Peer-to-Peer Networking and Applications (2021) 14:1650–1665
https://doi.org/10.1007/s12083-021-01078-6

Impact of data correlation on privacy budget allocation in continuous publication of location statistics

D. Hemkumar[1] · **S. Ravichandra**[1] · **D. V. L. N. Somayajulu**[1]

Received: 17 July 2020 / Accepted: 13 January 2021 / Published online: 20 March 2021

Abstract

Continuous publication of statistics collected from various location-based applications may compromise users' privacy as the statistics could be procured from users' private data. Differential Privacy (DP) is a new privacy notion that offers a strong privacy guarantee to all users who participate in the statistics. However, the existing DP mechanism for continuous publication of location statistics provides a privacy guarantee with the assumption that the data-points of users stream at consecutive timestamps are independent. In reality, users' data-points may be temporally correlated, resulting in more privacy leakage due to an inadequate supply of privacy budget to the timestamps where the data-points are correlated. In this paper, we present a reformulated differential privacy definition to quantify the impact of temporal correlation on privacy leakage. Then, we introduce a privacy budget allocation method for allocating an adequate amount of privacy budget to each successive timestamps under the protection of differential privacy. Our solution adopts w-event privacy for continuously releasing statistics over infinite streams. The main idea is to check the dissimilarity between statistics at each timestamp and decide whether to publish current statistics or last release statistics. Finally, we evaluate the data utility of our proposed method by presenting experimental results for real and synthetic data sets.

Keywords Location data publication · Differential privacy · Correlated data · Privacy budget allocation

1 Introduction

Numerous applications require continuous publication of location statistics for providing various social benefits such as marketing analysis [1], smart healthcare [2], traffic surveillance system [3] and online advertisement [4]. However, sharing such location statistics without preserving users' privacy may lead to serious mistrust between the users and the data published organization [5]. For instance, a hospital wants to share a collected radio frequency identification (RFID) data of patients who suffer from a specific disease with the researchers for the analysis of disease and disorder outbreaks. Due to the inadvertent sharing of patients' data, the researchers can identify and disclose the type of disease a targeted patient suffers from, which leads to compromising patients' privacy. To overcome this, various privacy preservation techniques have been proposed for preserving users' privacy in the publication of location datasets [6–9, 42]. Anonymization [10, 11] is one of the techniques and is used to perturb the dataset before it is published. But it runs out to maintain the trade-off between utility and privacy. In addition, only four users moving (or location) data points are sufficient to identify 95% the users' trajectory uniquely in the published dataset. Therefore, a rigorous privacy mechanism is needed to offer a strong privacy guarantee to all users.

To this end, ϵ-Differential Privacy (ϵ-DP) [9] is a popular privacy mechanism. It is proved that DP provides strong privacy guarantees to users against an adversary with unbounded knowledge. It ensures that any user's privacy leakage is to be strictly bounded by at most a ϵ value, where ϵ is a user parameter. If the value of ϵ is small, it achieves a strong privacy guarantee and vice versa. The ϵ-DP releases a noisy output instead of true output for hiding user's sensitive information. This noisy output is computed by adding a

This article belongs to the Topical Collection: *Special Issue on Privacy-Preserving Computing*
Guest Editors: Kaiping Xue, Zhe Liu, Haojin Zhu, Miao Pan and David S.L. Wei

✉ D. Hemkumar
hemkumar.medar@gmail.com

[1] National Institute of Technology Warangal, Hanamkonda, India

random noise (derived from the Laplace distribution with scale λ) to the true output.

Recently, the ϵ-DP privacy notion has been applied in settings of continuous data publishing [12–14, 28–32, 41]. For example, the traffic surveillance system periodically publishes a count of people (or users) at each location per timestamps in privately. In the literature, there exist a few privacy approaches such as event-level privacy [15], user-level privacy [16] and w-event privacy [13] for continuous private data publishing. Event-level privacy provides a ϵ-DP guarantee to each event's (or each timestamp's) count. In other words, it protects only a single data-point of the user's entire stream. However, by combining all event's count, the adversary can reconstruct the user's stream, which leads to an effect on users' privacy [13]. In contrast, user-level privacy guarantees a ϵ-DP to finite event's (or timestamp's) count. In other words, it protects only a finite length of users' stream. Due to this, the user-level privacy has limited applicability in most of the real-world applications. The w-event privacy mechanism has been proposed to address the limited use of event-level privacy and user-level privacy. This mechanism offers a strong privacy guarantee to any user stream within a window of w timestamps. A w-event privacy presents a sliding window methodology that involves a broad range of w-event private mechanisms. Each mechanism constructs a separate sub-mechanism per timestamp, and each sub-mechanism uses a certain privacy budget to control the noise (higher privacy budget, lower perturbation, or less noise added). The w-event privacy achieves ϵ-DP when the sum of all privacy budgets used in any window of w timestamps is at most total privacy budget ϵ.

However, the w-event privacy mechanism provides less privacy guarantee than traditional ϵ-DP, especially when the user's data-points are not independent (i.e., temporally correlated) between consecutive timestamps. It happens due to the allotted privacy budget at timestamps within a window of size w is not adequate, especially where the data-points of users' stream involve temporal correlation. Therefore, the privacy budget distribution strategies in w-event privacy such as Budget Distribution (BD) and Budget Absorption (BA) are not suitable in the presence of correlated datasets within a window. The following example illustrates how privacy guarantee is degrading when users' data-points have a temporal correlation.

Example 1 Assume that a trusted curator collects users' location data-points in continuous timestamps, as shown in Fig. 1a. The curator aims to publish a statistic (*i.e., how many users are in each location)* at each timestamp without breach of any user privacy. According to the Laplace mechanism, the curator publishes private statistics at every timestamp using an independent random noise derived from the Laplace distribution with scale $Lap(1/\epsilon)$, where ϵ is a privacy budget. However, if the nature of two location data-points of users' stream at consecutive timestamps is temporally correlated, then the independent random noise with a scale $Lap(1/\epsilon)$ achieves 2ϵ-DP instead of ϵ-DP. It happens due to the modification or removal of one data

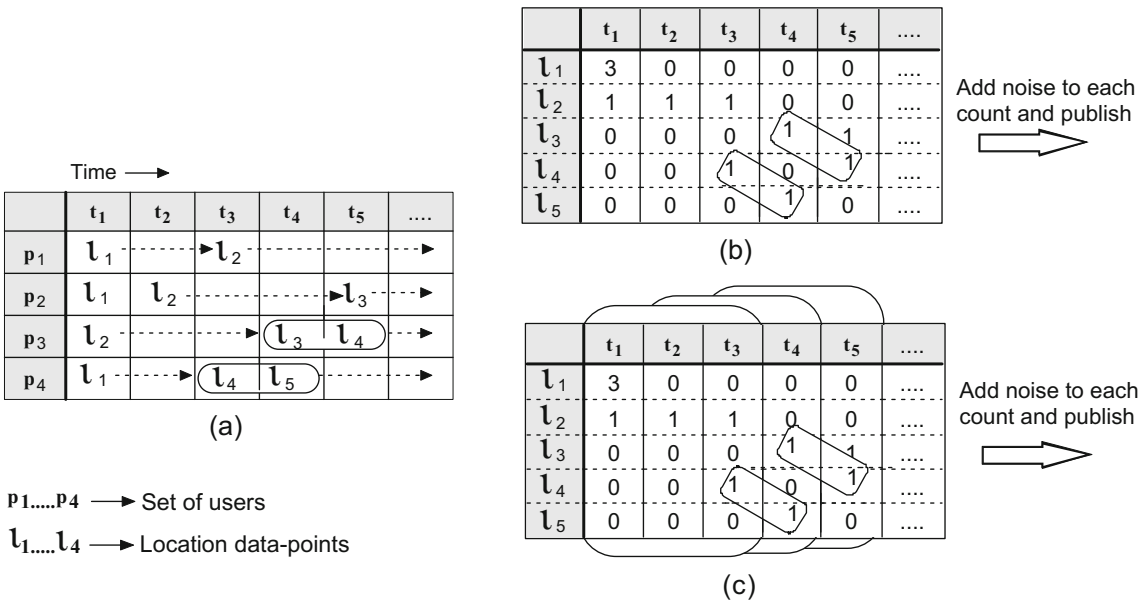

Fig. 1 Illustration of example 1 (**a**) Collection of users location data-points in continuous timestamps (**b**) Statistics for event-level or user-level privacy (**c**) Statistics for w-event privacy while set $w = 3$

point affects two counts in the published statistics (i.e., the global sensitivity (Δ) is 2), as shown in the Fig. 1b. Consequently, the presented two privacy budget distribution approaches in w-event privacy achieve $w\epsilon$-DP instead of ϵ-DP, especially when the nature of all location data-points of users' stream at consecutive timestamps are temporally correlated. Hence, the w-event privacy mechanism is not suitable for the publication of temporally correlated user streams, as shown in Fig. 1c.

Figure 2 shows that the distribution of total privacy budget ϵ to each timestamp within a sliding window of size 3. In the first window of size 3, the traditional ϵ-DP privacy mechanism allots a privacy budget ($\epsilon/3$) uniformly to each timestamp. Then, compute noise lap($1/(\epsilon/3)$) to perturb each timestamp counts by using the allotted privacy budget. Hence, the required ratio of the privacy budget at each timestamp is $\epsilon/3$ to achieve ϵ-DP in the first sliding window. In the second sliding window, the required ratio of privacy budget at timestamps 2, 3 and 4 is $\epsilon/3$, $\epsilon/3$ and $2\epsilon/3$ respectively due to the presence of temporal correlation between the timestamps 3 and 4 as shown in the Fig. 1c. The sum of the privacy budgets in the second window is exceeded than the total privacy budget ϵ (i.e., $\epsilon/3 + \epsilon/3 + 2\epsilon/3 > \epsilon$). Thus, the second sliding window violates ϵ-DP privacy mechanism. Similarly, the third sliding window also violates ϵ-DP due to the presence of temporal correlation between the timestamps 4 and 5 (i.e., the required ratio of privacy budget at timestamp 5 is $2\epsilon/3$). Therefore, it is necessary to design a privacy budget distribution method for allocating a sufficient privacy budget to all timestamps within the sliding window of size w.

