2024 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics | 979-8-3503-5163-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics62450.2024.00059

A Framework for Privacy-Preserving Efficient Collaborative Learning

Jianxiang Cao
State Key Laboratory of Media Convergence and Communication
Communication University of China
Beijing, China
jxcao@126.com

Xing Song
State Key Laboratory of Media Convergence and Communication
Communication University of China
Beijing, China
songxing9903@163.com

Wenqian Shang
State Key Laboratory of Media Convergence and Communication
Communication University of China
Beijing, China
shangwenqian@cuc.edu.cn

***Abstract*—In recent years, federated learning has been widely used in deep learning training tasks such as image recognition. The federated learning mechanism allows multiple participants to collaboratively train a common model without having to aggregate data. However, participants may infringe on the privacy of data owners when collecting personal data. And when there are significant differences in data distribution among participants, traditional federated learning methods may yield less than ideal training results. To solve these problems, we propose an efficient privacy-preserving collaborative learning framework (EPPCL), which provides secure access authorization and training for personal data. While introducing an authorization mechanism, we design an efficient collaborative training method that can improve the model performance in cases of data heterogeneity. Furthermore, we evaluate the system performance of EPPCL. The experimental results demonstrate that EPPCL is a secure and effective way to achieve multi-party collaborative training for personal data.**

Keywords*—*privacy preserving, federated learning, encryption

I. Introduction

With the application of the Internet of Things (IoT) across various fields, IoT devices have reached a certain scale worldwide. These devices can collect a large amount of data in a short period of time through real-time monitoring. Different devices produce different types of data, which can be used for analysis to optimize real-world problems [1]. The efficient analysis of IoT data benefits from the combination with deep learning. Due to their good performance in specific domains such as image recognition and suitability for handling large-scale data, deep learning models are widely deployed for analyzing IoT data [2]. However, to achieve more accurate results, large amounts of data are required in deep learning training. For traditional training approach of deep learning, data from individual devices are aggregated to a central server for centralized training. This approach poses significant privacy risks and brings new challenges for deep learning training [3].

The introduction of federated learning (FL) effectively solves the above issues [4]. Federated learning can be utilized for deep learning training without having to aggregate sensitive data together. When isolated data held by individual devices cannot produce ideal models, the federated learning mechanism allows different devices to jointly train a common model using their own data while ensuring the data security of each device [5]. In federated learning, multiple participants collaboratively train a common model, and each participant has full control over the data held [6]. However, in practical applications, there may be multiple data owners for the local data collected by participants, and each owner has different privacy requirements for his or her own data. Therefore, how to achieve collaborative training while protecting the privacy of data owners becomes a critical problem.

To protect the privacy of data owners, we need to change the way data is provided in traditional federated learning. We envision a new scenario: where data owners and trainers are not the same roles, trainers cannot directly acquire local data, and each data owner decides whether the personal data will be involved in federated learning. This realizes the separation of data ownership and usage rights. Therefore, in this paper, we construct a new federated learning framework, an efficient privacy-preserving collaborative learning framework, named EPPCL. In EPPCL, owners hold their personal data and authorize it to trainers through attribute-based encryption. This mechanism ensures that trainers no longer have control over the data, effectively reducing privacy risks. More importantly, we design an improved federated learning method in EPPCL that overcomes the limitations of federated learning in authorization scenarios. This method ensures that trainers can achieve efficient training while protecting privacy.

The rest of the paper is organized as follows. Section II introduces some preliminaries. Section III describes the system architecture of EPPCL, the proposed collaborative training method, and the construction of EPPCL in detail. Section IV presents the experimental results of the EPPCL system. Finally, conclusions are drawn in section V.

II. Preliminaries

In this section, some preliminaries used in our collaborative learning scheme are introduced.

A. Ciphertext Policy Attribute-Based Encryption

Ciphertext Policy Attribute-Based Encryption (CP-ABE) [7] allows users to encrypt and decrypt data based on specific access policies and attributes, enabling fine- grained access control for data. In this scheme, the access policy designated by the

2836-3701/24/$31.00 ©2024 IEEE

DOI 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics62450.2024.00059

encryptor is embedded into the ciphertext, while the user's key is associated with a set of attributes. Only when a user's attribute set matches the access policy can they decrypt the ciphertext associated with that policy. The classic scheme [7] comprises four algorithms: initialization, key generation, encryption, and decryption.

B. *Federated Learning*

Federated learning provides a safe collaborative training approach among multiple parties. By sharing local training results, participants jointly train an effective model. The process of the classic federated averaging method FedAvg [4] involves the following steps.

