PBPAFL: A Federated Learning Framework with Hybrid Privacy Protection for Sensitive Data

Ruichu Yao[1], Kunsheng Tang[1,2], Yongshi Zhu[1], Bingbing Fan[1(✉)], Tian Luo[1], and Yide Song[1]

[1] South China Normal University, Guangzhou 510631, Guangdong, China
fanbb1962@qq.com, {luotian,yide.song}@m.scnu.edu.cn
[2] University of Science and Technology of China, Hefei 230026, Anhui, China

Abstract. Due to the difficulties of exchanging data securely, data silos have become a critical issue in the era of big data. Federated learning provides an advantageous approach by enabling data holders to train a model collaboratively without sharing local data. However, multiple known inference attacks have made it impossible for a purely federated learning approach to protect privacy well enough. We present a PBPAFL algorithm that combines differential privacy with homomorphic encryption based on federated learning with an assessment module that enables the privacy budget parameters to be flexible in response to varying training requirements. The models trained using our proposed PBPAFL algorithm are capable of preventing inference assaults without a severe loss of precision. To demonstrate the efficacy of our proposed framework, we employ the PBPAFL algorithm to train a collection of face image-sensitive data. The experimental results show that our approach can improve the privacy protection of the model while maintaining precision.

Keywords: Federated Learning · Privacy Budget Parameter Adaptive · Differential Privacy · Homomorphic Encryption

1 Introduction

With the growing adoption of cloud computing, there has been a significant rise in the focus on big data. Data may be an essential and valuable commodity, and the value of big data can be used to improve life and serve society through the precise analysis of massive volumes of data using machine learning (ML) techniques. Although these services may appear appealing, it is challenging to ensure that sharing specific data will not violate privacy regulations. For instance, in May 2018, the European Union (EU) enacted the General Data Protection Regulation (GDPR) bill, which stipulates that any information connected to an individual is personal data and that the use of such data must be expressly permitted by users [1]. With the growing emphasis on data security and the adoption of privacy protection regulations, "data silos" must be resolved. Data silos relate to the inability of disparate companies or corporate divisions to share and interact with data.

S. Goel et al. (Eds.): ICDF2C 2022, LNICST 508, pp. 403–416, 2023.
https://doi.org/10.1007/978-3-031-36574-4_24

For machine learning (ML) and deep learning (DL) to deliver the intended outcomes, the quantity and quality of training data have a substantial impact on the results. Accordingly, there is an increasing demand to enhance the quantity of data through sharing [2]. To better use data, increase its value, and enable artificial intelligence (AI) for the benefit of human civilization, data silos must be eliminated. Google was the first organization to introduce the notion of Federated Learning (FL) [3] and apply it for mobile input prediction in order to overcome the issue of data leakage, therefore fulfilling the aim of data sharing and overcoming the data silo conundrum. FL enables data holders interested in training to execute machine learning (ML) algorithms locally and train a model jointly by communicating just model parameters, which will be aggregated by the server. Thus, the possibility of data leaking as a result of leaving the local region is eliminated. Nonetheless, this method does not yet provide enough privacy protection for training data. On the basis of the parameters communicated during the FL process, it is feasible to establish not only whether certain samples belong to a certain node [4], but also to invert some training data [5]. Consequently, the local data still presents some security problems.

We provide the PBPAFL, a framework that extends FL with privacy-preserving approaches like differential privacy (DP) and homomorphic encryption, which can protect model parameters from two main threatening attacks. The issue of low accuracy due to the incorporation of DP noise is resolved. The framework's algorithm, which we name the PBPAFL, constantly changes the amount of noise injected during training to meet different needs, reaching a balance between practicality and privacy protection. The primary results from this study are as follows:

- We propose a protection approach that combines differential privacy and homomorphic encryption to inject noise and encrypt the gradient throughout the training process in order to defend against gradient leak attacks that may be faced in FL and enhance its privacy protection capacity. This resolves the remaining data security concerns within the FL procedure.
- We describe a privacy budget parameter adaptive algorithm, PBPAFL, that can change the amount of noise injected during training to tackle the problem of excessive noise diminishing accuracy and maintaining the model's utility while safeguarding data privacy.
- We verify our approach using the accessible face dataset LFW, and experimental findings demonstrate that our algorithm can control the accuracy loss (ACL) within 3% while maintaining privacy. Our method is relevant to more picture categorization issues.
- We investigate the generalizability of the technique to data that does not follow the ideal IID distribution. In comparison to FL methods trained on IID data, those trained on non-IID data provide less effective models. The results show that our technique is effective even when used with Non-IID data.

