Real-Time Stream Statistics via Local Differential Privacy in Mobile Crowdsensing

Teng Wang[1(✉)] and Zhi Hu[2]

[1] Xi'an University of Posts and Telecommunications, Xi'an 710121, China
wangteng@xupt.edu.cn
[2] Northwest University, Xi'an 710127, China

Abstract. Mobile crowdsensing has enabled the collection and analysis of stream data. However, the direct processing of gigantic stream data will seriously compromise users' privacy since those stream data involve numerous sensitive information. To address the challenges of the vulnerabilities of untrusted crowdsensing servers and low data utility, we propose an effective real-time stream statistics mechanism that can not only achieve strong privacy guarantees, but also ensure high data utility. We firstly apply local perturbation on each user's stream data on the client side, which achieves ω-event ϵ-local differential privacy for each user at each timestamp. Then, we propose a retroactive grouping-based noise smoothing strategy that adaptively exploits the time correlations of stream data and smooths excessive noises, thus improving data utility. Finally, experimental results on real-world datasets show the strong effectiveness of our mechanism in terms of improving data utility.

Keywords: Stream data · Local differential privacy · Time correlation · Data utility

1 Introduction

Mobile crowdsensing has greatly promoted and facilitated the big data collection, analysis and utilization [15,20]. Without deploying thousands of static sensors, a large scale mobile crowdsensing system can be easily formed with portable smart mobile devices, such as mobile phones, smart watch/bands, etc. In such a system, each participant will joint one or more monitoring tasks and continuously contribute her/his personal stream data for statistics. Therefore, mobile crowdsensing has been widely adopted for many data-driven applications, such as traffic flow control, population density monitoring, health management [2,28,29].

However, the continual collection and analysis of stream data seriously violates the privacy of each participant and causes severe impacts [18,30]. Because

J. Xiong et al. (Eds.): MobiMedia 2021, LNICST 394, pp. 432–445, 2021.
https://doi.org/10.1007/978-3-030-89814-4_31

such stream data involve in individuals' identity, location, health status, or other sensitive information. What's worse, the privacy leaks will gradually accumulate as time continues to grow [6], which will produce negative consequences.

Differential privacy (DP) [12], as a rigorous privacy paradigm, has been widely adopted to provide users with privacy protection in real-time monitoring systems [10,11]. Many existing studies [5,7,14,17,22] have applied DP on stream data publishing when facing on a trustworthy server, which cannot be directly used for stream data processing in mobile crowdsensing.

As a distributed variant of DP, local differential privacy (LDP) [25] is enable to provide privacy guarantees for each user locally and is independent of any assumptions on servers. Based on LDP, Erlingsson et al. [13] proposed to provide longitudinal privacy protection for user's successive data. Ding et al. [8] designed an LDP-compliant method for continual data collection and estimation. Moreover, Joseph et al. [16] also focused on collecting up-to-date statistics over time under LDP. When considering the dynamic and time correlations of stream data, the privacy-preserving mechanism for real-time stream data processing is still in the early stage of research. In nature, the privacy-preserving concerns of real-time statistics mainly cover two-fold challenges in mobile crowdsensing:

1) **Non-local privacy.** Existing studies on real-time stream statistics mainly using centralized differential privacy that assumes a trusted server. But in mobile crowdsensing systems, the servers may be untrustworthy and interested in users' sensitive data.
2) **Low data utility.** In nature, the stream data are highly sparse and dynamic, and accompanied by complex time correlations. In its simplest way of LDP deployment, the direct perturbation on stream data is vulnerable to low data utility.

In this paper, to address the above challenges and further improve data utility, we propose a local differential privacy (LDP)-based stream data collection and statistics mechanism which incorporates a retroactive grouping strategy to guide noise smoothing in an adaptive way. In summary, we make the following contributions.

- We propose to perturb stream data of each user on the client side by applying LDP, which prevents privacy leakage from the data source when considering untrusted servers in crowdsensing. Based on the randomized response technique, the local perturbation is performed on each user's stream data to achieve ω-event ϵ-LDP at each timestamp.
- We design a retroactive grouping-based noise smoothing strategy to reduce noise error in stream aggregation and estimation. The retroactive grouping can adaptively exploit the time correlations of the dynamic streams, thus making noise smoothing effective and improving data utility.
- We conduct extensive experiments on two real-world datasets. The results demonstrate that our mechanism can improve data utility while ensuring strong privacy protections.

The remainder of the paper is organized as follows. Section 2 provides a literature review. Section 3 outlines the preliminaries and problem statement. Section 4 introduces our proposed mechanism. Section 5 presents the evaluation results. Finally, Sect. 6 concludes the paper.

