IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 26, 2018, accepted March 5, 2018, date of publication March 20, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2817523

Machine Learning Differential Privacy With Multifunctional Aggregation in a Fog Computing Architecture

MENGMENG YANG[1], TIANQING ZHU[1,2], (Member, IEEE), BO LIU[3], YANG XIANG[4], (Senior Member, IEEE), AND WANLEI ZHOU[1], (Senior Member, IEEE)

[1]School of Information Technology, Deakin University, Melbourne, VIC 3125, Australia
[2]School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan 430023, China
[3]Digital Research and Innovation Capability Platform, Swinburne University of Technology, Melbourne, VIC 3122, Australia
[4]Department of Engineering, La Trobe University, Melbourne, VIC 3086, Australia

Corresponding author: Tianqing Zhu (t.zhu@deakin.edu.au)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502362 and in part by the Australia Research Council Linkage under Grant LP170100123.

ABSTRACT Data aggregation plays an important role in the Internet of Things, and its study and analysis has resulted in a range of innovative services and benefits for people. However, the privacy issues associated with raw sensory data raise significant concerns due to the sensitive nature of the user information it often contains. Thus, numerous schemes have been proposed over the last few decades to preserve the privacy of users' data. Most methods are based on encryption technology, which is computationally and communicationally expensive. In addition, most methods can only handle a single aggregation function. Therefore, in this paper, we propose a multifunctional data aggregation method with differential privacy. The method is based on machine learning and can support a wide range of statistical aggregation functions, including additive and non-additive aggregation. It operates within a fog computing architecture, which extends cloud computing to the edge of the network, alleviating much of the computational burden on the cloud server. And, by only reporting the results of the aggregation to the server, communication efficiency is improved. Extensive experimental results show that the proposed method not only answers flexible aggregation queries that meet diversified aggregation goals, but also produces aggregation results with high accuracy.

INDEX TERMS Data aggregation, differential privacy, fog computing.

I. INTRODUCTION

Data aggregation is considered to be an essential research topic in the Internet of Things (IoT). For example, energy companies collect and aggregate utility data from sensors installed at customer sites, which is used to improve the overall reliability and efficiency of their infrastructure [1]. Likewise, in traffic monitoring systems, traffic flow data is collected by road-side sensors and used to analyze the network to improve services for drivers [2]. In wireless body area networks (WBANs), health data is collected through mobile or wearable devices to monitor a user's health indicators, but aggregated data is needed for medical research [3].

Given the often sensitive nature of the data involved, privacy is an important issue in data aggregation. For instance, health data, such as blood pressure and temperature, can reveal a user's health status, and electricity usage patterns can be used to profile a customer's lifestyle and daily routines [4]. For this reason, many people choose not to participate in sensory systems without a strong guarantee of privacy.

Methods to preserve the privacy of aggregated data have been developed by several scholars [5]–[9]. However, most are based on encryption technology, such as homomorphic encryption. For example, Dong *et al.*'s [10] data aggregation method for smart grids is based on EIGamal-based homomorphic privacy preservation, while Abdallah and Shen 's scheme [11] introduces lightweight lattice-based homomorphic privacy preservation.

Despite these efforts, there are many problems with the existing methods.

- *Computation overhead.* Homomorphic encryption typically results in massive computational overheads [1], which increases the burden of processing and analysis

on cloud services. Additionally, these methods are not practical for sensors with limited energy.

- *Communication efficiency.* The communication overheads are high, especially when the system contains thousands of sensors with high reporting frequency, because each sensor needs to report its encrypted data to the cloud at the same time.
- *Single aggregation function calculation.* Most existing methods can only calculate a single aggregation function. In practice, the ideal aggregation scheme would allow flexible aggregation queries to meet diversified aggregation goals with only one round of communication.

To solve these problems, we propose a privacy-preserving data aggregation method based on machine learning within a fog computing architecture. Fog computing architectures distribute computation and data storage to the edge of the network, i.e., to devices that sit between the data source and the cloud server. This type of architecture reduces the amount of data transported to the cloud, improving efficiency and alleviating much of the burden on the server itself. Additionally, in our method, the aggregator resides at the center of the fog and only the aggregation results are reported to the cloud server, which significantly increases communication efficiency. Aggregation queries are answered by learning a model, which is trained to predict the query results through a process that satisfies differential privacy. Multiple aggregation functions can be calculated, including additive aggregation and non-additive aggregation. Finally, the method does not apply encryption technology, so the sensors only need to report raw data without the need for a complex cipher process.

In summary, this paper offers the following contributions.

- We propose a novel privacy-preserving data aggregation method under fog computing architecture, which reduces the communication overhead and releases the cloud burdens.
- The proposed privacy-preserving data aggregation method is based on machine learning. The trained learning model can be used to predict the aggregation query results and supports multiple aggregation functions, which allows the server provides various services.
- The proposed data aggregation method satisfies differential privacy, which provides rigorous privacy protection for sensory data. Efficiently defend the differential attack that appears in most aggregation functions.
- We theoretically analyse the privacy and utility of proposed methods and extensive experimental results show that the proposed method generates highly accurate aggregated results.

The rest of the paper is organized as follows. In Section II, we introduce the preliminaries. Section III proposes the research problem. We present our privacy preservation method and theoretically analyze its privacy and utility in Sections IV and V, respectively. Section VI details the results of the experiments. Section VII discusses the related work, and Section VIII concludes the paper.

II. PRELIMINARIES

A. NOTATIONS

Let $S_{fc} = \{s_1, s_2, \ldots, s_g\}$ be a group of sensors. These sensors report the sensory data to the fog nodes f_1 and f_2. The fog node trains a learning model M using the collected data and predicts the query results. Let $Q\{q_1, q_2, \ldots, q_t\}$ be a set of queries which generated by the fog center. Additional notations are shown in Table 1.

TABLE 1. Notations.

Notation	Description
q	query
Q	query set
s_i	sensor
f_i	fog node
S_Q	query sensitivity
S_{max}	biggest value of query sensitivity
QA	query results
$\hat{QA}$	perturbed query results
M	training model

B. DIFFERENTIAL PRIVACY

Differential privacy is a provable privacy notation, developed by Dong *et al.* [12] that has emerged as an essential standard for preserving privacy in a variety of areas.

Definition 1 (ϵ-Differential Privacy): A randomized algorithm $\mathcal{M}$ gives ϵ-differential privacy for any pair of *neighboring datasets* D and D^* where, for every set of outcomes Ω, $\mathcal{M}$ satisfies

$$Pr[\mathcal{M}(D) \in \Omega] \leq \exp(\epsilon) \cdot Pr[\mathcal{M}(D^*) \in \Omega]. \quad (1)$$

This definition ensures that the presence or absence of an individual will not significantly affect the output of the query.

