

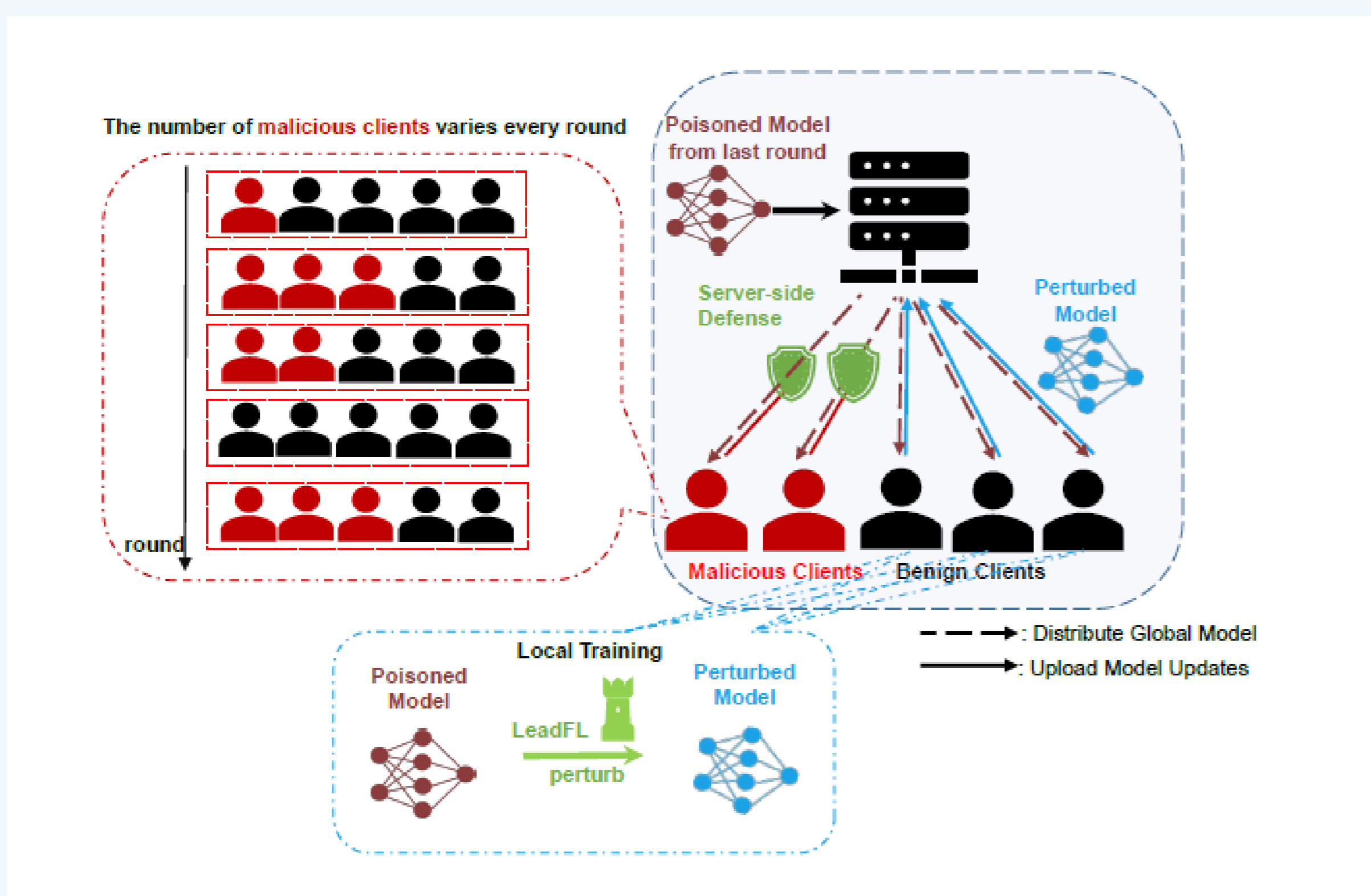
LeadFL: Client Self-Defense against Model Poisoning in Federated Learning

Ramazan Tan, Claus Guthmann

Introduction

Federated Learning (FL) is highly vulnerable to model poisoning attacks, where malicious clients manipulate local data and models without oversight.