2 Related Works

The privacy-preserving mechanisms for real-time stream data processing have been widely studied under differential privacy (DP). The design principle of existing schemes consists of transformation, modeling, sampling, grouping, and clustering.

Based on the compressibility of time-series data, Xiao et al. [27] proposed to transform the frequency domain of the raw data into a wavelet coefficient matrix and then add Laplace noise to the coefficients to achieve DP. As for large-scale spatio-temporal data, Acs et al. [4] combined sampling, clustering, Fourier perturbation, and smoothing to improve the utility of the published data. Besides, both [19] and [3] adopted discrete Fourier transformation technique under differential privacy protection.

As for dynamic stream data, Fan et al. [14] proposed FAST which adaptively samples data points and adds noise based on Kalman filter and PID feedback error. FAST can capture the dynamic changes of stream data by an adaptive sampling and filtering mechanism, thus reducing the total noise. Moreover, Wang et al. [21] proposed to use an unscented Kalman filter to publish time-series data in non-linear scenes. To publish infinite streams, Kellaris et al. [17] proposed ω-event privacy model which can protect any event sequences within any window of ω timestamps. Correspondingly, they also introduced two privacy budget allocation mechanisms BD and BA that ensure better data utility than uniform distribution mechanisms. Wang et al. [22,23] proposed RescueDP which integrates adaptive sampling and budget allocation, dynamic grouping, perturbation, and filtering to publish multiple infinite streams with high data utility. Chen et al. [7] proposed PeGaSus, which adopts a perturber-grouper-smoother framework to smooth excessive noises, thus improving data utility greatly.

However, the above mechanisms adopt DP to achieve privacy protection under the assumption of a trusted server, which cannot prevent insider attacks from inferring the privacy of users. To address this, local differential privacy (LDP) [9] can be accepted to ensure data privacy for distributed crowdsensing tasks. In this direction, LDP is widely used to alleviate the privacy concerns for many data collection and analytic tasks, such as frequency and mean value estimation, frequent items mining, marginal release, empirical risk minimization (ERM), deep learning, recommendation system, etc. [25].

So far, most studies focus on data collection or analysis with LDP only for one-time computation. As for evolving data and repeated collection, Erlingsson et al. [13] proposed the RAPPOR mechanism to publish binary attributes with LDP, which tries to achieve longitudinal privacy protection for each user. Besides, Ding et al. [8] developed LDP protocols for repeated collection and computation

of counter data such as daily APP usage statistics. Moreover, Joseph et al. [16] proposed the THRESH mechanism under LDP for collecting up-to-date statistics over time, which only updates the global estimation when it becomes sufficiently inaccurate, thus improving data utility. Nonetheless, the LDP-compliant mechanism for evolving stream data is still in its infancy.

3 Preliminaries and Models

In this section, we introduce some background knowledge of local differential privacy, formalize the system/stream model, and present the problem definition.

3.1 Local Differential Privacy

Local differential privacy has been widely adopted to achieve privacy protection in distributed crowdsensing [25].

Definition 1 (ϵ-Local Differential Privacy (ϵ-LDP) [9]). *A randomized algorithm $\mathcal{A}$ satisfies ϵ-LDP if and only if for any pairs of input values v, v' in the domain of $\mathcal{A}$, and for any possible output $y \in Y$, it holds that*

$$\mathbb{P}[\mathcal{A}(v) = y] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{A}(v') = y], \tag{1}$$

where $\mathbb{P}[\cdot]$ denotes probability and ϵ is the privacy budget. A smaller ϵ means stronger privacy protection, and vice versa.

Moreover, in the context of continuous stream processing, we consider ω-event privacy [17] as our privacy model since it can protect event sequences within any window of ω timestamps. The definition of ω-event privacy can be found in [17]. Under ω-event privacy, the sequential composition is a key character of LDP, which is defined as follows.

Theorem 1 (Sequential Composition). *Let $\mathcal{A}_i(v)$ be an ϵ_i-LDP algorithm on an input value v, and $\mathcal{A}(v)$ is the sequential composition of $\mathcal{A}_1(v), \cdots, \mathcal{A}_k(v)$. Then algorithm $\mathcal{A}(v)$ satisfies $\sum_{i=1}^{k} \epsilon_i$-LDP.*

3.2 Model Formalization

System Model. We consider a large distributed mobile crowdsensing system that consists of abundant local sensing nodes (i.e., mobile smart devices) and the central server. Each node randomly participates in a sensing task and uploads the sensing state to the central server in real-time. The central server will aggregate the data stream and conduct statistics. We don't make any assumptions about the credibility of the server. That is, the central server can be *honest* or *honest-but-curious*.

