Loss Based Byzantine Resilience for Decentralized Learning

Shinnosuke Masuda
Kyoto University
Kyoto, Japan

Kazuyuki Shudo
Kyoto University
Kyoto, Japan

***Abstract*—Decentralized learning has gained attention for overcoming the disadvantage of centralized learning such as single point of failure and scalability. However, when there are faulty nodes or attackers (Byzantine nodes) in the network, and these nodes perform Byzantine attacks by sending abnormal models, the model may fail to converge. Existing defense methods against Byzantine attacks have been pointed out to fail when facing sophisticated attacks or an increasing proportion of Byzantine nodes. To address these issues, we propose a novel defense method based on loss function values. Each node calculates the loss values of both its own model and the models of neighboring nodes using its local data. If the difference between the two loss values exceeds a certain threshold, the model is used for aggregation. This defense method ensures that only models with generalization performance above a certain level are included in the aggregation process, effectively filtering out Byzantine models as well as low-performing models from benign nodes. Experimental results using the CIFAR-10 dataset demonstrate that the proposed method achieves better convergence speed and accuracy compared to existing methods, and it remains robust even as the proportion of Byzantine nodes increases.**

***Index Terms*—Decentralized Learning, Federated Learning, Deep Learning**

I. Introduction

Decentralized learning [1], [2] has gained attention as a peer-to-peer (P2P) network-based distributed deep learning approach that that overcomes the disadvantage of centralized learning [3] such as single point of failure and scalability. In decentralized learning, each node exchanges the models learned on its local data with neighboring nodes and aggregates the models, allowing for the construction of an accurate model while preserving data privacy. According to prior research, decentralized learning has been shown to perform faster than federated learning in environments with limited bandwidth or high network latency [4].

However, in decentralized environments, there may be faulty nodes or Byzantine nodes executing malicious attacks. Byzantine nodes hinder model convergence by transmitting abnormal models to neighboring nodes. Attacks executed by Byzantine nodes are broadly categorized into Data Poisoning Attacks (DPA) and Model Poisoning Attacks (MPA) [5]. DPA involves attackers training a model using intentionally manipulated datasets and transmitting a low-accuracy model to neighboring nodes. A representative attack is the label-flipping attack [6], where the correct labels in the dataset are swapped prior to training. In contrast, MPA involves attackers directly adding noise to the model and sending the tampered model to neighboring nodes. MPA is generally considered more effective than DPA in obstructing model convergence [5]. A prominent example of MPA is the Max attack [7], where the attacker inverts the maximum value in each dimension of the model parameters from all benign nodes and transmits it to neighboring nodes. More sophisticated attacks include "A little is enough" [8] and "Fall of Empires" [9].

To counter such attacks, methods have been proposed that filter out malicious actors or employ robust aggregation techniques [8], [9]. However, existing methods have been criticized for their inability to prevent sophisticated attacks or for becoming ineffective as the proportion of Byzantine nodes increases [5]. To address these issues, we propose a novel filtering method based on loss function values. In our approach, each node calculates the loss values of its own model and the models of neighboring nodes using its local data, and if the difference exceeds a certain threshold, the model is adopted. In the early stages of training, when it is difficult to distinguish between Byzantine and benign nodes, setting a higher threshold allows for the adoption of only models with high generalization performance. Through simulation experiments, we confirmed that the proposed method continues to function effectively even under sophisticated attacks and increasing proportions of Byzantine nodes, outperforming existing methods.

II. Related Work

Filtering methods are divided into two categories: distance-based and performance-based approaches. Distance-based methods typically measure deviations in model parameters, such as the Euclidean distance between the model parameters of a given node and those of neighboring nodes, excluding outliers. Blanchard et al. proposed Krum [10], a method that selects a model from neighboring nodes whose parameter distances to other nodes are minimal, and uses it for aggregation. However, this method becomes ineffective under sophisticated attacks or when the proportion of Byzantine nodes increases.

This work was supported by JSPS KAKENHI Grant Number JP24H00691.

Proc. IEEE 43rd Int'l Conf. on Consumer Electronics
(IEEE ICCE 2025), January 2025

Performance-based methods, on the other hand, evaluate the performance of neighboring models by measuring metrics such as loss values or accuracy using the node's local data, excluding models with poor generalization performance. Xie et al. proposed a method that scores models based on the loss function values of neighboring nodes, and aggregates the models with the lowest scores [11]. The problem with this approach is that in the early stages of training, most models exhibit uniformly high loss values or low accuracy, making it difficult to distinguish between Byzantine and benign nodes, which can lead to the inclusion of Byzantine models in the aggregation.

