IEEE TRANSACTIONS ON
DEPENDABLE AND
SECURE COMPUTING

Real-time and Spatio-temporal Crowd-sourced Social Network Data Publishing with Differential Privacy

Qian Wang, *Member, IEEE,* Yan Zhang, Xiao Lu, Zhibo Wang, *Member, IEEE,*
Zhan Qin, *Student Member, IEEE,* and Kui Ren, *Fellow, IEEE*

Abstract—Nowadays gigantic crowd-sourced data from mobile devices have become widely available in social networks, enabling the possibility of many important data mining applications to improve the quality of our daily lives. While providing tremendous benefits, the release of crowd-sourced social network data to the public will pose considerable threats to mobile users' privacy. In this paper, we investigate the problem of real-time spatio-temporal data publishing in social networks with privacy preservation. Specifically, we consider continuous publication of population statistics and design RescueDP - an online aggregate monitoring framework over infinite streams with w-event privacy guarantee. Its key components including adaptive sampling, adaptive budget allocation, dynamic grouping, perturbation and filtering, are seamlessly integrated as a whole to provide privacy-preserving statistics publishing on infinite time stamps. Moreover, we further propose an enhanced RescueDP with neural networks to accurately predict the values of statistics and improve the utility of released data. Both RescueDP and the enhanced RescueDP are proved satisfying w-event privacy. We evaluate the proposed schemes with real-world as well as synthetic datasets and compare them with two w-event privacy-assured representative methods. Experimental results show that the proposed schemes outperform the existing methods and improve the utility of real-time data sharing with strong privacy guarantee.

Index Terms—Crowd-sourced data, social networks, privacy preservation, realtime data publishing, differential privacy.

1 Introduction

In the past few years, social networks, especially mobile social networks, have become an essential part of people's daily life, which are mainly driven by the crazy growth on both quantity and computation capabilities of mobile devices. With the powerful sensors in mobile devices, mobile social networks have promoted the development of many new services. For example, it has become a fashion that users "check in" at some places and share their location information to their friends. In addition to the proactive behavior sharing information by users themselves, the service providers also collect users' location data more and more frequently on the ground of providing more accurate and personalized services. These gigantic data crowd-sourced from users could enable the possibility of many appealing data mining applications to improve the quality of our daily lives, e.g., traffic monitoring, route planning, pedestrian counting, hot spot tracing, targeted advertising and accident warning, which gives the service providers the impetus to perform various data mining tasks on users' data, and even share or release the data to the third parties or the greater public to make more profit.

However, the crowd-sourced data collected from mobile users in social networks are usually private, e.g., location information, and releasing these sensitive data to third parties would raise users' concerns from a privacy perspective, which may even discourage the users to access social network services [2]–[5]. A recent study [6] on human mobility data obtained from mobile phone carriers found that human mobility traces are highly unique and even coarse datasets provide little anonymity, i.e., with some outside information the data can be linked back to an individual. This and other related findings indicate fundamental constraints to an individual's privacy and have important implications for the development of private data publishing frameworks to protect the privacy of individuals [7], [8]. Figure 1 shows the typical architecture of crowd-sourced private data publishing.

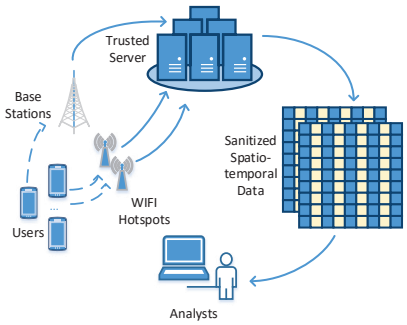


Fig. 1: An architecture of real-time crowd-sourced data collecting and publishing.

Differential privacy [9], which provides privacy for statistics publishing with strong theoretical guarantee, has emerged as a compelling privacy model. It makes almost no assumption about attackers background information, which means that even given the background information about a user, the attacker cannot learn

- *Qian Wang, Yan Zhang, Xiao Lu, and Zhibo Wang are with The State Key Lab of Software Engineering, and School of Computer Science, Wuhan University, P. R. China.*
  E-mail: {qianwang, stong, luxiao, zbwang}@whu.edu.cn
- *Zhan Qin and Kui Ren are with the Department of Computer Science and Engineering, The State University of New York at Buffalo, USA.*
  E-mail: {zhanqin, kuiren}@buffalo.edu
- *A preliminary version [1] of this work has been accepted by the 35th IEEE International Conference on Computer Communications (INFOCOM), 10-15 April 2016, San Francisco, CA, USA.*

Digital Object Identifier 10.1109/TDSC.2016.2599873

any extra information about the user from the published data. A common approach to realize data publishing with differential privacy is to perturb data prior to their publishing and hide sensitive information about the individuals in the process of statistical analysis/data mining. Until now, most existing literatures on differentially private data publishing and analysis have focused on one-time release of static data [10]–[13]. However, in many applications, crowd-sourced data are collected and published in real-time (called data streaming), and the data miners conduct repeated computations to produce outputs continually, e.g., for the purpose of monitoring traffic conditions, search trends or incidence of influenza, etc. This recently motivates extensive research efforts on real-time data publishing with differential privacy guarantee [14], [15], hiding any single "action" taken by users at specific time stamps (event-level privacy) or all the "actions" of any user throughout the entire stream (user-level privacy) [14].

Most of the state-of-the-art on differentially private schemes either focus on event-level privacy on infinite streams [16]–[18] or user-level privacy on finite streams [15]. Kellaris et al. [14] merged the gap between event-level and user-level in streams by proposing the *w-event ϵ-differential privacy* model (w-event privacy for short) to strike a good balance between utility and privacy which protects any event sequence occurring within any window of w time stamps. In this paper, we study the problem of real-time spatio-temporal crowd-sourced data publishing. Specifically, we divide an area into several disjoint regions and publish the population statistics of each region in real-time for potential applications. w-event privacy is preferred for this infinite stream to protect any users' mobility trace over any successive w time stamps.

Our work is inspired by [14] where two schemes, called budget distribution (BD) and budget absorption (BA), were proposed to achieve w-event privacy over infinite streams. However, their schemes are not one-size-fits-all and have their own limitations. More specifically, BD and BA only allocated portion of the entire privacy budget ϵ for data perturbation at any successive w time stamps, so some portion of the budget was wasted and thus the utility [1] of the released data was reduced. In addition, they ignored the difference among regions and allocated equivalent budget for all regions at a time stamp, which results in large relative error to regions with small counts while streams with small counts are very common in many real-world applications, especially for real-time spatio-temporal data publishing with data sparsity [19], [20]. In this paper, we argue that the scheme design should take into account the characteristics of streaming data itself, and there exists much room to improve the accuracy of published data while satisfying w-event privacy.

In this paper, we propose RescueDP for REal-time Spatio-temporal Crowd-soUrcEd Data Publishing with Differential Privacy. Our scheme dynamically groups regions with small statistics together by taking into consideration the similarity of data change, and adds Laplace noise to each group instead of each region so that the effects of perturbation error on small statistics can be eliminated. To efficiently allocate privacy budget, we design an adaptive sampling mechanism and an adaptive budget allocation mechanism that dynamically adjust the sampling rate according to data change and allocate appropriate proportion of privacy budget to sampling points within any successive w time stamps. Due to the rich temporal correlation of the time series, we finally use Kalman Filter to improve the accuracy of the published data. We prove that RescueDP satisfies w-event privacy and its practicality in terms of high utility is validated through extensive experiments. Moreover, we further propose an enhanced RescueDP scheme by taking advantage of neural networks to accurately predict the statistics of each region. A new sampling mechanism and a dynamic programming based dynamic grouping mechanism are proposed to find the optimal grouping strategy and thus improve the accuracy of the released data. Our main contributions can be summarized as follows.

We design RescueDP, a novel real-time crowd-sourced data publishing framework for social networks over infinite streams with w-event privacy protection. The design of RescueDP includes adaptive sampling, adaptive budget allocation, dynamic grouping, perturbation and filtering, and it takes data dynamics into account and tackles the challenge raised by data sparsity. Our theoretical analysis proves that RescueDP satisfies w-event privacy.

We further propose an enhanced RescueDP scheme by taking advantage of neural networks to accurately predict the statistics of each region, and propose a new sampling and dynamic programming based dynamic grouping mechanisms to improve the accuracy of the released data while still maintaining w-event privacy

We implement and evaluate the proposed schemes with real-world as well as synthetic datasets, and compare them with two w-event privacy-assured representative benchmarks. Experimental results show that our solutions outperform existing methods and improve the accuracy/utility of real-time data sharing with strong privacy guarantee.

The remainder of the paper is organized as follows. Section 2 discusses the literature on privacy preserving data publishing techniques. Section 3 introduces some preliminary knowledge of differential privacy and presents the problem statement. We present RescueDP and analyze its privacy in Section 4. We present the enhanced RescueDP with neural networks in Section 5. We evaluate the performance of the proposed schemes with extensive experiments in Section 6 and finally conclude the paper in Section 7.

2 Related Work

Recent studies have pointed out the severe privacy risks of releasing users' data [21] [22], and a lot of privacy-preserving data publishing techniques have been proposed.

There are many proposals achieving k-anonymity in publishing users location data [23], moving traces [24], search logs or web browsing data [25] [26]. However, [27] and other studies pointed out that these methods always have a limitation on the background knowledge of attackers, which is not practical in many real world scenarios. Whereas differential privacy, first proposed by [9] in 2006, is an appealing notion that provides a much stronger privacy guarantee that one's privacy would be protected even the attackers already have the information of everyone else in the database. Meanwhile, compared to k-anonymity, differential privacy provides a more rigorous method to measure the privacy level. After the first differentially private mechanism (Laplace mechanism) presented in [10], a great number of techniques have been proposed for data publishing while achieving differential privacy in the past several years.

1. Without confusion, we interchangeably use utility and accuracy throughout the paper.

Several techniques have been proposed for publishing traditional statistical data derived from static database, such as [28] [29] [30], while recently several works have been focusing on releasing time series data. There are two major directions for the latter, one focuses on off-line data release and the other focuses on real-time data publishing. The off-line data release techniques process the whole time series data at one time, while the real-time data publishing techniques process the time series data in a streaming way.

The works of [31] [20] focused on off-line time series release. [31] proposed an algorithm based on Discrete Fourier Transform (DFT), which achieves differential privacy by injecting noise into the discrete Fourier coefficients. [20] proposed a differentially private scheme that combines sampling, clustering, Fourier perturbation and smoothing processes to release the statistics of every IRIS cells in Paris each hour over a whole week, while preserving privacy with high data utility. The works of [16] [18] studied continual counting queries on time series under differential privacy, which can be applied in real-time for monitoring purpose. However, they only provide *event-level* privacy guarantee, i.e., they can only protect a user's presence at a single time stamp in the whole data stream.