First, the central server initializes and distributes the global model parameters w_0 to each client. In each training round (t=1, 2, ...), each selected client k updates the local model using the current global model parameters w_t from the central server, trains the model on the local datasets based on the gradient descent method, and obtains the local model results w_{t+1}^k. These updates are then sent back to the central server. The server aggregates them using a weighted averaging method, updates the global model and distributes the updated parameters w_{t+1} for the next round of training. After a certain number of training rounds, the final global model is obtained.

C. *Knowledge Distillation*

Knowledge distillation generally transfers knowledge from complex teacher models to lightweight student models [8]. During this process, the student model learns from the teacher model's outputs based on the distillation dataset, and is trained using a linear combination of two loss functions with the following equation.

$$L = \lambda L_{CE}(q^S, y) + (1-\lambda) L_{KL}(q_\tau^S, q_\tau^T) \qquad (1)$$

In (1), L_{CE} represents the cross-entropy loss between the student model's predicted probabilities q and the true labels y. L_{KL} is the relative entropy loss between the output probability distributions of the student and the teacher model. Here, the temperature τ is used to adjusts the smoothness of the outputs. λ is used to denote the weights of L_{CE} and L_{KL}. By minimizing the total loss, the student model can learn the information from both original labels and the teacher model's probability distribution.

III. THE FRAMEWORK OF EPPCL

A. *System Architecture*

In this section, we propose an efficient privacy-preserving collaborative learning system, called EPPCL. EPPCL has a three-layer architecture, which includes data acquirement layer, data authorization layer and data training layer. The system architecture is shown in Fig. 1, and each layer is specifically described as follows.

Data Acquirement Layer. Before training data, data is initially collected from data providers. In this layer, a data provider consists of a group of data owners. A data owner possesses personal data that can be provided it to trainers to support their participation in subsequent training. To provide data to trainers securely, a data owner can select specific trainers as authorized users based on their willingness, and then upload the data in an encrypted form to the cloud server.

Data Authorization Layer. The data authorization layer mainly provides secure storage for encrypted data, granting users access permissions to further obtain authorized plaintext data. In this layer, the attribute authority within each organization (AA_i) as a trusted entity is responsible for managing and distributing user keys. Each cloud server CSP_i is responsible for storing the encrypted data of data owners associated with their respective organizations, and providing data access to trainers.

Data Training Layer. In this layer, to train a high- quality model, multiple trainers jointly participate in the machine learning training task. Each trainer does not possess any data themselves, but acquires authorized local data through the data authorization layer. Then trainers reach an agreement with the central server and train collaboratively based on the improved federated learning algorithm. Under the deployment of the central server, each trainer trains a local model using authorized data. Through multiple rounds of training, trainers collectively build a global model with superior performance.

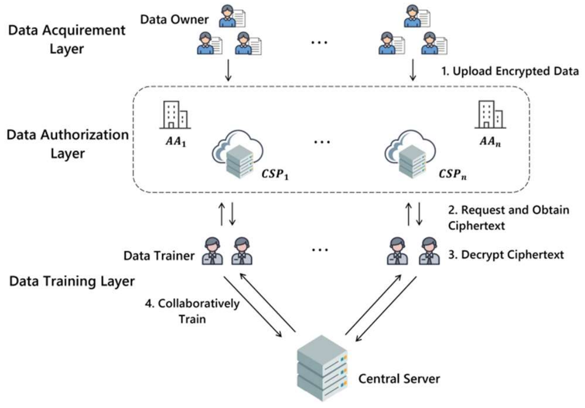


Fig. 1. System architecture of EPPCL

B. *Collaborative Training Method*

In the system architecture described in Section A, individual trainers in the training layer train models collaboratively through federated learning. Trainers collect training data according to the wishes of their respective users. Thus, the distribution of authorized data among trainers may vary significantly. In the case of data heterogeneity, when applying the federated averaging method, using global model parameters completely to guide local training may affect the performance improvement of local models on their respective datasets, thereby resulting in less than ideal training results. Therefore, a new collaborative training method is adopted instead in the training layer.

Collaborative Training Method. The improved federated learning algorithm serves as the collaborative training method. The training process is described as follows. First, the central server distributes the current global model parameters $global_i$ (or initialization parameters) to each trainer. Then, each trainer performs local distillation and training on the authorized data. For the t-th round of training, local distillation and training include the following two steps:1) Trainer k takes the global model $global_t$ as the teacher model and the local model $local_t^k$ as the student model, and performs local distillation of global knowledge using the local dataset D_k . By performing r_1 rounds of training using the loss function defined in formula (1), the local model $local_t^{k}{'}$ is obtained. 2) Trainer k further performs r_2 rounds of local training using the local dataset D_k to obtain the local model $local_{t+1}^k$. After local distillation and training have been completed, the updates of local model parameters from the trainers are uploaded to the central server. The central server updates the global model by aggregating multiple model parameter updates through weighted averaging. Each trainer receives the new global model parameters and proceeds to the next round of training. After performing the prescribed number of training rounds, the final global model is obtained.

C. System Workflow

The EPPCL system process mainly consists of four stages: initialization, data preparation, data acquisition, and collaborative training, as shown in Fig. 2, Fig. 3. The detailed descriptions of each stage are as follows.

Initialization. In this stage, we define n participating institutions P = $\{P_1, P_2, ..., P_n\}$. Each participating institution P_i has a corresponding attribute authorization center AA_i and a cloud server CSP_i. AA_i runs the initialization algorithm of CP-ABE to generate a public key PK_i and a master key MK_i, and provides the public parameter PK_i to data owners for encryption.

Data Preparation. At the participating institution P_i, data owner O_{P_i} encrypts and uploads the data permitted for training. For the data M_i to be shared, O_{P_i} specifies the access conditions, generates an access structure T_i, and encrypts the data M_i into ciphertext CT_i using the encryption algorithm of CP-ABE. The ciphertext CT_i contains the access structure T_i . Then O_{P_i} uploads CT_i to the cloud storage platform CSP_i.

Data Acquisition. At the participating institution P_i, when a data user U_{P_i} wants to access the data M_i within P_i, AA_i runs the key generation algorithm of CP-ABE and assigns a private key SK_{u_i} to U_{P_i} based on the user's attribute set S_i . Then, U_{P_i} obtains the ciphertext CT_i from the cloud and runs the decryption algorithm of CP-ABE to decrypt it. If the user's attribute set S_i aligns with the access structure T_i, the user's private key SK_{u_i} can decrypt CT_i to obtain the plaintext M_i. U_{P_i} repeats this process to decrypt multiple ciphertext data to obtain the original data shared by multiple owners. These decrypted data then form a data set D_i, which will serve as the training data for the next stage.

Collaborative Training. In participating institution P_i, a single data user U_{P_i} serves as trainer T_{P_i} for collaborative training. The trainers T = $\{T_{P_1}, \ldots, T_{P_n}\}$ from n institutions jointly participate in the training based on the collaborative training method proposed in Section B, as shown in Fig. 3. During the training process, taking the t-th round as an example, each selected trainer T_{P_i} performs local distillation of the global model on the local dataset D_i and then further trains the local model with the dataset D_i to obtain the new local model $local_{t+1}^i$. Next, the local update results $local_{t+1}^1, \ldots, local_{t+1}^m$ are aggregated by the central server to obtain $global_{t+1}$. After a certain number of training rounds, the final global model is obtained.

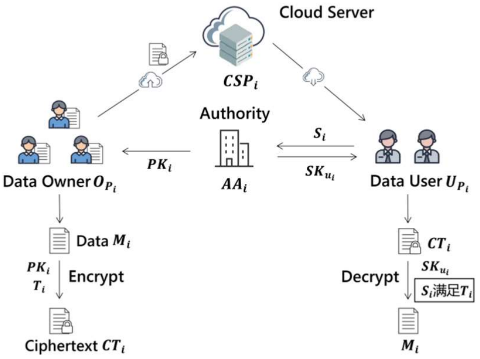


Fig. 2. Data authorization

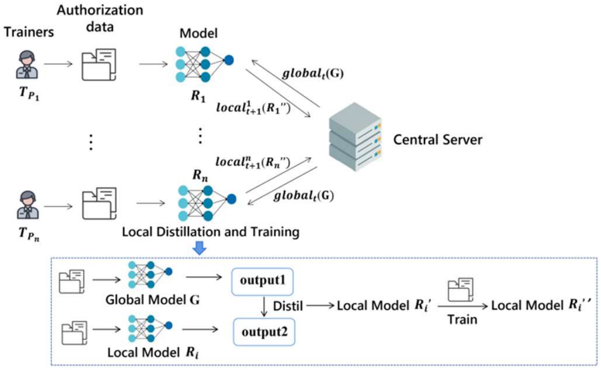


Fig. 3. Collaborative training

IV. EXPERIMENTS

In this section, we implement the prototype of the EPPCL system on a machine that is equipped with an Intel(R) Xeon(R) E5-2686 v4 2.30GHz CPU, 128GB RAM and installed with Ubuntu Server 22.04 64-Bit Version. The experiments focus on two parts: (1) the performance of data encryption and decryption; and (2) the performance of the improved federated learning algorithm.