There are also hybrid methods that combine distance-based and performance-based approaches. Guo et al. proposed UBAR [12], which employs two-stage filtering. In the first stage, multiple nodes with model parameters close to the node' s own parameters in terms of Euclidean distance are selected. In the second stage, the loss values of the neighboring models are evaluated using local data, and if a neighboring model has a lower loss value than the node's own model, that model is adopted. If no model has a lower loss value, the model with the smallest loss among the neighbors is selected. However, this method has two major issues. First, the distance-based filtering in the first stage cannot filter out sophisticated attacks, allowing many adversarial models to remain. Second, from the perspective of the neighboring nodes, the local data of the node receiving the models is unknown test data. Since the receiving node has been trained on that dataset, its model generally achieves high generalization performance on that data, making it rare for the loss value of neighboring models to be lower than the receiving node's model. As a result, adversarial models that pass through the first-stage filtering may be selected for aggregation, increasing the risk of attack.

In summary, distance-based methods fail under sophisticated attacks or when the proportion of Byzantine nodes increases, while performance-based methods face difficulties distinguishing between benign and Byzantine models during the early stages of training. Furthermore, as previous studies have shown, these methods do not utilize past gradient information, leading to issues with convergence [13]. Considering these challenges, a method that does not rely on distance-based filtering, overcomes the difficulty of distinguishing Byzantine nodes in the early stages of performance-based learning, and guarantees convergence through the use of historical information is needed.

III. Problem Formulation

A. Network

The network is represented as an undirected graph consisting of n nodes, denoted as $G = (V, E)$. Here, $V = \{1, 2, \ldots, n\}$ represents the set of node vertices. E is the set of edges, and if $(i, j) \in E$, then i and j are connected. Let N_i be the set of neighboring nodes of node i. Also, let $\bar{N}_i = N_i \cup \{i\}$. Assume that some nodes do not follow the protocol and send arbitrary messages, referred to as Byzantine nodes. Let H be the set of benign nodes and B be the set of Byzantine nodes.

B. Optimization Problem

The goal of the benign nodes in the network is to solve the following optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{H} \sum_{i \in H} \{f_i(x) := \mathbb{E}_{\xi_i \sim D_i} F_i(x; \xi_i)\}, \quad (1)$$

Let $x \in \mathbb{R}^d$ represent the model parameters, where f_i is the local objective function of benign node i, and $F_i(x; \xi_i)$ denotes the loss function for the model parameter x, with ξ_i being a random data sample drawn from the distribution D_i. To solve this optimization problem, each node repeats the following three steps:

1) Local update
   The gradient $g_i^t = \nabla F_i(x_i^t; \xi^t)$ is computed using a mini-batch. Then, the model is updated as follows:
   $$x_i^{t+1/2} = x_i^t - \eta g_i^t \quad (2)$$
2) Model exchange
3) Aggregation
   $$x_i^{t+1} = \sum_{j \in \bar{N}_i} W_{ij} x_j^{t+1/2} \quad (3)$$
   Here, W_{ij} is the mixing weight matrix, and if $i \neq j$ and $(i, j) \notin E$, then $W_{ij} = 0$. Otherwise, $W_{ij} > 0$. Additionally, t represents the iteration number.

When Byzantine nodes are present, abnormal models can be introduced into the aggregation process, deteriorating the accuracy of the model. Therefore, robust aggregation methods and filtering of attackers are necessary.

IV. Proposed Method

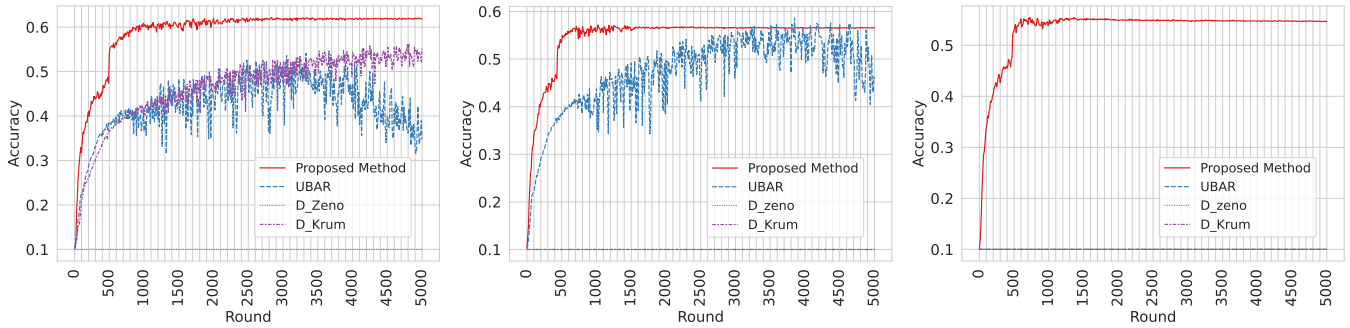
In order to overcome the challenge of distinguishing between benign and Byzantine models during the early stages of performance-based learning, the proposed method ensures that each node i integrates only models that satisfy the following condition, based on its local data.