The most related works to ours are [14] and [15]. Both of them focused on real-time time series release under differential privacy. In [15], Fan et al. proposed a framework called FAST, which is composed of two major components: sampling and filtering, and it can provide *user-level* privacy, i.e., protect the presence of a user in the whole time series. However, their work cannot be applied over infinite time stamps since FAST must pre-assign the maximum times of publications, and their sampling mechanism can only be applied under the condition that each time stamp has an equivalent budget. Kellaris et al. [14] merged the gap between *event-level* privacy and *user-level* privacy by proposing a novel model, called w-*event* ϵ-*differential privacy* (w-event privacy for short). They also proposed two new schemes to achieve w-event privacy. However, their schemes are not one-size-fits-all and have their own limitations as discussed in Section 1. In this paper, we adopt w-event privacy and proposed two schemes to protect users' mobility traces within any w successive time stamps. Our schemes address the above limitations by proposing a sequence of new mechanisms, which take data dynamics into account, allocate budget adaptively for each sampling point, tackle the challenge raised by the data sparsity and are especially suitable for spatio-temporal data publishing for social networks.

3 Preliminaries and Problem Statement

In this section, we introduce some preliminary knowledge of differential privacy and w-event privacy, and present the problem to be studied in this paper.

3.1 Differential Privacy

Differential privacy has become a *de facto* standard privacy model for statistics analysis with provable privacy guarantee. Intuitively, a mechanism satisfies differential privacy if its outputs are approximately the same even if a single record in the dataset is arbitrarily changed, so that an adversary infers no more information from the outputs about the record owner than from the dataset where the record is absent.

Definition 1 (Differential Privacy [9]). A privacy mechanism $\mathcal{M}$ gives ϵ-differential privacy, where $\epsilon > 0$, if for any datasets D and $D^{'}$ differing on at most one record, and for all sets $S \subseteq Range(\mathcal{M})$,

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(D^{'}) \in S], \quad (1)$$

where ϵ is the *privacy budget* representing the privacy level the mechanism provides. Generally speaking, a smaller ϵ guarantees a stronger privacy level, but also requires a larger perturbation noise.

Definition 2 (Sensitivity [10]). For any function $f : \mathcal{D} \rightarrow \mathcal{R}^d$, the sensitivity of f w.r.t. $\mathcal{D}$ is

$$\triangle(f) = \max_{D,D^{'} \in \mathcal{D}} ||f(D) - f(D^{'})|| \quad (2)$$

for all D, $D^{'}$ differing on at most one record.

Laplace mechanism is the most commonly used mechanism that satisfies ϵ-differential privacy. Its main idea is to add noise drawn from a Laplace distribution into the datasets to be published.

Theorem 1 (Laplace Mechanism [10]). For any function $f : \mathcal{D} \rightarrow \mathcal{R}^d$, the Laplace Mechanism $\mathcal{M}$ for any dataset $D \in \mathcal{D}$

$$\mathcal{M}(D) = f(D) + \langle Lap(\triangle(f)/\epsilon) \rangle^d \quad (3)$$

satisfies ϵ-differential privacy, where the noise $Lap(\triangle(f)/\epsilon)$ is drawn from a Laplace distribution with mean zero and scale $\triangle(f)/\epsilon$.

The composition property of differential privacy provides privacy guarantee for a sequence of computations.

Theorem 2 (Sequential Composition [32]). Let $\mathcal{M}_1$, $\mathcal{M}_2$, $\cdots$, $\mathcal{M}_r$ be a set of mechanisms and each $\mathcal{M}_i$ provides ϵ_i-differential privacy. Let $\mathcal{M}$ be another mechanism that executes $\mathcal{M}_1(D)$, $\cdots$, $\mathcal{M}_r(D)$ using independent randomness for each $\mathcal{M}_i$. Then $\mathcal{M}$ satisfies $(\sum_i \epsilon_i)$-differential privacy.

This theorem allows us to distribute the privacy budget ϵ among r mechanisms to realize ϵ-differential privacy.

3.2 w-Event Privacy

w-event ϵ-differential privacy, without confusion w-event privacy is used for short, is a new privacy model proposed in [14], which provides provable privacy guarantee for any event sequence occurring at any window of w time stamps.

Before introducing the definition of w-*event privacy*, we first explain some notions it requires. We call two datasets $D_i, D_i^{'}$ at time stamp i as neighboring if they differs in at most one row. A stream prefix of an infinite series $S = (D_1, D_2, ...)$ at time stamp t is defined as $S_t = (D_1, D_2, ..., D_t)$.

Definition 3 (w-neighboring [14]). Let w be a positive integer, two stream prefixes S_t, $S_t^{'}$ are w-neighboring, if

1) for each $S_t[i]$, $S_t^{'}[i]$ such that $i \in [t]$ and $S_t[i] \neq S_t^{'}[i]$, it holds that $S_t[i]$, $S_t^{'}[i]$ are neighboring, and
2) for each $S_t[i_1]$, $S_t[i_2]$, $S_t^{'}[i_1]$, $S_t^{'}[i_2]$ with $i_1 < i_2$, $S_t[i_1] \neq S_t^{'}[i_1]$ and $S_t[i_2] \neq S_t^{'}[i_2]$, it holds that $i_2 - i_1 + 1 \leq w$.

Definition 4 (w-Event Privacy [14]). A mechanism $\mathcal{M}$ satisfies w-event ϵ-differential privacy, if for all sets $S \subseteq Range(\mathcal{M})$ and all w-neighboring stream prefixes S_t, $S_t^{'}$ and all t, it holds that

$$\Pr[\mathcal{M}(S_t) \in S] \leq exp(\epsilon) \cdot \Pr[\mathcal{M}(S_t^{'}) \in S]. \quad (4)$$

A mechanism satisfying w-event privacy will protect the sensitive information that may be disclosed from a sequence of some length w.

Theorem 3 ([14]). Let $\mathcal{M}$ be a mechanism that takes stream prefix S_t as input, where $S_t[i] = D_i \in \mathcal{D}$, and outputs $\mathbf{s} = (\mathbf{s}_1,..., \mathbf{s}_t) \in Range(\mathcal{M})$. Suppose $\mathcal{M}$ can be decomposed into t mechanisms $\mathcal{M}_1, ..., \mathcal{M}_t$ such that $\mathcal{M}_i(D_i) = \mathbf{s}_i$, each $\mathcal{M}_i$ generates independent randomness and achieves ϵ_i-differential privacy. Then, $\mathcal{M}$ satisfies w-event privacy if

$$\forall i \in [t], \sum_{k=i-w+1}^{i} \epsilon_k \leq \epsilon. \tag{5}$$

This theorem enables a w-event private scheme to view ϵ as the total available privacy budget in any sliding window of size w, and appropriately allocate portions of it across the time stamps. This is the fundamental theorem, based on which we design a novel w-event privacy mechanism for real-time data publishing.

3.3 Problem Statement

Under the architecture of real-time data crowd-sourcing and data publishing in Figure 1, in this paper, we consider the popular statistics application where the "check in" information of mobile users in social networks are crowd-sourced to a trusted server and the statistics (e.g., the number of users) of each region are continually published in real-time to the public. We assume that users travel and "check in" at regions in a two-dimensional area. The "check in" information of a user (e.g. location) is crowd-sourced to a trusted server through cellular network or Wifi hotspots. Based on the "check in" information, the server creates a database D over time. The goal is to continually publish statistics computed on D in real-time with w-event privacy guarantee. Therefore, instead of releasing the true value of statistics, the trusted server applies an appropriate privacy protection mechanism and releases a sanitized version of the raw statistics to the third party (e.g., analysts).

Let D_i denote the database crowd-sourced at time stamp i, where each row corresponds to a user and each column corresponds to the occurrence of users of a region. The value in row u and column j is 1 if a user u appears at region j at time stamp i; 0 otherwise. Note that each row of D_i contains at most one 1 since a user cannot appear at two regions simultaneously. The trusted server wishes to publish the statistic of every column in D_i (the total number of users in each region). Considering the statistics as the results of a query Q on D_i, $Q(D_i) = X_i = (x_i^1, x_i^2, \cdots, x_i^d)$ where d is the total number of regions and x_i^j is the number of users at region j at time stamp i. Since each user can only appear at most one region per time stamp, the sensitivity $\triangle(Q) = 1$. In order to protect the privacy of statistics, a sanitized version of x_i^j, say r_i^j, is released instead of releasing x_i^j directly. Given the statistics at time stamp i, X_i, its sanitized version is denoted by $R_i = (r_i^1, r_i^2, \cdots, r_i^d)$. Therefore, the problem studied in this paper can be formally stated as follows:

Problem 1. Given an infinite multi-dimensional time series $\mathbf{X} = \{X_1, X_2, ..., X_i, ...\}$, for each X_i at time stamp i, release a sanitized version R_i in real-time such that the continued release $\mathbf{R} = \{R_1, R_2, ..., R_i, ...\}$ satisfies w-event ϵ-differential privacy.

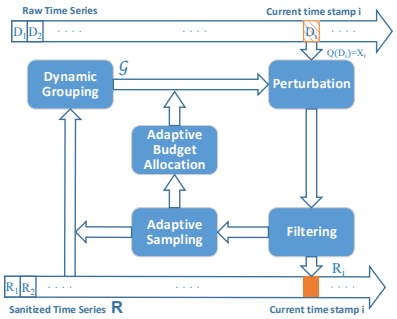


Fig. 2: A high-level overview of RescueDP.

4 RescueDP: Real-time Spatio-Temporal Crowd-sourced Data Publishing with Differential Privacy

In this section, we present RescueDP, a privacy protection scheme to realize real-time spatio-temporal data publishing with w-event privacy guarantee. Figure 2 shows a high-level overview of the RescueDP, which mainly consists of five mechanisms: adaptive sampling, adaptive budget allocation, dynamic grouping, perturbation and filtering.

We propose an adaptive sampling mechanism that adjusts the sampling rate according to data dynamics, which perturbs statistics at selected sampling time stamps and approximates the non-sampled statistics with perturbed statistics at last sampling time stamp. Then we propose an adaptive budget allocation mechanism that dynamically distributes the entire privacy budget ϵ at sampling points over any successive w time stamps. The idea behind the sampling mechanism and the adaptive budget allocation is that non-sampled statistics can be approximated without any budget allocation. Thus, given a fixed ϵ, more budget can be allocated to sampling points within any successive w time stamps, which can reduce the perturbation error introduced by Laplace noise and increase the utility of the released statistics.

