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

Claus Guthmann, Ramazan Tan

Abstract—**Existing defense mechanisms in federated learning often assume an approximately constant proportion of malicious clients throughout training. However, when only a subset of clients participates in each communication round, the fraction of malicious clients can vary substantially across rounds. Over a sufficient number of iterations, this variability makes it likely that some rounds will contain a majority of malicious participants. Under this premise, the paper *LeadFL: Client Self-Defense against Model Poisoning in Federated Learning* [1] demonstrates that poisoning attacks occurring in such rounds can have long-lasting effects on the global model. To address this issue, the authors propose a client-side defense mechanism that significantly improves the model's ability to recover from these transient but severe poisoning events.**

I. Introduction

Federated learning enables the collaborative training of machine learning models across distributed clients while keeping local data private. However, this decentralized setting also exposes the training process to model poisoning attacks, where malicious clients manipulate their updates to degrade the global model. Many existing defense mechanisms implicitly assume that the proportion of malicious clients remains approximately constant throughout training. In practice, federated learning systems typically select only a subset of clients to participate in each communication round, causing the fraction of malicious clients to fluctuate over time. As a result, it becomes increasingly likely that some rounds contain an unusually high—or even majority—share of malicious participants, especially over long training horizons.

The paper *LeadFL: Client Self-Defense against Model Poisoning in Federated Learning* [1] addresses this often-overlooked scenario. It shows that poisoning attacks occurring during such adversarial spikes can have long-lasting effects on the global model, even when the system later returns to benign conditions. To mitigate this issue, LeadFL introduces a client-side defense mechanism designed to limit the impact of these transient but severe attacks and to accelerate recovery in subsequent rounds.

In this report, we begin by presenting the preliminary concepts and results required to understand the LeadFL approach. We then provide an overview of the method itself and discuss its potential advantages in federated learning environments subject to spiking adversarial participation, followed by an examination of its limitations. To further assess its effectiveness, we implemented the LeadFL algorithm and extended the original evaluation to cover previously unaddressed cases. Finally, we present the results of our own experiments, which corroborate that LeadFL significantly improves robustness against the attack patterns considered in the paper.

II. Background

A central idea is to reduce the lasting influence of a burstial attack. One way to formally characterize this lingering effect was introduced in *FL-WBC: Enhancing Robustness against Model Poisoning Attacks in Federated Learning from a Client Perspective* [2] through the notion of the **Attack Effect on Parameters**. For a given communication round t, let W_t denote the global model parameters resulting from potentially malicious client updates, and let W_t^M denote the parameters that would have been obtained in the absence of malicious behavior. The attack effect is then defined as

$$\delta_t := W_t - W_t^M. \tag{1}$$

The authors further derive a recursive approximation of this quantity,

$$\delta_t \approx \hat{\delta}_t := \frac{1}{|C_t|}\left[\sum_{c\in C_t}\prod_{i=0}^{I-1}\left(I - \eta_t H_{t,i}^c\right)\right]\hat{\delta}_{t-1},$$

where C_t denotes the set of clients selected in round t (assumed to be equally weighted), η_t is the learning rate, I is the number of local training steps, and $H_{t,i}^c$ represents the Hessian matrix of the local loss function.

Empirically, the Hessian matrices $H_{t,i}^c$ are often highly sparse, which causes the aggregated transformation

$$\frac{1}{|C_t|}\left[\sum_{c\in C_t}\prod_{i=0}^{I-1}\left(I - \eta_t H_{t,i}^c\right)\right] \tag{2}$$

to remain close to the identity matrix. As a consequence, the attack effect δ_t decays only slowly over time, leading to the observed persistence of poisoning attacks.

The FL-WBC [2] client-side defense seeks to counteract this by injecting random noise into local updates, thereby accelerating the decay of the attack effect. Building on this insight, LeadFL [1] proposes a more targeted approach: instead of relying on random perturbations, it introduces a specifically designed regularization term.

Directly computing the full Hessian matrix $H_{t,i}^c$ for neural networks is, however, computationally prohibitive. To make the approach tractable in a federated learning setting, LeadFL relies on an efficient approximation of the Hessian inspired by the *Optimal Brain Damage* framework [3].

Specifically, the method restricts attention to the diagonal elements of the Hessian, thereby ignoring inter-parameter dependencies. While this simplification sacrifices second-order interactions, it significantly reduces computational complexity and has been shown to provide useful curvature information in large-scale models. The diagonal entries capture the sensitivity of the gradient with respect to individual parameters and can be approximated via finite differences of successive gradients.

Formally, the Hessian matrix is approximated as

$$H^c_{t,i} \approx \tilde{H}^c_{t,i} := \operatorname{diag}\big(\nabla L(W^c_{t,i+1}) - \nabla L(W^c_{t,i})\big), \quad (3)$$

where $W^c_{t,i}$ denotes the local model parameters of client c at global round t and local step i, and $L(\cdot)$ is the local loss function.

III. LeadFL

The core idea of LeadFL [1] is to reduce the persistence of poisoning effects by locally minimizing the term $I - H^c_{t,i}$ on each client, thereby accelerating the decay of the attack effect described in the previous section. To achieve this, LeadFL augments the standard local optimization objective with an additional regularization term whose gradient is incorporated into each update step.

Concretely, each local update is computed in two stages. First, a standard gradient descent step is performed,

$$\tilde{W}^c_{t,i+1} := W^c_{t,i} - \eta_t \nabla L(W^c_{t,i}), \quad (4)$$

yielding intermediate model parameters $\tilde{W}^c_{t,i+1}$. In a second step, the regularization update is applied,

$$W^c_{t,i+1} := \tilde{W}^c_{t,i+1} - \eta_t \alpha \operatorname{clip}\Big(\nabla\big(I - \eta_t \tilde{H}^c_{t,i}\big), q\Big), \quad (5)$$

where α controls the strength of the regularization and *clip* limits the absolute value of each entry to the clipping threshold q.

This two-stage formulation allows the intermediate parameters $\tilde{W}^c_{t,i+1}$ to be reused in constructing a computationally efficient approximation of the Hessian. However, the derivation of this approximation in the original paper contains errors. We therefore adopt the following formulation, which we believe to be a mathematically consistent approximation closely aligned with the intended method:

$$\tilde{H}^c_{t,i} \approx (W^c_{t,i} - \tilde{W}^c_{t,i+1} + \Delta W^c_{t,i})/\eta_t, \quad (6)$$

where $\Delta W^c_{t,i} := W^c_{t,i} - W^c_{t,i-1}$ denotes the parameter change between successive local steps. It is important to note that this Hessian approximation is only accurate up to the contribution of the regularization term applied in the previous local update.

Combining the previous steps, the local client iteration of LeadFL can be summarized as follows (cf. Algorithm 1 in [1]):

Algorithm 1 *LeadFL* Local Client Iteration

```
1: Compute gradients and update intermediate weights:
   $\tilde{W}^c_{t,i+1} \leftarrow W^c_{t,i} - \eta_t \nabla L(W^c_{t,i})$
2: Estimate the Hessian matrix:
   $\tilde{H}^c_{t,i} \leftarrow (W^c_{t,i} - \tilde{W}^c_{t,i+1} + \Delta W^c_{t,i})/\eta_t$
3: Compute the regularization term:
   $R^c_{t,i} \leftarrow \operatorname{clip}\Big(\nabla\big(I - \eta_t \tilde{H}^c_{t,i}\big), q\Big)$
4: Update the local weights with the regularization term:
   $W^c_{t,i+1} \leftarrow \tilde{W}^c_{t,i+1} - \eta_t \alpha R^c_{t,i}$
```

The authors of LeadFL subsequently implement and evaluate the proposed defense under a range of experimental settings. Their primary setup consists of 100 clients, of which 25 are malicious. Training is performed over 80 global rounds, with each client executing 10 local training steps per round. In each global round, 10 clients are selected uniformly at random; this client selection is kept consistent across different experimental runs to ensure comparability (additional results for alternative client selection strategies are reported in the appendix of [1]).

The evaluation considers several server-side aggregation defenses, namely SparseFed, Multi-Krum, and Bulyan. For the LeadFL client-side defense, the clipping threshold is fixed at $q = 0.2$. The regularization strength α is varied across datasets, with $\alpha = 0.4$ for FashionMNIST and $\alpha = 0.25$ for CIFAR10. Due to the randomized client selection process, the number of malicious participants fluctuates across rounds, leading to significant variability in backdoor success rates. To account for this effect, the authors report both the final backdoor accuracy and the average backdoor accuracy over the entire training process.

TABLE I
COMPARISON OF DEFENSES UNDER 9-PIXEL PATTERN BACKDOOR ATTACK ON IID FASHIONMNIST DATASET. (CF. TABLE 1 [1])

Server-side Defense	Multi-Krum			
Client-side Defense	None	LDP	FL-WBC	LeadFL
Maintask Accuracy	89.3	87	87.2	87.9
Backdoor Accuracy Avg.	82.6	76.0	77.5	32.9
Backdoor Accuracy Final	93.2	79.6	80.6	0.0

TABLE II
COMPARISON OF DEFENSES UNDER 9-PIXEL PATTERN BACKDOOR ATTACK ON IID CIFAR10 DATASET. (CF. TABLE 2 [1])

Server-side Defense	Multi-Krum			
Client-side Defense	None	LDP	FL-WBC	LeadFL
Maintask Accuracy	76.3	48.0	43.3	56.9
Backdoor Accuracy Avg.	77.5	53.1	56.9	35.6
Backdoor Accuracy Final	80.5	43.8	40.5	25.6

Tables I and II present a selection of results for the IID FashionMNIST and CIFAR10 datasets, respectively, under a 9-pixel pattern backdoor attack using the Multi-Krum server-side defense.

For the FashionMNIST dataset (Table I), LeadFL consistently reduces the final backdoor accuracy to nearly zero while maintaining the main task accuracy within a 10% reduction compared to the undefended baseline. In contrast, the alternative client-side defenses, LDP and FL-WBC, achieve only moderate reductions in backdoor accuracy.

For the more complex CIFAR10 dataset (Table II), the impact of LeadFL on the main task accuracy is more pronounced, reducing it to approximately 50–60%. Nevertheless, LeadFL still substantially lowers the final backdoor accuracy to around 25%, outperforming the other client-side defenses in terms of robustness.

Across both datasets and all evaluated server-side defenses, the absence of a client-side defense results in consistently high average backdoor accuracies—exceeding 75%—highlighting the severity of bursty poisoning attacks and the necessity of additional client-side protection mechanisms.

Fig. 1. 9-pixel back door attack on FashionMNIST when the attack is periodic(cf. Figure 3 [1])

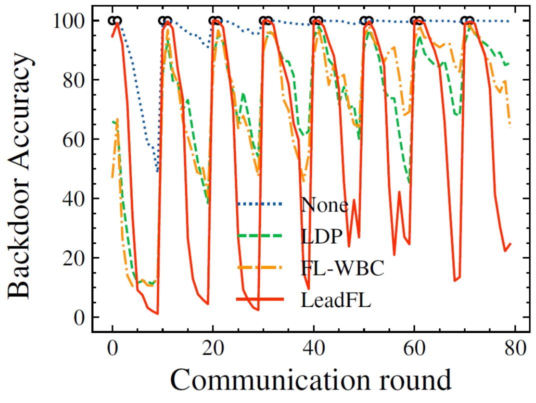
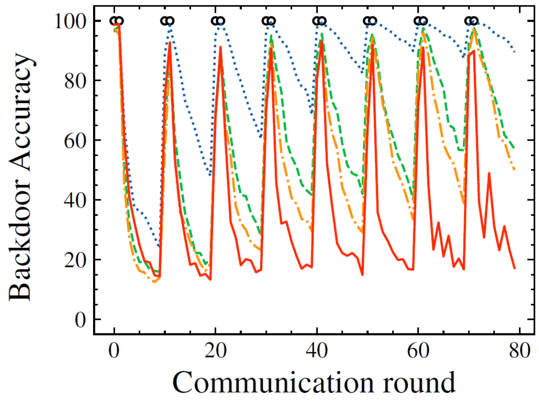


Fig. 2. 9-pixel back door attack on CIFAR10 when the attack is periodic (cf. Figure 3 [1])



The authors further investigate the behavior of LeadFL under explicitly bursty attack patterns. In this setting, client selection is no longer random. Instead, training proceeds in cycles of ten rounds, where the first two rounds contain six malicious clients out of ten selected participants, while the remaining eight rounds involve exclusively benign clients.

The results of this experiment are illustrated in Figure 1 for the FashionMNIST dataset and in Figure 2 for the CIFAR10 dataset. While LeadFL exhibits limited impact during the rounds dominated by malicious clients, its effect becomes apparent in the subsequent benign rounds. In particular, the recovery from each attack spike is significantly faster, and the achieved backdoor accuracy stabilizes at a lower level compared to the undefended baseline.

In summary, LeadFL introduces a novel client-side defense mechanism that is particularly effective at mitigating the long-lasting effects of spikes in malicious client participation. Although the method incurs a measurable reduction in main task accuracy, it demonstrates that client-side defenses can play a crucial role in complementing existing server-side aggregation strategies. These findings motivate a closer examination of LeadFL's practical limitations and empirical behavior, which we address in the remainder of this report through an extended experimental evaluation.

IV. Limitations and Extended Evaluation

Before presenting our own experimental evaluation, we first discuss several limitations and ambiguities encountered in the original LeadFL paper. These issues are relevant both for the correct interpretation of the reported results and for the reproducibility of the proposed method.

First, while the regularization strength α plays a crucial role in balancing robustness against performance degradation, the paper does not provide a principled justification or systematic tuning strategy for the selected values. Instead, α is varied empirically across experiments without clear guidance, limiting the interpretability and generalizability of the reported outcomes.

Fig. 3. 9-pixel back door attack on FashionMNIST when the malicious client selection is periodic

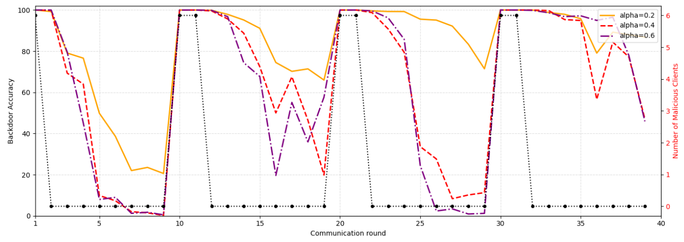
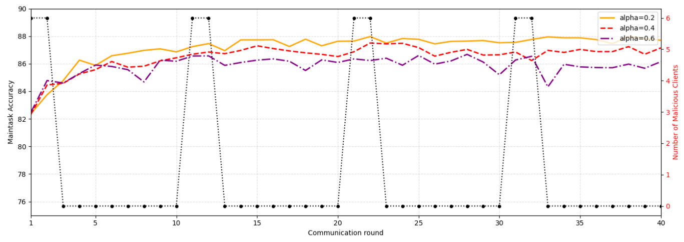


Fig. 4. 9-pixel back door attack on FashionMNIST when the malicious client selection is periodic



To investigate the sensitivity of LeadFL to the choice of α, we conducted additional experiments using the authors' original code with modified regularization strengths. Due to computational constraints, these experiments were limited to 40 global rounds. Figures 3 and 4 show the resulting backdoor accuracy and main task accuracy, respectively. All of the evaluations in this section use the bulyan server-side defense.

As shown in Figure 3, reducing the regularization strength to $\alpha = 0.2$ significantly weakens the defense: the decline in backdoor accuracy is slower and can not reach the same lower bound compared to the default setting of $\alpha = 0.4$. Increasing the regularization strength to $\alpha = 0.6$ yields a slight improvement on average, but the effect is not consistent across all rounds. Regarding the main task performance, illustrated in Figure 4, $\alpha = 0.2$ provides no observable benefit, while $\alpha = 0.6$ introduces occasional drops in accuracy.

Finally, as discussed in the previous sections, the Hessian approximation used in LeadFL appears to contain inconsistencies. As a result, any practical implementation, including our own, necessarily relies on an approximation that deviates from the formulation presented in the paper.

Fig. 5. 9-pixel back door attack on FashionMNIST when the malicious client selection is periodic
TODO: Replace with proper client selection

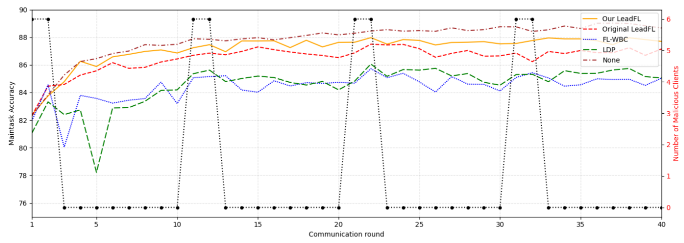
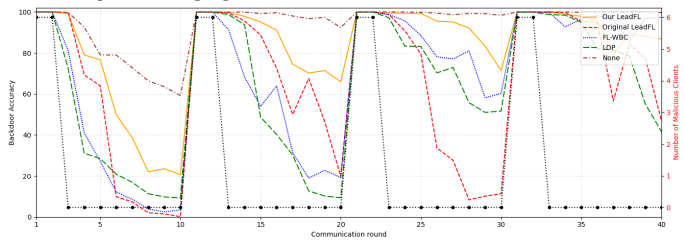
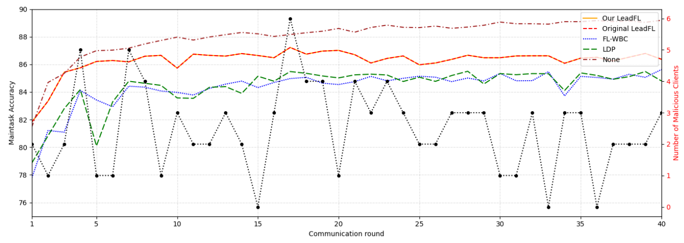


Fig. 6. 9-pixel back door attack on FashionMNIST when tthe malicious client selection is periodic
TODO: Replace with proper client selection



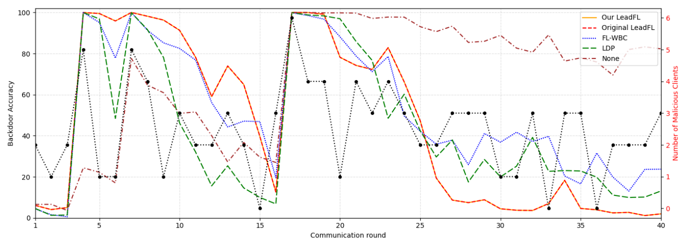
To evaluate this, we reproduced the periodic attack setup shown in Figure 1 and augmented it with results obtained from our own implementation. The corresponding main task accuracy and backdoor accuracy are shown in Figures 5 and 6, respectively. Although our implementation performs slightly worse than the original results reported by the authors, the central qualitative behavior of LeadFL remains intact. In particular, the defense induces only a moderate reduction in main task accuracy while substantially improving the recovery rate following bursty poisoning attacks.

Fig. 7. 9-pixel back door attack on FashionMNIST when the malicious client selection is random



We also repeated the same evaluation using random client selection instead of the periodic pattern. Interestingly, the main-task accuracy, shown in Figure 7, remains largely consistent with the periodic case. In contrast, the backdoor accuracy exhibits more pronounced differences. While LeadFL still provides the strongest defense toward the end of training, it is occasionally outperformed by other methods—including the baseline with no client-side defense—during the initial

Fig. 8. 9-pixel back door attack on FashionMNIST when the malicious client selection is random



rounds.

In summary, our evaluation confirms that LeadFL is effective at its core: it accelerates recovery from bursty poisoning attacks and lowers the long-term backdoor accuracy, though at a cost to main-task performance. Notably, even with this trade-off, LeadFL maintains higher main-task accuracy compared to alternative client-side defenses. The periodic bursty attack pattern considered in this report represents an optimal scenario for LeadFL, whereas random client selection reduces its overall impact and introduces greater variability in backdoor mitigation. These findings suggest that while LeadFL shows promise, further evaluation is necessary before deployment in real-world federated learning systems. In particular, large-scale experiments and a more systematic investigation of the regularization parameter α in complex models would be essential.

REFERENCES

[1] C. Zhu, S. Roos, and L. Chen. LeadFL: Client Self-Defense against Model Poisoning in Federated Learning. *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, pp. 43158–43180, Jul. 2023.

[2] J. Sun, A. Li, L. DiValentin, A. Hassanzadeh, Y. Chen and H. Li. FL-WBC: Enhancing Robustness against Model Poisoning Attacks in Federated Learning from a Client Perspective *Advances in Neural Information Processing Systems*, vol. 34, pp. 12613–12624, Dec. 2021.

[3] Y. LeCun, J. Denker and S. Solla. Optimal Brain Damage *Advances in Neural Information Processing Systems*, vol. 2, pp. 598-605, Nov. 1989.

V. Summary: Model Poisoning Attacks to Federated Learning via Multi-Round Consistency

Many existing attacks on federated learning rely on assumptions that are often unrealistic in practice, such as knowledge of other clients' updates or awareness of the aggregation rule used by the server. The paper *Model Poisoning Attacks to Federated Learning via Multi-Round Consistency* [2] proposes an attack strategy that relaxes these assumptions and operates effectively without direct knowledge of other clients or the aggregation mechanism.

The principal idea behind the attack is to overcome the natural cancellation of updates caused by random noise added by benign clients. To achieve this, the malicious clients first select a consistent attack direction s. They then estimate the honest update by removing their own previous malicious contribution from the global model update.

$$\tilde{h} = g^{t-1} - \frac{||g^{t-1}||}{||k^{t-1} \odot s||}(k^{t-1} \odot s) \tag{7}$$

where g^{t-1} is the last global model update and $k^{t-1} \odot s$ the last model updates on the fake clients. The factor $\frac{||g^{t-1}||}{||k^{t-1} \odot s||}$ normalizes the malicious update to have the same magnitude as the as the global one. This estimate is normalized to form a template of a benign update.

$$v^t = \frac{|\tilde{h}|}{||\tilde{h}||_2} \tag{8}$$

$$\lambda_t = c_t||g^{t-1}||_2 \tag{9}$$

A scaling factor λ_t is determined relative to the norm of the global update, with an additional check: if the global model does not drift in the intended attack direction, it is assumed that the malicious updates are being filtered, and the scaling factor is reduced accordingly: $c_t = \begin{cases} c_{t-1} & \text{Alignment } > 50\% \\ \beta c_{t-1} & \text{else} \end{cases}$ with $\beta < 1$. The final malicious update g_i^t is computed as the product of the normalized template, the scaling factor, and the attack direction.

$$g_i^t = (v^t \cdot \lambda_t) \odot s \tag{10}$$

The presenting group [1] conducted their own evaluation of this attack. They observed that the attack requires several rounds to become effective. Nevertheless, with at least 9% of selected clients acting maliciously, the attack successfully degrades the global model's accuracy.

The students further investigated the performance of a novel server-side defense, PRoDIGY, which was not included in the original paper. PRoDIGY proved extremely effective against this attack because the malicious updates are highly similar, making them easily detectable and filtered by the dissimilarity-based criteria.

To examine potential circumvention strategies, the group adapted the attack to counter PRoDIGY. They tested two techniques proposed in the original paper: noise adaptation, where random perturbations are added to the malicious update, and sign flipping, where each element of the malicious direction vector is flipped with a small probability. For both techniques, different scaling factors and probabilities were evaluated. The noise adaptation strategy was over all more effective than sign flipping. However, in all cases, the main task accuracy remained above 80%.

The group also noted that the original paper assumes a fixed, very small learning rate. They experimented with a learning rate scheduler, which accelerates the increase of main task accuracy and causes the learning rate to decay before the attack can fully take effect. Under this condition, the attack's effectiveness is further reduced.

Building on the work of the group, we believe that an additional direction for future research is to investigate the impact of client selection on attack effectiveness. In the paper, the percentage of malicious clients is fixed after selection, remaining consistent across rounds. Since one of the main observations is that the attack's effect accumulates over time, it would be valuable to study the impact of a fully random client selection process, where the fraction of malicious participants varies from round to round.

Further, if attackers are assumed to coordinate, an alternative strategy would be to partition the dimensions of the malicious direction vector among participants, with each attacker contributing only to their assigned subset of parameters. Against defenses such as PRoDIGY, which rely on detecting similarity across malicious updates, this approach could reduce the artificial correlation among adversarial updates and potentially increase the attack's effectiveness.

References

[1] Y. Xie, M. Fang and N. Z. Gong. Model Poisoning Attacks to Federated Learning via Multi-Round Consistency. *Trustworthy Distributed Learning, TUM*, Jan. 2026.

[2] L. Kunze, F. Durchdewald and N. Schenk. Presentation on: PoisonedFL via Multi-Round Consistency. *Tri*, vpp. 5454-15463, Jun. 2025.