

Robust Decentralized Federated Learning with Partial Weight Sharing Against Weight Poisoning Attacks

Reem Nabha

*Samovar, Télécom SudParis
Institut Polytechnique de Paris
91120 Palaiseau, France
0009-0005-0882-6370*

Anis Laouiti

*Samovar, Télécom SudParis
Institut Polytechnique de Paris
91120 Palaiseau, France
0000-0002-1287-4739*

Abed Ellatif Samhat

*Faculty of Engineering - CRSI
Lebanese University
Hadath Campus, Lebanon
0000-0002-1137-621X*

Abstract—Federated Learning (FL) enables collaborative model training while preserving data locality. However, collaborative learning remains vulnerable to malicious model updates, including weight poisoning attacks. This paper presents a decentralized federated learning framework that relies on partial sharing of dense-layer parameters in a peer-to-peer manner by eliminating reliance on a central server for aggregation and examines its resilience to weight poisoning attacks. Sharing partial parameters reduces the amount of information exposed during communication and helps contain the influence of malicious updates. The proposed framework is applied to chest X-ray image classification as a representative case study to evaluate its effectiveness; however, the framework can be applied to different application domains. The experiments conducted using three dataset sizes and two sharing ratios of parameters (25% and 100%) show that decentralized learning remains relatively stable performance under persistent poisoning, with robustness improving as dataset size increases. The results show the decentralization together with a controlled partial sharing can offer a balance between privacy and accuracy, while a lightweight median-based clipping further enhances stability. These findings highlight decentralization as an effective architectural defense against weight poisoning attacks.

Index Terms—Decentralized Federated Learning, Weight Poisoning Attack, Medical Image Classification, Privacy-Preserving AI, Partial Model Sharing, Secure Collaborative Learning

I. INTRODUCTION

The rapid spread of machine learning has enabled notable improvements in different fields, including medical image analysis and intelligent decision support systems. However, aggregating sensitive data in a centralized location is required in centralized training methods, raising serious concerns about privacy, regulatory compliance, and data ownership. Federated Learning addresses these problems by allowing collaborative model training while preserving the locality of raw data for each participant [25].

With all the advantages of Federated Learning (FL), standard FL architectures mainly rely on a centralized server for aggregating data, which presents a critical vulnerability [8],

[9]. Malicious actors can exploit this architecture by sending altered model updates, leading to poisoning attacks that degrade performance or cause targeted misclassifications [10]. Among these threats, weight poisoning attacks are particularly effective, where attackers directly alter shared parameters to influence the learning process. Previous studies have shown that centralized federated learning remains vulnerable to such attacks, even when applying robust aggregation mechanisms [12].

To address these limitations, this paper explores a decentralized FL framework that eliminates centralized aggregation and mitigates the effects of malicious attacks through peer-to-peer collaborative training and partial weight sharing. Participants share only specific portions of the final dense layers, minimizing information exposure while preserving the benefits of collaboration. By distributing the aggregation process among participants, the framework reduces centralized dependency and limits the amount of model information exchanged.

While our framework applies to various domains and fields, including finance and industrial IoT, we assess its effectiveness on COVID-19 chest X-ray classification as a use case. Experiments conducted across multiple dataset sizes demonstrate that decentralized learning maintains stable performance under weight poisoning attacks [12], particularly when combined with lightweight threshold-based defenses.

The main contributions of this paper are summarized as follows:

- We propose a decentralized federated learning framework with partial weight sharing for privacy-preserving collaborative training.
- We implement and evaluate weight poisoning attacks in a peer-to-peer federated setting.
- We integrate a lightweight defense based on median-MAD clipping and update stabilization.
- We conduct extensive experiments on chest X-ray images across multiple dataset scales to validate robustness.

The remainder of this paper is organized as follows. Section II reviews related work. Section III presents the proposed framework. Section IV describes the threat model. Section V introduces the defense mechanism. Section VI details the experimental setup, followed by results in Section VII. Section VIII concludes the paper.

II. RELATED WORK

A. Federated Learning in Healthcare

Federated learning has been extensively investigated in medical imaging and healthcare to enable collaborative model training while protecting patient privacy. Studies demonstrate privacy-preserving FL applications in radiological data, medical diagnostics, and multi-institutional collaboration, while highlighting challenges such as data heterogeneity and non-IID distributions [1]. Recent surveys also examine privacy-preserving healthcare architectures, emphasizing distributed learning, secure data sharing, and regulatory constraints. A comprehensive analysis in [11] discusses federated learning as a framework for collaborative health intelligence without direct data exchange.

B. Poisoning Attacks in Federated Learning

One of the most serious security issues in FL is poisoning attacks, in which malicious actors actively alter local training data or updates to degrade overall performance. Data poisoning, such as label flipping and targeted perturbations, has significantly impacted federated models. Label-flipping attack strategies have been studied extensively to understand their impact and defense limitations [2]. Model- and data-level attacks, their persistence, and their effectiveness based on attack technique are identified in a survey on poisoning in FL [3]. Recent work continues to develop novel poisoning and analytical frameworks that define these risks [4].

C. Defense Mechanisms Against Poisoning

Many works suggest strong aggregation and anomaly detection methods to prevent poisoning attacks. Krum, Median, and Trimmed Mean are examples of Byzantine-resilient aggregation rules that aim to reduce the impact of outliers in update aggregation [5]. Defenses based on similarity and clustering have also been proposed to detect malicious updates and dynamically modify client weights. To enhance robustness, other strategies for defense concentrate on detecting poisoned data points or filtering suspicious updates prior to aggregation [6].

D. Decentralized and Partial Sharing in Federated Learning

Researchers have investigated decentralized, or peer-to-peer, FL as an alternative to centralized aggregation to improve resilience and eliminate single points of failure. By enabling direct participant interaction without a trusted server, decentralized FL can improve privacy and reliability [7]. Researchers have also explored partial parameter sharing to reduce communication overhead and limit information exposure,

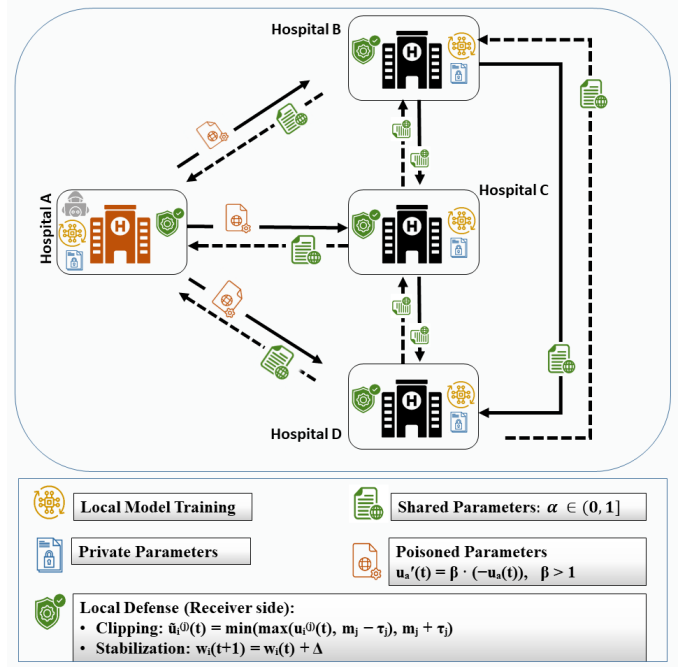


Fig. 1. Decentralized federated learning with partial dense-layer sharing under weight poisoning. Participants exchange selected dense parameters ($\alpha \in (0, 1]$) in a peer-to-peer manner, while receiver-side clipping and stabilization mitigate adversarial updates.

although its security implications in malicious environments remain under investigation.

[22] presented BrainTorrent, a peer-to-peer FL system designed for medical applications, where clients communicate directly without relying on a central server. However, BrainTorrent requires participants to exchange the full model, whereas our framework restricts sharing to selected portions of the final dense layers.

In contrast to existing clipping, robust aggregation, and personalized FL approaches [13], [15], [23], [24], our work investigates partial dense-layer sharing in a decentralized peer-to-peer setting under weight poisoning attacks.

III. SYSTEM MODEL

The framework is designed to enable collaborative training without relying on a central server. Through peer-to-peer collaboration and partial parameter exchange, the framework improves privacy, resilience, and scalability while focusing on binary medical image classification tasks (such as COVID vs. Normal).

