Towards Pattern-aware Privacy-preserving Real-time Data Collection

Zhibo Wang[†], Wenxin Liu[†,≀], Xiaoyi Pang[†], Ju Ren[‡,♯,*], Zhe Liu[¶], and Yongle Chen[§]
[†]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University, P. R. China
[≀]Shaanxi Key Laboratory of Network and System Security, Xidian University, P. R. China
[‡]Department of Computer Science and Technology, Tsinghua University, P. R. China
[♯]School of Computer Science and Engineering, Central South University, P. R. China
[¶]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, P. R. China
[§]College of Information and Computer, Taiyuan University of Technology, P. R. China
Email: {zbwang, wxliu111, xypang}@whu.edu.cn, renju@csu.edu.cn, sduliuzhe@gmail.com, chenyongle@tyut.edu.cn

***Abstract*—Although time-series data collected from users can be utilized to provide services for various applications, they could reveal sensitive information about users. Recently, local differential privacy (LDP) has emerged as the state-of-art approach to protect data privacy by perturbing data locally before outsourcing. However, existing works based on LDP perturb each data point separately without considering the correlations between consecutive data points in time-series. Thus, the important patterns of each time-series might be distorted by existing LDP-based approaches, leading to severe degradation of data utility.**

In this paper, we focus on real-time data collection under a honest-but-curious server, and propose a novel pattern-aware privacy-preserving approach, called PatternLDP, to protect data privacy while the pattern of time-series can still be preserved. To this end, instead of providing the same level of privacy protection at each data point, each user only samples remarkable points in time-series and adaptively perturbs them according to their impacts on local patterns. In particular, we propose a pattern-aware sampling method based on Piecewise Linear Approximation (PLA) to determine whether to sample and perturb current data point. To reduce the utility loss caused by pattern change after perturbation, we propose an importance-aware randomization mechanism to adaptively perturb sampled data locally while achieving better trade-off between privacy and utility. A novel metric-based w-event privacy is introduced to measure the privacy protection degree for pattern-rich time-series. We prove that PatternLDP can provide the above privacy guarantee, and extensive experiments on real-world datasets demonstrate that PatternLDP outperforms existing mechanisms and can effectively preserve the important patterns.

***Index Terms*—Local differential privacy, time-series, real-time data collection, pattern preservation**

I. Introduction

With the popularity of mobile devices (e.g., smartphone, smartwatch) and the development of crowdsourcing applications, service providers can easily collect data from users and utilize them to provide various services. With time-series data generated from users, it is possible to discover useful information that are of great benefits to both groups and individuals. For the *group*, a service provider aggregates data of each group at a certain moment and calculates the statistical results to obtain group characteristics, such as traffic monitoring and urban noise mapping. For the *individual*, time-series data of each user can be analyzed to recognize useful patterns and then personalized services can be provided to him/her, such as monitoring a patient's condition to predict his risk of some disease. However, the adjunct to the benefits is a serious privacy breach. As the personal data outsourced from mobile devices to service providers, a lot of personal sensitive information would be exposed. For example, a malicious service provider can utilize the medical information of a user to send targeted ads or even unjustly determine whether to accept the renewal of his insurances and labor contracts [1].

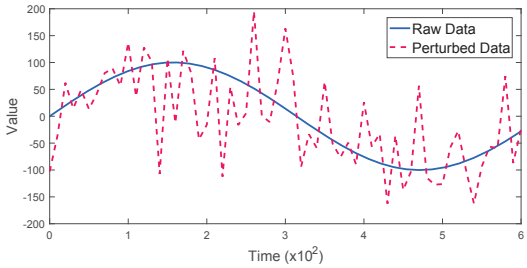


Fig. 1. Pattern change by perturbation.

Most of the privacy-preserving research on time-series focuses on data publishing, which considers the scenario of centralized databases, where a trustable data curator is assumed and users' private data are collected by it. The data curator aggregates the time-series data from each user and sanitizes them uniformly (e.g., via data anonymization techniques as in [2] [3], differential privacy technique as in [4] [5]) before publishing the dataset. However, the assumption of trustable data curators is not always true, since service providers may trade user's privacy for commercial interests or be attacked by malicious agents. Recently, LDP has emerged as the de facto notion for privacy protection in the local setting, where users take full control of their own data without the trust of any party. Existing works on LDP mainly focus on solving two problems. One is how to design suitable LDP mechanisms for different data types generated by users, such as numerical data [6] [7], categorical data [8] [9], set-valued data [10] [11]

*Ju Ren is the corresponding author.

and key-value data [12]. The other is how to design suitable LDP mechanisms for different analysis purposes, such as mean estimation [6] [7] and frequency estimation [13] [14].

However, an important issue is that existing LDP-based privacy preservation approaches perturb each data point separately without considering the correlations between consecutive data points in time-series. The patterns of time-series data, characterized by the correlations of data points, are crucial for personalized services analysis. It is highly possible that some significant patterns of time-series from a user are changed or even destroyed after data perturbation. As shown in Figure 1, the time-series data, which is perturbed by the Laplace noise [5], has completely failed to exhibit the sinusoidal pattern. In this case, although the aggregate statistic over multiple users might be accurate at each time point, the time-series data of each user has lost its *pattern utility* and cannot be utilized to provide personalized services. Therefore, it is urgently necessary to *design a novel privacy-preserving real-time data collection approach to protect users' privacy while the patterns of the time-series data can still be preserved.*

In this paper, we focus on real-time data collection against the honest-but-curious curator, and aim to protect each user's privacy while preserving the pattern utility. To this end, we are facing several nontrivial challenges. The first challenge is *how to determine whether a data point is a remarkable point or not in the real-time data collection scenario*. Since perturbing each data points results in bad utility, we intend to only perturb remarkable points (also referred to as key-points [15], break-points [16] and change-points [17]), which can be thought as those points in the time-series that are obviously remarkable (peaks, valleys and so on) and can effectively and efficiently represent the unique patterns. However, it is difficult to find out these remarkable points without global time-series. The second challenge is *how to achieve a good tradeoff between pattern utility and privacy*. Preserving patterns of time-series means violating users' privacy to some extent, while larger perturbation usually means better privacy protection but more pattern loss. It would be difficult to preserve useful pattern information of time-series while satisfying the required privacy protection requirement.