$$F_i(x; \xi_i) - F_j(x; \xi_i) \geqq \theta \quad (4)$$

θ is a threshold that is set high in the early stages of learning, allowing the selection of models with better generalization performance than the node's own model. However, from the perspective of neighboring node j, the local data of node i serves as test data, making it increasingly difficult for j's model to achieve a lower loss than that of node i, which has been training on this data. As a result, fewer models meet the condition, leading node i to continue learning in isolation, which can cause overfitting. When overfitting occurs, in addition to lowering the threshold, if no models exceed the threshold, node i will adopt models that satisfy the following condition:

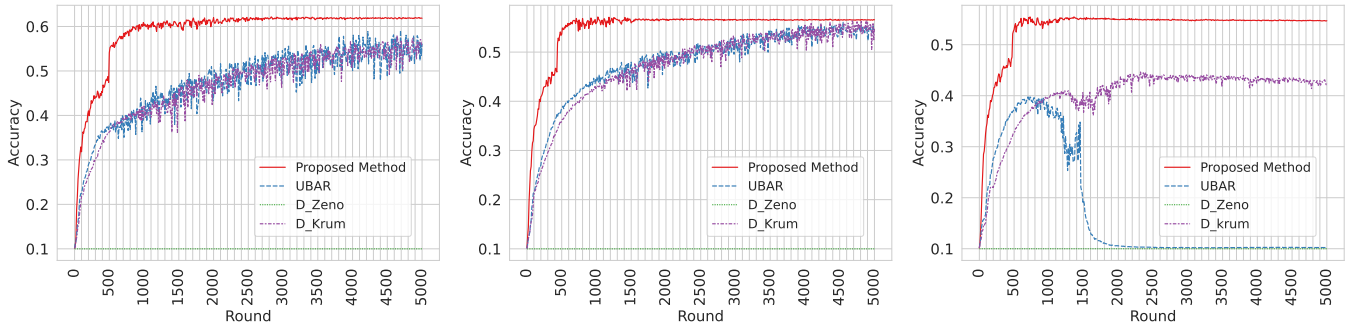
$$j \in \underset{j \in N_i}{\operatorname{argmin}} F_j(x; \xi_i) \quad (5)$$

In this study, each node calculates the test error at the end of each epoch and determines that overfitting has occurred if the minimum test error is not updated for 5 consecutive



(a) $Br = 0.1$ (b) $Br = 0.3$ (c) $Br = 0.5$

Fig. 1. Accuracies with "Max Attack". Note that Br is Byzantine rate



(a) $Br = 0.1$ (b) $Br = 0.3$ (c) $Br = 0.5$

Fig. 2. Accuracies with "A little is enough". Note that Br is Byzantine rate.

epochs. The learning process starts with $\theta = 0.1$, and each time overfitting is detected, the value of θ is reduced by 0.1. However, to prevent the infiltration of Byzantine models, the lower bound of θ is set to -0.1. Additionally, momentum was employed to stabilize model convergence, with the momentum coefficient set to 0.9. Let $N_f^{i,t}$ denote the set of neighboring nodes of node i after filtering at iteration t, and the model is aggregated as follows:

$$F_i^t = \frac{1}{|N_f^{i,t}|} \sum_{j \in N_f^{i,t}} x_j \tag{6}$$

$$x_i^{t+1} = \alpha x_i^{t+1/2} + (1-\alpha) F_t^i \tag{7}$$

In this study, α was set to 0.5.

V. Experiments

In this section, we verify through simulation experiments whether the proposed method is more resistant to both simple and sophisticated attacks compared to existing methods. In this study, we assume an environment where each node communicates and learns synchronously.

A. Simulation Settings

1) Dataset & Model: The CIFAR-10 dataset [14] was used in the experiment. The dataset was distributed so that the data distribution for each client was independent and identically distributed (IID). The model used was a CNN with two convolutional layers and two fully connected layers. The learning rate was set to 0.01 for all nodes, and the batch size was fixed at 256.

2) Network Topology: The number of nodes in the network was set to 10 for the experiments. The topology of the decentralized network was constructed based on the connection rate between nodes, which is commonly used in many previous studies. The connection rate, denoted as C_r, represents the probability that an edge exists between any two nodes. Since many prior studies fix C_r at 0.4 for experiments, the same value was used in this study. The proportion of Byzantine nodes in the network is represented by B_r, and experiments were conducted with three different values for B_r. Additionally, it is assumed that the network is static in this study.

3) Baselines: As baselines for the proposed method, D_Krum [10], UBAR [12], and D_Zeno [11] were used in the simulation experiments.