In such a system, suppose there are N sensor nodes (i.e., N participating users) that monitor and report on the same sensing task in real time. At each time t, let $r_i(t)$ denote the real state value of the user u_i under the current sensing task. If $r_i(t) = 1$, it means the user u_i holds the target state at time t. If $r_i(t) = 0$, it means the user u_i is not in the target state at time t.

Stream Model. Since the system model is distributed, the stream model is also distributed. Each user u_i holds an infinite source stream dataset D_i. Each tuple (u, s, t) in D_i is an atomic record denoting the user u was in state s at time t. Based on system model, $r_i(t)$ denotes the user u_i's stream data at time t for a given state s. Therefore, for each user u_i, her/his infinite stream data X_i can be represented as $X_i = \{r_i(1), r_i(2), \cdots, r_i(t), \cdots\}$.

3.3 Problem Definition

Let $X_i = \{r_i(1), r_i(2), \cdots, r_i(t), \cdots\}$ denote the real infinite stream of user u_i. Let $Z_i = \{z_i(1), z_i(2), \cdots, z_i(t), \cdots\}$ denote the reported noisy stream data of user u_i. Let $\widehat{C} = \{\hat{c}(1), \hat{c}(2), \cdots, \hat{c}(t), \cdots\}$ denote the statistics of the infinite stream that is estimated in the server side. Based on system model and stream model, the problem in this paper can be formalized as: *collecting the stream data of each user with ω-event ϵ-local differential privacy and estimating the stream statistics while guaranteeing a good data utility.*

4 Our Solution

In this section, we first show the design rationales and an overview of our mechanism. Then, we detail our solution for real-time stream statistics with local differential privacy.

4.1 Design Rationales and Overview

In mobile crowdsensing, the privacy-preserving of real-time stream statistics should cover two-fold concerns:

1) *Local privacy protection.* The users participate in the mobile crowdsensing task in a distributed way. Thus, the stream data of each user can be distributedly perturbed on the user side (i.e., client side) and then sent to the central server. To this end, we apply the randomized response technique to achieve local differential privacy for each user.
2) *Time correlation exploitation.* The direct perturbation on each user's stream data will destroy the time correlation of stream data, leading to a low data utility. Thus, we propose to learn the underlying time correlations of the stream under LDP and smooth the excessive noise to improve data utility.

Based on the above considerations, we proposed a local privacy-preserving mechanism for real-time stream statistics in mobile crowdsensing, as shown in Fig. 1. Our mechanism mainly consists of two components, that is, 1) local perturbation on the client side and 2) aggregation and estimation with error reduction on the server side. The details of each component will be introduced in the next sections.

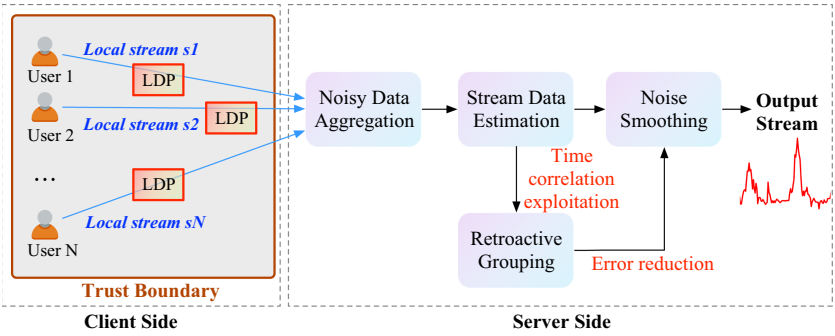


Fig. 1. A high-level overview of our mechanism

4.2 Local Perturbation on the Client Side

In the context of continuous stream processing, event-level [11], user-level [10], and ω-event [17] privacy models are three common paradigms. Since ω-event privacy can protect the stream data within any window of ω timestamps, we adopt it to protect stream in real-time settings in this paper. Besides, concerning the untrusted server in crowdsensing, the local perturbation of our mechanism will achieve ω-event ϵ-LDP for each user's stream data at each timestamp.

