Privacy-Preserving Federated Learning in Fog Computing

Chunyi Zhou, *Member, IEEE,* Anmin Fu, *Member, IEEE,* Shui Yu, *Senior Member, IEEE,* Wei Yang, Huaqun Wang, and Yuqing Zhang, *Member, IEEE*

***Abstract*—Federated learning can combine a large number of scattered user groups and train models collaboratively without uploading data sets, so as to avoid the server collecting user sensitive data. However, the model of federated learning will expose the training set information of users, and the uneven amount of data owned by users in multiple users' scenarios will lead to the inefficiency of training. In this paper, we propose a privacy-preserving federated learning scheme in fog computing. Acting as a participant, each fog node is enabled to collect Internet of Things (IoT) device data and complete the learning task in our scheme. Such design effectively improves the low training efficiency and model accuracy caused by the uneven distribution of data and the large gap of computing power. We enable IoT device data to satisfy ε-differential privacy to resist data attacks and leverage the combination of blinding and Paillier homomorphic encryption against model attacks, which realize the security aggregation of model parameters. In addition, we formally verified our scheme can not only guarantee both data security and model security, but completely resist collusion attacks launched by multiple malicious entities. Our experiments based on the Fashion-MNIST dataset prove that our scheme is highly efficient in practice.**

***Index Terms*—IoT, fog computing, federated learning, privacy-preserving.**

I. Introduction

TODAY, Internet of Things (IoT) plays a vital role in smart applications such as smart medical, smart home, smart transportation, and smart education [1], [2]. It is estimated that the number of IoT devices will exceed 50 billion by the end of 2020. Every day, extraordinary amounts of data are generated from IoT systems and injected into various storage spaces [3]. Such massive data provides a hotbed for machine learning to make effective use of these data [4]–[6]. In the face of the proliferation of machine learning applications in the IoT, fog computing has become a promising supplement to cloud computing by extending the cloud to the edge of the network to meet the computing and storage requirements of such applications [7].

Fog nodes usually collect IoT device data and send it to the cloud server to complete the centralized machine learning task. However, the centralized training mode of machine learning may cause some hidden security troubles [8], [9]. Centralized training mode needs a great number of data sets as training support, which often contains private information that users do not want to disclose [10], such as medical information, financial materials and so on. As a result, with the increasing awareness of large companies compromising on data security and user privacy, the emphasis on data privacy and security has become a major worldwide issue [11], [12]. For example, the General Data Protection Regulation (GDPR) was enforced by the European Union on May 25, 2018, which gives individual users inspection right, forgetting right, processing restriction right, and so on. Unfortunately, in centralized machine learning, users lose control of data after uploading data. Thus it is challenging to meet the requirements of GDPR under such a scenario.

To solve this problem, some work has applied cryptography algorithms and differential privacy to protect users' privacy. For example, Li et al. [13] used additive homomorphic encryption technology to classify data in the ciphertext, and Abadi et al. [14] added noise to the gradient in deep learning to satisfy differential privacy. Besides, secure multi-party computation is also adopted to realize privacy-preserving machine learning [15]. Nevertheless, the above proposals cannot fit perfectly in privacy-preserving machine learning. Furthermore, when differential privacy mechanisms add noise to model parameters, the utility of the learning result degrades as the privacy requirement becomes more stringent [16]. The above technologies under encryption operations and secure multi-party computation are difficult to extend in deep learning because of the complex calculation of the deep learning model.

In 2015, Shokri proposed a collaborative deep learning framework based on Stochastic Gradient Descent (SGD). It allows two or more participants, each with a local training dataset, to construct a joint model [17]. The framework does not need to centralize data, which fundamentally guarantees the users' control over their data. Unfortunately, the training process of this model is asynchronous and atomic, which

This work is supported by National Natural Science Foundation of China (61572255, U1836210, 61802186, 61941116), Six talent peaks project of Jiangsu Province, China (XYDXXJS-032), Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, NJUPT(BDSIP1909), and CERNET Innovation Project (NGII20190405). (Corresponding authors: Anmin Fu).

C. Zhou, A. Fu, and W. Yang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, PR China. A. Fu is also with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, PR China. (e-mail: zhouchunyi@njust.edu.cn; fuam@njust.edu.cn; generalyzy@njust.edu.cn).

S. Yu is with the School of Software, University of Technology Sydney, NSW, Australia. (e-mail: shui.yu@uts.edu.au).

H. Wang is with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, PR China. (email: whq@njupt.edu.cn).

Y. Zhang is with the National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing, PR China. (e-mail: zhangyq@ucas.ac.cn).

causes that the rest of the users will be waiting when one user trains, leading to inefficient training. To solve this problem, Google proposed federated learning [18] in 2016, enabling all users to complete training in parallel and much improved training efficiency. Smith et al. [19] propose a distributed, multi-task approach to accelerate the speed of federated learning. Bonawitz et al. [20] designed a parameter aggregation scheme in a blinding way, which realizes the model parameter privacy of each federated learning user. In order to meet users' privacy requirements, distributed training such as collaborative learning and federated learning are undoubtedly better choices.

Each user in federated learning needs long-term online continuous training and frequent interaction with the server. However, some IoT devices are usually equipped with limited computing capability and battery issues, which will make it difficult to maintain training. In addition, these devices may have a colossal data gap in reality. All of the above will reduce the training efficiency. Moreover, model security will be threatened through some attacks [21], [22], which obtains the information of training data set from model parameters. Although some schemes have proposed to satisfy model privacy [23], [24], they do not consider collusion attacks. Once a malicious user colludes with the server, the model security.

Our contributions. In this paper, we introduce fog computing to the federated learning framework. In the scheme, we use differential privacy to protect IoT device data, and design a model parameter security aggregation method based on blinding and homomorphic encryption. Our scheme can protect the data security of IoT devices and model security, and has the ability to resist collusion attacks.

Our contributions are as follows:

- We propose a federated learning scheme in fog computing, which compromises centralized and distributed training modes and provides practical application scenarios for federated learning.
- We use differential privacy technology to ensure IoT device data security, and protect the model security against the honest-but-curious parameter server by using Paillier homomorphic encryption. Our scheme effectively prevents the server from inferring training set information of a specific fog node from model parameters.
- We design a secure aggregation method of model parameters based on blinding and Paillier homomorphic encryption, which makes our scheme can resist collusion attacks. Our scheme can protect the privacy of model parameters, even if the untrusted parameter server or malicious fog node colludes with k nodes. Note that $k \leq n-2$, where n is the number of fog nodes.
- We give a security analysis for our scheme, which provides the guarantee of data security, model security as well as anti-collusion attacks. We evaluate our scheme over the Fashion-MNIST dataset and fully connected neural network (FCNN). The results demonstrate that our scheme is highly efficient in practice.

Roadmap. The rest of this paper is organized as follows. Section II presents the related work of centralized learning and distributed learning. Section III introduces the problem statement, including the system model and adversary model of our scheme. Section IV describes the detailed scheme. We analyze the security of our scheme in Section V, and report the experimental results in Section VI. Finally, Section VII concludes this paper.

II. Related Work and Motivation

In this section, we summarize the related work of centralized learning and distributed learning, respectively. Besides, we describe the motivation of this paper.

A. Centralized training

For privacy-preserving centralized learning, some researches focus on encrypting data sets and classifying them on the ciphertext. Bost et al. [25] was devoted to the classification of machine learning in ciphertext, and designed several building blocks as part of machine learning training process, so as to realize the privacy of user input data. Based on this work, Li et al. [13] proposed a privacy-preserving outsourcing scheme to classify encrypted data, in which the test data and the classifier are in ciphertext state. Work [26] also classified encrypted data sets in natural language processing. Although classification in ciphertext can guard the privacy of user data, the computing cost is not within the acceptable range of practical applications. Therefore, researchers took advantage of differential privacy and secure multi-party computation instead of ciphertext classification to balance the trade-off between privacy and efficiency.

Work [27] implemented naive Bayesian classifiers on multiple data sources using differential privacy and homomorphic encryption. Xu et al. [28] proposed a differential privacy GAN (Generative Adversarial Network) scheme GANobfuscator, which can achieve different privacy under GANs by adding carefully designed noise to grades during the learning procedure. The scheme can generate synthetic data according to the training task instead of using real training data, which provides privacy protection of training data.

