Hindawi
Security and Communication Networks
Volume 2021, Article ID 9968905, 14 pages
https://doi.org/10.1155/2021/9968905

Research Article

Differentially Private Web Browsing Trajectory over Infinite Streams

Xiang Liu,[1] **Yuchun Guo**,[1] **Xiaoying Tan**,[2] **and Yishuai Chen**[1]

[1]*Beijing Jiaotong University, Beijing, China*
[2]*China Justice Big Data Institute Co., Ltd., Beijing, China*

Correspondence should be addressed to Xiang Liu; 16111017@bjtu.edu.cn

Received 24 March 2021; Accepted 21 July 2021; Published 5 August 2021

Academic Editor: Liguo Zhang

Nowadays, a lot of data mining applications, such as web traffic analysis and content popularity prediction, leverage users' web browsing trajectories to improve their performance. However, the disclosure of web browsing trajectory is the most prominent issue. A novel privacy model, named Differential Privacy, is used to rigorously protect user's privacy. Some works have applied this privacy model to spatial-temporal streams. However, these works either protect the users' activities in different places separately or protect their activities in all places jointly. The former one cannot protect trajectories that traverse multiple places; while the latter ignores the differences among places and suffers the degradation of data utility (i.e., data accuracy). In this paper, we propose a (w, n)-differential privacy to protect any spatial-temporal sequence occurring in w successive timestamps and n-range places. To achieve better data utility, we propose two implementation algorithms, named Spatial-Temporal Budget Distribution (STBD) and Spatial-Temporal RescueDP (STR). Theoretical analysis and experimental results show that these two algorithms can achieve a balance between data utility and trajectory privacy guarantee.

1. Introduction

Service providers collect users' web browsing activities and share them with many data mining applications, such as web traffic analysis and content popularity prediction. Figure 1 shows a typical data publishing system. A trusted curator aggregates users' visiting activities (a.k.a "events") and publishes the number of visits to each page periodically. The published data is a spatial-temporal stream. A user's web browsing trajectory is a sequence of timestamped visiting activities. Web browsing trajectories can reveal users' sensitive information, such as user preference. Therefore, the published stream should be protected to prevent the leakage of trajectory privacy.

A lot of excellent works [1–5] have been conducted to protect users' privacy in statistic data. Among all these works, the state-of-the-art privacy model is differential privacy [5], which can provide rigorous privacy guarantees. A differential privacy model takes a dataset as the input and outputs sanitized statistic data that remain roughly the same even if any single record in the dataset is absent. Given the output, people cannot tell whether a record is included in the original dataset or not.

Some recent works have studied the problem of releasing the spatial-temporal stream with differential privacy. According to their protection range on the time dimension, privacy models are divided into two categories: event-level privacy and user-level privacy, which are represented in Figure 1(a). Event-level privacy [6–8] protects the events on different timestamps independently. In contrast, user-level privacy [9, 10] protects users' events on all timestamps jointly. The former cannot protect the trajectories that have multiple events. The latter cannot be applied to infinite streams. The reason is that the length of a user's event sequence is infinite, which requires an infinite amount of perturbation noise. To bridge the gap between event-level privacy and user-level privacy, Kellaris et al. [11] propose the w-event differential privacy, which protects any event sequence occurring in successive w timestamps.

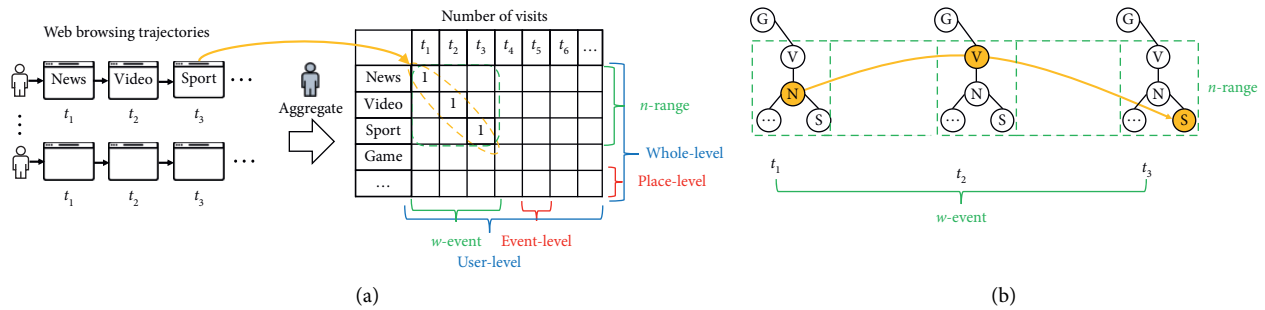


FIGURE 1: Data publishing system. (a) Event-level, user-level, place-level, and whole-level privacy and (b) a spatial-temporal window covers w timestamps and n-range places.

According to the protection range on the spatial dimension, we further categorize privacy models into two new types: place-level privacy and whole-level privacy, as shown in Figure 1(a). Place-level privacy protects the events on different places (i.e., pages) separately [10, 12, 13], whereas whole-level privacy protects the events on all places jointly [11, 14]. Since a trajectory can traverse multiple places, the place-level privacy fails to protect the trajectory privacy. However, whole-level privacy ignores the characteristics of data among different places, resulting in lower data utility (i.e., data accuracy). Specifically, the data series on different pages always have different characteristics. But whole-level privacy perturbs the data on different places with the same random noise and publishes the data on different places with the same publication cycle.

To bridge the gap between place-level privacy and whole-level privacy, it is necessary to design a new privacy model that can protect event sequences occurring in multiple places, e.g., n places.

Since the webspace is a high-dimensional space, we cannot simply arrange all pages in different rows as in Figure 1(a). Researchers usually describe the webspace as a page graph, which is shown in Figure 1(b). Intuitively, the web pages that a user browses during a short period, e.g., w successive timestamps, are related or similar. In other words, web browsing trajectories are located around a center node (e.g., topic). Therefore, in this page graph, we define that an n-range spatial window covers a center node with its $(n-1)$-hop neighbors. Given this definition, we can propose a novel privacy model, (w,n)-differential privacy, to protect any spatial-temporal sequence occurring in any spatial-temporal window of w successive timestamps and n-range paces. When $n = 1$ or N, where an N-range window covers all places, it is equivalent to place-level privacy or whole-level privacy. To provide a strong trajectory privacy guarantee, we can choose a spatial-temporal window that can cover most trajectories. Furthermore, (w,n)-differential privacy allows the data on different pages to have different perturbation noises and different publication cycles. Therefore, we can design effective implementation algorithms, which can adaptively allocate the privacy budget that controls the scale of perturbation noise and dynamically publish data according to data change. In addition to protecting web browsing trajectories, this work is also suitable for other privacy protection scenarios [15], especially non-Euclidean spaces. For example, we can use it to protect user transaction data between banks, the information flow in social networks, etc. Our contributions are summarized as follows:

(1) We demonstrate the way to construct the spatial window. Generally, the pages visited by a user within a short period, i.e., w successive timestamps, are similar or related. Therefore, we use a deep learning model to obtain the page similarity, and then we construct a page graph $\mathcal{G}$, where two similar pages are connected by an edge. In $\mathcal{G}$, we define that a spatial window of size n covers a page with its $(n-1)$-hop neighbor pages.

(2) We formulate (w,n)-differential privacy and design two implementation algorithms, Spatial-Temporal Budget Distribution (STBD) and Spatial-temporal RescueDP (STR). These two algorithms detect the data change and publish data when data change significantly. They also efficiently allocate the privacy budget for each publication to ensure that they can achieve better data utility and satisfy (w,n)-differential privacy.

(3) We compare our algorithms with two baseline algorithms, BD and RescueDP. These two algorithms belong to $(w,1)$-differential privacy and (w,N)-differential privacy, respectively. Experimental results on a real-world dataset show that the proposed two algorithms can achieve a balance between data utility and trajectory privacy guarantee.

(4) This work can be further applied to other privacy protection scenarios, especially non-Euclidean spaces, such as protecting user transaction data between banks and information flow in social networks.