To address the above challenges, in this paper, we propose a novel pattern-aware privacy-preserving real-time data collection approach, called PatternLDP, to protect the sensitive time-series while the patterns in time-series can still be effectively preserved. To this end, instead of providing the same level of privacy protection at each data point, each user only samples remarkable points in time-series and adaptively perturbs them according to their impacts on local patterns. In particular, we propose a pattern-aware sampling method based on PLA to determine whether to sample and perturb current data point. To reduce the pattern loss, we propose an importance-aware randomization mechanism to adaptively perturb sampled data locally while achieving better trade-off between privacy and utility. A novel metric-based w-event privacy is introduced to measure the privacy protection degree for pattern-rich time-series. Our main contributions are summarized as follows:

- To the best of our knowledge, this is the first work to preserve both privacy and pattern utility of time-series in real-time data collection scenario. A novel pattern-aware privacy-preserving real-time data collection approach is proposed to preserve patterns of time-series while satisfying privacy protection requirement.
- By taking the importance of remarkable points into consideration, a pattern-aware sampling method and an importance-aware randomization mechanism are proposed to adaptively perturb sampled data to realize a better tradeoff between data privacy and pattern utility of time-series data.
- We prove that PatternLDP satisfies metric-based w-event privacy, and the experimental results on three real-world datasets show that PatternLDP outperforms existing mechanisms and can effectively preserve the important patterns.

The remainder of this paper is organized as follows. Section II reviews the related literature. Section III introduces our system model and formulates the problem. Section IV presents the design of PatternLDP in detail and its theoretical analysis. Section V presents the performance evaluation. Finally, Section VI concludes our work.

II. Related Work

Many approaches have been proposed to protect time-series data privacy. We first review the privacy-preserving approach for time-series data, and then present the LDP-based privacy protection approaches.

Time-series Data Privacy Preserving. According to different utility goals, privacy-preserving approaches for time-series data can be categorized into two aspects: one is for the statistical analysis and the other is for the time-series analysis. To achieve the first goal, Castelluccia *et al.* [18] designed an inexpensive symmetric-key homomorphic encryption scheme that allows an aggregator to compute aggregates on encrypted data. Kellaris *et al.* [19] first put forth the notion of w-event privacy over infinite streams based on differential privacy, which protects any event sequence occurring in w successive time instants. To achieve the second goal, Papadimitriou *et al.* [20] firstly consider projecting time-series data into the frequency domain for perturbation, such that the structure of the time-series data can still be preserved. Fan *et al.* [21] studied time-series sanitization with metric-based privacy and preserved the unique patterns in time-series. However, it is worth noting that all these pattern-preserving methods are not for real-time data collection, and cannot provide strict guarantee for the accuracy of statistical estimation.

Local Differential Privacy (LDP). The notion of differential privacy (DP) was first introduced by Dwork in [22]. To overcome the threat of centralized server, many differential privacy-preserving frameworks were proposed for different crowdsensing tasks (such as task allocation [23], incentive [24] and data publishing [25]). For privacy-preserving data collection, LDP [26], which allows each user perturb data locally before uploading, has recently been proposed and

applied widely. Ren *et al.* [14] proposed LoPub to achieve LDP on high-dimensional crowdsourced data and took advantage of the correlations among multiple attributes to reduce the dimensionality of crowdsourced data, so that achieving high utility. Qin *et al.* proposed LDPMiner [10] to obtain accurate heavy hitters over set-valued data and LDPGen [27] to generate representative synthetic social graphs. Chen *et al.* [28] proposed a new privacy model, called personalized LDP, for the untrusted server to learn user distribution over a spatial domain. Wang *et al.* [29] studied ordinal data aggregation for distribution estimation while preserving individuals' data privacy. Ye *et al.* [12] studied the problem of frequency and mean estimation on key-value data, and designed a sanitized mechanism LPP to perturb the key-value pair. In [7], a practical, accurate and efficient system named Harmony was proposed, which can support complex machine learning tasks by collecting the gradient in a private manner.

To the best of our knowledge, this is the first work to preserve both privacy and pattern utility of generic time-series in real-time data collection scenario. Besides the pattern utility of each time-series of a user, the statistic accuracy of a group of users can be also guaranteed simultaneously.

III. System Model and Problem Formulation

In this section, we first introduce the privacy-preserving real-time data collection model, and then describe some notations used throughout this paper. Finally, we introduce the proposed metric-based w-event privacy to measure the privacy protection degree for pattern-rich time-series and formalize the problem we want to solve.

A. System Model

There are usually two parties in a privacy-preserving real-time data collection system: a server and a group of users. We assume the server is honest-but-curious, which implies that it will honestly execute every operation in analyzing tasks but might curiously pry into users' sensitive information.

As shown in Figure 2, each user generates data, perturbs the raw data locally and sends perturbed data to the server at each timestamp. To achieve statistical utility, the server can perform statistical analysis tasks for a group of users and provide public service. To achieve pattern utility, the server can perform individual time-series analysis tasks for each user and provide personalized service.

- **Statistical analysis tasks:** Given the collected data from all the users, the server is able to aggregate these data and obtain the statistics, such as *mean* and *median*, which indicate the features of the group of these users.
- **Individual time-series analysis tasks:** This task focuses on each time-series and aims to extract meaningful information to provide personalized services.

Existing works mainly focus on achieving one of the utility goals, but how to achieve both utility for personalized service and public service while preserving each user's data privacy has not been well explored.

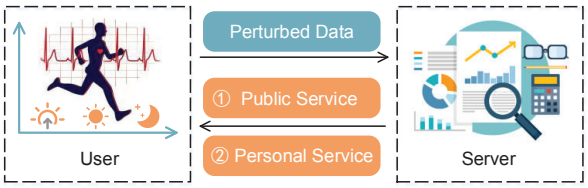


Fig. 2. The model of real-time data collection.

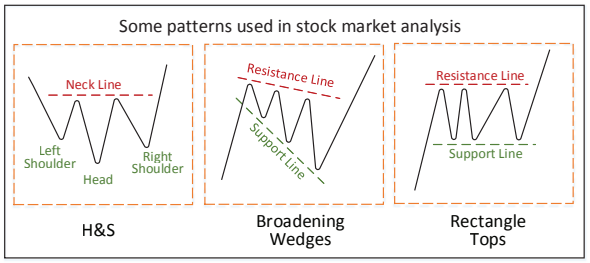


Fig. 3. Patterns in financial time-series.

B. Preliminaries

In this section, we introduce the preliminaries including the definition of time-series data and LDP.

1) Time-series: Any data with temporal attribute can be called a time series, such as location traces and event sequence. However, our work aims to design a privacy-preserving mechanism for generic time-series which is defined as follows.

***Definition** 1 (Time-series):* A time-series S is an ordered sequence of data points: $S = \{s[1], s[2], \ldots\}$, where each data point $s[i]$ is a tuple $(i, v[i])$ described by timestamp i and the corresponding univariate value $v[i]$.

Each time-series usually has its unique patterns, which can be mined to provide personalized service.

***Definition** 2 (Pattern):* A pattern P in time-series can be defined as a sequence of limited data points (e.g., $P = \{s[i], s[i+1], \ldots, s[i+k-1]\}$), which can characterize some meaningful trends of time-series.

Patterns in time-series are highly related with the application scenarios. Figure 3 shows some patterns in the financial time-series. Existing works [30] [31] on time-series have demonstrated that different data points have different impacts on the pattern. The missing of some data points would not affect the trend, which allows us to use a few remarkable points to represent a pattern, such as an H&S pattern consisting of a head point, two shoulder points, and a pair of neck points.