For any ω timestamps, the stream data of user u_i can be represented as an ω-bit vector B_i. At time t, the ω-bit vector of user u_i can be denoted as $B_i^t = \{r_i(t-\omega+1), \cdots, r_i(t-1), r_i(t)\}$. Then, our purpose can always achieve ω-event ϵ-LDP for ω-bit vector B_i^t at time t. To this end, we apply randomized response technique on stream data $r_i(t)$ at time t with privacy budget ϵ/ω. The specific perturbation rule is as follows.

$$z_i(t) = \begin{cases} r_i(t), & \text{with probability } f \\ 1 - r_i(t), & \text{with probability } 1-f \end{cases} \tag{2}$$

where f determines the privacy level and $z_i(t)$ is the noisy stream data of real stream data $r_i(t)$.

To guarantee ω-event ϵ-LDP, we set f as $\frac{e^{\epsilon/\omega}}{e^{\epsilon/\omega}+1}$, which will ensure the local privacy for the stream data within any window of ω timestamps. The Algorithm 1 shows the pseudocode of local perturbation. As we can see, after perturbation, each user's noisy data $z_i(t)$ will be sent to the server.

Theorem 2. *The local perturbation in Algorithm 1 satisfies ω-event ϵ-LDP. That is, Algorithm 1 provides ω-event ϵ-LDP for each user.*

Proof. Based on the definition of LDP, for any pairs of input stream data $r(t)$ and $r'(t)$ at time t, the ratio of probabilities for different outputs will satisfy

$$\frac{\mathbb{P}[\mathcal{A}(r(t))]}{\mathbb{P}[\mathcal{A}(r'(t))]} \leq \frac{f}{1-f} = \frac{\frac{e^{\frac{\epsilon}{\omega}}}{e^{\frac{\epsilon}{\omega}}+1}}{1-\frac{e^{\frac{\epsilon}{\omega}}}{e^{\frac{\epsilon}{\omega}}+1}} = e^{\frac{\epsilon}{\omega}} \tag{3}$$

Algorithm 1: Local Perturbation with ω-event ϵ-LDP on the Client Side

Input: $\mathbf{X} = [X_1, X_2, \cdots, X_N]^\top$: real infinite stream data, where each $X_i = \{r_i(1), r_i(2), \cdots, r_i(t), \cdots\}$,
ϵ: privacy budget,
ω: sliding window.
Output: $\mathbf{Z} = [Z_1, Z_2, \cdots, Z_N]^\top$: sanitized infinite stream data, where each $Z_i = \{z_i(1), z_i(2), \cdots, z_i(t), \cdots\}$.

```
for each timestamp t do
    for each user u_i (i = {1, 2, ..., N}) do
        Obtain the stream data r_i(t) of user u_i;
        Perturb r_i(t) to z_i(t) according to Eqn. (2);
        Send z_i(t) to server;
Return Z;
```

Thus, each user achieves $\frac{\epsilon}{\omega}$-LDP at each time t.

In addition, within any ω timestamps, it always holds that $\sum_{t-\omega+1}^{t} \frac{\epsilon}{\omega} = \epsilon$. Hence, Algorithm 1 provides ω-event ϵ-LDP for each user.

4.3 Aggregation and Estimation with Error Reduction on the Server Side

After local perturbation, the server will receive the noisy stream data $\mathbf{Z}(t)$ at each time t, where $\mathbf{Z}(t) = [z_1(t), z_2(t), \cdots, z_N(t)]^\top$. From $\mathbf{Z}(t)$, we can compute the received count of stream data, denoted as $\widehat{N}_t$. Let $x(t)$ be the unbiased estimation of stream data at time t. Based on the perturbation rule (i.e., Eq. (2)), it holds that

$$\frac{e^{\frac{\epsilon}{\omega}}}{e^{\frac{\epsilon}{\omega}} + 1} \cdot x(t) + \frac{1}{e^{\frac{\epsilon}{\omega}} + 1} \cdot (N - x(t)) = \widehat{N}_t \tag{4}$$

Therefore, the unbiased estimation of stream data can be computed as

$$x(t) = \left(\widehat{N}_t - \frac{N}{e^{\frac{\epsilon}{\omega}} + 1}\right) \cdot \frac{e^{\frac{\epsilon}{\omega}} + 1}{e^{\frac{\epsilon}{\omega}} - 1} \tag{5}$$

Theorem 3. *The variance of the unbiased estimation of stream data is* $\frac{N e^{\frac{\epsilon}{\omega}}}{(e^{\frac{\epsilon}{\omega}} - 1)^2}$ *at each time* t.

Proof. The specific proof refers to [26].

However, adding noise at each time t will incur high perturbation errors due to the sparsity and time correlations of stream data. Direct perturbation will break the time correlations among stream data, leading to a low data utility. Therefore, from the perspective of time correlation exploitation, we propose a retroactive grouping strategy to reduce total noise and improve the data utility.