- **Bursty adversarial patterns**, characterized by high variance in the number of malicious clients, can effectively bypass current defense mechanisms.
- Although these bursts occur rarely, their effect leaves lingering impacts on the remaining rounds of training. Once the global model is polluted in a single strong attack round, **the attack effect persists for many subsequent rounds** even without additional attacks—existing server-side defenses cannot eliminate this lingering effect.



Bursty adversarial patterns: Number of malicious clients varies between rounds

- **Why Existing Defenses Fail**, Server-side defenses are designed assuming constant and low number of malicious clients.
- **Previous client-side defense (FL-WBC):**
 - Adds random noise to reduce Hessian sparsity
 - **Problem:** Uncalibrated noise degrades model accuracy
 - **Problem:** No theoretical robustness guarantees

LeadFL Solution

Core Idea: Perturb the Hessian matrix to minimize the coefficient $(I - \eta_t H_{t,i}^k)$, reducing the lingering attack effect.

- **Problem:** Computing Hessian Matrix is expensive

Approximation Strategy:

- **Diagonalization:** We approximate H using only its diagonal elements to reduce complexity:

$$H \approx \text{diag}(H)$$

- **Finite Difference:** The diagonal is estimated via the change in gradients between iterations:

$$H \approx \text{diag}(\nabla L(\theta_{t,i+1}^k) - \nabla L(\theta_{t,i}^k))$$

- **Parameter Estimation:** To avoid extra backpropagation, we approximate gradient changes using model weight differences:

$$\tilde{H}_{t,i}^k \approx \frac{\text{diag}(\tilde{\theta}_{t,i+1}^k - \theta_{t,i}^k - \Delta\theta_{t,i}^k)}{\eta_t} \quad (3)$$

LeadFL Solution

This estimation allows LeadFL to regularize the Hessian without incurring significant computational overhead[cite: 130, 134].

Local Weight Update Protocol:

Step 1 - Standard gradient descent:

$$\tilde{\theta}_{t,i+1}^k \leftarrow \theta_{t,i}^k - \eta_t \nabla L(\theta_{t,i}^k) \quad (4)$$

Step 2 - Hessian regularization:

$$\theta_{t,i+1}^k \leftarrow \tilde{\theta}_{t,i+1}^k - \eta_t \alpha \text{clip}[\nabla(I - \eta_t \tilde{H}_{t,i}^k), q] \quad (5)$$

Evaluation

Setup:

- Dataset: FashionMNIST
- 100 clients (25% malicious), 10 selected per round
- Attacks: 9-pixel backdoor, Server side defense: Bulyan
- Compared with: No defense, FL-WBC, LDP, Original LeadFL, Our LeadFL