Mohassel et al. [29] used secure multi-party computation to train a machine learning model between multiple users and two non-collusive servers. Jia et al. [30] proposed a similarity evaluation of the machine learning model in a distributed system based on Oblivious Transfer (OT) and Oblivious Evaluation of Multivariate Polynomials (OMPE). It effectively evaluates the similarity among models of different trainers without disclosing the model to test the similarity of their datasets. Unfortunately, they suffer from the collusion attacks, and the implementation conditions of OMPE are strict.

B. Distributed training

In centralized machine learning, the security operations added in the process of server collecting data limit the training efficiency. To solve this problem, Shokri et al. [9] gave the formal model and security definition of collaborative learning. Then Google's federated learning framework [18] extended collaborative learning to the parallel scene. The main idea is to build machine learning models based on datasets that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2987958, IEEE Internet of Things Journal

are distributed across multiple devices with preventing data leakage. Collaborative learning and federated learning transform centralized training into distributed training, which can ensure the user's data security without uploading data sets. Therefore, the framework of distributed training has attracted wide attention as soon as it was proposed. Aiming at the security problem of model parameters in federated learning, Bonawitz et al. [20] designed a secure aggregation scheme, which enables users to update the weighted average of vectors and cancel the blind factors among users during aggregation. In 2017, Hitaj et al. [31] designed the GAN attack based on collaborative learning, which enables malicious users to join the training process and generate data sets of honest users. It is a serious form of attack in distributed learning. Some other studies have devoted to solving the statistical challenges and improving security in federated learning [28].

In recent research, Abadi et al. [14] added Laplacian noise to gradients in the neural network to satisfy differential privacy. Nevertheless, Xiang et al. [16] pointed that although adding noise to the gradients could guarantee privacy, it would significantly reduce the accuracy of the model. Phong et al. [23] proposed a collaborative learning system based on additively homomorphic encryption. It uses asynchronous stochastic gradient descent to construct a neural network to ensure that user information is not accessed by semi-honest servers. However, it increases the communication overhead between users and cloud computing servers. In addition, the cloud server may intentionally delete or modify computing results [32]. To solve this problem, Xu et al. [33] designed a secure and verifiable federated learning, which protects the users' privacy and verifies the integrity of aggregated results.

C. Motivation

Existing works of centralized and distributed training usually utilize differential privacy, secure multi-party computation, and homomorphic encryption to satisfy data and model security. However, these tools have some drawbacks that discount the applicability of existing works. For example, the differential privacy that adds noise to the gradients of the neural network could lead to accuracy loss of models. Secure multi-party computation and homomorphic encryption suffer from high computing overhead, especially in large-scale scenarios. In addition, most of the existing distributed learning schemes only focus on data and model security, ignoring the fact that they are not suitable in practice. Currently, the discrete users train models locally in distributed learning, which dose satisfy the data security but sacrificing efficiency for security when the amount of data held by each user is uneven. Some researches [20], [23], [33] also fail in preventing the collusion attacks though guaranteeing the model security of distributed training.

Motivated by above issues, we introduce fog computing into our scheme and design the "IoT Devices – Fog Nodes – Cloud" architecture, which comprises the advantages of both centralized and distributed training modes. Particularly, we focus on reducing the impact of uneven data distribution on distributed training efficiency and model accuracy loss to improve its practicability, with the protection against the data attacks, model attacks, and collusion attacks.

III. Problem Statement

In this section, we first describe our system model. Later, we describe the adversary model of our scheme.

A. System Model

Extending from the cloud computing, fog computing deploys geographically distributed fog nodes in the edge network, providing computing power closer to both the IoT devices and users [7], [34]. Therefore, fog nodes with excellent computing and storage capabilities can be used as a link between IoT devices and cloud server. Figure 1 shows the system model in our scheme. There are three kinds of entities in the model: IoT devices, fog nodes, and cloud, which constitute the "IoT Devices – Fog Nodes – Cloud" three-tier architecture.

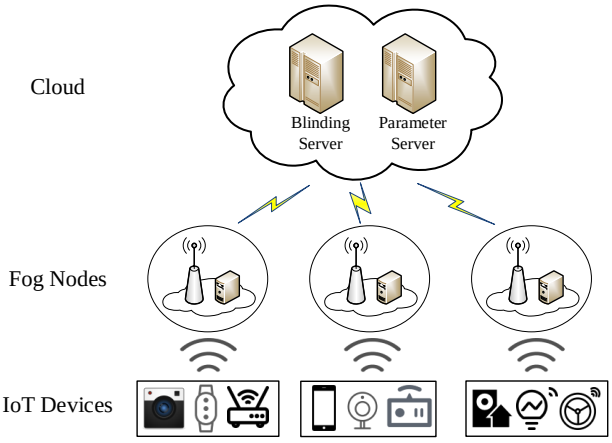


Fig. 1: The "IoT Devices – Fog Nodes – Cloud" architecture.

The entities are as follows:

- *IoT Devices* : They act as all participants in federated learning tasks. They have different amounts of data sets and different computing power, and this gap can not be ignored. IoT devices' data may contain private information, such as user health information, and their data sets are intended to train a model.
- *Fog Nodes* : They are the links between IoT Devices and Cloud, which have higher storage and computing power than IoT devices. They collect and store the IoT devices' data before the training. Then, each fog node trains the model locally and uploads the model parameters to the Cloud for parameter aggregation. In our scheme, the fog nodes are also responsible for the encryption and blinding of model parameters.
- *Cloud* : It consists of a parameter server and a blinding server in our scheme. They are responsible for training and distribution of initial model, aggregation, and update of model parameters in each round. The blinding server is trusted, while the parameter server is semi-honest.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2987958, IEEE Internet of Things Journal

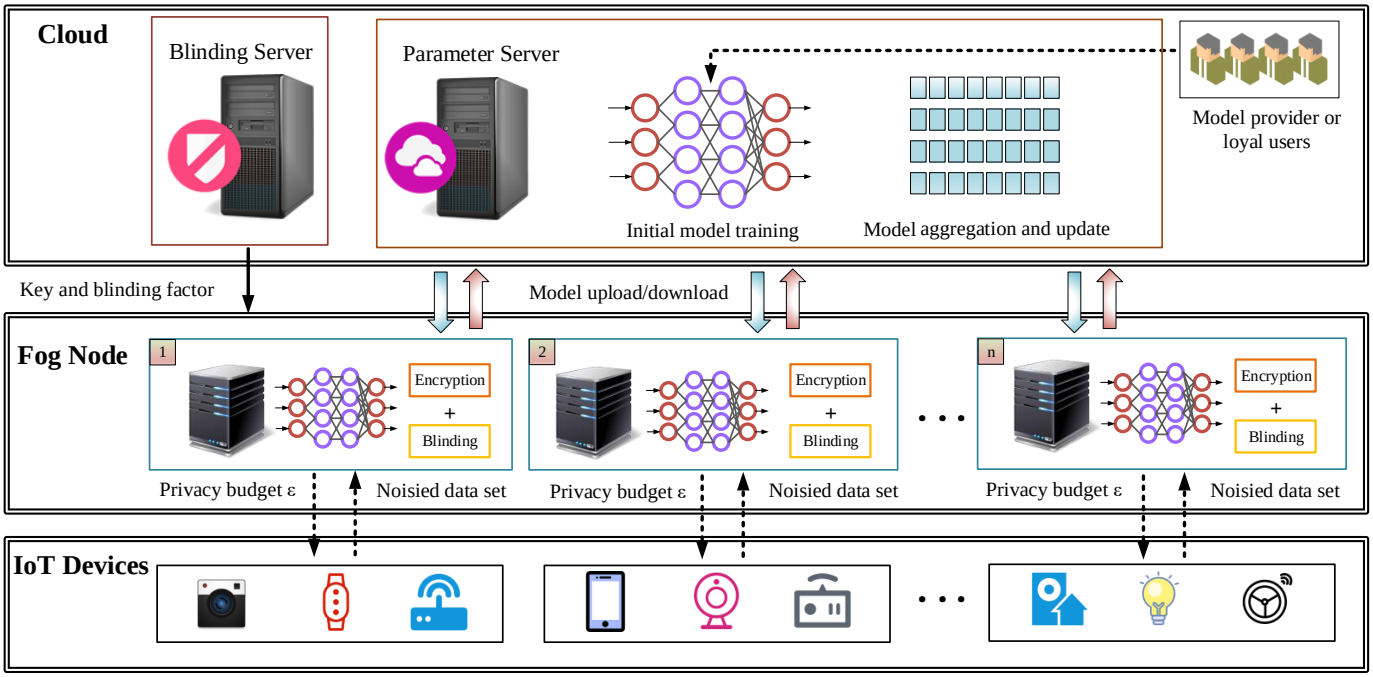


Fig. 2: Privacy-Preserving Federated Learning Scheme in Fog Computing.