A. Participants and Local Models

We consider a system with N participants $\{C_i\}_{i=1}^N$, where each participant holds a private local dataset D_i containing sensitive medical images. Each participant maintains an identical neural network architecture composed of a feature extraction backbone and final dense layers for classification.

- $w_i^{private}$: private parameters (feature extraction layers) that remain strictly local.

- w_i^{shared} : shareable parameters (final dense layers) selected for collaborative learning.

Only the dense layers are eligible for sharing, as they encode high-level task representations, while earlier layers retain low-level data-dependent features that could increase reconstruction risks.

B. Decentralized Training Protocol

Training proceeds over synchronous communication rounds indexed by $t = 1, 2, \dots, T$. During each round, every participant performs local training on its private dataset by minimizing a standard classification loss [19]:

$$F_i(w_i) = \frac{1}{|D_i|} \sum_{(x,y) \in D_i} L(w_i, x, y) \quad (1)$$

where $L(\cdot)$ denotes the binary cross-entropy loss.

After local optimization, each participant extracts the shareable parameters from its final dense layers and constructs a sparse partial update $u_i(t)$ by randomly selecting a proportion $\alpha \in (0, 1]$ of parameter values. Unselected entries are masked to zero, ensuring that only partial information is disclosed.

Participants then engage in peer-to-peer communication, directly exchanging their partial updates without centralized coordination. In our experiments, participants operate in a fully connected peer-to-peer topology, where each participant exchanges its selected parameters with all other participants during every communication round. Upon receiving updates from peers, each participant computes an aggregated update using dataset-size weighted averaging [18]:

$$\bar{u}(t) = \sum_{j=1}^N \frac{|D_j|}{\sum_{k=1}^N |D_k|} u_j(t) \quad (2)$$

The aggregated parameters replace the shared dense-layer weights, while private layers remain unchanged. This process repeats iteratively across communication rounds.

C. Comparison with Centralized Federated Learning

Participants in traditional centralized FL share full model updates to a central server for aggregation, creating a single point of failure, which leaves the system more susceptible to poisoning attacks [25]. However, the proposed decentralized framework eliminates centralized aggregation and limits peer-to-peer communication to partial dense-layer parameters.

By distributing model integration and restricting parameter exchange, this architectural approach avoids centralized aggregation and limits the parameters exposed during peer-to-peer collaboration [26].

IV. THREAT MODEL AND WEIGHT POISONING ATTACK

In this section, we discuss the centralized federated learning vulnerability based on previous literature [25], outline the adversarial assumptions considered in our work, and describe the weight poisoning attack tested within our decentralized federated learning system (see Fig. 1).

A. Threat Model

We consider an adversarial setting in which one or more participants behave maliciously during collaborative training. Specifically, we assume a Byzantine adversary controlling a participant C_a , which aims to degrade global model performance by injecting poisoned updates while otherwise following the protocol. The attacker performs a weight poisoning attack by manipulating its shared update at communication round t . Let $u_a(t)$ denote the legitimate partial update generated by the attacker [27]. The poisoned update is constructed as [17]:

$$u'_a(t) = \beta \cdot (-u_a(t)), \quad \beta > 1, \quad (3)$$

where β is a scaling factor that amplifies the adversarial impact, and sign inversion is applied to maximize disruption.

The private datasets and parameters of other participants are not accessible to the attacker. The attacker is considered to hold full control over its local model and training data. Peer-to-peer communication takes place in the absence of a central server. Previous studies have demonstrated that such poisoned updates are directly aggregated by the server in centralized federated learning, possibly corrupting the global model. However, by distributing aggregation among participants, our decentralized approach reduces single-point risk.

Furthermore, since only specific parts of the final dense layers are shared, we believe that attackers cannot alter private layers, limiting the exposure of sensitive feature representations.

B. Robust Aggregation via Median-MAD Clipping

To mitigate poisoning attacks, each participant applies a coordinate-wise robust aggregation strategy based on the median and Median Absolute Deviation (MAD) [23], [24].

Let $u_i^{(j)}(t)$ denote the j^{th} shared parameter received from participant C_i at round t . The coordinate-wise median is computed as [14]:

$$m_j = \text{median} \left(u_i^{(j)}(t)_{i=1}^N \right), \quad (4)$$

and the MAD is defined as [15]:

$$\text{MAD}_j = \text{median} \left(|u_i^{(j)}(t) - m_j| \right). \quad (5)$$

A clipping threshold is then determined by:

$$\tau_j = k \cdot \text{MAD}_j, \quad (6)$$

The threshold τ_j defines a coordinate-wise adaptive clipping bound, scaled by the robustness coefficient k .

Each received update is clipped coordinate-wise [15]:

$$\tilde{u}_i^{(j)}(t) = \min \left(\max(u_i^{(j)}(t), m_j - \tau_j), m_j + \tau_j \right). \quad (7)$$

This ensures that each parameter remains within the bounded interval $[m_j - \tau_j, m_j + \tau_j]$, effectively suppressing extreme deviations on a per-coordinate basis.

The clipped updates are then aggregated using weighted averaging proportional to local dataset sizes:

$$\bar{u}(t) = \sum_{i=1}^N \frac{|D_i|}{\sum_{k=1}^N |D_k|} \tilde{u}_i(t). \quad (8)$$

C. Update Stabilization via Delta Norm Clipping

To further limit adversarial influence, we introduce an additional stabilization mechanism based on ℓ_2 clipping [23].

Let $w_i(t)$ denote the current shared parameters of participant C_i . The update difference is computed as:

$$\Delta = \bar{u}(t) - w_i(t). \quad (9)$$

The update vector Δ represents the adjustment introduced by aggregation relative to the participant's current model.

The update is then constrained as [16]:

$$\Delta \leftarrow \Delta \cdot \min \left(1, \frac{\lambda}{\|\Delta\|_2} \right) \quad (10)$$

The ℓ_2 norm $\|\Delta\|_2$ measures the overall magnitude of the update, while λ defines a global bound that limits the total model shift.

The stabilized update is finally applied as:

$$w_i(t+1) = w_i(t) + \Delta. \quad (11)$$

This prevents abrupt parameter shifts caused by adversarial scaling.

Together, coordinate-wise Median-MAD clipping and global ℓ_2 norm clipping provide complementary protection by constraining both per-parameter deviations and overall update magnitude.

Moreover, updates are applied exclusively to indices explicitly shared by at least one participant, preventing zero-injection and unintended overwriting of private parameters.

V. DEFENSE MECHANISM

The Weight Guard defense combines the Median-MAD clipping and delta-norm stabilization described in Sections IV-B and IV-C with lightweight threshold clipping of shared parameters. The latter bounds extreme parameter values before aggregation as:

$$u_j^{\text{clip}}(t) = \text{clip}(u_j(t), -\tau, \tau) \quad (12)$$

where τ denotes the clipping threshold and $u_j(t)$ represents the partial update received from participant C_j at round t . Together, these mechanisms constrain coordinate-wise deviations, extreme parameter values, and the overall magnitude of aggregated updates.

It is crucial to note that this defense does not rely on sophisticated aggregation algorithms and is purposefully lightweight. The main goal is to show that even minimal defenses combined with partial parameter sharing and decentralized learning effectively minimize the impact of poisoning. However, in centralized FL, thresholding may not prevent poisoned updates from propagating over the global model, which highlights the fundamental vulnerability of centralized aggregation.

VI. EXPERIMENTAL SETUP

A. Dataset and Preprocessing

The COVID-19 Radiography Dataset [20], [21], which included images of chest X-rays from the COVID and Normal classes, was used for the experiments. To ensure balanced evaluation, 400 images from each class were selected for a total of 800 samples. The dataset was split as follows: 80% of the dataset was used for training, 10% for validation, and 10% for testing.

In order to simulate a realistic data imbalance, training and validation data were spread among four participants (Hospitals A–D) in unequal proportions. All models were evaluated using a common global test set. Images were resized to 224×224 normalized to $[0, 1]$, and augmented using random horizontal flipping and rotation.

B. Model Architecture

Each participant employed an identical convolutional neural network based on MobileNetV2 pretrained on ImageNet. The backbone was frozen during training and followed by a Global Average Pooling layer, a fully connected layer with 128 ReLU units, and a sigmoid output neuron for binary classification.

The model was optimized using the Adam optimizer and binary cross-entropy loss. Performance was evaluated using accuracy, precision, recall, and F1-score.

C. Federated Training Configuration

Four participants formed a fully connected peer-to-peer network and collaboratively trained their models over 10 communication rounds, with each round consisting of 3 local epochs and a batch size of 32. Only parameters from the final two dense layers (kernel and bias) were eligible for sharing.

By choosing a percentage of parameters at random from the shared layers, partial weight sharing was implemented. In our experiments, we evaluated two representative configurations: A low sharing proportion is represented by 25% sharing, whereas full sharing of the applicable layers is represented by 100% sharing. During training, private feature extraction layers remained totally local. Moreover, Dataset-size weighted averaging was used for aggregation.

D. Weight Poisoning Attack Setup

The attacker was identified as a single malicious participant (Hospital A). The attacker used sign flipping and scaling with a factor of 5.0 to change its shared parameters starting from the first communication round. Throughout every round, this poisoned update was continuously injected.

The attack targets the shared dense-layer parameters only, consistent with the partial sharing protocol.

E. Defense Configuration

A lightweight Weight Guard mechanism combining Median-MAD clipping, threshold clipping, and delta-norm stabilization was applied to received updates prior to aggregation. For each shared parameter, median-based clipping was performed using a threshold defined as $k = 3.0$ times the

TABLE I
BASELINE PERFORMANCE: CLEAN SCENARIO (NO ATTACK/DEFENSE)

Data Scale	Share (%)	Hosp-A	Hosp-B	Hosp-C	Hosp-D
Small (400–400)	25	0.850	0.875	0.813	0.838
	100	0.900	0.913	0.825	0.875
Medium (1808–5096)	25	0.928	0.919	0.894	0.938
	100	0.942	0.948	0.858	0.939
Large (3616–10192)	25	0.936	0.935	0.944	0.917
	100	0.937	0.945	0.950	0.949

Note: Values represent Test Accuracy.

MAD, with an absolute floor of 0.05. Additionally, a delta-norm constraint limited per-round parameter changes to 1.0 on shared indices only. Updates were applied exclusively to parameters actively shared by participants, preventing zero-injection effects.

F. Evaluation Protocol

Each round’s training and validation metrics were tracked, and the combined test dataset was used for the final assessment. Thresholded model outputs were used to calculate precision, recall, and F1-score. To guarantee reproducibility, fixed random seeds were used in every experiment.

VII. RESULTS AND DISCUSSION

The robustness of the suggested decentralized federated learning system against weight poisoning attacks is assessed in this section using two sharing configurations (25% and 100% of the final dense layers) and three dataset scales (small, medium, and large). Accuracy, precision, recall, and F1-score on a common global test set are used to report performance. Results are shown for each dataset size under three scenarios: clean training (no attack) Table I, weight poisoning without defense Table II, and weight poisoning with the suggested lightweight defense Table III. In Table III to quantify the improvement introduced by the proposed defense, we define the accuracy gain over the attack-only baseline as:

$$\text{Gain} = \text{Defended}_{\text{Acc}} - \text{Attack}_{\text{Acc}} \quad (13)$$

This metric measures the absolute recovery in test accuracy achieved by the defense compared to the poisoning-only scenario.

A. Small Dataset (COVID = 400, Normal = 400)

Under clean training, accuracy ranges from 0.813–0.875 with 25% sharing and 0.825–0.913 with 100% sharing (Table I). Weight poisoning causes the largest degradation in this setting, particularly at 25% sharing, where attacker accuracy falls to 0.500, and benign average accuracy and F1-score decrease to 0.700 and 0.772, respectively (Table II). With Weight Guard, accuracy recovers to 0.887 with an F1-score of 0.888, yielding the largest observed gain (+0.187). At 100% sharing, defended accuracy and F1-score reach 0.871 and 0.872, with a smaller gain of +0.013 (Table III).

