (2020). SeqPT model and SafePath model allocate privacy budget evenly on each level of the prefix tree. Every node at the same level shares the privacy budget, regardless of the number of nodes at each level, due to the characteristics of the prefix tree. And Li et al's model (2020) is designed to assign privacy budgets and threshold to different levels. However, Li et al's model (2020) uses the same budget for all nodes in the same level and their allocation function is qualitive with the equation $\epsilon_l = \frac{\lg(l+\sigma)}{\sum_{l=1}^{h} \lg(l+\sigma)} \times \epsilon$.

For those reasons, we propose a new algorithm to optimize the budget allocation policy and try to allocate more budget to the nodes that have higher probability of being

queried, which finally contributes to increase of the utility of the differential privacy model.

4.3.2. Privacy Guarantee. Algorithm 1 consists of four steps: BuildRawTree, BudgetAllocation, BuildChildTree, and Output. Given the total privacy budget ϵ, the first step converts the original trajectory dataset into the data structure of the trajectory prefix tree, and the second step only calculates and optimizes the budgets allocation of all nodes. Thus, there is no privacy budget consumption in the first two steps.

In third step, we added noises by iteratively constructing one level at a time based on the output of the first step. According to the parallel composition theorem, the entire privacy budget consumed in a level is shared by all the nodes on the same level since all nodes on the same level contain a disjoint set of trajectories. Each level is a dedicated privacy budget portion, since the height of the noisy prefix tree is h, the BuildChildTree consumes the privacy budget in an amount $\sum_{l=1}^{h} \epsilon$. The summation of privacy budget along any path equals to the total privacy budget.

In the last step, we processed the noise prefix tree without accessing the underlying raw trajectories which is similar to the first two steps. Therefor, there is no privacy budget consumption in this step.

In conclusion, with the given privacy budget, Algorithm 1 is ε-differentially private.

5. NUMERICAL EXPERIMENT

This section analyzes the utility and efficiency of proposed algorithm. We used the same data as Li et al. (2020) and follow the evaluation method from previous works (Chen et al. 2012a, Chen et al. 2012b, Al-Hussaeni et al. 2018, Li et al. 2020). The real-life datasets from the Shenzhen Metro smart card records, that are used, cover 2.8 million smart card users. 4 different datasets with different trajectory size, time domain and max trajectory length are listed in Table 5.1 are utilized in the algorithm evaluation. Time Domain represents the time domain size, and every time interval equals to 15 minutes. We evaluate the efficiency and scalability of the proposed algorithm, as well as the utility of the sanitized trajectory data used for counting queries.

Table 5.1 Datasets in numerical experiment.

Dataset	Trajectory Size	Time Domain	Max Trajectory Length
Dataset 1	393,552	16	6
Dataset 2	772,606	48	16
Dataset 3	824,957	64	18
Dataset 4	845,727	80	20

5.1. UTILITY ANALYSIS

This section shows the utility examination result of a sanitized algorithm output. Same as the previous evaluation method, 40,000 random count queries of length $|q| = 2$ are generated to examinate our algorithm to evaluate its utility.

Figure 5.1 indicates the average relative error difference under different total budget and prefix tree height of four datasets. X-axis represents different prefix tree height and Y-axis represents the average relative error. It can be observed that the average relative error generally decreases when the prefix tree height gets higher in dataset1 and Dataset3. However, in Dataset2, the average relative error does not change a lot with the change of prefix tree height. In the other hand, the utility is also influenced by total budget ϵ.

