Research Article

Re-ADP: Real-Time Data Aggregation with Adaptive ω -Event Differential Privacy for Fog Computing

Yan Huo , Chengtao Yong, and Yanfei Lu

School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing, China

Correspondence should be addressed to Yan Huo; yhuo@bjtu.edu.cn

Received 28 April 2018; Accepted 24 June 2018; Published 8 July 2018

Academic Editor: Fuhong Lin

Copyright © 2018 Yan Huo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the Internet of Things (IoT), aggregation and release of real-time data can often be used for mining more useful information so as to make humans lives more convenient and efficient. However, privacy disclosure is one of the most concerning issues because sensitive information usually comes with users in aggregated data. Thus, various data encryption technologies have emerged to achieve privacy preserving. These technologies may not only introduce complicated computing and high communication overhead but also do not work on the protection of endless data streams. Considering these challenges, we propose a real-time stream data aggregation framework with adaptive ω -event differential privacy (Re-ADP). Based on adaptive ω -event differential privacy, the framework can protect any data collected by sensors over any dynamic ω time stamp successively over infinite stream. It is designed for the fog computing architecture that dramatically extends the cloud computing to the edge of networks. In our proposed framework, fog servers will only send aggregated secure data to cloud servers, which can relieve the computing overhead of cloud servers, improve communication efficiency, and protect data privacy. Finally, experimental results demonstrate that our framework outperforms the existing methods and improves data availability with stronger privacy preserving.

1. Introduction

Driven by the development of cyberphysical networks, cloud computing, mobile Internet, context-aware smart devices, and the corresponding data experience explosive growth [1]. Cloud computing provides a good solution to deal with the explosive data growth and realize resource sharing [2]. However, cloud-based services may face many challenges, such as high latency and high overhead at cloud servers, due to the centralized structure and the limitation of network bandwidth. Some researches present a distributed service computing paradigm, called fog networking [3–5]. It allocates the capabilities of data gathering, data processing, computing, and applications to devices located at the edge of the network, so as to provide intelligent services for nearby users.

Although fog computing provides great benefits, sensitive and private information mined from raw data (e.g., social relationships and financial transactions) is also exposed to the risk of disclosure. Even more, due to the complexity and diversity of fog nodes, user privacy in a fog network can easily be disclosure. For example, more than 400,000

electronic eyes in Beijing may lead to privacy leakage (e.g., vehicle location information) by data sharing in vehicular ad hoc networks (VANETs) [6–8]. Similarly, we can also gain illegal access to personal health datasets gathered from various sensors of physical sign in body sensor networks (BSNs) and publish these private data without permission [9–11]. As a result, how to protect user privacy is one of the important research issues in fog computing.

Currently, the protection of aggregated data privacy is mainly divided into two types. The first one is designed based on various encryption technologies, such as homomorphic encryption [6, 8–10]. In this type, the encryption technology may cause huge computational overhead as well as lots of computing resources of cloud services [12]. In addition, the cryptography-based schemes may lower communication system efficiency, especially when the system contains many sensors with high reporting frequency. The reason is that a great number of communication resources may be wasted on transmission of encryption information and the corresponding keys. As a result, this is not suitable for energy-limited sensor networks.

The other type of aggregated data privacy preserving is explored by using differential privacy [13]. Compared with the traditional cryptography-based schemes, differential privacy can protect individuals privacy while improving data accuracy as much as possible. For example, the authors of [14] protect privacy of aggregated data with differential privacy by using machine learning. Although there exist many studies based on differential privacy, some challenges cannot be addressed. These studies do not consider the high correlation of time series so as not to generate real-time aggregated data with high accuracy. However, a practical framework should be able to satisfy batch queries in continuous time by exchanging information only once.

To address these challenges, we propose a real-time privacy-preserving stream data aggregation framework based on adaptive ω -event differential privacy under fog computing architecture. In fog computing, data storage, processing, and applications are concentrated in devices on the edge of the network rather than all in the cloud. This type of architecture reduces the amount of data transmitted to the cloud, increases efficiency, and significantly lowers overhead on the server itself. In addition, a fog center is considered as a data aggregator in our framework. It only reports the aggregation secure results to cloud server. In this way, the efficiency of communication can be greatly improved. Moreover, sensors only report raw data instead of encrypted data because our framework does not utilize complex encryption technology. Finally, many techniques for processing time-series data is exploited in our framework to improve the accuracy of aggregation data, such as adaptive sampling, time-series prediction, and filter.

In a nutshell, the main contributions of the paper are summarized as follows.

- (i) We propose a real-time privacy-preserving stream data aggregation framework based on adaptive ω -event differential privacy under fog computing architecture. The framework releases the overhead of cloud servers and generates aggregation data with differential privacy preserving.
- (ii) In order to promote ω -event differential privacy, we pioneer a novel metric, i.e., quality of privacy (QoP). The QoP design takes into account both the window size ω and errors of published statistics. Using the metric, we adjust the size of window ω adaptively by dint of the design of QoP-based adaptive ω -event mechanism.
- (iii) We exploit the long short-term memory (LSTM) to predict time-series data and design the adaptive sampling scheme to improve the accuracy of aggregation data.
- (iv) We theoretically analyze the privacy of the proposed Re-ADP framework and demonstrate the high accuracy of aggregated data through numerical simulation results.

The rest of the paper is organized as follows. In Section 2, we introduce preliminaries of differential privacy and ω -event privacy. Then, we provide the system model, the adversary model, and the whole Re-ADP framework to illustrate our

FIGURE 1: System model.

problem. In Section 3, we present a QoP-based adaptive ω -event privacy algorithm that includes a dynamical adjustment method of the window size ω . Section 4 presents a smart grouping-based perturbation algorithm, which can reduce the noise added to data significantly. In Section 5, we analyze whether the Re-ADP framework satisfies differential privacy and provides a series of simulation results to discuss the performance of each mechanism in our framework. We then review previous works related to the privacy preserving of aggregated data and differential privacy in Section 6. Finally, Section 7 concludes our paper and explains promising research directions for future work.

2. Problem Statement and Preliminaries

2.1. System Model. The system model, shown as Figure 1, is composed of four layers: the things layer, the fog layer, the core layer, and the cloud layer. The function of each layer is described as follows.