The remainder of this paper is organized as follows. Section 2 surveys the related work and provides the preliminaries on differential privacy. Section 3 proposes our privacy model. Section 4 presents the STBD and STR algorithm, as well as their technical details. Section 5 presents a set of empirical studies and results. Section 6 concludes this paper.

2. Related Work and Preliminaries

2.1. Related Work

(1) Differential Privacy. Dwork et al. [5] firstly propose the differential privacy to rigorously protect individual privacy. Then, many works [16–20] apply it to the static dataset, which will not be updated in the future.

For streaming data, Dwork et al. [6] give the definitions of event-level privacy and user-level privacy and propose a "counter" algorithm to achieve event-level privacy. RTP-DMM algorithm [8] uses a Fenwick tree to reorganize data items in the stream and uses a matrix mechanism to reduce the global sensitivity. Compared with the "counter" algorithm, RTP-DMM achieves better data utility. Sun et al. [7] propose the PriStream algorithm to protect the thresholded percentile statistics under event-level privacy. Chen et al. [21] propose an event-level privacy model, PeGaSus, to simultaneously support a variety of tasks, such as counts, sliding windows, and event monitoring. To achieve user-level privacy on finite streams, Acs and Castelluccia [10] use Fourier Perturbation Algorithm [22] to perturb the data streams, and Fan and Xiong [9] propose the FAST algorithm. To bridge the gap between event-level privacy and user-level privacy, Kellaris et al. [11] propose the w-event differential privacy and its implementation algorithm, BD algorithm. Ren et al. [23] propose the average w-event differential privacy to relax the requirement of privacy budget consumed in w timestamps.

The algorithm in [10] protects the finite data series on each place separately. RescueDP [12] and AdaPub [13] independently protect the infinite data series on each place under w-event differential privacy. All these algorithms belong to the place-level privacy, which cannot rigorously protect trajectory privacy. Works [11, 14] perturb the data on different places with the same noise and publish data with the same publication cycle. Thus, they belong to whole-level privacy. Since whole-level privacy considers all places as a whole, it ignores the differences among places and achieves lower data utility. To reduce the impact of data sparsity on the spatial dimension, work [24] uses a Quadtree to group similar places together. Our previous work [25] proposes a (w, n)-differential privacy algorithm for protecting users' GPS trajectories. Because the GPS points are located in the 2D Euclidean plane, the n-range spatial window is a square area of size $n \times n$. This algorithm cannot be applied to the non-Euclidean webspace. Works [26, 27] perturb the activities within a trajectory and publish the sanitized trajectory. However, this paper aims to protect aggregated statistic data from the leakage of trajectory privacy. DADP [28] and DPCrowd [29] consider a different scenario, where the system contains multiple servers aggregating partial crowd-sourced data.

(2) Page Graph. Works [30–32] build a page graph, where the nodes are pages and the directed edges are hyperlinks. Since the hyperlinks are added by the authors, this kind of page graph cannot reflect the page relationship in users' minds. Considering users' click behaviors, Yu and Iwaihara [33] construct a click graph, where the directed and weighted edges represent click counts on each hyperlink. However, click counts often exhibit a power-law distribution [34], which makes the click graph sparse and unconnected. Inspired by recent researches on representational learning for natural language processing (NLP), Bing et al. [35] propose the session2vec algorithm extended from word2vec [36] to obtain the feature representations of pages. Their experimental results show that the cosine similarity of features can represent the relationship between any two pages.

2.2. Differential Privacy.

Differential privacy is a rigorous privacy model proposed by Dwork et al. [5]. Intuitively, given a randomized mechanism K, differential privacy requires that the output of K is insensitive to the change of a single record in the input of K. The formal definition of differential privacy is given as follows.

Definition 1. ε-differential privacy [5].

A randomized mechanism K satisfies ε-differential privacy, if for any two datasets $\mathscr{D}$ and $\mathscr{D}\prime$ differing on at most one record, and for any possible output $O \in \mathrm{Range}(K)$,

$$\Pr[K(\mathscr{D}) = O] \leq e^{\varepsilon} \times \Pr[K(\mathscr{D}\prime) = O], \tag{1}$$

where the probability is taken over the randomness of K.

The parameter ε controls the degree of privacy protection. A lower value of ε offers a stronger privacy guarantee. Two datasets $\mathscr{D}$ and $\mathscr{D}\prime$ that differ on at most one record are called neighboring datasets.

Laplace mechanism is the most commonly used mechanism that satisfies ε-differential privacy. Its main idea is to add Laplace random noise to the statistic data.

Definition 2. Sensitivity [5].

For any function $q: \mathscr{D} \longrightarrow \mathbb{R}^d$, the sensitivity of q is $\Delta(q) = \max_{\mathscr{D},\mathscr{D}\prime} \|q(\mathscr{D}) - q(\mathscr{D}\prime)\|_1$ for any two neighboring datasets $\mathscr{D}$ and $\mathscr{D}\prime$.

The sensitivity of function q is the maximum L1 distance between function outputs of any two neighboring datasets $\mathscr{D}$ and $\mathscr{D}\prime$.

Theorem 1. *Laplace mechanism [5].*

For any function $q: \mathscr{D} \longrightarrow \mathbb{R}^d$, the mechanism K

$$K(\mathscr{D}) = q(\mathscr{D}) + < \mathscr{L}_1\left(0, \frac{\triangle(q)}{\varepsilon}\right), \ldots, \mathscr{L}_d\left(0, \frac{\triangle(q)}{\varepsilon}\right) >, \tag{2}$$

satisfies ε-differential privacy, if $\mathscr{L}_i(0, \triangle(q)/\varepsilon)$ are i.i.d zero-mean Laplace noises with scale $(\triangle(q)/\varepsilon)$.

A smaller value of ε offers a larger scale of Laplace noise and a stronger privacy guarantee. Differential privacy maintains two composition properties, which are given as follows.

Theorem 2. *Sequential Composition [37].*

Let $\{K_1, K_2, \ldots, K_T\}$ be a set of mechanisms, and each K_t provides ε_t-differential privacy, where $t \in [1, T]$. Let K be another mechanism that executes $K_1(\mathscr{D})$, $K_2(\mathscr{D})$, …,

$K_T(\mathscr{D})$ independently and returns the outputs of these mechanisms. Then, K satisfies $\sum_{t=1}^{T}\varepsilon_t$-differential privacy.

Theorem 2 allows us to distribute ε among T independent mechanisms. Therefore, ε is called the privacy budget.

Theorem 3. *Parallel Composition [37].*

Let $\{D^1, D^2, \ldots, D^M\}$ be a set of disjoint subsets of dataset $\mathscr{D}$. Let $\{K^1, K^2, \ldots, K^M\}$ be a set of mechanisms, and each K^m provides ε^m-differential privacy, where $m \in [1, M]$. Let K be another mechanism that executes $K^1(D^1)$, $K^2(D^2)$, $\ldots$, $K^M(D^M)$ independently and returns the outputs of these mechanisms. Then, K satisfies $\max_{1\le m\le M}\varepsilon^m$-differential privacy.

If a dataset is partitioned into disjoint subsets, and each part is protected under differential privacy, then the ultimate privacy guarantee depends on the worst of the guarantees.

2.3. w-Event Differential Privacy. w-event differential privacy [11] (short for w-event ε-differential privacy) is used to protect any event sequence occurring in w timestamps. An infinite stream is denoted by $S = [\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_t, \ldots]$, where $\mathscr{D}_t$ denotes the dataset on timestamp t. A prefix of stream S of length T is denoted by $S_T = [\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_T]$. The definition of two w-neighboring prefixes is given as follows.

Definition 3. w-neighboring [11].

Let w be a positive integer. Two stream prefixes S_T and S_T' are w-neighboring if

(1) for each $\mathscr{D}_t, \mathscr{D}_t'$ such that $t \in [1, T]$ and $\mathscr{D}_t \neq \mathscr{D}_t'$, it holds that $\mathscr{D}_t, \mathscr{D}_t'$ are neighboring, and

(2) for each $\mathscr{D}_t, \mathscr{D}_t', \mathscr{D}_\tau, \mathscr{D}_\tau'$ with $1 \le t < \tau \le T$, $\mathscr{D}_t \neq \mathscr{D}_t'$ and $\mathscr{D}_\tau \neq \mathscr{D}_\tau'$, it holds that $\tau - t + 1 \le w$.