We also propose a dynamic grouping mechanism to reduce the perturbation error introduced to regions with small values of statistics. Instead of grouping regions according to their spatial correlation, the dynamic grouping mechanism groups regions with small values of statistics together according to the similarity of their changing trends. This strategy has two advantages: groups dynamically change over time as the statistics change, and regions far away in the space domain can also be grouped together to resist noise. With the grouping strategy and the adaptive budget allocation, we apply Laplace mechanism to perturb the true values of statistics, and then we adopt the filtering mechanism of FAST [15] to improve the accuracy of released data.

Algorithm 1 gives a high-level description of the proposed RescueDP. In the following, we describe the main components of RescueDP in detail.

4.1 Adaptive Sampling

Every noisy data release comes at the cost of budget consumption while the entire budget ϵ is constant. Thus, publishing noisy data at every time stamp will introduce large magnitude of noise when the window size w is large. A common way to overcome this problem is using sampling mechanism which queries and perturbs statistics at selected time stamps and approximates the non-sampled statistics with perturbed sampled statistics. In this way, non-sampled statistics can be approximated without any

Algorithm 1 RescueDP

Input: The raw database D_i at time stamp i
Output: The released statistics R_i
1: Obtain statistics $X_i = Q(D_i)$
2: Find the set of sampling regions at current time stamp
3: Perform *Dynamic Grouping* mechanism on the set of sampling regions to obtain the grouping strategy $\mathcal{G}$
4: Obtain the allocated budget for each sampling region from the *Adaptive Budget Allocation* mechanism
5: Add Laplace noise to groups $\mathcal{G}$ with allocated budget at *Perturbation* mechanism
6: Improve the accuracy of the perturbed statistics by performing *Filtering* mechanism and release the sanitized statistics R_i
7: Determine the new sampling interval with the *Adaptive Sampling* mechanism

budget allocation, and more budget can be allocated to sampling points within any successive w time stamps given a fixed ϵ.

FAST algorithm [15] adopts PID control [33] to adjust the sampling rate according to historical data dynamics. However, it predefines the number of sampling points and allocates equivalent budget to each sampling point, which is not applicable for w-event privacy protection in a sliding window methodology. Note that the private dissimilarity calculation phase in BD and BA [14] is also a kind of sampling mechanism. However, it uses the current raw data to calculate dissimilarity and will always consume a fixed portion of budget which leaves less budget for perturbation. Moreover, it uses the change of total statistics of all regions per time stamp to determine whether to skip the publication, which ignores the difference of changing trend of statistics among regions, i.e. publications of some regions can be skipped while others cannot. Although the authors optimize their mechanism with a column partitioning method, the improvement of optimization is limited.

In this paper, we design a new adaptive sampling mechanism that takes data dynamics and remaining budget into consideration. In particular, we adopt the PID control to characterize data dynamics, and then determine the next sampling interval for each region with the PID error and the remaining budget at next time stamp. Note that different from FAST, we use a different *feedback error* measure to calculate the PID error because the *feedback error* defined in FAST is very sensitive to data changes especially when data value is small which affects the performance of adaptive sampling.

Suppose the current sampling point is k_n and the last sampling point is k_{n-1}. For a region j, the *feedback error* measure is defined as:

$$E^j_{k_n} = |r^j_{k_n} - r^j_{k_{n-1}}|,$$

which is the error between the released data values at current and last sampling points. We use the released data instead of raw data for the concern of privacy protection, which may introduce some error of reflecting data dynamics. However, the error is relative small due to the carefully design of RescueDP, such as the dynamic grouping and the filtering mechanisms. The evaluation results shown in Figure 8 also validate the correctness of this point.

The PID error δ^j for statistics on the jth column of D_{k_n} (i.e., region j) is calculated as follows.

$$\delta^j = K_p E^j_{k_n} + K_i \frac{\sum_{o=n-\pi-1}^{n} E^j_{k_o}}{\pi} + K_d \frac{E^j_{k_n}}{k_n - k_{n-1}}, \quad (6)$$

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 $K_p E^j_{k_n}$ is the proportional error standing for present error; the second term $K_i \frac{\sum_{o=n-\pi-1}^{n} E^j_{k_o}}{\pi}$ is the integral error standing for the accumulation of past error, and π is how many recent errors are taken for integral error; the third term $K_d \frac{E^j_{k_n}}{k_n - k_{n-1}}$ is the derivative error predicting the future error.

Intuitively, the sampling interval should be small when data changes rapidly. However, if the remaining budget is very small, sampling and perturbing statistic at next time stamp may introduce very high perturbation error. In contrast, a better choice might be using a relative large sampling interval so that previous allocated budget can be recycled and approximating the statistic at next time stamp with previous publication. The next sampling interval is determined as follows.

$$I = \max\{1, I_l + \theta(1 - (\frac{\delta^j}{\lambda_r})^2)\}, \quad (7)$$

where I and I_l are the next and last sampling interval of region j, respectively. We let $\lambda_r = 1/\epsilon_r$ to measure the scale of Laplace noise where ϵ_r is the remaining budget at next time stamp. θ is a pre-defined scale factor to adjust the sampling interval and is set to 10 in our experiments. Here we use the relative value of PID error δ^j and the scale of Laplace noise λ_r to decide whether to increase or decrease the sampling interval. In particular, the sampling interval increases when $\delta^j < \lambda_r$ and decreases when $\delta^j > \lambda_r$.

4.2 Adaptive Budget Allocation

w-event privacy requires that the sum of budgets within any sliding window of w time stamps is at most the entire privacy budget ϵ. Although BD and BA were proposed to allocate the privacy budget across w event sequences, they have their limitations. First, only part of privacy budget is used for data perturbation since some budget is used for private dissimilarity calculation, which may introduce large perturbation error. Second, both BD and BA rely on an assumption that the statistics may not change significantly in successive time stamps which makes them not applicable for many real-world scenarios where statistics may change significantly over time.

In this paper, we propose an adaptive budget allocation mechanism taking the trend of data change into consideration to adaptively allocate some portion of the budget at *each sampling point*. In the adaptive sampling mechanism, we dynamically change the sampling interval according to the trend of data change. Generally speaking, the new sampling interval is small when data change rapidly and large when slowly. Therefore, when the sampling interval is small, we can infer that data change rapidly and there will be many sampling points within a window of w time stamps. In this case, we will allocate a small portion of the remaining budget to next sampling point so that more available budget will be left to the successive potential sampling points. While if the sampling interval is large, we can infer that data change slowly and there will be few sampling points within a window of w time stamps. In this case, we will allocate a large portion of the remaining budget to next sampling point.

With this objective in mind, we find that the natural logarithm can perfectly characterize the relationship between the portion p and the sampling interval I. Even better, the portion increases slowly as the interval is large enough which is just what we expect.

Algorithm 2 Adaptive Budget Allocation

Input: Privacy budget ϵ, new sampling interval I, allocated budget for each time stamp $(\epsilon_1, ..., \epsilon_{i-1})$, and the maximum allocated budget at each sampling point ϵ_{max}. Note that $\epsilon_k = 0$ if time stamp k is not a sampling point.

Output: Budget allocation ϵ_i for sampling time stamp i

1: Calculate the remaining budget $\epsilon_r = \epsilon - \sum_{k=i-w+1}^{i-1} \epsilon_k$
2: Calculate the portion $p = \min(\phi \cdot \ln(I+1), p_{max})$
3: Calculate the allocated budget $\epsilon_i = \min(p \cdot \epsilon_r, \epsilon_{max})$

Thus, we let $p = \phi \cdot \ln(I+1)$, where ϕ is a scale factor varies in (0,1]. Since the minimum value of I is 1, we use $\ln(I+1)$ instead of $\ln I$ so that p will not be 0.

The adaptive budget allocation mechanism is formally presented in Algorithm 2. The algorithm first calculates the remaining budget ϵ_r in window $[i-w+1, i]$, where ϵ_r is equal to ϵ minus the sum of budget allocated in window $[i-w+1, i-1]$, to make sure that the total budget spent in the current window is no more than ϵ. The algorithm then calculates the portion p that decides how much budget to be used for current sampling point i. Note that we let $p \leq p_{max}$ to avoid leaving too few budget to next sampling point. Finally, the budget allocated to the current time stamp is calculated, $\epsilon_i = \min(p \cdot \epsilon_r, \epsilon_{max})$ where ϵ_{max} limits the maximum budget to be allocated at each sampling point. The reason we have the limit ϵ_{max} is because that it is not necessary to allocate more budget than ϵ_{max} at each sampling point since the improvement of utility is small when the allocated budget is larger than a threshold ϵ_{max}, say $\epsilon_{max} = 0.2$ when $\epsilon = 1$.

4.3 Dynamic Grouping

The most straight forward way to realize differential privacy is to injecting Laplace noise to each statistic directly which, however, may result in high perturbation error especially for statistics with small values. To overcome the problem, [19] proposes a spatial estimation algorithm that groups regions with small statistics into partitions according to the spatial correlation among regions. However, the grouping is performed off-line at one time, which cannot work well for dynamic scenarios where the spatial correlation among regions are changing over time and the similarity may not hold all the time.

In this paper, we propose a dynamic grouping algorithm that dynamically aggregates regions with small statistics together according to the trend of statistics change other than the spatial correlation among regions. The main idea is that regions with small statistics can be grouped together if their statistics are close and their trends of statistics change are similar. To realize this objective, we use released statistics at previous sampling points to predict the statistic at current sampling point as well as characterize the trend of statistics change. Let $(r^j_{k_{i-\kappa}}, r^j_{k_{i-\kappa+1}}, \cdots, r^j_{k_{i-1}})$ denote the released statistics at previous κ sampling points, and $\bar{x}^j_{k_i}$ denote the predicted statistic at sampling point k_i for region j. We let $\bar{x}^j_{k_i} = \sum_{o=i-\kappa}^{i-1} r_{k_o}/\kappa$, and adopt Pearson Correlation Coefficient, the most commonly used measure of correlation in statistics, to measure the similarity of trend of statistics change. Finally, regions with small statistics and high similarity are grouped together.

The procedures of the dynamic grouping mechanism are described as follows. Note that at each time stamp, dynamic grouping only considers regions that need to be sampled, denoted by P. Let $\mathcal{G}_{k_i}$ be the group strategy at k_i.

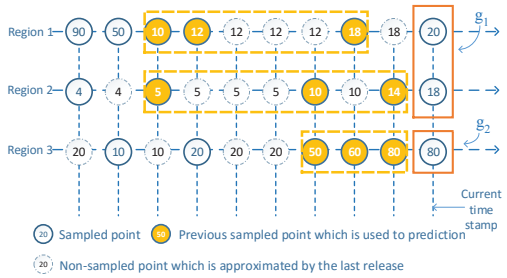


Fig. 3: An illustration of dynamic grouping mechanism.