Definition 2 (Global Sensitivity): For a query $Q : D \to \mathbb{R}$, the global sensitivity of Q is defined as follow:

$$GS = max_{D,D'} \|Q(D) - Q(D')\|_1 \quad (2)$$

*Definition 3 (*Laplace *Mechanism):* Given a function $f : D \to \mathbb{R}$ over a dataset D, Eq. 3 provides ϵ-differential privacy.

$$\widehat{f}(D) = f(D) + Laplace(\frac{s}{\epsilon}). \quad (3)$$

A *Laplace* mechanism is used to produce numeric output, and differential privacy is achieved by adding *Laplace* noise to the true answer.

III. PROBLEM STATEMENT

A. SYSTEM MODEL

As shown in Fig. 1, the system model is composed of four entities: sensors, fog nodes, the fog center, and a cloud server. A description of each entity follows.

- *Sensors:* The sensors, which might be embedded in smart devices, collect the data. To address privacy concerns, the original data is partitioned and separately reported to two fixed fog nodes.

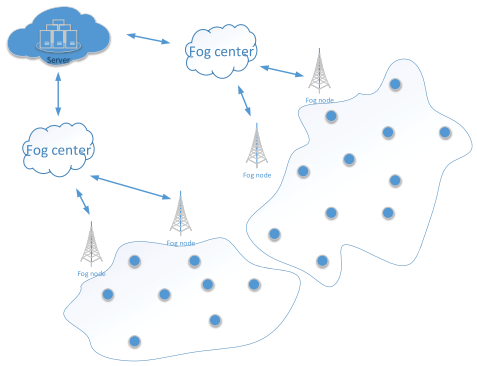


FIGURE 1. The private multifunctional aggregation system model.

- *Fog nodes:* The fog nodes are efficient devices for computing and storing data that extend the edge of the cloud service. These devices serve as storage to answer aggregation queries sent from the fog center.
- *Fog center:* The fog center is in charge of three important tasks. First, it transfers queries to the appropriate aggregation query set to be answered by the fog nodes. Second, it gathers the returned query results from the fog nodes. Third, it calculates the original query results and reports them to the cloud server.
- *Cloud server:* The cloud server is managed by the service provider and deployed as the aggregation application. This server is powerful and is used to process and analyze large amounts of aggregation data to provide information and assist with a wide range of services.

B. ADVERSARY MODEL

In this paper, we assume that the cloud server and the fog center are untrusted. Both will try to acquire the true values of the collected data, which is either sensitive or could be used to infer private information about the service users, or both. The fog nodes are semi-trusted, which means they are curious about the collected data but are not able to collude with each other.

C. DESIGN OBJECTIVES

Our objective is to design an efficient data aggregation method that preserves the privacy of the users' data and allows for multifunctional aggregation queries in an IoT setting. Within this problem, there are three primary objectives:

- *to ensure multifunctional aggregation is implemented correctly.* To suit practical requirements, the method must include flexible aggregation functions to meet diverse analysis requirements for a wide and diverse range of services. Therefore, a mechanism that can satisfy multifunctional aggregation requirements and flexibly answer a range of data aggregation queries is highly desirable.
- *to guarantee the privacy of the collected user data.* Adversarial models consider possible privacy threats to an individual's privacy and, given that the data collected often pertains to a user's health or behavioral habits, the aggregation scheme developed must satisfy each individual's privacy with a guarantee of ϵ-differential privacy.
- *to ensure the aggregation results are close to the results without privacy protection.* As the proposed system needs to satisfy ϵ - differential privacy, any noise added to the training set will reduce the accuracy of the aggregation results. (How accuracy is evaluated is defined in Definition 4.) Hence, the method must include a way to adjust the sensitivity and the amount of added noise to ensure the accuracy of the aggregation results are (α, β)-useful.

IV. PROPOSED SCHEME

In this paper, we propose a multifunctional aggregation framework based on machine learning. In general, the data collected from each region are used to train a learning model, which, in turn, is used to predict multiple query results. The predicted query results are then further processed to calculate the required aggregation function. This framework is able to deliver multifunctional aggregation in one round of communication without disclosing the sensory data to any party.

Fig. 2 illustrates the complete aggregation process. Within the framework, two fog nodes are in charge of collecting data from each region. Once a sensor collects some information, it randomly partitions the data into two parts and separately transmits one part to each of the two fog nodes. Because the fog nodes cannot collude, neither node can integrate or infer the true values of the sensor data. Each fog node receives data from many sensors, and once assembled, the fog node trains a learning model using the data it has received. Once trained, the learning model is able to predict the summation of any sensor's value. To defend against differential attacks, the training dataset is generated using a process that satisfies differential privacy. The fog center fetches the query results from the two fog nodes, calculates the aggregation results, and returns those results to the cloud server.

A. DATA AGGREGATION PROTOCOL

This section presents the proposed privacy-preserving data aggregation method. The method includes three stages: processing the query, generating the sensor report, and predicting the query results while preserving privacy.

1) QUERY PROCESSING

As previously mentioned, this method supports multiple functions simultaneously. Allowable query functions are *min*, *max*, *medium*, σ*-percentile*, *average*, and *summation* aggregation. The cloud server sends all these queries together to the fog center. The fog center sends each newly generated query set to a fog node to be answered, and the fog node returns the results to the cloud server. In detail, the process is as follows:

- *Step 1:* Query set generation. The fog nodes cannot answer *min*, *max*, *medium*, σ*-percentile*, and average queries directly, which means the fog center must

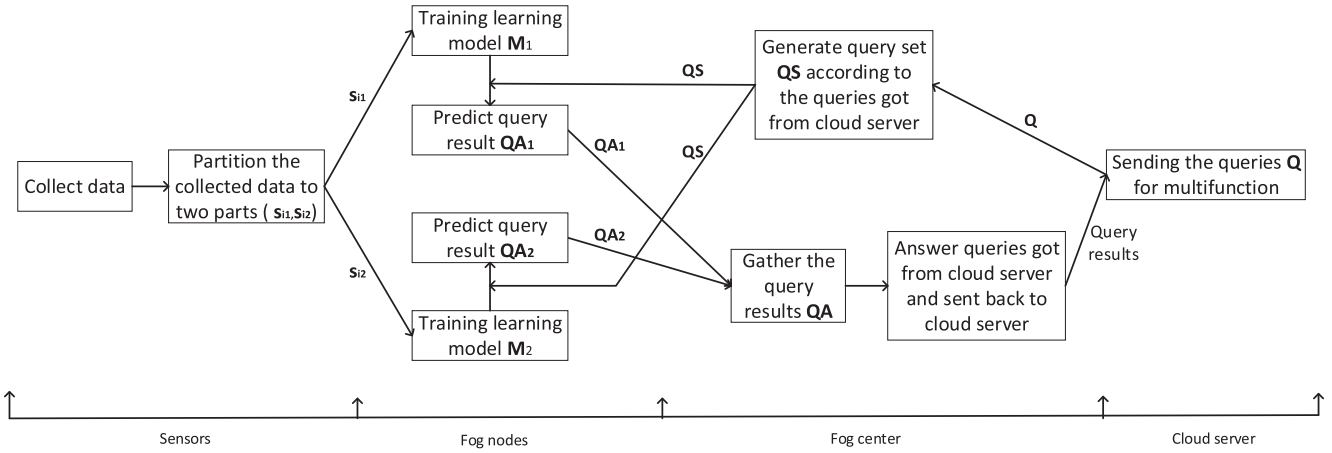


FIGURE 2. Aggregation process.

TABLE 2. New query generation.

New Query	s_1	s_2	s_3	s_4
q_1	1	0	0	0
q_2	0	0	1	0
q_3	0	0	0	1

generate the proper queries first. To illustrate this process, consider a min aggregation as an example. Assuming the query $q = min(s_1, s_3, s_4)$ represents the *min* value of sensors s_1, s_3 and s_4, the fog center generates three independent queries to determine the value of each sensor, as shown in Table 2. The same method is used for *max*, *medium*, and σ-*percentile* queries.

To calculate average queries, we simply sum the values of the queried sensor.

- Calculating the original query results. Assume a query set $Q(q_1, q_2, \ldots q_n)$ is a newly generated query that requires different aggregation functions, say, *min*, *max*, *medium*, σ-*percentile* and *average*. The methods for calculating the corresponding query results are shown below:
  - Min: $min_D = min\{q_1(D), q_2(D), \ldots q_n(D)\}$
  - Max: $max_D = max\{q_1(D), q_2(D), \ldots q_n(D)\}$
  - Medium: If n is odd, $med_D = q_i(D)$, where

$$\begin{cases} |[min, q_i(D)| \geq (n+1)/2 \\ |[q_i(D), max|) \geq (n+1)/2 \end{cases}$$

If n is even, $med_D = (q_i(D) + q_j(D))/2$, where

$$\begin{cases} |[min, q_i(D)]| \geq n/2 \\ |[q_i(D), max]| \geq n/2 + 1 \\ |[min, q_j(D)]| \geq n/2 + 1 \\ |[q_j(D), max]| \geq n/2 \end{cases}$$

  - σ-percentile: $per_D = q_i(D)$, where

$$\begin{cases} |[min, q_i(D)]| \geq \lfloor \sigma n/100 \rfloor \\ |[q_i(D), max]| \geq \lfloor (100-\sigma) n/100 \rfloor \end{cases}$$

  - Average: $ave_D = \frac{\sum q_i(D)}{n}$
  - Summation: $sum_D = \sum_{i=1}^{n} q_i(D)$

In the above, *min*, *max*, *medium*, σ-*percentile*, and *average* of a dataset D are denoted as min_D, max_D, med_D, per_D, and ave_D, respectively. $|[a, b]|$ refers to the number of values that fall within the range $[a, b]$. Once calculated, the fog center sends the aggregation results back to the cloud server for further processing.

2) SENSOR REPORT GENERATION

Assume that the sensors report their sensory data to the fog nodes every 15 minutes. And to provide the required range of services, they must report their data simultaneously. To avoid disclosing any real information to the fog nodes, a simple algorithm that resides on the sensor device partitions the data before it is sent. Specifically, each sensor $s_i \in S_{fc}$ gathers sensory data m at time point t_γ and carries out the following protocol:

- *Step 1:* A random number $\kappa \in 0$ m is generated for the current time point t_γ.
- *Step 2:* The sensor reports κ to the fog node f_1 through a wireless network.
- *Step 3:* The sensor calculates the value of ι and reports it to the fog node f_2, where $\iota = \mathfrak{m} - \kappa$.

3) PREDICTING THE QUERY RESULTS WHILE PRESERVING PRIVACY

After receiving all the reported data from the sensors, the fog node predicts the query results according to the following steps:

- *Step 1:* Generate a training set.
  The fog node generates a query set QS with ν queries. Each query includes S_{fc} features, which are the features

of the sensory data. Sensitivity needs to be considered during the process of generating the training set because, without proper calibration, substantial errors can occur. Query sensitivity is defined as follows:

Definition 1 (Query Sensitivity): Given a group of queries $Q(q_1, q_2, \ldots, q_v)$ over a data set D, the query sensitivity S_Q is defined as follow:

$$S_Q = max \sum_{i=1}^{v} sign(\mid q_i(D) - q_i(D') \mid), \tag{4}$$

where D' is the neighbouring dataset of D.

Query sensitivity evaluates how many queries results are affected by a single record. To reduce the query sensitivity, the feature being queried is controlled within S_{max} times in each query set, where $S_{max} \leq v$.

To ensure the model satisfies differential privacy and can defend against differential attacks, *Laplace* noise is added to the query results. Specifically, the noisy answer $\hat{QA} = QA + \{Lap(S_{max}/\epsilon), Lap(S_{max}/\epsilon), \ldots, Lap(S_{max}/\epsilon)\}$, where QA represents the vector of the query results.

- *Step 2:* Training the learning model.
  The training set generated in the last step $< Q, \hat{QA} >$ is used to train the learning model. Given the sensory data is made up of numerical values, the model M could be trained using a variety of regression algorithms. In this paper, we used a simple linear regression algorithm that demonstrated good performance during the experiments.
- *Step 3:* Predicting the query results.
  The trained model is then used to predict the results of fresh queries Q sent by fog center. Specifically, $Q(\hat{D}) = MQ$, $Q(\hat{D})$ is the noisy answers of queries.

In summary, the proposed method addresses the three challenges mentioned in Section I - computation overheads, communication overheads, and multifunctional aggregation. The lack of required encryption technology ameliorates the computation overhead, and introducing a machine learning process coupled with a fog architecture allows for more powerful computing power and greater storage capabilities. As such, the sensor nodes only need to report raw, unprocessed data, and the fog center distributes tasks to a number of fog nodes, which reduces the burden on the cloud server. Communication efficiency is improved by only reporting the aggregation results to the cloud server rather than all the sensory data. And the last section demonstrates the power of multifunctional aggregation within the proposed protocol.

V. PRIVACY AND UTILITY

In this section, we theoretically analyze the privacy and utility of our method.

A. PRIVACY ANALYSIS

In the proposed method, generating the training set is the only process that consumes the privacy budget. Theorem 1 shows that the proposed data release method satisfies ϵ-differential privacy.

Theorem 1: Each record in a given dataset D represents the sensory data of one sensor, and each record is independent of the others. Thus, the proposed privacy-preserving aggregation method can provide ϵ - differential privacy.

Proof: Let Q be a set of training queries. *Laplace* noise is added to the query results, generating a noisy answer $Q(\hat{D}) = Q(D) + Laplace(s_{max}/\epsilon)$. Throughout the entire process, the original dataset D can only be accessed by the training queries. The process for training the model is based on the training dataset, whereas the prediction process is based on the trained learning model. These processes do not consume any of the privacy budget and cannot disclose any private information because the original dataset is not interrogated. Therefore, every aspect of this aggregation method satisfies ϵ-differential privacy. Additionally, the original sensory data is divided into two parts and reported separately to the two fog nodes. Each fog node conducts its protocols independently. Hence, each fog node also satisfies ϵ-differential privacy.

In the analysis below, we examine the composite property of the privacy budget for the entire dataset to determine the privacy guarantee is satisfied.

Theorem 2 (Parallel Composition [13]): Assume we have a set of privacy mechanisms $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_m\}$, and each $\mathcal{M}_\rangle$ provides ϵ_i privacy guarantee on a disjoint subset of the entire dataset, M provides $max(\epsilon_i)$ - differential privacy.

Theorem. 2 directly illustrates the privacy guarantee in the proposed method. The sensory data is sliced into two parts; therefore, the data received by the fog nodes are disjoint and independent of each other. According to Theorem. 2, the set of privacy mechanisms $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_m\}$ will consume the $max\{\epsilon_1, \epsilon_2, \ldots, \epsilon_m\}$ of the privacy budget. In our method, each fog node is assigned the same privacy budget; therefore, the proposed method preserves differential privacy. □

B. UTILITY ANALYSIS

In this section, we apply a well-known utility definition suggested by Abdallah and Shen [14] to measure the accuracy of the proposed privacy framework.

Definition 4 ((α, β)-Useful): A mechanism $\mathcal{M}$ is (α, β)-useful with respect to a set of queries, if for every data set D, with a probability of at least $1 - \beta$, the output of the mechanism $\mathcal{M}$ satisfies

$$Pr[max|\mathcal{M}(\hat{cell}_i) - \mathcal{M}(cell_i)| \leq \alpha] \geq 1 - \beta. \tag{5}$$

Based on the definition of accuracy (Definition 4), we demonstrate that a certain value of α bounds the errors caused by our method with a high probability.

Theorem 3: The output errors of a set of the queries on collected data caused by the proposed method is bounded by α with a probability of at least $1 - \beta$. The proposed method is satisfied with (α, β)-usefulness when $\alpha < max\{\sqrt{\frac{4sln\frac{2|H|}{\beta}}{m\epsilon}}, \sqrt{\frac{n^2 ln\frac{|H|}{\beta}}{m}}\}$.

Proof: The errors caused by the proposed method occur when noise is added to the training set and when training the model. Suppose the chosen learning algorithm for the model

has a hypothesis set $H = \{h_1, h_2, \ldots, h_i\}$ of size $|H|$. The error probability is denoted as follows:

$$Pr[error] \leq Pr[error_n] + Pr[error_m], \tag{6}$$

where $error_n$ refers to the errors caused by adding noise, and $error_m$ refers to the errors caused by training model.

To satisfy differential privacy, *Laplace* noise is added to the entire training set. The level of error can be calculated using the properties of *Laplace* noise, presented as sums of *Laplace* random variables, as shown in Lemma 1.

Lemma 1 (Sums of Laplace Random Variables [15]): Let $\lambda_1, \lambda_2, \ldots, \lambda_m$ be a set of independent random variables drawn from $Laplace(\sigma)$. Then for every $\alpha > 0$,

$$Pr(|\frac{\sum \lambda_i}{m}| > \alpha) = exp(-\frac{m\alpha^2}{4\sigma}). \tag{7}$$

As $error_n = \frac{\sum_i |f_i(\hat{D}) - f_i(D)|}{m}$, we have $Pr[error_n > \alpha] = Pr[\frac{\sum_i |f_i(\hat{D}) - f_i(D)|}{m} < \alpha]$. For each f_i, the number of errors are equal to the random variable λ_i sampled from $Laplace(\frac{s}{\epsilon})$. Therefore, $Pr[error_n > \alpha] = Pr(|\frac{\sum \lambda_i}{m}| > \alpha)$. According to the Lemma 1,

$$Pr[error_n > \alpha] = exp(-\frac{m\alpha^2}{4\sigma}) = exp(-\frac{m\epsilon\alpha^2}{4s}) \tag{8}$$

For all hypotheses $h \in H$, we then have

$$Pr[error_n > \alpha] = |H|exp(-\frac{m\epsilon\alpha^2}{4s}). \tag{9}$$

Let $\beta = 2|H|exp(-\frac{m\epsilon\alpha^2}{4s})$, we have $\alpha = \sqrt{\frac{4sln\frac{2|H|}{\beta}}{m\epsilon}}$.

The error $error_m$ can be analyzed with the help of the *Chernoff_Hoeffding* bound [15], shown as follow.

Lemma 2 (Real-Valued Chernoff-Hoeffding Bound [15]): Let $X_1, \ldots, X_m$ be independent random variables with $E[X_i] = u$ and $a \leq X_i \leq b$ for all i, then for every $\alpha > 0$,

$$Pr(|\frac{\sum_i X_i}{m}| > \alpha) \leq 2exp(\frac{-2\alpha^2 m}{(b-a)^2}). \tag{10}$$

All queries to train the model are range queries. If the dataset has n records and each value is 1, the output of the query range from 0 to n. As $error_m = \frac{\sum_i |f_i(D) - f_i(M)|}{m}$, $Pr[error_m > \alpha] = Pr[\frac{\sum_i |f_i(D) - f_i(M)|}{m}) > \alpha]$. According to Lemma 2, for each hypothesis $h \in H$, we have $Pr[error_m > \alpha] = Pr[\frac{\sum_i |f_i(D) - f_i(M)|}{m}) > \alpha] \leq 2exp(\frac{-2\alpha^2 m}{n^2})$. Thus, for all hypothesis, we then have

$$Pr[error_m > \alpha] \leq 2|H|exp(\frac{-2\alpha^2 m}{n^2}). \tag{11}$$

Let $\beta = 2 \times 2|H|exp(\frac{-2\alpha^2 m}{n^2})$, we have $\alpha = \sqrt{\frac{n^2 ln\frac{|H|}{\beta}}{m}}$.

Therefore,

$$\begin{aligned} Pr[error > \alpha] &\leq Pr[error_n > \alpha] + Pr[error_m > \alpha]] \\ &\leq |H|exp(-\frac{m\epsilon\alpha^2}{4s}) + 2|H|exp(\frac{-2\alpha^2 m}{n^2}) \end{aligned} \tag{12}$$

Let $\beta = |H|exp(-\frac{m\epsilon\alpha^2}{4s}) + 2|H|exp(\frac{-2\alpha^2 m}{n^2})$, we get that when $\alpha < max\{\sqrt{\frac{4sln\frac{2|H|}{\beta}}{m\epsilon}}, \sqrt{\frac{n^2 ln\frac{|H|}{\beta}}{m}}\}$, the accuracy of proposed method satisfies the $(\alpha, \beta) - useful$ definition. In other worlds, the error is controlled by $\alpha = max\{\sqrt{\frac{4sln\frac{2|H|}{\beta}}{m\epsilon}}, \sqrt{\frac{n^2 ln\frac{|H|}{\beta}}{m}}\}$ with a probability of at least $1 - \beta$.

□

VI. EXPERIMENT EVALUATION

A. EXPERIMENT SETUP

1) DATASET

We used two real-world datasets to evaluate the performance of our method. The Reference Energy Disaggregation Data Set (REDD) contains specific information about the electricity consumption of many real homes over several months. MHEALTH [16] is a mobile health dataset, which contains more than 1 million records, each comprising the data from 24 different sensor signals. Given each signal is at the same scale, we randomly chose one type of signal for evaluation.

2) METRICS

We used the mean absolute error (MAE) to evaluate the accuracy of the results, defined as follows:

$$MAE = \frac{1}{m} \sum_{Q_i \in Q} |Q_i(\hat{D}) - Q_i(D)|, \tag{13}$$

where $Q_i(D)$ is the true aggregation result for one query, and $Q(\hat{D})$ is the perturbed aggregation result that calculated through our aggregation framework. A lower MAE represents a higher accuracy.

3) COMPARISON

Within our proposed aggregation framework, multifunctional aggregation could be achieved very simply using a traditional *Laplace* differential privacy method (LapDP). The fog node could be used as regional storage and to release the query results used in the aggregation function calculations. Hence, we compared our machine learning-based method (MLDP) to the traditional LapDP method.

TABLE 3. Parameters.

Parameter	Description	Value	Default
T	Size of training set	$1 - 500$	200
Q_s	Size of query set	$1 - 100$	100
ϵ	Privacy budget	$0.1 - 1$	1
σ	Percentile query parameter	–	0.3

4) PARAMETERS

Table 3 lists the parameter settings for our experiment.

B. EXPERIMENT RESULTS

To compare the performance of the proposed method with LapDP, we assessed the results of several aggregation

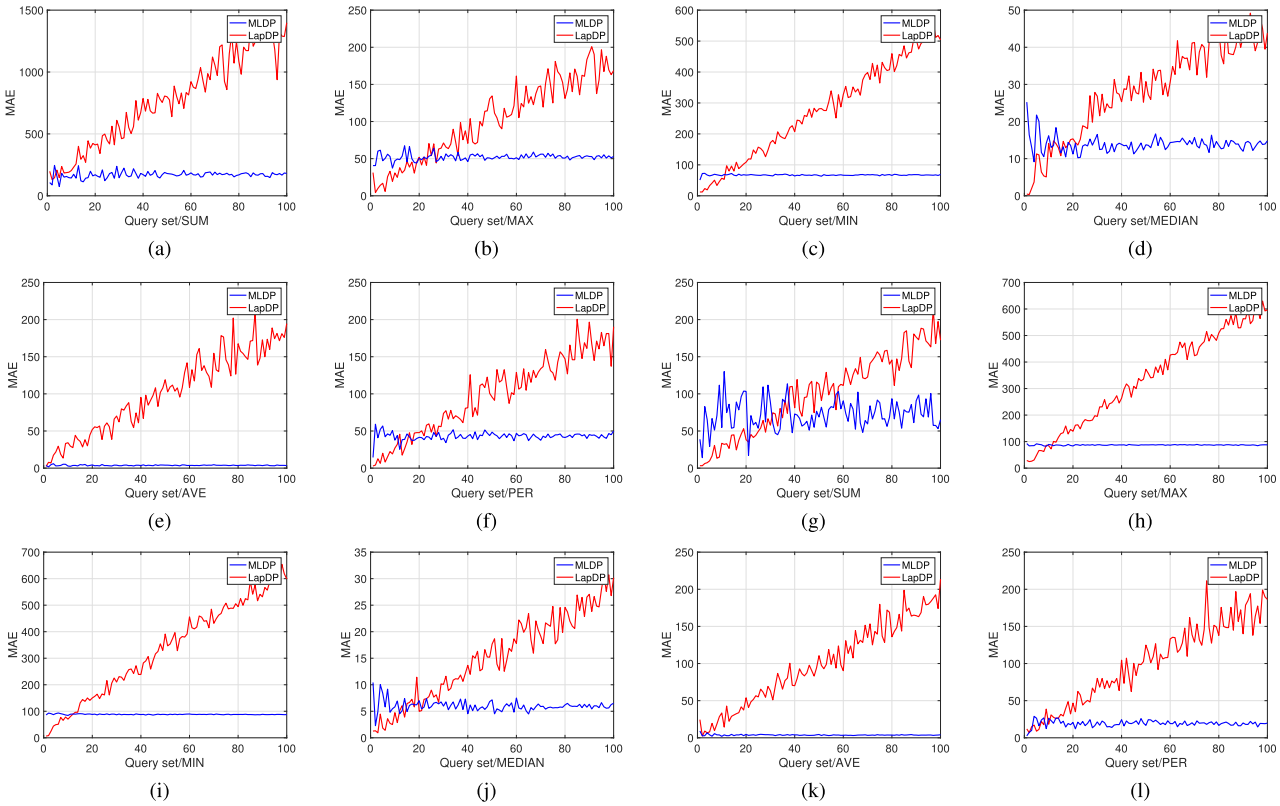


FIGURE 3. Performance with different sizes of query set. (a)–(f) REDD. (g)–(l) MHEALTH.

functions - *sum*, *max*, *min*, σ *-percentile* and *average* - in terms MAE with a number of different conditions. These were:

1) PERFORMANCE BY VARYING SIZE OF QUERY SET

Varying size of query set: The query set is used to calculate all aggregation functions. This experiment examined the performance of the proposed method on both the REDD and MHEALTH datasets with query sets Qs ranging from 1 to 500.

Fig. 3 shows the impact of the size of the query set on the performance of both methods in terms of MAE. With all aggregation functions on all the datasets, LapDP's MAE linearly increased as the size of query set grew, while MLDP remained stable. This is because, given a fixed privacy budget, the sensitivity in LapDP increases linearly with the growth of query set and, in turn, the amount of noise added to the query result also increases linearly. However, because MLDP satisfies differential privacy during the training process, the size of the query set has no effect on performance with a fixed privacy budget.

We also observed that LapDP showed better performance than MLDP with a small enough query set. But MLDP significantly outperformed LapDP as the size of the query set grew. For example, Fig. 3h shows the performance results for the *max* function on the REDD dataset. At a $Qs < 20$, MLDP has a higher MAE than LapDP, whereas at $Qs > 20$, MLDP's MAE is lower than LapDP. Similarly, Fig. 3d shows MLDP with a higher MAE than LapDP up to $Q_s \approx 18$, at which point it starts to perform better than LapDP. We find the same results for other aggregation functions on both REDD (Figs. 3g-3f) and the MHEALTH dataset (Figs. 4g-4l). For example, LapDP performed 50% better than MLDP with the min function on the MHEALTH dataset, with an MAE of 42.2 compared to MLDP's 94.1 at $Qs = 5$. However, at $Qs = 14$, LapDP and MLDP show similar performance, with an MAE of 96.4 and 94.3, respectively, and at $Qs > 20$, MLDP significantly outperforms LapDP. These results indicate that MLDP performs well, and significantly outperforms the traditional *Laplace* method, when calculating aggregation functions on large datasets.

2) VARYING THE LEVELS OF PRIVACY BUDGET

The privacy budget determines the amount of noise that is added to the training set and the query results. To determine how the privacy budget contributes to the final aggregation results, we changed the budget from 0.1 to 1 in steps of 0.1 for both datasets and fixed the training and query sets.