A. Evaluation of Encryption and Decryption

To evaluate the efficiency of data encryption and decryption, we measure the time overhead of the encryption and decryption processes for various file sizes (1MB, 10MB, 20MB, 30MB) and formulate access policies with varying numbers of attributes (10-50, in intervals of 10) to analyze their impact. The experimental results are presented in Fig. 4–Fig. 7.

Time Overhead of Encryption. According to Fig. 4, it can be seen that for the same data, as the number of attributes increases, the encryption time also approximately increases in a linear proportion. Moreover, it can be inferred from Fig. 5 that when the number of attributes for encryption remains constant, the encryption time increases as the size of the data increases, and the relationship between them is close to a linear one. It is evident that both the number of attributes and the size of data are important factors affecting encryption efficiency.

Time Overhead of Decryption. According to the decryption results in Fig. 6, it can be observed that the decryption time for 1MB and 10MB files remains relatively stable under different numbers of attributes. However, for the other two types of files, the decryption time increases significantly under complex attributes. It can be concluded that changes in the number of attributes have a relatively small impact on decryption efficiency. Furthermore, Fig. 7 indicates that there is an approximately linear relationship between decryption time and file size under certain numbers of attributes.

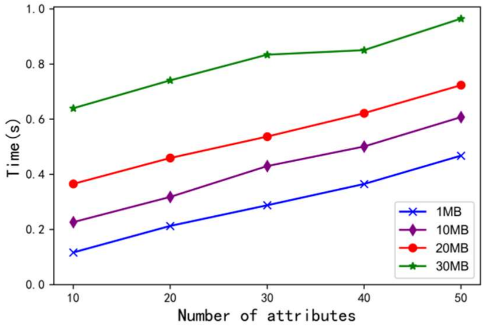


Fig. 4. Time overhead of encryption with different numbers of attributes

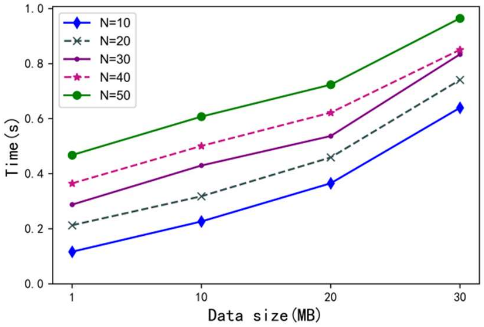


Fig. 5. Time overhead of encryption with different sizes of data

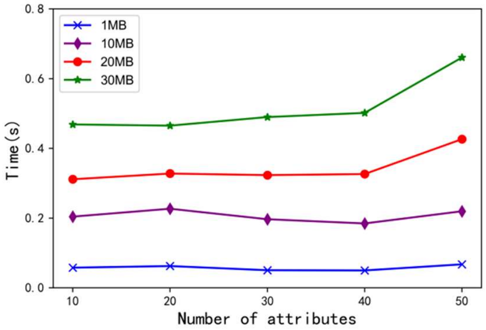


Fig. 6. Time overhead of decryption with different numbers of attributes

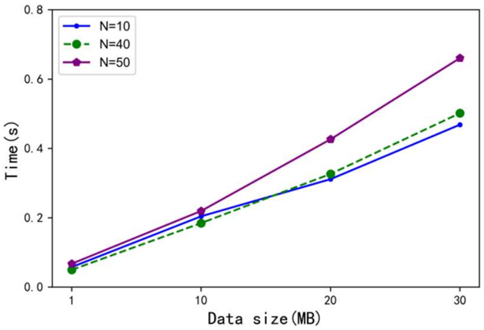


Fig. 7. Time overhead of decryption with different sizes of data

Based on the above experimental results, it is evident that the performance of data encryption and decryption in terms of time is feasible for this system.

B. Evaluation of Collaborative Training Method

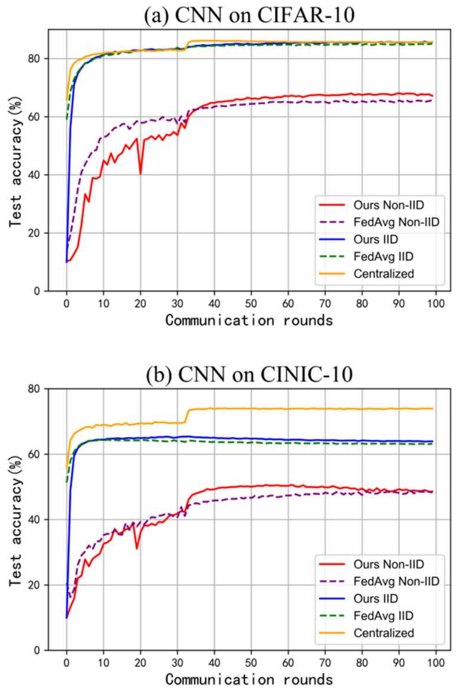
Datasets and Parameters. To validate the effectiveness of the proposed collaborative training method, we test it on the CIFAR-10, CINIC-10, and MNIST datasets with IID and Non-IID data distributions. In the Non-IID setting, each participating node is assigned only data samples of two specific categories. In addition, centralized training is also considered. We trained CNN, ResNet50 models on CIFAR-10 and CINIC-10, and MLP, LeNet5 models on MNIST. Performance comparisons are made between our method, FedAvg, and centralized training. With 10 local iterations and 100 communication rounds, we use a ratio of 8:2 or 7:3 for local distillation and training. The results are presented in Table I.

Comparison. Table I shows the model accuracy of the proposed method compared to the baseline method FedAvg under different data distribution. As can be seen from the Table I, for both IID and Non-IID data, our proposed method demonstrates higher model accuracy compared to the baseline method in these training tasks. In particular, in the Non-IID setting of the CIFAR-10 and CINIC-10 datasets, the proposed method can significantly improve prediction accuracy. This indicates the effectiveness of the proposed method for Non-IID data.

TABLE I. THE TEST ACCURACY OF GLOBAL MODEL FOR THE PROPOSED METHOD AND FEDAVG ON THREE DATASETS

Settings	**Methods**	**CIFAR-10**		**CINIC-10**		**MNIST**	
		CNN	***ResNet 50***	***CNN***	***ResNet 50***	***MLP***	***LeNet 5***
Non-IID	ours	**68.11**	**71.69**	**50.62**	**50.78**	**92.41**	**94.11**
	FedAvg	65.70	70.23	49.04	50.11	91.98	93.86
IID	ours	**85.84**	**89.47**	**65.40**	**73.09**	**98.78**	**99.37**
	FedAvg	85.11	89.22	64.45	72.86	98.69	99.29
	Centralized	86.44	90.30	74.39	76.67	98.84	99.29

We further analyze the relationship between communication rounds and model accuracy in federated learning under IID and Non-IID data. We conduct 100 rounds of experiments and compare our method with the baseline method FedAvg, with the results shown in Fig. 8. It can be observed that on CIFAR-10 and CINIC-10, our method can achieve higher accuracy in later communication rounds, especially for Non-IID data.


(a) CNN on CIFAR-10

(b) CNN on CINIC-10

Fig. 8. Test accuracy over communication rounds of the proposed method compared with FedAvg using a CNN model on IID and non-IID data of (a) CIFAR-10 (b) CINIC-10

V. CONCLUSION

In this paper, an efficient privacy-preserving collaborative learning framework is proposed, named EPPCL, which provides privacy preservation for personal data while collaborative training. In the EPPCL system, we design a new collaborative training method, which can enable trainers to learn the knowledge of global model while training a local model. This method solves the problem of training effect caused by data heterogeneity. The joint design of the attribute-based access control mechanism and the collaborative training method ensures data access and training securely. Moreover, we implement a prototype of EPPCL system and conduct performance tests. The experimental results show that EPPCL can train efficient models while protecting data privacy.

REFERENCES

[1] Y. Sasaki, "A survey on IoT big data analytic systems: Current and future," IEEE Internet of Things Journal, vol. 9, pp. 1024–1036, 2021.

[2] S. B. Atitallah, M. Driss, W. Boulila, and H. B. Ghezala, "Leveraging Deep Learning and IoT big data analytics to support the smart cities development: Review and future directions," Computer Science Review, vol. 38, 2020.

[3] F. Sattler, S. Wiedemann, K. R. Muller, and W. Samek, "Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, pp. 3400–3413, 2019.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," Artificial intelligence and statistics, pp. 1273–1282, 2017.

[5] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 10, pp. 1–19, 2019.

[6] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," IEEE Transactions on Knowledge and Data Engineering, 2021.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," IEEE Symposium on Security and Privacy, 2007.

[8] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," Computer Science, vol. 14, pp. 38–39, 2015.