- Step 1: Predict the statistics at k_i for each region in P (e.g., $\bar{x}^j_{k_i} = \sum_{o=i-\kappa}^{i-1} r_{k_o}/\kappa$ for region j);
- Step 2: Let the region j itself as a group if $\bar{x}^j_{k_i} > \tau_1$ and add the group to $\mathcal{G}_{k_i}$;
- Step 3: Sort all regions not in $\mathcal{G}_{k_i}$ in increasing order according to $\bar{x}^j_{k_i}$, denoted by Ψ;
- Step 4: Initialize a new group (e.g., g) with the first region $\Psi(1)$ in Ψ;
- Step 5: Check the next region in Ψ, denoted by $\Psi(k)$, if the difference between the predicted statistics of $\Psi(1)$ and $\Psi(k)$ is less than τ_3, and the sum of the predicted statistics of the regions in the group g is less than τ_1, go to next step; otherwise, go to step 7;
- Step 6: Calculate the similarity between $\Psi(k)$ and $\Psi(1)$ with the Pearson Correlation Coefficient by using their released statistics at previous κ sampling points. If their similarity is larger than τ_2, remove $\Psi(k)$ from Ψ and put it into group g, otherwise, skip this region. Go to step 5;
- Step 7: Add the group g to $\mathcal{G}_{k_i}$ and remove $\Psi(1)$ from Ψ. If Ψ is not empty, go to step 4; otherwise, return the final strategy $\mathcal{G}_{k_i}$.

Note that several thresholds, τ_1, τ_2 and τ_3, are used for dynamic grouping. τ_1 in step 2 is the noise resistance threshold that reflects whether the statistics of regions have sufficient capacity to resist noise. τ_2 in step 6 is the similarity threshold that decides whether two regions have similar trends of statistics change. τ_3 in step 5 is the error threshold that decides whether two regions are close in terms of statistics.

We use Figure 3 as an example to illustrate the dynamic grouping mechanism. Suppose there are three regions needed to be sampled at current time stamp k_i. Let $\kappa = 3$ so the released statistics at last 3 sampling points (e.g., yellow blocks in each row) are used to predict the statistic at current time stamp. Let $\tau_1 = 50$, $\tau_2 = 0.8$ and $\tau_3 = 20$. The predicted statistics for three regions are 13.3, 9.7 and 63.3, respectively. Since $63.3 > 50$, region 3 is a separate group and is added to $\mathcal{G}_{k_i}$. For region 1 and region 2, their Pearson Correlation Coefficient with $[10, 12, 18]$ and $[5, 10, 14]$ is 0.94. As $0.94 > 0.8$, and $13.3 - 9.7 = 3.6$ is smaller than 20, we can group the two regions together. Thus, the final group strategy is $\mathcal{G}_{k_i} = \{\text{region 3}, \{\text{region 1, region 2}\}\}$.

Dynamic Grouping Optimization: In perturbation phase, we calculate the sum of statistics for each group, perturb it with Laplace mechanism and then average the perturbed data to every region in a group. Note that the average operation may introduce high error since regions having similar trends may still be grouped together even if they differ greatly later. The good thing is that we adopt a filtering mechanism that improves the accuracy of released

data for a region based on previously released data of this region. In this way, the released data of two regions are different even their perturbed data are the same. As the grouping mechanism is based on the released data instead of the perturbed data, the tie can be broken to some extent. In addition, we further break the tie by applying Laplace mechanism to a region separately if it is grouped with other regions for a predefined time period which cannot be long. Our experiments reveal that the problem can be solved efficiently and the errors are small.

4.4 Perturbation

At each time stamp, we apply Laplace mechanism to inject Laplace noise to statistics at sampling regions to provide differential privacy protection. For each non-sampling region, the publication is approximated by its last release.

As shown in Figure 2, the adaptive budget allocation mechanism allocates budget for each region, and the dynamic grouping mechanism provides the grouping strategy. We apply Laplace mechanism on each group instead of each region, and then average the perturbed statistic to each region. To ensure that the total budget for each region at each w time stamps no larger than ϵ, the budget allocated for a group is the smallest budget allocated for regions in the group.

Suppose there is a group g with φ regions. Thus, g consists of φ columns of D_i and $g \subseteq D_i$. Let $f(g)$ denote the statistic function that aggregates the total number of 1 in g. Since each user can only appear at most one region at each time stamp, the sensitivity of f is $\triangle(f) = 1$. Let $\lambda(g)$ denote the scale of Laplace noise injected to $f(g)$. We apply the Laplace mechanism on group g,

$$\begin{aligned}\mathcal{M}(g) &= f(g) + Lap(\lambda(g)) \\ &= \sum_{j=1}^{\varphi} \sum g[j] + Lap(\triangle(f)/\min(\epsilon_{g[j]})),\end{aligned} \tag{8}$$

where $g[j]$ is the jth column of g and $\triangle(f) = 1$.

Then the perturbed statistic for each column/region at group g is calculated as the average of $\mathcal{M}(g)$. That is,

$$\mathcal{M}(g[j]) = \mathcal{M}(g)/\varphi, \quad \forall j = 1, \cdots, \varphi. \tag{9}$$

According to Axiom 2.1.1 in [34], post-processing sanitized data maintains privacy as long as the post-processing algorithm does not use the sensitive information directly. Therefore, if $\mathcal{M}(g)$ provides $\min(\epsilon_{g[j]})$-differential privacy, $\mathcal{M}(g[j])$, $\forall j = 1, \cdots, \varphi$ will also provides $\min(\epsilon_{g[j]})$-differential privacy. However, we would not release $\mathcal{M}(g[j])$ directly since Laplace mechanism may introduce too much noise which affects the data utility. Thus we further use the filtering mechanism to improve the accuracy of released statistics.

4.5 Filtering

Inspired by FAST algorithm [15], we also use Kalman Filter [35] to improve the accuracy of released data by releasing the posterior estimation of the perturbed data. We perform filtering for each sampling region separately. Kalman filter uses a state-space model to realize the posterior estimation, which consists of two steps: prediction and correction. Suppose current time stamp is i and the perturbed statistic is $\mathcal{M}(D_i[j])$ for region j at i. The filtering mechanism is shown in Algorithm 3.

Algorithm 3 Filtering with Kalman Filter

Input: Last release $r_{j|i-1} = r^j_{i-1}$, and noisy observation $z_{j|i} = \mathcal{M}(D_i[j])$
Output: Posterior estimation $\hat{x}_{j|i}$
Prediction
1: $\hat{x}^-_{j|i} = r_{j|i-1}$
2: $P^-_{j|i} = P_{j|i-1} + Q_j$
Correction
3: $K_{j|i} = P^-_{j|i}/(P^-_{j|i} + R_{j|i})$
4: $\hat{x}_{j|i} = \hat{x}^-_{j|i} + K_{j|i}(z_{j|i} - \hat{x}^-_{j|i})$
5: $P_{j|i} = P^-_{j|i-1}(1 - K_{j|i})$

The posterior estimation $\hat{x}_{j|i}$ will be the final released statistic for region j at time stamp i. That is, $r^j_i = \hat{x}_{j|i}$ is the sanitized version of the true statistic x^j_i. Since the filtering mechanism is exactly the same as that in FAST algorithm. Please refer to [15] for more detail.

4.6 Privacy Analysis

Theorem 4. RescueDP satisfies w-event ϵ-differential privacy.

Proof: Among the components of RescueDP, the only one accessing raw data is perturbation, while the others operate on sanitized data. Thus, if we can prove that the perturbation mechanism satisfies w-event ϵ-differential privacy, RescueDP will satisfy w-event ϵ-differential privacy.

According to the grouping strategy $\mathcal{G}$, D_i is divided into n disjoint groups $\{g_1, g_2, ..., g_n\}$ and each group consists of several columns of D_i. Without loss of generality, we take g_1 as an example and assume g_1 has φ_1 columns. According to Equation 8, the laplace mechanism on group g_1 is

$$\begin{aligned}\mathcal{M}(g_1) &= f(g_1) + Lap(\lambda(g_1)) \\ &= \sum_{j=1}^{\varphi_1} \sum g_1[j] + Lap(\triangle(f)/\min(\epsilon_{g_1[j]})),\end{aligned}$$

where $g_1[j]$ is the jth column of g_1 and $\triangle(f) = 1$.

According to Theorem 1, $\mathcal{M}(g_1)$ satisfies $\min(\epsilon_{g_1[j]})$-differential privacy. According to Axiom 2.1.1 in [34], post-processing sanitized data maintains privacy as long as the post-processing algorithm does not use the sensitive information directly. Thus, $\mathcal{M}(g_1[j])$, $\forall j = 1, \cdots, \varphi$, also satisfies $\min(\epsilon_{g_1[j]})$-differential privacy. Similarly, each group performs an independent Laplace mechanism on a column/region in a group g_k satisfying $\min(\epsilon_{g_k[j]})$-differential privacy. Let $\hat{\epsilon}_k$ and ϵ_k denote the budget used for perturbation and the allocated budget given by the adaptive budget allocation mechanism for a region at time stamp k, and we know $\hat{\epsilon}_k \leq \epsilon_k$.

According to Theorem 3, in order to prove that the perturbation mechanism for a region satisfies w-event ϵ-differential privacy, we must need to prove that, for every t and $i \in [t]$, it holds that $\sum_{k=i-w+1}^{i} \hat{\epsilon}_k \leq \epsilon$. Since the adaptive budget allocation mechanism already guarantees that $\sum_{k=i-w+1}^{i} \epsilon_k \leq \epsilon$ for any sliding window w time stamps, and there exists $\hat{\epsilon}_k \leq \epsilon_k$, $\sum_{k=i-w+1}^{i} \hat{\epsilon}_k \leq \epsilon$ can be proved. Therefore, the perturbation mechanism on each group satisfies w-event ϵ-differential privacy. Consequently, RescueDP satisfies w-event ϵ-differential privacy. □

Algorithm 4 Enhanced RescueDP with RNN

Input: The raw database D_i at time stamp i
Output: The released statistics R_i
1: Obtain statistics $X_i = Q(D_i)$
2: Obtain the predicted statistics $P_i = (p_i^1, p_i^2, \cdots, p_i^d)$ for all regions from RNN
3: Perform the *Sampling* mechanism based on P_i to determine which series need to be sampled
4: Perform the *Dynamic Grouping* mechanism on the set of sampling regions to obtain the grouping strategy
5: Obtain the allocated budget for each sampling region from the *Adaptive Budget Allocation* mechanism
6: Add Laplace noise to groups with allocated budget at *Perturbation* mechanism
7: Improve the accuracy of the perturbed statistics by performing *Filtering* mechanism and release the sanitized statistics R_i

5 Enhanced RescueDP with Neural Networks

In RescueDP, the adaptive sampling mechanism and the dynamic grouping mechanism are proposed to better allocate privacy budget within any sliding window of w time stamps accordingly to changing trends of released data. That is, the two mechanisms affect the accuracy of released data and the released data in turn affect the two mechanisms. In particular, in the dynamic grouping mechanism, only the statistics at sampling regions but not all regions are predicted, which may result in lower accuracy for the statistics at non-sampling regions. This motivates us to improve the accuracy of released data by finding a better model to characterize the trend of data change and predict the statistic for all regions at each time stamp.