- (i) **Things layer**, consisting of various smart devices, e.g., sensors, mobiles, and actuators, generates and reports raw data to fog layer.
- (ii) **Fog layer**, typically located between IoT devices and core networks, is composed of lots of fog devices. The fog devices can be considered as traditional network devices, such as routers, switches, gateways, or local servers that are specially deployed. In this paper, the devices are mainly composed of local servers and are responsible for (i) gathering and storing data reported from things layer, (ii) computing and aggregating data to satisfy differential privacy, and (iii) responding to query requests from the cloud layer.
- (iii) **Core layer** is in charge of transferring and exchanging data between the fog layer and the cloud layer through network protocols such as IP and MPLS.
- (iv) **Cloud layer** deploys many cloud servers that can analyze massive aggregated data. Using the analyzing results, cloud services can provide a wide range of services.

2.2. Adversary Model. In this paper, we assume that both the cloud layer and the core layer are untrustworthy. They will try to acquire actual values of gathered data or maliciously tamper data. And the fog layer is considered trusted, which means it can acquire raw data but do not disclose data to the third party.

2.3. Differential Privacy Basics. Differential privacy is one of the most popular notions of privacy in the current research field of privacy preservation. The basic idea is that the record of an individual, regardless of whether or not he is in the dataset, has little impact on the final output, thus protecting the privacy of the individual.

Definition 1. (ϵ -differential privacy [13]): a randomized algorithm $\mathcal{A}$ over datasets can provide ϵ -differential privacy guarantee, if any neighboring datasets D and D' are different on at most one record, and any output $\mathcal{O} \in \text{Range}(\mathcal{A})$ satisfies

$$\log \frac{\Pr[\mathcal{A}(D) \in \mathcal{O}]}{\Pr[\mathcal{A}(D') \in \mathcal{O}]} \leq \epsilon, \quad (1)$$

where $\text{Range}(\mathcal{A})$ denotes the range of the randomized algorithm $\mathcal{A}$.

Note that ϵ , called privacy budget, is an important parameter in differential privacy. It represents the privacy level of the randomized algorithm $\mathcal{A}$. More specifically, the level of privacy is inversely proportional to ϵ . Then, a mostly used method to achieve ϵ -differential privacy is the Laplacian mechanism as shown below.

Theorem 2 (the Laplacian mechanism [15]). Let $\mathcal{D}$ denote a set of datasets. Considering a function $f : \mathcal{D} \rightarrow \mathbb{A}$, the Laplacian mechanism $\mathcal{A}$ for any dataset $D \in \mathcal{D}$ is

$$\mathcal{A}(D) = f(D) + \text{Lap}\left(\frac{\Delta(f)}{\epsilon}\right), \quad (2)$$

where the noise follows a Laplacian distribution with mean zero and scale $\Delta(f)/\epsilon$. Here, $\Delta(f)$ denotes sensitivity of f , which is defined as the maximum L_1 norm for any neighboring datasets D_1 and D_2 .

Theorem 3 (sequential composition [16]). Let $\mathcal{A}_i$ provide ϵ_i -differential privacy. Then the sequence of $\mathcal{A}_i(D)$ provides $(\sum_i \epsilon_i)$ -differential privacy.

Obviously, **Theorem 3** shows that the secrecy level of a combination of several differential privacy-preserving algorithms is the sum of all budgets.

2.4. ω -Event Privacy. ω -event privacy, the abbreviation of ω -event ϵ -differential privacy, is a new privacy model proposed by Kellaris et al. [17]. It can protect privacy for any event sequence occurring at any window of ω time stamp.

We define two neighboring datasets at the i th time stamp as D_i and D'_i and a stream prefix of an infinite series $S = (D_1, D_2, \dots)$ at the t th time stamp as $S_t = (D_1, D_2, \dots, D_t)$.

Definition 4 (ω -neighboring [17]). Two stream prefixes S_t, S'_t are ω -neighboring if one of the following two conditions holds:

- (i) for each $S_t[i], S'_t[i]$ such that $i \in [t]$ and $S_t[i] \neq S'_t[i]$,
- (ii) for each $S_t[i_1], S_t[i_2], S'_t[i_1]$, and $S'_t[i_2]$ 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 $i_2 - i_1 + 1 \leq \omega$,

where ω , a positive integer, denotes the length of a sequence that can be protected at the same time.

Definition 5 (ω -event privacy [17]). A mechanism $\mathcal{A}$ satisfies ω -event differential privacy, if, for all $S \subseteq \mathcal{O}$, all S_t, S'_t at all t , it holds that

$$\log \frac{\Pr[\mathcal{A}(S_t) \in \mathcal{O}]}{\Pr[\mathcal{A}(S'_t) \in \mathcal{O}]} \leq \epsilon, \quad (3)$$

where $\mathcal{O}$ is the set of all possible outputs of $\mathcal{A}$. A mechanism satisfying ω -event privacy will protect the sensitive information that may be disclosed from a sequence of length ω .

According to the above definitions, we refer to [17] to conclude **Theorem 6**. The theorem enables a ω -event private scheme to view ϵ as the total available privacy budget in any sliding window of size ω and appropriately allocate portions of it across the time stamps.

Theorem 6. Assume that $\mathcal{A}$ is a mechanism with input stream prefix $S_t[i] = D_i \in \mathcal{D}$ and output $\mathbf{s} = (s_1, \dots, s_t) \in \mathcal{S}$. Supposing $\mathcal{A}$ can be decomposed into $\mathcal{A}_i(D_i) = s_i$, $i \in [1, t]$, each $\mathcal{A}_i$ generates independent randomness and achieves ω -differential privacy. Then, $\mathcal{A}$ satisfies ω -event privacy if

$$\forall i \in [t], \quad \sum_{k=i-\omega+1}^i \epsilon_i \leq \epsilon. \quad (4)$$

Based on this fundamental theorem, we will explore a novel adaptive ω -event differential privacy mechanism in our work. The proposed mechanism is designed for real-time privacy-preserving stream data aggregation under fog computing architecture.

2.5. Motivation and System Framework. Our motivation is to design a real-time stream data aggregation framework that can protect user privacy in any ω time stamp, allow batch queries, and obtain high-accuracy results. In order to achieve the motivation, we divide our work into two main tasks.

(i) **Protect privacy in any window of ω time stamp.**

Servers may query aggregated data within ω time only one round of communication. Therefore, the proposed framework must protect privacy of data generated in ω time stamp. Besides the size of window ω should be adaptively adjusted according to the state of data changes.