2 Methodology

2.1 Overview

In conventional machine learning setups, training datasets are centralized on the server where the algorithm is executed. It will lead to an unavoidable problem where data is out of control when it leaves the local area. Even if all participants can be trusted, we would still like a mechanism to avoid revealing local data to others, particularly when this data contains sensitive personal information such as face images. Federated learning overcomes these restrictions by permitting all participants to train locally and sending only trained parameters to collectively train the same model. But Florida is not absolutely safe either. We also need to look into the possibility of getting back to the original data by changing the parameters.

In this research, we propose a PBPAFL algorithm that upgrades FL to resist eavesdropping attacks by adding noise and homomorphic encryption to the parameters. In addition, our algorithm is adaptive and may dynamically alter the quantity of noise to preserve the model's effect. Our algorithm comprises the steps listed below: 1) add noise to the parameters, which is obtained by local training on the client side; 2) evaluate model effects and re-add noise to modify parameters (if needed); and 3) homomorphic encryption of the noise-added parameters. It can adjust the privacy budget parameters to provide appropriate privacy protection based on the needs of different situations and lower the amount of noise through homomorphic encryption to balance the accuracy of the model. Experiments reveal that the concept of limiting the amount of noise added by altering the privacy budget settings in order to finally guarantee model performance is viable. And the final accuracy loss of our method can be managed to within 3%.

This paper is structured as follows: In Sect. 2, we define FL and detail the fundamental components of our proposed method, including the noise mechanism, homomorphic encryption, and privacy budget parameter adaptive FL algorithm. Section 3 describes the dataset portion and the experimental details. In Sect. 4, we summarize the outcomes of the experiment. In Sect. 5, we explore a few issues that arose during the tests. Finally, we conclude by summarizing the experiment and drawing conclusions.

2.2 Federated Learning

Federated learning is a machine learning approach in which multiple individuals can train a model cooperatively without trading or combining data. The following is a general description of FL.

Assuming that a FL system consists of 1 server and N clients, and the full training data is D. The i-th client has the training data as D_i, where $i \in \{1, 2, \ldots, N\}$, then $\sum_{i=1}^{N} D_i = D$. During the training phase, each client trains the model with its local data in order to find a vector w_i that minimizes a specific loss function. The server side needs to aggregate the parameters w_i received from each client, i.e.,

$$w = \sum_{i=1}^{N} p_i w_i, \tag{1}$$

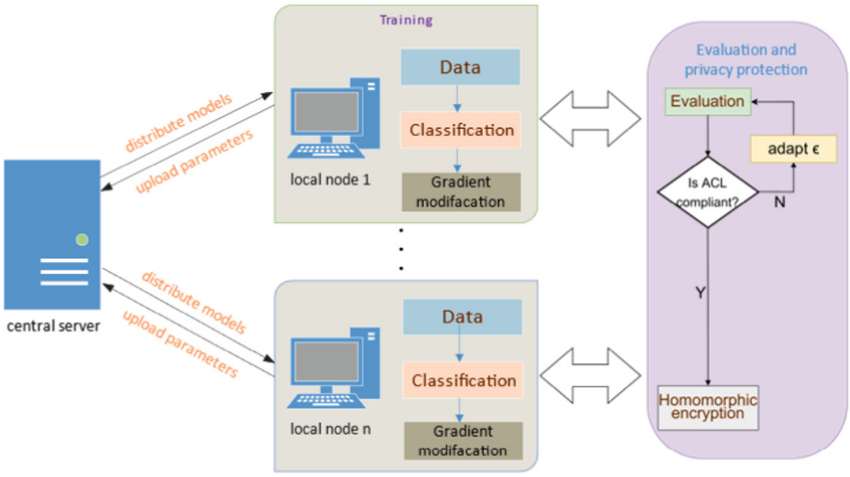

Fig. 1. A privacy budget parameters adaptive framework based on FL.