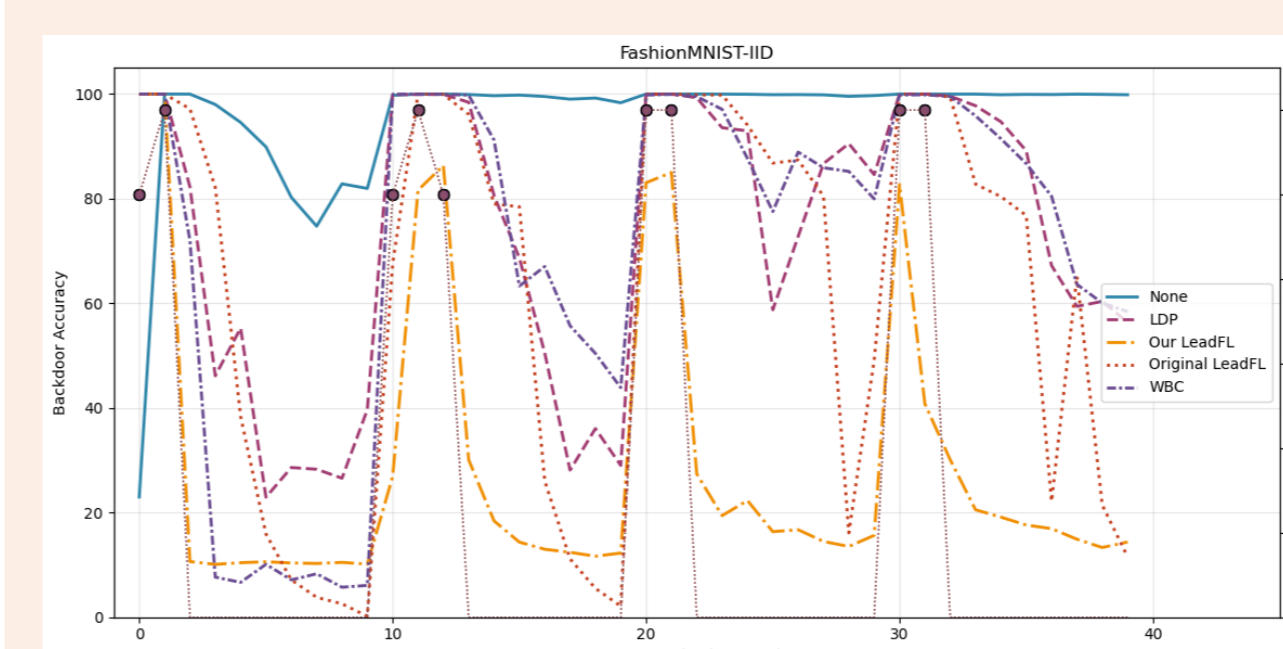


Fig 1: FashionMNIST IID

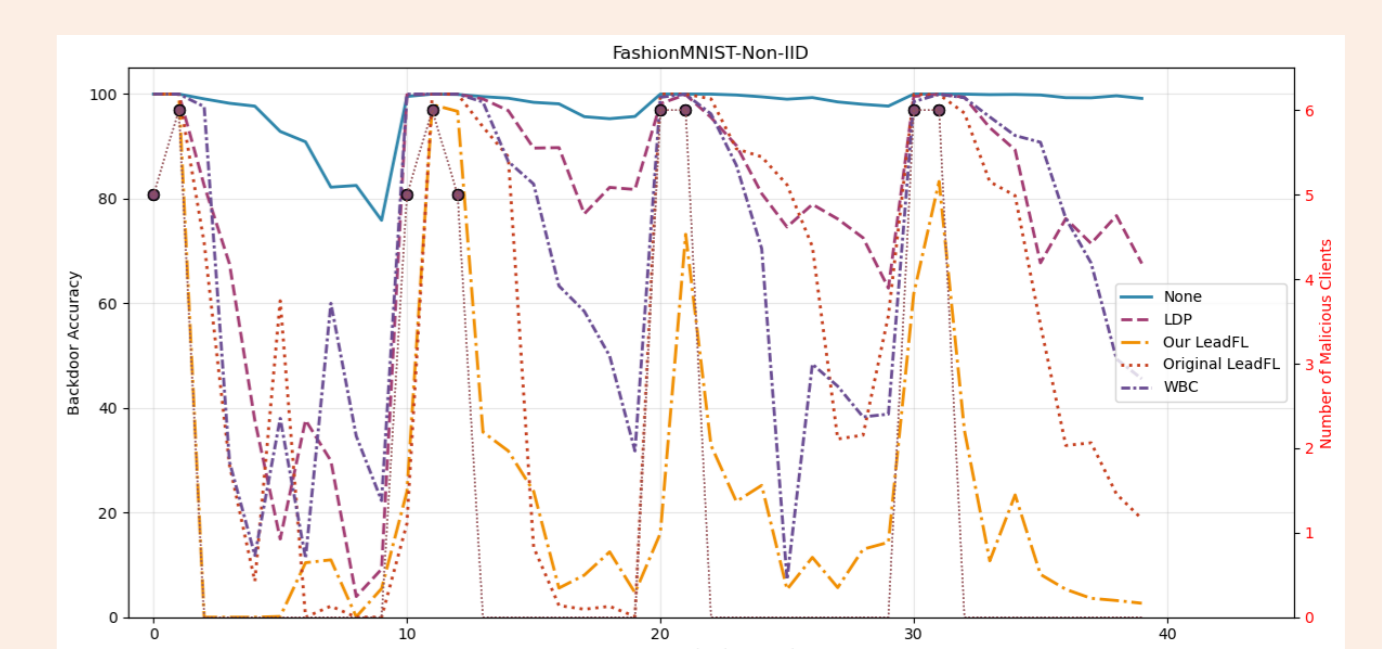


Fig 2: FashionMNIST non-IID

	IID					Non-IID				
	None	LDP	Our LeadFL	Original LeadFL	WBC	None	LDP	Our LeadFL	Original LeadFL	WBC
MA	88.8	85.0	72.9	86.9	85.8	60.6	74.3	38.1	73.0	65.0
BA Avg	95.5	73.1	29.4	62.2	70.8	97.2	76.8	25.6	54.3	68.0
BA Final	99.9	56.4	14.4	11.6	58.4	99.2	67.6	2.7	18.8	45.6

Table 1: Comparison of client-side defenses under 9-pixel pattern backdoor attack

Both LeadFL implementations achieves:

- Up to 65% reduction in backdoor accuracy
- Faster recovery from attack bursts
- Minimal impact on main task accuracy

Conclusion

LeadFL provides a principled client-side defense against model poisoning in federated learning by:

- Optimally perturbing local Hessian matrices to reduce attack persistence
- Offering provable convergence and robustness guarantees
- Outperforming existing defenses against bursty attack patterns

References

- [1] Zhu, C., Roos, S., & Chen, L. Y. (2023). LeadFL: Client Self-Defense against Model Poisoning in Federated Learning. *ICML 2023*.
- [2] Sun, J., et al. (2021). FL-WBC: Enhancing Robustness Against Model Poisoning Attacks in Federated Learning from a Client Perspective. *NeurIPS 2021*.