The retroactive grouping strategy aims to divide the timestamps into groups based on the stream received so far. The grouping rule is that the stream data in the same group have a small deviation value from their average. Specifically, this paper adopts deviation function to compute the deviation value between the stream data and the current group. Let $\mathcal{G}_t$ be the group at time t, and $X[\mathcal{G}_t]$ be the set of corresponding stream values in group $\mathcal{G}_t$. The deviation function is formalized as

$$f\left(X[\mathcal{G}_t]\right) = \sum_{j \in \mathcal{G}_t} \left| x(j) - \frac{\sum_{j \in \mathcal{G}_t} x(j)}{|\mathcal{G}_t|} \right| \tag{6}$$

where $|\mathcal{G}_t|$ is the size of group.

The deviation value essentially reflects the absolute difference of a set of data from their average. Thus, based on deviation value, we can easily evaluate the quality of each potential group and conduct grouping at each timestamp.

The specific process of stream aggregation and estimation with error reduction on the server side is shown in Algorithm 2. We can see that the unbiased estimation of stream data will be computed firstly at each timestamp t, as shown in lines 2–3.

Algorithm 2: Stream Aggregation and Estimation with Error Reduction on the Server Side

Input: $\mathbf{Z} = [Z_1, Z_2, \cdots, Z_N]^\top$: the reported noisy stream data of N users.
Output: $\widehat{C} = \{\hat{c}(1), \hat{c}(2), \cdots, \hat{c}(t), \cdots\}$: the estimated count of infinite stream data.

1 **for** *each timestamp* t **do**
2 Estimate the received count $\widehat{N}_t$ of stream data;
3 Compute the unbiased estimation of stream data as $x(t) = \left(\widehat{N}_t - \frac{N}{e^{\epsilon/\omega}+1}\right) \cdot \frac{e^{\epsilon/\omega}+1}{e^{\epsilon/\omega}-1}$;
4 **if** $t = 1$ **then**
5 $\mathcal{G}_t = \{t\}$, and set $\mathcal{G}_{state} = open$;
6 **if** $\mathcal{G}_{state} = open$ **then**
7 Compute group deviation value $v_t = f(X[\mathcal{G}_{t-1} \cup t])$;
8 **if** $v_t < \theta_t$ **then**
9 $\mathcal{G}_t = \mathcal{G}_{t-1} \cup \{t\}$, and set $\mathcal{G}_{state} = open$;
10 **else**
11 $\mathcal{G}_t = \{t\}$, and set $\mathcal{G}_{state} = close$;
12 **else**
13 $\mathcal{G}_t = \{t\}$, and set $\mathcal{G}_{state} = open$;
14 Smooth noise as $\hat{c}(t) = \text{median}\{x(j) | j \in \mathcal{G}_t\}$;
15 **Return** $\widehat{C} = \{\hat{c}(1), \hat{c}(2), \cdots, \hat{c}(t), \cdots\}$

Next, lines 4–13 show the procedure of the retroactive grouping strategy. As we can see, the first timestamp is directly taken as a group and the group state is open, that is $\mathcal{G}_t = \{t\}$ and $\mathcal{G}_{state} = open$. Here, the open state of a group means that a new timestamp can be added to it. For each of the successive timestamps, depending on the group state, there will be two grouping operations.

1) If group state is open, we then compute group deviation value v_t as $v_t = f(X[\mathcal{G}_{t-1} \cup t])$ based on Eq. (6). If the deviation value v_t is smaller than threshold θ_t, the group will be updated by adding the current timestamp t into it and the group state is still open, that is $\mathcal{G}_t = \mathcal{G}_{t-1} \cup \{t\}$ and $\mathcal{G}_{state} = open$, as shown in lines 8–9. Otherwise, the current timestamp t is taken as a new group and the group state is close, that is $\mathcal{G}_t = \{t\}$ and $\mathcal{G}_{state} = close$, as shown in lines 10–11.
2) If group state is close, the current timestamp t will be directly taken as a new group and the group state is open, that is $\mathcal{G}_t = \{t\}$ and $\mathcal{G}_{state} = open$, as shown in lines 12–13.

Based on the noisy stream count and time group result, we can smooth the excessive noise at each time t and obtain the final estimated count of stream, that is $\hat{c}(t) = \text{median}\{x(j)|j \in \mathcal{G}_t\}$, as shown in the $14th$ line. Here, we adopt median smoothing in this paper.