where p_i indicates the weight of the i-th client, then $p_i = \frac{D_i}{D}$ and $\sum_{i=1}^{N} p_i = 1$. We define the loss function for the i-th client as $F_i()$, then we can state the optimization problem as

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N} p_i F_i(D_i, w) \tag{2}$$

Through federated learning, more data can be utilized to train a model collectively, tackling the problem of low model accuracy resulting from insufficient data volume. At the same time, the training data stays in the local area, so data leakage is successfully avoided.

Despite the fact that FL provides a novel solution to the problem of local data leakage in the conventional data-centralized ML model, there are still certain security risks involved. The most significant aspect is the transmission parameters for eavesdropping, which may lead to privacy leakage.

2.3 Privacy-Preserving Mechanisms in Federated Learning

Generally speaking, privacy-preserving approaches for distributed learning systems serve two primary objectives: 1) safeguarding the confidentiality of the training data and 2) safeguarding the confidentiality of model parameters shared by the local client and the server. Differential privacy approaches proposed by Zhang et al. [6], data anonymization techniques proposed by Bayardo et al. [7], and homomorphic encryption techniques proposed by Gentry [8] are mainstream privacy-preserving strategies in the field of FL. Following is a detailed description of the differential privacy and homomorphic encryption algorithms utilized by the proposed framework.

Differential Privacy. Differential privacy is primarily the alteration of the original data by introducing noise that fulfills mostly Laplace and Gaussian distributions, such that it

differs to some extent from the actual data. It prevents attackers from accessing sensitive information via the published datasets or the given service interface. Zhang et al. [6] define differential privacy as follows:

Definition 2.1 (ϵ - differential privacy (ϵ-DP)). If there are m users, then each user will have their own record. If the function F satisfies the following inequality, then for every two input records m_0, m_1 ($m_0, m_1 \in D_F$), the same output m_* ($m_* \in R_F$) is attained.

$$pr[F(m_0) = m_*] \leq e^{\in} \times pr[F(m_1) = m_*] \tag{3}$$

Then it is claimed that the function F satisfies ϵ- differential privacy, where ϵ is the privacy budget. The above D_F, R_F represent the function F's definition and value domains, respectively. This mechanism's noise satisfies the Laplace distribution.

However, using only differential privacy to protect privacy cannot completely prevent an attacker from gaining access to sensitive information, such as data source and data distribution, from supplied settings. Consequently, we must additionally consider the possibility of data leakage as a result of inference-based attacks on the output.

Homomorphic Encryption. Using homomorphic encryption techniques is another method for protecting data privacy and security in machine learning, particularly in centralized systems such as cloud servers where data is collected for collaborative training without disclosing the original information. Homomorphic encryption makes it possible to execute computations on encrypted forms of data without requiring a decryption key [9]. The result of the computation is delivered in an encrypted format that can only be decrypted by the requester. In addition, homomorphic encryption ensures that the decrypted output is identical to the original output computed on the unencrypted data set.

Homomorphic encryption can be categorized as slightly homomorphic encryption (SWHE) and fully homomorphic encryption (FHE) [10], based on the encryption technique and the class of computational operations that can be performed on the encrypted form. Some conventional cryptographic approaches, such as FHE developed by Gentry et al. in [9], can conduct any arbitrary action on the cipher text (and hence achieve any desired functionality) in order to create encrypted outputs. Using the decryption function, original data or ciphertext calculations may be conveyed mathematically in FHE without conflict. Paillier is the most prevalent homomorphic encryption algorithm in the FL situation. Paillier is defined as

$$\mathrm{Enc}(\mathrm{m}_1) + \mathrm{Enc}(\mathrm{m}_2) = \mathrm{Enc}(\mathrm{m}_1 + \mathrm{m}_2) \tag{4}$$

However, performing calculation operations on ciphertexts is computationally intensive. Therefore, the application of homomorphic encryption to large-scale data training remains impracticable.

Our Research Work. Both differential privacy and homomorphic encryption can only guarantee privacy in FL from a single viewpoint, which limits their privacy-preserving impact and model performance. With equal privacy-preserving guarantees, 1) due to differential privacy's susceptibility to output inference attacks that compromise data, this approach requires more noise to achieve the same level of privacy as the method employing a single application of differential privacy. Additionally, more noise leads to

a decrease in model prediction accuracy; 2) if only homomorphic encryption is used, it may produce a greater level of model prediction accuracy while maintaining the same level of privacy, since it does not involve direct interference with the original data, hence improving data availability. But there is still a chance that this method could leave sensitive information open to inference attacks.

In this research, we present a hybrid privacy protection mechanism that combines homomorphic encryption and differential privacy approaches to secure privacy in FL from several dimensions and obtain higher privacy guarantees with high accuracy. We add differential privacy noise to the gradient parameters, and then encrypt the noise-added parameters using homomorphic encryption to further enhance privacy protection. Since homomorphic encryption already provides some level of privacy protection, a modest quantity of noise is sufficient to achieve enhanced privacy protection, reducing the necessary privacy budget and achieving enhanced privacy protection while enhancing accuracy.

2.4 Privacy Budget Parameter Adaptive Federated Learning

FL provides a function for the local data privacy concern within the multi-party ML paradigm, hence enabling federated data silos. There is nevertheless a danger of data leaking from the local client in the event of an external attack during the FL process. We consider both adding noise and homomorphic encryption to solve this problem. However, we must also consider the applicability of the model, so we propose a Privacy Budget Parameter Adaptive Federated Learning (PBPAFL) algorithm that can dynamically modify the amount of noise added during each FL round to guarantee the final model's accuracy meets expectations. As illustrated in Fig. 1, our architecture is made up of three parts.

Part 1. Federated Learning. We'll assume the system has one central server and n local clients. The central server launches the training task, picks the local clients that will participate in the training, and selects a model to be trained based on the task while initializing the model. FL enables the use of data from different clients to train a model

cooperatively, extending the number of training sets while guaranteeing that the data does not leave the local region.

Part 2. Parameters Protection. Combining differential privacy and homomorphic encryption, two ways to protect privacy, is used to change the gradient and encrypt it. This step is intended to increase the security of data.

Part 3. Privacy Budget Parameters Adaptive FL. The model is checked after noise is added, and the privacy budget parameters are dynamically changed to find a balance between protecting privacy and making the model effective.

Algorithm 1:PBPAFL

Initialization

for each epoch i = 1,2,⋯,t do

  Distribute w_i

  for each client c_k,k=1,2,⋯,n do

    Update the local gradients:

    $w_i^k \leftarrow w_i - \eta \nabla F(w_i, d_k)$

    Add noise:

    $g_i^k \leftarrow w_i^k + Gaussion(\epsilon_k)$

    Evaluation:

    if $acc_{g_i^k} \leq (1 - x\%) * acc_{w_i^k}$

      $\epsilon_k' \leftarrow \epsilon_k$

      $g_i^k \leftarrow w_i^k + Gaussion(\epsilon_k')$

    Homomorphic encryption:

    $\hat{g}_i^k \leftarrow g_i^k$

    return $\hat{g}_i^k$ to server

  Aggregation:

  $w_{i+1} = \sum_{k=1}^{n} \lambda_k \hat{g}_i^k$

  (return w_{i+1})