Input: The raw data database X_i at the i th time stamp
Output: The aggregation secure statistics R_i
 (1) Find out the optimal number of sampling points N
 (2) Find out sets of sampling sensors at the current time stamp
 (3) Obtain the grouping strategy G_{k_i} via *smart grouping*
 (4) Allocate the budget for all sampling sensors
 (5) Add Laplacian noise to group G_{k_i} with allocated budget at *the perturbation mechanism*
 (6) Report the aggregated secure statistics R_i that is filtered by Kalman filtering.
 (7) Update the sampling interval by *adaptive sampling*

ALGORITHM 1: A real-time stream data aggregation framework with adaptive ω -event differential privacy (Re-ADP).

- (ii) **Improve the accuracy of aggregated data.** Because of the Laplacian differential privacy, the proposed framework needs to add random noise to data to guarantee privacy protection. Thus, the framework must reduce extra errors of aggregated data as much as possible on premise of privacy preserving.

In this article, we intend to design an adaptive ω -event based differential privacy-preserving strategy. This strategy in Figure 2 is composed of adaptive ω -event privacy analysis, smart grouping-based perturbation, and the filtering mechanism. Here, we outline the complete process of the proposed Re-ADP strategy, shown in **Algorithm 1**. The first component, illustrated in Section 3, is achieved based on the adaptive sampling and QoP measurement. The second one is presented in Section 4, which is designed based on K-means smart grouping and the corresponding perturbation mechanism. And we exploit the similar filtering mechanism in [18] to reduce errors of aggregated data so as to improve data availability.

3. QoP-Based Adaptive ω -Event Privacy Design

For privacy protection on infinite stream data aggregating, ω -event privacy is a convincing model. The objective is to make a trade-off between utility and privacy to protect all data sequences that occur within all windows of ω time stamp. However, it is not applicable to many realistic scenarios due to the fixed size of the sliding window. The key issue of the unrealistic assumption is that most real-time aggregate data streams collected from sensors are significantly different in various time periods. For example, within successive time stamps, it can be seen that traffic data varies sharply in the daytime but is relatively stable at night. Thus, we introduce a new QoP-based adaptive ω -event privacy mechanism in this section to dynamically adjust the window size ω within different time stamps. The following three subsections describe the key parts to achieve this mechanism, including the QoP definition, the adaptive sampling design, and the adaptive ω -event privacy design.

3.1. Quality of Privacy. Considering the window size ω and errors of aggregated statistics, QoP is proposed to measure the corresponding privacy quality. Assume $\mathbf{x} = \{x_1, \dots, x_k\}$ and $\mathbf{r} = \{r_1, \dots, r_k\}$ represent the raw time series in a window

and the sanitized time series, respectively. Then, we exploit mean absolute error (MAE) to measure difference between these two time series.

$$MAE(x, r) = \frac{1}{k} \sum_{i=1}^k |r_i - x_i|. \quad (5)$$

Next, we employ a sampling mechanism in the proposed Re-ADP. It may perturb statistics at selected time stamp and approximate the nonsampled statistics with perturbed sampled statistics. Thus, (5) can be rewritten as follows:

$$MAE(x, r) = \frac{1}{k} \sum_i \sum_{j=i}^{i-1} |r_i - x_j|. \quad (6)$$

As a result, QoP in a window is defined as

$$QoP(x, r) = \sigma\left(\beta \frac{\omega}{MAE(x, r)}\right), \quad (7)$$

where ω is a window size and β is the weight between ω and $MAE(x, r)$. Here, β is set to 0.002 in our experiments. In addition, $\sigma(\cdot)$ is a logistic sigmoid function that is equal to

$$\sigma(x) = \frac{1}{1 - e^{-x}}. \quad (8)$$

The reason that we employ the logistic sigmoid function for normalization is that we do not need to know the general characteristics of the data. Intuitively, as sensor data generated in contiguous time stamps is not independent, there is close correlation among these data when data changes slowly. Meanwhile, with the possibility that sensitive information may be disclosed, the windows size ω should be increased when data changes slowly.

3.2. The Adaptive Sampling Design. In general, a report of noisy data denotes the expenditure of fixed budget ϵ . When protecting all time stamps, the budget allocated to each time stamp will be small if the window size ω is large. In this case, the report will show gigantic errors. This problem can be addressed by using a sampling mechanism (this mechanism can perturb sampled statistics while skipping nonsampled statistics). In this case, we can employ skipping some data points to save budget for future perturbation as

FIGURE 2: A high-level overview of Re-ADP.

well as improve communication efficiency. Without exploiting the model controller, our mechanism with a proportional-integral-derivative (PID) controller has the advantages of strong robustness and low complexity. Therefore, we exploit the PID controller to change the sampling rate based on dynamic historical data. Firstly, we define the feedback errors of sensor j .

$$E_{k_n}^j = |r_{k_n}^j - r_{k_{n-1}}^j|, \quad (9)$$

where k_n and k_{n-1} indicate the current sampling data point and the last sampling data point, respectively. It shows that data changes rapidly when the error $E_{k_n}^j$ increases. Then, the full PID error δ^j of sensor j is defined as follows:

$$\delta^j = C_p E_{k_n}^j + C_i \frac{\sum_{o=n-\pi-1}^n E_{k_o}^j}{\pi} + C_d \frac{E_{k_n}^j}{k_n - k_{n-1}}, \quad (10)$$

where C_p , C_d , and C_i denote the proportional gain, the integral gain, and the derivative gain, respectively.

Intuitively, the sampling interval needs to be small with rapid data change. Thus, a new sampling interval T is calculated by the following methods.

$$T = \max \left\{ 1, T_l + \theta_1 \left(1 - \left(\frac{\delta^j}{\theta_2} \right)^2 \right) \right\}. \quad (11)$$

In (11), T and T_l denote the current sampling interval and the previous one of sensor j . θ_1 is used to regulate the sampling interval, and θ_2 is used to control the sensitivity of the PID error.

3.3. The Adaptive ω -Event Privacy Algorithm. On the basis of the two sections above, the adaptive ω -event privacy algorithm is proposed in **Algorithm 2**. Note that pseudocodes from line 1 to line 6 are experiment offline over the training set.