B. Adversary Model

This paper focuses on the "semi-honest model", which means that the parameter server and fog nodes are honest-but-curious (we assume that the blinding server is trusted). They have the following characteristics: (1) They will honestly execute a protocol and does not intentionally terminate the protocol at any time; (2) They will not tamper with the calculation results; (3) They are curious about the input of other entities in the scheme (IoT devices and other fog nodes in this paper).

In our scheme, we consider three types of attackers: the **data attackers**, the **model attackers**, and the **collusive attackers**. The **data attackers** are malicious data collectors that may steal the privacy information of IoT devices. The **model attackers** may infer training data information through some model parameters. In addition, the **collusive attackers** may obtain model parameters of a specific fog node by multiple malicious entities. We assume that the entities in the scheme do not trust each other and there is collusion between (1) *the parameter server and the fog node* or (2) *the fog node and the fog node* in order to obtain the model parameters of a specific fog node.

Hence, in our system, we assume the following conditions are satisfied:

- Assume that the blinding server is trusted, because it is the institution that is responsible for generating the key and blinding factors. The blinding server does not participate in the transfer of model parameters.
- Assume that at most (n-2) fog nodes can collude with parameter server or a malicious fog node, where n is the number of all fog nodes.

IV. Our Scheme

In this chapter, we first give an overview of the privacy-preserving federated learning scheme in fog computing and then describe each process in detail.

A. Overview

Figure 2 is the specific framework of our scheme. As we discussed in Section III.A, our scheme structure is clearly divided into three-tier "IoT Devices – Fog Nodes – Cloud". These three tiers make up our privacy-preserving federated learning scheme, among which the Fog Nodes tier plays a key role in privacy protection. The fog nodes not only collect and transmit data but also process data and train local models. Therefore, the security of the Fog Nodes tier is quite vital, otherwise a lot of privacy information will be leaked. To prevent this, we utilize ε-differential privacy, Paillier homomorphic encryption and blinding to protect the privacy information and model parameters.

Our scheme consists of four phases: *Initialization*, *Collection of IoT Device Data*, *Fog Node Training* and *Aggregation of Model Parameters*, as shown in Figure 3. The details of each phase are as follows:

(1) *Initialization*: before the training, the cloud server (parameter server and blinding server) needs some initialization preparation. The parameter server will collect some data sets of loyal users in advance to train the initial model. The blinding server is responsible for generating the key pairs of Paillier homomorphic encryption and blinding factors. Then the blinding server transmits them to each fog node.

(2) *Collection of IoT Device Data*: after the initialization, each fog node starts to download the initial model from the parameter server. Then the fog node allocates the privacy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2987958, IEEE Internet of Things Journal

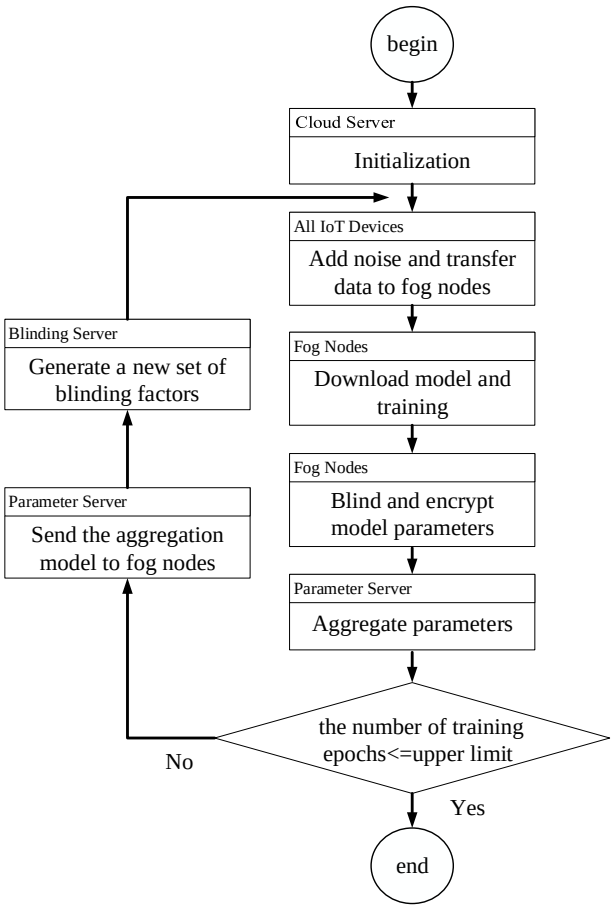


Fig. 3: The training process of our scheme.

budget according to the sensitivity of each IoT device node and collects their noise data set after adding Laplacian noise.

(3) *Training of fog nodes*: each fog node trains the neural network model locally. Note that the model attacker may infer training data through model parameters, and the collusive attackers may obtain the model parameters of a specific fog node, as we discussed in Section III. Thus, each fog node encrypts and blinds parameters before uploading them to the parameter server for secure aggregation.

(4) *Model parameter aggregation*: after the parameter server collects the model parameters uploaded by all fog nodes, it aggregates and updates the global model. Then the blinding server generates a new set of blinding factors and sends them to each fog node to start the next round of training.

Next, we will describe the details of each process.

B. Initialization

The initialization phase consists of blind factor generation, key distribution, and initial model training. The steps for initialization are as follows:

(1) First, the blinding server selects blinding factors $\{\eta_n\}$ such that $\eta_1 \cdot \eta_2 \cdot \ldots \cdot \eta_n = z^i$, where n is the number of fog nodes and $z^i \in \mathbb{Z}$. Then the z^i will be sent to the parameter server.

(2) The parameter server trains the initial model by downloading the model through a model provider or collecting data sets from loyal users who trust the server.

(3) Next, the blinding server generates the key pair of the Paillier homomorphic encryption, and transmits the key pair $\{pk,sk\}$ to each fog node. The private key sk will never be disclosed to the parameter server.

(4) Finally, the blinding server transfers η_n to each fog node, where i represents the round number of the current iteration. The content of $\{\eta_n^i\}$ will not be revealed to the parameter server, and z^i will not be revealed to any fog node.

C. Collection of IoT Device Data

After the initialization, each fog node assigns a privacy budget ε to each IoT device according to the amount of IoT devices. Then the IoT device generates Laplacian noise according to ε and sensitivity Δf and adds it to his data set. Finally, each IoT device uploads the noisy data set to the fog node. The Laplace distribution (centred at 0) with scale b is the distribution with probability density function :

$$Lap(x|b) = \frac{1}{2b} exp(-\frac{|x|}{b}) \tag{1}$$

and the noise is independently identically distribution random variable drawn from $\Delta f/\varepsilon$. After uploading, IoT devices can be offline without having to occupy limited computing resources for training tasks. The IoT device's sensitivity is held locally by the device without informing other devices and fog node.

D. Fog Node Training

Algorithm 1 Training and Privacy Processing in Fog Node

Input: global model for *(i-1)th* round $[\omega^{i-1}]_{pk}$, the number of fog nodes *n*, upper limit of training rounds *epoch*.

Output: blind encrypted model parameters $\eta_n^i * \left[\omega_n^i\right]_{pk}$.