After the setup is complete, the iteration begins, and in the i-th round, the central server sends the initial model parameters $\boldsymbol{w_i}$ to each client $\boldsymbol{c_k}$, $\boldsymbol{k} \in (1, \boldsymbol{n})$. After training with its local data $\boldsymbol{d_k}$, each local client obtains the model parameters $\boldsymbol{w_i^k}$. The local client will choose the privacy budget parameter $\boldsymbol{\epsilon_k}$ and add the corresponding noise to it. The parameter after adding noise is noted as $\boldsymbol{g_i^k}$. Next, the noise-added model is evaluated and the result will be compared to those models without adding noise. If the expected requirement is not met (assuming the desired accuracy loss rate is $\boldsymbol{x\%}$), $\boldsymbol{\epsilon_k}$ must be selected again until it passes. After finding $\boldsymbol{\epsilon_k}$, all local clients will use homomorphically encrypt to ensure that the parameters can successfully withstand eavesdropping attacks, and then provide the encrypted parameters $\hat{\boldsymbol{g}}_{\boldsymbol{i}}^{\boldsymbol{k}}$ to the central server.

The central server aggregates all received parameters to update the global model and sends the new parameter w_{i+1} to each local client. Then the next round of training will begin. Due to the twofold security provided by noise perturbation and homomorphic encryption, it has become almost unfeasible for eavesdropping attempts to revert the original data. In the meanwhile, we may dynamically modify the quantity of noise during model training to preserve its accuracy. Algorithm 1 describes the execution flow of our framework.

To address the issue of data privacy in FL, we provide a PBPAFL algorithm to deal with the risk of local training data leakage produced by eavesdropping attacks. Furthermore, since adding noise affects accuracy, we analyze the accuracy loss rate in each round in order to dynamically alter the amount of noise added. We will next illustrate the viability of our methodology through experiments.

3 Experimental Setup

In this section, we will first discuss the dataset utilized for the experiments as well as the data preparation portion. Then, we present the experimental framework in this study, including the face recognition algorithms, the privacy budget parameter setup, and the evaluation criteria. Finally, we describe the experimental procedures. We are interested in finding answers to the following questions via our experiments:

Q1. Why are noise and homomorphic encryption necessary?

Q2. What are the advantages of including noise in the parameter as opposed to the dataset?

Q3. How does adaptation reflect?

3.1 Experimental Dataset

In this research, the Labeled Faces in the Wild (LFW) dataset is utilized to validate the feasibility of our experiment. The LFW dataset is widely used for face recognition in uncontrolled scenarios, with a total of 13233 distinct face images, 5749 identity labels (virtual names), and 1680 individuals having two or more face shots. Each image is a jpeg with a resolution of 250 $\times$ 250. We exclude items that have more than 40 face images. Then we chose the remaining portion of items with close to 40 images and augmented them such that the final size of our experimental dataset is 30 individuals with 60 images each, for a total of 1800 face images.

We split the dataset into two portions of 80% and 20%, respectively, as training data for the local clients and a test set to evaluate the performance of the final model when training has been completed. In our research, we treat the independent identical distribution (IID) and the non-independent identical distribution (Non-IID) of training data as two distinct training sets. In the IID case, the training set data is distributed equally among the number of local clients. In the Non-IID case, we split the training data ratio into two categories, 4:3:1 and 4:3:2.

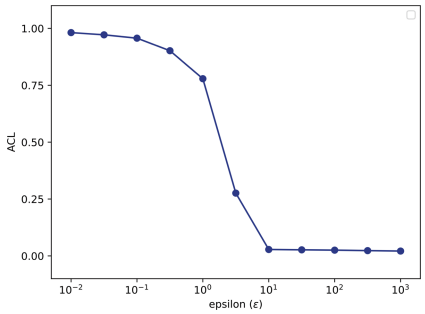

Fig. 2. The correspondence between ACL and ϵ

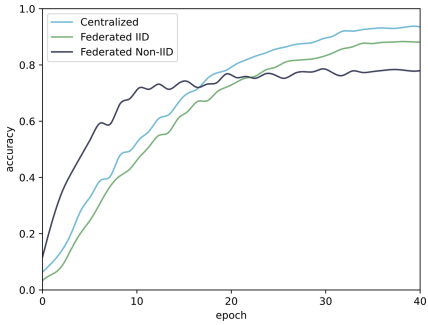

Fig. 3. Accuracy comparison of three cases without DP noise.

3.2 Model Evaluation Metrics

The objective of machine learning is to learn the characteristics of training data in order to predict the labels of input examples not included in training. To assess the efficacy of the training model, predictions are made on the test set (which was not utilized for training) and then compared to the actual labels. The accuracy of the training model is defined as the ratio of the set of predictions whose outcomes are consistent with the true labels ($n_{correct}$) to the set of true label holdouts ($n_{holdout}$) (ACC).

$$ACC = \frac{n_{correct}}{n_{holdout}} \tag{5}$$

Accuracy is a straightforward indicator of ML prediction performance. Other prevalent measures are AUC, precision, recall, and F-Score. Also, new criteria like model fairness and reducing the amount of work that needs to be done [11] may be needed to evaluate FL performance.

What we focus on is accuracy loss (ACL), which is measured as the performance loss when our privacy-preserving mechanism is applied to the FL process (m) in comparison to a ML model trained without applying the privacy mechanism ($\epsilon = \text{inf}$).

$$Accuracy\ Loss(ACL) = 1 - \frac{ACC(m, \epsilon)}{ACC(m, \epsilon = inf)} \tag{6}$$

3.3 Convolutional Neural Network

In this paper, we use a convolutional neural network (CNN) for training and apply the approach to the problem of face image classification. Convolutional neural networks are multilayer neural networks that consist of overlapping convolutional layers for the purpose of feature extraction and sampling layers for the purpose of feature processing. Face images in our training data are the input of CNN. After a number of "convolution" and "sampling" steps, a fully connected layer network is used to map the output to the target. The output is the result of classification after recognition.

In our experiments, the CNN model contains two convolutional (CONV) layers, ReLU, Softmax of 30 classes, and stochastic gradient descent (SGD) as the optimizer. Additionally, we use a cross-entropy loss function. The default settings of CNN used in this paper's experiments are listed in Table 1. We set the following parameters: learning rate = 0.0001, batch size = 64, epoch = 40, weight decay = 0.001, total number of images = 1800, and number of labels = 30.

Table 1. Default parameters for CNN.

Parameter	Default value	Description
learning rate	0.0001	step size for updating the weight matrix
batch size	64	number of samples in each epoch
epoch	40	training rounds
weight decay	0.001	adjust the effect of model complexity on the loss function

3.4 Privacy Budget Parameters and Evaluation Methodology

Privacy Budget Parameters. To avoid the model's accuracy severely degrading owing to excessive noise addition, we manage the amount of noise added by altering the privacy budget parameter. The privacy budget for adding noise is set as

$$\epsilon \in \{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000\}$$

We believe that the actual accuracy needed will vary depending on the situation. When the accuracy threshold value of a situation's requirement changes, the quantity of added noise has a direct impact on the final ACL. To solve this problem, local clients compare the value after adding noise to the value without noise in all rounds. If the result does not meet the requirement this time, local clients should adjust the amount of added noise until the requirement is met.

Evaluation Methodology. The training data will be duplicated k times. Perform k rounds of iterations, with k-1 subsets in each iteration serving as the training set and the remaining 1 subset serving as the validation set. Also, make sure that each subset is used as the validation set. We assess the appropriateness of the quantity of noise added by ACL. In each round, we write down the model's ACL values, and then we take the average of these k ACLs to get our final evaluative criterion.

3.5 Experimental Procedure

Our experimental procedure consists of six steps: 1) data preprocessing; 2) model initialization and training; 3) privacy budget selection and noise addition; 4) assessment and dynamic adjustment of; 5) homomorphic encryption of the parameters; and 6) aggregation of the encrypted parameters. First, we need to preprocess the LFW dataset and divide

the face data into local clients according to the planned proportions. When the training is launched, the server delivers the training model to local clients. After training, local clients will add noise and evaluate the ACL. Depending on the evaluation results, local clients dynamically alter the chosen privacy budget. Then the modified parameters will be sent to the server after homomorphic encryption. The server gets all of the encrypted parameters, performs aggregation, and sends the updated model to the local clients so that the next round can begin.

When training is finished, the ACL of the final model will be evaluated on the test set. We set up several sets of comparison experiments to verify the effectiveness of our methodology. Moreover, we also performed experiments in the Non-IID case.