- [3] Panda, A., et al. (2022). SparseFed: Mitigating Model Poisoning Attacks in Federated Learning with Sparsification. *AISTATS 2022*.

Understanding the Attack Effect

Attack Effect on model Parameter: We quantify the impact using the Attack Effect on Parameter (AEP), denoted as δ_t . This metric represents the change in global model parameters accumulated until the t -th round due to attacks conducted by malicious devices.

$$\delta_t \triangleq \theta_t - \theta_t^M \quad (1)$$

Benign global model Poisoned global model

Propagation of AEP: When malicious clients attack in rounds τ_1 and τ_2 , we estimate the attack effect for the intermediate rounds ($\tau_1 < t < \tau_2$) as follows:

$$\hat{\delta}_t = \frac{N}{K} \left[\sum_{k \in S_t} p^k \prod_{i=0}^{t-1} (I - \eta_i H_{t,i}^k) \right] \hat{\delta}_{t-1} \quad (2)$$

Where **Hessian Matrix:** $H_{t,i}^k \triangleq \nabla^2 L(\theta_{t,i}^k)$

Key Insight: If $\hat{\delta}_{\tau_1}$ resides in the kernel (null space) of $H_{t,i}^k$, then $\hat{\delta}_t = \hat{\delta}_{\tau_1}$, causing the attack effect to persist unchanged.

$$\text{If } H_{t,i}^k \text{ is highly sparse} \implies \hat{\delta}_t \approx \hat{\delta}_{t-1}$$

Why server-side defenses fail: The propagation of attack effects is determined by $H_{t,i}^k$ during *local client training*, which is inaccessible to the central server.