1: **for** i $\leq$ *epoch* **do**
2: Download global model parameters for this round $[\omega^{i-1}]_{pk}$.
3: Decrypt the above ciphertexts using the secret key *sk* to obtain $\frac{\omega^{i-1}}{n}$.
4: Get a mini-batch of data from its local dataset.
5: Train local model by using the BP algorithm based on SGD:

$$\omega_n^i := \frac{\omega_n^{i-1}}{n} - \alpha \frac{\delta E_n^i}{\delta \frac{\omega^{i-1}}{n}}$$

6: Encrypt and blind model parameter: $\eta_n^i * \left[\omega_n^i\right]_{pk}$.
7: Upload $\eta_n^i \left[\omega_n^i\right]_{pk}$ to the parameter server.
8: **end for**

We notice the fact that the earlier the training starts, the more epochs will be trained and the more accurate the training will be. Thus, when current amount of data is enough for training, the fog node will be active and dedicated training tasks. The training and privacy processing in fog node is provided in Algorithm 1. Each fog node downloads the initial model and collects IoT device nodes data with Laplacian noise.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2987958, IEEE Internet of Things Journal

When a fog node receives noisy data, it will get a mini-batch of data from its local dataset and compute gradients. They use the Back-Propagation (BP) algorithm based on stochastic gradient descent (SGD) to train the model:

$$\omega_n^i := \omega_n^{i-1} - \alpha \frac{\delta E_n^i}{\delta \omega_n^i} \tag{2}$$

where α stands for the learning rate and E be the error function. The resulting model parameter vector in the *ith* round is defined as ω_n^i. After training the model, the fog node first uses the *pk* to encrypt the parameters. Then, it blinds the ciphertext by the blinding factor η_n^i. The blind factor η_n^i and the *sk* of the fog node will never be disclosed to any entity (other fog nodes and parameter server). Finally, the fog node uploads $\eta_n^i \left[\omega_n^i\right]_{pk}$ to the parameter server.

E. Aggregation of Model Parameters

The aggregation of model parameters in server is provided in Algorithm 2. After receiving all model parameters from fog nodes, the parameter server performs the following aggregation operation:

$$\begin{aligned}[\omega_{sum}^i]_{pk} &= \eta_1^i[\omega_1^i]_{pk} \cdot \eta_2^i[\omega_2^i]_{pk} \cdot \ldots \cdot \eta_n^i[\omega_n^i]_{pk} \\ &= \prod_{j=1}^{n} \eta_n^i \cdot ([\sum_{j=1}^{n} \omega_n^i]_{pk}) \\ &= z^i \cdot [\sum_{j=1}^{n} \omega_n^i]_{pk}\end{aligned} \tag{3}$$

Algorithm 2 Aggregation and Update of Models in Sever

Input: blind encrypted model parameters of all fog nodes $\eta_n^i * \left[\omega_n^i\right]_{pk}$, upper limit of training rounds *epoch*.

Output: global model for *(i)th* round $[\omega^i]_{pk}$ and new set of blinding factors η_n^{i+1}.

1: **for** i $\leq$ *epoch* **do**
2: **Parameter Server:**
3: Receive $\eta_n^i \left[\omega_n^i\right]_{pk}$.
4: Aggregate model parameters:

$$\begin{aligned}[\omega_{sum}^i]_{pk} &= \eta_1^i[\omega_1^i]_{pk} \cdot \eta_2^i[\omega_2^i]_{pk} \cdot \ldots \cdot \eta_n^i[\omega_n^i]_{pk} \\ &= z^i \cdot [\sum_{j=1}^{n} \omega_n^i]_{pk}\end{aligned}$$

5: Compute the new global model: $[\omega^{i+1}]_{pk}=\frac{[\omega_{sum}^i]_{pk}}{z^i}$.
6: Send $[\omega_{sum}^i]_{pk}$ to fog nodes.
7:
8: **Blinding Server:**
9: Generate a new set of blinding factors η_n^{i+1}.
10: Send η_n^{i+1} to each fog node.
11: Iteration round $i = i + 1$
12: **end for**

The parameter server can get the $\frac{[\omega_{sum}^i]_{pk}}{z^i}$ as the aggregation global model by Paillier homomorphic properties. Then all fog nodes can download the global model.

The blinding server generates a new set of blinding factors η_n^{i+1} and finally delivers them to each fog node, which means the beginning of the second iteration. This iterative process will continue until the number of iterations is reached.

V. Security Analysis

In this section, the theoretical analysis is carried out on our proposed scheme, which provides the guarantee of data security, model security as well as anti-collusion attacks.

A. Data Security

In our scheme, IoT devices upload data to fog nodes. But a malicious fog node may obtain privacy information of IoT devices from the data it collects. Then we would prove that the collected data satisfies ε-differential privacy, thus our scheme could defend against the **data attackers**.

Theorem 1. *In our scheme, the IoT device data of each fog node is satisfied with ε-differential privacy.*

Proof: According to the sequence composition of differential privacy: let $\mathcal{M}_1$, $\mathcal{M}_2$, ..., $\mathcal{M}_n$ be n mechanisms, where each mechanism $\mathcal{M}_i$ (i∈[1,n]) provides ε-differential privacy. Let D_1,D_2,...,D_n be n arbitrary disjoint data sets of the input domain D. For a new mechanism $\mathcal{M}$, the sequence of $\mathcal{M}(\mathcal{M}_1(D_1), \mathcal{M}_2(D_2),...,\mathcal{M}_n(D_n))$ provides $\sum_{i=1}^{n}$ ε-differential privacy. This is called the sequence composition of differential privacy.

In our scheme, the fog node allocates the privacy budget ε to each IoT device according to the IoT device numbers. In order to facilitate calculation, we uniformly allocate $\frac{\varepsilon}{m}$ to each user, where m is the number of users. Users generate Laplacian noise according to their sensitivity Δf and $\frac{\varepsilon}{m}$ and add it to the data set to satisfy the following requirements:

$$Pr[\mathcal{M}(I) \in \mathcal{O}] \leq e^{\frac{\varepsilon}{m}} \cdot Pr[\mathcal{M}(I') \in \mathcal{O}] \tag{4}$$

in which I represent all adjacent inputs, and $\mathcal{O}$ represent all possible output. According to sequence composition, we can see that the mechanism of fog node provides $\sum \frac{\varepsilon}{m} = \varepsilon$-differential privacy, because data sets of different IoT devices may intersect. Therefore, we confirm that the data attackers cannot steal privacy information in our scheme.

B. Model Security

A malicious parameter server may obtain the model parameters of a specific fog node, and then infer its training data by GAN attack [21], model inversion attack [22], and so on. Next, we would demonstrate that the **model attacker** can only obtain the global model after aggregation. Even if he initiates such an attack, he can not obtain the data characteristics of the IoT devices under the specific fog node.

Theorem 2. *In our scheme, the parameter server will never infer the training set information from the global model parameters.*

Proof: Since the model of federated learning is trained by all users, the security of the model is not guaranteed from the server's point of view. An attacker or a malicious user can easily intercept the model plaintext, which leads to

model information disclosure. At present, there are inference attacks, such as GAN attack [21], model inversion attack [22] and so on, which can obtain the training set information from the trained model interface through testing methods. Moreover, there is an obvious security risk that the parameters uploaded by IoT devices are plaintext in each round of model uploading process, which is easy to be intercepted by attackers, resulting in model information leakage and loss. However, in our scheme, the server can only get the ciphertext of the global parameters $[\omega^i]_{pk}$ in the end. In the ciphertext, the condition of the inference attack is destroyed, so we can consider that our scheme can resist the inference attacks. Because in general, parameter servers will never get the content of global model parameters unless collusion occurs. Therefore, we can believe that our scheme can resist a series of model attacks against specific fog nodes to meet the model security.

C. Anti-Collusion Attacks

In this section, we analyze the impact of collusion attacks on our scheme. As described in subsection III.B, collusion may occur between the parameter server and fog node, or between fog node and fog node (blinding server is trusted and will not participate in collusion).

Theorem 3. *Our scheme is secure in the semi-honest model, even if the malicious fog node F_{mal} colludes with k ($k \leq n-2$) fog nodes.*

Proof: Suppose that there are *k* ($k \leq n-2$) fog nodes which collude with the malicious fog node F_{mal}. Let *W* $=\{\omega_1^i, \omega_2^i, ..., \omega_k^i\}$ denote the set of gradients with colluding with F_{mal}. Let $\eta = \{\eta 1, \eta 2, ..., \eta_k\}$ denote the set of gradients with colluding with F_{mal}. All information F_{mal} can obtain is *W* and η. Thus, if F_{mal} wants to get model parameter of F_x, for $x \notin k$, F_{mal} can learn it from :

(1) F_{mal} obtains ω_1^{i+1} by downloading from blinding server. So F_{mal} can easily compute the aggregation of all fog nodes' model parameters in the previous round: $\omega^{i+1} \cdot n$. Then model parameter of F_x can only be obtained in the following ways: $\omega_x^i = \omega^{i+1} \cdot n - \omega_1^i - \omega_2^i - ... - \omega_{x-1}^i - \omega_{x+1}^i - ... - \omega_n^i$. So if F_{mal} cannot collude with all fog nodes except F_x, then the model parameters of F_x will never be revealed from ω^{i+1}.

(2) F_{mal} obtains $\eta_x^i \cdot [\omega_x^i]_{pk}$ by monitoring channels, and model parameter of F_x can only be obtained in the following ways: $\omega_x^i = \text{dec}(\frac{\eta_x^i \cdot [\omega_x^i]_{pk}}{z^i / (\eta_1 \cdot \eta_2 \cdot ... \eta_{x-1} \cdot \eta_{x+1} \cdot ... \cdot \eta_n)})$. So if F_{mal} cannot obtain z^i from blinding server, then the model parameters of F_x will never be revealed from $\eta_x^i \cdot [\omega_x^i]_{pk}$.

Theorem 4. *Our scheme is secure in the semi-honest model, even if the parameter server colludes with k ($k \leq n-2$) fog nodes.*

Proof: Suppose that there are *k* ($k \leq n-2$) fog nodes which collude with the parameter server. The purpose of parameter server is the model parameter of F_i. Thus, the parameter server can compute:

$$\frac{\eta_1[\omega_1]_{pk} \cdot \eta_2[\omega_2]_{pk} \cdot ... \cdot \eta_n[\omega_n]_{pk}}{\eta_1[\omega_1]_{pk} \cdot ... \cdot \eta_{i-1}[\omega_{i-1}]_{pk} \cdot \eta_{i+1}[\omega_{i+1}]_{pk} \cdot ... \cdot \eta_{n-1}[\omega_{n-1}]_{pk}} = \eta_i[\omega_i]_{pk} \cdot \eta_n[\omega_n]_{pk} \quad (5)$$

TABLE I: Two practical IoT scenarios

Scenarios	Fog nodes	IoT devices	Data volume
Smart Homes	$\leq$10	50-100	500-1000
Smart Healthcare	10-20	200-400	2000-4000

We can see that if the parameter server colludes with k ($k \leq n-2$) fog nodes, the model parameters of F_x will never be revealed from ω^i.

VI. Experiment

In this section, we conduct extensive experiments on the representative dataset in different aspects to evaluate the performance of our scheme.

A. Experimental Setup

Our experiments are carried out by using one desktop computer as fog nodes and Alibaba Cloud as the parameter server and the blinding server. In our experimental environment, we use Python to simulate multiple IoT device nodes. The desktop computer has an identical configuration: Intel Core i5 (64 bit) processors for a total of 4 cores running at 3.20GHz and 12GB RAM. The specification of Alibaba Cloud is ecs.g5.4xlarge, with identical configuration: Intel Xeon Platinum 8163 processors for a total of 16 cores running at 2.5GHz and 64GB RAM. Considering the computing overhead, we use 64-bit Paillier encryption keys.

To evaluate our scheme, we simulate two types of application scenarios. Note that the scenarios present different data distribution. Thus, in order to make the experiment more representative, we choose a small-scale data distribution scenario (i.e., Smart Homes) and a large-scale data distribution scenario (i.e., Smart Healthcare). The data distributions are shown in Table I. The characteristics of Smart Homes are that the data set is relatively scattered and the total amount of data is small, while the data of Smart Healthcare is relatively centralized and the total amount is large.

```
model=Sequential()
Self.img_shape = (28, 28, 1)
model.add(Flatten(input_shape=img_shape))
model.add(Dense(512))
model.add(LeakyReLU(alpha=0.2))
model.add(Dense(1024))
model.add(LeakyReLU(alpha=0.2))
model.add(Dense(256))
model.add(LeakyReLU(alpha=0.2))
model.add(Dense(10, activation='sigmoid'))
```

Fig. 4: Neural network architecture (with 784 inputs).

The dataset we use is Fashion-MNIST, which is a dataset of Zalando's article images—consisting of a training set of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2987958, IEEE Internet of Things Journal

TABLE II: Layers and parameter number in our model

Layer (type)	Output Shape	Parameter number
dense_5 (Dense)	(256)	25856
leaky_re_lu_4 (LeakyReLU)	(256)	0
batch_normalization_1 (Batch)	(256)	1024
dense_6 (Dense)	(512)	131584
leaky_re_lu_5 (LeakyReLU)	(512)	0
batch_normalization_2 (Batch)	(512)	2048
dense_7 (Dense) (Batch)	(1024)	525312
leaky_re_lu_6 (LeakyReLU) (Batch)	(1024)	0
batch_normalization_3 (Batch)	(1024)	4096
dense_8 (Dense) (Batch)	(784)	803600
reshape_1 (Reshape)	(28, 28, 1)	0

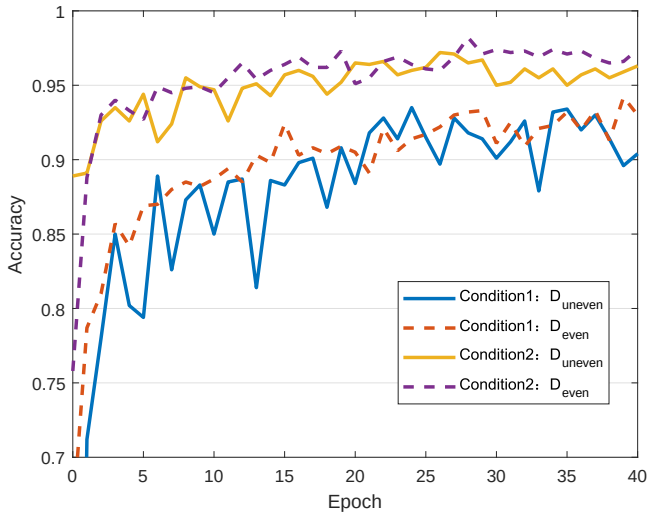


Fig. 5: Accuracy comparison between uneven data D_{uneven} and even data D_{even}.

60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. We represent each image as a column vector by reading the image gray value column by column and normalize the gray value vector as the input of the fully connected neural network (FCNN). We implemented our scheme based on FCNN and the structure of it is shown in Figure 4 and trained model details are shown in Table II.

B. Influence of Unevenly Distributed Data on Training

We carried out two kinds of experiments: the accuracy of the model under different data distribution and the training time of the model under different data distribution.

(1) Model Accuracy under Different Data Distribution

Figure 5 is the accuracy varying with the number of training rounds between uneven data D_{uneven} and even data D_{even} under two different data distributions. The data distribution of condition 1 is less discrete than that of condition 2. From the results, we can see that when the total amount of data is the same, the even distribution of data can achieve higher model accuracy and convergence speed than the uneven distribution, and the convergence process is smoother. Experimental results demonstrate the impact of data distribution on the model accuracy and training efficiency. That is, the more uneven the data distribution is, the lower the efficiency of model training is. This is because our scheme introduces fog computing, where the IoT devices upload data to fog nodes and transform the uneven data distributions into the even ones. Therefore, the three-tier architecture of our scheme is a quite suitable way that can adjust the overall data distribution and improve the model accuracy training efficiency.

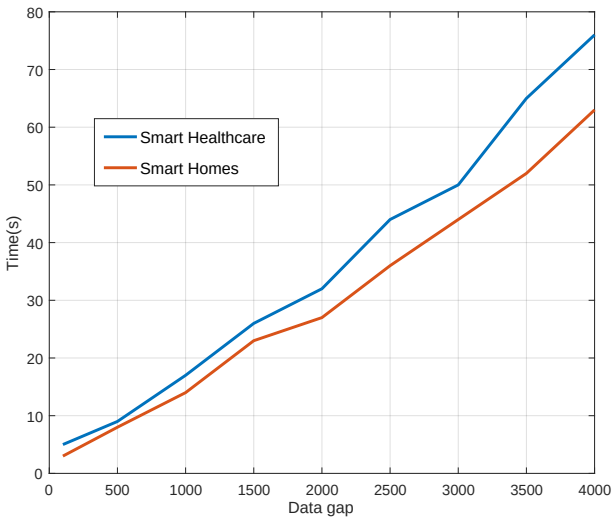


Fig. 6: Influence of data gap on training time.

(2) Training Time under Different Data Distribution

Figure 6 shows the training time gap caused by the data gap of the nodes we evaluated. From the figure, we can see that the data gap and the training time gap are linear in both scenarios. The larger the data gap is, the more the training time gap is. Therefore, the lag of training time caused by uneven data distribution cannot be ignored. From these experimental results, we can know that the uneven data distribution will not only cause the prolongation of training time, but also the convergence rate of the model is slow, unstable, and the final accuracy is low. Because the nodes with fewer data have to wait for the end of the node with more data. Additionally, both scenarios have the similar experimental result, which demonstrates that our view can be suitable in various data distributions. Therefore, we can conclude that our scheme can improve the training efficiency by selecting fog node to collect the data of IoT device nodes for training.

C. Accuracy

We compared the accuracy of Federated Learning [18] and our scheme. Then we evaluated the impact of the privacy budget on model accuracy.

(1) Model Accuracy Comparison

We compare our scheme with Federated Learning [18] in two scenarios in Figure 7. We use SGD for optimization, which the learning rate is 0.001. The training epoch is 18,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2987958, IEEE Internet of Things Journal

Fig. 7: Accuracy comparison between our scheme and Federated Learning [18].

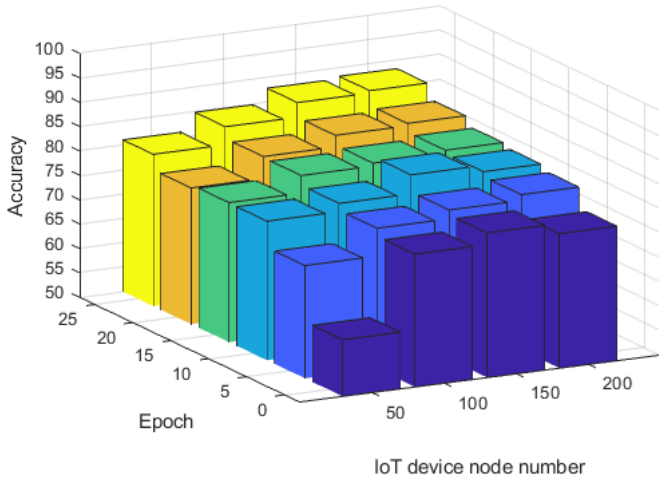


Fig. 8: Relationship between model accuracy and the number of IoT device nodes.

and each batch selects the appropriate number of samples according to the scenarios. We can see that our scheme has similar accuracy to Federated Learning [18] in both scenarios, which demonstrates that our scheme cannot have the impact of differential privacy, blinding, and Paillier homomorphic encryption on model accuracy. In our scheme, the model parameters will be decrypted and their blinding will be removed at the parameter server, so that the model accuracy would not reduce by the impact of blinding and encryption. Moreover, differential privacy also has little influence on accuracy due to the noise added to data.

Figure 8 depicts the relationship among the number of IoT device nodes, the number of training epochs, and the accuracy rate of the model. We set that each node has the same number of data. From the figure, we can see that the different number of nodes first has a direct impact on the convergence rate of the model. The more nodes, the faster the convergence rate of the model, the higher the upper limit of the final model accuracy rate. It makes the convergence curve smooth and the accuracy of the model stable. In addition, it indicates that the model can converge fast in our scheme and show good performance in different IoT devices nodes, which also proves the practicability of our scheme.

(2) Model Accuracy under Different Privacy Budgets

We have evaluated the accuracy changes of the model under different privacy budgets. It can be seen from Figure 9 that the smaller the ε is, the greater the added Laplacian noise is, so as to obtain higher privacy requirements. But at the same time, the accuracy of the model will be reduced, which is the trade-off between privacy and data availability. It is straightforward that, $\varepsilon > 0.1$ is a reasonable privacy budget when the accuracy of the model remains within the acceptable range. Moreover, as shown in Figure 9 that our scheme does not lead to accuracy loss due to the differential privacy. This is because other works add Laplacian noise to gradients so that the accumulation of noise in each epoch may reduce the accuracy of the model. In contrary, our scheme adds the one-time noise to the data, and the situation mentioned above would not happen in the training process. Thus, our scheme can maintain model accuracy while satisfies data security.

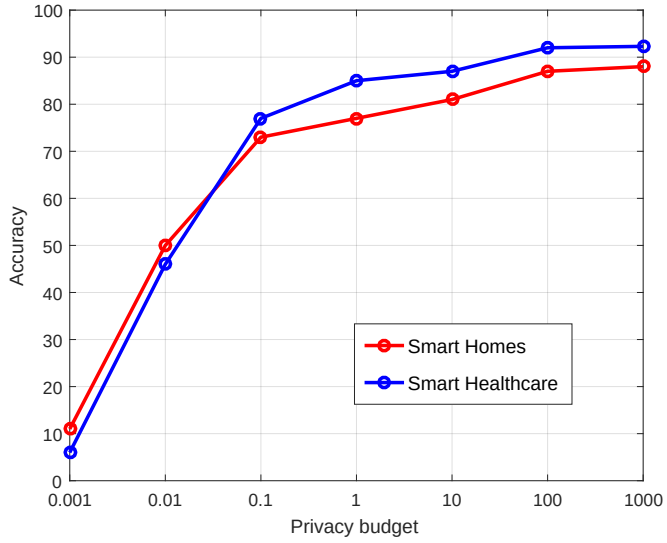


Fig. 9: Model accuracy under different privacy budgets.

D. Computing overhead

Figure 10 presents the relationship between the time of each operation and the number of training epochs in our scheme, which is mainly divided into four parts: encryption, decryption, blinding, and aggregation. It is clear from Figure 10 that encryption and decryption take up most of the time in each epoch, and the aggregation and blinding time of ciphertext is within 2 seconds. With the training goes on, due to storage and memory reasons, the occupation time of each operation increases a little.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2987958, IEEE Internet of Things Journal

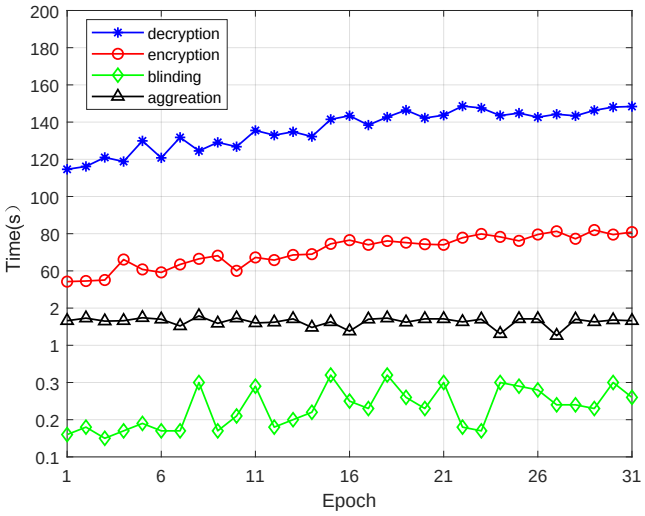


Fig. 10: Time consumption of each operation in our scheme.

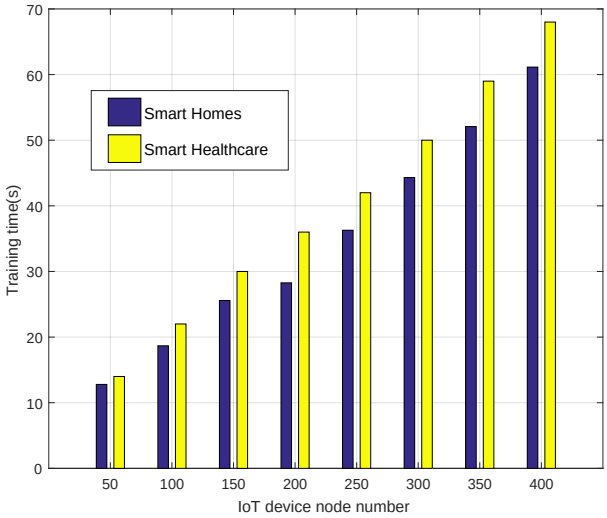


Fig. 11: Relationship between training time and number of IoT device nodes.

