LeadFL: Client Self-Defense against Model Poisoning in Federated Learning

Chaoyi Zhu [1] **Stefanie Roos** [1] **Lydia Y. Chen** [1]

Abstract

Federated Learning is highly susceptible to backdoor and targeted attacks as participants can manipulate their data and models locally without any oversight on whether they follow the correct process. There are a number of server-side defenses that mitigate the attacks by modifying or rejecting local updates submitted by clients. However, we find that bursty adversarial patterns with a high variance in the number of malicious clients can circumvent the existing defenses. We propose a client-self defense, `LeadFL`, that is combined with existing server-side defenses to thwart backdoor and targeted attacks. The core idea of `LeadFL` is a novel regularization term in local model training such that the Hessian matrix of local gradients is nullified. We provide the convergence analysis of `LeadFL` and its robustness guarantee in terms of certified radius. Our empirical evaluation shows that `LeadFL` is able to mitigate bursty adversarial patterns for both iid and non-iid data distributions. It frequently reduces the backdoor accuracy from more than 75% for state-of-the-art defenses to less than 10% while its impact on the main task accuracy is always less than for other client-side defenses.

1. Introduction

Federated Learning (FL) realizes collaborative learning without the need to share possibly sensitive raw data. Clients submit intermediate local models to a server, the federator, who aggregates these models. In order to achieve models of high accuracy, high-quality local models and effective aggregation algorithms are needed. Adversarial clients can reduce the accuracy, either overall or on specific tasks, by manipulating their local data and the submitted model. For instance, malicious clients can launch backdoor attacks, which mislead the model to make inaccurate inferences on images with certain triggers.

The attack severity is closely related to the number of malicious clients that are chosen over time. Federated learning proceeds in rounds. Usually, in each round, a certain number of clients are selected from a large pool. If the selection is random and by the server, the number of malicious clients chosen varies greatly even if the overall fraction of malicious clients in the pool stays constant. Figure 1 displays an example for 5 selected clients over 5 rounds. As a consequence, in some rounds, the fraction, malicious clients make up the majority of the clients, which allows them to launch a strong attack.

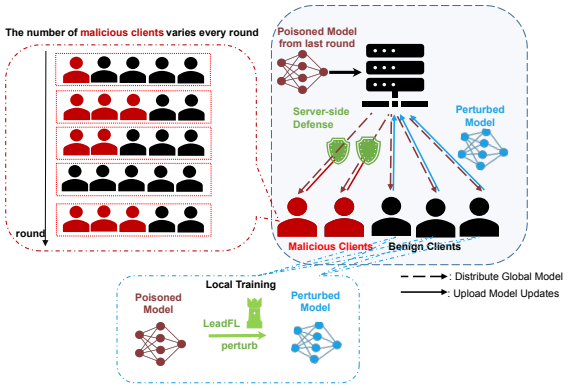


Figure 1: Bursty adversarial patterns with the number of malicious clients chosen varying greatly between rounds

Defense mechanisms (Blanchard et al., 2017; Fung et al., 2020; Muñoz-González et al., 2019; Xia et al., 2019; Mhamdi et al., 2018; Ozdayi et al., 2021; Panda et al., 2022; Yin et al., 2018; Nguyen et al., 2022; Rieger et al., 2022; Gupta et al., 2022; Xu et al., 2022) have been designed to mitigate the attacks. The majority of these attacks are server-side, meaning the federator assigns updates that appear to be malicious a low weight during aggregation or completely excludes them from the aggregation. These defenses have been shown to be effective against sophisticated attacks when the number of malicious selected clients is constant and low (Nguyen et al., 2022; Panda et al., 2022; Rieger et al., 2022). In addition to demonstrating the empirical effectiveness, theoretical frameworks, such as certified radius on models (Panda et al., 2022) and inference samples (Xie

[1]Delft University of Technology, Delft, Netherlands. Correspondence to: Lydia Y. Chen <lydiaychen@ieee.org>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

et al., 2021) provide theoretical guarantees of the defense effectiveness.

In contrast, client-side defenses (Sun et al., 2021) have the client modify the training process. The most notable client-side defense is FL-WBC (Sun et al., 2021), which can deal with bursty attack patterns. The authors find that a strong bursty attack in one round has a lingering effect on the model and the duration and severity of that effect depends on the sparsity of the Hessian matrix of gradients: the higher the sparsity, the longer the attack effect. FL-WBC perturbs the Hessian matrix of gradients by adding random noise into clients' local models to reduce sparsity. Such uncalibrated random noise unfortunately leads to the degradation in the global model accuracy. Moreover, there is no theoretical guarantee that FL-WBC is robust against backdoor attacks.

In this paper, we design `LeadFL`, a client-side defense that enhances server-side defenses to deal with bursty adversarial patterns while not affecting global model accuracy significantly. The core of `LeadFL` is an optimization framework that optimally perturbs the Hessian matrix of local gradients and local models using a regularisation term such that their Hessian matrix is close to the identity matrix.We verify the effectiveness of the proposed regularized Hessian optimization by deriving the convergence analysis and certified radius analysis, which quantifies the distances between benign and poisoned models. Specifically, we make the following technical contributions:

- We design `LeadFL`, a novel client-side defense based on Hessian matrix optimization, to mitigate the impact of bursty adversarial patterns for backdoor and targeted attacks.
- We derive the convergence analysis and certified radius analysis, proving `LeadFL` that is effective.
- We empirically combine `LeadFL` with different server-side defenses and find that the combination can effectively defend against strong attacks while other combinations of server-side and client-side defenses fail. We reduce the backdoor accuracy by up to 65% and achieve a lower impact on main task accuracy than other combined defenses.

2. Background and Prior Art

We first introduce necessary concepts and then analyze the state-of-the-art defenses against model poisoning.

2.1. Model Poisoning Attacks in Federated Learning

Federated Learning In Federated Learning (FL) (Konečný et al., 2016), K clients indexed by $k = \{1 \cdots K\}$ are selected from a total of N clients at global round t to train a learning model by using the local data to minimize the loss function $\mathcal{L}(\boldsymbol{\theta}^{\boldsymbol{k}})$ with model weights of $\boldsymbol{\theta}^{\boldsymbol{k}}$). Specifically, at its local round of i, the client k uses stochastic gradient descent to update weights as follows:

$$\boldsymbol{\theta}^k_{t,i+1} \leftarrow \boldsymbol{\theta}^k_{t,i} - \eta_{t,i} \nabla \mathcal{L}\left(\boldsymbol{\theta}^k_{t,i}\right)$$

where $\eta_{t,i}$ is the learning rate and each local round is computed on a mini-batch of data samples uniformly chosen from client k's local data set.

Periodically, namely, at every global round t, the federator selects a subset of clients and updates the global model weights. The most common aggregation method is FedAvg (McMahan et al., 2017), which averages the selected local models with weights proportional to their sample sizes.

Poisoning Attacks Malicious clients may join the training process. We assume them to have the similar computational capability as benign clients and they cannot access the weights or data of other clients. Their objectives are to reduce the model accuracy on certain tasks, termed targeted attacks (Chen et al., 2017; Bhagoji et al., 2019), or to mislead the global model to make wrong inferences on data sets with certain triggers, termed backdoor attacks (Xie et al., 2019; Bagdasaryan & Shmatikov, 2021), without degrading the overall model accuracy. To obtain such a poisoned model, malicious clients train their local models on malicious data to minimize the malicious loss functions $\mathcal{L}_M$ as follows:

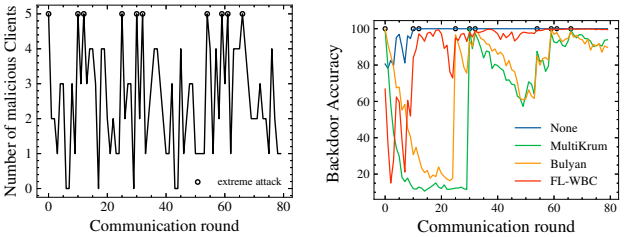
$$\boldsymbol{\theta}^k_{t,i+1} \leftarrow \boldsymbol{\theta}^k_{t,i} - \eta_{t,i} \left[\pi \nabla \mathcal{L}\left(\boldsymbol{\theta}^k_{t,i}\right) + (1-\pi) \nabla \mathcal{L}_M\left(\boldsymbol{\theta}^k_{t,i}\right)\right]$$

Note that data samples in the malicious data set are assumed have the same distribution as the benign training data. The only difference is that for targeted attacks, the labels are altered to belong to a certain target class whereas for backdoor attacks, data samples with certain patterns are inserted into the dataset.

Model poisoning attacks are typically stealthy and difficult to detect, as the malicious dataset is usually small and does not affect the accuracy of the global model(Fung et al., 2020; Steinhardt et al., 2017; Tolpegin et al., 2020; Bagdasaryan & Shmatikov, 2021).

2.2. Prior Art on Defenses

Server-side defenses To defend against the adversarial parties, the federator may employ (i) robust aggregation by computing the median (Yin et al., 2018) of all or subset of client updates, e.g., Trimmed-mean (Yin et al., 2018), or (ii) filtering by removing outliers in the set of updates based on pair-wise distance, e.g., MultiKrum (Blanchard et al., 2017). Bulyan (Mhamdi et al., 2018) combines both approaches by first filtering the outliners using MultiKrum and then applying robust aggregation using Trimmed Mean. These defenses are designed for general adversarial attacks where the number of malicious clients is strictly less than the benign clients.



(a) Number of malicious clients selected out of 5 selected clients per round.

(b) Backdoor accuracy of 3 server-side defenses and a client-side defense (FL-WBC).

Figure 2: The lingering impact of bursty backdoor attack on federated learning for FashionMNIST.

Defenses for poisonous model attacks. Recognizing the increasing threat from poisoning attacks, the prior art designs attack-specific defenses by bounding the norms of updates or adding noise. SparseFed (Panda et al., 2022) mitigates model poisoning attacks in FL by only updating the most relevant weights of the aggregated models. DeepSight (Rieger et al., 2022) mitigates backdoor attacks in FL through clustering the last layer of deep models to filter outliers. CRFL (Xie et al., 2021) exploits clipping and smoothing methods to provide certified robustness against backdoor attacks.

Aforementioned defenses mainly take place at the federator, under an implicit assumption that the number of malicious clients selected in each global round is lower than the number of benign clients. To the best of our knowledge, FL-WBC (Sun et al., 2021) and Local Differential Privacy (LDP) (Naseri et al., 2022) are the only client-side defenses against model poisoning attacks in federated learning. In LDP, benign clients add noise to updates before sending updates to the server.

3. Hessian Matrix

We first demonstrate the long-term impact of bursty adversarial patterns on state-of-the-art defenses that we already discussed in Section 2.2. Details about the framework used can be found in Section 5. Using a total of 100 clients with 25 of them being malicious, we selected 5 clients per round. Fig. 2a shows the number of malicious clients selected. The learning task is image classification on FashionMNIST and the malicious clients execute a 9-pixel attack (Bagdasaryan & Shmatikov, 2021). As displayed in Figure 2b, none of the defenses can defend against the attack, i.e., the final backdoor accuracy is around 90%, though Bulyan and Multikrum are able to filter out the malicious updates occasionally. This example highlights the ineffectiveness of existing defenses against bursty adversarial patterns. While the attack only directly affects some rounds, the effect lingers.

Attack Effect and Hessian Matrix The effect of attacks taking place at round t, δ_t, can be formalized (Sun et al., 2021) as follows $\delta_t \triangleq \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^M$ where $\boldsymbol{\theta}_t$ represents the global model weights at round t without the presence of malicious updates and $\boldsymbol{\theta}_t^M$ is the model weights from the malicious clients.

Based on (Sun et al., 2021), the estimated attack effect, $\hat{\delta}_t$, can be written as the function of Hessian matrix

$$\hat{\delta}_t = \frac{N}{K}\left[\sum_{k\in\mathbb{S}_t} p^k \prod_{i=0}^{I-1}\left(\boldsymbol{I} - \eta_t \boldsymbol{H}_{t,i}^k\right)\right]\hat{\delta}_{t-1}, \qquad (1)$$

where p^k is aggregation weight for client k, $\boldsymbol{H}_{t,i}^k \triangleq \nabla^2\mathcal{L}\left(\boldsymbol{\theta}_{t,i}^k\right)$ is the Hessian matrix at local iteration i of global round t and $\boldsymbol{I}$ is the identify matrix.

The Hessian matrix $\boldsymbol{H}_{t,i}^k$ is observed to be highly sparse during the training process, for both benign and malicious clients. Therefore, the weights of $\hat{\delta}_{t-1}$ in Eq. 1 are close to $\sum_{k\in\mathbb{S}_t} p^k \prod_{i=0}^{I-1}(\boldsymbol{I})$. As a consequence, δ_t causes notable changes and due to the relation between $\hat{\delta}_{t-1}$ and $\hat{\delta}_t$, the effect lingers.

Insight To mitigate the effect of poisoned weights, benign clients can perturb the Hessian matrix such that that coefficient of δ_{t-1} is minimized, i.e., $\prod_{i=0}^{I-1}(\boldsymbol{I})$. As noted earlier, the Hessian matrix here is sparse, FL-WBC (Sun et al., 2021) proposes to add random noise to the benign clients' model weights such that their Hessian matrix is no longer sparse and the impact of δ_{t-1} is thus reduced. However, as the noise is randomly added, the coefficient may not necessarily be reduced, unfortunately enhance the attacking effects, and further degrade the model accuracy, shown by extensive experiments in Appendix C. We are thus motivated to find alternatives to perturb the Hessian Matrix more effectively to reduce coefficients without degrading the model's accuracy.

4. LeadFL

In this section, we describe `LeadFL`, a novel client-side defense and can be agilely combined with any existing server-side defense.

4.1. Algorithm Design

The core idea of `LeadFL` is to mitigate the attack effect by minimizing the coefficient term $\left(\boldsymbol{I} - \eta_t \boldsymbol{H}_{t,i}^k\right)$ in Equation 1. Essentially, we aim to add perturbation to the Hessian matrix such that this coefficient term vanishes. We show that this is equivalent to adding the same amount of perturbation in model updates $\boldsymbol{\theta}_{t,i+1}^k$, which motivates our proposed novel regularization term. We first summarize the proposed regularized model update protocol before conducting an analysis:

$$\widetilde{\boldsymbol{\theta}}_{t,i+1}^k \leftarrow \boldsymbol{\theta}_{t,i}^k - \eta_t \nabla \mathcal{L}\left(\boldsymbol{\theta}_{t,i}^k\right) \tag{2}$$

$$\boldsymbol{\theta}_{t,i+1}^k \leftarrow \widetilde{\boldsymbol{\theta}}_{t,i+1}^k - \eta_t \alpha \operatorname{clip}\left(\nabla\left(\boldsymbol{I} - \eta_t \widetilde{\boldsymbol{H}}_{t,i}^k\right), q\right) \tag{3}$$

where $\widetilde{\boldsymbol{\theta}}_{t,i+1}^k$ is the intermediate model weights in local iteration $t+1$ of client k. $\widetilde{\boldsymbol{H}}_{t,i}$ is the estimation of the Hessian matrix of the local model before adding the regularization term in this local iteration, α is a hyper-parameter to control the magnitude of the regularization term, and clip is the operation of bounding the regularization term to a threshold q to ensure convergence.

Hessian Matrix Estimation As the Hessian matrix is the second-order derivative of the loss function, we resort to the estimation of the Hessian matrix proposed in (LeCun et al., 1989). We only focus on the diagonal terms due to the intractability of estimating non-diagonal terms. Specifically, the diagonal elements in $\boldsymbol{H}_{t,i}^k$ can be estimated from the change of the gradient between local iteration i and $i+1$: $\nabla \mathcal{L}\left(\boldsymbol{\theta}_{t,i+1}^k\right) - \nabla \mathcal{L}\left(\boldsymbol{\theta}_{t,i}^k\right)$. In this term, the change of the gradient can be approximated by the change of the model parameters during the local iterations, i.e.,

$$\hat{\boldsymbol{H}}_{t,i}^k = \left(\Delta \boldsymbol{\theta}_{t,i+1}^k - \Delta \boldsymbol{\theta}_{t,i}^k\right) / \eta_t.$$

where $\Delta \boldsymbol{\theta}_{t,i+1}^k = \boldsymbol{\theta}_{t,i+1}^k - \boldsymbol{\theta}_{t,i+1}^k$ and $\Delta \boldsymbol{\theta}_{t,i}^k = \boldsymbol{\theta}_{t,i}^k - \boldsymbol{\theta}_{t,i}^k$.

The estimation of the Hessian Matrix before adding the regularization term in Equation 3 can be rewritten as a function of model parameter changes.

$$\widetilde{\boldsymbol{H}}_{t,i}^k = \left(\widetilde{\boldsymbol{\theta}}_{t,i+1}^k - \boldsymbol{\theta}_{t,i}^k - \Delta \boldsymbol{\theta}_{t,i}^k\right) / \eta_t \tag{4}$$

Adding Perturbation Our objective now is to perturb the estimated Hessian matrix such that the coefficient term $\left(\boldsymbol{I} - \eta_t \boldsymbol{H}_{t,i}^k\right)$ is minimized, i.e.,

$$\hat{\boldsymbol{H}}_{t,i}^k \leftarrow \operatorname{argmin}_{\widetilde{\boldsymbol{H}}_{t,i}^k}\left(\boldsymbol{I} - \eta_t \widetilde{\boldsymbol{H}}_{t,i}^k\right)$$

Combining it with Equation 4, optimizing $\widetilde{\boldsymbol{H}}$ is then equivalent to

$$\boldsymbol{\theta}_{t,i+1}^k \leftarrow \operatorname{argmin}_{\widetilde{\theta}_{t,i+1}^k}\left(\boldsymbol{I} - \left(\widetilde{\boldsymbol{\theta}}_{t,i+1}^k - \boldsymbol{\theta}_{t,i}^k - \Delta \boldsymbol{\theta}_{t,i}^k\right)\right) \tag{5}$$

Gradient Clipping To ensure that the model can converge after the regularization term is added, the gradients are clipped with the threshold q during the local training. The clipping function is defined as:

$$\operatorname{clip}\left(\nabla\left(\mathbf{I} - \eta_{t,i}\widetilde{\mathbf{H}}_{t,i}^k\right), q\right)_{r,c} = \begin{cases} \nabla\left(\mathbf{I} - \eta_{t,i}\widetilde{\mathbf{H}}_{t,i}^k\right)_{r,c}, & \left|\nabla\left(\mathbf{I} - \eta_{t,i}\widetilde{\mathbf{H}}_{t,i}^k\right)_{r,c}\right| \leq q \\ q, & \left|\nabla\left(\mathbf{I} - \eta_{t,i}\widetilde{\mathbf{H}}_{t,i}^k\right)_{r,c}\right| > q \end{cases}$$

Algorithm 1 `LeadFL` and robust aggregation

Input: number of global rounds T, local learning rate η_t, regularization rate α, clipping bound q, # of clients selected in a round K
for communication round $t = 0, 1, \cdots, T-1$ **do**
  Server randomly chooses K clients
  parallel on clients $k = 1, 2, \cdots, K$ **do**
    Update model weights as global weights from the last round $\boldsymbol{\theta}_t^k \leftarrow \boldsymbol{\theta}_t$;
    for local iteration $i = 0, 1, ...$ **do**
      Compute gradients and update weights $\widetilde{\boldsymbol{\theta}}_{t,i+1}^k \leftarrow \boldsymbol{\theta}_{t,i}^k - \eta_t \nabla \mathcal{L}\left(\boldsymbol{\theta}_{t,i}^k\right)$;
      Estimate Hessian matrix $\widetilde{\boldsymbol{H}}_{t,i}^k = \left(\widetilde{\boldsymbol{\theta}}_{t,i+1}^k - \boldsymbol{\theta}_{t,i}^k - \Delta \boldsymbol{\theta}_{t,i}^k\right) / \eta_t$
      Compute and Clip gradients of the regularization term $\boldsymbol{R}_{t,i}^k = \operatorname{clip}\left(\nabla\left(\boldsymbol{I} - \eta_t \widetilde{\boldsymbol{H}}_{t,i}^k\right), q\right)$;
      Update weights $\boldsymbol{\theta}_{t,i+1}^k \leftarrow \widetilde{\boldsymbol{\theta}}_{t,i+1}^k - \eta_t \alpha \boldsymbol{R}_{t,i}^k$;
    end for
    Compute updates $\boldsymbol{u}_t^k = \boldsymbol{\theta}_t^k - \boldsymbol{\theta}_t$;
  end parallel
  Aggregate updates using server-side defense: $\boldsymbol{u}_t = \operatorname{Aggregation}\left(\{\boldsymbol{u}_t^k\}_{k=1}^K\right)$
  Update $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \boldsymbol{u}_t$
end for
Output: $\{\theta^t\}_{t=0}^{T-1}$

where r and c are the indexes of rows and columns of the Matrix.

Algorithm To compute the model updates as shown in Equations 2 and 3, we adopt a two-step backpropagation process. We first allow the losses to backpropagate and then estimate the diagonal values of the Hessian matrix. Our second step is to use the estimated Hessian Matrix to compute the proposed regularization term and to allow the regularization loss to backpropagate. We summarize the key steps of `LeadFL` in Algorithm 1 and includes option of combining it with a server-side defense.

4.2. Convergence Analysis

In this part, we show that `LeadFL` converges under the same assumptions as other methods when there are no malicious clients attacking the FL system.We summarize these common assumptions in Appendix A.1.

In our defense, we add a new backpropagation process to perturb the Hessian matrix as shown in Equation 3. This extra backpropagation can be seen as a modification of gradients $\nabla \mathcal{L}$:

$$\nabla \mathcal{L}'\left(\boldsymbol{\theta}_{t,i}^k\right) = \nabla \mathcal{L}\left(\boldsymbol{\theta}_{t,i}^k\right) + \operatorname{clip}\left(\nabla\left(\boldsymbol{I} - \eta_t \widetilde{\boldsymbol{H}}_{t,i}^k\right), q\right) \tag{6}$$

Based on Assumption A.1 to A.5, we can derive the convergence guarantee of our defense on FedAvg as follows.

Theorem 4.1 (Convergence Guarantee)**.** *Let Assumptions A.1 to A.5 hold and $l, \mu, \sigma_k, G, K, N, \Gamma, \mathcal{L}^*$ be as defined therein and in Definition A.6 . Choose $\kappa = \frac{l}{\mu}$, $\gamma = \max\{8\kappa, E\}$ and the learning rate $\eta_t = \frac{2}{\mu(\gamma+t)}$. Then we have the following bound for* `LeadFL`*:*

$$\mathbb{E}\left[\mathcal{L}\left(\boldsymbol{\theta}_T\right)\right] - \mathcal{L}^* \leq \frac{\kappa}{\gamma + T - 1}\left(\frac{2(B+C)}{\mu} + \frac{\mu\gamma}{2}\mathbb{E}\left\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\right\|^2\right)$$

where

$$C = \frac{N-K}{N-1}\frac{4}{K}E^2(d^2q^2 + G^2)$$

$$B = \sum_{k=1}^{N} p_k^2(d^2q^2 + \sigma_k^2) + 6l\Gamma + 8(E-1)^2(d^2q^2 + G^2)$$

The proof is shown in Appendix A.2

4.3. Robustness Analysis

In this subsection, we use the certified radius framework proposed by (Panda et al., 2022) to analyze the robustness of `LeadFL`. We consider two types of threat models: periodic poisoned model submissions and bursty poisoned model submissions. Due to the space limitation, we provide the definitions and assumptions in Appendix A.1.

The certified radius is the upper bound on the distance between a poisoned model and a benign model. From (Xie et al., 2021), minimizing the certified radius improves robustness because close models are likely to have the same predictions. Based on the aforementioned assumptions and definitions, the certified radius for general protocols is proposed by (Panda et al., 2022).

Theorem 4.2. *Let f be a c-coordinatewise-Lipschitz protocol on a dataset D. Then $R(\rho) = \Lambda(T)(1+dc)^{\Lambda(T)}\rho$ is a certified radius for f, where $\Lambda(t)$ is the cumulative learning rate $\Lambda(t) = \sum_{t=0}^{T-1}\eta_t$, d is the dimension of model parameters.*

4.3.1. Scenario I

Scenario I assumes a simplified model of bursty adversarial patterns, namely the most extreme pattern where periodically, a large number of clients is malicious and there are no malicious clients in the other rounds. Concretely, malicious clients submit poisoned updates in global round T_A. Afterwards, there are no malicious updates submitted between round T_A and round $T-1$.

Theorem 4.3 (Certified Radius in Scenario I)**.** *Let Assumptions A.9 hold and T_A,c be as defined therein. We assume that* `LeadFL` *with FedAv aggregation is used. The certified radius satisfies:*

$$R(\rho) = \left(\frac{N}{K}\right)^{T-T_A}\left|\prod_{t=T_A}^{T}\left[\sum_{k\in S_t}p^k\prod_{i=0}^{I-1}\left(I - \eta_t H_{t,i}^k\right)\right]\right| \cdot \sum_{t=0}^{T_A-1}\eta_t(1+dc)^{\sum_{t=0}^{T_A-1}\eta_t}\rho$$

The proof is in Appendix A.3. As `LeadFL` aims to minimize $\left(I - \eta_t H_{t,i}^k\right)$ in the local training as by Equation 6. Hence, `LeadFL` achieves a low certified radius.

4.3.2. Scenario II

Here, we consider a more general threat model, the number of malicious clients varies between rounds with resulting bursty adversarial patterns. Concretely, we assume that the clients are selected randomly. We furthermore assume the presence of a server-side defense that filters out updates.

The probability of a server-side defense filtering out all malicious updates is correlated to the number of malicious clients selected in a communication round. For an attack atk, we use $g_{atk}(\cdot)$ to represent the above correlation. The probability of a server-side defense filtering out all malicious updates in global round t can be presented as $\phi_{atk}^t = g_{atk}(K_m^t)$, where K_m^t is the number of malicious clients selected in round t. We then can derive the certified radius of `LeadFL` combined with any given server-side defense under bursty adversarial patterns as:

Theorem 4.4 (Certified Radius in Scenario II)**.** *Let Assumption A.9 hold. The certified radius of the threat model is*

$$R(\rho) = (1+dc)^{\sum_{t\in\Phi_T}\eta_t}\rho \cdot \left(\left|\prod_{t\in\Gamma_T}\left[\frac{N}{|S_t^*|}\sum_{k\in S_t^*}p^k\prod_{i=0}^{I-1}\left(I - \eta_t H_{t,i}^k\right)\right]\right| + |\Phi_T|\sum_{t\in\Phi_T}\eta_t\right)$$

where Φ_T is the set of communication rounds that server-side defenses cannot filter out all malicious updates. Γ_T is the set of communication rounds that server-side defenses filter out all malicious updates. S_t^* is a set of clients whose updates are not filtered out by the server-side defense in round t. $|\Phi_T|$ and $|S_t^*|$ are the cardinality of the set Φ_T and S_t^*, where $\mathbb{E}\left[|\Phi_T|\right] = \sum_{t=0}^{T-1} g_{atk}(K_m^t)$.

Note that the value $|\Phi_T|$ depends entirely on the server-side defense. In the absence of a server-side defense, the certified radius is hence large, so we need the server-side defense to lower it.

Table 1: Comparison of defenses under 9-pixel pattern backdoor attack on IID and non-IID FashionMNIST dataset.

Distribution	IID									Non-IID								
Server-side Defense	SparseFed	Multi-Krum				Bulyan				SparseFed	Multi-Krum				Bulyan			
Client-side Defense	None	None	LDP	FL-WBC	Ours	None	LDP	FL-WBC	Ours	None	None	LDP	FL-WBC	Ours	None	LDP	FL-WBC	Ours
MA	85.9	89.3	87.0	87.2	**87.9**	89.2	86.0	86.0	**86.3**	84.9	85.6	76.7	77.2	**79.1**	77.4	73.4	71.7	**74.0**
BA Avg	97.9	82.6	76.0	77.5	**32.9**	78.8	74.1	70.6	**21.6**	99.8	88.7	80.4	74.0	**39.5**	92.5	71.9	73.7	**32.3**
BA Final	99.9	93.2	79.6	80.6	**0.0**	90.6	62.2	86.5	**0.3**	99.9	93.3	86.7	70.3	**1.2**	88.6	94.7	69.0	**2.0**

Table 2: Comparison of defenses under 9-pixel pattern backdoor attack on IID and non-IID CIFAR10 dataset.

Distribution	IID									Non-IID								
Server-side Defense	SparseFed	Multi-Krum				Bulyan				SparseFed	Multi-Krum				Bulyan			
Client-side Defense	None	None	LDP	FL-WBC	Ours	None	LDP	FL-WBC	Ours	None	None	LDP	FL-WBC	Ours	None	LDP	FL-WBC	Ours
MA	50.9	76.3	48.0	43.3	**56.9**	76.2	41.5	42.2	**54.8**	55.3	70.7	43.2	42.9	**55.3**	61.7	36.7	36.2	**51.4**
BA Avg	95.8	77.5	53.1	56.9	**35.6**	79.1	46.7	51.3	**43.9**	45.2	85.8	55.4	54.4	**45.2**	87.5	48.8	48.1	**46.8**
BA Final	98.5	80.5	43.8	40.5	**25.6**	87.0	23.4	35.5	**21.4**	34.4	96.2	52.4	35.4	**34.4**	95.2	29.8	47.7	**27.3**

5. Evaluation

In this section, we demonstrate the effectiveness of `LeadFL` for multiple server-side defenses. We consider heterogeneous data distributions and compare against state-of-the-art client-side defense mechanisms. Furthermore, our ablation study confirms that a combination of server-side and client-side defenses succeeds in mitigating attacks that are highly effective in the presence of either of the two.

We perform all experiments using PyTorch's deep learning framework (Paszke et al., 2019) in combination with the FLTK Testbed [1]. We reimplemented FL-WBC, LDP, and the targeted attacks based on the source code of FL-WBC [2] to compare them with our defense. Additionally, we reimplement SparseFed and backdoor attacks based on the source code provided by (Panda et al., 2022) [3] and (Bagdasaryan & Shmatikov, 2021) [4], respectively. Our code can be found at `https://github.com/CarlosChu-c/LeadFL`.

5.1. Evaluation Metrics

Our goal is to achieve high accuracy for the main task but mitigate the backdoor. Thus, we primarily focus on the following three metrics:

- **Main Task Accuracy (MA)**: The main task accuracy is the fraction of correctly classified samples of the model on test data without the trigger. As other works, we consider the maximum accuracy achieved during training.
- **Backdoor Accuracy (BA)**: The backdoor accuracy qualifies how successful the attacker is in integrating a backdoor into the model. We measure backdoor accuracy as the percentage of samples with the trigger that are classified as intended by the attacker. We found that the backdoor accuracy does not converge during our experiments, hence we consider both the average and the final backdoor accuracy. The final backdoor accuracy is the one of the model that is later used but it does not give a full picture due to the high variance in backdoor accuracy over rounds, which is why we also include the average backdoor accuracy.
- **Mitigation rounds**: Our attacks do not have the same strength in every round due to the fact that the number of malicious clients selected varies between rounds. When a lot of malicious clients are involved, the backdoor accuracy spikes and then decreases again. After a strong attack that achieves a temporary backdoor accuracy of more than 50%, we define the mitigation rounds as the number of communication rounds until the backdoor accuracy drops below 50%.

5.2. Evaluation Setup

In each simulation run, we have a set of clients. During each round, the server selects clients. The clients train and apply the client-side defense during training. Afterwards, the server aggregates the local updates submitted by the clients, applying the server-side defense during aggregation.

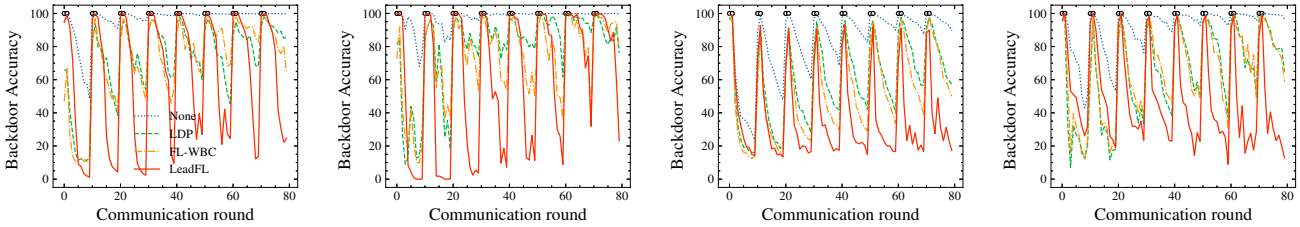
Client Selection and Rounds There are 100 clients in total, of which 25 are malicious. There are 80 global rounds and 10 local rounds. The server selects 10 clients per global round. For most experiments, the selection is random but consistent over experiments, i.e., for two experiments, the clients selected in round t are the same to enable comparison between the different settings. Figure 2a displays the number of malicious clients per round. In order to ensure that our results are not an artifact of this one specific client selection, we present results for other selections in the Appendix D.1

In previous work, periodic attacks alternating between a large number of malicious selected clients and no malicious selected clients have been evaluated. In addition to random selecting, we hence also use a selection corresponding to such a periodic attack: For every 10 global rounds, 6 of

[1] `https://github.com/JMGaljaard/fltk-testbed`
[2] `https://github.com/jeremy313/FL-WBC`
[3] `https://github.com/sparsefed/sparsefed`
[4] `https://github.com/ebagdasa/backdoors101`

Table 3: Comparison of defenses under 9-pixel pattern backdoor attack on IID CIFAR100 dataset.

Distribution	IID								
Server-side Defense	SparseFed	Multi-Krum				Bulyan			
Client-side Defense	None	None	LDP	FL-WBC	Ours	None	LDP	FL-WBC	Ours
MA	32.3	38.4	28.3	28.8	**30.6**	37.4	25.6	25.8	**27.2**
BA Avg	85.1	58.0	57.3	53.8	**29.3**	56.1	49.3	48.3	**29.0**
BA Final	68.3	52.2	34.5	29.2	**6.4**	32.8	21.7	20.4	**3.5**



(a) FashionMNIST-IID (b) FashionMNIST-NonIID (c) CIFAR10-IID (d) CIFAR10-NonIID

Figure 3: Comparison of 9-pixel back door attack on FashionMNIST and CIFAR10 when the attack is periodic.

10 clients are malicious in the first two rounds while the remaining 8 rounds only have honest clients.

Datasets and Model Architecture We conduct experiments on FashionMNIST, CIFAR10 and CIFAR100, which are both benchmark tasks in image classification. For FashionMNIST, each of the 100 clients receives 600 images out of 60.000. For CIFAR10 and CIFAR100, each client gets 500 out of 50.000.

In the IID setting, samples are uniformly distributed to clients. In the non-IID setting, we deploy the limited label strategy (McMahan et al., 2017) that is also used for the evaluation of FL-WBC in FashionMNIST and CIFAR10: Of the 10 classes in each of the two datasets, each client is assigned 5 random classes. They are then assigned an equal number of randomly selected samples from each of their classes. The clients' datasets are selected independently.

We adopt the same model architectures as FL-WBC (Sun et al., 2021) on FahionMNIST and CIFAR10. On FashionMNIST, we employ two convolutional layers and one fully-connected layer. Our CIFAR10 model consists of two convolutional layers and three fully-connected layers. And on CIFAR100, we employ ResNet9 (He et al., 2016), which is a more complicated model. The detail of the model architecture and hyperparameters can be found in Appendix B.

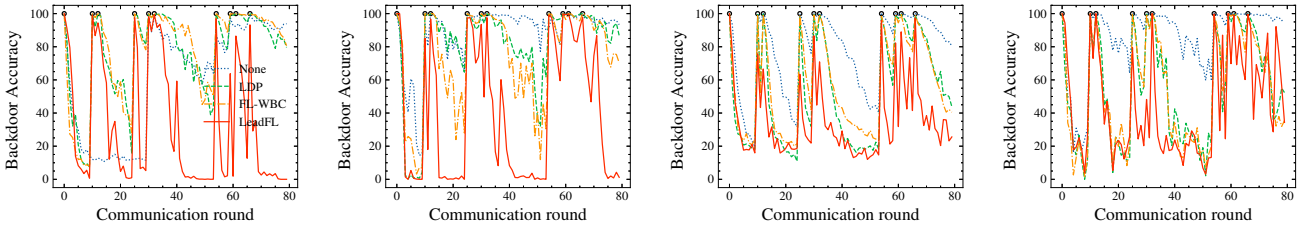
Attacks and Defenses For attacks, we evaluate both state-of-the-art backdoor and targeted attacks. In terms of backdoor attacks, we use the 9-pixel pattern backdoor attacks and the single-pixel backdoor attacks from (Bagdasaryan & Shmatikov, 2021). As a targeted attack, we evaluate the single-image targeted attacks from (Bhagoji et al., 2019): All malicious clients add one incorrectly classified image to their otherwise clean dataset; it is the same image for all clients. We use the settings that achieved the best results in the original papers.

Here, we use Multi-Krum (Blanchard et al., 2017) and Bulyan (Mhamdi et al., 2018) as server-side defenses. We also compare SparseFed (Panda et al., 2022), one of the state-of-the-art defenses against poisoning attacks in FL. Moreover, we considered CMA (Yin et al., 2018) and CTMA (Yin et al., 2018) but they had very little effect in comparison to the other defenses, so we only include the corresponding results in the Appendix D.3.Note that our protocol can enhance any other server-side defense as well.

For client-side defenses, we choose FL-WBC and LDP as the baseline. For these two defenses, we apply Laplace noise with $mean = 0$ and $std = 0.2$ as in the original papers. For our defense, we set the clipping norm $q = 0.2$. For the regularization term, we use hyperparameter tuning to choose $\alpha = 0.4$ for FashionMNIST, $\alpha = 0.25$ for CIFAR10, and $\alpha = 0.15$ for CIFAR100.

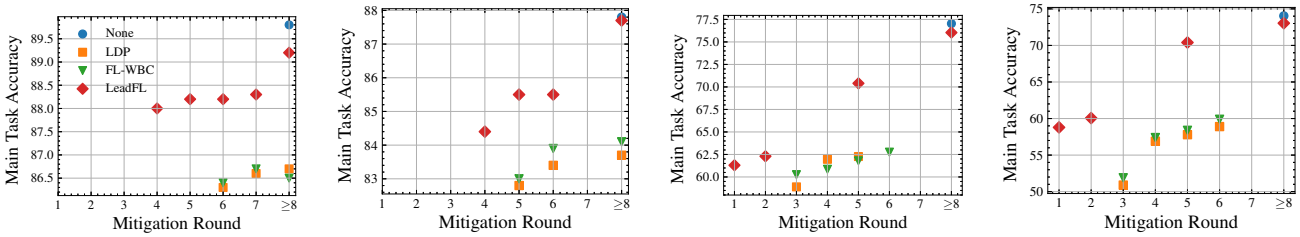
5.3. Results

Table 1, 2 and 3 show the results for the 9-pixel backdoor attack. In our threat model, SparseFed presents limited effectiveness in defending against poisoning attacks, achieving higher Backdoor accuracy than Multi-Krum and Bulyan across all three datasets. And it can be seen that our defense achieves the highest main task accuracy and lowest backdoor accuracy. In comparison to the case without a client-side defense, the main task accuracy is reduced by less than 10% whereas the final backdoor accuracy is 0 or close to 0 for FashionMNIST. For CIFAR10, the main task accuracy of our defense is between 50% and 60% and the final backdoor accuracy is between 20% and 35%. The average backdoor



(a) FashionMNIST-IID (b) FashionMNIST-NonIID (c) CIFAR10-IID (d) CIFAR10-NonIID

Figure 4: Comparison of 9-pixel pattern backdoor accuracy on FashionMNIST and CIFAR10. The server-side defense here is Multi-Krum. Black hollow circles indicate that the system is attacked very strongly in that round.



(a) FashionMNIST-IID (b) FashionMNIST-NonIID (c) CIFAR10-IID (d) CIFAR10-NonIID

Figure 5: Various defenses' tradeoffs between main task accuracy and mitigation rounds on FashionMNIST and CIFAR10.

accuracy for our defense is higher than the final accuracy but always below 50% and lower than the backdoor accuracy of the state-of-the-art defenses. For CIFAR100, the main task accuracy is about 30% and the final backdoor accuracy is only between 3% and 7%. Indeed, without a client-side defense, the final and average backdoor accuracy is always above 75% in FashionMNIST and CIFAR10, and 55% in CIFAR100, meaning that the server-side defense on its own is ineffective. The other client-side defenses are considerably less effective than ours: For FashionMNIST, they have a final and average backdoor accuracy of above 69%, a stark contrast to our defense, especially for the final backdoor accuracy. For CIFAR10, the difference is less pronounced, with backdoor accuracies that are only about 10% higher than for our defense. However, the main task accuracy of the other defenses falls below 50% for CIFAR10. For CIFAR100, although the main task accuracy of our defense is only about 2% higher than other client-side defenses, the average backdoor accuracy and final backdoor accuracy of our defense is about 20% lower than other defenses.

While our defense is hence an improvement over existing defenses, there are notable differences between settings. Non-IID distributions of data reduce the main task accuracy and increase the backdoor accuracy for all defenses. The result is unsurprising: The more uniform benign clients are, the easier it is to detect malicious clients whose model updates differ. However, if client data and hence models already differ between benign clients, it becomes more difficult to identify and mitigate malicious behavior.

In order to analyze how the backdoor accuracy is affected by the number of attackers, we consider the backdoor accuracy over the duration of the experiment. Figure 4 displays the backdoor accuracy. We can see that whenever the number of malicious clients exceeds the number of benign clients, i.e., if there are at least 6 malicious clients selected in a round, the backdoor is successfully embedded into the model, as shown by a high backdoor accuracy of close to 100%. In subsequent rounds with a lower amount of malicious clients, the backdoor accuracy decreases. Our defense exhibits a faster decrease in backdoor accuracy than the other defenses, which results in the lower final and average backdoor accuracy seen above. The same pattern is observed for both datasets and levels of data heterogeneity, although the speed of recovery is faster for iid data distributions.

We compare this behavior for random client selection with the periodic attack described in Section 5.2. We observe the same pattern, displayed in Figure 3, as when selecting clients randomly, only that it is now periodic. For the periodic setting, we derive the number of mitigation rounds, as defined in Section 5.1. As the delay between two attacks is always the same and the attacks are of the same severity, the periodic setting enables use to compare recovery in a fair manner. We can then analyze whether there is a trade-off between mitigation rounds, i.e., strength of the defense, and main task accuracy.

Concretely, for each experiment and each attack, we compute the number of mitigation rounds. If the backdoor ac-

curacy does not recover during the 8 rounds between two attacks, we use ≥ 8 for the number of mitigation rounds. For a defense d, we then compute $MA_{r,d}$, the average main task accuracy over all experiments for d that have r mitigation rounds[5]. Figure 5 shows the results. Our defense achieves a better trade-off between main task accuracy and recovery, i.e., for the same number of mitigation rounds, it has a higher main task accuracy. An exception is the case ≥ 8 with no client-side defense having a higher main task accuracy, which makes sense as if our defense does not lead to recovery, not applying a defense is the better option. However, usually our defense successfully mitigates the attack and if it does so, it has a higher main task accuracy than other defenses.

All the presented results are for the 9-pixel attack. The results for the 1-pixel attack and the single-image targeted attack are similar (see Appendix D.4 and D.5).

6. Conclusion

To defend against model poisoning attacks with bursty adversarial patterns, we propose a novel client-side self defense, `LeadFL`, which perturbs the local model updates by adding a novel regularization term based on the Hessian matrix of the gradients. Thanks to the optimized regularization, `LeadFL` effectively thwarts backdoor and targeted attacks with a low degradation of the main task accuracy, proven theoretically and empirically. Evaluated on FashionMNIST, CIFAR10 and CIFAR100, `LeadFL` combined with a server-side defense can reduce the backdoor accuracy by up to 65 % in comparison only using a server-side defense.

Acknowledgment

This work has been partly funded by the Dutch National Science Foundation Perspectief Project, DEPMAT.

References

Bagdasaryan, E. and Shmatikov, V. Blind backdoors in deep learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1505–1521, 2021.

Bhagoji, A. N., Chakraborty, S., Mittal, P., and Calo, S. Analyzing federated learning through an adversarial lens. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 634–643. PMLR, 09–15 Jun 2019.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Fung, C., Yoon, C. J., and Beschastnikh, I. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pp. 301–316, 2020.

Gupta, A., Luo, T., Ngo, M. V., and Das, S. K. Long-short history of gradients is all you need: Detecting malicious and unreliable clients in federated learning. In *European Symposium on Research in Computer Security*, pp. 445–465. Springer, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Konečný, J., McMahan, H. B., Yu, F. X., Richtarik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

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

Mhamdi, E. M. E., Guerraoui, R., and Rouault, S. The hidden vulnerability of distributed learning in byzantium. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 3521–3530. PMLR, 2018. ISSN: 2640-3498.

Muñoz-González, L., Co, K. T., and Lupu, E. C. Byzantine-robust federated machine learning through adaptive model averaging, 2019.

Naseri, M., Hayes, J., and De Cristofaro, E. Local and central differential privacy for robustness and privacy in federated learning. In *Proceedings 2022 Network and Distributed System Security Symposium*. Internet Society, 2022. ISBN 978-1-891562-74-7. doi: 10.14722/ndss.2022.23054.

[5]We round it up the nearest integer.

Nguyen, T. D., Rieger, P., Chen, H., Yalame, H., Möllering, H., Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Zeitouni, S., et al. {FLAME}: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1415–1432, 2022.

Ozdayi, M. S., Kantarcioglu, M., and Gel, Y. R. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9268–9276, 2021.

Panda, A., Mahloujifar, S., Bhagoji, A. N., Chakraborty, S., and Mittal, P. Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In *International Conference on Artificial Intelligence and Statistics*, pp. 7587–7624. PMLR, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Rieger, P., Nguyen, T. D., Miettinen, M., and Sadeghi, A.-R. DeepSight: Mitigating backdoor attacks in federated learning through deep model inspection. In *Proceedings 2022 Network and Distributed System Security Symposium*. Internet Society, 2022. ISBN 978-1-891562-74-7. doi: 10.14722/ndss.2022.23156.

Steinhardt, J., Koh, P. W. W., and Liang, P. S. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30, 2017.

Sun, J., Li, A., DiValentin, L., Hassanzadeh, A., Chen, Y., and Li, H. Fl-wbc: Enhancing robustness against model poisoning attacks in federated learning from a client perspective. *Advances in Neural Information Processing Systems*, 34, 2021.

Tolpegin, V., Truex, S., Gursoy, M. E., and Liu, L. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*, pp. 480–501. Springer, 2020.

Xia, Q., Tao, Z., Hao, Z., and Li, Q. Faba: An algorithm for fast aggregation against byzantine attacks in distributed neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4824–4830. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/670.

Xie, C., Huang, K., Chen, P.-Y., and Li, B. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019.

Xie, C., Chen, M., Chen, P.-Y., and Li, B. CRFL: Certifiably robust federated learning against backdoor attacks. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 11372–11382. PMLR, 2021. ISSN: 2640-3498.

Xu, J., Huang, S.-L., Song, L., and Lan, T. Byzantine-robust federated learning through collaborative malicious gradient filtering. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pp. 1223–1235. IEEE, 2022.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5650–5659. PMLR, 2018. ISSN: 2640-3498.

Nomenclature

α	weights of regularization term in loss function
$\boldsymbol{\theta}$	model weights
$\boldsymbol{\theta}^*$	poisoned model weights
$\boldsymbol{H}$	Hessian Matrix
$\boldsymbol{I}$	Attack Effect on Parameter
$\boldsymbol{I}$	Identity Matrix
$\boldsymbol{M}$	mask matrix in FL-WBC
$\boldsymbol{u}$	model updates
δ	Attack Effect on Parameter
ℓ	parameter in smoothness assumption
η	learning rate
Γ_T	Set of communication rounds that server-side defenses filter out all malicious updates before round T
$\hat{\delta}$	estimate of Attack Effect on Parameter
$\mathbb{S}$	set of clients selected in a round
$\mathcal{G}$	Gradient oracle
$\mathcal{L}$	loss function of benign clients
$\mathcal{L}_M$	loss function of malicious clients
Φ_T	Set of communication rounds that server-side defenses cannot filter out all malicious updates before round T
π	portion of benign loss in malicious clients
ρ	Bound of poisoning Attacks
σ	bound of variance of stochastic gradients
c	index of columns in the Matrix
G	bound of norm of stochastic gradients
I	total number of local iterations
i	local iteration index
K	total number of clients selected in a round
k	index of clients selected in a round
K_m	the number of malicious clients selected in a round
N	number of clients in a system
N_m	number of malicious clients in a system
p	Weights in aggregation
p_X	probability mass function of the number of malicious clients selected in a round
q	clipping bounds in our method
r	index of rows in the Matrix
T	total number of communication rounds
t	communication round index

A. Proofs

A.1. Assumptions and Definitions

Assumption A.1 (Smoothness)**.** $\mathcal{L}$ is ℓ-smooth if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{R}^d$

$$\mathcal{L}(\boldsymbol{x}) - \mathcal{L}(\boldsymbol{y}) + (\boldsymbol{x} - \boldsymbol{y})^T \mathcal{L}(\boldsymbol{x}) \leq \frac{\ell}{2} \|x - y\|_2^2$$

Assumption A.2 (Convex)**.** $\mathcal{L}$ is μ-strongly convex if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{R}^d$,

$$\mathcal{L}(\mathbf{x}) - \mathcal{L}(\mathbf{y}) + (\mathbf{x} - \mathbf{y})^T \nabla \mathcal{L}(\mathbf{y}) \geq \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|_2^2$$

Assumption A.3 (Bound of Variance)**.** Let ξ_t^k be sampled from the k-th device's local data uniformly at random. The variance of stochastic gradients in each device is bounded: $\mathbb{E} \left\| \nabla \mathcal{L}^k \left(\boldsymbol{\theta}_t^k, \xi_t^k \right) - \nabla \mathcal{L}^k \left(\boldsymbol{\theta}_t^k \right) \right\|^2 \leq \sigma_k^2$ for $k = 1, \cdots, N$.

Assumption A.4 (Bound of Norm)**.** The expected squared norm of stochastic gradients is uniformly bounded, i.e., $\mathbb{E} \left\| \nabla \mathcal{L}^k \left(\boldsymbol{\theta}_{t,i}^k, \xi_{t,i}^k \right) \right\|^2 \leq G^2$ for all $k = 1, \ldots, N, i = 0, \ldots, I-1$ and $t = 0, \ldots, T-1$

In our FL system, K clients are randomly selected among N clients each round. Then we adapt the following assumption from (Li et al., 2019).

Assumption A.5 (Selection of Clients)**.** Assume $\mathcal{S}_t$ contains a subset of K indices uniformly sampled from $[N]$ without replacement. Assume the data is balanced in the sense that $p_1 = \cdots = p_N = \frac{1}{N}$. The aggregation step of FedAvg performs $\boldsymbol{\theta}_t \longleftarrow \frac{N}{K} \sum_{k \in \mathcal{S}_t} p_k \boldsymbol{\theta}_t^k$.

Definition A.6 (Loss of clients)**.** Denote $\mathcal{L}^*$ and $\mathcal{L}_k^*$ as the minimum value of $\mathcal{L}$ and $\mathcal{L}_k$, where $\mathcal{L}$ is the loss of a model trained on the combination of datasets from all the clients and $\mathcal{L}_k$ is the loss of a model trained on the dataset of client k. We can set $\Gamma = \mathcal{L}^* - \sum_{k=1}^N p_k \mathcal{L}_k^*$ which can quantify the degree of noniid. If the data are iid, then Γ goes to zero as the number of samples grows. If the data are noniid, then Γ is nonzero, and its magnitude reflects the heterogeneity of the data distribution.

Definition A.7 (Poisoning Attack)**.** For a protocol $f = (\mathcal{G}, \mathcal{A}, \eta)$ we define the set of poisoned protocols $F(\rho)$ to be all protocols $f^* = (\mathcal{G}^*, \mathcal{A}, \eta)$ that are exactly the same as f except that the gradient oracle $\mathcal{G}^*$ is a ρ-corrupted version of $\mathcal{G}$. That is, for any round t and any model θ_t and any dataset D we have we have $\mathcal{G}^* (\theta_t, D) = \mathcal{G} (\theta_t, D) + \epsilon$ for some ϵ with $\|\epsilon\|_1 \leq \rho$

Definition A.8 (Certified Radius)**.** Let f be a protocol and $f^* \in F(\rho)$ be a poisoned version of the same protocol. Let θ_T, θ_T^* be the benign and poisoned final outputs of the above protocols. We call R a certified radius for f if $\forall f^* \in F(\rho); R(\rho) \geq |\theta_T - \theta_T^*|_1$

Assumption A.9 (Coordinate-wise Lipschitz)**.** The protocol $f(\mathcal{G}, \mathcal{A}, \eta)$ is c-coordinate-wise Lipschitz if for any round $t \in [T]$, models $\theta_t, \theta_t^* \in \mathcal{M}$, and a dataset D we have that the outputs of the gradient oracle on any coordinate cannot drift too much farther apart. Specifically, for any coordinate index $i \in [d]$

$$|\mathcal{G} (\theta_t^*, D) [i] - \mathcal{G} (\theta_t, D) [i]| \leq c \cdot |\theta_t^* - \theta_t|_1$$

A.2. Proof of Theorem 4.1

Theorem 4.1 (Convergence Guarantee)**.** *Let Assumptions A.1 to A.5 hold and $l, \mu, \sigma_k, G, K, N, \Gamma, \mathcal{L}^*$ be as defined therein and in Definition A.6 . Choose $\kappa = \frac{l}{\mu}$, $\gamma = \max\{8\kappa, E\}$ and the learning rate $\eta_t = \frac{2}{\mu(\gamma + t)}$. Then we have the following bound for* `LeadFL`*:*

$$\mathbb{E} \left[\mathcal{L} (\boldsymbol{\theta}_T) \right] - \mathcal{L}^* \leq \frac{\kappa}{\gamma + T - 1} \left(\frac{2(B + C)}{\mu} + \frac{\mu \gamma}{2} \mathbb{E} \left\| \boldsymbol{\theta}_0 - \boldsymbol{\theta}^* \right\|^2 \right)$$

where

$$C = \frac{N-K}{N-1}\frac{4}{K}E^2(d^2q^2+G^2)$$

$$B = \sum_{k=1}^{N} p_k^2(d^2q^2+\sigma_k^2) + 6l\Gamma + 8(E-1)^2(d^2q^2+G^2)$$

Proof: The expected distance between the gradients before and after regularization can be bounded.

$$\begin{aligned}
&\mathbb{E}\left\|\nabla\mathcal{L}'_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)-\nabla\mathcal{L}_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)\right\|_2^2\\
=&\mathbb{E}\left\|\text{clip}\left(\nabla\left(\mathbf{I}-\eta_t\hat{\mathbf{H}}^k_{t,i}\right),q\right)\right\|_2^2\\
\leq&\mathbb{E}\|q\|_2^2 = d^2q^2
\end{aligned} \tag{7}$$

Using the bounds above and Assumption A.3, we can derive new bounds for the variance of modified gradients $\mathbb{E}\left\|\nabla\mathcal{L}'_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)-\nabla\mathcal{L}_k\left(\boldsymbol{\theta}^k_{t,i}\right)\right\|^2$

$$\begin{aligned}
&\mathbb{E}\left\|\nabla\mathcal{L}'_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)-\nabla\mathcal{L}_k\left(\boldsymbol{\theta}^k_{t,i}\right)\right\|^2\\
\leq&\mathbb{E}\left\|\nabla\mathcal{L}'_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)-\nabla\mathcal{L}_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)\right\|^2\\
&+\mathbb{E}\left\|\nabla\mathcal{L}_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)-\nabla\mathcal{L}_k\left(\boldsymbol{\theta}^k_{t,i}\right)\right\|^2\\
\leq& d^2q^2+\sigma_k^2,
\end{aligned}$$

where we use the triangle inequality.

Similarly, we can also derive bounds the expected squared norm of modified gradients using Assumption A.4.

$$\begin{aligned}
&\mathbb{E}\left\|\nabla\mathcal{L}'_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)\right\|^2\\
\leq&\mathbb{E}\left\|\nabla\mathcal{L}'_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)-\nabla\mathcal{L}_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)\right\|^2\\
&+\mathbb{E}\left\|\nabla\mathcal{L}_k\left(\boldsymbol{\theta}^k_{t,i},\xi^k_{t,i}\right)\right\|^2\\
\leq&d^2q^2+G^2,
\end{aligned}$$

Applying the bounds for the variance and the expected squared norm of modified gradients after applying `LeadFL`, we can derive our convergence guarantee from Theorem 3 in (Li et al., 2019) by replacing these bounds.

A.3. Proof of Theorem 4.3

Theorem 4.3 (Certified Radius in Scenario I)**.** *Let Assumptions A.9 hold and T_A,c be as defined therein. We assume that* `LeadFL` *with FedAv aggregation is used. The certified radius satisfies:*

$$\begin{aligned}
R(\rho) = &\left(\frac{N}{K}\right)^{T-T_A}\left|\prod_{t=T_A}^{T}\left[\sum_{k\in\mathrm{S}_t}p^k\prod_{i=0}^{I-1}\left(I-\eta_t H^k_{t,i}\right)\right]\right|\\
&\cdot\sum_{t=0}^{T_A-1}\eta_t(1+dc)^{\sum_{t=0}^{T_A-1}\eta_t}\rho
\end{aligned}$$

Proof. Equation 1 can be rewritten as follows:

$$\boldsymbol{\theta}_t-\boldsymbol{\theta}^*_t = \frac{N}{K}\left[\sum_{k\in S_t}p^k\prod_{i=0}^{I-1}\left(I-\eta_t H^k_{t,i}\right)\right]\left(\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}^*_{t-1}\right) \tag{8}$$

This equation can be used iteratively to get:

$$\begin{aligned}
\boldsymbol{\theta}_T-\boldsymbol{\theta}^*_T = &\left(\frac{N}{K}\right)^{T-T_A}\prod_{t=T_A}^{T}\left[\sum_{k\in S_t}p^k\prod_{i=0}^{I-1}\left(I-\eta_t H^k_{t,i}\right)\right]\\
&\left(\boldsymbol{\theta}_{T_A}-\boldsymbol{\theta}^*_{T_A}\right)
\end{aligned} \tag{9}$$

Apply the Theorem 4.2, we can get:

$$\left|\boldsymbol{\theta}_{T_A} - \boldsymbol{\theta}^*_{T_A}\right| \leq \cdot \sum_{t=0}^{T_A-1} \eta_t (1+dc)^{\sum_{t=0}^{T_A-1}\eta_t}\rho \tag{10}$$

Combine Equations 9 and 10, the certified radius can be derived:

$$\begin{aligned} |\boldsymbol{\theta}_T - \boldsymbol{\theta}^*_T| \leq \left(\frac{N}{K}\right)^{T-T_A} \left| \prod_{t=T_A}^{T} \left[\sum_{k\in \mathrm{S}_t} p^k \prod_{i=0}^{I-1} \left(I - \eta_t H^k_{t,i}\right)\right]\right| \\ \cdot \sum_{t=0}^{T_A-1} \eta_t (1+dc)^{\sum_{t=0}^{T_A-1}} \rho \end{aligned} \tag{11}$$

A.4. Proof of Theorem 4.4

Theorem 4.4 (Certified Radius in Scenario II)**.** *Let Assumption A.9 hold. The certified radius of the threat model is*

$$\begin{aligned} R(\rho) = (1+dc)^{\sum\limits_{t\in\Phi_T}\eta_t}\rho\cdot \\ \left(\left|\prod_{t\in\Gamma_T}\left[\frac{N}{|S^*_t|}\sum_{k\in S^*_t} p^k \prod_{i=0}^{I-1}\left(I-\eta_t H^k_{t,i}\right)\right]\right| + |\Phi_T| \sum_{t\in\Phi_T}\eta_t\right) \end{aligned}$$

where Φ_T is the set of communication rounds that server-side defenses cannot filter out all malicious updates. Γ_T is the set of communication rounds that server-side defenses filter out all malicious updates. S^*_t is a set of clients whose updates are not filtered out by the server-side defense in round t. $|\Phi_T|$ and $|S^*_t|$ are the cardinality of the set Φ_T and S^*_t, where $\mathbb{E}\left[|\Phi_T|\right] = \sum_{t=0}^{T-1} g_{atk}(K^t_m)$.

Proof. Denote $f^* = (\mathcal{G}^*, \mathcal{A}, \eta) \in f(\rho)$ as an arbitrary ρ-poisoned version of f in Definition A.7. And let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T$ and $\boldsymbol{u}^*_1, \ldots, \boldsymbol{u}^*_T$ be the model updates that the benign oracle $\mathcal{G}$ would produce on models $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{T-1}$ and $\boldsymbol{\theta}^*_0, \ldots, \boldsymbol{\theta}^*_{T-1}$, respectively. We also define $\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_T$ to be the output of the adversarial gradient oracle $\mathcal{G}^*$ on models $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{T-1}$. By the definition of ρ-poisoning in Definition A.7, we have $|\hat{\boldsymbol{u}}_t - \boldsymbol{u}^*_t|_1 \leq \rho$.

Based on the definition of model updates, we use the triangle inequality to get the following inequality between $|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*_t|$ and $\left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*_{t-1}\right|$ when the system is attacked in round $t-1$

$$|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*_t| = \left|\boldsymbol{\theta}_{t-1} - \eta_t u_t - \boldsymbol{\theta}^*_{t-1} + \eta_t \hat{u}_t\right| \leq \left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*_{t-1}\right| + \eta_t \left|u_t - \hat{u}_t\right| \tag{12}$$

Using the triangle inequality again, we can get

$$|u_t - \hat{u}_t| = |u_t - u^*_t + u^*_t - \hat{u}_t| \leq |u_t - u^*_t| + |u^*_t - \hat{u}_t| \tag{13}$$

According to Definition A.7 and coordinate-wise Lipshitz in Assumption A.9:

$$|u_t - \hat{u}_t| \leq |u_t - u^*_t| + |u^*_t - \hat{u}_t| = dc\left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*_{t-1}\right| + \rho \tag{14}$$

By plugging the above equation into Equation 12, we get

$$|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*_t| \leq \left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*_{t-1}\right| + \eta_t\left(dc\left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*_{t-1}\right| + \rho\right) = (1+dc\eta_t)\left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*_{t-1}\right| + \rho\eta_t \tag{15}$$

According to Bernoulli's inequality, we have

$$|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*_t| \leq (1+dc)^{\eta_t}\left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}^*_{t-1}\right| + \rho\eta_t \tag{16}$$

Table 4: Model Architectures for FashionMNIST and CIFAR10 dataset

FashionMNIST	CIFAR10
5×5 Conv2d 1-16	3×3 Conv2d 3-32
5×5 Conv2d 16-32	3×3 Conv2d 32-32
FC-10	2×2 MaxPool
	3×3 Conv2d 32-64
	3×3 Conv2d 64-64
	2×2 MaxPool
	3×3 Conv2d 64-128
	3×3 Conv2d 128-128
	2×2 MaxPool
	FC-128
	FC-10

Now we get the inequality between $|\boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*|$ and $\left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_{t-1}^*\right|$ when the system is attacked in round $t-1$.

Since we introduced server-side defense, we rewrite Equation 8

$$\boldsymbol{\theta}_t - \boldsymbol{\theta}_t^* = \frac{N}{|S_t^*|}\left[\sum_{k\in S_t^*} p^k \prod_{i=0}^{I-1}\left(I - \eta_t H_{t,i}^k\right)\right]\left(\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_{t-1}^*\right) \tag{17}$$

Then we get the following relationship between $|\boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*|$ and $\left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_{t-1}^*\right|$ when server-side defense filters out all malicious updates in round $t-1$.

$$|\boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*| \leq \frac{N}{|S_t^*|}\left|\sum_{k\in S_t^*} p^k \prod_{i=0}^{I-1}\left(I - \eta_t H_{t,i}^k\right)\right|\left|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_{t-1}^*\right| \tag{18}$$

Finally, we can use Equation 16 and 18 to prove the Theorem by induction hypothesis

$$R(\rho) = |\boldsymbol{\theta}_T - \boldsymbol{\theta}_T^*| = (1+dc)^{\sum_{\in\Phi_T}\eta_t}\rho\left(\left|\prod_{t\in\Gamma_T}\left[\frac{N}{|S_t^*|}\sum_{k\in S_t^*} p^k \prod_{i=0}^{I-1}\left(I - \eta_t H_{t,i}^k\right)\right]\right| + |\Phi_T|\sum_{t\in\Phi_T}\eta_t\right) \tag{19}$$

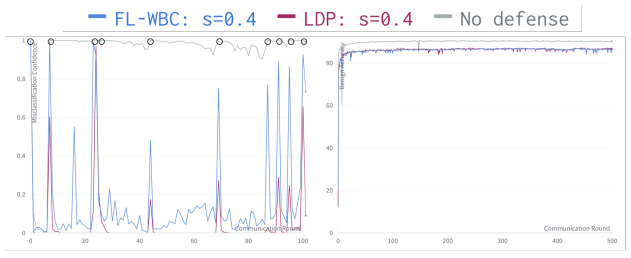
B. Experiments Detail

For all datasets, we choose the learning rate $\eta = 0.01$ and batch size $Ba = 32$ for all clients. The model architectures for two datasets are shown in Table 4.

C. Comparison Between FL-WBC and LDP

The only difference between FL-WBC and LDP (Local Differential Privacy) is that FL-WBC adds noise to only the smaller elements in Hessian Matrix by estimating the matrix, whereas LDP includes noise for all elements. Therefore, LDP can also be used to perturb the null space of the Hessian Matrix. We, therefore, believe that a detailed comparison between the two is necessary.

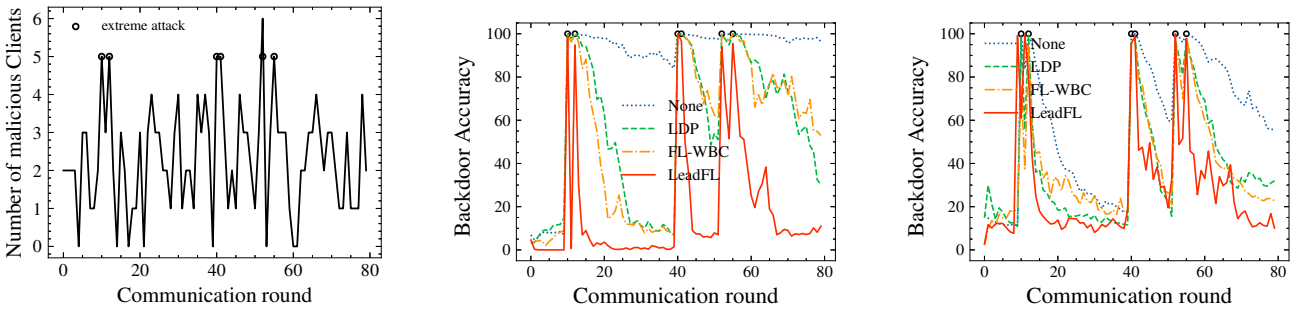
The experiment compares the FL-WBC given std of Laplace noise $s = 0.4$ with LDP $s = 0.4$ on both FashionMNIST and CIFAR10 datasets with IID settings under single-image targeted attack. The threat model is the same as the paper (Sun et al., 2021). The results in Figure 6 show a slight difference between FL-WBC and LDP in all settings. The FashionMNIST-IID dataset shows almost no difference between the two defenses approach. Both FL-WBC and LDP successfully defend the attack and maintain almost the same benign accuracy in the first 100 communication rounds. With the CIFAR10-IID setup, the FL-WBC and LDP successfully defend the attack for the first 80 communication rounds. However, both defenses lead to a loss of model accuracy. The benign accuracy of FL-WBC and LDP have the same distribution, and both results are below 50%. In other words, there is no significant difference between the results of FL-WBC and LDP in this experiment.



(a) FashionMNIST-IID (b) FashionMNIST-IID

(c) CIFAR10-IID (d) CIFAR10-IID

Figure 6: Comparison between FL-WBC and LDP on different datasets. The black circles represent the communication round that malicious clients conduct the attack.



(a) Number of malicious clients per round (b) FashionMNIST-IID (c) CIFAR10-IID

Figure 7: Results of different random client selections. The backdoor attack here is the 9-pixel-pattern backdoor attack. The server-side defense here is Multi-Krum. Black hollow circles indicate that the system is attacked very strongly in that round.

D. Additional Results

D.1. Results of different client selections

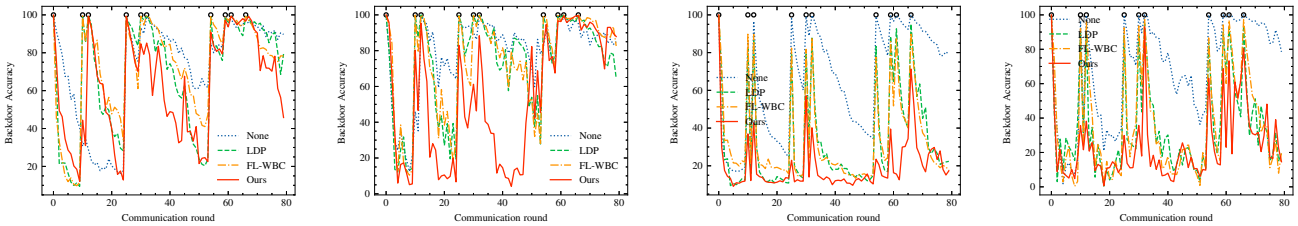
In the main part, only one client selection result is shown. In order to ensure that our results are not an artifact of this one specific client selection, we present results for another selection result shown in Figure 7. We can observe that, the backdoor accuracy is still very high at the round when the extreme attack is conducted. Our defense still exhibits a faster decrease in backdoor accuracy than the other defenses.

D.2. Results of the larger scale setting

In the main part, the number of clients in our system is 100. In this subsection, we present the results of experiments with an increased number of clients, totaling 1000, while maintaining 25% of malicious clients. The dataset is evenly distributed among all clients, and each round involves the selection of 100 clients, with other settings remaining consistent with Table 1 in the main part. As shown in Table 5, our method still achieves the highest MA and lowest BA compared to other client-side defenses in the larger-scale experiments.

Table 5: Comparison of defenses under 9-pixel pattern backdoor attack on IID FashionMNIST dataset. The number of clients is 1000

Distribution	IID							
Server-side Defense	Multi-Krum				Bulyan			
Client-side Defense	None	LDP	FL-WBC	Ours	None	LDP	FL-WBC	Ours
MA	86.7	84.1	83.7	**84.4**	85.2	83.9	84.0	**84.2**
BA Avg	77.2	73.5	67.9	**52.8**	82.8	76.1	72.9	**45.7**
BA Final	71.0	65.1	62.3	**35.7**	79.9	53.2	46.9	**26.9**



(a) FashionMNIST-IID (b) FashionMNIST-NonIID (c) CIFAR10-IID (d) CIFAR10-NonIID

Figure 8: Comparison of backdoor accuracy on FashionMNIST and CIFAR10 with both IID and non-IID settings. The backdoor attack here is single pixel backdoor attack. The server-side defense here is Multi-Krum. Black hollow circles indicate that the system is attacked very strongly in that round.

D.3. Results of CMA and CTMA under 9-pixel backdoor attack

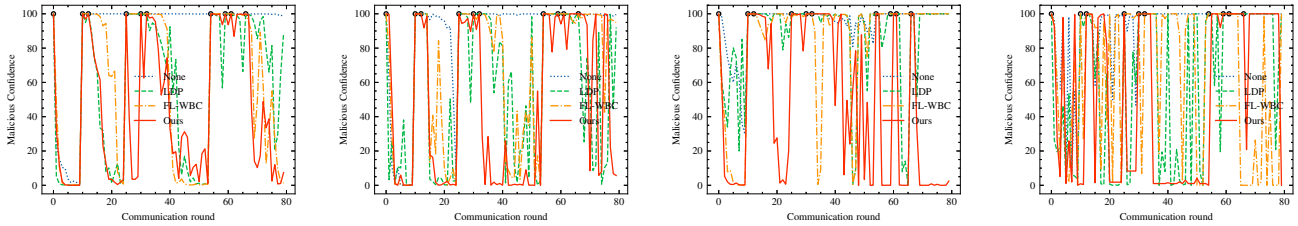
In the main part, we only show the results of MultiKurm and Bulyan server-side defenses. Table 6 and 7 contain the results of CMA and CTMA.

D.4. Results of single-pixel backdoor attack

Table 8, 9 and Figure 8 show the performance of defenses under single-pixel backdoor attacks.

D.5. Results of single image targeted attack

We also measure **Malicious Confidence (MC)**: In (Bhagoji et al., 2019), the authors present a single-image attack where a malicious client inserts exactly one image with the wrong label in their dataset. The malicious confidence is the confidence of the global model in their classification of the malicious image. We consider both average and final confidence. Note that this metric is only relevant for single-image attacks. Table 10, 11 and Figure 9 show the performance of defenses under single image targeted attacks.



(a) FashionMNIST-IID (b) FashionMNIST-NonIID (c) CIFAR10-IID (d) CIFAR10-NonIID

Figure 9: Comparison of backdoor accuracy on FashionMNIST and CIFAR10 with both IID and non-IID settings. The attack here is single image targeted attack. The server-side defense here is Multi-Krum. Black hollow circles indicate that the system is attacked very strongly in that round.

Table 6: Comparison of defenses under 9-pixel pattern backdoor attack on FashionMNIST dataset with both IID and non-IID settings.

Distribution	Server-side Defense	Client-side Defense	MA	BA Avg	BA Final
IID	None	None	89.8	98.4	100.0
		LDP	88.7	90.7	98.8
		FL-WBC	88.1	**90.2**	99.5
		Ours	**89.0**	95.0	**95.7**
	CMA	None	90.0	**93.8**	100.0
		LDP	87.1	95.8	99.3
		FL-WBC	87.2	96.7	**99.2**
		Ours	**87.6**	96.7	99.6
	CTMA	None	89.7	96.6	100.0
		LDP	88.4	97.8	99.9
		FL-WBC	**90.0**	98.9	99.6
		Ours	87.5	**91.9**	**96.8**
	Multi-Krum	None	89.3	82.6	93.2
		LDP	87.0	76.0	79.6
		FL-WBC	87.2	77.5	80.6
		Ours	**87.9**	**32.9**	**0.0**
	Bulyan	None	89.2	78.8	90.6
		LDP	86.0	74.1	62.2
		FL-WBC	86.0	70.6	86.5
		Ours	**86.3**	**21.6**	**0.3**
Non-IID	None	None	87.6	99.7	100.0
		LDP	82.2	94.9	99.4
		FL-WBC	84.0	**94.7**	**96.6**
		Ours	**84.9**	97.5	97.1
	CMA	None	85.6	98.9	100.0
		LDP	78.5	**97.6**	99.9
		FL-WBC	78.5	**97.6**	99.9
		Ours	**80.2**	98.2	**92.5**
	CTMA	None	85.3	99.2	100.0
		LDP	81.9	99.6	99.9
		FL-WBC	**82.4**	99.3	99.9
		Ours	80.8	**95.1**	**67.3**
	Multi-Krum	None	85.6	88.7	93.3
		LDP	76.7	80.4	86.7
		FL-WBC	77.2	74.0	70.3
		Ours	**79.1**	**39.5**	**1.2**
	Bulyan	None	77.4	92.5	88.6
		LDP	73.4	71.9	94.7
		FL-WBC	71.7	73.7	69.0
		Ours	**74.0**	**32.3**	**2.0**

Table 7: Comparison of defenses under 9-pixel pattern backdoor attack on CIFAR10 dataset with both IID and non-IID settings.

Distribution	Server-side Defense	Client-side Defense	MA	BA Avg	BA Final
IID	None	None	71.3	93.5	99.6
		LDP	55.1	77.0	79.7
		FL-WBC	56.2	77.1	89.5
		Ours	**60.4**	**70.2**	**58.7**
	CMA	None	74.1	93.3	99.2
		LDP	12.8	67.6	70.6
		FL-WBC	12.6	**62.3**	**69.9**
		Ours	**64.6**	82.3	79.0
	CTMA	None	75.8	95.2	99.7
		LDP	56.6	95.1	96.4
		FL-WBC	56.4	94.8	97.1
		Ours	**61.3**	**79.4**	**49.4**
	Multi-Krum	None	76.3	77.5	80.5
		LDP	48.0	53.1	43.8
		FL-WBC	43.3	56.9	40.5
		Ours	**56.9**	**35.6**	**25.6**
	Bulyan	None	76.2	79.1	87.0
		LDP	41.5	46.7	23.4
		FL-WBC	42.2	51.3	35.5
		Ours	**54.8**	**43.9**	**21.4**
Non-IID	None	None	73.7	97.0	100.0
		LDP	50.8	86.3	92.5
		FL-WBC	52.6	83.2	89.6
		Ours	**60.5**	**76.3**	**67.7**
	CMA	None	69.6	97.3	99.9
		LDP	13.5	**58.4**	**69.2**
		FL-WBC	13.2	63.9	69.7
		Ours	**60.3**	87.5	90.1
	CTMA	None	73.0	98.3	100.0
		LDP	54.2	97.5	99.1
		FL-WBC	51.1	97.2	99.8
		Ours	**56.9**	**90.1**	**84.5**
	Multi-Krum	None	70.7	85.8	96.2
		LDP	43.2	55.4	52.4
		FL-WBC	42.9	54.4	35.4
		Ours	**55.3**	**45.2**	**34.4**
	Bulyan	None	61.7	87.5	95.2
		LDP	36.7	48.8	29.8
		FL-WBC	36.2	48.1	47.7
		Ours	**51.4**	**46.8**	**27.3**

Table 8: Comparison of defenses under single pixel backdoor attack on FashionMNIST dataset with both IID and non-IID settings.

Distribution	Server-side Defense	Client-side Defense	MA	BA Avg	BA Final
IID	None	None	90.1	96.2	99.7
		LDP	88.0	88.0	95.1
		FL-WBC	87.8	85.2	97.7
		Ours	**88.1**	91.2	95.7
	CMA	None	89.9	89.1	99.8
		LDP	87.0	96.0	97.9
		FL-WBC	87.0	96.7	99.3
		Ours	**87.7**	91.2	98.0
	CTMA	None	89.8	92.0	99.7
		LDP	88.2	96.3	99.6
		FL-WBC	**87.6**	96.0	99.2
		Ours	**88.2**	84.4	92.1
	Multi-Krum	None	89.4	28.1	39.7
		LDP	87.0	70.7	90.7
		FL-WBC	86.9	71.4	76.8
		Ours	**87.5**	**70.3**	**43.4**
	Bulyan	None	89.1	72.2	89.7
		LDP	85.9	68.8	79.4
		FL-WBC	85.3	72.4	**78.3**
		Ours	**86.7**	**67.6**	85.8
Non-IID	None	None	88.0	98.8	99.9
		LDP	**83.4**	90.9	97.8
		FL-WBC	82.2	91.0	98.1
		Ours	**82.8**	93.9	97.0
	CMA	None	85.7	95.4	99.9
		LDP	76.4	96.3	99.2
		FL-WBC	79.3	96.6	99.6
		Ours	**79.5**	93.4	93.7
	CTMA	None	85.5	97.4	99.9
		LDP	81.0	98.2	99.7
		FL-WBC	**81.7**	98.6	99.9
		Ours	**81.5**	94.5	92.8
	Multi-Krum	None	86.5	79.2	85.1
		LDP	80.5	72.7	**64.5**
		FL-WBC	78.2	73.8	82.6
		Ours	**81.7**	**54.4**	87.9
	Bulyan	None	85.2	83.9	85.9
		LDP	73.7	66.9	85.2
		FL-WBC	70.1	72.5	71.4
		Ours	**75.0**	**62.6**	**46.2**

Table 9: Comparison of defenses under single pixel backdoor attack on CIFAR10 dataset with both IID and non-IID settings.

Distribution	Server-side Defense	Client-side Defense	MA	BA Avg	BA Final
IID	None	None	71.3	93.5	99.6
		LDP	55.1	77.0	79.7
		FL-WBC	56.2	77.1	89.5
		Ours	**62.1**	70.2	58.7
	CMA	None	74.1	93.3	99.2
		LDP	12.8	67.6	70.6
		FL-WBC	12.6	62.3	69.9
		Ours	**64.6**	82.3	79.0
	CTMA	None	75.8	95.2	99.7
		LDP	56.6	95.1	96.4
		FL-WBC	56.4	94.8	97.1
		Ours	**61.3**	79.4	49.4
	Multi-Krum	None	76.3	77.5	80.5
		LDP	48.0	53.1	43.8
		FL-WBC	43.3	56.9	40.5
		Ours	**56.9**	**35.6**	**25.6**
	Bulyan	None	76.2	79.1	87.0
		LDP	41.5	46.7	23.4
		FL-WBC	42.2	51.3	35.5
		Ours	**55.8**	**43.9**	**26.4**
Non-IID	None	None	74.5	92.3	99.8
		LDP	52.7	73.0	81.2
		FL-WBC	51.4	70.2	85.0
		Ours	**62.3**	62.8	55.5
	CMA	None	67.8	89.5	98.4
		LDP	13.1	67.5	69.5
		FL-WBC	13.7	66.3	68.5
		Ours	**59.2**	82.7	74.1
	CTMA	None	73.6	92.6	99.8
		LDP	54.8	93.6	94.2
		FL-WBC	**48.7**	94.0	95.2
		Ours	**58.1**	77.7	78.7
	Multi-Krum	None	71.0	68.7	78.5
		LDP	40.7	34.8	**19.3**
		FL-WBC	39.6	35.9	20.0
		Ours	**51.5**	**26.6**	31.9
	Bulyan	None	62.7	69.8	78.3
		LDP	33.6	25.4	16.6
		FL-WBC	37.5	22.1	15.5
		Ours	**49.1**	**17.6**	**10.3**

Table 10: Comparison of defenses under single image targeted attack on FashionMNIST dataset with both IID and non-IID settings.

Distribution	Server-side Defense	Client-side Defense	MA	MC Avg	MC Final
IID	None	None	89.9	94.6	98.9
		LDP	87.4	79.7	95.8
		FL-WBC	**88.3**	82.7	75.5
		Ours	88.2	82.0	95.8
	CMA	None	89.3	98.1	99.9
		LDP	86.1	88.6	77.9
		FL-WBC	86.7	86.1	83.6
		Ours	**87.6**	93.6	91.0
	CTMA	None	89.0	94.2	99.4
		LDP	87.3	95.2	93.9
		FL-WBC	87.4	95.2	98.8
		Ours	**87.5**	89.4	99.9
	Multi-Krum	None	89.1	89.5	99.5
		LDP	86.5	57.9	88.7
		FL-WBC	86.4	58.1	1.3
		Ours	**86.8**	**44.9**	**7.4**
	Bulyan	None	89.1	92.0	100.0
		LDP	85.0	34.0	**0.1**
		FL-WBC	85.5	86.2	84.1
		Ours	**86.7**	**33.9**	14.0
Non-IID	None	None	87.7	97.2	99.6
		LDP	83.3	86.1	99.2
		FL-WBC	83.7	92.9	95.7
		Ours	**83.9**	78.9	96.7
	CMA	None	87.1	94.6	99.9
		LDP	81.2	87.5	92.6
		FL-WBC	79.4	81.8	94.5
		Ours	**82.2**	81.5	83.1
	CTMA	None	86.7	95.4	99.8
		LDP	84.8	93.3	87.1
		FL-WBC	**84.8**	92.9	93.7
		Ours	84.4	78.4	84.0
	Multi-Krum	None	86.3	87.6	99.6
		LDP	81.6	55.2	92.1
		FL-WBC	80.6	59.4	77.0
		Ours	**84.5**	**46.1**	**5.7**
	Bulyan	None	76.8	86.8	99.9
		LDP	71.8	54.2	77.1
		FL-WBC	73.9	83.5	99.8
		Ours	**77.2**	**39.8**	**0.0**

Table 11: Comparison of defenses under single image targeted attack on FashionMNIST dataset with both IID and non-IID settings.

Distribution	Server-side Defense	Client-side Defense	MA	MC Avg	MC Final
IID	None	None	70.5	98.9	99.9
		LDP	51.1	97.2	97.2
		FL-WBC	50.8	95.5	100.0
		Ours	**60.4**	89.1	100.0
	CMA	None	71.0	99.8	99.9
		LDP	13.9	96.2	100.0
		FL-WBC	13.9	99.6	100.0
		Ours	**60.6**	97.0	100.0
	CTMA	None	68.4	99.8	99.9
		LDP	47.0	98.9	100.0
		FL-WBC	47.4	98.4	100.0
		Ours	**60.8**	96.1	99.9
	Multi-Krum	None	75.4	94.7	100.0
		LDP	46.3	86.6	99.9
		FL-WBC	46.7	82.5	99.9
		Ours	**56.6**	**47.7**	**2.4**
	Bulyan	None	73.6	84.1	93.8
		LDP	41.6	49.0	0.0
		FL-WBC	41.9	6.5	0.0
		Ours	**53.6**	**1.7**	**0.0**
Non-IID	None	None	72.6	99.0	100.0
		LDP	48.6	83.3	100.0
		FL-WBC	50.6	86.0	97.7
		Ours	**60.3**	90.2	100.0
	CMA	None	65.9	99.6	99.6
		LDP	13.0	98.0	100.0
		FL-WBC	15.1	96.0	100.0
		Ours	**61.5**	98.4	100.0
	CTMA	None	70.5	98.9	99.9
		LDP	47.3	97.3	99.1
		FL-WBC	47.8	96.9	100.0
		Ours	**55.0**	95.5	100.0
	Multi-Krum	None	70.5	92.4	100.0
		LDP	43.8	60.6	100.0
		FL-WBC	43.3	65.0	100.0
		Ours	**51.7**	**55.3**	**0.0**
	Bulyan	None	68.1	84.7	100.0
		LDP	39.8	63.9	100.0
		FL-WBC	38.4	62.9	100.0
		Ours	**49.1**	**25.1**	**3.5**