2) LDP: We aim to preserve privacy of each participant's time-series data. LDP, the de facto data privacy notion, was proposed to protect data privacy in the local setting where no one else can get access to the original data directly except the user himself. w-event privacy [19] is a privacy model proposed for infinite stream, which provides privacy guarantee for any event sequence occurring at any successive w timestamps.

Here we will give the definition of the variant LDP specially for time-series which provides meaningful privacy guarantees over long time scales. We first explain some notions it requires.

***Definition** 3 (w-neighboring [19]):* For a positive integer w, two time-series S, S' of length t are w-neighboring, if

1) for each $S[i]$, $S'[i]$ such that $i \in [t]$ and $S[i] \neq S'[i]$, it holds that $S[i]$, $S'[i]$ are neighboring, and

2) for each $S[i_1]$, $S[i_2]$, $S'[i_1]$, $S'[i_2]$ with $i_1 < i_2$, $S[i_1] \neq S'[i_1]$ and $S[i_2] \neq S'[i_2]$, it holds that $i_2 - i_1 + 1 \leq w$.

Formally, let $\mathcal{M}$ denote the randomization mechanism, and $\mathcal{M}(S)$ denote the output when the input is S. $\mathcal{L}$ denotes the whole database and $Range(\mathcal{M})$ is the set of all possible outputs of $\mathcal{M}$. The definition of LDP is described as follows:

***Definition** 4 (LDP):* A randomized mechanism $\mathcal{M}$ satisfies ϵ-local differential privacy if and only if for any w-neighboring time-series $S, S' \in \mathcal{L}$ and for any possible output $R \in Range(\mathcal{M})$, we have

$$Pr[\mathcal{M}(S) = R] \leq e^{\epsilon} \times Pr[\mathcal{M}(S') = R] \quad (1)$$

where ϵ is the privacy budget quantifying the level of privacy protection. Intuitively, LDP means that when obtaining the output R, the server cannot infer whether the input time-series is S or S' with high confidence.

C. Problem Formulation

In this section, we will introduce a novel notion called metric-based w-event privacy to measure the privacy-preserving degree according to the distance metrics between time-series and present the problem formulation formally.

It has been proved that LDP in general requires more noise than DP to achieve the same level of protection [32]. As the time-series dimension increases, the accumulated noise will substantially spoil the utility of the time-series data. Thus, by combining $d_{\mathcal{X}}$-privacy [33] and w-event privacy [19], we propose a local privacy notion, called *metric-based w-event privacy*, improving the tradeoff between privacy and utility.

***Definition** 5 (Metric-based w-Event Privacy):* A mechanism $\mathcal{M}$ satisfies metric-based w-event ϵ-differential privacy, if for any possible output $R \subseteq Range(\mathcal{M})$ and all w-neighboring time-series S,$S^{'}$, it holds that,

$$\Pr[\mathcal{M}(S) = R] \leq e^{\epsilon d(S,S')} \times \Pr[\mathcal{M}(S') = R], \quad (2)$$

where $d(S, S')$ is the Euclidean distance between S and S'.

By introducing the distance metrics, metric-based w-event privacy can provide low distinguishability between similar time-series which makes it harder to infer true values and high distinguishability between dissimilar time-series which prevents the output of outliers. Therefore, a mechanism satisfying metric-based w-event privacy imposes the minimum utility loss for protecting generic time-series data privacy at any successive w timestamps.

Formally, the problem we study in this paper is described as follows. Given the time-series data $S_c = \{s_c[1], s_c[2], \cdots\}$ possessed by user u_c, the objective is to obtain a sanitized version $R_c = \{r_c[1], r_c[2], \ldots\}$ and send it to the untrusted server in real-time, such that R_c satisfies the privacy protection requirement while its utility (i.e. statistical utility and pattern utility) can still be guaranteed.

IV. Pattern-aware privacy preserving Mechanism

In this section, we present PatternLDP, a novel pattern-aware privacy-preserving mechanism for real-time data collection. In the following, we first introduce the overview of PatternLDP and then describe the key components of PatternLDP in more detail. Finally, we give the strict theoretical analysis of privacy protection and utility guarantee.

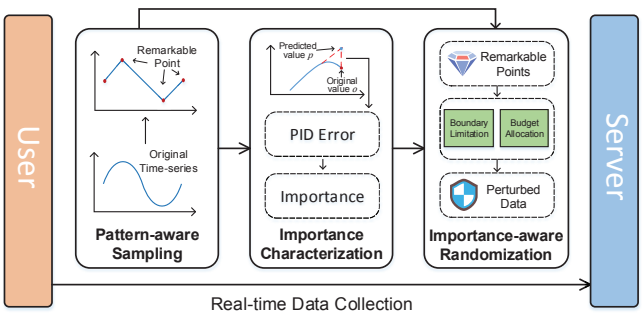


Fig. 4. The overview of the PatternLDP.

A. Overview of PatternLDP

We aim to perturb time-series while preserving the useful patterns in real-time. Thus, our main idea is to sample the remarkable points which represent useful patterns in real-time and adaptively perturb them based on their importance. Figure 4 shows the overview of PatternLDP, which mainly consists of three mechanisms: pattern-aware sampling, importance characterization and importance-aware randomization.

Pattern-aware Sampling: This component aims to sample the remarkable points in real-time. We model it as an optimization problem whose objective is to minimize the number of remarkable points while the representation error does not exceed the error tolerance. To solve the problem efficiently, a criterion called *feasible space* [34] is adopted to determine the sampling points.

Importance Characterization: This component aims to characterize the importance of each remarkable point. Since remarkable points in short-term patterns can have greater impacts than in long-term patterns, we use data dynamics, characterized by PID control, as a measure of the importance.

Importance-aware Randomization: This component aims to perturb time-series while minimizing the pattern loss. In particular, we design an importance-aware budget allocation method to ensure the sensitivity to patterns and an importance-aware randomized response method to achieve metric-based data perturbation. The above local protection process achieves metric-based w-event privacy.

Finally, the server collects the perturbed sampling points and approximates the time-series based on PLA. The outcome of approximation can be further used to support statistical estimation and time-series analysis. In the following, we will describe each component in detail.

B. Pattern-aware Sampling

In this section, we propose a pattern-aware sampling method based on PLA, which is consistent with human visual experience, to search for remarkable points. In order to improve the utility, fewer points are expected to be sampled to achieve the goal of representing patterns. Thus, in real-time sampling, the objective is to find the farthest next remarkable points

$s[j]$ while the representation error does not exceed the error tolerance δ, which can be described as follows:

$$\begin{aligned} \max \quad & |i-j| \\ \text{s.t.} \quad & error(s[k]) \leq \delta \quad \forall k \in [i,j] \end{aligned} \tag{3}$$

where i is the timestamp of last remarkable point, $error(s[k])$ is the representation error for data point $s[k]$ and δ is the user-specified error tolerance.