From Algorithm 2, it can be observed that the threshold θ_t is a key parameter and plays an important role in grouping. An appropriate threshold determines the effectiveness of the retroactive grouping strategy. Intuitively, due to the dynamics of stream data, the threshold that selected according to the changing trend of the stream data will be a good threshold. Therefore, we no longer predefine a fixed threshold and instead update the threshold based on the observed feedback errors at the current timestamp. In this paper, we update the threshold based on the feedback error like [24].

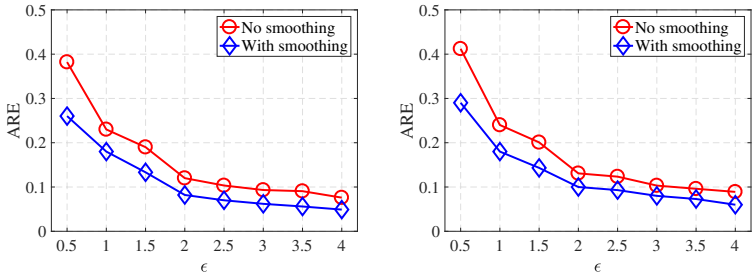
5 Experiments

This section presents the performance evaluations of our proposed framework for real-time stream statistics with ω-event ϵ-LDP.

5.1 Evaluation Setup

Datasets. Our experiments are conducted on two real-world datasets.

- Retail [1] is a retail market basket dataset, which contains 16,470 unique items. We take the Retail dataset as stream data by taking the size of items as the length of the stream.
- Kosarak [1] is a webpage click-stream dataset, which was collected from a Hungarian online news portal. Kosarak dataset contains around one million users and 41,270 categories. We pre-process the Kosarak dataset into stream data by taking the size of click categories as the length of the stream.



(a) Retail (b) Kosarak

Fig. 2. Utility evaluation of retroactive grouping-based noise smoothing when ϵ changes ($\omega = 20$)

Metrics. We use the metric of Average Relative Error (ARE) to evaluate the data utility of our proposed mechanism. In formal, ARE is defined as

$$\mathrm{ARE}(C, \widehat{C}) = \frac{1}{T} \sum_{t=1}^{T} \frac{c(t) - \hat{c}(t)}{\max\{c(t), \delta\}} \tag{7}$$

where C and $\widehat{C}$ are the real and noisy stream statistics, respectively. The parameter δ is used to mitigate the effect of zero value or excessively small value.

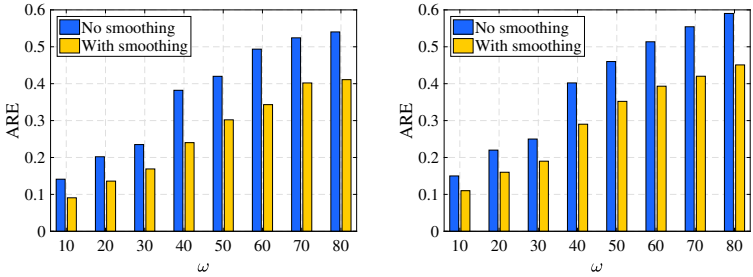
Experimental Environment. We simulate a crowdsensing system for real-time stream statistics. The system has an aggregation server. Each user acts as a participant node and perturbs her/his stream using Algorithm 1 before sending data to the server. The privacy-preserved stream data are aggregated in the server end to compute the stream statistics. All algorithms and experiments are implemented using Python 2.7, running on a Windows 10 PC with CPU i7-10700, 16 GB RAM.

5.2 Experimental Results

In what follows, we present the evaluation results of our mechanism from different aspects varying from different privacy parameters. In all experiments, we consider privacy budget $\epsilon \in \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4\}$ and sliding window $\omega \in \{10, 20, 30, 40, 50, 60, 70, 80\}$. Each experiment is conducted 100 times and the average result is reported.

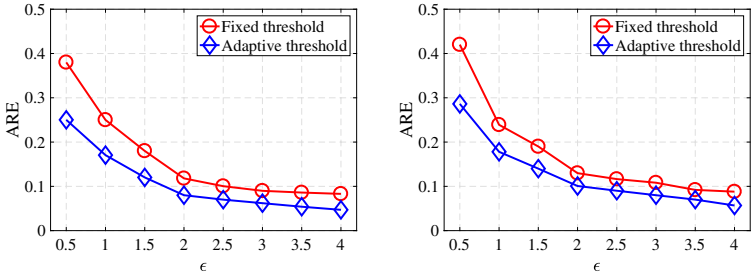
1) Performance evaluation of retroactive grouping-based noise smoothing.