4) Byzantine Attacks: To verify whether the proposed method has robustness against attacks compared to existing methods, we implemented the "Max Attack," representing a simple attack, and "A little is enough," representing an sophisticated attack.

(a) Max Attack: The attacker adds noise to the model

as follows and sends it to the benign nodes. Here, d represents the dimension of the model parameters.

$$x_i^{t+1/2}[d] = -max\{x_j[d]|j \in H\} \tag{8}$$

(b) A little is enough: The attacker sends a model to the benign nodes, which is perturbed from the average model parameters of the benign nodes.

$$x_i^{t+1/2} = x_{avg}^{t+1/2} - z\sigma \tag{9}$$

Here, $x_{avg}^{t+1/2}$ is the average model parameter of the benign nodes, σ is the standard deviation, and z is determined as follows:

$$z = \max_z \left\{ \varphi(z) < \frac{n - B - s}{n - B} \right\} \tag{10}$$

$\varphi(z)$ is the cumulative standard normal function, and s is given by $s = \left\lceil \frac{n}{2} + 1 \right\rceil$.

B. Simulation Results

1) Max Attack: Figure 1 illustrates the accuracy improvement over rounds for each defense method as the Byzantine ratio increases during the Max Attack. The proposed method converges faster compared to existing methods and remains effective even as the Byzantine ratio increases. Additionally, a rapid improvement in accuracy is observed after around 500 rounds. This is because overfitting begins around 500 rounds, leading to a reduction in the threshold, and at least one model is incorporated into the aggregation process.

2) A little is enough: Figure 2 shows the accuracy improvement over rounds for each defense method as the Byzantine ratio increases during the "A little is enough" attack. The proposed method shows faster model convergence compared to other methods and continues to function even as the Byzantine ratio increases. Furthermore, when the Byzantine ratio is 0.5, similar to the Max Attack, while the accuracy of other methods deteriorates, the proposed method maintains an accuracy of over 0.5.

VI. Conclusion

In this study, we proposed a novel filtering method that overcomes the challenges of performance-based approaches. As a result of the simulation, the proposed method not only converges faster than other existing methods, but also maintains accuracy even when the Byzantine ratio increases. Specifically, when the Byzantine ratio is 0.5, the accuracy of models in existing methods is significantly degraded, whereas the proposed method maintains an accuracy of over 0.5. Future work will focus on the dynamic adjustment of the threshold θ and the convergence analysis of the proposed method.

References

[1] M. Blot, D. Picard, M. Cord, and N. Thome, "Gossip training for deep learning," ArXiv, vol.abs/1611.09726, p.5, 2016. https://api.semanticscholar.org/CorpusID:11058126

[2] H. Oguni and K. Shudo, "Communication scheduling for gossip sgd in a wide area network," IEEE Access, vol.9, pp.77873–77881, 2021.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Trans. Intell. Syst. Technol., vol.10, no.2, p.19, 2019.

[4] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," Proceedings of the 31st International Conference on Neural Information Processing Systems, pp.5336–5346, 2017.

[5] G. Xia, J. Chen, C. Yu, and J. Ma, "Poisoning attacks in federated learning: A survey," IEEE Access, vol.11, pp.10708–10722, 2023.

[6] V. Shejwalkar, A. Houmansadr, P. Kairouz, and D. Ramage, "Back to the drawing board: A critical evaluation of poisoning attacks on federated learning," CoRR, vol.abs/2108.10241, pp.1354–1371, 2021.

[7] J. Hou, F. Wang, C. Wei, H. Huang, Y. Hu, and N. Gui, "Credibility assessment based byzantine-resilient decentralized learning," IEEE Transactions on Dependable and Secure Computing, pp.1–12, 2022.

[8] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," Advances in Neural Information Processing Systems, pp.1–11, 2019.

[9] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation," Proceedings of The 35th Uncertainty in Artificial Intelligence Conference, pp.261–270, 2020.

[10] P. Blanchard, E.M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," Advances in Neural Information Processing Systems, p.11, 2017.

[11] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," Proceedings of the 36th International Conference on Machine Learning, pp.6893–6901, 2019.

[12] S. Guo, T. Zhang, H. Yu, X. Xie, L. Ma, T. Xiang, and Y. Liu, "Byzantine-resilient decentralized stochastic gradient descent," IEEE Transactions on Circuits and Systems for Video Technology, vol.32, no.6, pp.4096–4106, 2022.

[13] S.P. Karimireddy, L. He, and M. Jaggi, "Learning from History for Byzantine Robust Optimization," ICML 2021 - Proceedings of International Conference on Machine Learning, p.26, 2021. https://arxiv.org/abs/2012.10333

[14] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009. https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf