ESP-IJACT
ESP International Journal of Advancements in Computational Technology
ISSN: 2583-8628 / Volume 4 Issue 2, April-June, 2026 / Page No: 1-10
Paper Id: IJACT-V4I2P101/Doi: 10.56472/25838628/IJACT-V4I2P101

Research Articles

PBDFL: A Privacy-Preserving and Byzantine-Robust Scheme for Decentralized Federated Learning

Wenjuan Lian[1], Li Liu[2], Fengnian Cai[3], Hongbao Zhang[4], Xin Chen[5], Bin Jia[6]
[1,2,3,5,6]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China.
[4]Eversec Technology Co.,Ltd., Beijing, China.

Received Date: 12 May 2026 Revised Date: 25 May 2026 Accepted Date: 13 Jun 2026

Abstract: *Federated learning (FL) effectively addresses data silos but remains vulnerable to Byzantine attacks and privacy leakage, particularly in the presence of untrusted servers. While existing robust aggregation algorithms mitigate the impact of malicious participants, accurately identifying them without compromising privacy remains a significant challenge. To address these issues, this paper proposes PBDFL, an efficient privacy-preserving and Byzantine-robust scheme for decentralized federated learning. First, we introduce a Byzantine-robust decentralized training mechanism that dynamically identifies and excludes Byzantine participants via a reputation-based system, thereby securing the global model's accuracy. Second, we design a hybrid privacy-preserving aggregation (PPA) mechanism combining Homomorphic Encryption (HE) and Differential Privacy (DP). This mechanism selectively encrypts parameters to defend against both internal inference and external eavesdropping while maintaining communication efficiency. Theoretical analysis and extensive experiments on MNIST, FashionMNIST, and EMNIST datasets demonstrate that PBDFL significantly outperforms state-of-the-art baselines. It effectively safeguards participant privacy and maintains high model accuracy even under intensive Byzantine attacks.*

Keywords: *Federated Learning, Byzantine Robustness, Decentralized Architecture, Privacy-Preserving Aggregation, Homomorphic Encryption.*

I. INTRODUCTION

Machine learning has revolutionized fields such as image recognition[1] and biology [2]. However, traditional centralized learning faces regulatory challenges and data privacy concerns. To address this, Google introduced Federated Learning (FL) **Error! Reference source not found.**, enabling distributed collaborative training by exchanging model p arameters instead of raw data. Despite its advantages, FL is not immune to privacy risks, as adversaries can reconstruct user data [3] or infer membership [5] from uploaded gradients. Furthermore, FL faces critical challenges regarding privacy preservation, Byzantine robustness, and centralized architecture risks.

Privacy-Preserving Techniques. To mitigate privacy threats, methods based on Differential Privacy (DP) [6] and Homomorphic Encryption (HE) [7] have been widely adopted. For instance, Shokri et al.[17] and Hamm et al.[18] applied DP to protect gradients and predictions, respectively. While DP effectively protects privacy, it often degrades model utility due to added noise. Alternatively, HE-based methods [11][12], , such as those by Li et al.[19] and Yang et al. [20] use cryptography to encrypt parameters. Although HE maintains accuracy, its high computational cost limits scalability in large-scale applications.

Byzantine Robustness and Decentralization. Beyond privacy, FL is vulnerable to Byzantine attacks [13], where malicious participants disrupt convergence. Traditional robust aggregation rules like Krum [14], Median [15], GeoMedian [16], and RSA [21] can filter outliers but typically require access to raw model parameters, thereby compromising privacy. While recent hybrid approaches attempt to combine privacy and robustness—such as Ma et al. [23] (combining RSA with HE) and Zhao et al. [24] (using TEE)—they often rely on a trusted central server. This reliance creates a single point of failure. To eliminate this bottleneck, decentralized schemes like Gossip-based protocols [25]-[26] and committee mechanisms [28] have been proposed. However, existing decentralized solutions still struggle to effectively identify Byzantine participants while maintaining efficiency and privacy.

Our Contribution. Balancing privacy, robustness, and efficiency remains a significant challenge. To address these issues, we propose **PBDFL**, a novel federated learning scheme that integrates communication efficiency, privacy protection, and Byzantine robustness.

The main contributions are as follows:

- We propose PBDFL, which is a federated learning scheme that considers privacy protection and Byzantine robustness. In response to the current situation where existing Byzantine-robust algorithms are unable to identify

Byzantine participants, we achieve the decentralization and Byzantine robustness of PBDFL by designing a Byzantine-robust decentralized training mechanism. The Byzantine-robust decentralized training mechanism is able to identify Byzantine participants and exclude them from the training process, providing stronger Byzantine robustness.

- We design a privacy-preserving aggregation mechanism (PPA) for decentralized federated learning, which selects different methods for different parameters to protect the privacy of participants while ensuring training efficiency, and can resist internal and external security threats at the same time.
- We conducted extensive experiments on the MNIST, FashionMNIST, and EMNIST datasets to verify the performance of PBDFL. The experimental results showed that PBDFL reduced the influence of Byzantine participants and protected the privacy of participants on the premise of satisfying efficient communication, and its test accuracy was higher than that of other Byzantine-robust federated learning schemes in most cases when subjected to Byzantine attacks.

II. PRELIMINARIES

A. Federated Learning

While FL avoids raw data sharing, it remains vulnerable to various security threats, primarily categorized into internal and external threats. Internal threats mainly stem from Byzantine participants and untrusted servers. Malicious participants may launch poisoning attacks by injecting crafted data or models to disrupt global model convergence [14]. Unlike simple noise injection, Byzantine participants can collude strategically, making their attacks highly destructive and difficult to defend against. A critical limitation of existing defenses is the inability to identify specific Byzantine participants. Furthermore, in centralized FL, the server controls parameter aggregation and model distribution. An untrusted server can infer private information from uploaded updates or tamper with the global model [27], posing a single point of failure and privacy leakage. External threats typically involve adversaries eavesdropping on communication channels to intercept sensitive information or compromise the integrity of data transmission. To address these challenges, our proposed scheme, PBDFL, incorporates a Byzantine-robust decentralized training mechanism and a privacy-preserving aggregation protocol.

B. Privacy Preserving for Federated Learning

Common privacy-preserving strategies in federated learning fall into two categories: HE and DP. HE focuses on secure transmission by performing computations on encrypted data, thereby concealing intermediate parameters. Conversely, DP emphasizes data anonymity by adding noise to perturb gradients, making it computationally infeasible to link updates to specific users.

a) Homomorphic encryption (HE)

HE allows computations to be performed directly on encrypted data, ensuring that intermediate parameters remain concealed during transmission. In this work, we adopt the CKKS scheme [29], which supports approximate arithmetic on real numbers—a crucial feature for training machine learning models. Compared to traditional schemes like Paillier [30], CKKS offers superior encryption and decryption efficiency by operating on polynomial rings $Z_q[X]/(X^N + 1)$. However, despite its security benefits, the high computational and communication overhead of HE makes it impractical to apply to the entire model, motivating our hybrid approach.

b) Differential privacy (DP)

While HE secures the computation process, DP protects the output against inference attacks by ensuring statistical indistinguishability. In this work, we employ Local Differential Privacy (LDP), which perturbs model updates on the client side before transmission. The level of protection is controlled by the privacy budget ε; a smaller ε guarantees stronger privacy but introduces larger noise variance. This creates a fundamental trade-off: excessive noise can degrade model accuracy, requiring a careful selection of ε to balance privacy and utility.

C. Treat Model and Goals

We assume that each participant in our proposed PBDFL is an "honest and curious" semi-trustworthy entity. Semi-honest model is a common assumed model in federated learning. According to this hypothetical model, each participant in PBDFL will abide by the set protocol, but will obtain the privacy information of other participants through intermediate data inference during training. At the same time, we also consider Byzantine participants, who modify the uploaded parameters during training to achieve the purpose of preventing the convergence of the global model.

Our research goal is that each participant cannot obtain sensitive information (such as training samples) of other participants during the training process of federated learning, detect Byzantine participants and exclude them from the training process, and ensure the availability of the global model.

III. OUR PROPOSED SCHEME

A. Architecture Overview

Unlike traditional FL which relies on a vulnerable central server, PBDFL adopts a decentralized architecture (see *Figure 1*). In each round, participants are dynamically divided into Training Nodes (who train local models) and Referee Nodes (who validate models and aggregate results). This separation of duties mitigates the risk of single-point failure and Byzantine attacks.

B. Byzantine-Robust Decentralized Training Mechanism

The core mechanism involves identifying malicious nodes through cross-validation and excluding them via a reputation system. This process consists of two stages: reliability calculation and weighted aggregation.

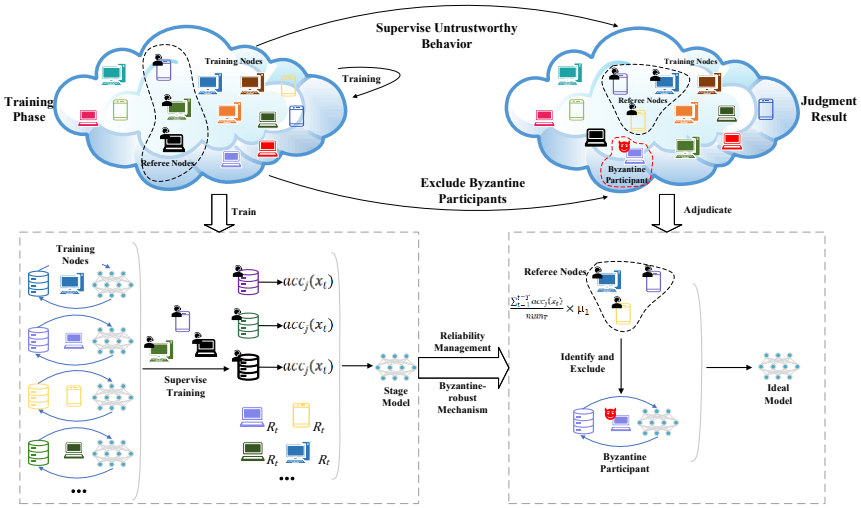

Figure 1: The overview of PBDFL

a) Reliability calculation

Referees validate uploaded models using their local datasets (assuming IID distribution). Let x_t be the model from training node t. A referee j records the accuracy $acc_j(x_t)$. If this accuracy falls below a dynamic threshold, the node is flagged as "untrustworthy." The threshold is defined as:

$$Threshold_j = \frac{\sum_{t=1}^{t=T} acc_j(x_t)}{num_T} \times \mu_1 \qquad (1)$$

Where num_T is the number of training nodes. μ_1 is used to adjust the size of $Threshold_j$. Based on the actual situation and experimental tests, the value of μ_1 ranges from 0.2 to 0.6.

Based on these validations, the jury calculates a Reliability Score R_t for each training node:

$$R_t = [\frac{\sum_{j=1}^{j=J} acc_j(x_t)}{num_J}]^{\mu_2} \qquad (2)$$

Where μ_2 is a constant greater than 1, amplifies the gap between honest and malicious nodes, ensuring Byzantine participants receive negligible weights.

In PBDFL, 20% of participants serve as referees. They are alternately selected before the training rounds reach $\frac{num_{iters}}{100}$. Subsequently, selection becomes probabilistic based on the history of untrustworthy behaviors, calculated as follows:

$$P_i = \frac{\frac{1}{unrelia_i+1}}{\sum\frac{1}{unrelia_i+1}} \qquad (3)$$

This mechanism naturally phases out Byzantine nodes from the referee committee over time.

Model aggregation. After calculating the reliability of each training node, the jury uses the reliability and the model of each training node to aggregate the ideal model of this round of training

$$x_* = \frac{\sum_{t=1}^{t=T} R_t x_t}{\sum_{t=1}^{t=T} R_t} \qquad (4)$$

This ensures that high-quality models dominate the global update while malicious contributions are suppressed.

Algorithm 1 shows how the Byzantine-robust decentralized training mechanism aggregates the details of the ideal model in each round of training.

Algorithm 1. Aggregate ideal model
Input: Local datasets$\{D_1, D_2, \dots, D_n\}$ **Output**: Ideal model x_*
1: Byzantine-robust Decentralized Training Mechanism: Select M participants as referee nodes J (e.g., Eqn. (3)) and the rest as training nodes T. 2: Training node t: 3: Use local dataset for training to obtain model parameters x_t. 4: Send x_t to all referee nodes. 5: Referee nodes j: 6: **for** (each $t \in T$)**do** 7: Use partial local datasets to calculate the model accuracy $acc_j(x_t)$ of training node t. 8: Send $acc_j(x_t)$ to others referee node. 9: **end for** 10: Jury, i.e., all referee nodes: 11: **for** (each $t \in T$)**do** 12: Calculate the reliability R_t of training node t (e.g., Eqn. (2)); 13: **end fo**r 14: Aggregate ideal model x_* (e.g., Eqn. (4)); 15: Send x_* to all training nodes.

All referee nodes in the jury perform the same reliability calculation as well as the model aggregation process. If all referee nodes are honest, the ideal model computed by them is exactly the same. After calculating the ideal model, the referee nodes broadcast it to all the training nodes, and each training node receives the ideal model from all the referee nodes. To prevent Byzantine participants from modifying the ideal model when they are selected as referee nodes, the training node compares the received ideal models and selects the exact same model among them as the ideal model.

b) Exclude byzantine participants

The exclusion of Byzantine participants is achieved through the participant management mechanism. In the setting of the participant management mechanism, after each round of training, the jury will judge whether a participant is a Byzantine participant based on the number of untrustworthy behaviors. When the number of untrustworthy behaviors of a participant reaches the $Threshold_{byz}$, the jury will consider this participant as a Byzantine participant and remove it from the training process. After selecting the referee nodes for the next round of training, the jury sends the untrustworthy behavior number of all participants to them. As shown in *Figure 1*, participants who are judged to be Byzantine participants will be identified and excluded from the subsequent training process. The setting of threshold is proportional to the number of referee nodes and the number of training rounds, and its calculation formula is as follows:

$$Threshold_{byz} = \mu_3 \times num_J \times num_{iters} \qquad (5)$$

Where $num_J$the number of referee nodes is, num_{iters} is the number of training rounds. μ_3 is used to adjust the size of $Threshold_{byz}$. Based on the practical situation and experimental tests, the value of μ_3 ranges from 0.001 to 0.01. When the number of referee nodes and training rounds is high, there are more records of untrustworthy behaviors. The value of $Threshold_{byz}$ is higher, which can prevent honest participants from being judged as Byzantine participants. Conversely, when the number of referee nodes and training rounds is low, the value of $Threshold_{byz}$ is lower, which can prevent the failure to identify Byzantine participants.

Through the Byzantine-robust decentralized training mechanism, PBDFL can achieve the following functions:

- Decentralization: PBDFL realizes the decentralized design through the Byzantine-robust decentralized training mechanism, and no longer requires the central server to be responsible for updating the global model, so it will not be attacked by the untrusted central server.
- K-fold cross validation: During the training process, each referee node uses portion of the local data set to validate the training node's model, serving as the validation set. As the judge nodes change, the validation set also changes, achieving K-fold cross-validation in this setting.
- Byzantine robustness: PBDFL tests and aggregates the training node model through the Byzantine-robust decentralized training mechanism, which can reduce the proportion of the model uploaded by Byzantine participants in the aggregation model. During the training process, the referee node records the untrustworthy

behavior of the participant, and judges the participant whose untrustworthy behavior number reaches the threshold as a Byzantine participant, and excludes the participant from the training process, thereby achieving Byzantine robustness.

C. Privacy-preserving aggregation Mechanism

In this section, we discuss the privacy protection scheme designed for the proposed decentralized federated learning system. Common privacy protection methods for decentralized systems include secure multiparty computation (SMC). However, the design process of SMC is often complex and requires high computational overhead, making it inefficient.

In many previous federated learning studies, homomorphic encryption was used to protect the privacy of participants. However, homomorphic encryption involves significant computational costs, which can increase the computational overhead of federated learning. Additionally, in a decentralized federated learning system with a full homomorphic encryption design, each participant possesses both the private and public keys, which means they can access the plaintext of the parameters uploaded by other participants. This makes it vulnerable to inference attacks from internal participants, posing a privacy threat.

The privacy protection scheme based on differential privacy can effectively mitigate internal privacy threats with relatively low additional computational overhead.[31] However, this approach requires adding some amount of noise to the transmitted data, which can impact the accuracy of the global model. Meanwhile, differential privacy cannot prevent eavesdroppers outside the federated learning system from obtaining models with availability.

Given these considerations, we propose a privacy-preserving aggregation mechanism that combines differential privacy and homomorphic encryption. We classify the intermediate data that needs protection and apply different privacy protection methods accordingly. This hybrid privacy protection mechanism leverages the advantages of both homomorphic encryption and differential privacy, ensuring that malicious participants within the decentralized federated learning system cannot threaten the privacy of legitimate participants. Ultimately, this approach achieves the privacy protection and global model accuracy requirements with lower computational overhead.

The PPA that we have designed is as follows. The forward propagation process of convolution layer and full connection layer in the model trained by federated learning can be defined as:

$$a_{i+1} = f(W_i a_i + b_i) \tag{6}$$

Where a_i represents the input vector at layer i in the model, $W_i a_i + b_i$ represents the affine transformation performed in the i-th layer, and f represents nonlinear operation. We divide the affine transformation into two parts, W as the weight unit and b as the bias unit.

To minimize computation overhead, PPA applies Differential Privacy to the high-dimensional weights (W) and CKKS Homomorphic Encryption to the low-dimensional biases (b). After decrypting \$b\$ with the private key S_k, the node computes local gradients and clips the gradients of W as follows:

$$g_i = \frac{g_i}{\max\left(1, \frac{|g_i|}{C}\right)} \tag{7}$$

The purpose of gradient clipping is to limit the L1 norm of the gradient to a certain range, so that the sensitivity can be calculated, which is used to calculate the size of the added Laplace noise. The training node uses the clipped gradient to update the weight units W, and then adds Laplace noise:

$$\widetilde{W_i} = W_i + Lap(\Delta f/\varepsilon) \tag{8}$$

After the training node applies differential privacy to W, the public key P_k of CKKS is used to encrypt the bias unit to obtain Enc (b_i). After completing the DP and HE operations, the training nodes send $\widetilde{W_i}$ and Enc (b_i) to all the referee nodes. Upon receiving the model parameters, the referee nodes first need to decrypt the bias unit Enc (b_i), and then load the plaintext of bias unit b and weight unit W into the model. They use part of the local data to test the accuracy of the model, and then assign reliability to each training node according to the accuracy calculated by the other referee nodes. Finally, the jury aggregates the ideal model according to the reliability and the parameters uploaded by each training node. To ensure the security of the model during transmission, the model parameters used in the aggregation are not decrypted. Therefore, the bias unit b in the model parameters sent to each participant below is the ciphertext of CKKS. The CKKS is homomorphic, and the decrypted plaintext after calculation is almost the same as the plaintext calculated directly. The specific details are shown in *Figure 2*.

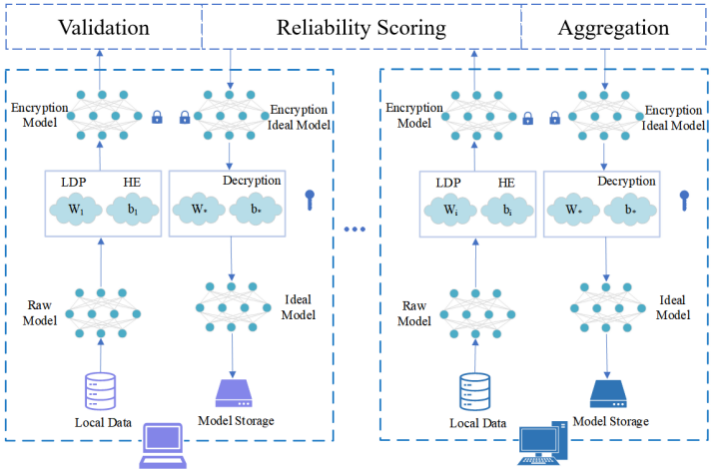

Figure 2: Details of Byzantine-Robust Decentralized Training Mechanism and Privacy-Preserving Aggregation Mechanism

Under this privacy protection setting, even if the adversary intercepts the model parameters, without collusion with the participants in PBDFL, the adversary cannot obtain the private key S_k of the CKKS, cannot obtain all the plaintext model parameters, and cannot obtain a usable model.

D. Security analysis

Table 1 compares PPA with schemes [23], [32] and [33]. While [23] uses HE, it remains vulnerable to internal-external collusion. Scheme [32] combines DP and secure multiparty computation but suffers from high computational overhead. Although [33] resists collusion via DP, the required noise significantly degrades model accuracy. In contrast, our PPA integrates HE and DP to effectively resist collusion, maintaining high model stability with low computational cost.

Table 1: Security Comparison with Existing Works

Scheme	Privacy protection technology	Resist internal inference attacks	Resist internal and external collusion	Low additional computation	High global model accuracy
[23]	HE	√	×	×	√
[32]	SMC+DP	√	√	×	√
[33]	DP	√	√	√	×
PPA	HE+DP	√	√	√	√

We evaluate the privacy of PBDFL under two threat models: (1) external eavesdropping without collusion, and (2) collusion between internal participants and external attackers. Our analysis is based on the standard assumption that the underlying CKKS homomorphic encryption scheme satisfies Indistinguishability under Chosen Plaintext Attack (IND-CPA) security [11]. This ensures that without the secret key, computationally bounded adversaries cannot distinguish between the ciphertexts of different messages.

Based on this cryptographic foundation and the hybrid mechanism of DP, we analyze the specific scenarios:

- Eavesdropping (No Collusion): An external adversary without the private key cannot decrypt the bias units due to the IND-CPA security of CKKS. Since the weights are also perturbed by DP, the adversary cannot reconstruct the original model parameters from the intercepted updates.
- Collusion Attacks: Even if malicious participants collude with external attackers to share the private key and decrypt the bias units, the weight parameters remain protected by Local Differential Privacy (LDP). As proven in [6], the privacy budget ε ensures that the adversary cannot infer the presence of specific training data with high confidence, effectively mitigating privacy leakage even under collusion.

IV. PERFORMANCE EVALUATION

A. Experiment settings

- Attack: The Same-value attack, adversaries replace all model parameters with a constant c=0 and Gaussian attack, adversaries inject standard normal noise $N(0,1)$ into the parameters.
- Datasets and Environment: We evaluated PBDFL using MNIST, FashionMNIST, and EMNIST datasets. MNIST and FashionMNIST contain 60,000 training and 10,000 testing samples, while EMNIST contains 240,000 and 40,000, respectively. All experiments were implemented in Python on an AMD Ryzen 7 5800H CPU.
- Experimental Setup: We simulated 20 participants with IID data. To test robustness, we introduced 2, 4, 6, and

8 colluding Byzantine participants. When selected as referee nodes, these adversaries assign high accuracy scores to other Byzantine nodes. We trained a softmax regression model (batch size 32) with a learning rate of 0.001 for MNIST/EMNIST and 0.0001 for FashionMNIST.

- Hyperparameter settings: In each round, 20% of participants (4 nodes) act as referees, using 20% of their local data for validation. We set the system parameters as μ_1=0.5, μ_2 =10 and μ_3 =0.0125, resulting in a reputation threshold of 100 for exclusion. The LDP privacy budget is set to ε =10.

B. Robustness comparison

In the robustness experiments, we compared the PBDFL, CMFL-selection I[28], which is suitable for FL scenarios with Byzantine attacks, RSA[21], Krum[14], Median[15], and federated average (FedAvg) algorithms. For FedAvg, the central server aggregates the model parameters uploaded by participants on average, which is the most primitive approach federated learning aggregation. With the exception of our proposed PBDFL, the above algorithms do not have privacy protection functionality.

Robustness experiments conducted on the MNIST, EMNIST, and FashionMNIST datasets under same-value attacks demonstrate that the PBDFL algorithm exhibits excellent robustness and stability across varying proportions of Byzantine attacks, outperforming the baseline algorithms overall. As shown in *Figure 3*. Specifically, when the proportion of Byzantine nodes is low (b=10), the accuracy of PBDFL is comparable to that of the best-performing RSA and CMFL algorithms. As the attack proportion increases (b $\geqslant$20%), the superiority of PBDFL becomes evident, as it achieves the highest final accuracy across all three datasets (MNIST$\approx$91%, EMNIST$\approx$92%, and FashionMNIST > 82%). In contrast, the performance of Median and FedAvg degrades significantly as attack intensity increases, and RSA performs poorly on the FashionMNIST dataset. Notably, PBDFL is the only scheme among the compared algorithms that incorporates privacy protection capabilities, successfully ensuring high robustness without sacrificing model performance.

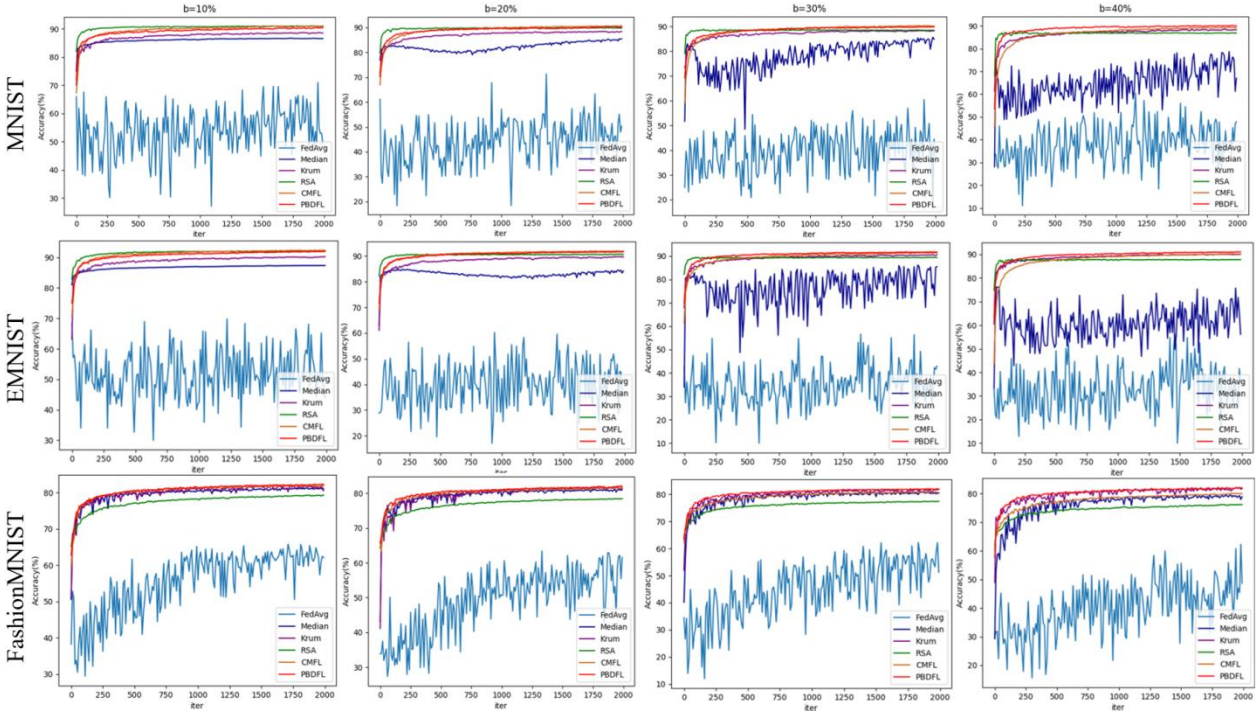

Figure 3: Comparison of the robustness of each algorithm under the same-value attack launched by various proportions of Byzantine participants on the MNIST, EMNIST, and FashionMNIST datasets

Robustness experiments conducted on the MNIST, EMNIST, and FashionMNIST datasets under Gaussian attacks demonstrate that the PBDFL algorithm exhibits robustness comparable to, or even superior to, state-of-the-art non-private defense algorithms (such as RSA), while simultaneously guaranteeing data privacy. As shown in Figure 4. Although RSA exhibits high accuracy in certain low-attack scenarios or during the early stages of training, PBDFL leverages its ability to detect and exclude Byzantine nodes early on. Consequently, its accuracy gradually improves as training progresses, eventually reaching parity with RSA (approximately 90% on MNIST and EMNIST). Notably, under a high attack proportion (b=40%) on the FashionMNIST dataset, PBDFL achieved the highest accuracy among all algorithms (82.01%). In contrast, FedAvg is rendered almost ineffective under attack, while the overall performance of

Krum and Median is slightly inferior to that of PBDFL. Crucially, PBDFL overcomes Krum's limitation of requiring prior knowledge of the number of attackers and addresses the privacy leakage issues associated with RSA and CMFL, which necessitate the transmission of parameters in plaintext.

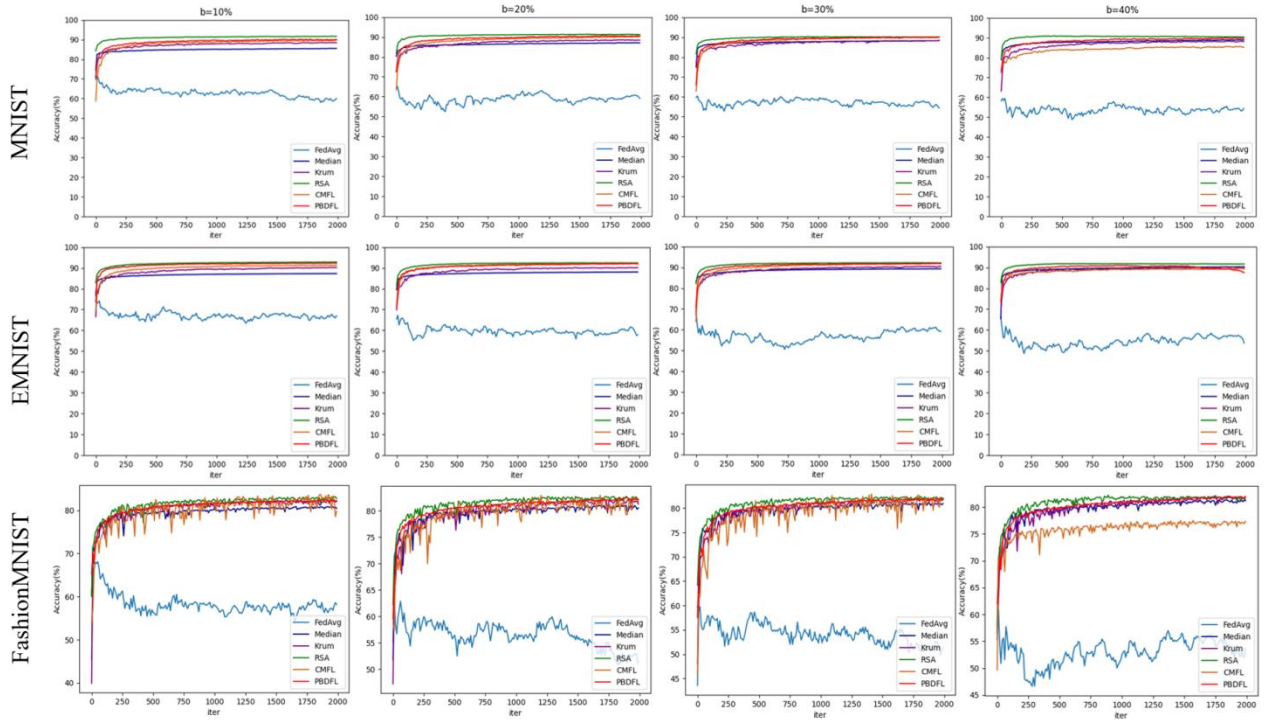

Figure 4: Comparison of the robustness of each algorithm under the Gaussian attack launched by various proportions of Byzantine participants on the MNIST, EMNIST, and FashionMNIST dataset

In the experimental results on the three datasets, our proposed PBDFL had excellent Byzantine robustness compared with the other four algorithms under the premise of adding the privacy protection function. In all the Byzantine robustness experiments, PBDFL achieved the highest or near-highest accuracy in most cases, and detected and excluded Byzantine participants in the early stages of training. PBDFL had good prevention and detection capabilities against Byzantine participants, which greatly reduced their opportunities for malicious behavior, thereby accelerating the convergence speed of the model and improving the accuracy of the global model. When algorithms other than PBDFL aggregate model parameters, the central server obtains the parameter plaintext uploaded by participants, which results in a great threat to the privacy of participants. Krum needs to know the number of Byzantine participants in advance, which is almost impossible to achieve in a practical situation.

C. Performance and Functional comparison

We compared PPA with pure CKKS HE on the MNIST dataset. PPA significantly outperforms CKKS, reducing the average training time per round from 28.73s to 0.97s and the model parameter size from 5871.71 MB to 11.38 MB. While PBDFL incurs a marginal cost compared to unencrypted baselines (e.g., RSA, Krum), this trade-off is negligible given the privacy guarantees provided. To illustrate the functional advantages of PBDFL, we compare the functionality of PBDFL with four federated schemes, including RSA, PBFL, DBPFL and CMFL. As shown in Table 2, we propose PBDFL, protects participants' privacy through the design of privacy-preserving aggregation mechanism. We also design a Byzantine-robust decentralized training mechanism to achieve decentralized federated learning and reduce the proportion of Byzantine participants in the global model, as well as to detect which participants are Byzantine, thus achieving Byzantine robustness.

Table 2: Functionality Comparison with Existing Works

Scheme	Byzantine robustness	Privacy protection	Decentralization	Byzantine participant detection
RSA[21]	√	×	×	×
PBFL [23]	√	√	×	×
DPBFL[33]	√	√	×	×
CMFL [28]	√	×	√	×
PBDFL	√	√	√	√

V. CONCLUSION

In this paper, we presented PBDFL, a novel decentralized federated learning scheme that simultaneously ensures Byzantine robustness and data privacy. By implementing a decentralized training mechanism with reputation-based validation, PBDFL successfully identifies and excludes malicious nodes, overcoming the limitations of traditional robust aggregation rules. Furthermore, our proposed hybrid privacy-preserving aggregation mechanism balances computational overhead with security, effectively resisting internal collusion and external threats. Theoretical proofs and experimental results on multiple datasets verify that PBDFL achieves superior accuracy and stability compared to existing methods, such as RSA and Krum, particularly in adversarial environments. In future work, we plan to extend this framework to handle non-IID data distributions more effectively and explore adaptive privacy budget allocation to further optimize the trade-off between model utility and privacy protection.

Interest Conflicts

The authors declare that there is no conflict of interest concerning the publishing of this paper.

Funding Statement

No funding was received for conducting this study.

VI. REFERENCES

[1] Nguyen M, Yan W Q. Temporal colour-coded facial-expression recognition using convolutional neural network[M]//International Summit Smart City 360°. Cham: Springer International Publishing, 2021: 41-54.

[2] Su Y, Huang C, Yin W, et al. Diabetes Mellitus risk prediction using age adaptation models[J]. Biomedical Signal Processing and Control, 2023, 80: 104381.

[3] Jia B, Zhang X, Liu J, et al. Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT[J]. IEEE Transactions on Industrial Informatics, 2021, 18(6): 4049-4058.

[4] Shokri R, Stronati M, Song C, et al. Membership inference attacks against machine learning models[C]//2017 IEEE symposium on security and privacy (SP). IEEE, 2017: 3-18.

[5] Song C, Ristenpart T, Shmatikov V. Machine learning models that remember too much[C]//Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security. 2017: 587-601.

[6] Dwork C, Roth A. The algorithmic foundations of differential privacy[J]. Foundations and trends® in theoretical computer science, 2014, 9(3-4): 211-487.

[7] Ku H, Susilo W, Zhang Y, et al. Privacy-preserving federated learning in medical diagnosis with homomorphic re-encryption[J]. Computer Standards & Interfaces, 2022, 80: 103583.

[8] Zeng Y, Lin Y, Yang Y, et al. Differentially private federated temporal difference learning[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 33(11): 2714-2726.

[9] Lian W, Zhang Y, Chen X, et al. IPCADP-Equalizer: An Improved Multibalance Privacy Preservation Scheme against Backdoor Attacks in Federated Learning[J]. International Journal of Intelligent Systems, 2023, 2023(1): 6357750.

[10] Truex S, Liu L, Chow K H, et al. LDP-Fed: Federated learning with local differential privacy[C]//Proceedings of the third ACM international workshop on edge systems, analytics and networking. 2020: 61-66.

[11] Xu G, Li H, Zhang Y, et al. Privacy-preserving federated deep learning with irregular users[J]. IEEE Transactions on Dependable and Secure Computing, 2020, 19(2): 1364-1381.

[12] Aono Y, Hayashi T, Wang L, et al. Privacy-preserving deep learning via additively homomorphic encryption[J]. IEEE transactions on information forensics and security, 2017, 13(5): 1333-1345.

[13] Yuan K, Wu Z, Ling Q. A Byzantine-resilient dual subgradient method for vertical federated learning[C]//ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022: 4273-4277.

[14] Blanchard P, El Mhamdi E M, Guerraoui R, et al. Machine learning with adversaries: Byzantine tolerant gradient descent[J]. Advances in neural information processing systems, 2017, 30.

[15] Xie C, Koyejo O, Gupta I. Zeno: Byzantine-suspicious stochastic gradient descent[J]. arXiv preprint arXiv:1805.10032, 2018, 24.

[16] Chen Y, Su L, Xu J. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent[J]. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2017, 1(2): 1-25.

[17] Shokri R, Shmatikov V. Privacy-preserving deep learning[C]//Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. 2015: 1310-1321.

[18] Hamm J, Cao Y, Belkin M. Learning privately from multiparty data[C]//International Conference on Machine Learning. PMLR, 2016: 555-563.

[19] Li Y, Li H, Xu G, et al. Efficient privacy-preserving federated learning with unreliable users[J]. IEEE Internet of Things Journal, 2021, 9(13): 11590-11603.

[20] Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 1-19.

[21] L. Li, W. Xu, T. Chen, G. B. Giannakis, Q. Ling, RSA: byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets, in: Proceedings of the AAAI Conference on Artificial Intelligence,2019, pp. 1544–1551.

[22] Cao, X., Jia, J., & Gong, N. Z. (2021). Provably Secure Federated Learning against Malicious Clients. Proceedings of the AAAI Conference on Artificial Intelligence, 35(8), 6885-6893. https://doi.org/10.1609/aaai.v35i8.16849

[23] Ma X, Zhou Y, Wang L, et al. Privacy-preserving Byzantine-robust federated learning[J]. Computer Standards & Interfaces, 2022, 80: 103561.
[24] Zhao L, Jiang J, Feng B, et al. Sear: Secure and efficient aggregation for byzantine-robust federated learning[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 19(5): 3329-3342.
[25] Hu C, Jiang J, Wang Z. Decentralized federated learning: A segmented gossip approach[J]. arXiv preprint arXiv:1908.07782, 2019.
[26] Hegedűs, I., Danner, G., Jelasity, M. (2019). Gossip Learning as a Decentralized Alternative to Federated Learning. In: Pereira, J., Ricci, L. (eds) Distributed Applications and Interoperable Systems. DAIS 2019. Lecture Notes in Computer Science(), vol 11534. Springer, Cham. https://doi.org/10.1007/978-3-030-22496-7_5
[27] Zhao J, Zhu H, Wang F, et al. PVD-FL: A privacy-preserving and verifiable decentralized federated learning framework[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 2059-2073.
[28] Che C, Li X, Chen C, et al. A decentralized federated learning framework via committee mechanism with convergence guarantee[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 4783-4800.
[29] Cheon, J.H., Kim, A., Kim, M., Song, Y. (2017). Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Takagi, T., Peyrin, T. (eds) Advances in Cryptology – ASIACRYPT 2017. ASIACRYPT 2017. Lecture Notes in Computer Science(), vol 10624. Springer, Cham. https://doi.org/10.1007/978-3-319-70694-8_15
[30] Ou W, Zeng J, Guo Z, et al. A homomorphic-encryption-based vertical federated learning scheme for rick management. Computer Science and Information Systems, 2020, 17(3): 819–834
[31] Tian H, Zeng C, Ren Z, et al. Sphinx: Enabling privacy-preserving online learning over the cloud[C]//2022 IEEE Symposium on Security and Privacy (SP). IEEE, 2022: 2487-2501.
[32] Truex S, Baracaldo N, Anwar A, et al. A hybrid approach to privacy-preserving federated learning[C]//Proceedings of the 12th ACM workshop on artificial intelligence and security. 2019: 1-11.
[33] Ma X, Sun X, Wu Y, et al. Differentially private byzantine-robust federated learning[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 3690-3701.