Figure 2 gives the ARE of our mechanism on Retail and Kosarak datasets when using noise smoothing or not. On the whole, the AREs decrease constantly with an increase of ϵ since the privacy protection level becomes lower when ϵ increases. Besides, as we can see, the ARE under noise smoothing is always smaller than that without noise smoothing. This is because the noise smoothing



(a) Retail (b) Kosarak

Fig. 3. Utility evaluation of retroactive grouping-based noise smoothing when ω changes ($\epsilon = 1$)



(a) Retail (b) Kosarak

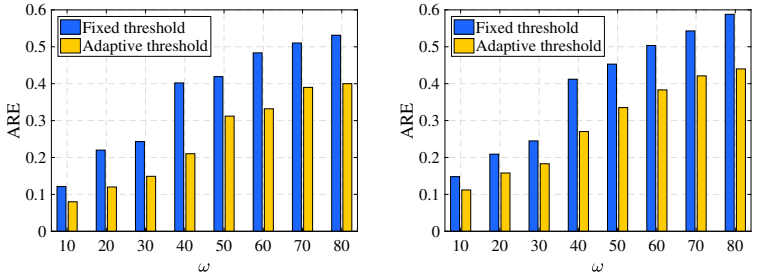
Fig. 4. Utility evaluation of threshold selection when ϵ changes ($\omega = 20$)

can reduce the total noise based on the consideration of time correlations of stream data. Therefore, it demonstrates that the retroactive grouping strategy can learn the stream changing trends, thus making noise smoothing effective.

Figure 3 shows the evaluation results of retroactive grouping-based noise smoothing when ω varies from 10 to 80. As we can see, the AREs increase constantly with the increase of ω since the privacy budget at each timestamp becomes smaller when ω increases. A smaller privacy budget leads to a larger variance. Nonetheless, it can be observed that the ARE is certainly reduced when applying noise smoothing. Thus, this demonstrates again that the retroactive grouping-based noise smoothing can greatly improve the data utility when dealing with dynamic stream data.

2) Performance evaluation of adaptive threshold selection.

Figure 4 gives the evaluation results of threshold selection varying ϵ from 0.5 to 4. At first, the AREs on both datasets decrease with the increase of ϵ since a larger ϵ means a lower privacy protection level, thus holding a higher data utility. Moreover, we can see that the adaptive threshold selection in our mechanism gives better accuracy in all cases than that using a fixed threshold, which is as expected. The reason here is that an adaptive threshold essentially



(a) Retail (b) Kosarak

Fig. 5. Utility evaluation of threshold selection when ω changes ($\epsilon = 1$)

reflects the changing trends of the dynamic stream data, which will make the time grouping more accurate, thus improving the data utility.

Figure 5 presents the AREs on two datasets varying ω from 10 to 80. Overall the performance gets worse when the size of the sliding window (i.e., ω) increases. A large ω can ensure the privacy of each user within a long time slot, thus reducing the accuracy. Moreover, the AREs are effectively reduced in both figures when using an adaptive threshold. This proves again that the utility improvement using an adaptive threshold is much significant than using a fixed threshold, which is as expected.

6 Conclusion

In this paper, we propose a privacy-preserving mechanism with ω-event ϵ-local differential privacy (LDP) for real-time stream statistics in mobile crowdsensing. Our mechanism applies LDP to introduce perturbation on the client side, which provides strong privacy guarantees for each user. Concerning the dynamics and time correlations of stream data, we leverage a retroactive grouping strategy to learn the time correlations and then smooth the excessive noise. All experiments have shown the effectiveness of our proposed mechanism in terms of improving data utility.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (No. 62102311) and in part by the Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 21JK0913).

References

1. Frequent itemset mining dataset repository. http://fimi.ua.ac.be/data/
2. Abdelhameed, S.A., Moussa, S.M., Khalifa, M.E.: Restricted sensitive attributes-based sequential anonymization (RSA-SA) approach for privacy-preserving data stream publishing. Knowl. Based Syst. **164**, 1–20 (2019)