Representation error: In PLA, patterns are represented by the linear connection of remarkable points. Thus, we choose Vertical Distance (VD) between the actual data point $v[k]$ and the fit line $\widehat{ij}$ between two adjacent sampling points, as the certain measure for evaluating the representation error:

$$error(s[k]) = VD(s[k], \widehat{ij}). \tag{4}$$

Real-time Sampling: For the purpose of solving the problem more efficiently to achieve real-time requirement, we adopt the slope limitation to indirectly describe the representation error of each point. Table I illustrates some definitions in the slope limitation as follows.

TABLE I
THE DEFINITIONS OF SLOPE CALCULATION

Name	Description
$l(s[i], s[j])$	The slope of the straight line connected by points $s[i]$ and $s[j]$
$low(s[i], s[j])$	The slope of the straight line connected by points $s[i]$ and $(j, v[j]-\delta)$
$up(s[i], s[j])$	The slope of the straight line connected by points $s[i]$ and $(j, v[j]+\delta)$

The constraint in Eq. 3 can be transformed as:

$$low(s[i], s[k]) \leq l(s[i], s[j]) \leq up(s[i], s[k]). \tag{5}$$

Consequently, for the coming data point $s[j]$, the limits of the representation error for its slope are:

$$\begin{aligned} l_{low} &= \max_{i<k<j} low(s[i], s[k]) \\ l_{up} &= \min_{i<k<j} up(s[i], s[k]). \end{aligned} \tag{6}$$

We call the interval $[l_{low}, l_{up}]$ as the *feasible space*, which will be updated every time a new data point arrives. If the slope of the new line $l(s[i], s[j])$ falls into the current feasible space, which means that the representation error will not exceed δ for all data points between $s[i]$ and $s[j]$, current data point $s[j]$ can be chosen as the candidate of next remarkable point.

When the feasible space becomes empty ($l_1 > l_2$), there will be no points which can join into current candidate set of remarkable points. Finally, we sample the new remarkable point, which is exactly the last point in current candidate set.

C. Importance Characterization

Since different remarkable points have different extents of influence on the patterns, in this section, we propose to characterize their importance according to data dynamics.

We observe that the influence of the remarkable points is mainly determined by the duration of the pattern to which the current point belongs. For remarkable points in the long-term pattern, the importance is very low because the perturbation will not cause significant changes in the pattern. Since the specific pattern to be preserved is unknown, the duration of the pattern cannot be directly obtained, but can be estimated by data dynamics. Intuitively, when data changes rapidly, the time-series presents a short-term pattern in which the remarkable points are of higher importance.

We adopt the PID control to characterize data dynamics, and then determine importance of each sampling point by the PID error. We first define the feedback as the error between the true value and the predicted value:

$$F[k_n] = |v[k_n] - \hat{v}[k_n]| \tag{7}$$

where k_n is the timestamp of the n-th sampling point and $\hat{v}[k_n]$ is the predicted value through the analysis on time-series.

Thus, the importance, characterized by the PID error, can be calculated as follows:

$$\gamma[k_n] = K_p F[k_n] + K_i \frac{\sum_{o=n-\pi-1}^{n} F[k_o]}{\pi} + K_d \frac{F[k_n] - F[k_{n-1}]}{k_n - k_{n-1}} \tag{8}$$

where the parameters K_p, K_i, and K_d are the standard PID scale factors representing proportional gain, integral gain and derivative gain, respectively. The first term $K_p F[k_n]$ is the proportional error standing for the current data dynamics; the second term $K_i \frac{\sum_{o=n-\pi-1}^{n} F[k_o]}{\pi}$ is the integral error standing for the accumulation of π past errors, eliminating the offset caused by outlier points; the third term $K_d \frac{F[k_n]-F[k_{n-1}]}{k_n - k_{n-1}}$ is the derivative error standing for the future data dynamics.

D. Importance-aware Randomization

In this section, we propose an importance-aware randomization mechanism, which adaptively injects noise to sampling points, to reduce pattern loss. As shown in Algorithm 1, we specify the privacy-preserving degree for each sampling point by importance-aware budget allocation and importance-aware boundary limitation. With the specific privacy degree, we apply randomized response mechanism which achieves metric-based privacy to perturb the values of sampling points.

1) Importance-aware Budget Allocation: For time-series data, metric-based w-event privacy requires that the sum of budgets within any sliding window of w timestamps is at most the entire privacy budget ϵ, which provides an opportunity to allocate budget for each sampling point.

In PatternLDP, we aim at preserving useful patterns. Those remarkable points which have great impacts on patterns are supposed to be slightly perturbed. Thus, a larger portion of the remaining privacy budget should be allocated to more important sampling point. Simply, the proportional function which decides the portion of the remaining budget allocated to current sampling point can be defined as:

$$p = 1 - exp(-\gamma[i]). \tag{9}$$

The exponential function guarantees that p ranges from 0 to 1. The final budget allocated to the current timestamp is calculated as $\epsilon[i] = p \cdot \epsilon'$, where ϵ' is the remaining budget in the window $\epsilon' = \epsilon - \sum_{j=i-w+1}^{i-1} \epsilon[j]$.

However, once multiple important sampling points come consecutively, the remaining budget within the sliding window will be almost used up, thus few available budgets are reserved for subsequent potential sampling points. A simple solution [4] is to allocate a small portion of the remaining budget to current sampling point when data changes rapidly, that is:

$$p = 1 - exp(-\frac{1}{\gamma[i]}). \tag{10}$$

In this way, we have two opposing requirements for budget allocation: one is to ensure the adequate budget for successive potential sampling points and the other is to ensure the sensitivity to the importance of sampling points.

We must discover a balance to reasonably allocate the budget so that both requirements can be achieved very well. In other words, if the remaining budget is sufficient, the budget allocation process should be as sensitive as possible to the importance for the purpose of preserving useful patterns. While if the remaining budget is insufficient, the budget reserved for subsequent data points should be as much as possible for the purpose of protecting data privacy.

To this end, we refine the proportional function as follows:

$$p = 1 - exp(-(\frac{\alpha}{\gamma[i]} + \beta\gamma[i])) \tag{11}$$

where α, β are weight factors and $\alpha + \beta = 1$.

Through the dynamic update of weight factors, the equilibrium between the two allocation strategies can be achieved.

We call $\mathfrak{B} = \frac{\alpha}{\beta}$ the equilibrium factor. Intuitively, the equilibrium will move in the direction of allocating less budget to more important sampling points (i.e. increasing $\mathfrak{B}$) when the remaining budget left for the current data point is too small (i.e. $\epsilon' < \frac{\epsilon}{w}$). The equilibrium will move in the direction of allocating more budget to more important sampling points (i.e. decreasing $\mathfrak{B}$) when the remaining budget left for the current data point is sufficient (i.e. $\epsilon' > \frac{\epsilon}{2}$). The specific dynamic update mechanism is as follows:

$$\alpha[i+1] = \begin{cases} \alpha[i] - \nabla_\alpha \mathfrak{B}, & \epsilon' > \frac{\epsilon}{2}, \\ \alpha[i], & \frac{\epsilon}{w} \le \epsilon' \le \frac{\epsilon}{2}, \\ \alpha[i] + \nabla_\alpha \mathfrak{B}, & \epsilon' < \frac{\epsilon}{w}, \end{cases} \tag{12}$$

where $\nabla_\alpha \mathfrak{B}$ is the gradient of the equilibrium factor $\mathfrak{B}$ with respect to the weight factor α. In this case, we can achieve the dynamic equilibrium to reasonably allocate the budget according to the importance.