Remark

1) We utilize face image data for our research, which is a representative type of sensitive data. Face data is commonly dispersed across several organizations and difficult to integrate, presenting the problem of "data silos".
2) Our methodology primarily analyzes external eavesdropping threats and protects local clients' data from leaking. But we can't protect against every possible threat, like attacks that come from inside the system.
3) With Non-IID data, the final model generally performs badly. So we perform experiments for these cases and discuss the findings in Sect. 4.

4 Experimental Results

Our experiments are conducted on IID and Non-IID data to compare the proposed methodology to traditional data-centralized ML and FL. This section will explain the results of our experiments and answer the questions asked in Sect. 3.

4.1 Performance Analysis

We propose a strategy for controlling noise by modifying the value of ϵ to suit the real feasible demand (ACL value) without compromising privacy protection. Figure 2 shows the effect of our approach on face recognition, where the ACL depends on the value of ϵ. We notice that as ϵ decreases, the quantity of adding noise increases and the ACL value also rises.

Table 2 shows the related data for ACL- ϵ, demonstrating that the value of ϵ may be adjusted to accommodate different ACL values.

Figure 3 depicts the accuracy performance of the LFW dataset for FL and data-centralized ML, demonstrating that while FL may offer some protection, it still leads to a drop in accuracy. Figure 4 shows how our method performs in both the data-centralized and data-distributed cases. It can be seen that our method protects privacy better and has a lower ACL than data-centralized training.

IID vs Non-IID. Two cases are created in Non-IID for the experiments, with a data volume split of 4:3:1 and 4:3:2, respectively. The results are shown in Fig. 5. Our methodology differs in ACL by just 3% between the data-centralized case and the IID case, which indicates that it is possible to balance privacy protection and ACL. But in Non-IID case,

Table 2. The values corresponding to different ACLs.

ACL	ϵ
0.02	100
0.03	10
0.26	5
0.76	1
0.89	0.5

the ACL is slightly lower than in other cases, which may be attributable to the uneven data distribution.

4.2 Summary of Findings

We propose a method to prevent training data leakage during the FL process. After local clients complete training, noise will be added and homomorphic encryption will be applied. In addition, we propose an assessment process to modify the amount of noise added and maintain the model's ultimate result. In answering the questions raised in Sect. 3, we obtained the following conclusions through experimentation:

1) This methodology enables silo data to participate in the model training to improve model quality while providing better protection for training data, which is also suitable for other sensitive data.
2) Typically, there are three options for adding noise to safeguard data privacy: adding noise to the dataset, adding noise to the parameters (e.g., gradients), or adding noise to the model. By adding noise to the parameters, it can prevent participants in FL from accessing training data. Therefore, this step is effective and necessary in our proposed approach.
3) From the experimental results, it can be seen that the ACL is gradually lowered and eventually becomes stable, which demonstrates the method's adaptability to varying ACL demands. In addition, we conducted experiments to compare our method to available methods, both in the ideal IID setting and the more realistic Non-IID case, with conclusive findings demonstrating the superiority of our approach.

5 Discussion

Despite the unique advantages of FL, it still cannot solve all the data security problems in ML. Whether a model can be trained successfully is dependent on data quality, bias, and standard deviation [12]. Appropriate action is needed to counteract these effects. The heterogeneity of face data is a significant factor affecting the effectiveness of FL models. Face data is particularly diverse—not only because of the diversity of patterns, dimensions, and general features but also because of factors such as acquisition differences and local demographics, even within specific protocols. There are substantial distinctions in the characteristics of these several types of faces.

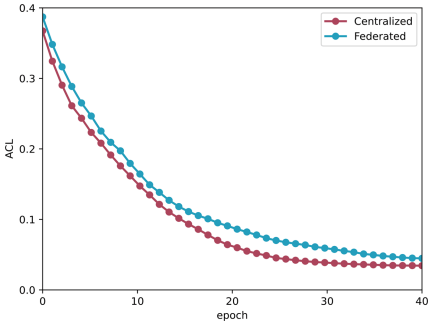

Fig. 4. ACL for FL and data-centralized ML.