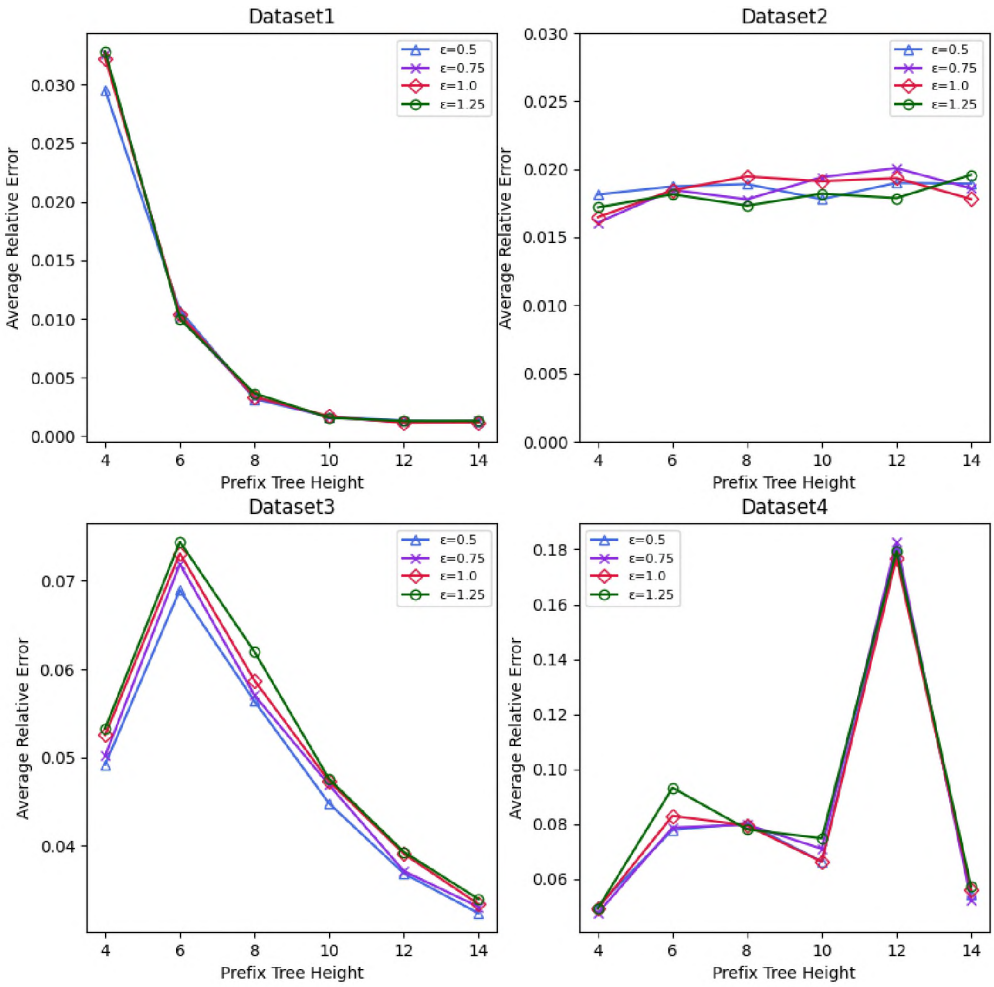

Figure 5.1 Average error comparison under different tree height.

5.2. SCALABILITY ANALYSIS

Figure 5.2 shows the runtime difference under different parameters and datasets. The parameter $h = 14$ and $\varepsilon = 1$.

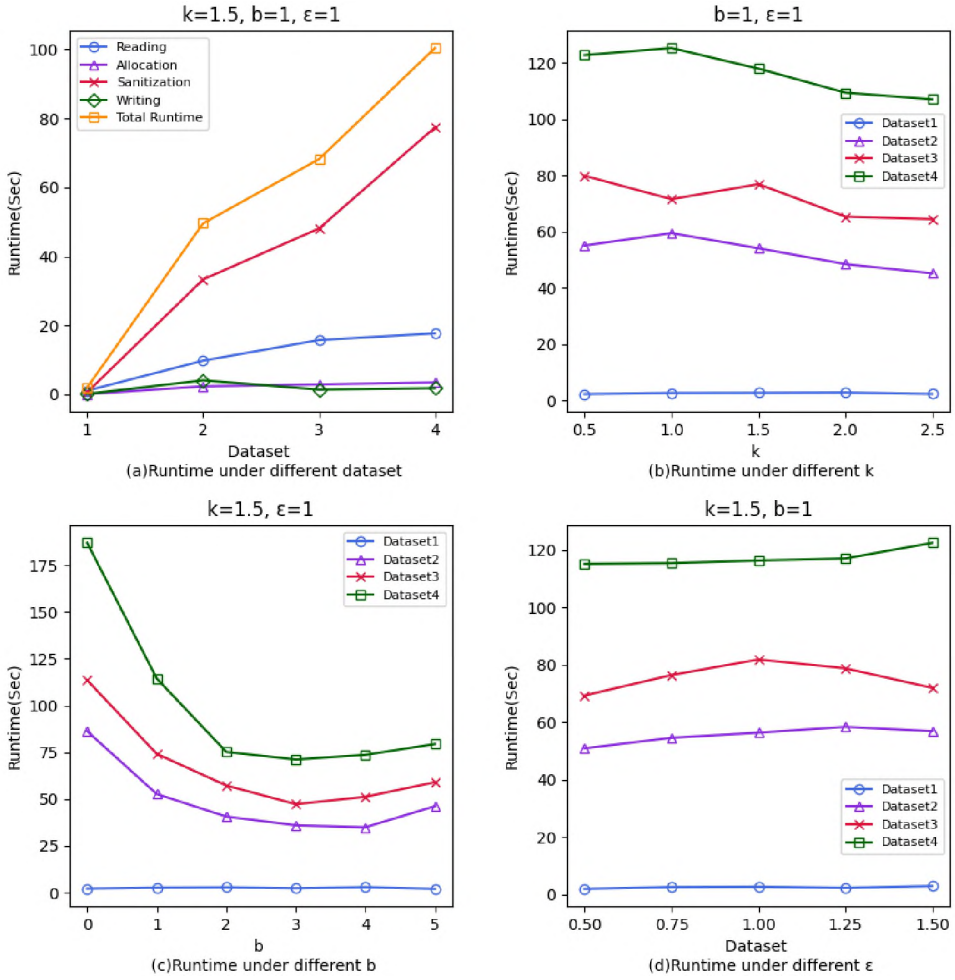

Figure 5.2 Scalability analysis.

Subgraph (a) describes the runtime of each part and the total runtime of the algorithm, including reading, budget allocation, sanitization, writing. X-axis represents different dataset and Y-axis represents runtime. We can conclude that the sanitization

takes most of the total runtime while budget allocation and writing account for a very small partition in subgraph (a).

Subgraph (b), (c), and (d) show the runtime varies under different parameters including threshold function parameters k, b and total budget ε. X-axis represents k, b and ε respectively and Y-axis represents runtime. It can be observed that the algorithm keeps steady in all situations of all 4 datasets. And the largest dataset, dataset 4 spends the longest time in every scenario.

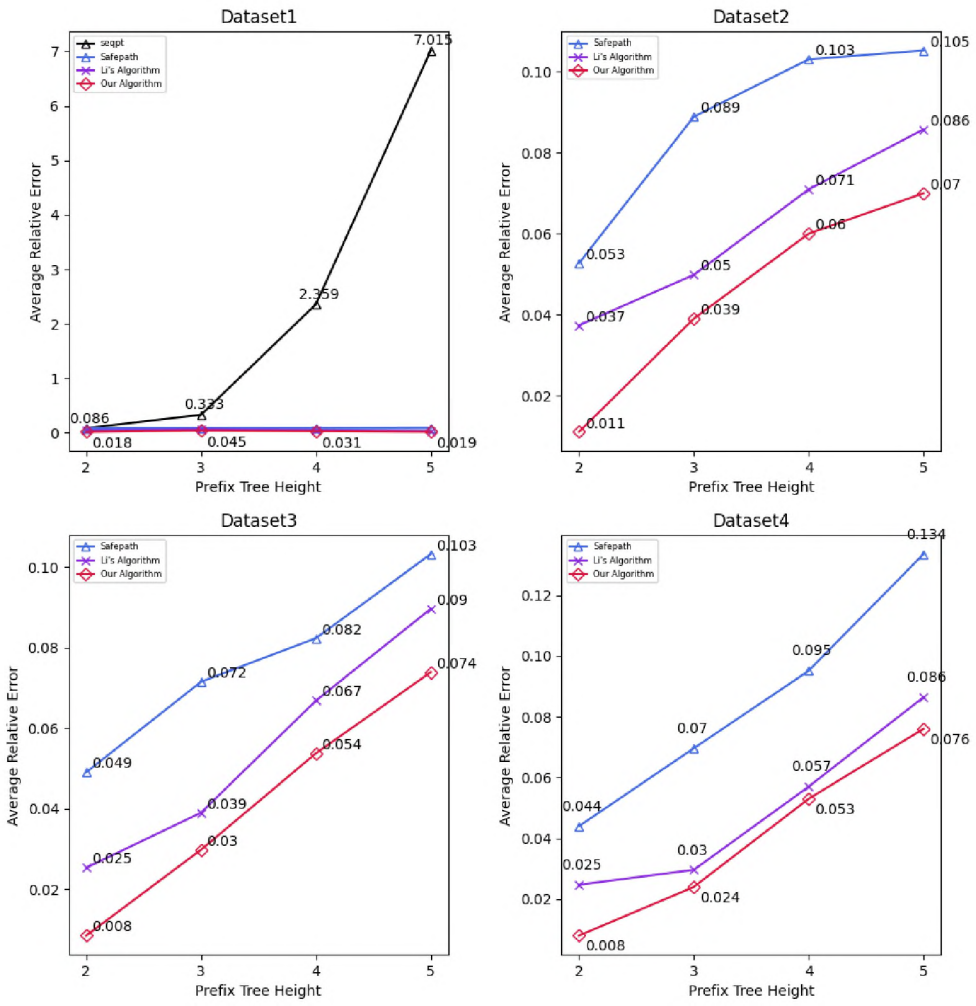

Figure 5.3 Relative error comparison of different datasets under different tree heights.

5.3. COMPARISON WITH OTHER MODELS

This section compares our algorithm's relative error and runtime with SeqPT and SafePath.

Figure 5.3 shows the results of utility comparison, with four subgraphs representing results under four different datasets, max |q| = 2 in this experiment. X-axis represents different h, Y-axis represents average relative error. The proposed algorithm outperforms the other three algorithms at different prefix tree heights, from 2 to 5. This especially occurs under Dataset 1, and the error rate of our algorithm is from 0.018 to 0.045 which is about far less than of Seqpt and it fails to generate a sanitized tree from other three datasets. Our algorithm has a better performance with all metro smart card datasets, which include both a smaller dataset with a lower domain size and a larger dataset with a higher domain size.

Figure 5.4 shows the results of efficiency comparison, with four subgraphs representing results under four different datasets, max |q| = 2 in this experiment. X-axis represents different height and Y-axis represents average relative error. In Dataset 1, the proposed algorithm has a similar runtime with Safepath and the algorithm of Li et al. (2020). However, it is far smaller than Seqpt model which spends almost 120 seconds. In other three datasets, though the proposed algorithm has lower efficiency than the algorithm of Li et al. (2020), which meets the theoretical analysis because of the extra step, budget allocation, it still performs better than SafePath when the height is low and slightly poorer when the prefix tree height gets higher. In general, the runtime keeps increasing in a stable and approximately linear rapid, which guarantees the practicality in big trajectory data sanitization.

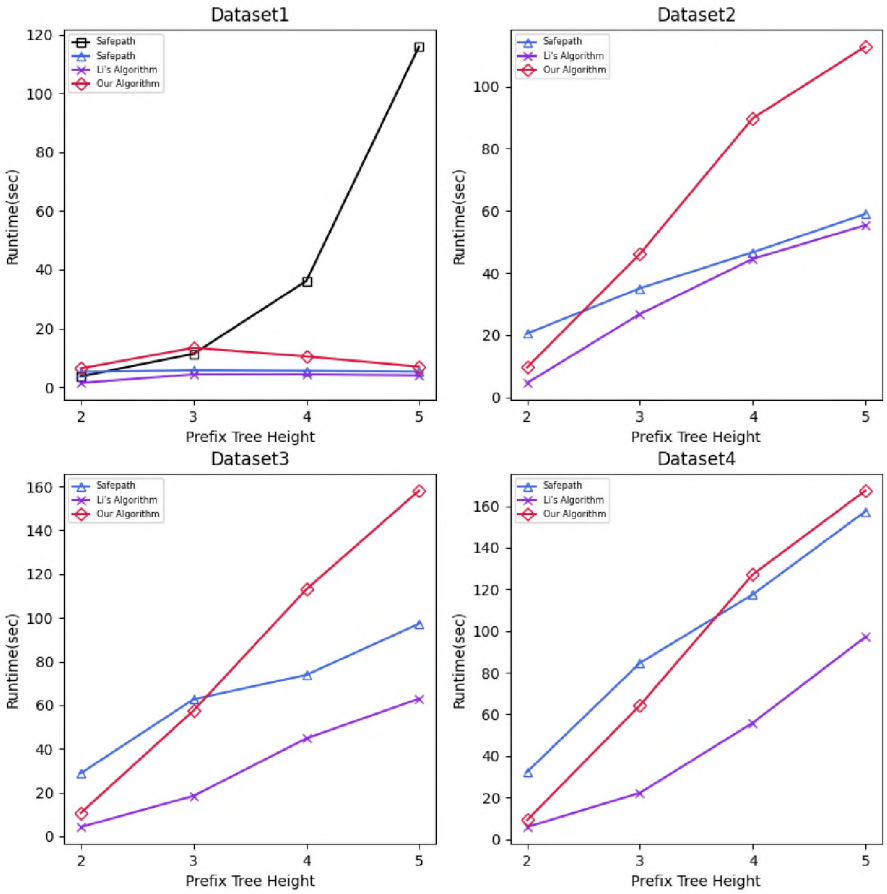

Figure 5.4 Runtime comparison of different datasets under different height.

6. CONCLUSIONS

In this work, we focus on the budget allocation optimization on trajectory data publish with ϵ- differentially private. The trajectory data that we handle is spatial-temporal data with features that are large scale, high-dimensional, and sparse, which brings challenges to improving algorithm efficiency and data utility. A Lagrangian relaxation method is proposed to optimize the budget allocation model, and an incremental privacy budget allocation model is developed to improve data utility. Through theoretical analysis and comparisons with previous works based on real-life trajectory datasets, the proposed algorithm demonstrates more efficient and scalable results. The sanitized trajectory dataset is also shown to have better utility. In addition to the transit smart card data, our method has the potential of being directly applied to other types of trajectory data, such as those from social media, navigation apps, ridesharing, and so on.

REFERENCES

[1] Abul, O., et al. (2008). Never walk alone: Uncertainty for anonymity in moving objects databases. *2008 IEEE 24th international conference on data engineering*, Ieee.

[2] Al-Hussaeni, K., et al. (2018). SafePath: Differentially-private publishing of passenger trajectories in transportation systems. *Computer Networks*. 143126-139.

[3] An, K., et al. (2017). A Network Partitioning Algorithmic Approach for Macroscopic Fundamental Diagram-Based Hierarchical Traffic Network Management. *IEEE Transactions on Intelligent Transportation Systems*. PP(99): 1-10.

[4] Barak, B., et al. (2007). Privacy, accuracy, and consistency too: a holistic solution to contingency table release. *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*.

[5] Chen, C., et al. (2020). Analysis of Electric Vehicle Charging Behavior Patterns with Function Principal Component Analysis Approach. *Journal of Advanced Transportation*. 20208850654.

[6] Chen, R., et al. (2012a). Differentially private sequential data publication via variable-length n-grams. *Proceedings of the 2012 ACM conference on Computer and communications security*.

[7] Chen, R., et al. (2012b). Differentially private transit data publication: a case study on the montreal transportation system. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*.

[8] Cheng, Y., et al. (2020). Monte Carlo Tree Search-Based Mixed Traffic Flow Control Algorithm for Arterial Intersections. *Transportation Research Record*. 2674(8): 167-178.

[9] Cicek, A. E., et al. (2014). Ensuring location diversity in privacy-preserving spatio-temporal data publishing. *The VLDB Journal*. 23(4): 609-625.

[10] Deng, Y.-J., et al. (2020a). Reduce Bus Bunching with a Real-Time Speed Control Algorithm Considering Heterogeneous Roadway Conditions and Intersection Delays. *Journal of Transportation Engineering, Part A: Systems*. 146(7): 04020048.

[11] Deng, Y., et al. (2020b). Heterogenous Trip Distance-Based Route Choice Behavior Analysis Using Real-World Large-Scale Taxi Trajectory Data. *Journal of Advanced Transportation*. 20208836511.

[12] Deng, Y., et al. (2020c). Modeling and Prediction of Bus Operation States for Bunching Analysis. *Journal of Transportation Engineering, Part A: Systems*. 146(9): 04020106.

[13] Fung, B. C., et al. (2009). Privacy protection for RFID data. *Proceedings of the 2009 ACM symposium on Applied Computing*.

[14] Fung, B. C., et al. (2010). Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (Csur)*. 42(4): 1-53.

[15] Ghasemzadeh, M., et al. (2014). Anonymizing trajectory data for passenger flow analysis. *Transportation Research Part C: Emerging Technologies*. 3963-79.

[16] Gursoy, M. E., et al. (2018). Differentially private and utility preserving publication of trajectory data. 18(10): 2315-2329.

[17] He, X., et al. (2015). DPT: differentially private trajectory synthesis using hierarchical reference systems. 8(11): 1154-1165.

[18] Hu, H., et al. (2010). Privacy-aware location data publishing. *ACM Transactions on Database Systems (TODS)*. 35(3): 1-42.

[19] Hu, X., et al. (2015). Studying Driving Risk Factors using Multi-Source Mobile Computing Data. *International Journal of Transportation Science and Technology*. 4(3): 295-312.

[20] Hu, X., et al. (2020). Will information and incentive affect traveler's day-to-day departure time decisions?—An empirical study of decision making evolution process. *International Journal of Sustainable Transportation*. 14(6): 403-412.

[21] Jiang, K., et al. (2013). Publishing trajectories with differential privacy guarantees. *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*.

[22] Li, Y., et al. (2020). A differential privacy-based privacy-preserving data publishing algorithm for transit smart card data. *Transportation Research Part C: Emerging Technologies*. 115102634.

[23] Li, Z., et al. (2018). Reconstructing Vehicle Trajectories to Support Travel Time Estimation. *Transportation Research Record*. 2672(42): 148-158.

[24] Liu, B., et al. (2019). VTDP: Privately Sanitizing Fine-grained Vehicle Trajectory Data with Boosted Utility.

[25] Ma, Q., et al. (2019). Taxicab crashes modeling with informative spatial autocorrelation. *Accident Analysis & Prevention*. 131297-307.

[26] Ma, Y.-L., et al. (2018). The use of context-sensitive insurance telematics data in auto insurance rate making. *Transportation Research Part A: Policy and Practice*. 113243-258.

[27] Machanavajjhala, A., et al. (2007). l-diversity: Privacy beyond k-anonymity. 1(1): 3-es.

[28] McSherry, F. and K. Talwar (2007). Mechanism design via differential privacy. *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, IEEE.

[29] Mir, D. J., et al. (2013). Dp-where: Differentially private modeling of human mobility. *2013 IEEE international conference on big data*, IEEE.

[30] Monreale, A., et al. (2010). Movement data anonymity through generalization. *Trans. Data Priv.* 3(2): 91-121.

[31] Nergiz, M. E., et al. (2008). Towards trajectory anonymization: a generalization-based approach. *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*.

[32] Primault, V., et al. (2015). Time distortion anonymization for the publication of mobility data with high utility. *2015 IEEE Trustcom/BigDataSE/ISPA*, IEEE.

[33] Qi, H. and X. Hu (2020). Real-time headway state identification and saturation flow rate estimation: a hidden Markov Chain model. *Transportmetrica A: Transport Science*. 16(3): 840-864.

[34] Sweeney, L. J. I. J. o. U., Fuzziness and K.-B. Systems (2002). k-anonymity: A model for protecting privacy. 10(05): 557-570.

[35] Tang, Q. and X. Hu (2020). Modeling Individual Travel Time with Back Propagation Neural Network Approach for Advanced Traveler Information Systems. *Journal of Transportation Engineering, Part A: Systems*. 146(6): 04020039.

[36] Tang, Q., et al. (2021). Analytical characterization of multi-state effective discharge rates for bus-only lane conversion scheduling problem. *Transportation Research Part B: Methodological*. 148106-131.

[37] Tang, Q., et al. (2020). Modeling Routing Behavior Learning Process for Vacant Taxis in a Congested Urban Traffic Network. *Journal of Transportation Engineering, Part A: Systems*. 146(6): 04020043.

[38] Terrovitis, M. and N. Mamoulis (2008). Privacy preservation in the publication of trajectories. *The Ninth International Conference on Mobile Data Management (mdm 2008)*, IEEE.

[39] Xiao, Y. and L. Xiong (2015). Protecting locations with differential privacy under temporal correlations. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*.

[40] Yu, X., et al. (2019). A Markov decision process approach to vacant taxi routing with e-hailing. *Transportation Research Part B: Methodological*. 121114-134.

[41] Zhu, X., et al. (2017). A Bayesian Network model for contextual versus non-contextual driving behavior assessment. *Transportation Research Part C: Emerging Technologies*. 81172-187.

VITA

Chenxi Chen was born in Zhejiang, China. Chenxi received his bachelor's degree in Civil Engineering from Zhejiang University, China, in June 2018. He came to Missouri University of Science and Technology in September 2018 to pursue his Ph.D. degree in transportation engineering and joined the research team as a Graduate Research Assistant under the supervision of Dr. Xianbiao Hu. He published one paper titled "Analysis of Electric Vehicle Charging Behavior Patterns with Function Principal Component Analysis Approach" by Journal of Advanced Transportation. He switched to a MS degree in June 2021. In July 2021, he received his MS degree in Civil Engineering from Missouri University of Science and Technology.