Concretely, if S_T, S_T' are w-neighboring prefixes, then (1) their pairwise elements $\mathscr{D}_t$ and $\mathscr{D}_t'$, where $t \in [1, T]$, are the same or neighboring, and (2) all neighboring element pairs can fit in a window of w timestamps.

Definition 4. w-event ε-differential privacy [11].

A mechanism K satisfies w-event ε-differential privacy, if, for any two w-neighboring stream prefixes S_T, S_T', any possible output $O \in \mathrm{Range}(K)$, and any $T > 0$,

$$\Pr[K(S_T) = O] \le e^{\varepsilon} \times \Pr[K(S_T') = O]. \tag{3}$$

According to Theorem 2, to satisfy w-event differential privacy, a privacy mechanism can distribute the privacy budget ε to several mechanisms that protect the data on different timestamps independently.

Theorem 4. *Implementation of w-event differential privacy [11].*

Let K be a mechanism that takes a stream prefix $S_T = [\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_T]$ as the input and outputs $O_T = [\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T] \in Range(K)$. Suppose that we can decompose K into T mechanisms $K_1, K_2, \ldots, K_T$, such that $\mathbf{o}_t = K_t(\mathscr{D}_t)$, where $t \in [1, T]$, and each K_t satisfies ε_t-differential privacy independently. Then, K satisfies w-event ε-differential privacy if

$$\sum_{t=\tau-w+1}^{\tau}\varepsilon_t \le \varepsilon, \quad \forall \tau \in [1, T]. \tag{4}$$

2.4. Dataset. In this paper, we use a real-world dataset, the WorldCup98 dataset [38]. This dataset contains 1.3 billion logs of the requests to the FIFA 1998 website from 2.76 million clients from April 30, 1998, to July 26, 1998. Each log contains client ID, URL, and timestamp. We choose 10000 most popular URLs from June 15, 1998, to June 30, 1998, and publish the number of visits every 30 minutes.

3. Privacy Model

We study the problem of how to publish the number of visits to each web page while protecting user browsing trajectories. A web browsing trajectory is defined as a sequence of web pages visited by a user with timestamps. Generally, the pages visited by a user during a short period are most likely to be similar or related. In other words, users' trajectories can fit into a spatial-temporal window that covers a group of similar pages and several successive timestamps. Therefore, to provide a strong trajectory privacy guarantee, we propose a privacy model that can protect any possible spatial-temporal sequence within a spatial-temporal window. In this section, we first construct a page graph and introduce the idea of the spatial-temporal window. Then, we formulate our privacy model.

3.1. Page Graph. Inspired by [35, 36], we use the word2vec algorithm to obtain the similarity between pages. The source code is available at [39]. We use the logs on June 15, 1998, as the training data. The embedding size, context window, and learning rate are set to 128, 3, and 0.001, respectively. After training 100000 steps with Adam optimizer [40], we obtain the feature vectors of pages. If the cosine similarity of two pages is higher than ρ, we connect them with an unweighted and undirected edge. We also connect each page to the other two pages, which are the top 2 similar pages among all pages to ensure that we can obtain a connected graph.

If ρ is too large, the generated graph will become very sparse. On the contrary, if ρ becomes too small, pages will directly connect to each other and the graph will become very dense. We test ρ from 0.1 to 0.8 and find that when $\rho = 0.4$, pages will have a proper distance to other pages.

We use $\mathscr{G}$ to denote this page graph, which contains 10000 nodes and 28594 edges. The maximum and minimum node degree are 96 and 2, respectively. Figure 2 shows the degree distribution in the log-log plot. We can observe that the distribution approximately follows the power-law [41], and the exponent of the power-law is 2.46.

The distance between two connected pages is 1, and the distance between two unconnected pages is the length of the shortest path in $\mathscr{G}$. As shown in Figure 3, the distance between page 1 and page 2 is 1. Since the shorted path from

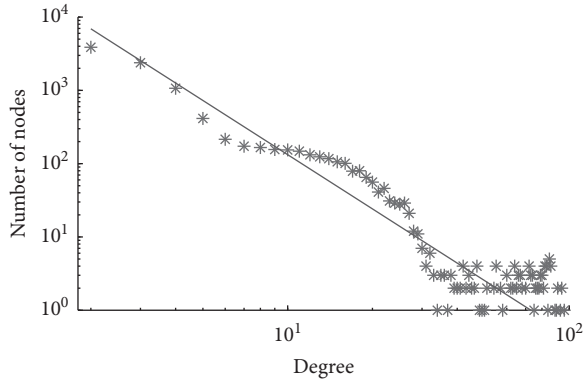
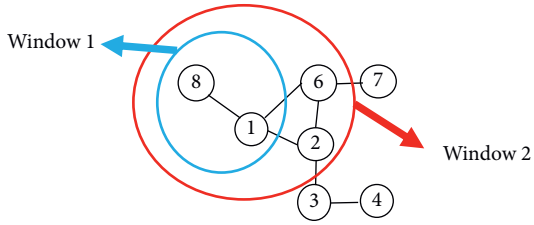


Figure 2: Degree distribution.



Trajectory A: 8-2-6
Trajectory B: 8-1-2-3-4

Figure 3: Example of the page graph.

page 1 to page 4 is $1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4$, the distance between page 1 and page 4 is 3. We also call page 1 as a 1-hop neighborhood of page 2 and a 3-hop neighborhood of page 4. Note that a user can visit the pages that are not directly connected to each other, and then the trajectory will not be a continuous path in $\mathcal{G}$. Given the distance between pages, we define the spatial range of a trajectory as follows.

Definition 5. The spatial range of a web browsing trajectory.

Given a graph $\mathcal{G}$ and a web browsing trajectory, the center of this trajectory is a page, whose greatest distance to all pages in the trajectory is as small as possible. This distance is the spatial range of the trajectory.

Trajectory center can be regarded as the "topic" of this trajectory, and the spatial range represents the "topic scope." The trajectory center is not necessarily included in the trajectory. For example, in Figure 3, the greatest distance from page 1 to pages 8, 2, and 6 is 1, while the greatest distance from other pages to pages 8, 2, and 6 is larger than 1. Therefore, the center of trajectory A is page 1, and its spatial range is 1. Similarly, the center of trajectory B is page 2, and its spatial range is 2.

Given a page, we define that a spatial window of size $\mathbf{n}$ ($\mathbf{n}$-range spatial window) covers the page with its $(n-1)$-hop neighborhoods. As shown in Figure 3, the spatial window 1 takes page 8 as the center and covers page 8 and page 1. Similarly, the spatial window 2 takes page 1 as the center and covers page 1, page 2, page 6, and page 8. If the spatial range of a trajectory is n, this trajectory can be covered by a spatial window of size $n+1$. In $\mathcal{G}$, a spatial window of size 9 can cover all pages. A temporal window of size $\mathbf{w}$ covers w successive timestamps. We use (w, n) to denote the size of a spatial-temporal window, where the first element is the temporal size, and the second element is the spatial size.

We randomly select 50,000 users and analyze the spatial range of their trajectories during w timestamps. The interval between two consecutive timestamps is 30 minutes. The results are shown in Table 1. When $w = 120$, the spatial range of 88.7% of the trajectories is less than 5. Thus, a spatial-temporal window of 120 successive timestamps and 6-range places can cover most trajectories.

3.2. (w, n)-Differential Privacy. Without loss of generality, we assume that there are M pages hosted on a server. A user can visit at most $l \ll M$ pages per timestamp. A trusted curator collects users' visiting activities and creates a dataset $\mathcal{D}_t$ at each timestamp t. This infinite stream is denoted by $S = [\mathcal{D}_1, \mathcal{D}_2, \ldots]$. We define a stream prefix of S as $S_T = [\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T]$. Dataset $\mathcal{D}_t$ can be partitioned into M disjoint subsets $D_t^1, D_t^2, \ldots, D_t^M$, where D_t^m contains users' visiting activities to page m at timestamp t. A query function q asks for the number of visits on each page. Then, $q(\mathcal{D}_t) = [q(D_t^1), q(D_t^2), \ldots, q(D_t^M)] = \mathbf{r}_t = [r_t^1, r_t^2, \ldots, r_t^M]$, where r_t^m denotes the number of visits on page m at timestamp t. (w, n)-differential privacy protects any spatial-temporal sequence occurring in any spatial-temporal window of w timestamps and n-range places. The formal definition is given as follows.