We assume that the starting and ending points of the window are both sampling points and there are n sampling points in the current window. As a result, the window size $\omega = \sum_{k=1}^{n-1} T_k$. According to (6) and (7), the QoP in a window can be calculated as follows:

$$QoP(x, r) = \sigma \left(\theta \frac{\sum_{k=1}^{n-1} T_k}{(1/k) \sum_{i=1}^{i-1} |r_i - x_j|} \right). \quad (12)$$

After obtaining N over training test, the adaptive ω -event privacy mechanism is described from line 7 to line 10. In particular, we can adjust the new window size by moving the start point of the window forward or backward Δ time stamps.

4. Smart Grouping-Based Perturbation

A naive method to achieve differential privacy is to inject the Laplacian noise to statistics. Nonetheless, it is likely to introduce more perturbation errors, especially in statistics with small values. Therefore, we present a smart grouping-based perturbation to aggregate sensors with small statistics together in a dynamic way with the change of statistics.

The Smart Grouping Algorithm is presented in **Algorithm 3**. It mainly is divided into three steps. First, it screens out the sensors that needs to be grouped according to the predicted statistics (denoted by p_t^j) by exploiting the LSTM model. Then, it groups sensors that need to be grouped using the K-means algorithm. Finally, aggregated data will be perturbed based on the grouping result. We will elaborate on each step in detail in the following subsections.

4.1. Statistics Prediction with LSTM. To protect privacy of raw data, we use the predicted data instead of real values in the smart grouping-based perturbation algorithm. As mentioned above, whether a sensor needs to be grouped depends on the prediction data of the sensor. In addition, which group each sensor is assigned to also depends on the predicted value.

Input: The current window size ω_l ; The first and last sampling interval T_0, T_l at current window
Output: The new window size ω

- (1) Initialize the optimal number of sampling points: $N = 0$
- (2) Initialize the maximal QoP: $QoP_{max} = 0$
- (3) **for** each $n \in \{2, 3, \dots, n_{max}\}$ **do**
- (4) Calculate the current QoP: QoP'
- (5) **if** $QoP' > QoP_{max}$ **then**
- (6) $QoP_{max} \leftarrow QoP', N \leftarrow n$
- (7) Count the number of sampling: n'
- (8) **if** $n' \leq N$ and i is sampling point **then**
- (9) $\Delta \leftarrow T_l - T_0$;
- (10) Update the new window size: $\omega \leftarrow \omega_l + \Delta$;
- (11) **return** The new window size ω

ALGORITHM 2: The adaptive ω -event privacy.

Input: S_{t_i} : the sensors that need to be sampled at t_i
Output: The noised data of sensors in S_{t_i}

- (1) **for** each sensors $i \in S_{t_i}$ **do**
- (2) Predict $p_{t_i}^i$ by using the LSTM model
- (3) **if** $p_{t_i}^i > \tau$ **then**
- (4) Let the sensor i itself as a group
- (5) Add the group to G_{t_i}
- (6) **else**
- (7) Add the sensor into Φ
- (8) Employ K-means algorithm to cluster sensors in Φ according to $p_{t_i}^i$.
- (9) Add each group into G_{t_i} based on the cluster results.
- (10) Introduce Laplacian noise into each group G_{t_i} .
- (11) Allocate the group perturbed statistic to each sensor according to $p_{t_i}^i$.
- (12) **return** The noised data of sensors in S_{t_i}

ALGORITHM 3: Smart grouping-based perturbation.

This means that the accuracy of the predicted value is critical to the accuracy of the final aggregated data. Thus, a good model must be formulated, which can describe characteristics of data change well and predict data accurately.

To achieve accurate prediction, we introduce the LSTM model. In general, a LSTM network [19] has been gradually applied to the time-series analysis [20–22] by profiting from some advantages. In particular, it is a special type of recurrent neural network (RNN), which skillfully solved the problem of gradient vanishing of RNN. A common LSTM unit is composed of a memory cell, an undate gate, an output gate, and a forget gate. The memory cell stores a value (or state) for either long or short terms. It has the ability to remove or increase information to cell state through the well-designed three gates that can transfer information. As a result, we adopt the LSTM network to formulate our model to characterize the nonlinear characteristics of data in our algorithm.

Considering the effectiveness of our Smart Grouping Algorithm, our LSTM network only consists of three layers (shown in Figure 3), i.e., the input layer, the hidden layer, and the output layer. The input layer has k neurons, where the value of k is determined by the number of previously aggregated data to be used for prediction. The output is just

one neuron because we just need to predict the value of next time stamp. The hidden layer consists of several LSTM units. W_1 is a weight matrix between the input layer and the hidden layer, while W_2 is that between the output layer and the hidden layer. In addition, each context unit corresponds to a neuron in the hidden layer, which is used to record the output of the hidden layer in one recurrence.

As shown in Figure 3, the historical aggregated data is used as the training data to input to the LSTM model so as to predict the value for each sensor at current time. For example, suppose we need to predict the value generated from sensor j at time t_i (e.g., $p_{t_i}^j$). The previously aggregated data used for prediction is $(r_{t_i-k}^j, r_{t_i-k-1}^j, \dots, r_{t_i-1}^j)$. We first calculate the output of a hidden layer unit (i.e., $a_{t_i}^j$). Figure 4 shows the detailed structure of a LSTM unit, and $a_{t_i}^j$ is calculated as follows.

First, the LSTM unit determines what information should be forgotten from the cell state by using

$$F_f = \sigma(W_f \cdot [a_{t_i-1}^j, m_{t_i}^j] + b_f) \quad (13)$$

where $\sigma(\cdot)$ is the logistic sigmoid function, W_f is weight matrix of the forget gate, and b_f is the bias vector of forget gate

FIGURE 3: The architecture of the LSTM network.

FIGURE 4: The structure of the LSTM unit.

layer. $a_{t_{i-1}}^j$ is the output of the hidden layer at time t_{i-1} , while $m_{t_i}^j$ is the input of the hidden layer at current time, which is computed as follows.

$$m_{t_i}^j = W_1 \cdot r_{t_{i-1}}^j \quad (14)$$

Next, LSTM employs the following equations to decide what new information needs to be stored in the cell state by using the update gate layer.

$$F_u = \sigma(W_u \cdot [a_{t_{i-1}}^j, m_{t_i}^j] + b_u) \quad (15)$$

$$\bar{C}_{t_i} = \tanh(W_c \cdot [a_{t_{i-1}}^j, m_{t_i}^j] + b_c) \quad (16)$$

where F_u indicates which value will be updated and $\bar{C}_{t_i}$ represents a vector of new candidate values. W_u and W_c are weight matrices of the input gate layer. And $\tanh(\cdot)$ is defined as follows:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (17)$$

Then, the cell state C_{t_i} is updated based on (18) in current time t_i .

$$C_{t_i} = F_u * \bar{C}_{t_i} + F_f * C_{t_{i-1}} \quad (18)$$

Here, the output at current time of the hidden layer is controlled by the update gate F_u and the forget gate F_f .