TABLE II
SYSTEM VULNERABILITY UNDER WEIGHT POISONING (NO DEFENSE)

Data Scale	Share (%)	Attack Acc.	Benign Avg. Acc.	Benign Avg. F1
Small (400–400)	25	0.500	0.700	0.772
	100	0.837	0.858	0.851
Medium (1808–5096)	25	0.884	0.841	0.902
	100	0.880	0.896	0.930
Large (3616–10192)	25	0.923	0.893	0.936
	100	0.923	0.921	0.950

B. Medium Dataset (COVID = 1808, Normal = 5096)

Under clean training, accuracy ranges from 0.894–0.938 with 25% sharing and 0.858–0.948 with 100% sharing (Table I). Under poisoning, benign average accuracy reaches 0.841 and 0.896 for 25% and 100% sharing, respectively, while the corresponding F1-scores are 0.902 and 0.930 (Table II). With Weight Guard, accuracy improves to 0.898 and 0.941, while F1-score reaches 0.935 and 0.961, yielding gains of +0.057 and +0.045, respectively (Table III).

C. Large Dataset (COVID = 3616, Normal = 10192)

Clean accuracy remains high, ranging from 0.917–0.944 with 25% sharing and 0.937–0.950 with 100% sharing (Table I). Under poisoning, benign average accuracy reaches 0.893 and 0.921 for 25% and 100% sharing, respectively, with corresponding F1-scores of 0.936 and 0.950 (Table II). With Weight Guard, both configurations achieve nearly identical performance: accuracy reaches 0.936 and 0.935 for 25% and 100% sharing, respectively, while both achieve an F1-score of 0.958 (Table III).

D. Impact of Sharing Ratio and Defense

Full sharing (100%) of the specific dense-layer parameters generally provides higher performance under weight poisoning without defense, particularly for the medium and large datasets. However, this advantage is not consistent after applying Weight Guard. Partial sharing (25%) achieves higher defended accuracy for the small dataset, while 100% sharing performs better for the medium dataset, and both configurations achieve nearly identical performance for the large dataset. Partial sharing also limits parameter exposure and reduces communication overhead.

TABLE III
DEFENSE EFFECTIVENESS WITH WEIGHT GUARD ENABLED.

Data Scale	Share (%)	Defended Acc	Defended F1	Gain
Small (400–400)	25	0.887	0.888	+0.187
	100	0.871	0.872	+0.013
Medium (1808–5096)	25	0.898	0.935	+0.057
	100	0.941	0.961	+0.045
Large (3616–10192)	25	0.936	0.958	+0.043
	100	0.935	0.958	+0.014

Note: Defended metrics are the average of all participants.

Weight poisoning is most disruptive in the small-dataset setting, where Weight Guard provides the largest recovery, particularly with 25% sharing. As dataset size increases, the system becomes less affected by the attack, suggesting greater inherent robustness in medium- and large-scale settings, while Weight Guard provides additional stability.

E. Discussion

The findings demonstrate that decentralized federated learning with partial dense-layer sharing maintains robust performance against the evaluated weight poisoning attack, particularly as dataset size increases. The peer-to-peer design also avoids reliance on a central aggregation server.

Sharing only 25% of the selected dense-layer parameters reduces parameter exposure and communication overhead while maintaining competitive performance compared with 100% sharing. With Weight Guard, partial sharing achieves higher defended accuracy in the small-data setting and nearly identical performance to full sharing in the large-data setting, showing that full parameter sharing is not always necessary for robust collaborative learning. Weight Guard further improves stability, with the largest recovery observed when data availability is limited.

VIII. CONCLUSION

This paper evaluated decentralized federated learning with partial dense-layer sharing under a persistent weight poisoning attack. Experiments across three dataset scales and two sharing ratios showed stable performance, particularly at medium and large scales. Weight Guard, combining Median-MAD, threshold, and delta-norm clipping, further improves robustness, while partial sharing reduces parameter exposure compared with full sharing. Although full sharing provides performance advantages in some settings, partial sharing offers an effective balance between reduced parameter exposure and collaborative performance. Weight Guard is particularly beneficial in limited-data scenarios.

The current evaluation is limited to four participants, a fully connected topology, a single attacker, and chest X-ray classification. Future work will consider larger networks, additional datasets, non-IID settings, and more diverse attack models.

REFERENCES

- [1] M. H. U. Rehman, W. H. L. Pinaya, P. Nachev, J. T. Teo, S. Ourselin, and M. J. Cardoso, "Federated learning for medical imaging radiology," *The British Journal of Radiology*, vol. 96, no. 1150, p. 20220890, 2023.
- [2] N. Jebreel, J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "Defending against the Label-flipping Attack in Federated Learning," *arXiv preprint arXiv:2207.01982*, 2022.
- [3] G. Xia, J. Chen, C. Yu, and J. Ma, "Poisoning attacks in federated learning: A survey," *IEEE Access*, vol. 11, pp. 10708–10722, 2023.
- [4] K. N. Kumar, C. K. Mohan, L. R. Cenkeramaddi, and N. Awasthi, "Minimal data poisoning attack in federated learning for medical image classification: An attacker perspective," *Artificial Intelligence in Medicine*, vol. 159, p. 103024, 2025.
- [5] W. Ma, Q. Zhao, and W. Tian, "A defense method against multi-label poisoning attacks in federated learning," *Scientific Reports*, vol. 15, no. 1, p. 26197, 2025.
- [6] P. R. Ovi and A. Gangopadhyay, "Robust federated learning against data poisoning attacks: Prevention and detection of attacked nodes," *Electronics*, vol. 14, no. 15, p. 2970, 2025.
- [7] H. Cheng, Y. Qu, W. Liu, L. Gao, and T. Zhu, "Decentralized federated learning for private smart healthcare: A survey," *Mathematics*, vol. 13, no. 8, p. 1296, 2025.
- [8] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., et al.: Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems* 1, 374–388 (2019)
- [9] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. *arXiv preprint arXiv:1806.00582* (2018)
- [10] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 634–643, 2019.
- [11] R. Nabha, A. Laouti, and A. E. Samhat, "Internet of Things-Based Healthcare Systems: An Overview of Privacy-Preserving Mechanisms," *Applied Sciences*, vol. 15, no. 7, p. 3629, 2025.
- [12] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proceedings of the 29th USENIX Security Symposium (USENIX Security 20)*, pp. 1605–1622, 2020.
- [13] X. Zhang, X. Chen, M. Hong, Z. S. Wu, and J. Yi, "Understanding clipping for federated learning: Convergence and client-level differential privacy," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.
- [14] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Machine Learning (ICML)*, pp. 5650–5659, 2018.
- [15] Y. Wang, D.-H. Zhai, and Y. Xia, "RFVIR: A robust federated algorithm defending against Byzantine attacks," *Information Fusion*, vol. 105, p. 102251, 2024.
- [16] X. Chen, S. Z. Wu, and M. Hong, "Understanding gradient clipping in private SGD: A geometric perspective," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 13773–13782, 2020.
- [17] Y. Xie, M. Fang, and N. Z. Gong, "Model poisoning attacks to federated learning via multi-round consistency," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 15454–15463, 2025.
- [18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017.
- [19] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.
- [20] Chowdhury, M.E.H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M.A., Mahbub, Z.B., Islam, K.R., Khan, M.S., Iqbal, A., Al-Emadi, N., Reaz, M.B.I., Islam, M.T.: Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access* 8, 132665–132676 (2020)
- [21] Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S.B.A., Islam, M.T., Maadeed, S.A., Zughaier, S.M., Khan, M.S., Chowdhury, M.E.: Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. Preprint (2020)
- [22] Roy, A.G., Siddiqui, S., Pölsterl, S., Navab, N., Wachinger, C.: Brain-torrent: A peer-to-peer environment for decentralized federated learning. *arXiv preprint arXiv:1905.06731* (2019)
- [23] M. Qiu, Z. Liu, R. Du, and X. Liang, "BRAFL: Byzantine-Robust Aggregation Scheme for Federated Learning under Data Heterogeneity," in *Proc. 2025 Int. Joint Conf. Neural Networks (IJCNN)*, 2025, pp. 1–8.
- [24] X. Wen, Y. Wang, M. Yuan, A. Wang, G. Zheng, H. Yu, and L. Meng, "DC-PFL: A dynamic clustering-based personalized federated learning method for human activity recognition," *Eng. Sci. Technol., Int. J.*, vol. 72, p. 102230, 2025.
- [25] P. Kairouz and H. B. McMahan, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [26] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.
- [27] S. Lee, T. Zhang, and A. S. Avestimehr, "Layer-wise adaptive model aggregation for scalable federated learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 7, pp. 8491–8499, 2023.