Definition 6. (w, n)-neighboring.

Let w and n be two positive integers. Two spatial-temporal stream prefixes S_T and S_T' are (w, n)-neighboring if

(1) each pair of D_t^m, $D_t'^m$ is neighboring or the same and

(2) all neighboring pairs D_t^m, $D_t'^m$ can fit into a spatial-temporal window of size (w, n).

The above definition means that two (w, n)-neighboring stream prefixes differ on a single spatial-temporal sequence that can fit in a spatial-temporal window of size (w, n). (w, n)-differential privacy ensures that the output results of any two (w, n)-neighboring stream prefixes are indistinguishable.

Definition 7. w-event n-range ε-differential privacy.

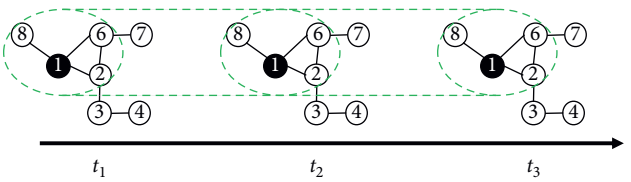
A mechanism K satisfies w-event n-range ε-differential privacy (short for (w, n)-differential privacy), if for any two (w, n)-neighboring prefixes S_T and S_T', any possible output $O \in \text{Range}(K)$, and any $T > 0$,

$$\Pr[K(S_T) = O] \leq e^{\varepsilon} \times \Pr[K(S_T') = O]. \tag{5}$$

This model can be widely used in various scenarios, where trajectories need to be protected. To implement this privacy model, we can decompose mechanism K into submechanisms K_t^m, and each K_t^m protects dataset D_t^m under ε_t^m-differential privacy independently. Then, we ensure that the total privacy budget inside any spatial-temporal window of size (w, n) is less than ε. Figure 4 gives an example. A

Table 1: Spatial range of trajectories during w timestamps.

% n	w				
	40	80	120	160	200
0	3.37	3.07	2.85	2.67	2.56
1	14.05	12.87	11.96	11.27	10.74
2	26.49	24.48	22.78	21.46	20.47
3	48.11	44.46	41.61	39.41	37.81
4	84.13	78.57	74.57	71.14	68.75
5	96.16	92.03	88.7	85.53	83.4
6	99.21	97.26	95.19	93.57	92.41
7	100	100	99.99	99.98	99.93
8	100	100	100	100	100



(a)

Page \ Time	t_1	t_2	t_3	…
1	r_1^1	r_2^1	r_3^1	
8	r_1^8	r_2^8	r_3^8	
2	r_1^2	r_2^2	r_3^2	
6	r_1^6	r_2^6	r_3^6	
…				
M				

(b)

Page \ Time	t_1	t_2	t_3	…
1	ε_1^1	ε_2^1	ε_3^1	
8	ε_1^8	ε_2^8	ε_3^8	
2	ε_1^2	ε_2^2	ε_3^2	
6	ε_1^6	ε_2^6	ε_3^6	
…				
M				

(c)

Figure 4: Example of the spatial-temporal window and privacy budget. (a) The spatial-temporal window, (b) number of visits, and (c) privacy budget.

spatial-temporal window of size $(3, 2)$ covers the data on page 1, page 2, page 6, and page 8 from t_1 to t_3. Since $r_t^1, r_t^2, r_t^6, r_t^8$ belong to disjoint datasets, the ultimate privacy guarantee is $\max(\varepsilon_t^1, \varepsilon_t^2, \varepsilon_t^6, \varepsilon_t^8)$-differential privacy based on Theorem 3. According to Theorem 2, the privacy guarantee inside this window is $\sum_{t=1}^{3}(\max(\varepsilon_t^1, \varepsilon_t^2, \varepsilon_t^6, \varepsilon_t^8))$-differential privacy. If $\sum_{t=\tau-2}^{\tau}(\max_{m\in win}\varepsilon_t^m) <= \varepsilon$ for any $\tau \in [1, T]$, and any 2-range spatial window *win*, mechanism K satisfies (3,2)-differential privacy.

Theorem 5. *Implementation of (w, n)-differential privacy.*

Let K be a mechanism that takes a spatial-temporal stream prefix $S_T = [\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_T]$ as the input and outputs $O_T = [\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T] \in Range(K)$, where $\mathscr{D}_t = [D_t^1, D_t^2, \ldots, D_t^M]$ and $\mathbf{o}_t = [o_t^1, o_t^2, \ldots, o_t^M]$. Suppose that we can decompose K into $T \times M$ mechanisms K_t^m, where $t \in [1, T], m \in [1, M]$, such that $o_t^m = K_t^m(D_t^m)$, and each K_t^m satisfies ε_t^m-differential privacy independently. Then, K satisfies (w, n)-differential privacy if

$$\sum_{t=\tau-w+1}^{\tau}\left(\max_{m\in win}\varepsilon_t^m\right)\le\varepsilon,\forall\tau\in[1,T],\quad \forall win\in\{\text{n-range spatial windows}\}. \tag{6}$$

4. Algorithms

4.1. Spatial-Temporal Budget Distribution. The structure of the Spatial-Temporal Budget Distribution (STBD) algorithm is shown in Figure 5. STBD consists of three components: evaluation, perturbation, and budget allocation. The evaluation component evaluates the data change and decides whether to publish new data at each timestamp. The perturbation component perturbs the data with Laplace noise and publishes the sanitized data. The budget allocation component distributes the privacy budget to each component. Our STBD algorithm is extended from the Budget Distribution (BD) algorithm [11]. STBD uses a new evaluation component to capture the data change on small page groups and uses a new budget allocation component to satisfy (w,n)-differential privacy. Algorithm 1 shows the pseudocode of the STBD algorithm.

4.1.1. Evaluation and Perturbation. Publishing data consumes the privacy budget, while the total privacy budget inside the spatial-temporal window is constant. If we publish data on every timestamp, the privacy budget for each publication becomes small, which results in significant perturbation noise. Conversely, if we skip some publications and approximate current data to previously published data, the perturbation noise decreases, while the approximation error increases. The approximation error is calculated as follows:

$$a\_\mathrm{err}_t^m=\left|r_t^m-o_{t-1}^m\right|+\mathscr{L}\left(0,\frac{l}{\alpha_t^m}\right), \tag{7}$$

r_t^m denotes the number of visits to page m at timestamp t. o_{t-1}^m denotes the previously published data on page m at timestamp $t-1$. Because the approximation error contains the sensitive data r_t^m, we should add Laplace noise to it. Since one user can visit at most l pages per timestamp, the sensitivity of the approximation error is l. α_t^m is the privacy budget for evaluation. $\mathscr{L}(0,l/\alpha_t^m)$ is called the evaluation noise.

The perturbation component adds perturbation noise to data r_t^m. The sanitized output data is given as follows:

$$o_t^m=r_t^m+\mathscr{L}\left(0,\frac{l}{\beta_t^m}\right), \tag{8}$$

β_t^m is the privacy budget for perturbation. Because the perturbation noise is a random variable, we define the perturbation error as the expectation of the absolute error between r_t^m and o_t^m. The perturbation error is calculated as follows:

$$p\_\mathrm{err}_t^m=E\left(\left|r_t^m-o_t^m\right|\right)=E\left(\left|\mathscr{L}\left(0,\frac{l}{\beta_t^m}\right)\right|\right)=\frac{l}{\beta_t^m}. \tag{9}$$

The evaluation component compares these two errors and decides whether to publish new data. If the data change dramatically, the approximation error becomes larger than the perturbation error. Then, the perturbation component consumes the perturbation budget and publishes new data. Otherwise, it skips this publication and saves the perturbation budget.