Fig. 4 shows the variations in the tendencies of all aggregation functions for the REDD and MHEALTH datasets along with the privacy budget ϵ. We observe that the MAE decreased as the privacy budget ϵ increased with both MLDP and LapDP. This is because a smaller privacy budget ϵ means more noise needs to be added. Correspondingly, as the privacy

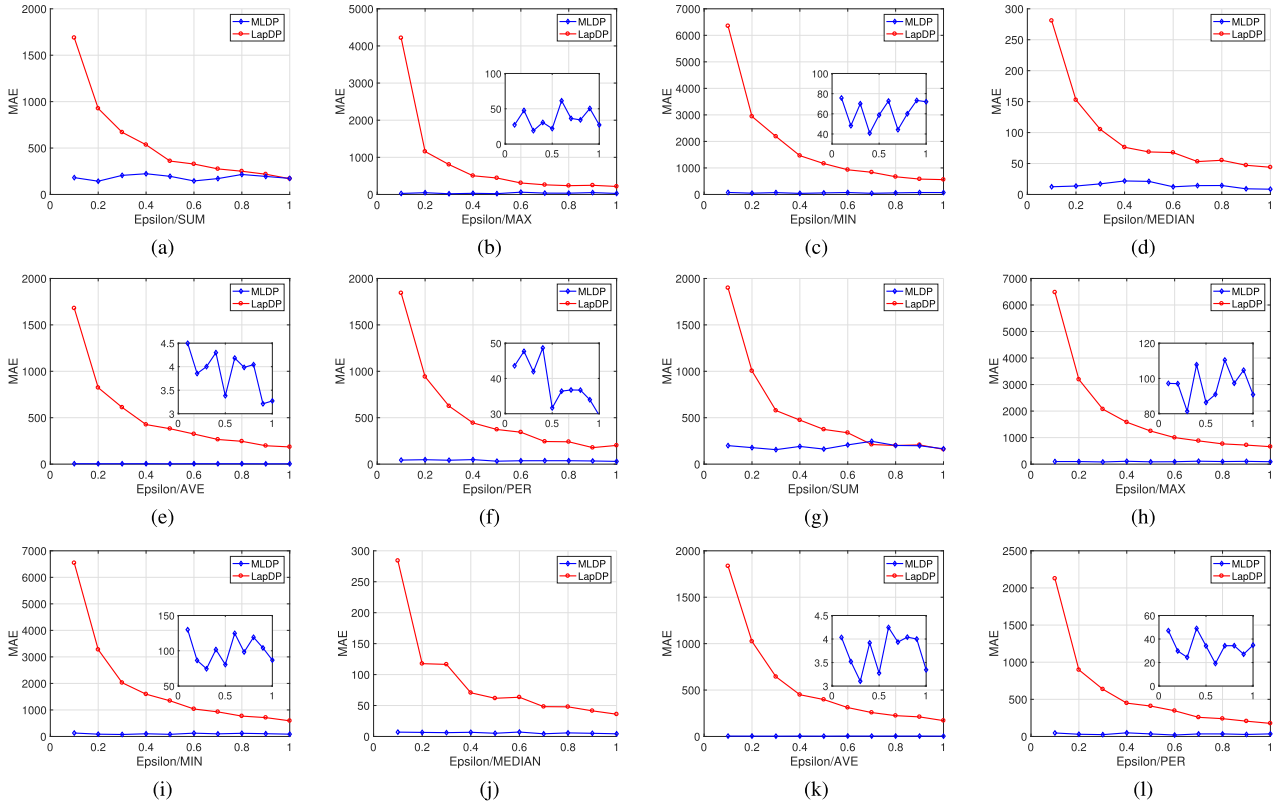


FIGURE 4. Performance with different privacy budgets. (a)–(f) REDD. (g)–(l) MHEALTH.

budget increases, less noise needs to be added, which means the results are less perturbed, leading to higher accuracy and a smaller MAE. In addition, we observed that our method consistently outperformed LapDP, with a lower MAE for all aggregation functions. As shown in Figs. 4b and 4c, when $\epsilon = 0.2$, LapDP scored an MAE of 1156 and 2941 for the max and min functions, respectively, while MLDP scored 47.61 and 48.17 - a significant improvement. When $\epsilon = 0.8$, LapDP resulted in an MAE of 236 for the max function and 667 for the min function, which is much larger than the MAE values of 34.24 and 60.00 for our method. We observed similar results for the other aggregation functions, as shown in Figs. 4a, 4d, 4e, and 4f. In addition, we observed that varying the privacy budget had a tremendous impact on LapDP's performance, while MLDP only showed small changes in performance. For example, in Fig. 4e, when $\epsilon = 1$, LapDP's MAE was 190 for average aggregation, yet at $\epsilon = 0.1$, LapDP's MAE rose to 1781 - an increase of around 90%. In contrast, MLDP's MAE rose from 171 to 180 - an increase of only around 6%. Figs. 4g - 4l show the results for the MHEALTH dataset with similar observations. In Fig. 4g, the sum aggregation at $\epsilon = 0.2$ resulted in an MAE of around 200 for MLDP, outperforming LapDP by about 80% at 1000. But at $\epsilon = 0.5$, LapDP's MAE reduced dramatically to 352; however, MLDP's MAE was around 184, still performing better than LapDP. Fig. 4l shows the results for the σ-percentile aggregation. Here, MLDP consistently performed better than LapDP, and the MAE changed rapidly as the privacy budget increased.

This is because LapDP has a much higher sensitivity than MLDP to begin with, which means it adds much more noise to the original data. Hence, when the privacy budget is halved, the amount of noise doubles. In LapDP's case, this doubling results in a huge amount of noise which significantly impacts accuracy, while for MLDP, doubling the small level of initial noise does not result in nearly as great a drop in accuracy.

3) PERFORMANCE BY VARYING SIZE OF TRAINING SET

Our theoretical analysis indicated that the size of the training set would play a vital role in the accuracy of the aggregation result. To observe the change in performance with different sized training sets, we increased the number of instances from 1 to 500 and tested all the aggregation functions using MLDP on both datasets. We then compared the results to the MAEs for the LapDP method with both a fixed privacy budget and fixed query set size.