2) Importance-aware Randomized Response: Existing methods of randomized response on numeric data limit the output to two discrete values, which guarantees statistical utility but can cause significant pattern loss in time-series. To preserve patterns, the output of randomized response should be limited to a certain range related to the original value. Therefore, we take boundary limit on private data into consideration and define the adaptive error bound $b[i]$ for sampling timestamp i. The randomized response range is determined as all the possible output data with values in a range $[v[i] - b[i], v[i] + b[i]]$. Intuitively, the probability of the generated perturbed value $v^*[i]$ that violates boundary limitation will be set to 0.

Algorithm 1: Importance-aware Randomization

Input: The importance of data points γ, privacy budget ϵ, true value v, the size of sliding window w.
Output: Perturbed value v^*.

```
α = β = 0.5;
for i = 1 → ∞ do
    ε' = ε − Σ_{j=i−w+1}^{i−1} ε[j];
    Update α according to Eq. 12;
    while s[i] is a sampling point do
        ε[i] = ε' · (1 − exp(−(α/γ[i] + βγ[i])));
        b[i] = ln(θ/γ[i] + μ);
        Update probability density function Pr;
        v*[i] ← random sample from Pr(v*[i]|v[i]);
    end
end
```

Since different sampling points have different requirements for boundary limitation, the value of $b[i]$ need to be updated according to the importance of sampling points. To improve the pattern utility, it is always hoped that the important points won't change a lot after perturbation. Thus, for the data points with larger importance, we set smaller b, which can be described as follows:

$$b[i] = \ln(\frac{\theta}{\gamma[i]} + \mu), \tag{13}$$

where θ is a scale factor and the role of μ is to guarantee that the size of output range of data perturbation is at least $\ln \mu$ even if $\gamma[i]$ is too large.

Given the true value $v[i]$, the response range $[v[i] - b[i], v[i] + b[i]]$, and the privacy budget $\epsilon[i]$, a perturbed value $v^*[i]$ can be returned at timestamp i. For each sampling point, the user reports his true answer $v[i]$ with probability $q[i]$, and a random answer, which is chosen from the output set, with probability $1 - q[i]$. Since metric-based privacy ensures up to $e^{\epsilon d(v[i], v'[i])}$ factor of distinguishability in output probabilities for the pair $(v[i], v'[i])$, the probability of reporting a random answer is related to the distance from true value. To satisfy metric-based privacy at each timestamp, we set $q[i]$ as follows:

$$q[i] = \frac{1}{2} \cdot \frac{\epsilon[i]}{1 - exp(-\epsilon[i] \cdot b[i])}. \tag{14}$$

When current data point is determined to be a sampling point, the perturbed value $v^*[i]$ is sampled from the following distribution:

$$\Pr(v^*[i]|v[i]) = q[i] \cdot exp(-\epsilon[i] \cdot d(v^*[i], v[i])) \tag{15}$$

where $d(v^*[i], v[i]) = |v^*[i] - v[i]|$ denotes the distance between the output value and the input value. When sampling

perturbed data from the above distribution, there will be high distinguishability between dissimilar data value which improves data utility, and low distinguishability between similar data which provides strong privacy protection.

Finally, the server will collect the perturbed outputs of each user and synthesize them into the time-series. With the pattern-aware sampling and importance-aware perturbation, the sanitized time-series preserve the essential characteristics and thus support statistical analysis and time-series analysis.

E. Theoretical Analysis

In this section, we prove that PatternLDP achieves metric-based w-event privacy and the statistical utility.

***Theorem** 1:* The importance-aware randomized response mechanism satisfies $\epsilon[i] \cdot d(s[i], s'[i])$-LDP at timestamp i.

***Proof** 1:* For two data point $s[i], s'[i]$, we define their values as $v[i], v'[i]$, which we write as v, v' in short to facilitate the proof. Assuming that the budget allocated to current data point is $\epsilon[i]$. According to Eq. 15, we have the following equation:

$$\frac{Pr(v^*|v)}{Pr(v^*|v')} = \frac{q[i]/e^{\epsilon[i]d(v^*,v)}}{q[i]/e^{\epsilon[i]d(v^*,v')}} = e^{\epsilon[i](d(v^*,v')-d(v^*,v))} \quad (16)$$

Since v, v' have the same perturbation range, we have

$$d(v, v') = \begin{cases} |d(v^*, v') + d(v^*, v)|, & v^* \in (v, v'), \\ |d(v^*, v') - d(v^*, v)|, & v^* \notin (v, v'). \end{cases} \quad (17)$$

It can be always true that $d(v, v') \geq d(v^*, v') - d(v^*, v)$. Thus, Eq. 16 can be updated:

$$\frac{Pr(v^*|v)}{Pr(v^*|v')} = e^{\epsilon[i](d(v^*,v')-d(v^*,v))} \leq e^{\epsilon[i]d(v,v')} \quad (18)$$

Since v is the value of data point $s[i]$, we have:

$$d(v, v') = |v - v'| = d(s[i], s'[i]). \quad (19)$$

Thus, our importance-aware randomized response mechanism satisfy $\epsilon[i] \cdot d(s[i], s'[i])$-LDP at timestamp i.

***Theorem** 2:* PatternLDP satisfies metric-based w-event ϵ-differential privacy.

***Proof** 2:* Since all the mechanisms use independent randomness, the following holds for time-series S and any mechanism output $R \in Range(\mathcal{M})$:

$$Pr[\mathcal{M}(S) = (r[1], ..., r[t])] = \prod_{k=1}^{t} Pr[\mathcal{M}_k(s[k]) = r[k]] \quad (20)$$

By Definition 3, there exists $i \in [t]$, such that $s[k] = r[k]$ for $1 \leq k \leq i - w$ and $i + 1 \leq k \leq t$. Thus, we have

$$\frac{Pr[\mathcal{M}(S) = (r[1], ..., r[t])]}{Pr[\mathcal{M}(S') = (r[1], ..., r[t])]} = \prod_{k=i-w+1}^{i} \frac{Pr[\mathcal{M}_k(s[k]) = r[k]]}{Pr[\mathcal{M}_k(s'[k]) = r[k]]} \quad (21)$$

Since each mechanism $\mathcal{M}_k$ satisfies $\epsilon[k] \cdot d(s[k], s'[k])$-LDP, and $d(s[k], s'[k]) \leq d(S, S')$ for $k \in [t]$, we derive that

$$\begin{aligned} \frac{Pr[\mathcal{M}(S) = (r[1], ..., r[t])]}{Pr[\mathcal{M}(S') = (r[1], ..., r[t])]} &\leq \prod_{k=i-w+1}^{i} e^{\epsilon[k] \cdot d(s[k], s'[k])} \\ &\leq e^{\sum_{k=i-w+1}^{i} \epsilon[k] \cdot d(S, S')} \end{aligned} \quad (22)$$

Since the importance-aware budget allocation mechanism already guarantees that $\sum_{k=i-w+1}^{i} \epsilon[k] \leq \epsilon$ for any sliding window w timestamps, we have $\frac{Pr[\mathcal{M}(S)=R]}{Pr[\mathcal{M}(S')=R]} \leq e^{\epsilon d(S,S')}$. Thus, PatternLDP satisfies metric-based w-event privacy.