Recently, the recurrent neural network (RNN) has been regraded as the most suitable neural network for time series analysis. In addition to the non-linear modeling capability and generalization ability from traditional neural networks (e.g., feedforward neural network), RNN employs feedback connections to neurons in the network, which means the outputs of RNN are affected not only by the current outside input, but also the historical outputs from the network. These appealing features give RNN the ability to simulate more complicated systems, and make it a natural alternative model to learn patterns that occur in temporal order. Note that the crowd-sourced spatio-temporal data is obviously non-linear and it is difficult to build an accurate mathematical model for the data without sufficient background information.

In this section, we propose an enhanced RescueDP scheme by utilizing RNN to improve the accuracy of released data while satisfying w-event privacy. In the design, it seems reasonable that a single monolithic network implemented on all regions simultaneously may learn patterns in both temporal and spatial manners. However, it will consume extensive time to train the network and even cannot achieve an acceptable training loss, since the number of weights is too large. Thus, here we choose to use independent RNN for each region to get a better trade-off between the accuracy and the training time. Specifically, after building and training the RNNs off-line, we utilize them to predict the statistics at current time stamp in real-time based on the previously released data, and use the predicted value to perform the new sampling and dynamic grouping mechanisms. Algorithm 4 gives a high-level description of the enhanced RescueDP scheme with RNN.

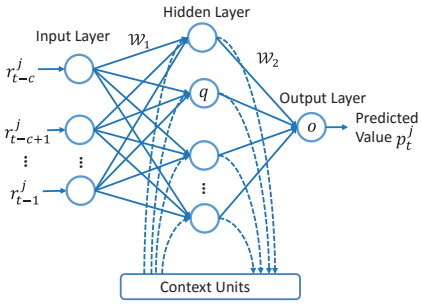


Fig. 4: The architecture of using Elman network with one hidden layer to predict statistics at time stamp t for region j.

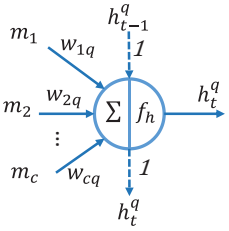


Fig. 5: The output of a neuron in the hidden layer.

Most steps except steps 2-4 of the enhanced RescueDP with RNN are the same with the RescueDP scheme in Algorithm 1. The step 2 predicts the statistic at each region with RNN. We then propose a new *sampling* mechanism in step 3 to determine the time series to be sampled, and a dynamic grouping mechanism in step 4 to group regions together. In the following, we mainly describe the three steps in detail.

5.1 Statistics Prediction with RNN

Elman network [36] also known as simple recurrent neural network is the most widely used RNN, which is powerful in modeling time series while has a compact structure and is easily to be trained. In the enhanced RescueDP scheme, we use Elman network to predict the statistic at each region. Figure 4 shows the architecture of using Elman network for one-step prediction. It has three layers: the input layer, the hidden layer and the output layer. The input layer has c neurons, where c equals the number of previously released data to be used for prediction. The output layer has only 1 neuron since only one-step prediction is needed. The hidden layer contains several hidden neurons and each of them corresponds to a context unit. There are links from the input layer to the hidden layer and from the hidden layer to the output layer where each link is associated with a weight. There are also recurrent connections between the hidden neurons and the context units and each connection has a fixed weight which usually is 1.

As shown in Figure 4, we use the previously released data to predict the statistic at current time stamp for each region with a trained Elman network. Let's use region j for example and the current time stamp is t. The set of previously released data used for prediction is $(r^j_{t-c}, r^j_{t-c-1}, ..., r^j_{t-1})$. The predicted statistic of region j at t is denoted by p^j_t. As shown in Figure 5, for a neuron in hidden layer, say q, the output of q denoted by h^q_t is calculated as follows:

$$h^q_t = f_h(\sum_{i=1}^{c} w_{iq} m_i + h^q_{t-1}), \tag{10}$$

where m_i is the output from the neuron i in the input layer, w_{iq} is the weight of the link from neuron i in the input layer to neuron q in the hidden layer, and h^q_t is the final output of neuron q in the hidden layer. f_h is the activation function which is usually a sigmoid function to capture the non-linearity of the system. Equation 10 clearly indicates that the output of the hidden node at current time stamp (h^q_t) is affected by both the current inputs and the output at previous time stamp (h^q_{t-1}), which gives the network the capability to learn from sequential data.

Algorithm 5 Sampling in Enhanced RescueDP for each region

Input: The current privacy budget ϵ_i, the predicted statistic p_i^j, and the release data at last sampling time stamp r_l^j
Output: Sampling or not
1: Calculate $dis = |p_i^j - r_l^j|$
2: Calculate $\lambda_i^j = 1/\epsilon_i$
3: **if** $dis > \lambda_i$, **then**
4: i is a sampling point for region j, update the sampling interval $I = i - l$.
5: **else**
6: i is not a sampling point for region j
7: **end if**

For the neuron in the output layer, say o, the final output of neuron o can be simply calculated as follows:

$$p_t^j = f_o(W_2 \cdot H_t), \tag{11}$$

where W_2 is the vector of weights from the hidden layer to the output layer, H_t is the vector of outputs of the hidden neurons, f_o is the activation function for the output layer.

Training of Elman network: In order to make prediction accurately and in real-time, the parameters of Elman network, say W_1 and W_2, should be trained offline in advance. It is worth noting that we sample a training set from the database, and use the real values of statistics of the training set to train a more accurate Elman network. Therefore, in the training phase, at each time stamp t for region j, the input is $(x_{t-c}^j, x_{t-c-1}^j, ..., x_{t-1}^j)$, and the target output is the real statistic x_t^j.

With the Elman network, the output denoted by p_t^j is the predicted value of x_t^j. The training error between them can be calculated as follows:

$$Err_t^j = \frac{1}{2}(x_t^j - p_t^j)^2. \tag{12}$$

We then use the back propagation algorithm [37] to propagate the training error back to the neurons in the Elman network, compute the contribution of each neuron to the training error and adjust the weights of links accordingly to reduce the training error. The detail of the training process can be founded in [37]. Finally, the trained Elman network model can be used to predicte the statistics for each region at each time stamp based on previously released data.

5.2 Sampling in Enhanced RescueDP

With the accurate prediction of statistics of each region at each time stamp from the Elman network, it is not necessary to use the PID algorithm with many parameters to adjust the sampling interval. Instead, we propose a simple but practical sampling mechanism in the enhanced RescueDP scheme. The basic idea of the sampling algorithm is that, if the error between the predicted statistic and the last release is smaller than the perturbation error, the statistic at current time stamp can be approximated by the last release and no sampling or privacy budget is needed; otherwise, the current time stamp is a sampling point, and the privacy budget should be allocated.

Algorithm 5 shows the pesudocode of the new sampling algorithm. We use region j as an example and suppose current time stamp is i. The privacy budget that may be allocated at time stamp i is ϵ_i. The predicted statistic of region j at time stamp i is p_i^j and the last release of region j is r_l^j where l is the time stamp of last release. The error between them is $dis = |p_i^j - r_l^j|$, and the perturbation error is $\lambda_i^j = 1/\epsilon_i$. If $dis > \lambda_i^j$, current time stamp i is a sampling point and privacy budget ϵ_i is allocated for region j; otherwise, no sampling or privacy budget is needed for region j, and the statistic can be approximated by r_l^j.

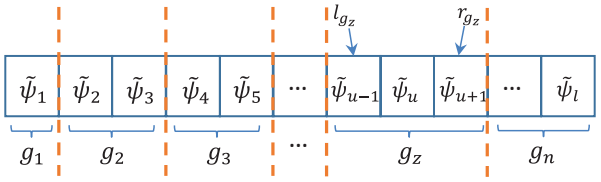


Fig. 6: Finding the optimal grouping strategy with cuts.

5.3 Dynamic Programming based Dynamic Grouping

In RescueDP, although the proposed dynamic grouping works well, it cannot yield the optimal grouping strategy. Since we can obtain more accurate prediction of statistics for each region with Elman network, we are allowed to propose a new dynamic grouping mechanism to realize better grouping of regions. In particular, we propose a dynamic programming based dynamic grouping mechanism to obtain the optimal grouping strategy.

We only consider the grouping of regions to be sampled at each time stamp. Similar to the dynamic grouping strategy in RescueDP, a sampling region itself is a group if its predicted statistic at current time stamp is larger than τ_1. So in the following we only consider the grouping of regions with small predicted statistics (e.g., $p_i^j < \tau_1$).

As shown in Figure 6, let $\tilde{\psi}$ denote the set of regions to be grouped whose statistics are smaller than τ_1 and l denote the total number of regions in $\tilde{\psi}$. Since we would like to group regions with close predicted statistics together, we firstly sort the regions in increasing order according to their predicted statistics p_i^j. That is, $\tilde{\psi}_1 \leq \tilde{\psi}_2 \leq \cdots \leq \tilde{\psi}_l$. Our objective is to find the optimal grouping strategy that minimizes the total group errors. From Figure 6, we can see that grouping regions into groups is actually inserting cuts into the sequence. Thus, the problem of finding the optimal grouping strategy is equivalent to finding the optimal cuts in the sequence of regions.

Let's first give some definitions. Remember that in the perturbation mechanism in RescueDP, we use the average value of statistics in a group to approximate the statistic of each region in the group. Suppose there are n groups and let $\bar{g}_z$ denote the average statistic in group g_z. Thus, the approximate error of a region, say j, in group g_z is $|\bar{g}_z - p_i^j|$, which is the absolute error between the average statistic and the predicted statistic. The group error of g_z is the sum of the approximation error of each region in g_z. Let P_i denote the predicted statistics of regions in $\tilde{\psi}$, and G_i denote a grouping strategy. The total group error of G_i is calculated as follows:

$$G_{err}(P_i, \mathcal{G}_i) = \sum_{g_z} \sum_{j \in g_z} |\bar{g}_z - p_i^j|. \tag{13}$$

The objective is to find the optimal grouping strategy G_i that minimizing the total group error $G_{err}(P_i, \mathcal{G}_i)$. As we mentioned, the problem of finding the optimal grouping strategy is equivalent to finding the optimal cuts in the sequence of regions. The problem is similar to but different from the classical rod-cutting problem, which exhibits optimal substructure and can be solved by the

Algorithm 6 Dynamic Grouping in enhanced RescueDP

Input: The set of sampling regions with statistics smaller than τ_1 at time stamp i: $\tilde{\psi}$, the predicted statistic for each sampling region in $\tilde{\psi}$, and the number of groups: n
Output: The grouping strategy $\mathcal{G}_i$

```
Sort the regions in ψ̃ in increasing order according to p_i^j
Initialize array S[n][l]
for r = 1 to n do
    for m = r to l do
        SAE_min ← ∞
        for μ = r to m do
            error = S[r − 1][μ − 1] + SAE(μ, m)
            SAE_min = min(SAE_min, error)
        end for
        S[r][m] = SAE_min
    end for
end for
Obtain the optimal grouping strategy G_i from S[n][l]
```

dynamic programming solution. Therefore, we propose a dynamic programming based dynamic grouping mechanism to find the optimal grouping strategy.