Finally, based on the latest cell state C_{t_i} , the output of the hidden layer at current time, $a_{t_i}^j$, can be calculated as follows:

$$F_o = \sigma(W_o \cdot [a_{t_{i-1}}^j, m_{t_i}^j] + b_o) \quad (19)$$

$$a_{t_i}^j = F_o * \tanh(C_{t_i}) \quad (20)$$

where F_o is the output gate that determines which part of C_{t_i} should be output.

According to (13), (15), and (19), we can be aware that the LSTM unit has the ability of determining which information is forgotten, updating and outputting intelligently. This ability enables us to predict time series of our network more accurately.

Final prediction data of sensor j at time t_i , e.g., $p_{t_i}^j$, is calculated as follows:

$$p_{t_i}^j = g(W_2 \cdot a_{t_i}^j) \quad (21)$$

where g is the activation function of the output layer.

Training of the LSTM network: in order to achieve real-time prediction, we should train the network related parameters offline in advance. In addition, we must employ the true statistics of the training set for the sake of the accuracy of the training model. Therefore, for sensor j at time t_i , the input is $(x_{t_{i-c}}^j, x_{t_{i-c-1}}^j, \dots, x_{t_{i-1}}^j)$, and the expected output is the true statistic $x_{t_i}^j$. And, based on the predicted statistic $p_{t_i}^j$, the loss function of our network is defined as below.

$$\mathcal{L}_{t_i}^j = \frac{1}{2} (x_{t_i}^j - p_{t_i}^j)^2 \quad (22)$$

Using the backpropagation algorithm [23], the training error is propagated to the neurons in the LSTM network. Then, we further calculate training errors caused by each neuron and adjust the corresponding weights to reduce the errors. Details of the training process can be established in [23]. Finally, given the historical aggregated data, the trained LSTM model can predict sensors data in real time.

4.2. K-Means Based Smart Grouping Algorithm. In this subsection, we present a Smart Grouping Algorithm based on the K-means method. The algorithm can smartly aggregate small statistics obtained from sensors in the noise scenarios. First of all, we allocate the budget to each sampling point ϵ_i and then generate an antinoise threshold τ dynamically. Clearly, we can utilize an inverse proportion to characterize the relationship between τ and the corresponding allocated budget. Then, we can obtain the predicted data of each sensor $p_{t_i}^j$ for each time t_i by using the trained LSTM model. According to the above processing, we can exploit K-means algorithm [24] to achieve

FIGURE 5: An example of smart grouping.

sensors data aggregation in the premise that $p_{t_i}^i$ is smaller than the antinoise threshold τ .

Compared with other algorithms, k-means algorithm is fast and efficient, which is suitable for large data scenarios. Thus, it accords with the data size and real-time requirement of our algorithm. Next, we will introduce how the K-means algorithm works in our scenario. Note that the input is the predicted data of sensors at each time stamp t_i , which need to be grouped as $\{\dots, p_{t_i}^i, p_{t_i}^j, \dots\}$. In particular, we first randomly initialize the K cluster centers and then divide it into clusters where each cluster is closest to its nearest cluster centers for each sensor i . Here, we intend to employ the Euclidean distance to calculate the distance from the current point to the center point. Next, we update the cluster center according to the new clusters obtained from the previous step. The method to update cluster centers is to calculate the mean of all points in the cluster. And the convergent condition is that the minimum squared error of every point to the center point is less than a threshold value or the preset maximum number of iterations. Finally, the algorithm repeats the above two steps until convergence.

Figure 5 is an example to explain the whole process of the Smart Grouping Algorithm. Assuming there are four sensors that need to sampled at time stamp t_i , we define that the predicted statistics are 12, 11, 23, and 80, respectively. The antinoise threshold τ is 50. For S_4 , S_4 is an independent group because $80 > 50$, which is added to G_{t_i} (the group strategy of the current time stamp). For $S_1, S_2,$ and S_3 , we input them to the K-means algorithm. Note that S_1 and S_2 are clustered into a group while S_3 becomes a single group. Thus, the final group strategy is $G_{t_i} = \{\{S_1, S_2\}, S_3, S_4\}$.

4.3. Smart Grouping-Based Perturbation. To achieve additional noise loading, we intend to exploit the Laplacian mechanism to directly inject noises into aggregated statistics [15] based on results from adaptive sampling. The aggregated statistics do not include the nonsampled statistics that can be

approximated by the last aggregated statistics. In this article, we present a scheme of smart grouping-based perturbation. This scheme is composed of a perturbation component and an allocation component. Considering the utilization of the grouping algorithm, we apply the Laplacian mechanism in each group rather than in each sensor in our scheme design.

We assume that a group g has κ sensors and $f(g)$ represents a function to aggregate the number of data contributors in g . Intuitively, because all contributors can only appear in the collection range of a sensor at one time stamp, the sensitivity of the function f is equal to 1; i.e., $\Delta(f) = 1$. Then, the Laplacian mechanism can be employed in group g as follows:

$$\begin{aligned} \mathcal{A}(g) &= f(g) + \text{Lap}(\lambda(g)) \\ &= \sum_{j=1}^{\kappa} g[j] + \text{Lap}\left(\frac{\Delta(f)}{\epsilon_i}\right), \end{aligned} \quad (23)$$

where $g[j]$ is the j th sensor of the group g and $\lambda(g)$ denotes the scale of Laplacian noises injected into $f(g)$. In order to avoid exceeding the total budget, our scheme considers the smallest budget of a sensor in a group as the budget of the whole group. In this case, the proposed RescueDP strategy does not make full use of the total budget. In addition, we also fix the sampling points in our scheme and allocate the total budget to each sampling point uniformly. Therefore, $\epsilon_i = \epsilon/N$, which leads to making full use of the total budget as well as ensuring not exceeding the total budget.