Further, there is a limited state of art methods for distributing a privacy budget in continuous data publishing settings. There exist a few baseline approaches for allocating privacy budgets in order to publish continuous location statistics privately. A Uniform method is to uniformly allocate a privacy budget to N timestamp's dataset. This approach achieves ϵ-DP since it combines

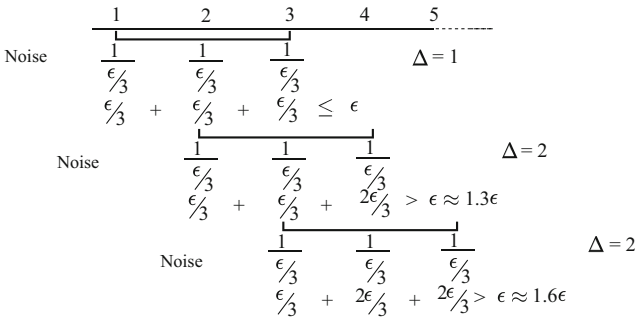

Fig. 2 Distribution of privacy budget over timestamps (or event) within the sliding window of size $w = 3$

all privacy budgets of N timestamp's datasets [24]. In our problem settings, the datasets at consecutive timestamps require more privacy budget than the baseline approach due to the presence of temporal-correlation. The fixed sampling [16, 25] is another approach for allocating a privacy budget at a given sampling interval I among N timestamps. So, the privacy budget at each interval I is $(\epsilon * I)/N$. It is also preserved ϵ-DP by combining the privacy budgets of all samples. This approach is not useful because pre-defined sampling intervals are not determined accurately if location data points arrive dynamically, and occur high perturbation errors if sampling intervals are too frequent.

In summary, the privacy budget distribution into the series of temporally correlated data-points in users stream remains unclear in the w-event privacy method. The major contributions of this paper are as follows.

1. We present a reformulated differential privacy definition for continuous data publication and prove that it can achieve ϵ-DP. Then we quantify the impact of temporal correlation on privacy leakage in reformulated ϵ-DP and analyze the privacy leakage in ϵ-DP with a numerical example.
2. We introduce a Privacy Budget Allocation method for allocating an adequate amount of privacy budget to each successive timestamps under the protection of ϵ-Differential privacy.
3. Finally, we evaluate the data utility of our method by computing the average error per timestamps through conducting a series of experiments on real and synthetic datasets.

The rest of this paper is as follows. Section 2 introduces the necessary notations and definitions of differential privacy under continual observation. In Section 3, we analyze the impact of temporal correlation on privacy leakage with a numerical example. Section 4 proposes a Privacy Budget Allocation method (PBA) and presents a theoretical analysis of privacy leakage and utility leakage of this method. A numerical experiment conducted on various datasets to evaluate the data utility of our method is presented in Section 5. Section 6 concludes the paper.

2 Background

In this section, we describe the necessary notations and definitions of differential privacy under continual observation.

2.1 Preliminaries

Let χ be the universe of possible input location data-points. In our setting, we assume that each location data-point is a

vector of size L, where L is a total number of all possible locations. Suppose, if a user is at location $l_i \in L$, then the corresponding bit (or column) i in the vector is set to be 1, and all remaining bits are set to zero. At every timestamp t, curator collects a dataset D_t with k rows, denoted as the set of indices $[k] = \{1, 2, ...k\}$. Let S be the stream prefixes of location data-points and we represent the stream prefix S up to t timestamp is $S_t = (D_1, D_2, ..., D_t)$. Let $q : D \rightarrow \mathbb{R}^L$ be the counting query function where D is the set of all datasets with L columns. The curator aims to publish a result (or statistics) for a counting query at each timestamp. a_i is the output of the dataset $S[i] = D_i$. The curator publishes a series of outputs $(a_1, a_2, a_3, ...)$ in contiguous timestamps. To preserve the privacy of k users, the curator releases a noisy output (say ω_i) by adding the random noise derived from the Laplace distribution.

Definition 1 (Adj(S_t, S'_t)) Let S_t and S'_t are the stream prefixes of location data-points drawn from the χ. The S_t is adjacent to S'_t if and only if they are differing in one or more data-points of any one user stream prefixes. More formally, Adj(S_t, S'_t) iff $\exists m, n \in \chi$ and $\exists k \subseteq [|S_t|]$ such that $S_t|_{k:m \rightarrow n} = S'_t$. Here, k is a set of indices in the stream prefix S_t and $S_t|_{K:m \rightarrow n}$ is the result of modifying all the occurrences of m at these indices with n.

2.2 Differential privacy under continual observation

Let $\mathcal{M}$ be a privacy mechanism which takes stream prefixes S_t as input and produced a series of outputs $\omega = (\omega_1, \omega_2, ..., \omega_t) \in \Omega$ at each timestamp. The privacy mechanism $\mathcal{M}$ is said to be ϵ-Differentially private iff the following logarithmic function is to be bounded by maximum ϵ value for any adjacent stream prefixes S, Adj(S_t, S'_t) and any possible output Ω of $Range(\mathcal{M})$.

$$log \frac{P_r(\mathcal{M}(S_t) = (\omega_1, \omega_2, ..., \omega_t))}{P_r(\mathcal{M}(S'_t) = (\omega_1, \omega_2, ..., \omega_t))} \leq \epsilon$$

Where the parameter ϵ quantifies the degree of a user privacy leakage. Suppose, $\mathcal{M}$ is decomposed into $(\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_t)$ sub-mechanisms. Each sub-mechanism $\mathcal{M}_i(D_i)$ produce an output ω_i with independent randomness. Hence, it holds $\frac{P_r(\mathcal{M}_i(D_i)=\omega_i)}{P_r(\mathcal{M}_i(D'_i)=\omega_i)} \leq e^{\epsilon_i}$ and guarantees ϵ_i-DP. Therefore we derive

$$\frac{P_r(\mathcal{M}(S_t) = (\omega_1, \omega_2, ..., \omega_t))}{P_r(\mathcal{M}(S'_t) = (\omega_1, \omega_2, ..., \omega_t))} = \prod_{i=1}^{t} \frac{P_r(\mathcal{M}_i(D_i) = \omega_i)}{P_r(\mathcal{M}_i(D'_i) = \omega_i)} \tag{1}$$

$$\leq \prod_{i=1}^{t} e^{\epsilon_i} \leq exp(\sum_{i=1}^{t} \epsilon_i) \leq e^{\epsilon} \tag{2}$$

Definition 2 (Laplace mechanism) It is one of the most common methods for achieving ϵ-DP. Given a counting query q, the Laplace mechanism generates random noise (say x) derived from Laplace distribution with scale $Lap(\lambda)$ and is added to the true answer of a query q i.e., $r_t = q(D_t) + x$. The probability density function of Laplace distribution is

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

Where $\lambda = \Delta q/\epsilon$, Δq is a global-sensitivity of a query q, which is a maximum difference between the outputs over the adjacent stream prefixes S_t and S'_t i.e., $\Delta q = \max_{S_t, S'_t} ||q(S_t) - q(S'_t)||$. The parameters Δq and ϵ plays a significant role in calibrating the noise to the query outputs.

Let $A = \{A_i : i \in [1, k]\}$ be the set of adversaries with arbitrary knowledge and are interested in the user's private data. Consider an adversary A_i whose target is i^{th} user private data and has knowledge of all other users' private data except i^{th} user, i.e., A_i knows $S_t = S_t \backslash \{i\}$. The privacy leakage of privacy mechanism $\mathcal{M}$ (or i^{th} user) at timestamp t against A_i is as follows, in which l_i^t and $l_i^{t'}$ are two possible data points of i^{th} user at timestamps t.

$$\mathcal{L}_{A_i}(\mathcal{M}_t) = \sup_{\omega, l_i^t, l_i^{t'}} log \frac{P_r(\omega | l_i^t, S_t)}{P_r(\omega | l_i^{t'}, S'_t)} \tag{3}$$

$$\mathcal{L}(\mathcal{M}_t) = \max_{\forall A_i, i \in [k]} \mathcal{L}_{A_i}(\mathcal{M}_t)$$

Table 1 Notations summary

χ	Universe of location data-points
L	Set of all possible locations
D_t	Dataset with k rows at timestamps t
S_t, S'_t	Adjacent Stream prefixes of S ends with current timestamps t
$S[i]$	i^{th} dataset of stream prefixes S
M	ϵ-Differential Privacy mechanism
q	Count query
a_t, r_t	True answer and noisy answer at timestamp t respectively
$a_t[n]$	n^{th} column result of a_t
θ	Transition probability between the location data-points at different timestamps

The $\mathcal{L}(\mathcal{M}_t)$ is the maximum privacy leakage at timestamp t caused by any k adversary. Here, we considered a privacy budget ϵ as a metric of privacy leakage. If lesser ϵ value, then lesser the privacy leakage. The summary of the notations is listed in Table 1.

3 Temporal correlation (TC) privacy leakage analysis

3.1 Adversary's knowledge

In the stream data publication, it is fair to consider that an adversary knows the transition probability between the possible location data-points. In our settings, we adopted a Markov chain process (MC) for modeling a transition probability between the location data-points (according to certain probabilistic rules) and is denoted as $\theta \in \Theta$, where Θ is a set of all transition probability distributions. In MC, the transition matrix describes the probabilities of transition from one data-point to another data-point, and the sum of transition probabilities in each row is equal to 1. Let consider a transition matrix of size 2 as shown in the Table 2(a). If a user i is at loc_1 (current location), then the probability of coming from loc_2 (previous location) is 0.4, represented as $P_r[l_i^{t-1} = loc_2 | l_i^t = loc_1] = 0.4$.

3.2 TC privacy leakage

Consider an adversary A_i with knowledge of $S_t = S_t \backslash \{i\}$ and transition probability distributions θ, named as A_i^θ. Let A_i^θ collects all private outputs which were published under the protection of ϵ-DP mechanisms $\mathcal{M}$ at each timestamps $t \in [1, T]$. Now, the aim of the adversary is to infer user $i's$ location data-point at timestamp t.