The key of a dynamic programming solution is to characterize the optimal substructure of the problem. Recall that l is the length of the sequence of regions. Let $S[n][l]$ denotes the minimum total group error of cutting the sequence of length l to n groups. For a group g_z, let (l_{g_z}, r_{g_z}) denote the range of group g_z. The error of g_z is $Err(g_z) = SAE(l_{g_z}, r_{g_z}) = \sum_{j=l_{g_z}}^{r_{g_z}} |\bar{g}_z - p_i^j|$. Then, the optimal substructure can be characterized as follows:

$$S[n][l] = \min_{n-1 \le l' \le n-1} (S[n-1][l'] + SAE(l'+1, l)), \quad (14)$$

where $SAE(l'+1, l)$ is the group error for the group with range $(l'+1, l)$. With this optimal substructure, we can find the optimal grouping strategy with a bottom-up dynamic programming method.

Algorithm 6 shows the pseudocode of the dynamic programming based dynamic grouping mechanism. Recall that a sampling region itself is a group if its predicted statistic is larger than τ_1. These individual groups should be added to the grouping strategy G_i obtained from Algorithm 6 to form the final grouping strategy.

The dynamic programming based dynamic grouping mechanism has a time complexity of $\mathcal{O}(l^2 n)$, where l is the number of regions needed to be grouped and n is the number of final groups. It is worth noting that at each time stamp, only a small portion of regions need to be sampled and the number of regions needed to be grouped is even less, so the grouping process for a large database usually is very quick. Compared to the dynamic grouping mechanism in the RescueDP scheme, this mechanism uses the more accurate predicted statistics and could find the optimal grouping strategy with solid mathematical theory, which could lead to higher utility of released data.

Theorem 5. The enhanced RescueDP scheme satisfies w-event ϵ-differential privacy.

Proof: We propose three new mechanisms in the enhanced RescueDP scheme, statistics prediction with RNN, the new sampling, and the dynamic programming based dynamic grouping mechanisms, while the other mechanisms, adaptive budget allocation, perturbation and filtering, are the same with the RescueDP scheme. The same as RescueDP, only the perturbation mechanism access the raw data. Since we already proved that the perturbation mechanism satisfies w-event ϵ-differential privacy in Theorem 4, the enhanced RescueDP scheme satisfies w-event ϵ-differential privacy as well. □

6 Experimental Evaluation

In this section, we evaluate the performance of RescueDP and the enhanced RescueDP [2] on both real-world and synthetic datasets. We compare their performance with BD and BA, the only two schemes designed for real-time data publishing with w-event privacy guarantee and optimize them according to [14]. All of the schemes are implemented in Python on a machine with Intel Core i5-4460S CPU 2.9GHz and 12GB RAM, running Windows 10. Each experiment is conducted 100 times and the point in the performance figures is the average value of 100 times of each experiment.

Datasets: We conduct the experiments on two real-world datasets, namely ECML/PKDD 15: Taxi Trajectory Prediction (I) [38] and World Cup [39]. The former provides an accurate dataset containing the trajectories of 442 taxies running in the city of Porto for one year from $01/07/2013$ to $30/06/2014$. The dataset contains a total of 1,710,671 trajectories and each contains a list of GPS coordinates reported by a taxi every 15 seconds. We divide the area of Porto into 40×120 disjoint regions and each region is about $80 \times 110m^2$. We report the number of taxies in each region every 15 minutes and compress the 1-year data to 7 days with 672 time stamps. In such a scenario, our algorithm protects any single trajectory of any taxi over at most w time stamps.

The World Cup dataset consists of all the requests made to the FIFA 1998 World Cup Web site between April 30, 1998 and July 26, 1998. It contains 1,352,804,107 web server logs and each log consists of a client ID, a requested URL, a time stamp, etc. We randomly choose 2,000 URLs as the test set, create a stream from the set and publish the data per hour, which has a total of 2010 time stamps. At each time stamp, say i, we aggregate the data to a vector X_i, such that $X_i[j]$ is the number of users requesting URL j at time stamp i (each user can only browse a single URL per time stamp). In this scenario, our scheme can protect any sequence of browsing histories of a user over at most w time stamps.

We also conduct experiments on a synthetic spatio-temporal dataset generated by the famous network-based moving users generator Brinkhoff [40]. We use the road map of San Joaquin and creat a dataset with 1000 time stamps, There are 100000 users at the beginning and then every 5000 new users are added per time stamp. The generator randomly creates a trajectory for each user, and a user disappears from the map once it reaches its destination. We divide the map into 200×300 disjoint regions and each region is approximately $270 \times 280\ m^2$ in reality. Since there are too much regions that have never been visited in the suburbs of the area, we choose a 50×50 part of area in the center of San Joaquin. We calculate the number of users in each region to get the statistic vector X_i at every time stamp i. Note that our scheme protects any single trajectory of any user over at most w time stamps.

Table 1 shows the statistics distribution of each dataset. We calculate the average statistics of each region for all time stamps and see the distribution. For example, for the Taxi dataset, there are 3158 regions whose average statistics fall in [0,1], and there is

2. We use "E-RescueDP" to denote the enhanced RescueDP in the figures

Range	Taxi	World Cup	San Joaquin
0∼1	3158	1753	952
1∼10	1467	197	397
10∼100	174	44	843
>100	1	6	308

TABLE 1: The data distribution of each dataset.

only 1 region whose statistic is larger than 100. We can see that the the three datasets behaves strong data sparsity, which pose a great challenge for privacy-preserving data publishing. This also implies the importance of designing dynamic grouping mechanisms.

Utility of the released data: We use the Mean Absolute Error (MAE) and Mean Relative Error (MRE) as the utility metric to evaluate the performance of these mechanisms.

For any region, let $\mathbf{x} = \{x_1, ..., x_n\}$ denote the raw time series and $\mathbf{r} = \{r_1, ..., r_n\}$ denote the sanitized time series. The MAE and MRE for this region are

$$\mathbf{MAE}(\mathbf{x}, \mathbf{r}) = \frac{1}{n}\sum_{i=1}^{n}|r_i - x_i|, \tag{15}$$

$$\mathbf{MRE}(\mathbf{x}, \mathbf{r}) = \frac{1}{n}\sum_{i=1}^{n}\frac{|r_i - x_i|}{\max(\gamma, x_i)}. \tag{16}$$

For the bound γ, we set its value to 0.1% of $\sum_{i=1}^{n} x_i$ to mitigate the effect of excessively small results. In experiments, we first calculate the MAE and MRE for each region and then figure out the average of all regions as the final results.

In the experiments, we set $K_p = 0.9$, $K_i = 0.1$, $K_d = 0$ and $\pi = 3$ for the PID controller. $\phi = 0.2$ and $p_{max} = 0.6$ are set for the adaptive budget allocation. In dynamic grouping, we let $\tau_1 = 30$ for Taxi and World Cup datasets, $\tau_1 = 40$ for San Joaquin dataset, $\tau_2 = 0.5$ and $\tau_3 = 25$ for all datasets. Without explanation, we set $w = 200$ and $\epsilon = 1$ for all experiments. For the group number n in E-RescueDP, we test different values of n with the training set of datasets, and find that E-RescueDP almost achieves the highest utility when n is around $\frac{\tau_1}{6}$. Specifically, we set n=5 for Taxi and World Cup datasets, and n=6 for San Joaquin dataset.

Training of Elman network. Table 2 shows the detailed settings of training Elman network. We use the last $c = 30$ released data to be the input of Elman network. The numbers of neurons are 30, 80, and 1 in the input layer, the hidden layer and the output layer, respectively. We take hyperbolic tangent function and linear function as the activation functions of the hidden layer and the output layer, respectively. For each dataset, we choose $10\% \sim 20\%$ of the whole dataset to be the training set and use the rest as the testing set. In our experiments, we found that the training loss (also training accuracy) of the three networks went stable after 2000 epochs. We then evaluated the prediction error (measured by MRE) of the three networks on the validation set, and the experimental results show that all of them achieve the error in the range of $12\% \sim 15\%$, which is small and acceptable in our application scenarios. Besides, training for each region individually with 2000 epochs on the training set costs about 3 minutes, which is time-consuming, so parallelization is used to accelerate the training process. Note that the training process is done off-line, which would not affect the performance of real-time data publishing.

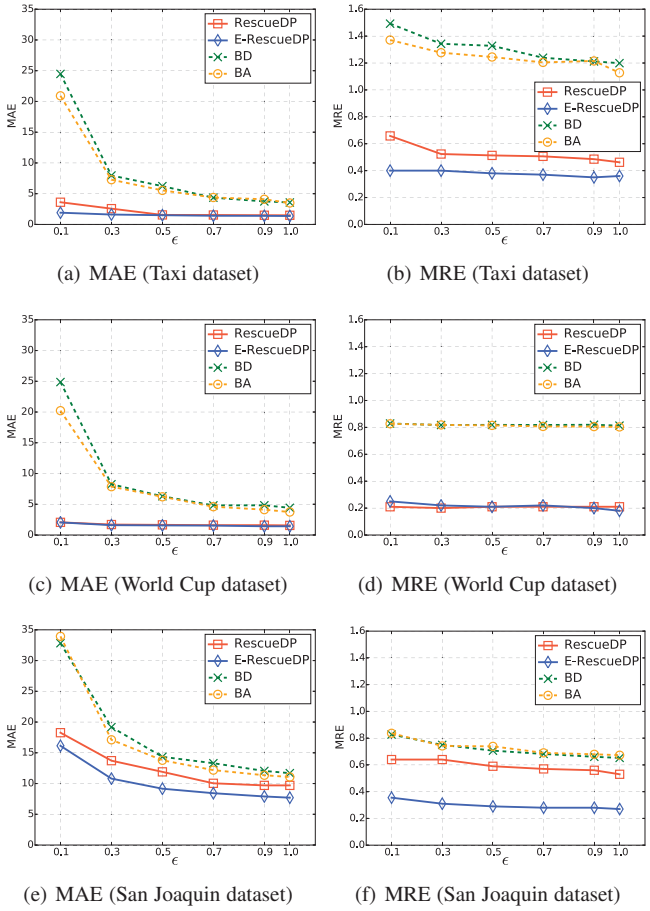
Utility vs Privacy. Figure 7 shows the comparison of MAE and MRE between the proposed schemes with BA and BD when ϵ varies from 0.1 to 1.0. We can see that both MAE and MRE of all the these mechanisms decrease when ϵ increases over all datasets. This is because that smaller noise is required when ϵ is larger. We can also observe that both MAE and MRE of RescueDP and E-RescueDP are much smaller than that of BA and BD over all datasets.