Next, considering the predicted statistics in each sensor, we allocate the perturbed statistic. The allocation method can avoid errors resulting from the average operation in the RescueDP strategy. Our allocation method is shown as follows:

$$\mathcal{A}(g[j]) = \alpha_j \mathcal{M}(g), \quad \forall j = 1, \dots, \kappa, \quad (24)$$

where the weight of a sensor, α_j , can be calculated by the predicted statistics of a sensor; i.e.,

$$\alpha_j = \frac{p_{t_i}^j}{\sum_{j=1}^{\kappa} p_{t_i}^j}. \quad (25)$$

According to the smart grouping-based perturbation scheme, the perturbed statistics of a sensor are more accurate.

5. Performance Discussion

In this section, we first analyze the privacy of our proposed Re-ADP framework in theory and then provide several numerical simulations to study the performance of our framework in terms of MAE and QoP.

5.1. Privacy Analysis

Theorem 7. The proposed Re-ADP framework satisfies ϵ -differential privacy.

Proof. In the Re-ADP framework, perturbation is the only possible mechanism to disclose private information because it is the only one to access raw data. As a result, if the perturbation mechanism can be proved to satisfy ϵ -differential privacy, the Re-ADP framework can meet the requirement of ϵ -differential privacy subsequently.

On the basis of the smart grouping strategy G_{t_k} at time stamp t_k , each group includes several sensors. We assume that g_m with κ_m sensors is an arbitrary group of G_{t_k} . According to (12), the Laplacian mechanism on group g_m is as follows:

$$\begin{aligned} \mathcal{A}(g_m) &= f(g_m) + \text{Lap}(\lambda(g_m)) \\ &= \sum_{j=1}^{\kappa_m} g_m[j] + \text{Lap}\left(\frac{\Delta(f)}{\epsilon_{t_k}}\right), \end{aligned} \quad (26)$$

where $g_m[j]$ is the j th sensor of g_m and $\Delta(f) = 1$.

Based on **Definition 1**, $\mathcal{A}(g_m)$ satisfies ϵ_{t_k} -differential privacy. According to Axiom 2.1.1 in [25], postprocessing sanitized data will not reveal privacy as long as sensitive information is not available directly in the postprocessing algorithm. As a result, $\mathcal{A}(g_m[j])$, $\forall j = 1, \dots, \kappa_i$, also satisfies ϵ_{t_k} -differential privacy. Assume that ϵ'_{t_k} and ϵ_{t_k} represent the budget consumed and the budget allocated for a sensor at timestamp t_k , respectively. If all allocated budget is employed for perturbation in our algorithm, then $\epsilon'_{t_k} = \epsilon_{t_k}$ holds.

Based on **Theorem 6**, the perturbation mechanism of a sensor satisfies ϵ -differential privacy for every t_i and $t_i \in [t]$, if it holds that

$$\sum_{t_k=t_i-\omega+1}^{t_i} \epsilon'_{t_k} \leq \epsilon. \quad (27)$$

The above formula always holds for any sliding window ω timestamp for the reason that $\epsilon'_{t_i} = \epsilon_{t_i}$ holds in our budget allocation algorithm. Thus, the perturbation mechanism on each group can satisfy ϵ -differential privacy. In other words, the Re-ADP algorithm also satisfies ϵ -differential privacy. And this completes the proof of **Theorem 7**. $\square$

5.2. Numerical Simulation. We compare the performance of the proposed Re-ADP strategy with MLDP in [14] and the RescueDP strategy in [26] over two real datasets. The MLDP is a privacy-preserving data aggregation scheme under fog computing based on machine learning, while the RescueDP is the latest strategy that provides ϵ -event privacy for real-time aggregate data publishing. In the simulation, we employ MAE and QoP as metrics to study the performance of the three schemes. The specific expressions of these metrics are given by (6) and (12). Our experiment is conducted in Python environment in Windows 10 operating system. Each experiment is run 100 times and points in the results are the average values of 100 times of each experiment.

The real-world test datasets to discuss the performance in our experiment are Bike data [27] and Station data [28]. The dataset of Bike provides an accurate data containing the bike share trip in Washington DC for one year from January 1, 2016, to December 31, 2016. It contains a total

of 3,333,791 bike share trajectories. Each of them consists of the bike number, the end station and time, and the start station and time. We transform it into a dataset that consists of 368 sensors to count the number of bikes at each bike parking spot in real time. The first three-quarters of the data is split as the training set and the fourth-quarter of the data is used as test set. The dataset of Station consists of the number of passengers of 2116 stations between January 1, 2016, and December 30, 2016. It contains 9,917,584 records and each record reports the number of passengers of a station. Because many stations have very little throughput to report. And the division of the test set and the training set is the same as the Bike dataset.

In our experiment, the parameters of the PID control are set as follows: $C_p = 0.9$, $C_d = 0.1$, $C_i = 0$, $\pi = 10$, $\theta_1 = 15$, and $\theta_2 = 10$ for the adaptive sampling mechanism. In **Algorithm 2**, n_{max} is set to be 100. In addition, we obtain $N = 10$ for the Bike dataset and $N = 8$ for the Station dataset by constantly iterating over training set. In K-means based Smart Grouping Algorithm, we set $K = 5$ for Bike dataset and $K = 8$ for the Station dataset, and the result is also the best performance on the training set. The parameters of the LSTM network are set as follows: the previous $k = 50$ history data is used for the input of the LSTM network. So the numbers of input layers' neurons is 50, and 100 is the number of hidden layers' neurons. Besides, we train by iterating 1,000 times. Note that a training process that takes about two minutes is time-consuming. However, we only conduct this process offline, which does not affect the real-time nature of the algorithm.

5.2.1. Utility versus Privacy. Figure 6 provides the trade-off analysis between utility and privacy. It is clear that when ϵ increases, the MAE of three schemes decreases gradually. The reason is that the larger ϵ represents the smaller noise that needs to be injected. Moreover, for two real-world test datasets, the Re-ADP scheme outperforms the other two schemes greatly, especially in a small privacy budget. Also, the QoP of Re-ADP is obviously superior to the other two schemes in a sufficient privacy budget.

The superior performance of the Re-ADP scheme results from the following three aspects. First, due to the design of the optimal number of sampling points and the corresponding privacy budget allocation mechanism, the privacy budget is fully used for private perturbation. Second, the adaptive ω -event privacy mechanism in the Re-ADP scheme satisfies the privacy window adaptively, which improves the practicability of the scheme. Finally, LSTM-based prediction can provide a high-accuracy prediction result for the smart grouping mechanism.