The TC privacy leakage ($\mathcal{TCL}$) of $\mathcal{M}_t$ w.r.t A_i^θ is the maximum ratio of two laplace distribution for all different values of $l_i^t, l_i^{t'}$ and for all possible transition probability distributions.

$$\mathcal{TCL}_{A_i^\theta}(\mathcal{M}_t) = \sup_{\omega, l_i^t, l_i^{t'}, \theta} log \frac{P_r(\omega \in \Omega | l_i^t, S_t, \theta)}{P_r(\omega \in \Omega | l_i^{t'}, S_t', \theta)} \tag{4}$$

The TC privacy leakage of $\mathcal{M}_t$ w.r.t any A_i^θ where $i \in [k]$ is less than or equal to ϵ, then we call $\mathcal{M}_t$ is ϵ-TC Differential privacy.

$$\sup_{\theta, \forall A_i, i \in [k]} \mathcal{TCL}_{A_i^\theta}(\mathcal{M}_t) \leq \epsilon \tag{5}$$

Further, to understand the impact of temporal correlation on privacy leakage in continuous data publish settings, Eq. 4 is expanded and simplified by Bayes theorem, i.e.,

$$\begin{aligned}
\mathcal{TCL}_{A_i^\theta}(\mathcal{M}_t) &= \sup_{\omega_1, ..., \omega_t, l_i^t, l_i^{t'}, \theta} log \frac{P_r(\omega_1, ..., \omega_t | l_i^t, S_t, \theta)}{P_r(\omega_1, ..., \omega_t | l_i^{t'}, S_t', \theta)} \\
&= \sup_{\omega_1, ..., \omega_{t-1}, l_i^t, l_i^{t'}, \theta} log \frac{\sum_{l_i^{t-1}} P_r(\omega_1, ..., \omega_{t-1} | l_i^{t-1}, S_{t-1}) P_r(l_i^{t-1} | l_i^t)}{\sum_{l_i^{t-1'}} P_r(\omega_1, ..., \omega_{t-1} | l_i^{t-1'}, S_{t-1}') P_r(l_i^{t-1} | l_i^t)} \\
&+ \sup_{\omega_t, l_i^t, l_i^{t'}, \theta} log \frac{P_r(\omega_t | l_i^t, S_t)}{P_r(\omega_t | l_i^{t'}, S_t')}
\end{aligned} \tag{6}$$

There are three annotated terms in Eq. 6. The first term determines privacy leakage at previous timestamp $t-1$; the second term indicates the probability of transition between the data-points of previous timestamp $(t-1)$ and current timestamp (t), and the last term is equal to the privacy leakage at time t. Hence, the privacy leakage at time t depends on the privacy leakage at time $t-1$, TC transition probability, and the privacy leakage at time t. Notice that, if $t = 1$, then $\mathcal{TCL}_{A_i^\theta}(\mathcal{M}_t) = \mathcal{L}_{A_i}(\mathcal{M}_1)$. Otherwise, if $t > 1$, then we have the following equation.

$$\mathcal{TCL}_{A_i^\theta}(\mathcal{M}_t) = \mathcal{TCL}_{A_i^\theta}(\mathcal{M}_{t-1}) + \mathcal{L}_{A_i}(\mathcal{M}_t) \tag{7}$$

The first term of the above Eq. 7 is calculated using the *temporal privacy loss function* given in [14]. We illustrate how the TC factor influences privacy leakage w.r.t adversary with and without knowledge of θ distribution through a numerical example. Consider a query to find the user $i's$ location value l_i is either loc_m or loc_n at timestamp t, where loc_m and $loc_n \in L$. For simplicity, we assume that l_i^t is,

$$l_i^t = \begin{cases} 1 & \text{i's true location at time t,} \\ 0 & \text{Otherwise.} \end{cases} \tag{8}$$

Example 2 Let a database D of two users u_1 and u_2 (as shown in the Table 2(b)), A_1^θ and A_1 are the two adversaries with and without knowledge of θ distribution respectively and are interested in finding the location of u_1 at timestamp 2. Assume that both adversaries know the location information of u_2. According to the definition of $TC - DP$, we compute $\mathcal{TCL}_{A_1}(\mathcal{M})$ and $\mathcal{TCL}_{A_1^\theta}(\mathcal{M})$. For A_1 without knowledge of θ, we get

$$\begin{aligned}
\mathcal{TCL}_{A_1}(\mathcal{M}_2) &= \sup_{\omega_1, \omega_2} log \frac{P_r(\omega_1, \omega_2 | l_1^2 = loc_1, l_2^2 = loc_2)}{P_r(\omega_1, \omega_2 | l_1^{2'} = loc_2, l_2^2 = loc_2)} \\
&= \sup_{\omega_1} log \frac{\sum_{l_1^1} exp(-|\omega_1 - (l_1^1, l_2^1 = loc_1)|) P_r(l_1^1 | l_1^2 = loc_1)}{\sum_{l_1^{1'}} exp(-|\omega_1 - (l_1^{1'}, l_2^1 = loc_1)|) P_r(l_1^{1'} | l_1^{2'} = loc_2)}
\end{aligned}$$

Table 2 Illustration

(a) Transition matrix

	loc_1	loc_2
loc_1	0.6	0.4
loc_2	0.1	0.9

(b) Database D

	D_1	D_2
u_1	loc_1	loc_1
u_2	loc_1	loc_2

$$+\sup_{\omega_2} log \frac{exp(-|\omega_2 - (l_1^2 = loc_1, l_2^2 = loc_2)|)}{exp(-|\omega_2 - (l_1^{2'} = loc_2, l_2^2 = loc_2)|)}$$
$$= 0 + \sup_{\omega_2} log \frac{exp(-|\omega_2 - 2|)}{exp(-|\omega_2 - 1|)} = 1$$

For A_1^θ with knowledge of θ, we get

$$\mathcal{TCL}_{A_1^\theta}(\mathcal{M}_2) = \sup_{\omega} log \frac{P_r(\omega_1, \omega_2 | l_1^2 = loc_1, l_2^2 = loc_2)}{P_r(\omega_1, \omega_2 | l_1^{2'} = loc_2, l_2^2 = loc_2)}$$
$$= \sup_{\omega_1} log \frac{\sum_{l_1^1} exp(-|\omega_1 - (l_1^1, l_2^1 = loc_1)|) P_r(l_1^1 | l_1^2 = loc_1)}{\sum_{l_1^{1'}} exp(-|\omega_1 - (l_1^{1'}, l_2^1 = loc_1)|) P_r(l_1^{1'} | l_1^{2'} = loc_2)}$$
$$+\sup_{\omega_2} log \frac{exp(-|\omega_2 - (l_1^2 = loc_1, l_2^2 = loc_2)|)}{exp(-|\omega_2 - (l_1^{2'} = loc_2, l_2^2 = loc_2)|)}$$
$$= 0.55 + 1 = 1.55$$

The above numeric analysis shows that TC has a significant influence on higher privacy leakage i.e., $\mathcal{TCL}_{A_1^\theta}(\mathcal{M}_2) > \mathcal{TCL}_{A_1}(\mathcal{M}_2)$. Hence, we can state that the curator (or data publisher) does not provide a strong privacy guarantee compared with traditional ϵ-DP in continuous data publication settings. In detail, a recent privacy method called w-event privacy allocates a ratio of privacy budget to each timestamp to achieve ϵ-DP guarantee of any user's stream within a window of size w by assuming the data-points in a user stream are independent. However, most of the location data-points are temporally correlated with a certain probability in real-time data collection. Due to this, the allotted privacy budget at timestamps within a window is not adequate to achieve ϵ-DP, resulting in more privacy leakage than the traditional ϵ-DP.

4 Proposed method

This section discusses our Privacy Budget Allocation (PBA) mechanism, which is allowed to compute and allocate the quantity of privacy budget to each publication in a continuous data release setting. Then, we theoretically prove that our PBA mechanism achieves ϵ-DP and shows the data utility of PBA mechanism.

This mechanism is motivated by limited use of previous mechanisms such as Uniform, Sampling, and w-event privacy, which are discussed in the introduction. In this mechanism, we adopt a w-event privacy concept called sliding window methodology, and it follows that the window is moving one timestamp ahead after every w timestamps. A sliding window consists of w number of timestamps and each timestamp t is operated by a sub-mechanism $\mathcal{M}_t$. Since each $\mathcal{M}_t$ uses independent randomness, $\mathcal{M}_t$ achieves ϵ_t-DP for some ϵ_t. The sum of the privacy budgets within the sliding window of size w must be lesser than or equal to the total privacy budget ϵ. Note that, at any timestamp t, span of sliding window is $t - w + 1$ to t.

The PBA mechanism $\mathcal{M}$ consists of series of sub-mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k, \ldots, \mathcal{M}_t$, where each $\mathcal{M}_k$ takes dataset $S_t[k] = D_k$ as input and publishes a private statistic ω_k as output by using allotted privacy budget ϵ_k. Thus, $\mathcal{M}$ publishes a series of private statistics, namely $\omega_1, \omega_2, .., \omega_k, .., \omega_t$. In detail, the mechanism $\mathcal{M}$ involves two phases $\mathcal{M}^1$ and $\mathcal{M}^2$. These two phases operate sequentially by using half of the total privacy budget, i.e., ϵ^1 and ϵ^2. In the first phase, $\mathcal{M}^1$ allocates a ratio of privacy budget from ϵ^1 to each timestamp uniformly within a sliding window. At timestamp k, the sub-mechanism $\mathcal{M}_k^1$ calculates a dissimilarity value between the true statistic a_k and last release private statistic ω_l. The mean of absolute error (MAE) is a metric which measures the dissimilarity between a_k and ω_l and is formulated as $\frac{1}{|L|}\Sigma_{j=1}^{|L|}|\omega_l[j] - a_k[j]|$ where a_k and ω_l are the vectors of length $|L|$. Then, the obtained dissimilarity value is forwarded into $\mathcal{M}_k^2$. In the second phase, $\mathcal{M}^2$ divides a privacy budget ϵ^2 into two parts; namely publication privacy budget and absorption privacy budget. The $\mathcal{M}^2$ allocates a publication privacy budget into each timestamp in an exponential decreasing fashion. At timestamp k, $\mathcal{M}_k^1$ forwards dissimilarity value to $\mathcal{M}_k^2$ to decide whether to publish a true publication with noise or null publication (last release private output). If $\mathcal{M}_k^2$ decides not to publish a true publication at timestamp k, then k^{th} allotted publication privacy budget is become free and can be used in the future publication if necessary. In contrast, if $\mathcal{M}_k^2$ decides true publication at timestamp k, then it uses allotted publication privacy budget to publish statistics privately. Further, the absorption privacy budget allocates an extra privacy budget at timestamp k only when a correlation exists between the current timestamp k and the previous timestamp $k - 1$. This is because a statistic at timestamp k requires more privacy budget compared to the normal publication.