Figure 11 displays the relationship between the number of IoT device nodes and training time. We increase the number of IoT device nodes by 50 nodes in turn and evaluate the training time in both scenarios, respectively. From the figure, it is evident that with the increase in the number of IoT device nodes, the training time of the model also increases. Moreover, Smart Homes has less training time than Smart Healthcare because of the fewer data. However, the difference between two scenarios is not huge. The reason is that the computing overhead of our scheme is encryption and decryption mainly, and they would not raise with the increase of IoT device nodes.

Figure 12 appears that the operation time and model accuracy when setting a different number of fog nodes in our scheme. We set $\{3, 4, 5, 6, 7, 8, 9\}$ fog nodes respectively, and set each situation to have the same number of IoT device nodes. It is evident from the experimental results that the time of encryption and decryption is longer than the training

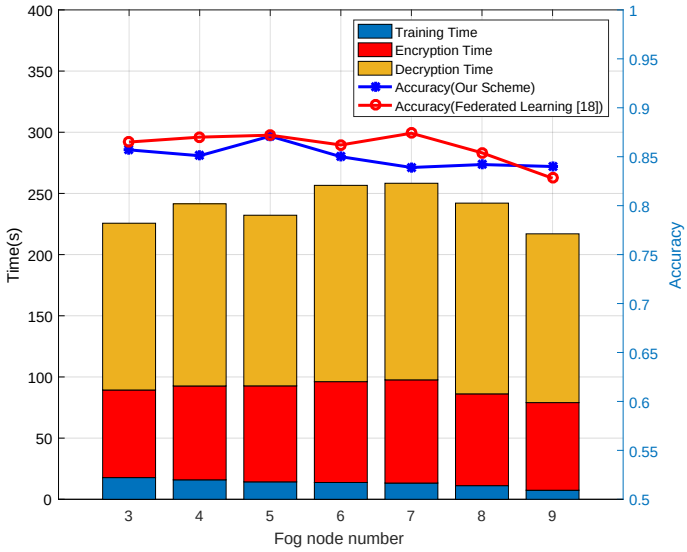


Fig. 12: Operation time and model accuracy under different number of fog nodes.

time (the blinding time is generally less than 1s, which is ignored here). Due to the complexity of the network structure, there are nearly 1.4 million parameters. In order to ensure model parameters security, we sacrifice part of the efficiency, which is a trade-off between privacy and efficiency. Although encryption and decryption increase the calculation cost, it has less impact on the accuracy of the model. As can be seen from the figure, model accuracy in our scheme is similar to that in plaintext training [18]. For some training cases that need to ensure security, it is worth sacrificing some time. Moreover, with the increase of the number of fog nodes, the training time of the model is significantly reduced, and the time of encryption and decryption will lead to a definite time difference due to a large number of parameters. However, the overall trend of model accuracy will gradually decrease with the increase in the number of fog nodes. Because the degree of data dispersion is too large, the amount of data is not enough to train the high accuracy model locally. From the figure that five fog nodes are the optimal allocation mode of IoT device nodes evidently, which has the highest model accuracy and lower operation time.

Finally, we compare our computing overhead with other schemes: Federated Learning [18] and Phong's Scheme [23], in the aspect of encryption and decryption time, training time, transmission time, and so on, which indicates in Table III. The time spent on ciphertext operations (encryption, decryption, aggregation) at every 300,000 gradients.

It is noticeable from TABLE III that the model training time of our scheme is similar to Federated Learning [18]. The increase of aggregation time and upload/download time of model parameters is due to the operation of ciphertext parameters. The blinding time of $23ms$ can be negligible. Compared with Phong's Scheme [23], our scheme has a faster training speed and a tremendous advantage in model aggregation time and encryption and decryption time. It is worth mentioning that our communication overhead is higher than that of Phong's

TABLE III: Computing overhead of our scheme

Computing Task	**Scheme [18]**	**Scheme [23]**	**Our Scheme**
Training	178ms	276ms	124ms
Encryption+Decryption	———	4028.1ms	2179.1ms
Blinding	———	———	23ms
Aggregation	6.6ms	481ms	163ms
Upload/Download	50.9ms	240ms	241ms
Communication	0.44MB	1.0MB	1.96MB

Scheme [23], because it uses 32-bit encryption, while we use 64-bit encryption. Thus the communication overhead increases nearly twice. In general, our scheme is more suitable for the uneven data distribution in practice than other works, and has lower training time and computing overhead.

In conclusion, the experimental results demonstrate that our scheme is suitable for various scenarios. Our scheme can be effective in terms of training time, and maintain similar accuracy compared with federated learning. In addition, it has a lower computing overhead than other works.

VII. Conclusions

In this paper, we proposed a privacy-preserving federated learning scheme in fog computing. Our design combinates centralized and distributed training modes to make it more practical, which can also solve the problem of inefficient training caused by the significant gap among IoT devices in the amount of data and computing power. Moreover, we take advantage of differential privacy, blinding, and Paillier homomorphic encryption to equip our scheme with the ability to resist data attacks, model attacks, and collusion attacks. On the one hand, we can protect against a privacy revealing fog node or a malicious parameter server. On the other hand, we can also prevent multiple malevolent entities from inferring the model parameters of a specific fog node. Security analysis demonstrates that our scheme can guarantee both data security and model security, and completely resist collusion attacks. Our experiments simulate two scenarios based on the Fashion-MNIST dataset and show that our scheme is efficient in dealing with multiple data distributions. Especially, with the same accuracy and low computing overhead, our scheme can greatly save training time compared with existing works.

Our future work can be devoted to improving the efficiency of our scheme and reducing the computational cost, which makes it more practical and maintains security.

References

[1] B. Kuang, A. Fu, S. Yu, G. Yang, M. Su and Y. Zhang, "Esdra: An efficient and secure distributed remote attestation scheme for iot swarms," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8372-8383, 2019.

[2] A. Rachedi and A. Benslimane, "Multi-objective optimization for Security and QoS adaptation in Wireless Sensor Networks," in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 1-7, 2016.

[3] S. Yu, G. Wang, X. Liu and J. Niu, "Security and Privacy in the Age of the Smart Internet of Things: An Overview from a Networking Perspective," *IEEE Communications Magazine*, vol.56, no. 9, pp. 14-18, 2018.

[4] A. L. Diedrichs, F. Bromberg, D. Dujovne, K. Brun-Laguna and T. Watteyne, "Prediction of Frost Events Using Machine Learning and IoT Sensing Devices," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4589-4597, 2018.

[5] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "ContextAware Computing, Learning, and Big Data in Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 1-27, 2018.

[6] A. Kaci, T. Bouabana-Tebibel, A. Rachedi and C. Yahiaoui, "Toward a big data approach for indexing encrypted data in Cloud Computing," *Security and Privacy*, vol.2, issue. 3, 2019.

[7] R. Yu, G. Xue and X. Zhang, "Application Provisioning in FOG Computing-enabled Internet-of-Things: A Network Perspective," in *Proc. of International Conference on Computer Communications (INFOCOM)*, pp. 783-791, 2018.

[8] G. Xu, H. Li, H. Ren, K. Yang and R. H. Deng, "Data Security Issues in Deep Learning: Attacks, Countermeasures, and Opportunities," *IEEE Communications Magazine*, vol.57, issue. 11, pp. 116-122, 2019.

[9] R. Shokri, V. Shmatikov, "Privacy-preserving deep learning," in *Proc. of ACM SIGSAC Conference on Computer and Communications Security. (CCS)*, pp. 1310-1321, 2015.

[10] A. Fu, Z. Chen, Y. Mu, W. Susilo, Y. Sun and J. Wu, "Cloud-based Outsourcing for Enabling Privacy-Preserving Large-scale Non-Negative Matrix Factorization," *IEEE Transactions on Services Computing*, DOI: 10.1109/TSC.2019.2937484, 2019.