5.2.2. Effect of Adaptive ω -Event Privacy Mechanism. In order to highlight the advantages of adaptive ω -event privacy mechanism, we compare our Re-ADP scheme with a variant version, Re-ADP(f), which only adapts fixed ω -event privacy mechanism. Figure 7 demonstrates the comparison results in terms of MAE and QoP. It can be clearly seen that the adaptive ω mechanism can increase QoP while decreasing MAE significantly in both real-world datasets. Therefore, we can draw

FIGURE 6: Utility comparison when ϵ changes.

the conclusion that the adaptive ω -event privacy mechanism advances the quality of reported data considerably.

5.2.3. Effect of Smart Grouping Mechanism. In this part, we investigate the performance of our smart grouping mechanism. As shown in Figure 8, both MAE and QoP of the smart grouping mechanism exceed the Re-ADP without the smart grouping. The excellent performance of smart grouping chiefly benefits from the K-means-based grouping algorithm and the application of the deep learning algorithm.

6. Related Work

Many methods have been proposed to ensure the privacy of aggregated data generated from IoT devices [29–33]. Wu et al. [34] proposed a Dynamic Trust Relationships Aware Data Privacy Protection (DTRPP) mechanism for Mobile Crowd Sensing (MCS), which evaluates the trust value of a public key ingeniously. Zhang et al. [35] designed a priority-based health data aggregation scheme (PHDA) in cloud-assisted wireless body area networks. In the scheme, a credible relay

FIGURE 7: The performance of the adaptive ω mechanism ($\epsilon = 1$). FIGURE 8: The performance of the smart grouping mechanism ($\epsilon = 1$).

node can be selected according to the social relationship between nodes to help aggregation data and then forward it to cloud servers. PHAD also provides a lightweight privacy-preserving aggregation scheme, which can not only resist the forgery attack but reduce communication overhead. Li et al. [36] presented a privacy-aware data aggregation protocol for mobile sensing, which can aggregate time-series data to prevent untrustworthy aggregators from disclosing privacy. Using an additive homomorphic encryption and a novel key management scheme, the aggregator can only obtain the sum

of all users data. Still, both schemes cannot cope with complex attacks that can also mine some privacy from the raw sum data.

In addition, all existing methods to achieve privacy preserving are based on encryption technologies. These complicated encryption technologies usually introduce high computation overhead, which may not be suitable for energy-constrained sensor networks. Some researchers suggest exploiting differential privacy, a convincing model for providing privacy, to protect aggregated data generated by IoT

devices. Han et al. in [37] proposed a scheme to provide privacy preserving for health data aggregation. It employs a differential privacy model to resist differential attacks that most existing data aggregation schemes have suffered from. Yang et al. in [14] also proposed a differential privacy model based on machine learning algorithms. The model can reduce communication overhead as well as protect the privacy of sensitive data rigorously for the fog computing architecture. Also in fog computing, Wang et al. [38] put forward a privacy-preserving content-based publish-subscribe scheme with differential privacy in a publish-subscribe system, which can protect against collusion attacks.

Although these works apply differential privacy to protect privacy of aggregated data, there is a serious deficiency in a real scenario. They may greatly reduce the availability of aggregated data streams. Thus, some studies are committed to solving this challenge. Cao et al. in [39] studied a protection method for sensitive streams within a window instead of the whole infinite stream. Considering window-based applications, they explored a stream-based management system to cope with numerous aggregate queries simultaneously. In [18], Fan and Xiong intended to hide all events of users and designed a user-level privacy strategy for a finite stream. For the received perturbing data, they employed the Kalman filter [40] in their strategy to improve accuracy of differentially private data release. Considering multiple events occurring at continuous time segments, Kellaris et al. presented a ω -event ϵ -differential privacy model in [17]. This model combined the advantages of event-level privacy model and user-level privacy model skillfully. In the model, they employed a sliding window to capture a wide range of ω -event privacy and designed a scheme to distribute and absorb the privacy budget on the assumption that statistics do not change significantly. On this basis, Wang and Zhang further designed an online aggregate monitoring scheme for infinite streams in [26]. Their scheme integrated adaptive sampling, a budget mechanism, and dynamic grouping and perturbation to provide privacy preserving of statistics.

Despite the fact that ongoing studies of differential privacy on streams data aggregation have played a vital role, there still exist challenges to be dealt with. We point out that the fixed sliding windows employed in most existing frameworks may not be practical. Moreover, existing metrics are only suitable for static data rather than streaming media. Motivated by these challenges, in this paper, we present a real-time privacy-preserving streams data aggregation framework based on adaptive ω -event differential privacy for the fog computing architecture.

7. Conclusion

Considering privacy disclosure of aggregated data in fog computing, we present a real-time stream data aggregation framework with adaptive ω -event differential privacy (Re-ADP) in this paper. For the four layers of our system model, this framework is composed of three components, i.e., adaptive ω -event privacy analysis, smart grouping-based perturbation, and the filtering mechanism. In particular, we can employ the first component to protect privacy of infinite

stream over any successive ω time stamps. Then the second component is to achieve smart grouping based on K-means and inject additional noise into aggregated data, and we exploit an existing filter to improve data availability in the third component. Finally, we provide a theory to prove that the proposed Re-ADP framework satisfies differential privacy in theory. Extensive experiments over real-world datasets show that the Re-ADP scheme outperforms existing methods and improves the utility of real-time data publishing with strong privacy preserving.

Data Availability

The datasets used to support the findings of this study have been openly accessed. The Bike dataset can be found at <https://www.capitalbikeshare.com/system-data>. And the Station dataset can be accessed at <https://www.kaggle.com/saulfuh/bart-ridership/data>. Also, the authors have cited these datasets in the References.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61471028 and 61571010) and the Fundamental Research Funds for the Central Universities (Grant nos. 2017JBM004 and 2016JBZ003).