3. Acs, G., Castelluccia, C., Chen, R.: Differentially private histogram publishing through lossy compression. In: Proceedings of IEEE ICDM, pp. 1–10 (2012)
4. Acs, G., Castelluccia, C.: A case study: privacy preserving release of spatio-temporal density in Paris. In: Proceedings of ACM SIGKDD, pp. 1679–1688 (2014)
5. Al-Hussaeni, K., Fung, B.C., Iqbal, F., Dagher, G.G., Park, E.G.: Safepath: differentially-private publishing of passenger trajectories in transportation systems. Comput. Netw. **143**, 126–139 (2018)
6. Cao, Y., Yoshikawa, M., Xiao, Y., Xiong, L.: Quantifying differential privacy under temporal correlations. In: Proceedings of IEEE ICDE, pp. 821–832 (2017)
7. Chen, Y., Machanavajjhala, A., Hay, M., Miklau, G.: PeGaSus: data-adaptive differentially private stream processing. In: Proceedings of ACM CCS, pp. 1375–1388 (2017)
8. Ding, B., Kulkarni, J., Yekhanin, S.: Collecting telemetry data privately. In: Advances in Neural Information Processing Systems, pp. 3571–3580 (2017)
9. Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Local privacy and statistical minimax rates. In: IEEE Annual Symposium on Foundations of Computer Science, pp. 429–438 (2013)
10. Dwork, C.: Differential privacy in new settings. In: Proceedings of ACM-SIAM SODA, pp. 174–183 (2010)
11. Dwork, C., Naor, M., Pitassi, T., Rothblum, G.N.: Differential privacy under continual observation. In: ACM Symposium on Theory of Computing, pp. 715–724 (2010)
12. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
13. Erlingsson, Ú., Pihur, V., Korolova, A.: Rappor: randomized aggregatable privacy-preserving ordinal response. In: Proceedings of ACM SIGSAC CCS, pp. 1054–1067 (2014)
14. Fan, L., Xiong, L.: An adaptive approach to real-time aggregate monitoring with differential privacy. IEEE Trans. Knowl. Data Eng. **26**(9), 2094–2106 (2014)
15. Guo, B., et al.: Mobile crowd sensing and computing: the review of an emerging human-powered sensing paradigm. ACM Comput. Surv. (CSUR) **48**(1), 1–31 (2015)
16. Joseph, M., Roth, A., Ullman, J., Waggoner, B.: Local differential privacy for evolving data. In: Advances in Neural Information Processing Systems, pp. 2381–2390 (2018)
17. Kellaris, G., Papadopoulos, S., Xiao, X., Papadias, D.: Differentially private event sequences over infinite streams. VLDB Endow. **7**(12), 1155–1166 (2014)
18. Li, M., et al.: All your location are belong to us: breaking mobile social networks for automated user location tracking. In: Proceedings of ACM MobiHoc, pp. 43–52 (2014)
19. Rastogi, V., Nath, S.: Differentially private aggregation of distributed time-series with transformation and encryption. In: Proceedings of ACM SIGMOD, pp. 735–746 (2010)
20. Sun, G., Sun, S., Yu, H., Guizani, M.: Toward incentivizing fog-based privacy-preserving mobile crowdsensing in the internet of vehicles. IEEE Internet Things J. **7**(5), 4128–4142 (2020)
21. Wang, J., Zhu, R., Liu, S.: A differentially private unscented Kalman filter for streaming data in IoT. IEEE Access **6**, 6487–6495 (2018)
22. Wang, Q., Zhang, Y., Lu, X., Wang, Z., Qin, Z., Ren, K.: Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy. IEEE Trans. Dependable Secure Comput. **15**(4), 591–606 (2016)

23. Wang, Q., Zhang, Y., Lu, X., Wang, Z., Qin, Z., Ren, K.: RescueDP: real-time spatio-temporal crowd-sourced data publishing with differential privacy. In: Proceedings of IEEE INFOCOM, pp. 1–9 (2016)
24. Wang, T., Yang, X., Ren, X., Zhao, J., Lam, K.Y.: Adaptive differentially private data stream publishing in spatio-temporal monitoring of IoT. In: IEEE 38th International Performance Computing and Communications Conference (IPCCC), pp. 1–8 (2019)
25. Wang, T., Zhang, X., Jingyu, F., Xinyu, Y.: A comprehensive survey on local differential privacy toward data statistics and analysis. Sensors **20**(24), 1–47 (2020)
26. Wang, T., Blocki, J., Li, N., Jha, S.: Locally differentially private protocols for frequency estimation. In: USENIX Security Symposium, pp. 729–745 (2017)
27. Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. IEEE Trans. Knowl. Data Eng. **23**(8), 1200–1214 (2011)
28. Yargic, A., Bilge, A.: Privacy-preserving multi-criteria collaborative filtering. Inf. Process. Manag. **56**(3), 994–1009 (2019)
29. Zhang, X., Hamm, J., Reiter, M.K., Zhang, Y.: Statistical privacy for streaming traffic. In: Network and Distributed System Security Symposium (NDSS), pp. 1–15 (2019)
30. Zheng, X., Cai, Z., Li, Y.: Data linkage in smart internet of things systems: a consideration from a privacy perspective. IEEE Commun. Mag. **56**(9), 55–61 (2018)