Algorithm 1 describes the mechanism of PBA in continuous data release settings. It takes Dataset D_k, total privacy budget ϵ, allotted budgets upto $(k-1)^{th}$ timestamp $(\epsilon_1^2, \epsilon_2^2, \ldots, \epsilon_{k-1}^2)$ and $(\epsilon_1^a, \epsilon_2^a, \ldots, \epsilon_{k-1}^a)$ are inputs and release a noisy statistic ω_k as an output. PBA aims to allocate an adequate amount of privacy budget at each timestamp within the sliding window to achieve ϵ-DP (line 2-20). At any timestamp k, the sliding window follows two phases: the first phase is to calculate noisy dissimilarity value between ω_l and a_k (line 3-8), and the second phase is to decide whether publish a private statistics of current timestamp k if publications occur, otherwise publish ω_l (line 9-19). The sub-mechanism $\mathcal{M}_k$ computes true answer (a_k) from the dataset $S[k] = D_k$ (line 4), then calculates dissimilarity value between a_k and ω_l by using

a metric called Mean of Absolute Error (line 5). After that, $\mathcal{M}_k$ utilizes an allotted privacy budget ϵ_k^1 to make *noisy dissimilarity value* shown in the lines (6-8). Finally, $\mathcal{M}_k$ computes MAE() value (MAE + noise) and forwards into phase 2 of $\mathcal{M}_k$. In the second phase, $\mathcal{M}_k$ starts with finding a remaining amount of privacy budget available at the time of k^{th} timestamp and assign half of the remaining budget to the phase 2 of k^{th} timestamp (line 10). If a correlation exists between the present and previous timestamps, then $\mathcal{M}_k$ adds extra budget from the absorbed privacy budget ϵ_A to the phase 2 of k^{th} timestamp, is shown in the lines (12-15). Once $\mathcal{M}_k$ computes noise (line 16), then it decides whether publish a_k with noise or ω_l based on the comparison between MAE() and computes noise λ_k^2 (lines 17-20). If MAE() is greater than λ_k^2, then it releases a_k with noise otherwise releases last release output ω_l.

Algorithm 1 Pseudocode of PBA mechanism at k^{th} timestamp ($\mathcal{M}_k$).

INPUT: Dataset D_k, total privacy budget ϵ, $\epsilon_1^2, \epsilon_2^2, \ldots, \epsilon_{k-1}^2$ and $\epsilon_1^a, \epsilon_2^a, \ldots, \epsilon_{k-1}^a$
OUTPUT: Release Noisy output ω_k.

1: At sub-mechanism $\mathcal{M}_k$
2: Compute noise for last release output ω_l
3: Calculate $a_k = M(D_k)$
4: Calculate $MAE(\omega_l, a_k)$
5: Allocated budget at time k: $\epsilon_k^1 = \epsilon \cdot |L|/(2 \cdot w)$
6: Compute noise $\lambda_k^1 = 1/\epsilon_k^1$
7: Set $MAE() = MAE(\omega_l, a_k) + Lap(\lambda_k^1)$
8: Compute noise for present publication ω_k
9: Calculate remaining budget: $\epsilon_k^2 = (\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1} \epsilon_j])/2$
10: If Correlation exists
11: $\epsilon_A = (\epsilon/4 - [\sum_{j=k-w+1}^{k-1} \epsilon_j^a])$
12: Set $\epsilon_k^a = \epsilon_k^2$ (absorbed from ϵ_A)
13: $\epsilon_k^2 = \epsilon_k^a + \epsilon_k^2$
14: Compute noise $\lambda_k^2 = 1/\epsilon_k^2$
15: If $MAE() > \lambda_k^2$
16: return $\omega_k = a_k + \langle Lap(\lambda_k^2) \rangle^{|L|}$
17: Else
18: return $\omega_k = \omega_l$
19: end for
20: end for

Figure 3 shows the operation of PBA mechanism in continuous data release settings of 5 timestamps while assuming the size of $w = 3$. Assume that $\mathcal{M}$ publishes private outputs at timestamps 1,3,4,5 and last release private output at timestamp 2 i.e., the noisy output of timestamp 1. At each timestamp in phase 1, $\mathcal{M}$ allocates a fixed privacy budget i.e., $\epsilon/2 \cdot w = \epsilon/6$ (fix $w = 3$). Then $\mathcal{M}$ allocates

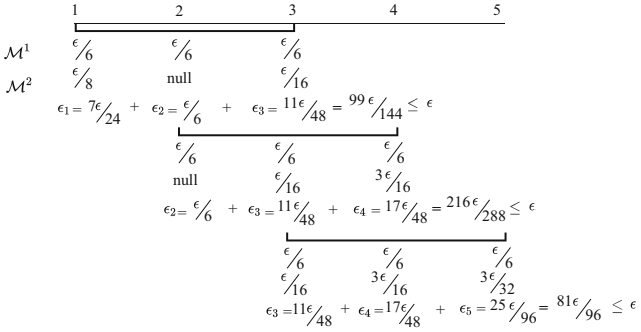

Fig. 3 Distribution of privacy budget over timestamps(or event) within the sliding window of size $w = 3$

half of the allotted privacy budget in phase 2 (i.e., $\epsilon/4$) in an exponential decreasing manner within the sliding window of size $w = 3$. In other words, at timestamp 1, it assigns $\epsilon_1^2 = (\epsilon/4 - 0)/2 = \epsilon/8$. At timestamp 2, $\epsilon_2^2 = 0$ because no output is generated at timestamp 2. At timestamp 3, $\epsilon_3^2 = (\epsilon/4 - (0 + \epsilon/8))/2 = \epsilon/16$. Since no correlation exists between the timestamps within the first sliding window, it is not required to add extra budget to any timestamps. At timestamp 4, $\epsilon_4^2 = (\epsilon/4 - (0 + \epsilon/16))/2 = 3\epsilon/32$ and adds extra budget $3\epsilon/32$ to ϵ_4^2 due to the existence of correlation in between the timestamps of 4 and 3 as shown in Fig. 1. Similarly at timestamp 5, the PBA assigns $\epsilon_5^2 = (\epsilon/4 - (\epsilon/16 + 3\epsilon/32))/2 + 3\epsilon/64$. Notice that the total sum of all privacy budgets in phases 1 and 2 of respective sliding windows is less than or equal to the total privacy budget ϵ.

Theorem 1 *Privacy Budget Allocation algorithm (PBA) achieves ϵ-Differential privacy.*

Proof A sub-mechanism $\mathcal{M}_k$ privately publishes either output of $q(D_k)$ or immediate last release output ω_l by utilizing a privacy budget ϵ_k. The sub-mechanism $\mathcal{M}_k$ has two phases that use independent privacy budgets, i.e., ϵ_k^1 and ϵ_k^2. Hence, we first prove that the sub-mechanism at phase 1 $\mathcal{M}_k^1$ satisfies ϵ_k^1-DP for $\epsilon_k^1 = \epsilon/2w$ and $\mathcal{M}_k^2$ satisfies ϵ_k^2-DP for $\epsilon_k^2 = (\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1} \epsilon_j])$ if it publishes output of $q(D_k)$, otherwise $\epsilon_k^2 = 0$. In phase 1, $\mathcal{M}_k^1$ publish private MAE() value i.e.,$q'(D_k) = \frac{1}{|L|} \Sigma_{j=1}^{|L|} |\omega_l[j] - a_k[j]|$. If add or remove a row from D_k, then the maximum alter in the result of $q'(D_k)$ is $1/|L|$. Hence, the sensitivity of q' is at most $1/|L|$. By using this sensitivity, $\mathcal{M}_k^1$ injects laplace noise with scale $\lambda_k^1 = \Delta(q')/\epsilon_k^1 = 2 \cdot w/(\epsilon \cdot |L|)$ to MAE() value. According to definition 2, $\mathcal{M}_k^1$ is ϵ_k^1-DP for $\epsilon_k^1 = \frac{1/|L|}{(2 \cdot w)/(\epsilon \cdot |L|)} = \epsilon/(2 \cdot w)$. In the second phase, $\mathcal{M}_k^2$ publishes either private output of $q(D_k)$ value or null. In former differential privacy, if add or remove a row from D_k, then the maximum alter in the result of $q(D_k)$ is 1.

Hence, the sensitivity of q is at most 1. The $\mathcal{M}_k^2$ injects laplace noise with scale $\lambda_k^2 = 2/(\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1}\epsilon_j])$. In case if a correlation exists between the current timestamp and previous timestamp, then $\mathcal{M}_k^2$ borrow extra budget from the available $\epsilon/4$. So, $\mathcal{M}_k^2$ injects laplace noise with scale $\lambda_k^2 = 2/(\epsilon/4-[\Sigma_{j=k-w+1}^{k-1}\epsilon_j])+2/(\epsilon/4-[\Sigma_{j=k-w+1}^{k-1}\epsilon_j]))$ if a correlation exists; otherwise, noise is $\lambda_k^2 = 2/(\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1}\epsilon_j])$. We assume that the mechanism $\mathcal{M}$ used an entire extra budget $(\epsilon/4)$ within a sliding window of size w. According to definition 2, $\mathcal{M}_k^2$ is ϵ_k^2-DP for $\epsilon_k^2 = (\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1}\epsilon_j])/2 + (\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1}\epsilon_j])/2) = (\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1}\epsilon_j])$.

Subsequently, we must prove that PBA holds $\Sigma_{j=k-w+1}^{k}\epsilon_j \leq \epsilon$, for every k within the sliding window. From composition property, PBA holds at j^{th} privacy budget is $\epsilon_j = \epsilon_j^1 + \epsilon_j^2$, then it equals to $\Sigma_{j=k-w+1}^{k}\epsilon_j = \Sigma_{j=k-w+1}^{k}\epsilon_j^1 + \Sigma_{j=k-w+1}^{k}\epsilon_j^2$. Since every ϵ_j^1 is set to $\epsilon/2 \cdot w$, the total privacy budget within the sliding window is $\Sigma_{j=k-w+1}^{k}\epsilon_j = \epsilon/2 + \Sigma_{j=k-w+1}^{k}\epsilon_j^2$. Now, it is required to prove that $\Sigma_{j=k-w+1}^{k}\epsilon_j^2 \leq \epsilon/2$. In our settings $\Sigma_{j=k-w+1}^{k}\epsilon_j^2$ is $\Sigma_{j=k-w+1}^{k}(\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1}\epsilon_j])/2 + (\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1}\epsilon_j])/2)$. These two terms can be proved using inequality by induction. Since both terms are equal, we prove either one of the terms is lesser than or equal to $\epsilon/4$. and then we can say that another term is also lesser than or equal to $\epsilon/4$. In the induction part, first, simplify the term using geometric series and prove that the term is lesser than or equal to $(\epsilon/2)$ using inductive steps (for more details in Appendix). Therefore, sub-mechanism $\mathcal{M}_k^2$ is always used up to half of the available privacy budget, i.e.,$(\epsilon/2)$. □

4.1 Utility analysis

In the PBA mechanism, the error at any timestamps depends on two reasons 1) a privacy budget utilized in true publications and 2) a privacy budget utilized in the last release publication, which is an approximated publication of the current timestamp. In detail, if publications occur at timestamp k, then $\mathcal{M}$ operates both the phases $\mathcal{M}_k^1$ and $\mathcal{M}_k^2$. We use the MAE metric of pair (ω_l, a_k) and pair (ω_k, a_k) for calculating error of publications at $\mathcal{M}_k^1$ and $\mathcal{M}_k^2$ respectively. If publication does not occur at timestamp k, then the mechanism $\mathcal{M}$ calculates the MAE metric of pair (ω_l, a_k) as a error of publications at timestamp k. Therefore, the mechanism $\mathcal{M}$ produces error per timestamps from one or both of the phases. Next, we show that the average error per timestamp in the PBA mechanism. Assume that there is an equal number of skipped publications between every occurrence of true publications, and n represents the total number of true publications that occur within a sliding window of size w.