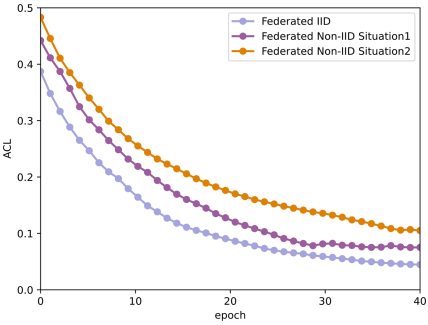

Fig. 5. ACL of PBPAFL in three data distributed cases.

FL may aid in addressing certain causes of variance by diversifying data sources, but the non-independent identical distribution of data is a challenge for FL algorithms. Recently, a study [13] demonstrated that FL is still achievable even if data is not provided evenly across institutions. However, the effectiveness of the model is drastically reduced compared to that achieved by training with IID data. The results of our experiments demonstrate that our methodology is suitable for Non-IID data as well.

6 Conclusion

Various significant advances in facial recognition technology have been made possible by machine learning. Federated learning is a potential way to generate robust, accurate, and unbiased models since all ML methods considerably benefit from access to close to genuine data distributions.

The primary contribution of this paper is the proposal of a framework based on PBPAFL that balances model performance and privacy protection to effectively tackle problems associated with eavesdropping attacks on sensitive data in FL by making use of combining differential privacy and homomorphic encryption. So, it could make the face recognition industry around the world more effective and secure, and it could also lead to new research on FL.

The future research direction of this paper is to try whether the methodology is applicable in more scenarios. If possible, we will consider whether PBPAFL can be applied to general applications of machine learning.

Acknowledgment. We appreciate the informative remarks made by the anonymous reviewers of this work. This research is supported by the Special Fund for the Key Program of Science and Technology of Guangdong Province, China (Grant No. 2016B030305003).

References

1. Colesky, M., Demetzou, K., Fritsch, L., Herold, S.: Helping software architects familiarize with the general data protection regulation. In: 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), pp. 226–229. IEEE, United States (2019)
2. Kim, J., Ha, H., Chun, B.G., Yoon, S., Cha, S.K.: Collaborative analytics for data silos. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), pp. 743–754. IEEE, United States (2016)
3. Ahmed, K.M., Imteaj, A., Amini, M.H.: Federated deep learning for heterogeneous edge computing. In: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1146–1152. IEEE, United States (2021)
4. Song, C., Ristenpart, T., Shmatikov, V.: Machine learning models that remember too much. In: Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security, pp. 587–601. ACM, United States (2017)
5. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. Ion: Advances in Neural Information Processing Systems, vol. 32. NeurIPS, Vancouver (2019)
6. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
7. Bayardo, R.J., Agrawal, R.: Data privacy through optimal k-anonymization. In: 21st International Conference on Data Engineering (ICDE 2005), pp. 217–228. IEEE, United States (2005)
8. Gentry, C.: Computing arbitrary functions of encrypted data. Commun. ACM **53**, 97–105 (2010)
9. Gentry, C., Halevi, S.: Implementing gentry's fully-homomorphic encryption scheme. In: Paterson, K.G. (ed.) Advances in Cryptology – EUROCRYPT 2011. EUROCRYPT 2011. Lecture Notes in Computer Science, vol. 6632. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_9
10. Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes: theory and implementation. ACM Comput. Surv. (CSUR) **51**(4), 1–35 (2018)
11. Wang, F., Casalino, L.P., Khullar, D.: Deep learning in medicine—promise, progress, and challenges. JAMA Intern. Med. **179**(3), 293–294 (2019)
12. Yaji, S., Bangera, K., Neelima, B.: Privacy preserving in blockchain based on partial homomorphic encryption system for AI applications. In: 2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW), pp. 81–85. IEEE, United States (2018)
13. Li, X., Gu, Y., Dvornek, N., Staib, L.H., Ventola, P., Duncan, J.S.: Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results. Med. Image Anal. **65**, 101765 (2020)