There are several reasons that RescueDP outperforms BA and BD. First, RescueDP has more available budget for private perturbation than BA and BD at any successive w time stamps, since some portion of the budget is used for similarity calculation at each time stamp in BA and BD. Second, the dynamic grouping strategy of RescueDP improves the capacity of resisting of Laplace noise for regions with small statistics by grouping them together. Third, the accuracy of released data is improved by the filtering mechanism of RescueDP.

We can also observe that E-RescueDP outperforms RescueDP in terms of utility for almost all datasets. This is because that the E-RescueDP achieves more accurate prediction of statistics for all

Parameter	Value
c	30
The structure of RNN	(30, 80, 1)
Activation functions	(tangent, linear)
Loss function	Mean square error
Batch size	10
Number of epochs	2000

TABLE 2: Setting of Elman network.

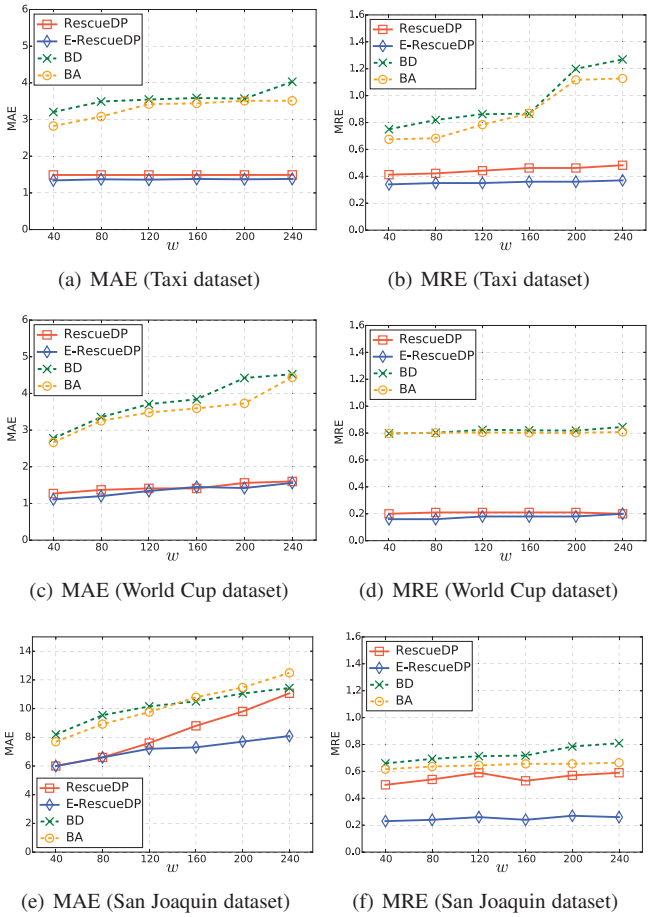


(a) MAE (Taxi dataset) (b) MRE (Taxi dataset)

(c) MAE (World Cup dataset) (d) MRE (World Cup dataset)

(e) MAE (San Joaquin dataset) (f) MRE (San Joaquin dataset)

Fig. 7: Utility comparison when ϵ changes ($w = 200$).

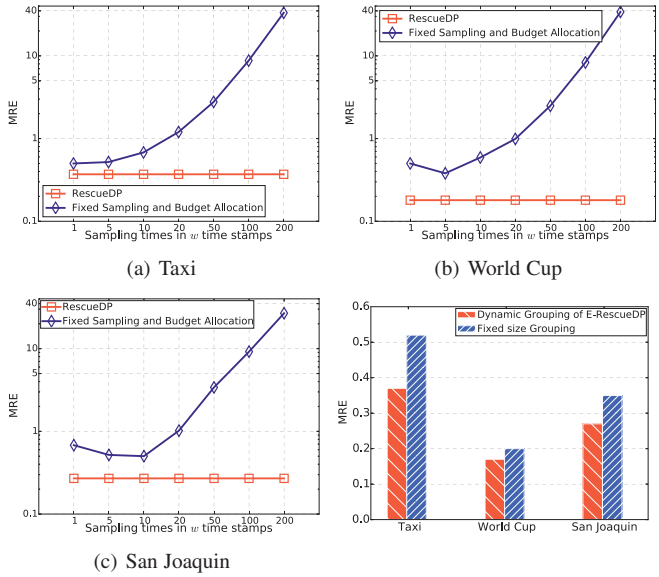


(a) MAE (Taxi dataset) (b) MRE (Taxi dataset)

(c) MAE (World Cup dataset) (d) MRE (World Cup dataset)

(e) MAE (San Joaquin dataset) (f) MRE (San Joaquin dataset)

Fig. 8: Utility comparison when w changes ($\epsilon = 1$).



(a) Taxi (b) World Cup

(c) San Joaquin

Fig. 9: Effects of Adaptive Sampling and Adaptive Budget Allocation mechanism.

Fig. 10: Effects of dynamic programming based dynamic grouping in E-RescueDP.

regions at each time stamp, and the optimal grouping strategy can improve the utility of the released data. Note that E-RescueDP does not improve the the utility of released data compared to RescueDP for the World Cup dataset. This is because that the World Cup dataset does not have the spatio-temporal property as the other datasets.

Utility vs. w. Figure 8 shows the comparison of utility between RescueDP and E-RescueDP with BA and BD when w changes. We can observe again that RescueDP and E-RescueDP outperforms BA and BD on all datasets when w changes. The MAE and MRE of BA and BD increase as w increases, which is due to the deficiency of their budget allocation schemes. The exponential decay of budget in BD will introduce very large noise as more publications occur in a larger window. When w increases, in order to ensure the total budget always less than ϵ within a window, BA may skip many potential publications and thus results in larger errors.

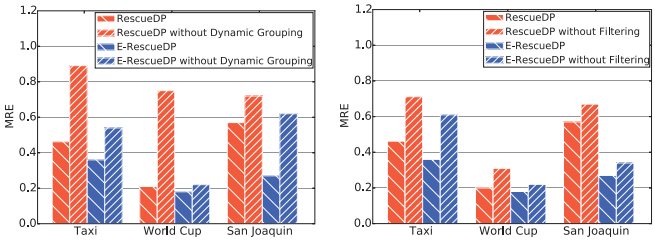
The MAE and MRE of RescueDP and E-RescueDP are relative stable when w changes. This is because that they employ an adaptive budget allocation mechanism which takes window size w and remaining budget into consideration to adaptively allocate budget on sampling points. The dynamic grouping strategy and the filtering mechanism also help increase the utility of the released data. Therefore, the careful design of RescueDP and E-ResueDP makes them robust to the changes of window size. Again, E-RescueDP achieves better utility than RescueDP in almost all datasets.

Effects of Adaptive Sampling and Adaptive Budget Allocation in RescueDP. In order to evaluate the effects of adaptive sampling and adaptive budget allocation mechanisms in RescueDP, we compare RescueDP with a naive scheme having fixed sampling and budget allocation mechanisms. Specifically, we vary the number of samples s in a window of w time stamps, and allocate $\frac{\epsilon}{s}$ for each sampling point in the naive scheme. Here, we set ϵ=1 and w=200.

As shown in Figure 9, the MRE of the naive scheme with fixed sampling and budget allocation mechanisms increases quickly as the sampling times increases. This is because that the allocated budget for each sampling point decrease as the number of sampling increases, which results in large perturbation error and lower utility of released data. In contrast, the utility of RescueDP would not be affected since it applies adaptive sampling and adaptive budget allocation mechanisms.

Effects of dynamic programming based dynamic grouping in E-RescueDP. We evaluate the effects of dynamic grouping mechanism of E-RescueDP. Specifically, we evaluate E-RescueDP with the dynamic grouping mechanism or a fixed grouping mechanism. For the E-RescueDP with a fixed grouping mechanism, the regions with small statistics are equally divided into n groups. As shown in Figure 10, the utility is better if the dynamic grouping mechanism is adopted for E-RescueDP. Note that the improvement of using dynamic grouping mechanism is not that obvious, which is because the utility is already high even E-RescueDP uses the fixed grouping mechanism.

Effects of Dynamic Grouping. We conduct experiments of RescueDP with and without dynamic grouping over three datasets to evaluate the effects of dynamic grouping mechanism. Figure 11(a) shows the results of utility comparison. We observe that dynamic grouping reduces MRE significantly in all three datasets. Therefore, we can conclude that dynamic grouping mechanism improves the utility of sanitized data significantly and it is practical in many real-world scenarios. For E-RescueDP, we notice that, the effect of dynamic grouping is not that significant, especially for the World cup dataset. This is because that the sampling mechanism of E-ResuceDP is based on the predicted statistics, and the majority



(a) MRE comparison (b) MRE comparison

Fig. 11: Utility comparison: (a) with and without dynamic grouping mechanism; (b) with and without filtering mechanism.

	BA	BD	RescueDP	E-RescueDP
Time Complexity	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d^2)$	$\mathcal{O}(nd^2)$
Taxi (d = 4800)	$5 \times 10^{-3}s$	$5 \times 10^{-3}s$	$0.8 \times 10^{-2}s$	$0.9s$
World Cup (d = 2000)	$2 \times 10^{-3}s$	$2 \times 10^{-3}s$	$0.6 \times 10^{-2}s$	$0.8s$
San Joaquin (d = 2500)	$2.4 \times 10^{-3}s$	$2.4 \times 10^{-3}s$	$1.1 \times 10^{-2}s$	$1.2s$

TABLE 3: Comparison of running time.

of regions in World Cup dataset have a very small statistics at each time stamp. Thus, the sampling mechanism would sample few regions based on the nearly constant predicted value from the well-trained network, which further leads to a fewer regions to be grouped, so the improvement is not that significant.

Effect of Filtering. We conduct experiments of RescueDP with and without filtering over three datasets to evaluate the effects of filtering mechanism. 11(b) shows the results of utility comparison. We can observe that MRE is smaller when the filtering mechanism is adopted. That is, the adoption of Kalman filter does improve the utility of the released data over all datasets. Therefore, it is appropriate to integrate the filtering mechanism to RescueDP to improve the utility of the released data.