Theorem 2 *The average error per timestamp in PBA is at most* $\frac{4}{n\epsilon} \cdot (2(2^n - 1) + 1) + \frac{2w}{\epsilon|L|}$

Proof At timestamp k, the private dissimilarity value of $\mathcal{M}_k^1$ guides $\mathcal{M}_k^2$ for deciding to publish either true publication or null publication. Hence, we consider both $\mathcal{M}_k^1$ and $\mathcal{M}_k^2$ for computing average error per timestamp in PBA mechanism. The $\mathcal{M}_k^1$ induces error when its private dissimilarity value suggests $\mathcal{M}_k^2$ to make a wrong decision, i.e., wrongly skips a publication or wrongly performs true publication. If $\mathcal{M}_k^1$ suggest true publication occur at time k, then the error at timestamp k is the error induced by $\mathcal{M}_k^2$, which is discussed later. Alternatively, if $\mathcal{M}_k^1$ suggest true skipped publication occur at time k, then the error at timestamp k is an original dissimilarity value between ω_l and a_k, it is bounded by the error of $\mathcal{M}_k^2$. However, if wrongly performs true publication at time k, then the original dissimilarity value is overrated due to noise of scale λ_k^1 added by the $\mathcal{M}_k^1$. Conversely, if wrongly skips a publication at time k, then the original dissimilarity value is underrated due to noise with scale λ_k^1 added by the $\mathcal{M}_k^1$. Therefore, the error induced by the $\mathcal{M}_k^1$ in PBA is at most $\frac{2w}{\epsilon|L|}$. Recall that PBA allocates privacy budget in exponential decreasing fashion in phase 2 of $\mathcal{M}$ i.e., $\epsilon/8, \epsilon/16, \epsilon/32, ...$ Hence, the error per each timestamp in phase 2 is $1/\epsilon_r$, where ϵ_r is a exponentially decreasing privacy budget. Moreover, $\mathcal{M}^2$ phase uses the extra budget from the absorbed privacy budget, so the error at $\mathcal{M}^2$ from the absorbed privacy budget is at most $(4/\epsilon)$. Therefore, the average error per timestamps within a sliding window of $\mathcal{M}^2$ in PBA is equal to

$$
\begin{aligned}
&= \tfrac{1}{n} \cdot (\tfrac{8}{\epsilon} + \tfrac{16}{\epsilon} + \cdots + \tfrac{2^{n+2}}{\epsilon}) + \tfrac{4}{n\epsilon} \\
&= \tfrac{8}{n\epsilon} \cdot (2^n - 1) + \tfrac{4}{n\epsilon} \\
&= \tfrac{4}{n\epsilon} \cdot (2(2^n - 1) + 1)
\end{aligned}
$$

The total average error per timestamps within the sliding window of size w in PBA mechanism is

$$= \frac{4}{n\epsilon} \cdot (2(2^n - 1) + 1) + \frac{2w}{\epsilon|L|}$$

□

5 Experimental results

In this section, we conduct an experiment to demonstrate users' privacy risk, especially when the datasets have temporal-correlation at certain timestamps. Furthermore, we validate the effectiveness of our proposed algorithm with existing states of art methods.

We employed three real-time trajectory datasets such as $Geolife$ [17], $T-Drive$ [18, 19], $ShangHai$ [20], and a synthetic dataset, namely $Metro100K$ [21] in

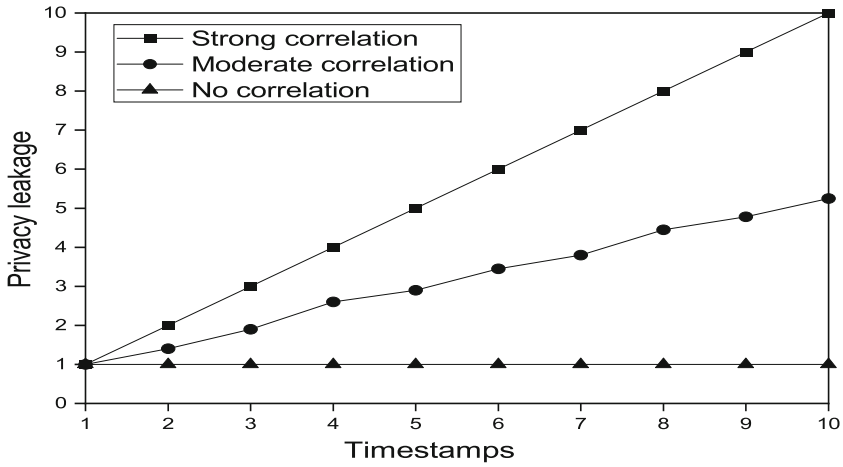

Fig. 4 Analysis of privacy risk (or leakage) of 1-DP under different types of temporal correlation

our experiments. A $Geolife$ dataset is a real-time GPS trajectory dataset collected from 182 users over three years. The average frequency of data (location value) collection from the users is about 35 seconds. A $T-Drive$ dataset contains the GPS trajectories of 10357 taxis over one year. The average sampling interval of data collection from users is about 177 seconds. A $ShangHai$ dataset is a public trajectory data of about 5000 buses and taxis in Shanghai collected by the Hong Kong University of Science and Technology on February 20, 2007. The sampling interval of data is approximately 60 seconds. Finally, a $Metro100K$ dataset is a synthetic dataset that contains 100000 trajectories gathered from a metropolitan area with 26 cities in 24 hours. We optimized all real-time datasets for our experiment by considering that a user is located at most one location at each timestamp and collected all samples (user's location data-point) every 5 minutes. The above trajectory datasets consist of a series of tuples containing ID, timestamp, latitude, and longitude. We filter the real-time datasets using two minimum requirements, i.e., each user is located at most one location at each timestamp, and collect samples from the datasets according to the curator's pre-defined timestamp frequency (ex: every 5 min).

According to our problem settings, we train the Markov model for modeling a transition probability between all possible location data-points. This transition probability matrix describes how a data-point is dependent on other possible remaining data-points. There are three types of temporal correlation, such as strong, moderate, and no correlation. Let assume that the temporal correlation between the data-points of a user stream is strong, i.e., $\theta^s = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The privacy-risk of users per timestamp is increasing linearly because the same datasets are released in contiguous timestamps, as shown in Fig. 4. In other words, the location data-points of users are at time t to time 1 is equivalent, i.e., $loc^t = loc^{t-1} = \cdots = loc^1$. Hence, the privacy risk is increasing linearly at each timestamp. In another extreme case, is a moderate temporal-correlation between the data-points, say $\theta^m = \begin{bmatrix} 0.8 & 0.2 \\ 0 & 1 \end{bmatrix}$. The line with circle shape in Fig. 4 shows users' privacy risk from timestamp 1 to t, which can be quantified by using Eq. 7. Finally, a line with a triangle shape in Fig. 4 shows each timestamp achieves 1-DP while assuming $\epsilon = 1$ because no temporal-correlation exists.

Further, we demonstrate the effectiveness on TC privacy leakage when the transition probability matrix involves a large (or small) number of dimensions. Let consider a transition probability matrix with a moderate correlation and d be the number of dimensions in the transition matrix. If d is large, then the probability value on cells is well scattered in transition matrix. Figure 5 shows the variation of privacy leakage when the size of d varies in the transition probability matrix. The results show that the ϵ-DP attains less privacy leakage when the vast number of dimensions in the transition probability matrix. It is depicted in the lines $d = 50$ and $d = 10$ of Fig. 5 by considering $\epsilon = 1$. This is because the data points in a matrix are very close to the stronger correlation. In other words, a stronger correlation in the transition matrix results in more privacy leakage. The transition matrix involves a weaker correlation when the matrix dimension is larger, as shown in the lines $d = 200$, $d = 50$ of Fig. 5.

5.1 Utility evaluation

We used two popular metrics such as *Mean of absolute error(MAE)* and *Mean of square error(MSE)* [22] for quantifying the data utility of our mechanism with existing states of art methods. These two metrics are used to measure the dissimilarity (or error) between the measured value and actual value. Moreover, these two metrics have good mathematical properties, and also the MSE metric helps to find larger errors. In our settings, the PBA method protects each user's data point per timestamp (or user stream) within

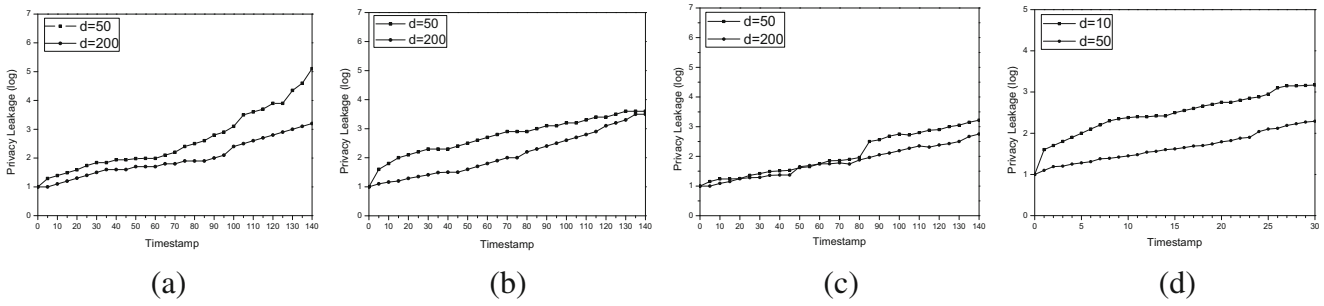

Fig. 5 Privacy leakage versus different degrees of correlation while set $\epsilon = 1$ (**a**) Geolife (**b**) T-Drive (**c**) ShangHai and (**d**) Metro100K datasets

the sliding window of size w. Hence, we measure the error per timestamp by using MAE and MSE metrics. The definition of MAE and MSE metric is as follows.

$$MAE() = \frac{1}{|T|}\frac{1}{|L|}\Sigma_{j=1}^{|L|}|\omega_l[j] - a_t[j]|$$

$$MSE() = \frac{1}{|T|}\frac{1}{|L|}\Sigma_{j=1}^{|L|}|\omega_l[j] - a_t[j]|^2$$

5.1.1 Baseline approaches

Firstly, we started to compare our proposed method with baseline approaches such as Uniform, Sampling, and w-event privacy, to analyze the effectiveness of our method while varying the size of the sliding window w and ϵ value. Figures 6 and 7 show the result of MAE and MSE values between the PBA method with baseline approaches, while varying the size of the sliding window w. We observed that our PBA method outperforms with baseline approaches on all datasets. This is because the rate of allotted privacy budget within the sliding window is minimized in baseline approaches due to an increase in the number of timestamps in the sliding window. In other words, the adequate amount of privacy budget is allotted at temporally correlated timestamps in the PBA method compared to baseline approaches. Notice that MAE and MSE value in uniform method increases linearly when w increases because fixed privacy budget allotted at each timestamp. Similarly, a privacy budget is allotted only at a given sample interval in a sampling approach. Furthermore, the MAE values in PBA and BD methods (see Fig. 6b, c and d) are approximately the same because both methods follow the same allocation scheme, i.e., exponential decreasing fashion. However, our PBA method achieves ϵ-DP even though the location data-points are temporally correlated at consecutive timestamps.