***Theorem** 3:* Let $m[i]$ be the mean of original values that a group of users generate at time i, and $\hat{m}[i]$ be the mean of perturbed values returned from the importance-aware randomization. Then, $E(\hat{m}[i]) = m[i]$.

***Proof** 3:* For a user u_c, let $v_c^*[i]$ denote the output at time i, $v_c[i]$ denote the input true value at time i, $b_c[i]$ denote the half of the size of output range, and $E(v_c^*[i])$ denote the expectation of the output distribution. In order to facilitate the next proof, we make $x = v_c^*[i] - v_c[i]$. According to Eq. 15, we have:

$$\begin{aligned} E(v_c^*[i]) &= \int_{v_c[i]-b_c[i]}^{v_c[i]+b_c[i]} v_c^*[i] \cdot Pr(v_c^*[i]) dv_c^*[i] \\ &= \int_{-b_c[i]}^{b_c[i]} (v_c[i] + x) \cdot \frac{q[i]}{e^{\epsilon|x|}} dx \\ &= v_c[i] \int_{-b_c[i]}^{b_c[i]} \frac{q[i]}{e^{\epsilon|x|}} dx = v_c[i] \end{aligned} \quad (23)$$

By aggregating data from all n users at time i, we have:

$$E(\hat{m}[i]) = E(\frac{1}{n} \sum_{c=1}^{n} v_c^*[i]) = \frac{1}{n} \sum_{c=1}^{n} v_c[i] = m[i] \quad (24)$$

Thus, $\hat{m}[i]$ is an unbiased estimate for $m[i]$ and the server can use the mean of perturbed values to approximate the mean of original values, which guarantees the statistical utility.

V. Performance Evaluation

In this section, we evaluate the performance of the proposed PatternLDP on three real-world time-series datasets. We conduct experiments in threefold: the evaluation on the utility of statistical estimation, the evaluation on the utility of time-series analysis and the evaluation on the effect of pattern-aware sampling and importance-aware randomization on pattern-preserving.

A. Dataset

We conduct experiments over three real-world time-series datasets which are summarized as follows:

- **ElectricDevices (ED)** [35]: This dataset is collected from the UCR time-series Data Mining Archive, containing the power usage of 16637 electrical devices, which can be classified into 7 categories, and each time-series data consists of 96 data points.
- **HRO** [36]: This dataset includes heart-rate time-series from 4 different groups of healthy subjects: (i) A Chi meditation group of 8 people wearing a Holter recorder for approximately 10 hours. (ii) A Kundalini Yoga meditation group of 4 people for approximately one and half hours. (iii) A spontaneously breathing group of 11 healthy subjects during sleeping hours. (iv) A group of 9 elite triathlon athletes in their pre-race period.
- **HRA** [37]: It contains heart-rate time-series data from one person in six days during which time he went

about his ordinary daily activities. The whole time-series contains 42963 heart-rate records.

B. Compared Approaches

In this section, we compare PatternLDP with two privacy-preserving mechanism for statistical analysis and two privacy-preserving mechanism for time-series analysis, which are represented as follows:

- **Harmony-mean** [7]: This mechanism first normalizes the value v to $[-1, 1]$. Then the perturbed value v^* is sampled from the following distribution:

$$v^* = \begin{cases} \frac{e^\epsilon+1}{e^\epsilon-1}, & w.p. \ \frac{v(e^\epsilon-1)+e^\epsilon+1}{2e^\epsilon+2} \\ -\frac{e^\epsilon+1}{e^\epsilon-1}, & w.p. \ \frac{-v(e^\epsilon-1)+e^\epsilon+1}{2e^\epsilon+2} \end{cases} \tag{25}$$

- **1BitMean** [6]:This mechanism first normalizes the input value v to $[0, h]$. Then the perturbed value v^* is sampled from the following distribution:

$$v^* = \begin{cases} 1, & w.p. \ \frac{1}{e^\epsilon+1} + \frac{v}{h}\frac{e^\epsilon-1}{e^\epsilon+1} \\ 0, & w.p. \ \frac{e^\epsilon}{e^\epsilon+1} - \frac{v}{h}\frac{e^\epsilon-1}{e^\epsilon+1} \end{cases} \tag{26}$$

- **Local w-event** [19]: We adapt the mechanisms in [5] [19] to preserve ϵ-differential privacy for time-series generated within w successive time instants in the local setting.
- **Metric-based Time-series Sanitization (MTSS)** [21]: This mechanism projects the time-series into the frequency domain through Discrete Cosine Transform (DCT) and gets the first k coefficients x_0. Then, the perturbed coefficients x are sampled from the following distribution:

$$D_{\epsilon,k}(x) = \frac{1}{2}(\frac{\epsilon}{\sqrt{\pi}})^k \frac{(\frac{k}{2}-1)!}{(k-1)!} e^{-\epsilon d_{\mathbb{R}^k}(x_0,x)} \tag{27}$$

C. Setup and Metrics

In our experiment, we are supposed to evaluate the utility in two aspects: statistical analysis and time-series analysis.

Specifically, we consider the mean estimation as the typical representation of statistical analysis and the estimation error as the utility metric.

Mean Relative Error (MRE): It measures the relative error of estimated means with respect to real means at each timestamp, which is defined as follows:

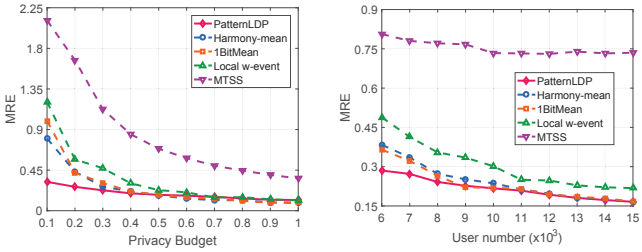
$$MRE = \frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}\frac{|m[i]-\widehat{m}[i]|}{m[i]}, \tag{28}$$

where, for $i \in \mathcal{T}$, $m[i]$ and $\widehat{m}[i]$ are the real and estimated means of all values from all users at time i.

And for time-series analysis, we consider the distance between original time-series and perturbed time-series as the utility metric to measure the degree of change in patterns.