Fig. 5 shows the results for the REDD and MHEALTH datasets, illustrating that the performance of the proposed method is greatly improved by increasing the size of the training set, initially, but once the training set reaches a certain value, the MAE reaches its nadir and become stable. As shown in Fig. 5e, the MAE continues to decrease until

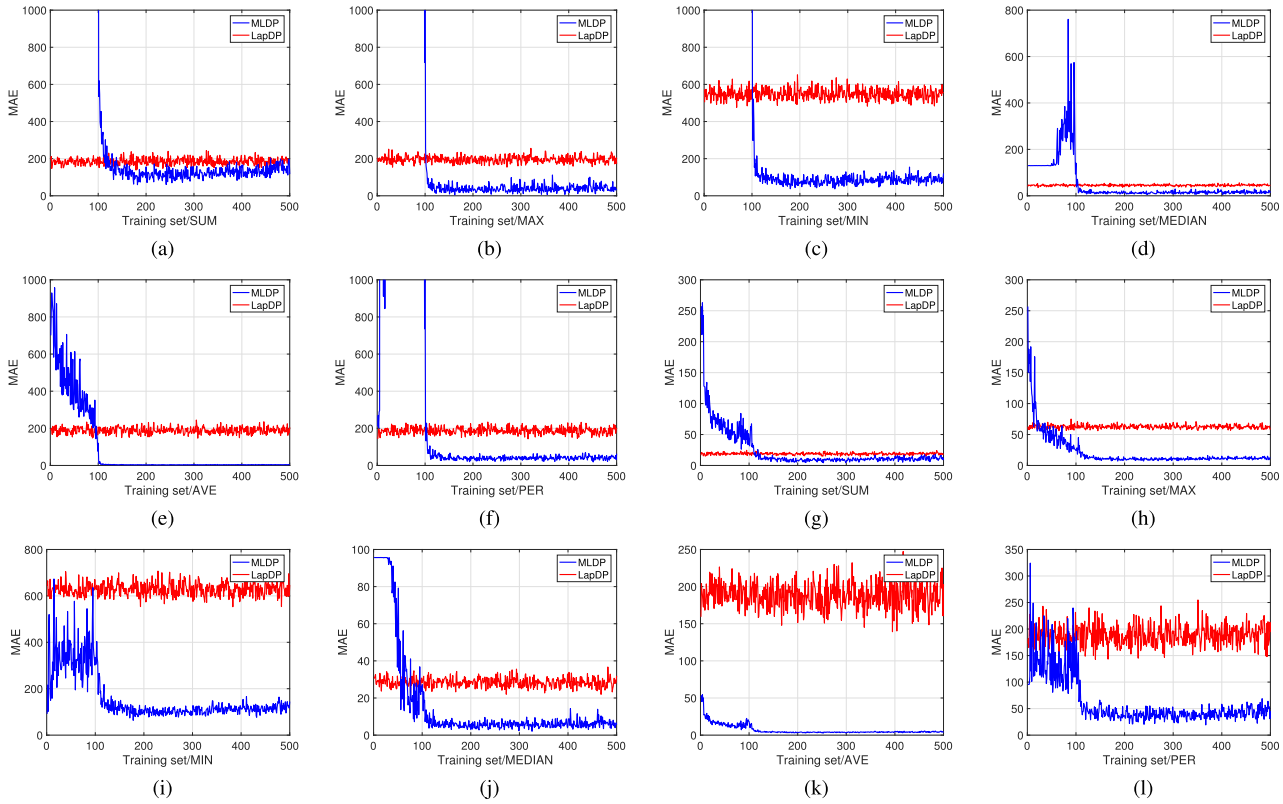


FIGURE 5. Performance with different sizes of training set. (a)–(f) REDD. (g)–(l) MHEALTH.

the training set contains 120 record where, at $MAE = 4.0$, the MAE reaches its lowest point. Subsequent increases in the size of the training set result in an MAE that fluctuates around 4. Fig. 5f shows the results for the σ - percentile aggregation. When the size of the training set is below 100, the MAE is very high but decreases significantly as the size of the training set increases, but at $T > 150$, the MAE no longer decreases. Similar results were observed on the MHEALTH dataset, as shown in Figs. 5g - 5l.

Given MLDP's performance is impacted by a mixture of noise and model errors, when the size of the training set is small, the sensitivity and noise levels are small, so the model errors play a more dominant role. Hence, the MAE decreases significantly with an increase in the size of the training set. However, beyond a certain threshold, a large training set carries too much sensitivity and noise to offset the increase in accuracy size brings. At this point, noise reduces the utility of the model and the MAE stops decreasing.

VII. RELATED WORK

Existing data aggregation methods typically use homomorphic encryption when aggregating data to ensure privacy [3], [17]–[21]. Zhang *et al.* [17] proposed a solution based on peer-to-peer protocols, called VPA, to preserve privacy in people-centric urban sensing systems. VPA supports a wide range of statistical additive and non-additive aggregations, but cannot defend against the differential attacks common to most data aggregation scenarios. Zhang *et al.* [19] proposed a priority-based aggregation solution for health data (PHDA), which includes privacy protection and also improves the cloud aggregation efficiency of the cloud service and the privacy of data privacy in WBANs. PHDA uses the relationships between its users and fixed social spots to choose the best relay for providing reliable data aggregation. In addition, PHDA can also withstand both internal and external forgery attacks, but it does not handle differential attack very well. Li *et al.* [21] proposed an efficient privacy-preserving protocol, called EPADA, which calculates sum aggregations from time-series data. The protocol uses additive homomorphic encryption and a novel key management technique to support a large plain-text space. Although the proposed method is easily extendable to min aggregations with just one round of communication, it is more difficult to adapt to compute multifunctional aggregations, especially non-additive aggregate functions, such as percentile and average. Han *et al.* [3] proposed a privacy-preserving multifunctional aggregation mechanism, also for health data. The cloud server is able to calculate multiple statistical functions and provides a range of services, each with privacy protection. This method supports both additive and non-additive aggregation functions.

However, all these schemes using encryption technology to protect the user's data and, since encryption usually results in a significant computational overhead, they are not practical for use with energy-limited sensors like smartphones. In addition, the computational burden on the cloud

server is heavy, especially when aggregating data that is reported with high frequency. A fog computing architecture allows computing services to reside at the edge of the network. Hence, a local aggregation device can be used to calculate the query results, which reduces the communication and computation overheads on the cloud server. Several papers have already explored privacy problems related to data aggregation in fog computing [22]–[25]. For example, Huang *et al.* [22] proposed a model that filters multiple encrypted XML streams and performs aggregation operations without decryption in a fog node. Lu *et al.* [25] proposed a lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT devices. However, most also include homomorphic encryption schemes, which does not solve the problem of sensors with limited energy resources.

VIII. CONCLUSION

In this paper, we proposed a privacy-preserving multifunctional data aggregation method based on machine learning. Within the method, a training dataset comprising the aggregation queries is used to train a machine learning model, which in turn predicts the aggregation results. The method allows for multiple aggregation functions without disclosing a user's privacy. The framework operates within a fog computing architecture, which means the computationally heavy aggregation tasks are distributed to the edge of the network, alleviating this burden from the cloud server. Additionally, only the aggregation results are sent to the server rather than all the sensory data, which significantly improves communication efficiency. Experimental results prove that the proposed method answers various aggregation queries with high accuracy.

REFERENCES

[1] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: Privacy-preserving multi-subset data aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 462–471, Feb. 2018.

[2] C. Xu, R. Lu, H. Wang, L. Zhu, and C. Huang, "PAVS: A new privacy-preserving data aggregation scheme for vehicle sensing systems," *Sensors*, vol. 17, no. 3, p. 500, Mar. 2017, doi: 10.3390/s17030500.

[3] 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 Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 1940–1955, Sep. 2016, doi: 10.1109/TIFS.2015.2472369.