References

- [1] L. Zhang, Z. Cai, and X. Wang, "FakeMask: A Novel Privacy Preserving Approach for Smartphones," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.
- [2] C. Hu, W. Li, X. Cheng, J. Yu, S. Wang, and R. Bie, "A Secure and Verifiable Access Control Scheme for Big Data Storage in Clouds," *IEEE Transactions on Big Data*, pp. 1–1, 2017.
- [3] W. Zhang, Z. Zhang, and H. Chao, "Cooperative Fog Computing for Dealing with Big Data in the Internet of Vehicles: Architecture and Hierarchical Resource Management," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 60–67, 2017.
- [4] Y. Huo, C. Hu, X. Qi, and T. Jing, "LoDPD: A Location Difference-Based Proximity Detection Protocol for Fog Computing," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1117–1124, 2017.
- [5] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing Fog Computing for Internet of Things Applications: Challenges and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2018.
- [6] F. Tzer, "Privacy issues in vehicular ad hoc networks," in *Proceedings of the International Conference on Privacy Enhancing Technologies*, pp. 197–209, 2005.
- [7] Y. Huo, W. Dong, J. Qian, and T. Jing, "Coalition game-based secure and effective clustering communication in vehicular cyber-physical system (VCPS)," *Sensors*, vol. 17, no. 3, article no. 475, pp. 1–23, 2017.

- [8] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, May 2017.
- [9] C. Hu, H. Li, Y. Huo, T. Xiang, and X. Liao, "Secure and Efficient Data Communication Protocol for Wireless Body Area Networks," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 2, pp. 94–107, 2016.
- [10] Y. Lu, X. Wang, C. Hu, H. Li, and Y. Huo, "A traceable threshold attribute-based signcryption for mHealthcare social network," *International Journal of Sensor Networks*, vol. 26, no. 1, pp. 43–53, 2018.
- [11] M. Li, W. J. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *IEEE Wireless Communications Magazine*, vol. 17, no. 1, pp. 51–58, 2010.
- [12] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: Privacy-Preserving Multisubset Data Aggregation in Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 462–471, 2018.
- [13] C. Dwork, "Differential privacy," in *Proceedings of the ICALP*, pp. 1–12, 2006.
- [14] M. Yang, T. Zhu, B. Liu, Y. Xiang, and W. Zhou, "Machine Learning Differential Privacy With Multifunctional Aggregation in a Fog Computing Architecture," *IEEE Access*, vol. 6, pp. 17119–17129, 2018.
- [15] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography*, vol. 3876 of *Lecture Notes in Comput. Sci.*, pp. 265–284, Springer, Berlin, 2006.
- [16] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," *Communications of the ACM*, vol. 53, no. 9, pp. 89–97, 2010.
- [17] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," in *Proceedings of the VLDB Endowment*, vol. 7, pp. 1155–1166, 2014.
- [18] 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.
- [19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [20] S. Gulshad, D. Sigmund, and J.-H. Kim, "Learning to reproduce stochastic time series using stochastic LSTM," in *Proceedings of the 2017 International Joint Conference on Neural Networks, IJCNN 2017*, pp. 859–866, May 2017.
- [21] S. Kaushik, A. Choudhury, N. Dasgupta, S. Natarajan, L. A. Pickett, and V. Dutt, "Using LSTMs for Predicting Patient's Expenditure on Medications," in *Proceedings of the 2017 International Conference on Machine Learning and Data Science (MLDS)*, pp. 120–127, December 2017.
- [22] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.
- [23] F. J. Pineda, "Generalization of back-propagation to recurrent neural networks," *Physical Review Letters*, vol. 59, no. 19, pp. 2229–2232, 1987.
- [24] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, University of California Press, Berkeley, CA, USA, 1967.
- [25] D. Kifer and B. Lin, "Towards an axiomatization of statistical privacy and utility," in *Proceedings of the the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems*, pp. 147–158, ACM, Indianapolis, Indiana, USA, June 2010.
- [26] 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 *Proceedings of the IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, 2016.
- [27] "Capital bikeshare data," <https://www.capitalbikeshare.com/system-data>.
- [28] "Station data," <https://www.kaggle.com/saulfuh/bart-ridership/data>.
- [29] D. He, N. Kumar, S. Zeadally, A. Vinel, and L. T. Yang, "Efficient and Privacy-Preserving Data Aggregation Scheme for Smart Grid Against Internal Adversaries," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2411–2419, 2017.
- [30] Z. You, S. Chen, and Y. Wang, "An efficient traffic data aggregation scheme for WSN based intelligent transportation systems," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 6, pp. 1117–1129, 2015.
- [31] A. Ara, M. Al-Rodhaan, Y. Tian, and A. Al-Dhelaan, "A Secure Privacy-Preserving Data Aggregation Scheme Based on Bilinear ElGamal Cryptosystem for Remote Health Monitoring Systems," *IEEE Access*, vol. 5, pp. 12601–12617, 2017.
- [32] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.
- [33] X. Dong, J. Zhou, K. Alharbi, X. Lin, and Z. Cao, "An ElGamal-based efficient and privacy-preserving data aggregation scheme for smart grid," in *Proceedings of the 2014 IEEE Global Communications Conference, GLOBECOM 2014*, pp. 4720–4725, December 2014.
- [34] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic Trust Relationships Aware Data Privacy Protection in Mobile Crowd-Sensing," *IEEE Internet of Things Journal*, no. 99, pp. 1–1, 2017.
- [35] K. Zhang, X. Liang, M. Baura, R. Lu, and X. Shen, "PHDA: a priority based health data aggregation with privacy preservation for cloud assisted WBANs," *Information Sciences*, vol. 284, pp. 130–141, 2014.
- [36] Q. Li, G. Cao, and T. F. L. Porta, "Efficient and privacy-aware data aggregation in mobile sensing," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 2, pp. 115–129, 2014.
- [37] S. Han, S. Zhao, Q. Li, C.-H. Ju, and W. Zhou, "PPM-HDA: privacy-preserving and multifunctional health data aggregation with fault tolerance," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1940–1955, 2016.
- [38] Q. Wang, D. Chen, N. Zhang, Z. Ding, and Z. Qin, "PCP: A Privacy-Preserving Content-Based Publish-Subscribe Scheme with Differential Privacy in Fog Computing," *IEEE Access*, vol. 5, pp. 17962–17974, 2017.
- [39] J. Cao, Q. Xiao, G. Ghinita, N. Li, E. Bertino, and K. Tan, "Efficient and accurate strategies for differentially-private sliding window queries," in *Proceedings of the the 16th International Conference*, pp. 191–202, ACM, Genoa, Italy, March 2013.
- [40] A. Jain, E. Y. Chang, and Y.-F. Wang, "Adaptive stream resource management using kalman filters," in *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pp. 11–22, ACM, 2004.