Dynamic Time Warping distance (DTW): DTW distance, often used in pattern matching to measure the similarity, is calculated by dynamic programming, which is as follows:

$$\begin{aligned} &DTW(A,B) = D(\omega,\varrho) \\ &D(i,j) = (a_i-b_j)^2 + min\begin{cases} D(i-1,j-1) \\ D(i,j-1) \\ D(i-1,j) \end{cases} \end{aligned} \tag{29}$$



(a) ElectricDevices (b) ElectricDevices

Fig. 5. Accuracy of mean estimation.

where $A = (a_1, a_2, \cdots, a_\omega)$ and $B = (b_1, b_2, \cdots, b_\varrho)$ are two time-series of length ω and ϱ.

In our evaluation, each time-series has been normalized to zero mean and unit variance. We set $\delta = 0.5$ for pattern-aware sampling and $K_p = 0.8, K_i = 0.1, K_d = 0.1$ for importance characterization. The privacy budget ranges from 0.1 to 1.0 and we set $\epsilon = 0.5$ as default. The number of users ranges from $6k$ to $15k$ in statistical analysis. Each data point is the average of 100 independent runs under the same setting.

D. Performance Comparison

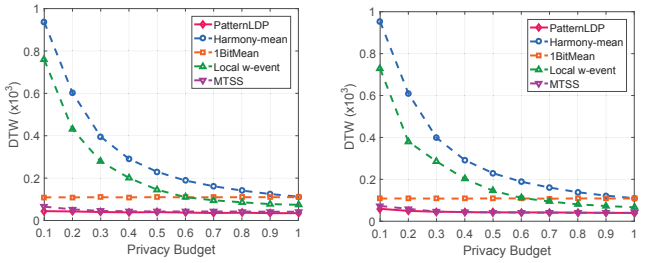
In this section, we show the performance comparison on the utility of statistical analysis and time-series analysis.

Utility of statistical analysis: Figure 5 shows the performance comparison on the accuracy of mean estimation implemented on dataset ED. Overall, the mechanisms designed specifically for data aggregation (e.g., Harmony-mean, 1BitMean) perform better on statistical analysis while the privacy-preserving mechanisms for time-series analysis often cannot guarantee the accuracy of statistical information well.

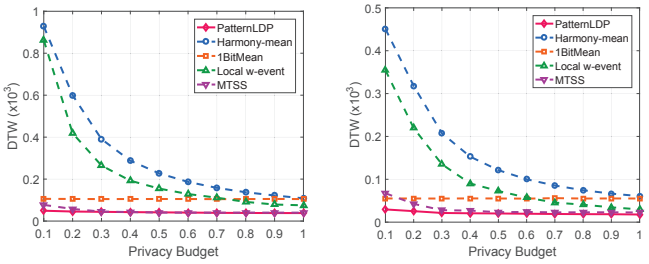
Figure 5(a) shows the variation of MRE under different privacy budgets. We can see that the MRE of all five mechanisms decrease with the increase of privacy budget, which is because larger privacy budget leads to smaller perturbation. Note that, PatternLDP doesn't change significantly, which shows the statistical stability of metric-based privacy. Figure 5(b) shows the MRE of five mechanisms against the number of users. We can also observe that the MRE of the all mechanisms except MTSS decrease significantly with the increase of the number of users. This is because MTSS performs perturbation operation in the frequency domain and the others perform in time domain based on unbiased statistics of which the accuracy comes from large number of data points.

It is worth noting that PatternLDP achieves the smallest MRE among all the mechanisms, which proves the effectiveness of our mechanism in statistical analysis.

Utility of time-series analysis: Figure 6 shows the performance comparison of five mechanisms on the effect of pattern-preserving under different types of users implemented on dataset HRO. We can observe that it is difficult for 1BitMean and Harmony-mean to effectively preserve useful patterns regardless of how much budget they are allocated. The reason is that these methods can guarantee the accuracy of statistical results by the cancellation of positive and negative noises, but the noise in temporal dimension cannot cancel each other out.

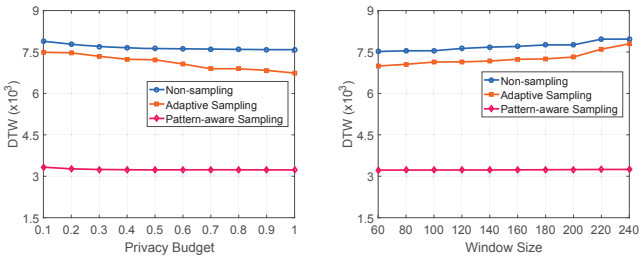


(a) Chi Meditator (b) Triathlon Athletes



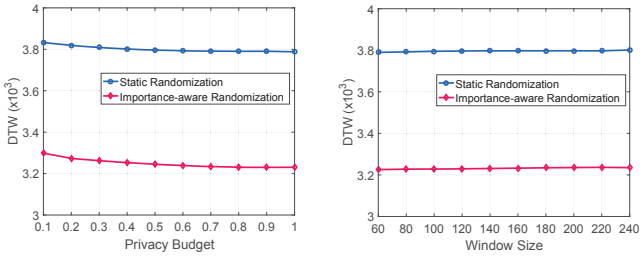
(c) Sleeper (d) Kundalini Yoga Meditator

Fig. 6. Impact of privacy budget on pattern-preserving.



(a) DTW Comparison (b) DTW Comparison

Fig. 7. Utility comparison with different sampling methods.



(a) DTW Comparison (b) DTW Comparison

Fig. 8. Utility comparison with different randomization methods.

As for other three privacy mechanism for time-series, we can see that DTW distance decreases when ϵ increases over all groups. This is because the more budget allocated, the larger probability that the perturbed value is similar to the real value.

Note that MTSS and our mechanism always have a better pattern-preserving performance than the others while Local w-event only has a certain ability of pattern-preserving as ϵ is large enough. The reason why the two mechanisms have the best performance is different: MTSS uses DCT to capture the patterns of time-series data and PatternLDP extracts remarkable points which characterize patterns.

E. Effects of PatternLDP

In this section, we evaluate the effect of two components in PatternLDP on pattern-preserving using dataset HRA.

Effect of pattern-aware sampling: We conduct experiments of PatternLDP with three different sampling methods to evaluate the effects of pattern-aware sampling on pattern-preserving. The first is sampling all the data in time-series, in other words, no sampling. The second is adaptive sampling used in [4] and the last is our pattern-aware sampling. Figure 7 shows the results of utility comparison. We observe that adaptive sampling reduces the DTW distance a little compared to no sampling, which means it mitigates the effect of noise on the pattern to a certain extent. As for our pattern-aware sampling, it reduces the DTW distance significantly which means our sampling method can better preserve the pattern.