Unfortunately, the evaluation noise can easily cause the evaluation component to make the bad decisions. To reduce the evaluation noise, we cluster all pages into X groups and calculate the mean approximate error in each group instead of the approximate error on each page. The mean approximate error is calculated as follows:

$$\overline{a\_\mathrm{err}_t^x}=\frac{1}{\mathrm{size}(x)}\sum_{m\in x}\left|r_t^m-o_{t-1}^m\right|+\mathscr{L}\left(0,\frac{l/\mathrm{size}(x)}{\alpha_t^x}\right), \tag{10}$$

where $\mathrm{size}(x)$ stands for the number of pages in group $x\in[1,X]$. The sensitivity of the mean approximate error is $l/\mathrm{size}(x)$, and α_t^x is the evaluation budget for each group.

We can see that the evaluation noise in equation (10) has a smaller variance compared to equation (7). However, the mean approximate error also has a shortcoming that the data change on a single page could be ignored. BD algorithm calculates the mean approximate error on all pages, which ignores many small changes among pages and results in a lower data utility. STBD clusters pages into several groups to strike a balance between reducing evaluation noise and capturing data change. The pseudocode of the evaluation and the perturbation are shown in Lines 2–18, Algorithm 1. The evaluation component first calculates the approximation error (Lines 2–7). Then, it calculates the perturbation error (Lines 8–10). If the perturbation error is less than the approximation error, the perturbation component adds perturbation noise to r_τ^m and publishes it (Lines 11–13). Otherwise, the perturbation component repeats the previously published data $o_{\tau-1}^m$ and saves the perturbation budget (Lines 14–17).

4.1.2. Budget Allocation in STBD. The budget allocation component distributes half of the privacy budget ε to the evaluation component, and the other half to the perturbation component. The evaluation budget is evenly distributed to w timestamps. Therefore, the evaluation budget $\alpha_t^x=(\varepsilon/2)/w$ for any timestamp t and any page group x. The perturbation budget for each publication is allocated by an exponential decay method, which allocates half of the remaining budget to each publication. Algorithm 2 presents the pseudocode of budget allocation.

We use Figure 6 as an example to illustrate Algorithm 2. Windows 1 and 2 are two 2-range spatial windows that cover page 8. Windows 1 and 2 take pages 8 and 1 as the center, respectively. The temporal length of windows 1 and 2 is w. Considering window 1, the perturbation budget consumed at timestamp t is calculated as follows:

$$\phi_t^{win1}=\max_{m\in win1}\beta_t^m. \tag{11}$$

Input: $\mathbf{r}_\tau, \mathbf{o}_{\tau-1}$
Output: $\mathbf{o}_\tau, (\beta_\tau^1, \ldots, \beta_\tau^M)$
(1) $\alpha_\tau^x, \beta_\tau^m \leftarrow$ Budget Allocation
//Evaluation
(2) for group x do
(3) $\overline{a\_\text{err}_\tau^x} = (1/\text{size}(x))\sum_{m\in x}|r_\tau^m - o_{\tau-1}^m| + \mathscr{L}(0, (l/\text{size}(x))/\alpha_\tau^x)$
(4) for page $m \in x$ do
(5) $a\_\text{err}_\tau^m = \overline{a\_\text{err}_\tau^x}$
(6) end for
(7) end for
(8) for page m do
(9) $p\_\text{err}_\tau^m = l/\beta_\tau^m$
(10) end for
//Perturbation
(11) for page m do
(12) if $a\_\text{err}_\tau^m > p\_\text{err}_\tau^m$ then
(13) $o_\tau^m = r_\tau^m + \mathscr{L}(0, l/\beta_\tau^m)$
(14) else
(15) $o_\tau^m = o_{\tau-1}^m$
(16) $\beta_\tau^m = 0$
(17) end if
(18) end for
(19) return $\mathbf{o}_\tau, (\beta_\tau^1, \ldots, \beta_\tau^M)$

Algorithm 1: STBD algorithm.

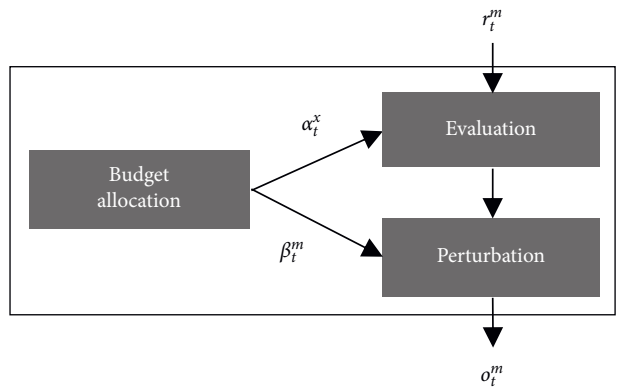
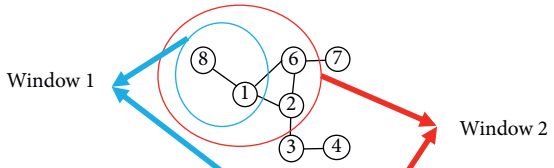


Figure 5: The structure of the STBD algorithm.

Input: $(\beta_1^1, \ldots, \beta_{\tau-1}^M), \varepsilon, w, n$
Output: $\alpha_\tau^x, \beta_\tau^m$
(1) for group x do
(2) $\alpha_\tau^x = (\varepsilon/2)/w$
(3) end for
(4) for n-range window win do
(5) $\phi_t^{win} = \max_{m\in win} \beta_t^m$
(6) $\psi_\tau^{win} = \varepsilon/2 - \sum_{t=\tau-w+1}^{\tau-1} \phi_t^{win}$
(7) end for
(8) for node m do
(9) $\eta_\tau^m = \min_{win\in m} \psi_\tau^{win}$
(10) $\beta_\tau^m = \eta_\tau^m/2$
(11) end for
(12) return $\alpha_\tau^x, \beta_\tau^m$

Algorithm 2: Budget Allocation in STBD.



Page \ Time	$\tau - w + 1$	…	$\tau - 1$
1	$\beta^1_{\tau-w+1}$	…	$\beta^1_{\tau-1}$
8	$\beta^8_{\tau-w+1}$	…	$\beta^8_{\tau-1}$
2	$\beta^2_{\tau-w+1}$	…	$\beta^2_{\tau-1}$
6	$\beta^6_{\tau-w+1}$	…	$\beta^6_{\tau-1}$

Window \ Time	$\tau - w + 1$	…	$\tau - 1$
1	$\phi^{win1}_{\tau-w+1}$	…	$\phi^{win1}_{\tau-1}$
2	$\phi^{win2}_{\tau-w+1}$	…	$\phi^{win2}_{\tau-1}$

Figure 6: Illustration of exponential decay method.

The remaining budget of window 1 at timestamp τ is

$$\psi^{win1}_{\tau} = \frac{\varepsilon}{2} - \sum_{t=\tau-w+1}^{\tau-1} \phi^{win1}_{t}. \tag{12}$$

Similarly, the remaining budget of window 2 at timestamp τ is

$$\psi^{win2}_{\tau} = \frac{\varepsilon}{2} - \sum_{t=\tau-w+1}^{\tau-1} \phi^{win2}_{t}. \tag{13}$$

To ensure that the perturbation budget inside windows 1 and 2 is no more than $\varepsilon/2$, the remaining budget for page 8 at timestamp τ is

$$\eta^{8}_{\tau} = \min_{win \ni 8} \psi^{win}_{\tau}. \tag{14}$$

We allocate half of the remaining budget for the perturbation budget, while reserving the other half for future publications. The perturbation budget is described as follows:

$$\beta^{8}_{\tau} = \frac{1}{2}\eta^{8}_{\tau}. \tag{15}$$

4.1.3. Privacy Analysis

Theorem 6. *STBD algorithm satisfies the (w, n)-differential privacy.*

Proof. Given a page m belonging to group x, the evaluation component calculates the mean approximate error on group x and the perturbation error on page m. The perturbation error does not contain sensitive information. The sensitivity of the mean approximate error on group x is $l/\text{size}(x)$. Based on Theorem 1, evaluation component adds $\mathscr{L}(0, (l/\text{size}(x))/\alpha^x_t)$ to the mean approximate error, so it satisfies α^x_t-differential privacy.

The perturbation component adds $\mathscr{L}(0, l/\beta^m_t)$ to the data r^m_t and the sensitivity of r^m_t is l. According to Theorem 1, the perturbation component satisfies β^m_t-differential privacy.

The evaluation and perturbation protect the data r^m_t independently. Therefore, r^m_t is protected under $(\alpha^x_t + \beta^m_t)$-differential privacy based on Theorem 2. Budget allocation guarantees that $\sum_{t=\tau-w+1}^{\tau} \max_{m \in win}(\alpha^x_t + \beta^m_t) \le \varepsilon$ for any n-range window win and any τ, so STBD algorithm satisfies (w, n)-differential privacy. □ □

4.2. Spatial-Temporal RescueDP. Spatial-Temporal RescueDP (STR) extends the RescueDP [12] to implement (w, n)-differential privacy. The structure of the STR algorithm is shown in Figure 7. It contains four components: sampling, perturbation, filtering, and budget allocation. The sampling component adaptively selects the timestamps to publish new data according to the data change and the remaining budget. The perturbation component adds Laplace noise to the statistic data at sampling timestamps. The filtering component uses the Kalman Filter [9, 12] to improve data utility. The budget allocation component allocates the privacy budget for each publication. The difference between STR and RescueDP is that STR uses a new budget allocation component to achieve (w, n)-differential privacy. Algorithm 3 outlines the steps in STR.

4.2.1. Sampling. The sampling component uses the PID control algorithm [12] to predict the data change. The PID error is defined as follows:

$$\begin{aligned} E^m_{t_n} &= \left| o^m_{t_n} - o^m_{t_{n-1}} \right|, \\ \delta^m_{t_n} &= K_p E^m_{t_n} + K_g \frac{\sum_{u=n-\pi-1}^{n} E^m_{t_u}}{\pi} + K_d \frac{E^m_{t_n}}{t_n - t_{n-1}}. \end{aligned} \tag{16}$$

The subscript t_n indicates the n-th sampling timestamp. $E^m_{t_n}$ is the feedback error [12], which calculates the absolute error between the published data at the current sampling timestamp and the last sampling timestamp. PID error $\delta^m_{t_n}$ contains three parts: (1) $K_p E^m_{t_n}$ is the proportional error standing for the current error; (2) $K_g \sum_{u=n-\pi-1}^{n} E^m_{t_u}/\pi$ is the integral error standing for the sum of past π feedback errors; (3) $K_d E^m_{t_n}/(t_n - t_{n-1})$ is the derivative error standing for the future error.

When the data change rapidly, the PID error becomes larger. To reduce approximation error, the sampling interval should become smaller. Meanwhile, if the remaining budget is small, the sampling interval should become larger to reduce the perturbation noise. Therefore, the sampling interval is calculated as follows:

Input: $\mathbf{r}_\tau$
Output: $\mathbf{o}_\tau$
(1) Find the set of sampling pages
(2) Budget Allocation component allocates privacy budget to each page
(3) Perturbation component adds Laplace noise to r_τ^m
(4) Filtering component uses Kalman filter to reduce the noise
(5) Sampling component calculates the next sampling interval

Algorithm 3: STR algorithm.

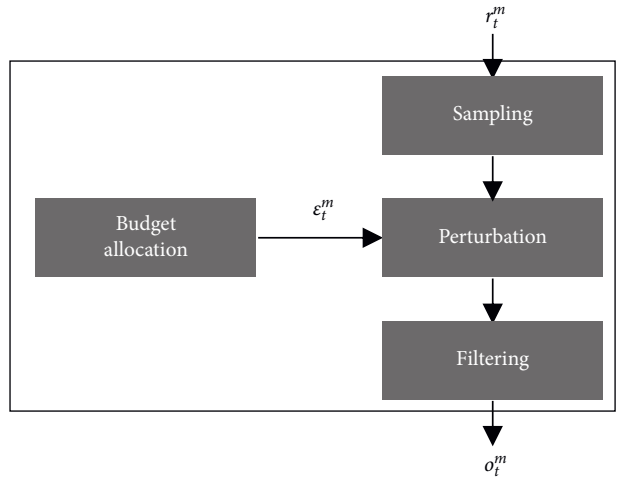


Figure 7: The structure of the STR algorithm.

$$I_{t_n}^m = \max\left(1, I_{t_{n-1}} + \theta\left(1 - \left(\delta_{t_n}^m \cdot \eta_{t_n}^m\right)^2\right)\right), \tag{17}$$

$I_{t_n}^m$ is the sampling interval of page m at timestamp t_n. $\eta_{t_n}^m$ is the remaining budget for page m at timestamp t_n. The parameter θ controls the changing rate of the sampling interval.

4.2.2. Perturbation and Filtering. The perturbation component adds the Laplace random noise into data r_t^m. The perturbation budget is denoted by ε_t^m. The sanitized output data is calculated as follows:

$$o_t^m = r_t^m + \mathscr{L}\left(0, \frac{l}{\varepsilon_t^m}\right). \tag{18}$$

The filtering component uses the Kalman Filter to reduce the perturbation noise. Since it only accesses the sanitized output data, it does not leak sensitive information. More details of Kalman Filter can be found in [9, 12].

4.2.3. Budget Allocation in STR. Budget allocation in STR allocates the perturbation budget based on the sampling interval. If the sampling interval is small, the number of sampling timestamps could be more. Therefore, we should allocate a small portion of the remaining budget to the current timestamp and leave more budget to the future. The pseudocode is shown in Algorithm 4. It first calculates the remaining budget for each page (Lines 1–6). Then, it calculates the proportion p of the remaining budget allocated to the perturbation component (Line 7). The $p_{\max}$ and $\varepsilon_{\max}$ are used to avoid consuming too much budget at a single publication (Lines 7-8).

4.2.4. Privacy Analysis

Theorem 7. *STR algorithm satisfies the (w, n)-differential privacy*

Proof. The perturbation component accesses the sensitive data r_t^m, while the other three components do not. The perturbation component adds $\mathscr{L}(0, l/\varepsilon_t^m)$ to the data r_t^m and the sensitivity of r_t^m is l. According to Theorem 1, the perturbation component satisfies ε_t^m-differential privacy. The budget allocation component guarantees that $\sum_{t=\tau-w+1}^{\tau} \max_{m \in win} \varepsilon_t^m < \varepsilon$ for any n-range window win and any τ. Therefore, STR algorithm satisfies (w, n)-differential privacy. □ □

5. Experimental Evaluation

Baseline models. In this section, we compare STBD and STR with two state-of-the-art algorithms, BD [11] and RescueDP [12]. BD is a (w, N)-differential privacy algorithm, and RescueDP is a $(w, 1)$-differential privacy

Input: $(\varepsilon_1^1, \ldots, \varepsilon_{\tau-1}^M), \varepsilon, w, n$
Output: ε_τ^m
(1) for n-range window win do
(2) $\quad \phi_t^{win} = \max_{m \in win} \varepsilon_t^m$
(3) $\quad \psi_\tau^{win} = \varepsilon - \sum_{t=\tau-w+1}^{\tau-1} \phi_t^{win}$
(4) end for
(5) for node m do
(6) $\quad \eta_\tau^m = \min_{win \in m} \psi_\tau^{win}$
(7) $\quad p_\tau^m = \min(s \cdot ln(I_\tau^m + 1), p_{\max})$
(8) $\quad \varepsilon_\tau^m = \min(p_\tau^m \cdot \eta_\tau^m, \varepsilon_{\max})$
(9) end for
(10) return ε_τ^m

Algorithm 4: Budget allocation in STR.

Table 2: STR parameters.

K_p	0.9	θ	10
K_g	0.1	s	0.2
K_d	0	$p_{\max}$	0.6
π	3	$\varepsilon_{\max}$	0.2ε

algorithm. Since the Uniform algorithm [11], which allocates the same budget to every timestamp, performs much worse than baseline models, we do not include it in the following experiments. All algorithms are written by Python on a computer with Intel Core i7-8700 CPU. We run each experiment 50 times and report the average results.

Utility metrics. We use the Mean Absolute Error (MAE) and Mean Relative Error (MRE) as the utility metrics to measure the performance of algorithms.

$$\begin{aligned} \mathrm{MAE}(R_T, O_T) &= \frac{1}{T \cdot M} \sum_{t=1}^{T} \sum_{m=1}^{M} \left| r_t^m - o_t^m \right|, \\ \mathrm{MRE}(R_T, O_T) &= \frac{1}{T \cdot M} \sum_{t=1}^{T} \sum_{m=1}^{M} \frac{\left| r_t^m - o_t^m \right|}{\max(\gamma, r_t^m)}, \end{aligned} \tag{19}$$

where $R_T = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_T]$ denotes the real number of visits, and $O_T = [\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T]$ is the sanitized output. γ is used to avoid the special case when r_t^m is 0. We set γ to 0.1% of $\sum_{m=1}^{M} r_t^m$, which is the same as [12].

In the STBD algorithm, we use the Louvain algorithm [42] to cluster pages into 39 groups. To fairly compare with RescueDP, STR uses the same parameters in [12], which are shown in Table 2. We refer readers to [12] for more details. We set l to 20.

Utility vs. ε. Figure 8 shows the MAE and MRE of four algorithms when ε changes from 0.2 to 1.0 under the condition that $w = 120$ and $n = 6$. Because larger privacy budget results in smaller Laplace random noise, the MAE and MRE of the four algorithms decrease when ε becomes larger. RescueDP and STR outperform BD and STBD. Since BD and STBD only allocate half of the privacy budget for the perturbation component, they get larger perturbation noise. Furthermore, RescueDP and STR use Kalman Filter to reduce the random noise.

Utility vs. w. Figure 9 shows the MAE and MRE of four algorithms when w changes from 40 to 200 under the condition that $\varepsilon = 1$ and $n = 6$. The MAE and MRE of BD and STBD increase when w becomes larger. This is because the evaluation budget becomes smaller when w becomes larger, which results in larger evaluation noise and more bad decisions. Although a larger w may result in more publications and less perturbation budget for each publication, STR and RescueDP increase the sampling interval when the remaining budget becomes small. Therefore, the perturbation budget cannot be too small, and the performance of STR and RescueDP is relatively stable when w changes.

Utility vs. n. Figure 10 shows the MAE and MRE of four algorithms when n changes from 4 to 9 under the condition that $\varepsilon = 1$ and $w = 120$. Because the n-range spatial window constrains the maximum perturbation budget consumed on each page, the performance of STR is worse than RescueDP. We can also observe that the performance of STBD and STR is stable as n increases. The reason is that STBD and STR can prevent performance degradation when n becomes too large. A much larger n brings a stronger budget constraint on the spatial dimension and can lead to greater perturbation noise. However, STBD and STR evaluate the data changes and skip some publications to save the budget for future publications. Thus, the performance degradation is not obvious. On the contrary, if we publish the data at every timestamp, data utility will significantly decrease as n increases.

Utility vs. Trajectory privacy guarantee. Comparing four algorithms in Figures 8–10, we can observe that BD gets the worst data utility. However, BD protects any spatial-temporal sequence within w successive timestamps, so it provides the strongest trajectory privacy guarantee. RescueDP

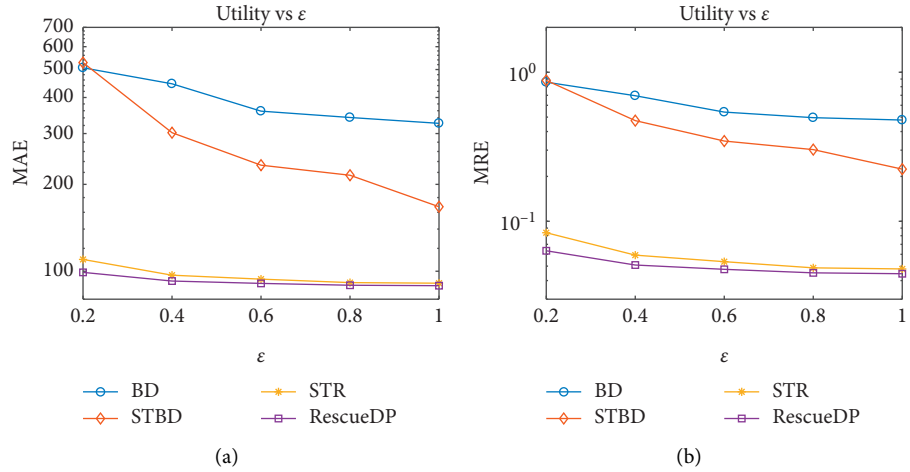


Figure 8: Utility vs. ε.

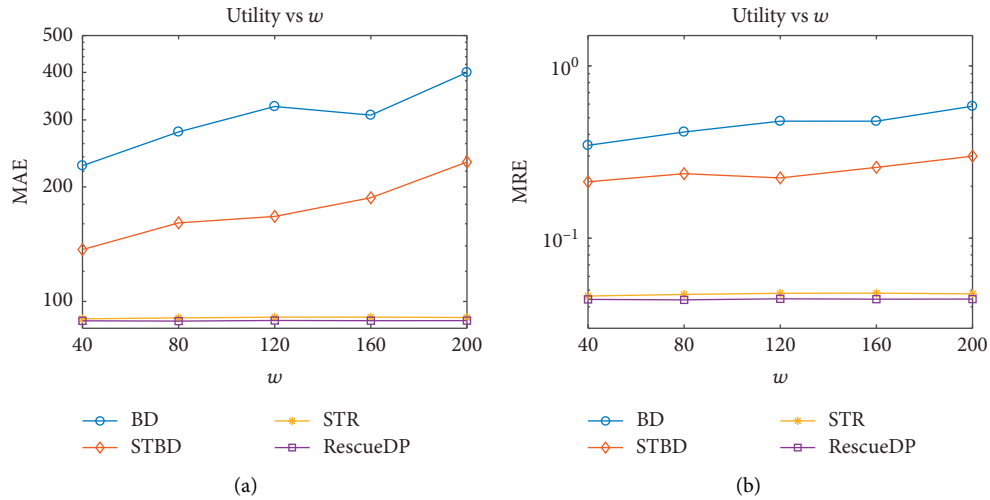


Figure 9: Utility vs. w.

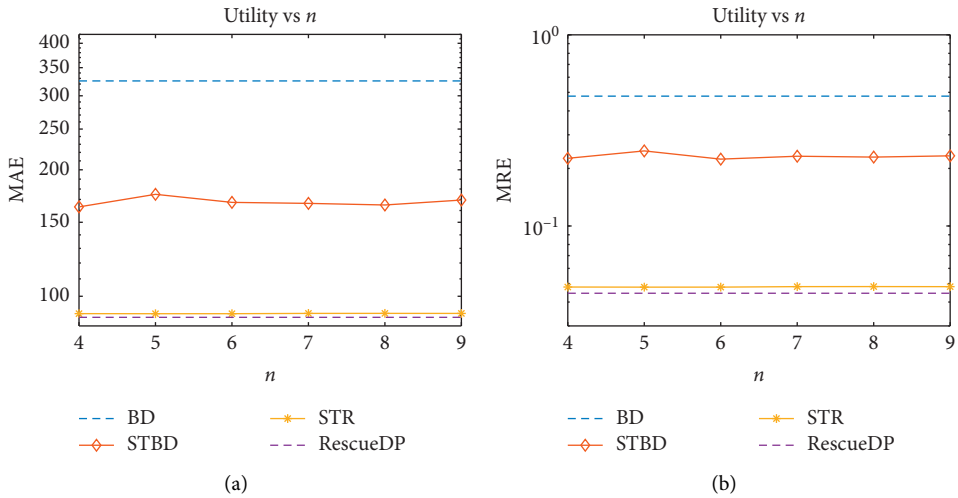


Figure 10: Utility vs. n.

gets the best data utility, but it cannot protect trajectories that traverse multiple places. STBD allows pages to have different publication decisions and different perturbation noises. Thus, STBD outperforms BD. STR constrains the perturbation budget consumed by the pages in the n-range spatial window. Therefore, the performance of STR is worse than that of RescueDP. STBD and STR protect any spatial-temporal sequence within any spatial-temporal window of size (w, n), which can cover most trajectories. Therefore, STBD and STR can achieve a balance between data utility and trajectory privacy guarantee.