[4] J. Sun, R. Zhang, J. Zhang, and Y. Zhang, "PriStream: Privacy-preserving distributed stream monitoring of thresholded PERCENTILE statistics," in *Proc. INFOCOM*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[5] 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 Trans. Smart Grid*, vol. 8, no. 5, pp. 2411–2419, Sep. 2017, doi: 10.1109/TSG.2017.2720159.

[6] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems," in *Proc. MobiHoc*, Paderborn, Germany, 2016, pp. 341–350.

[7] F. Qiu, F. Wu, and G. Chen, "Privacy and quality preserving multimedia data aggregation for participatory sensing systems," *IEEE Trans. Mobile Comput.*, vol. 14, no. 6, pp. 1287–1300, Jun. 2015, doi: 10.1109/TMC.2014.2352253.

[8] W. Zeng, Y. Lin, J. Yu, S. He, and L. Wang, "Privacy-preserving data aggregation scheme based on the P-function set in wireless sensor networks," *Ad Hoc Sensor Wireless Netw.*, vol. 21, nos. 1–2, pp. 21–58, Jan. 2014.

[9] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "An improved deep computation model based on canonical polyadic decomposition," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[10] 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 *Proc. GLOBECOM*, Austin, TX, USA, Dec. 2014, pp. 4720–4725.

[11] A. Abdallah and X. S. Shen, "A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 396–405, Apr. 2016, doi: 10.1109/TSG.2016.2553647.

[12] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. TCC*, Austin, TX, USA, 2006, pp. 265–284.

[13] F. McSherry and I. Mironov, "Differentially private recommender systems: building privacy into the net," in *Pro. SIGKDD*, Paris, France, 2009, pp. 627–636.

[14] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *J. ACM*, vol. 60, no. 2, pp. 12:1-12:25, Apr. 2013, doi: 10.1145/2450142.2450148.

[15] S. P. Kasiviswanathan, H. K. Lee, and K. Nissim, "What can we learn privately?" *SIAM J. Comput.*, vol. 40, no. 3, pp. 793–826, Jun. 2011, doi: 10.1137/090756090.

[16] P. Leandro, L. C. Liming, D. N. Chris, and B. Jose, "MHEALTH dataset," in *Ambient Assisted Living and Daily Activities*, 6th ed. New York, NY, USA: Springer, 2014.

[17] R. Zhang, J. Shi, Y. Zhang, and C. Zhang, "Verifiable privacy-preserving aggregation in people-centric urban sensing systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 268–278, Sep. 2013, doi: 10.1109/JSAC.2013.SUP.0513024.

[18] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "PPHOPCM: Privacy-preserving high-order possibilistic c-means algorithm for big data clustering with cloud computing," *IEEE Trans. Big Data*, to be published.

[19] 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," *Inf. Sci.*, vol. 284, pp. 130–141, Nov. 2014, doi: 10.1016/j.ins.2014.06.011.

[20] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and M. J. Deen, "Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning," *IEEE Internet Things J.*, to be published.

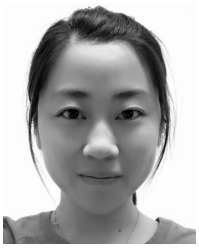
[21] Q. Li, G. Cao, and T. F. La Porta, "Efficient and privacy-aware data aggregation in mobile sensing," *IEEE Trans. Depend. Sec. Comput.*, vol. 11, no. 2, pp. 115–129, Mar. 2014, doi: 10.1109/TDSC.2013.31.

[22] J.-Y. Huang, W.-C. Hong, P.-S. Tsai, and I.-E. Liao, "A model for aggregation and filtering on encrypted XML streams in fog computing," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 5, pp. 1–14, May 2017, doi: 10.1177/1550147717704158.

[23] M. S. H. Nazmudeen, A. T. Wan, and S. M. Buhari, "Improved throughput for power line communication (PLC) for smart meters using fog computing based data aggregation approach," in *Proc. ISC2*, Trento, Italy, 2016, pp. 1–4.

[24] M. Gheisari, G. Wang, and M. Z. A. Bhuiyan, "A survey on deep learning in big data," in *Proc. EUC*, Guangzhou, China, 2017, pp. 173–180.

[25] 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, doi: 10.1109/ACCESS.2017.2677520.



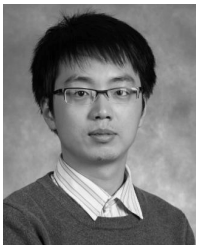
MENGMENG YANG received the B.Eng. degree from Qingdao Agricultural University, China, in 2011, and the M.Eng. degree from Shenyang Normal University, China, in 2014.

She is currently pursuing the Ph.D. degree with the School of Information Technology, Deakin University, Australia. Her research interests include privacy preserving, machine learning, and network security.

TIANQING ZHU (M'11) received the B.Eng. and M.Eng. degrees from Wuhan University, China, in 2000 and 2004, respectively, and the Ph.D. degree in computer science from Deakin University, Australia, in 2014.

From 2004 to 2011, she served as a Lecturer with Wuhan Polytechnic University, China. She is currently a Lecturer with the School of Information Technology, Deakin University. Her research interests include privacy preserving, data mining, and network security.

Dr. Zhu received the Best Student Paper Award in PAKDD 2014.



BO LIU received the B.Sc. degree from the Department of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004, and the M.Eng. and Ph.D. degrees from the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2007 and 2010 respectively.

He was an Assistant Research Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, from 2010 to 2014, and a Post-Doctoral Research Fellow with Deakin University, Australia, from 2014 to 2017. He has been a Lecturer with the Department of Engineering, La Trobe University, since 2017. His research interests include wireless communications and networking, security, and privacy issues in wireless networks.

YANG XIANG (SM'12) received the Ph.D. degree in computer science from Deakin University, Australia.

He is currently the Dean of Digital Research and Innovation Capability Platform, Swinburne University of Technology, Australia. He has published over 200 research papers in many international journals and conferences. His research interests include cyber security, which covers network and system security, data analytics, distributed systems, and networking. In particular, he is currently leading his team to develop active defense systems against large-scale distributed network attacks.

He is the chief investigator of several projects in network and system security, funded by the Australian Research Council. Two of his papers were selected as the featured articles in 2009 and 2013 issues of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. Two of his papers were selected as the featured articles in 2014 issues of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. He has been a PC Member for over 80 international conferences in distributed systems, networking, and security. He has served as a program/general chair for many international conferences.

WANLEI ZHOU (SM'98) received the B.Eng. and M.Eng. degrees in computer science and engineering from the Harbin Institute of Technology, Harbin, China in 1982 and 1984, respectively, the Ph.D. degree in computer science and engineering from Australian National University, Canberra, Australia, in 1991, and the D.Sc. degree from Deakin University in 2002.

He is currently the Alfred Deakin Professor and the Chair of information technology with the School of Information Technology, Deakin University. He has published over 300 papers in refereed international journals and refereed international conferences proceedings. His research interests include distributed systems, network security, bioinformatics, and e-learning.

Dr. Zhou has chaired many international conferences and has been invited to deliver keynote address in many international conferences.