Effect of importance-aware randomization: We conduct experiments of PatternLDP with and without consideration of each data's importance to evaluate the effects of importance-aware randomization. Figure 8 shows the results of utility comparison. We can observe that the DTW distance is smaller when we adopt the importance-aware randomized response. That is, the proposed importance-aware randomization does improve the effect of pattern-preserving. Therefore, it is appropriate to specify the privacy degree by characterizing the importance of each remarkable point to improve the utility of the sanitized time-series data.

VI. Conclusions

In this paper, we focused on real-time data collection under a honest-but-curious server, and proposed a novel pattern-aware privacy-preserving mechanism, called PatternLDP, to protect privacy while preserving patterns in time-series. Instead of providing the same level of privacy protection at each data point, a pattern-aware sampling method and an importance-aware randomized response mechanism were proposed to sample the remarkable points in time-series and adaptively perturb them according to their impacts on patterns. The theoretical analysis and extensive experiments on real-world datasets validated the effectiveness of PatternLDP in terms of improving pattern utility as well as statistical utility. The experimental results also demonstrated that PatternLDP is sensitive at capturing pattern change and robust to the change of privacy budget and window size.

VII. Acknowledgments

This work was supported by National Natural Science of China (Grants No. 61872274, 61702562, U19A2067, and 61802180), National Key Research and Development Program under Grants 2018YFB0803402 and 2019YFA0706403, the Young Tlite Scientists Sponsorship Program by CAST under Grant No. 2018QNRC001, Natural Science Foundation of Jiangsu Province (No. BK20180421), Key Research and Development Program of Shanxi Province under Grant 201903D121121, Fundamental Research Funds for the Central Universities (Grants No. 2042018gf0043, 2042019gf0098, and NE2018106), and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System (Wuhan University of Science and Technology).

REFERENCES

[1] J. Zhou, Z. Cao, X. Dong, and X. Lin, "Tr-mabe: White-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving e-healthcare cloud computing systems," in *Proc. of IEEE INFOCOM*, 2015, pp. 2398–2406.

[2] L. Shou, X. Shang, K. Chen, G. Chen, and C. Zhang, "Supporting pattern-preserving anonymization for time-series data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 877–892, 2013.

[3] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Matching anonymized and obfuscated time series to users profiles," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 724–741, 2019.

[4] 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, pp. 591–606, 2018.

[5] L. Fan and L. Xiong, "An adaptive approach to real-time aggregate monitoring with differential privacy," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 9, pp. 2094–2106, 2014.

[6] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Proc. of the NIPS*, 2017, pp. 3571–3580.

[7] T. T. Nguyên, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin, "Collecting and analyzing data from smart device users with local differential privacy," *arXiv preprint arXiv:1606.05053*, 2016.

[8] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proc. of ACM CCS*, 2014, pp. 1054–1067.

[9] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Proc. of NIPS*, 2014, pp. 2879–2887.

[10] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proc. of ACM CCS*, 2016, pp. 192–203.

[11] S. Wang, L. Huang, Y. Nie, P. Wang, H. Xu, and W. Yang, "Privset: Set-valued data analyses with locale differential privacy," in *Proc. of IEEE INFOCOM*, 2018, pp. 1088–1096.

[12] Q. Ye, H. Hu, X. Meng, and H. Zheng, "Privkv: Key-value data collection with local differential privacy," in *Proc. of IEEE S&P*, 2019, pp. 294–308.

[13] G. Fanti, V. Pihur, and Ú. Erlingsson, "Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries," *Privacy Enhancing Technologies*, vol. 2016, no. 3, pp. 41–61, 2016.

[14] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and S. Y. Philip, "Lopub: High-dimensional crowdsourced data publication with local differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.

[15] J. Bandera, R. Marfil, A. Bandera, J. A. Rodríguez, L. Molina-Tanco, and F. Sandoval, "Fast gesture recognition based on a two-level representation," *Pattern Recognition Letters*, vol. 30, no. 13, pp. 1181–1189, 2009.

[16] H. Shatkay and S. B. Zdonik, "Approximate queries and representations for large data sequences," in *Proc. of IEEE ICDE*, 1996, pp. 536–545.

[17] Y. Mohammad and T. Nishida, "Constrained motif discovery in time series," *New Generation Computing*, vol. 27, no. 4, p. 319, 2009.

[18] C. Castelluccia, A. C. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 20:1–20:36, 2009.

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

[20] S. Papadimitriou, F. Li, G. Kollios, and P. S. Yu, "Time series compressibility and privacy," in *Proc. of VLDB Endowment*, 2007, pp. 459–470.

[21] L. Fan and L. Bonomi, "Time series sanitization with metric-based privacy," in *Proc. of IEEE BigData Congress*, 2018, pp. 264–267.

[22] C. Dwork, "Differential privacy," in *Proc. of ICALP*, 2006, pp. 1–12.

[23] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, 2019.

[24] Z. Wang, X. Pang, J. Hu, W. Liu, Q. Wang, Y. Li, and H. Chen, "When mobile crowdsensing meets privacy," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 72–78, 2019.

[25] Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, and H. Qi, "Privacy-preserving crowd-sourced statistical data publishing with an untrusted server," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1356–1367, 2019.

[26] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. of IEEE FOCS*, 2013, pp. 429–438.

[27] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proc. of ACM CCS*, 2017, pp. 425–438.

[28] R. Chen, H. Li, A. Qin, S. P. Kasiviswanathan, and H. Jin, "Private spatial data aggregation in the local setting," in *Proc. of IEEE ICDE*, 2016, pp. 289–300.

[29] S. Wang, Y. Nie, P. Wang, H. Xu, W. Yang, and L. Huang, "Local private ordinal data distribution estimation," in *Proc. of IEEE INFOCOM*, 2017, pp. 1–9.

[30] F. Chung, T. C. Fu, R. Luk, and V. Ng, "Flexible time series pattern matching based on perceptually important points," in *Proc. of IJCAI Workshop*, 2001, pp. 1–7.

[31] T. Fu, F. Chung, K. Kwok, and C. Ng, "Stock time series visualization based on data point importance," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1217–1232, 2008.

[32] M. Alvim, K. Chatzikokolakis, C. Palamidessi, and A. Pazii, "Local differential privacy on metric spaces: optimizing the trade-off with utility," in *Proc. of IEEE CSF*, 2018, pp. 262–267.

[33] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *Proc. of PETS*, 2013, pp. 82–102.

[34] X. Liu, Z. Lin, and H. Wang, "Novel online methods for time series segmentation," *IEEE Transactions on knowledge and data engineering*, vol. 20, no. 12, pp. 1616–1626, 2008.

[35] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," July 2015, www.cs.ucr.edu/ eamonn/time_series_data/.

[36] C.-K. Peng, J. E. Mietus, Y. Liu, G. Khalsa, P. S. Douglas, H. Benson, and A. L. Goldberger, "Exaggerated heart rate oscillations during two meditation techniques," *International journal of cardiology*, vol. 70, no. 2, pp. 101–107, 1999.

[37] "Heart rate analysis," https://github.com/JenniferLing/heart_rate_analysis, 2017.