Further, we compared our PBA method with baseline approaches while varying ϵ values, as shown in the Figs. 8 and 9. The result shows that the error rate of MRE and MSE is comparatively low while assigning a larger privacy budget at each timestamp. The MRE and MSE values of baseline approaches have more update errors while comparing our PBA method. This is because the uniform and sample methods use a fixed privacy budget at each timestamp even-though the location data-points are temporally correlated. Further, the error rate of the PBA and BD approach has almost similar because the PBA adopts a similar allocation scheme (i.e., exponential decreasing fashion at stage 2) as

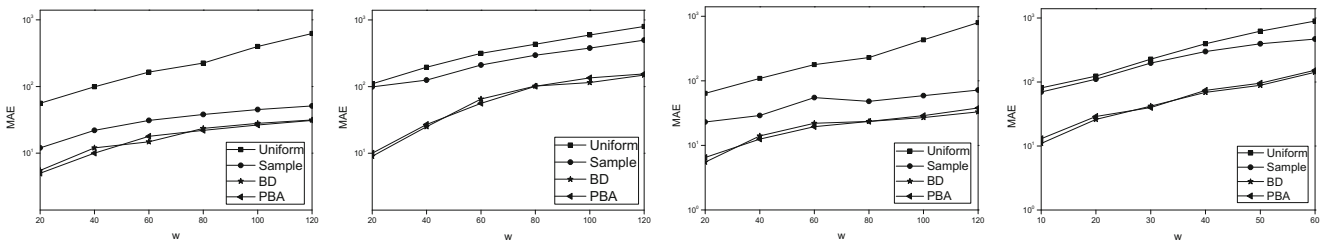

Fig. 6 MAE vs. w while fixing ϵ=1 (**a**) Geolife (**b**) T-Drive (**c**) ShangHai and (**d**) Metro100K datasets

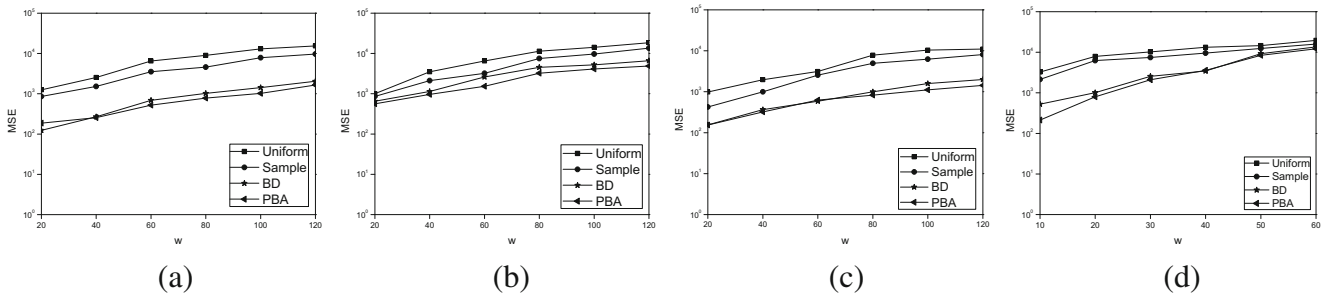

Fig. 7 MSE vs. w while fixing ϵ=1 (**a**) Geolife (**b**) T-Drive (**c**) ShangHai and (**d**) Metro100K datasets

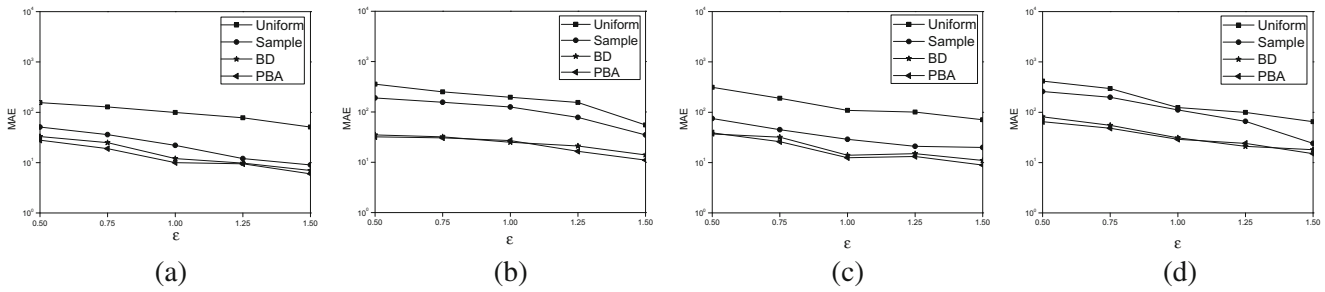

Fig. 8 MAE vs. ϵ while fixing w=40 (**a**) Geolife (**b**) T-Drive (**c**) ShangHai and (**d**) Metro100K datasets

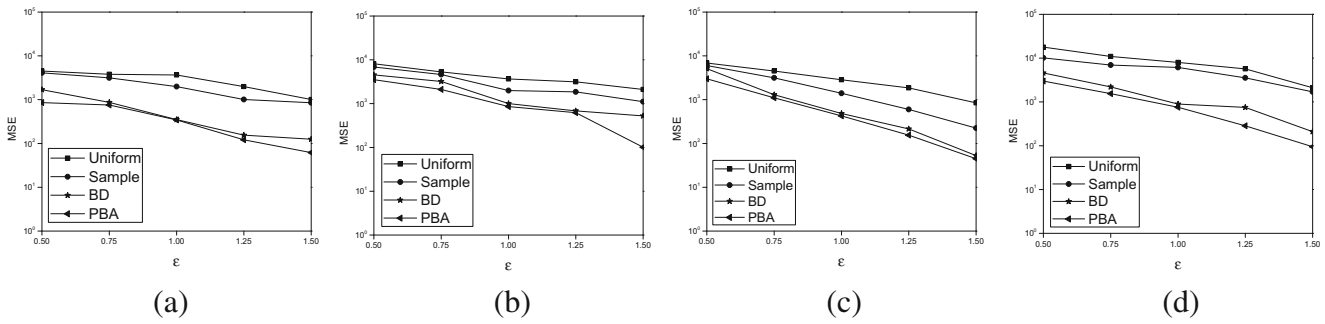

Fig. 9 MSE vs. ϵ while fixing w=40 (**a**) Geolife (**b**) T-Drive (**c**) ShangHai and (**d**) Metro100K datasets

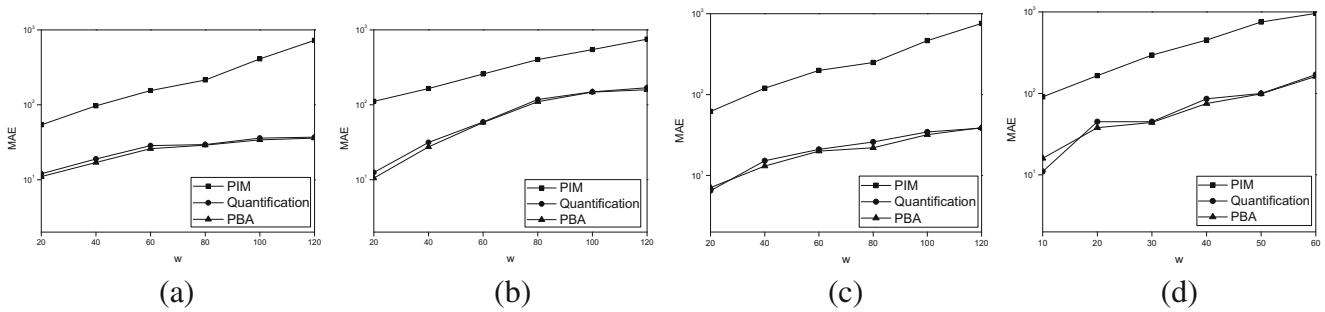

Fig. 10 MAE vs. w while fixing ϵ=1 (**a**) Geolife (**b**) T-Drive (**c**) ShangHai and (**d**) Metro100K datasets

in the BD approach. However, our PBA method allocates an adequate amount of privacy budget even though the location data-points are temporally correlated at consecutive timestamps.

5.1.2 Comparison

In literature, several works have been proposed that consider the correlation between the users in the dataset, i.e., user-user correlation, whereas we consider a correlation among the single user's location data-points at different timestamps (i.e., temporal correlation). We found limited research on finding the privacy risk of differential privacy under temporal correlation for the continuous location data release settings. However, Quantification [14] and Planar Isotropic Mechanisms (PIM) [33] are the recent privacy budget allocation methods for temporal correlation in continuous location data release settings. Figures 10 and 11 show the error rate of MAE and MSE between our PBA method with other states of art methods such as Quantification and PIM while varying the size of the sliding window w. The experimental results exhibit that the proposed PBA method provides significant data utility compared to the above two methods. The error rate of the quantification and PBA method are approximately similar. This is because the proposed method achieves ϵ-DP under a temporal correlation whereas the quantification method achieves α-DP under temporal correlation instead of ϵ-DP, where α is increased privacy leakage of range $\epsilon \leq \alpha \leq T\epsilon$ (assume that the length of temporally correlated data-points in user's stream is T). In other words, the quantification method allocates more privacy budget (i.e., exceeds than the allotted budget of traditional ϵ-DP) to each timestamps. Hence, the error rate (MAE and MSE) of quantification is almost same to the PBA method under temporal correlation. And since the privacy budgets are assigned to each timestamps uniformly in PIM, the error rate (MAE and MSE) of PIM increases linearly, as shown in Figs. 10 and 11.

Figures 12 and 13 show the error rate of MAE and MSE between our PBA method with Quantification and PIM while varying ϵ values. The result shows that the MRE and MSE values decreasing when assigning more privacy budget at each timestamp while setting w=40. The MAE and MSE of the PBA method are closely related to the quantification method because the quantification method uses an increased privacy budget α, which leads to fewer update errors even though the user's stream involves temporal correlation. Another side, the PIM method follows a uniform approach for allocating privacy budgets that leads to having more update errors, as shown in Figs. 12 and 13.

Table 3 shows that the privacy guarantee of various privacy budget allocation methods on temporally correlated data-points of length T. We observed that our PBA method achieves ϵ-DP under temporal correlation in continuous stream data publishing compared with other DP approaches under temporal correlation. The existing approaches such as Uniform, Sampling, BD, PIM, and the Quantification method are provided less privacy guarantee under a temporal correlation, i.e., Tϵ-DP, (T/I)ϵ-DP, $w\epsilon$-DP, Tϵ-DP and α-DP respectively. In other words, the existing approaches require more privacy budgets ($\geq \epsilon$) in continuous temporally correlated stream data publishing to satisfy the definition of ϵ-DP. Even though BD and PBA methods follow the same allocation strategy, BD achieves $w\epsilon$-DP (assume that the length of temporally correlated data-points in the user's stream is w) instead of ϵ-DP under temporal correlation. Hence, the PBA method is a better approach for allocating privacy budgets to temporally correlated data-points.

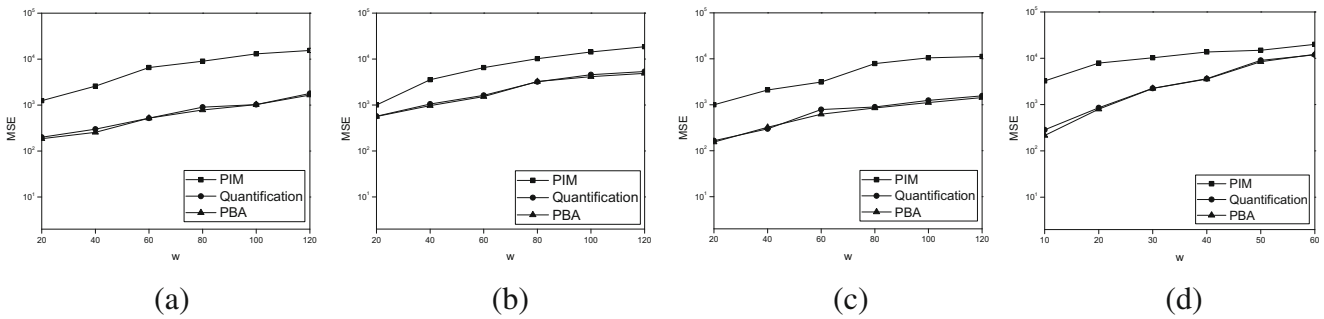

Fig. 11 MSE vs. w while fixing ϵ=1 (**a**) Geolife (**b**) T-Drive (**c**) ShangHai and (**d**) Metro100K datasets

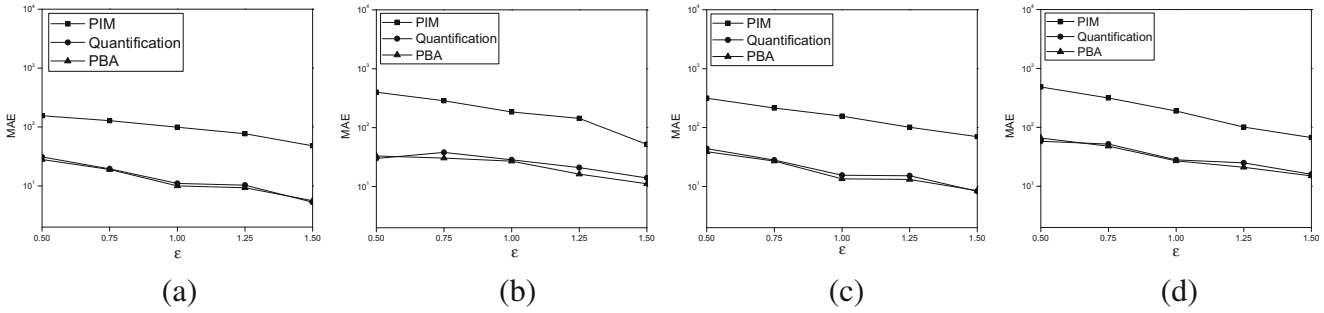

Fig. 12 MAE vs. ϵ while fixing w=40 (**a**) Geolife (**b**) T-Drive (**c**) ShangHai and (**d**) Metro100K datasets

6 Related work

Dwork et al. [15, 23] first initiated and proposed two privacy approaches such as event-level and user-level privacy for studying differential privacy under continual observation. This study is beneficial to investigate different problems related to continuous data release settings, such as infinite stream publishing [24], real-time data publishing [25], and high-dimensional data publication [26, 27]. However, Kifer et al. [34] claim that differential privacy may not provide a privacy guarantee if the dataset contains a correlation between users.

There exist a few ϵ-DP approaches to address the correlation between users in the dataset. Kifer et al. [35] first initiated this privacy issue and proposed a *pufferfish* framework, which requires three components that need to be explicitly specified: potential secrets, discriminative pairs, and data generation. Song et al. [36] present the Markov Quilt mechanism when the correlation can frame by Bayesian Network. Yang et al. [37] proposed a general perturbation algorithm that is ϵ-DP for prior knowledge of any subset of records in the dataset when the records are correlated. Zhu et al. [39] present an effective correlated differential privacy mechanism by defining the correlated sensitivity. This sensitivity significantly decreases the noise compared with traditional global sensitivity. Liu et al. [38] proposed a dependent perturbation mechanism against probabilistic dependence between users. It uses a dependence coefficient to find accurate query sensitivity for dependent data, leading to better data-utility at the same privacy level. Wu et al. [40] present a game-based definition of correlated differential privacy to evaluate the privacy level of a single user's record influenced by the other user. The above privacy methods deal with the correlation between the user's records in the dataset, i.e., user-user correlation, whereas, in our settings, we consider the correlation among single user's data at different timestamps, i.e., temporal-correlation.

On the other hand, to the best of our knowledge, there is very limited research on the analysis of privacy risk of differential privacy under temporal correlation in continuous location data release settings. Xiao et al. [33] propose a planar isotropic mechanism (PIM) for location perturbation at each timestamp, and it achieves differential privacy against the adversaries who have knowledge of temporal correlation in location datasets. Cao et al. [14] define a α-differential privacy for providing privacy guarantee against an adversary with knowledge of temporal correlation in the continuous data release settings.

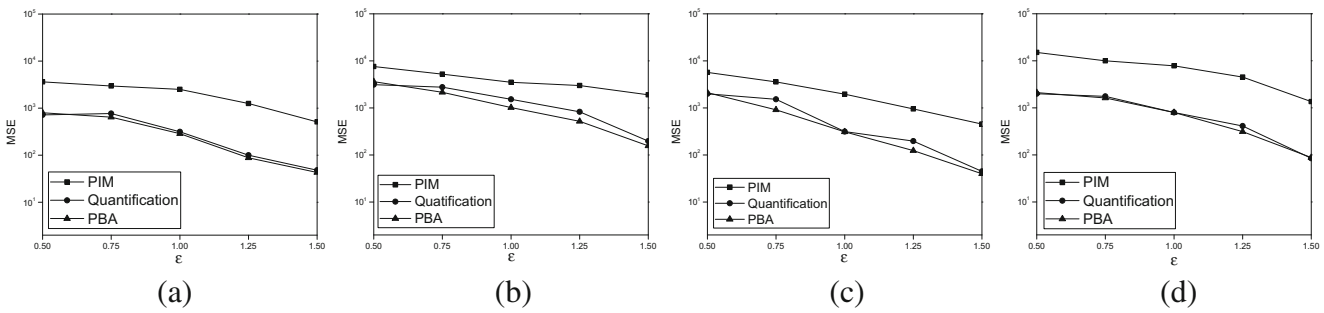

Fig. 13 MSE vs. ϵ while fixing w=40 (**a**) Geolife (**b**) T-Drive (**c**) ShangHai and (**d**) Metro100K datasets

Table 3 The privacy guarantee of various privacy budget allocation methods on temporally correlated data-points of length T

Privacy approaches	Privacy Budget Allocation scheme	Temporal correlation	privacy guarantee on T length user stream
Uniform [24]	✓	–	Tϵ-DP
Sampling [25]	✓	–	(T/I)ϵ-DP
Budget Distribution (BD) [13]	✓	–	$w\epsilon$-DP
PIM [33]	–	✓	Tϵ-DP
Quantification [14]	✓	✓	α-DP
Proposed method	✓	✓	ϵ-DP (i.e., $\Sigma_{j=k-w+1}^{k}\epsilon_j \leq \epsilon$)

7 Conclusion

In this paper, we present the definition of differential privacy under temporal correlation to quantify the impact of temporal correlation on privacy leakage. Then, we illustrate and prove that the adversaries who have knowledge of temporal correlation can disclose more privacy leakage than the traditional ϵ-DP. Our analysis result shows that the privacy leakage increases over time in w-event privacy when the dataset involves temporal correlation. Therefore, we introduce a privacy budget allocation (PBA) method for allocating an adequate amount of privacy budget to each successive timestamp under the protection of ϵ-differential privacy. This method protects any w length user stream that contains temporally correlated data-points. We conducted a series of experiments with real and synthetic datasets to evaluate the average error per timestamp for analyzing the data utility of our method. As future work, we are interested in investigating the $\mathcal{TC}$ privacy leakage under all types of correlation models and also quantify the impact of temporal correlation on privacy leakage in ϵ-Local Differential privacy.

Appendix

Theorem 3 *Prove that*

$$\sum_{j=k-w+1}^{k}(\epsilon/4 - [\Sigma_{j=k-w+1}^{k-1}\epsilon_j])/2 \leq \epsilon/4$$

.

Proof Sub-mechanism $\mathcal{M}^2$ allocates a privacy budgets per timestamps in exponential decreasing fashion i.e., $(\epsilon/8, \epsilon/16, \epsilon/32, ...)$. Then LHS is,

$$\sum_{j=k-w+1}^{k}(\epsilon/4 - \Sigma_{j=k-w+1}^{k-1}\epsilon_j)/2 = (\frac{\epsilon}{8} + \frac{\epsilon}{16} + \cdots + \frac{\epsilon}{2^{n+2}})$$

$$= \Sigma_{i=3}^{n}\frac{\epsilon}{2^i} \leq \epsilon/4$$

We can rewrite

$\Sigma_{i=3}^{n}\frac{\epsilon}{2^i} \leq \epsilon/4$ into $\Sigma_{i=1}^{n}\frac{\epsilon}{2^i} \leq \epsilon$

Let assume that $\epsilon = 1$

$$\Sigma_{i=1}^{n}\frac{1}{2^i} = \Sigma_{i=1}^{1}(\frac{1}{2})(\frac{1}{2})^{i-1} = \frac{1}{2}\frac{(1-(\frac{1}{2})^i)}{(1-(\frac{1}{2}))}$$

By mathematical induction:
Basis: $n = 1$

$$\Sigma_{i=1}^{1}(\frac{1}{2})(\frac{1}{2})^{1-1} = \frac{1}{2}\frac{(1-(\frac{1}{2})^1)}{(1-(\frac{1}{2}))} \leq 1$$

Inductive step: Assume true for $n = p$

$$\Sigma_{i=1}^{p}(\frac{1}{2})(\frac{1}{2})^0 + (\frac{1}{2})(\frac{1}{2})^1 + \cdots + (\frac{1}{2})(\frac{1}{2})^{p-1}$$

$$= \frac{1}{2}\frac{(1-(\frac{1}{2})^p)}{(1-(\frac{1}{2}))} \leq 1$$

Prove true for $n = p + 1$

$$\Sigma_{i=1}^{p+1}(\frac{1}{2})(\frac{1}{2})^0 + (\frac{1}{2})(\frac{1}{2})^1 + \cdots + (\frac{1}{2})(\frac{1}{2})^{p-1} + (\frac{1}{2})(\frac{1}{2})^p$$

$$= \frac{1}{2}\frac{(1-(\frac{1}{2})^{p+1})}{(1-(\frac{1}{2}))}$$

$$= \frac{1}{2}\frac{(1-(\frac{1}{2})^p)}{(1-(\frac{1}{2}))} + (\frac{1}{2})(\frac{1}{2})^p$$

$$= \frac{1}{2}\frac{(1-(\frac{1}{2})^p + (\frac{1}{2})^p - (\frac{1}{2})^{p+1})}{(1-(\frac{1}{2}))}$$

$$= \frac{1}{2}\frac{(1-(\frac{1}{2})^{p+1})}{(1-(\frac{1}{2}))} = \frac{1}{2}\frac{(1-(\frac{1}{2})^{p+1})}{(1-(\frac{1}{2}))} \leq 1$$

□

References

1. Schneider MJ, Jagpal S, Gupta S, Li S, Yan Y (2017) Protecting customer privacy when marketing with second-party data. Int J Res Mark 34:593–603
2. Lane ND, Mohammod M, Lin M, Yang X, Lu H, Ali S, Doryab A, Berke E, Choudhury T, Campbell A (2011) Bewell: a smartphone application to monitor, model and promote wellbeing. In: 5th International ICST conference on pervasive computing technologies for healthcare, pp 23–26
3. Thiagarajan A, Ravindranath L, LaCurts K, Madden S, Balakrishnan H, Toledo S, Eriksson J (2009) VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In: Proceedings of the 7th ACM conference on embedded networked sensor systems. ACM, pp 85–98
4. Guha S, Reznichenko A, Tang K, Haddadi H, Francis P (2009) Serving Ads from localhost for performance, privacy, and profit. HotNets
5. Malathi D, Logesh R, Subramaniyaswamy V, Vijayakumar V, Sangaiah AK (2019) Hybrid reasoning-based privacy-aware disease prediction support system. Comput Electr Eng 73:114–127
6. Chow C-Y, Mokbel MF (2011) Trajectory privacy in location-based services and data publication. ACM Sigkdd Explor Newslett 13(1):19–29
7. Dong Y, Pi D (2018) Novel privacy-preserving algorithm based on frequent path for trajectory data publishing. Knowl-Based Syst 148:55–65
8. Aggarwal CC, Philip SY (2008) A general survey of privacy-preserving data mining models and algorithms. In: Privacy-preserving data mining. Springer, pp 11–52
9. Dwork C (2011) Differential privacy. Encyclopedia of cryptography and security. Springer, pp 338–340
10. Sweeney L (2002) k-anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. World Scientific, pp 557–570
11. Hemkumar D, Ravichandra S, Somayajulu DVLN (2020) Impact of prior knowledge on privacy leakage in trajectory data publishing. Engineering Science and Technology, an International Journal. Elsevier
12. Fan L, Xiong L, Sunderam V (2013) Fast: differentially private real-time aggregate monitor with filtering and adaptive sampling. In: Proceedings of the 2013 ACM SIGMOD international conference on management of data. ACM, pp 1065–1068
13. Kellaris G, Papadopoulos S, Xiao X, Papadias D (2014) Differentially private event sequences over infinite streams. Proceedings of the VLDB Endowment, 1155–1166
14. Cao Y, Yoshikawa M, Xiao Y, Xiong L (2018) Quantifying differential privacy in continuous data release under temporal correlations. IEEE Trans Knowl Data Eng 31(7):1281–1295
15. Dwork C (2010) Differential privacy in new settings. In: Proceedings of the twenty-first annual ACM-SIAM symposium on discrete algorithm. SIAM, pp 174–183
16. Fan L, Xiong L (2012) Real-time aggregate monitoring with differential privacy. In: Proceedings of the 21st ACM international conference on information and knowledge management. ACM, pp 2169–2173
17. Zheng Y, Zhang L, Xie X, Ma W-Y (2009) Mining interesting locations and travel sequences from gps trajectories. In: Proceedings of the 18th international conference on world wide web. ACM, pp 791–800
18. Yuan J, Zheng Y, Xie X, Sun G (2011) Driving with knowledge from the physical world. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 316–324
19. Yuan J, Zheng Y, Zhang C, Xie W, Xie X, Sun G, Huang Y (2010) T-drive: driving directions based on taxi trajectories. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. ACM, pp 99–108
20. Cocchia A (2014) Smart and digital city: a systematic literature review. In: Smart city. Springer, pp 13–43
21. Mohammed N, Fung B, Debbabi M (2010) Preserving privacy and utility in rfid data publishing
22. Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geoscientific model development. Copernicus GmbH, pp 1247–1250
23. Dwork C, Naor M, Pitassi T, Rothblum GN (2010) Differential privacy under continual observation. In: Proceedings of the forty-second ACM symposium on theory of computing, pp 715–724
24. Cao Y, Yoshikawa M (2016) Differentially private real-time data publishing over infinite trajectory streams. IEICE Transactions on Information and Systems, pp 163–175
25. Li H, Xiong L, Jiang X, Liu J (2015) Differentially private histogram publication for dynamic datasets: an adaptive sampling approach. In: Proceedings of the 24th ACM international on conference on information and knowledge management. ACM, pp 1001–1010
26. Acs G, Castelluccia C (2014) 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. ACM, pp 1679–1688
27. Xiao Y, Xiong L, Fan L, Goryczka S (2012) DPCube: differentially private histogram release through multidimensional partitioning. arXiv:1202.5358
28. Chan T-HH, Chan ES, Song D (2011) Private and continual release of statistics. ACM Transactions on information and system security (TISSEC). ACM
29. Bolot J, Fawaz N, Muthukrishnan S, Nikolov A, Taft N (2013) Private decayed predicate sums on streams. In: Proceedings of the 16th international conference on database theory. ACM, pp 284–295
30. Fan L, Bonomi L, Xiong L, Sunderam V (2014) Monitoring web browsing behavior with differential privacy. In: Proceedings of the 23rd international conference on world wide web. ACM, pp 177–188
31. Mir D, Muthukrishnan S, Nikolov A, Wright RN (2011) Pan-private algorithms via statistics on sketches. In: Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems. ACM, pp 37–48
32. Jiang K, Shao D, Bressan S, Kister T, Tan K-L (2013) Publishing trajectories with differential privacy guarantees. In: Proceedings of the 25th International conference on scientific and statistical database management, pp 1–12
33. Xiao Y, Xiong L (2015) Protecting locations with differential privacy under temporal correlations. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. ACM, pp 1298–1309
34. Kifer D, Machanavajjhal A (2011) No free lunch in data privacy. In: Proceedings of the 2011 ACM SIGMOD international conference on management of data. ACM, pp 193–204
35. Kifer D, Machanavajjhala A (2012) A rigorous and customizable framework for privacy. In: Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on principles of database systems. ACM, pp 77–88
36. Song S, Wang Y, Chaudhuri K (2017) Pufferfish privacy mechanisms for correlated data. In: Proceedings of the 2017 ACM international conference on management of data. ACM, pp 1291–1306

37. Yang B, Sato I, Nakagawa H (2015) Bayesian differential privacy on correlated data. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data. ACM, pp 747–762
38. Liu C, Chakraborty S, Mittal P (2016) Dependence makes you vulnberable: differential privacy under dependent tuples. In: NDSS, pp 21–24
39. Zhu T, Xiong P, Li G, Zhou W (2014) Correlated differential privacy: hiding information in non-IID data set. IEEE Transactions on Information Forensics and Security. IEEE, pp 229–242
40. Xiaotong W, Wu T, Maqbool K, Qiang N, Wanchun D (2017) Game theory based correlated privacy preserving analysis in big data. IEEE Transactions on Big Data. IEEE
41. Cao Y, Yoshikawa M (2015) Differentially private real-time data release over infinite trajectory streams. In: 2015 16th IEEE international conference on mobile data management. IEEE, pp 68–73
42. Ma Z, Zhang T, Liu X, Li X, Ren K (2019) Real-time privacy-preserving data release over vehicle trajectory. IEEE Trans Veh Technol 68(8):8091–8102

D. Hemkumar has received his Masters degree from National Institute of Technology Karnataka (NITK), India in the year 2012. He is currently Ph.D. student in the Department of Computer Science and Engineering of National Institute of Technology Warangal (NITW), India. He has previously worked in several universities as assistant professor. He has published two journal papers in the area of privacy preserving data publishing. His research interests include Differential Privacy, Data mining and information retrieval.

S. Ravichandra has received his Ph.D. in computer science from National Institute of Technology (NIT) Warangal, India in the year 2015. He is currently an associate professor in the department of computer science and engineering, National Institute of Technology (NIT) Warangal, Telangana, India. He has published several papers in premium conferences and journals in his research area. His research interests include Software Engineering, Software Architecture, Design Patterns, Privacy Preserving Data Publishing and Service Oriented Architecture. He is a member of IEEE and ISTE.

D. V. L. N. Somayajulu has received his Ph.D. in computer science from Indian Institute of Technology Delhi, India and serving as Professor of Computer Science & Engineering since 2006. Currently, he is Director, Indian Institute of Information Technology, Design and Manufacturing, Kurnool, Andhra Pradesh (MHRD -CFTI) on deputation since Feb 28, 2019. He has published several papers in premium conferences and journals in his research area. His research interests include Databases, Information Extraction, Query Processing, Big Data and Privacy.