[11] Q. Yang, Y. Liu, T. Chen and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, Article. 12, 2019.

[12] A. Rachedi and A. Benslimane, "Security and Pseudo-Anonymity with a Cluster-based approach for MANET," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1-6, 2008.

[13] T. Li, Z. Huang, P. Li, Z. Liu, and C. Jia, "Outsourced privacy-preserving classifcation service over encrypted data," *Journal of Network and Computer Applications*, vol. 106, pp. 100-110, 2018.

[14] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar *et al.*, "Deep Learning with Differential Privacy," in *Proc. of ACM SIGSAC Conference on Computer and Communications Security. (CCS)*, pp. 308-318, 2016.

[15] Y. Rahulamathavan, R. C.-W. Phan, S. Veluru, K. Cumanan and M. Rajarajan, "Privacy-Preserving Multi-Class Support Vector Machine for Outsourcing the Data Classification in Cloud," *IEEE Transactions On Dependable And Secure Computing*, vol. 11, no. 5, pp. 467-479, 2014.

[16] L. Xiang, J. Yang, B. Li, "Differentially-Private Deep Learning from an optimization Perspective," in *Proc. of International Conference on Computer Communications (INFOCOM)*, pp. 559-567, 2019.

[17] C. Zhang, L. Zhu, C. Xu and R. Lu, "PPDP: An efficient and privacy-preserving disease prediction scheme in cloud-based e-Healthcare system," *Future Generation Computer Systems*, vol.79, pp. 16-25, 2017.

[18] H. B. McMahan, E. Moore, D. Ramage, and B. A. Arcas, "Federated learning of deep networks using model averaging," *arxiv:1602.05629*, 2016.

[19] V. Smith, C. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated Multi-Task Learning," in *Proc. of Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 4424-4434, 2017.

[20] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan and S. Patel *et al.*, "Practical Secure Aggregation for Privacy-Preserving Machine Learning,"in *Proc. of ACM SIGSAC Conference on Computer and Communications Security. (CCS)*, pp. 1175-1191, 2017.

[21] B. Hitaj, G. Ateniese and F. Pérez-Cruz, "Deep models under the GAN: information leakage from collaborative deep learning," in *Proc. of ACM SIGSAC Conference on Computer and Communications Security. (CCS)*, pp. 603-618, 2017.

[22] M. Fredrikson, S. Jha and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. of ACM SIGSAC Conference on Computer and Communications Security. (CCS)*, pp. 1322-1333, 2015.

[23] L. Phong, Y. Aono, T. Hayashi, L. Wang and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333-1345, 2018.

[24] L. Melis, C. Song, E.D. Cristofaro and V. Shmatikov, "Exploiting Unintended Feature Leakage in Collaborative Learning," in *Proc. of 40th IEEE Symposium on Security & Privacy (S&P)*, 2019.

[25] R. Bost, R. A. Popa, S. Tu, S. Goldwasser, "Machine Learning Classification over Encrypted Data," in *Proc. Annual Network and Distributed System Security Symposium. (NDSS)*, 2015.

[26] Q. Wang, M. Du, X. Chen, Y. Chen, P. Zhou and X. Chen *et al.*, "Privacy-preserving collaborative model learning: The case of word vector training," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2381-2393, 2018.

[27] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private Naive Bayes learning over multiple data sources," *Information Sciences*, vol. 444, pp. 89–104, 2018.

[28] C. Xu, J. Ren, D. Zhang, Y. Zhang, Z. Qin and K.Ren, "GANobfuscator: Mitigating Information Leakage under GAN via differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2358-2371, 2019.

[29] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *Proc. of IEEE Symposium on Security & Privacy (S&P)*, pp. 19-38, 2017.

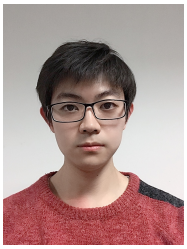
[30] Q. Jia Q, L. Guo, Z. Jin and Y. Fang, " Preserving model privacy for machine learning in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol.29, no. 8, pp. 1808-1822, 2018.

[31] B. Hitaj, G. Ateniese and F. Pérez-Cruz, "Deep models under the GAN: information leakage from collaborative deep learning," in *Proc. of ACM SIGSAC Conference on Computer and Communications Security. (CCS)*, pp. 603-618, 2017.

[32] L. Zhou, A. Fu, S. Yu, M. Su and B. Kuang, "Data Integrity Verification of the Outsourced Big Data in the Cloud Environment: A Survey," *Journal of Network and Computer Applications*, vol. 122, pp. 1-15, 2018.

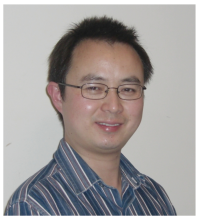
[33] G. Xu, H. Li,S. Liu, K. Yang and X. Lin, "VerifyNet: Secure and Verifiable Federated Learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911-926, 2019.

[34] H. Wang, Z. Wang and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 712-719, 2018.



Chunyi Zhou is currently a Ph.D. student in School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He received his Bachelor degree in Computer Science and Technology from Nanjing University of Science and Technology, China, in 2017. His research interest includes Machine Learning Security and Privacy Preserving.

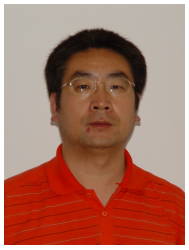
Anmin Fu received the Ph.D. degree in Information Security from Xidian University in 2011. From 2017 to 2018, he was a Visiting Research Fellow with the University of Wollongong, Australia. He is currently an associate professor and supervisor of Ph.D. students of Nanjing University of Science and Technology, China. Dr Fu's research interest includes IoT Security, Cloud Computing Security and Privacy Preserving. He has published more than 50 technical papers, including international journals and conferences, such as IEEE Transactions on Big Data, IEEE Transactions on Services Computing, IEEE Transactions on Vehicular Technology, IEEE Internet of Things Journal, IEEE communications letters, Information Sciences, Journal of Network and Computer Applications, Computers & Security, IEEE ICC, IEEE GLOBECOM and ACISP.



Shui Yu is currently a full Professor of School of Software, University of Technology Sydney, Australia. Dr Yu's research interest includes Security and Privacy, Networking, Big Data, and Mathematical Modelling. He has published two monographs and edited two books, more than 200 technical papers, including top journals and top conferences, such as IEEE TPDS, TC, TIFS, TMC, TKDE, TETC, ToN, and INFOCOM. He actively serves his research communities in various roles. He served IEEE Transactions on Parallel and Distributed Systems as an AE (2013-2015), and is currently serving the editorial boards of IEEE Communications Surveys and Tutorials (exemplary editor for 2014), IEEE Access, IEEE Internet of Thing Journal, IEEE Communications Letters (exemplary editor for 2016), and a number of other international journals. Moreover, he has organized several Special Issues either on big data or cybersecurity. He has served more than 70 international conferences as a member of organizing committee, such as publication chair for IEEE Globecom 2015 and IEEE INFOCOM 2016 and 2017, TPC co-chair for IEEE BigDataService 2015, IEEE ITNAC 2015, and General chair for ACSW 2017. He is a Senior Member of IEEE, a member of AAAS and ACM, the Vice Chair of Technical Committee on Big Data of IEEE Communication Society, and a Distinguished Lecturer of IEEE Communication Society.

Wei Yang received the Ph. D degree in University of Chinese Academy of Sciences in January 2018. He is now a lecturer of Nanjing University of Science and Technology. His main research interests include side-channel attacks and countermeasures.

Huaqun Wang received the B.S. degree in mathematics education from the Shandong Normal University and the M.S. degree in applied mathematics from the East China Normal University, both in China, in 1997 and 2000, respectively. He received the Ph.D. degree in Cryptography from Nanjing University of Posts and Telecommunications in 2006. Now, he is a professor in Nanjing University of Posts and Telecommunications. His research interests include applied cryptography, network security, and cloud computing security.



Yuqing Zhang received the Ph.D. degree in cryptography from Xidian University, Xi'an, China. He is a Professor and the Director of National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing, China. He has authored or co-authored over 100 research papers in international journals and conferences, such as the ACM CCS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. His research has been sponsored by NSFC, Huawei, Qihu360, and Google. His current research interests include network and system security and applied cryptography.