Running Time. Table 3 shows the comparison of time complexity of the four mechanisms. Here, d is the number of regions. As can be seen, BA and BD are the fastest mechanisms with time complexity $\mathcal{O}(d)$. RescueDP with time complexity of $\mathcal{O}(d^2)$ is slower than BA and BD but faster than E-RescueDP. E-RescueDP has the time complexity of $\mathcal{O}(nd^2)$, where n is the number of groups. As discussed above, the value of n is usually small (n is set to 5 and 6 in our experiments). Thus, we can conclude that our proposed methods greatly improve the data utility with moderate additional computation cost.

7 Conclusions

In this paper, we investigated the problem of privacy preserving data publishing in social networks, and proposed RescueDP, a real-time spatio-temporal crowd-sourced data publishing scheme with w-event privacy protection. We designed a framework for RescueDP consisting of mechanisms of adaptive sampling, adaptive budget allocation, dynamic grouping, perturbation and filtering. The carefully design of these mechanisms enables RescueDP to efficiently adjust the sampling rate and distribute privacy budget within any sliding window of w time stamps.

Moreover, we further proposed an enhanced RescueDP with neural networks to accurately predict the statistics of each region at each time stamp. With the accurate statistics prediction, we propose a new sampling mechanism to decide whether to sample a region or not, and a dynamic programming based dynamic grouping mechanism to find the optimal grouping strategy that minimizing the total group error and thus improving the utility of released data. Our theoretical analysis proved that both RescueDP and the enhanced RescueDP satisfy w-event privacy. Extensive experiments over real-world and synthetic datasets showed that the proposed schemes outperform existing method and improves the utility of real-time data publishing with strong privacy guarantee.

Acknowledgment

Qian and Zhibo's research are supported in part by National Natural Science Foundation of China (Grant No. 61373167, 61502352, 61272453), National Basic Research Program of China (Grant No. 2014CB340600), National High Technology Research and Development Program of China (863 Program, Grant No. 2015AA016004), Wuhan Science and Technology Bureau (Grant No. 2015010101010020), Fundamental Research Funds for the Central Universities (Grant No. 2042016kf0137), and Natural Science Foundation of Hubei Province and Jiangsu Province (Grant No. 2015CFB203, BK20150383). Kui's research is supported in part by US National Science Foundation under grant CNS-1262277.

References

[1] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Rescuedp: Real-time spatio-temporal crowd-sourced data publishing with differential privacy," in *Proc. of IEEE INFOCOM'16*, 2016, pp. 1152–1160.

[2] M. Li, H. Zhu, Z. Gao, S. Chen, L. Yu, S. Hu, and K. Ren, "All your location are belong to us: breaking mobile social networks for automated user location tracking," in *Proc. of ACM MobiHoc'14*, 2014, pp. 43–52.

[3] H. Li, L. Sun, H. Zhu, X. Lu, and X. Cheng, "Achieving privacy preservation in wifi fingerprint-based localization," in *Proc. of IEEE INFOCOM'14*, 2014, pp. 2337–2345.

[4] S. Ji, W. Li, M. Srivatsa, J. S. He, and R. A. Beyah, "General graph data de-anonymization: From mobility traces to social networks," *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 4, pp. 12:1–12:29, 2016.

[5] S. Hu, Q. Wang, J. Wang, Z. Qin, and K. Ren, "Securing sift: Privacy-preserving outsourcing computation of feature extractions over encrypted image data," *IEEE Transactions on Image Processing*, vol. PP, pp. 1–1, DOI: 10.1109/TIP.2016.2 568 460, 2016.

[6] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Scientific reports*, vol. 3, 2013.

[7] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proc. of ACM MobiCom'11*, 2011, pp. 145–156.

[8] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Show me how you move and i will tell you who you are," in *Proc. of ACM SIGSPATIAL'10*, 2010, pp. 34–41.

[9] C. Dwork, "Differential privacy," in *Proc. of ICALP'06*, 2006, pp. 1–12.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography*, 2006, pp. 265–284.

[11] X. Xiao, G. Bender, M. Hay, and J. Gehrke, "ireduct: Differential privacy with reduced relative errors," in *Proc. of ACM SIGMOD'11*, 2011, pp. 229–240.

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

[13] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proc. of VLDB Endowment*, vol. 3, no. 1-2, pp. 1021–1032, 2010.

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

[15] 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.

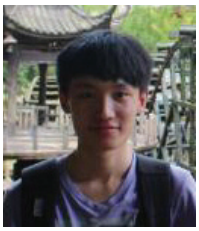
[16] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Transactions on Information and System Security*, vol. 14, no. 3, p. 26, 2011.

[17] T.-H. H. Chan, M. Li, E. Shi, and W. Xu, "Differentially private continual monitoring of heavy hitters from distributed streams," in *Proc. of PETS'12*, 2012, pp. 140–159.

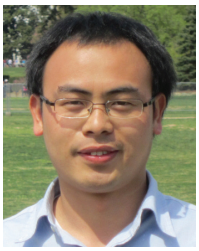
[18] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proc. of ACM STOC'10*, 2010, pp. 715–724.
[19] L. Fan, L. Xiong, and V. Sunderam, "Differentially private multi-dimensional time series release for traffic monitoring," in *Data and Applications Security and Privacy*, 2013, pp. 33–48.
[20] G. Acs and C. Castelluccia, "A case study: privacy preserving release of spatio-temporal density in paris," in *Proc. of ACM SIGKDD'14*, 2014, pp. 1679–1688.
[21] L. Olejnik, C. Castelluccia, and A. Janc, "Why johnny can't browse in peace: On the uniqueness of web browsing history patterns," in *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*, 2012.
[22] M. Barbaro, T. Zeller, and S. Hansell, "A face is exposed for aol searcher no. 4417749," *New York Times*, vol. 9, no. 2008, p. 8For, 2006.
[23] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *Pervasive computing*. Springer, 2009, pp. 390–397.
[24] C.-Y. Chow and M. F. Mokbel, "Trajectory privacy in location-based services and data publication," *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 1, pp. 19–29, 2011.
[25] E. Adar, "User 4xxxxx9: Anonymizing query logs," in *Proc. of Query Log Analysis Workshop, WWW'07*, 2007.
[26] Y. Hong, X. He, J. Vaidya, N. Adam, and V. Atluri, "Effective anonymization of query logs," in *Proc. of ACM CIKM'09*, 2009, pp. 1465–1468.
[27] M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke, "Publishing search logs−a comparative study of privacy guarantees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 520–532, 2012.
[28] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB Journal*, vol. 22, no. 6, pp. 797–822, 2013.
[29] G. Cormode, M. Procopiuc, D. Srivastava, and T. T. Tran, "Differentially private publication of sparse data," *arXiv preprint arXiv:1103.0825*, 2011.
[30] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas, "Releasing search queries and clicks privately," in *Proc. of ACM WWW'09*, 2009, pp. 171–180.
[31] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proc. of ACM SIGMOD'10*, 2010, pp. 735–746.
[32] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proc. of ACM SIGMOD'09*, 2009, pp. 19–30.
[33] M. King, *Process Control: A Practical Approach*. John Wiley & Sons, 2010.
[34] D. Kifer and B.-R. Lin, "Towards an axiomatization of statistical privacy and utility," in *Proc. of ACM PODS'10*, 2010, pp. 147–158.
[35] R. Kalman, "A new approach to linear filtering and prediction problem," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
[36] J. L. Elman, "Distributed representations, simple recurrent networks, and grammatical structure," *Machine learning*, vol. 7, no. 2-3, pp. 195–225, 1991.
[37] F. J. Pineda, "Generalization of back-propagation to recurrent neural networks," *Physical review letters*, vol. 59, no. 19, p. 2229, 1987.
[38] ECML/PKDD 15: Taxi Trajectory Prediction (I), https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i.
[39] World Cup dataset, http://ita.ee.lbl.gov/html/contrib/WorldCup.html.
[40] T. Brinkhoff, "A framework for generating network-based moving objects," *GeoInformatica*, vol. 6, no. 2, pp. 153–180, 2002.

Qian Wang is a Professor with the School of Computer Science, Wuhan University. He received the B.S. degree from Wuhan University, China, in 2003, the M.S. degree from Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, China, in 2006, and the Ph.D. degree from Illinois Institute of Technology, USA, in 2012, all in Electrical Engineering. His research interests include wireless network security and privacy, cloud computing security, and applied cryptography. Qian is an expert under "1000 Young Talents Program" of China. He is a co-recipient of the Best Paper Award from IEEE ICNP 2011. He is a Member of the IEEE and a Member of the ACM.

Yan Zhang received his bachelor's degree from China University of Geoscience. Currently he is pursuing for his master degree in the School of Computer Science, Wuhan University, China. His current research interests are differential privacy in data publishing, data mining and machine learning.

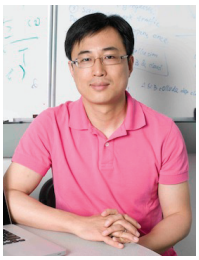


Xiao Lu received his bachelor's degree from China University of Geoscience. Currently he is pursuing for his master degree in the School of Computer Science, at Wuhan University, China. His current research interests are differential privacy in time series analysis and data mining.



Zhibo Wang received the B.E. degree in Automation from Zhejiang University, China, in 2007, and his Ph.D degree in Electrical Engineering and Computer Science from University of Tennessee, Knoxville, in 2014. He is currently an Associate Professor with the School of Computer, Wuhan University, China. His currently research interests include wireless sensor networks and mobile sensing systems. He is a member of IEEE and ACM.

Zhan Qin is currently working toward his Ph.D. degree at the director of the Ubiquitous Security and Privacy Research Laboratory (UbiSeC) in the Computer Science and Engineering Department of the State University of New York at Buffalo. His research interests are in the areas of cloud computing and security, with focus on differential privacy data collection and publication, cybersecurity in smart grid. He is a student member of the IEEE, IEEE COMSOC, and ACM.



Kui Ren is an Associate Professor of computer science at State University of New York at Buffalo. He received his PhD degree from Worcester Polytechnic Institute and both BE and ME degrees from Zhejiang University. Kui's research interests include Cloud Security, Wireless Security, and Smartphone-enabled Crowdsourcing Systems. His research has been supported by NSF, DoE, AFRL, MSR, and Amazon. He is a recipient of NSF CAREER Award in 2011 and Sigma Xi Research Excellence Award in 2012. Kui has published 135 peer-review journal and conference papers. Kui received several Best Paper Awards including IEEE ICNP 2011. Kui currently serves as an associate editor for IEEE Transactions on Information Forensics and Security, IEEE Wireless Communications, IEEE Internet of Things Journal, IEEE Transactions on Smart Grid, IEEE Communications Surveys and Tutorials, Elsevier Pervasive and Mobile Computing, and Oxford The Computer Journal. Kui is a Fellow of IEEE, a member of ACM, a Distinguished Lecturer of IEEE Vehicular Technology Society, and a past board member of Internet Privacy Task Force, State of Illinois.