6. Conclusion

In this paper, we propose (w, n)-differential privacy to protect trajectory privacy in spatial-temporal streams. This privacy model protects any spatial-temporal sequence occurring in any window of w timestamps and n-range places. We introduce the way of constructing the spatial-temporal window and finding the appropriate window size. To achieve better data utility, two implementation algorithms, STBD and STR, are proposed. Both of these two algorithms adaptively allocate the privacy budget and publish data according to the characteristics of data. Experiments on a real-world dataset show that our proposed algorithms can achieve a balance between data utility and trajectory privacy guarantee.

Data Availability

The pseudocode of our algorithm is given in the article. The web browsing history and source code of word2vec used to support the findings of this study are included within the references.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61572071 and 61872031 and the Fundamental Research Funds for the Central Universities under Grant no. 2019YJS020.

References

[1] L. Sweeney, "k-ANONYMITY: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[2] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: privacy beyond k-anonymity," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, p. 24, April 2006.

[3] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: privacy beyond k-anonymity and l-diversity," in *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering*, pp. 106–115, IEEE, Istanbul, Turkey, April 2007.

[4] M. Prakash and G. Singaravel, "A new model for privacy preserving sensitive data mining," in *Proceedings of the 2012 Third International Conference on Computing, Communication and Networking Technologies*, pp. 1–8, ICCCNT'12), Karur, India, July 2012.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Theory of Cryptography Conference*, pp. 265–284, Springer, New York, NW, USA, March 2006.

[6] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, pp. 715–724, ACM, Cambridge, Massachusetts, June 2010.

[7] J. Sun, R. Zhang, J. Zhang, and Y. Zhang, "Pristream: privacy-preserving distributed stream monitoring of thresholded percentile statistics," in *Proceedings of the IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, IEEE, San Francisco, CA, USA, April 2016.

[8] L. Sun, C. Ge, X. Huang, Y. Wu, and Y. Gao, "Differentially private real-time streaming data publication based on sliding window under exponential decay," *Computers, Materials & Continua*, vol. 58, no. 1, pp. 61–78, 2019.

[9] L. Fan and L. Xiong, "Real-time aggregate monitoring with differential privacy," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 2169–2173, ACM, Maui, HI, USA, October 2012.

[10] G. Acs and C. Castelluccia, "A case study: privacy preserving release of spatio-temporal density in paris," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1679–1688, ACM, New York, NW, USA, August 2014.

[11] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *Proceedings of the VLDB Endowment*, vol. 7, no. 12, pp. 1155–1166, 2014.

[12] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, p. 1, 2016.

[13] T. Wang, X. Yang, X. Ren, J. Zhao, and K.-Y. Lam, "Adaptive differentially private data stream publishing in spatio-temporal monitoring of iot," in *Proceedings of the 2019 IEEE 38th International Performance Computing and Communications Conference*, pp. 1–8, IPCCC), London, England, October 2019.

[14] L. Fan, L. Bonomi, L. Xiong, and V. Sunderam, "Monitoring web browsing behavior with differential privacy," in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 177–188, ACM, Seoul, Korea, April 2014.

[15] X. Yang, T. Wang, X. Ren, and W. Yu, "Survey on improving data utility in differentially private sequential data publishing," *IEEE Transactions on Big Data*, p. 1, 2017.

[16] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB Journal*, vol. 22, no. 6, pp. 797–822, 2013.

[17] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially private spatial decompositions," in *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, Arlington, VA, USA, April 2012.

[18] W. Qardaji, W. Weining Yang, and N. Ninghui Li, "Differentially private grids for geospatial data," *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, ICDE), in *Proceedings of the 2013 IEEE 29th International Conference on Data Engineering*, pp. 757–768, April 2013.

[19] Y. Xia, T. Zhu, X. Ding, H. Jin, and D. Zou, "Heterogeneous differential privacy for vertically partitioned databases," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 8, p. e5607, 2019.

[20] D. Lv and S. Zhu, "Achieving correlated differential privacy of big data publication," *Computers & Security*, vol. 82, pp. 184–195, 2019.

[21] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau, "Pegasus: data-adaptive differentially private stream processing, CCS '17," in *Proceedings of the Association for Computing Machinery*, pp. 1375–1388, New York, NY, USA, June 2017.

[22] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, pp. 735–746, ACM, Indianapolis, , IND, USA, June 2010.

[23] X. Ren, S. Wang, X. Yao, C.-M. Yu, W. Yu, and X. Yang, "Differentially private event sequences over infinite streams with relaxed privacy guarantee," in *Wireless Algorithms, Systems, and Applications Wireless Algorithms, Systems, and Applications*, E. S. Biagioni, Y. Zheng, and S. Cheng, Eds., , 2019.

[24] L. Fan, L. Xiong, and V. Sunderam, "Differentially private multi-dimensional time series release for traffic monitoring," in *Lecture Notes in Computer Science Data and Applications Security and Privacy XXVII*, L. Wang and B. Shafiq, Eds., Springer Berlin Heidelberg, Berlin, Germany, 2013.

[25] X. Liu, Y. Guo, Y. Chen, and X. Tan, "Trajectory privacy protection on spatial streaming data with differential privacy," in *Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, IEEE, Abu Dhabi, United Arab Emirates, December 2018.

[26] Y. Cao, Y. Xiao, L. Xiong, L. Bai, and M. Yoshikawa, "PriSTE," *Proceedings of the VLDB Endowment*, vol. 12, no. 12, pp. 1866–1869, 2019.

[27] H. Huang, X. Niu, C. Chen, and C. Hu, "A differential private mechanism to protect trajectory privacy in mobile crowd-sensing," in *Proceedings of the 2019 IEEE Wireless Communications and Networking Conference*, pp. 1–6, WCNC), Marrakesh, Morocco, April 2019.

[28] Z. Wang, X. Pang, Y. Chen et al., "Privacy-preserving crowd-sourced statistical data publishing with an untrusted server," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1356–1367, 2019.

[29] X. Ren, C.-M. Yu, W. Yu, X. Yang, J. Zhao, and S. Yang, "Dpcrowd: privacy-preserving and communication-efficient decentralized statistical estimation for real-time crowd-sourced data," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2775–2791, 2021.

[30] R. . Albert, H. Jeong, and L. A. Barabási, "Diameter of the world wide web," *Nature*, vol. 401, no. 6, pp. 130-131, 1999.

[31] A. Broder, R. Kumar, F. Maghoul et al., "Graph structure in the web," *Computer Networks*, vol. 33, no. 1-6, pp. 309–320, 2000.

[32] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. rep, 1999.

[33] L. Yu and M. Iwaihara, "Finding high quality documents through link and click graphs," *2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI)*, IIAI-AAI), in *Proceedings of the 2018 7th International Congress on Advanced Applied Informatics*, pp. 49–54, July 2018.

[34] M. Naldi, "Approximation of the truncated zeta distribution and zipf 's law," 2015.

[35] L. Bing, Z. Y. Niu, P. Li, W. Lam, and H. Wang, "Learning a unified embedding space of web search from large-scale query log," *Knowledge-Based Systems*, vol. 150, Article ID S095070511830100X, 2018.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computer Science*.

[37] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, pp. 19–30, ACM, New York, NW, USA, June 2009.

[38] B. Lab, "Worldcup1998," 1998, http://ftp://ita.ee.lbl.gov/html/contrib/WorldCup.html.

[39] "Google, word2vec," 2019, https://github.com/tensorflow/tensorflow/tree/r0.11/tensorflow/examples/tutorials/word2vec.

[40] D. Kingma, J. Ba, and Adam, "A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, p. 12, Banff, Canada, April 2014.

[41] M. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemporary Physics*, vol. 46, no. 5, pp. 323–351, 2005.

[42] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and experiment*, vol. 2008, no. 10, Article ID P10008, 2008.