LoMar: A Local Defence Against Poisoning Attack on Federated Learning

SYED NURJA [1] MOHAMMED TAUSEEF AHMED [2] EPPA SRUJAN REDDY [3] SHAIK SHOAIB HANNAN [4]

[1]Assistant Professor, Department of CSE(AI&ML), CMR College, Hyderabad, Telangana, India

[2 3 4] Department of CSE(AI&ML), CMRCET, Hyderabad, Telangana, India

EMAIL: sd.nurja@cmrcet.ac.in[1] tsfahmd01@gmail.com[2] srujanereddy@gmail.com[3] shaik.shoaibhannan05@gmail.com[4]

ABSTRACT

Federated Learning (FL) enables collaborative model training across distributed clients while preserving data privacy. However, its decentralized nature makes it vulnerable to poisoning attacks, where malicious participants manipulate local updates to degrade model performance or introduce backdoors. In this paper, we propose LoMar (Local Malicious Factor), a novel two-phase defense algorithm designed to detect and mitigate poisoning attacks in FL. The first phase applies Kernel Density Estimation (KDE) to analyze the statistical properties of model updates in local neighborhoods, identifying deviations indicative of malicious behavior. The second phase establishes an optimal threshold to distinguish benign and adversarial updates, preventing compromised data from influencing the global model.We evaluate LoMar's performance against label-flipping and model replacement attacks on real-world datasets, including MNIST, CIFAR-10, and Amazon Reviews. Experimental results demonstrate that LoMar improves target label accuracy from 96.0% to 98.8% and overall model accuracy from 90.1% to 97.0%, outperforming existing defense methods such as FG+Krum. Furthermore, LoMar introduces minimal computational overhead, making it suitable for large-scale FL deployments.

Keywords— Federated Learning, Poisoning Attacks, Kernel Density Estimation, Security, Machine Learning.

I. INTRODUCTION

Federated Learning (FL) is a decentralized machine learning technique that allows several clients to work together to build a common global model without sharing raw data. Applications where privacy is a concern, like healthcare, finance, and personalizing mobile devices, benefit greatly from this strategy. FL has been popular in domains such as edge computing, mobile networks, and the Internet of Things because it permits learning to take place locally while maintaining data security. But even with its benefits, FL is still susceptible to a number of security risks, including poisoning attacks that jeopardize the integrity of the model.

Data poisoning and model poisoning are the two main poisoning attack types that FL is vulnerable to. Malicious clients insert tainted or incorrectly labeled data into the training phase during data poisoning attacks, which causes the model to pick up the wrong patterns. However, adversaries that engage in model poisoning attacks alter their local model updates before to sending them to the central aggregator. These changes may result in adversarial behaviors, including backdoor vulnerabilities, or deteriorate the model's overall performance. The privacy-preserving nature of FL makes it impossible to directly view client data, therefore identifying and thwarting such attacks is still very difficult.

A number of defense mechanisms, such as anomaly detection and Byzantine-tolerant aggregation algorithms, have been developed to overcome this problem. These methods, however, frequently fall short against highly skilled adversarial tactics. To improve FL security, we respond by presenting LoMar (Local Malicious Factor), a novel two-phase anomaly detection technique. LoMar analyzes the statistical characteristics of client updates using Kernel Density Estimation (KDE), spotting abnormalities that can point to poisoning. Furthermore, to prevent compromised contributions from impacting the global model, a threshold-based filtering system aids in differentiating between malicious and benign updates. By comparing LoMar's accuracy, resilience, and computing efficiency to those of current defense mechanisms, our study shows that LoMar has the potential to be a reliable and scalable solution for protecting FL systems.

II. LITERATURE REVIEW

Numerous research projects have investigated security measures to defend Federated Learning (FL) from hostile attacks. The main defense tactics are reviewed in this part, along with their advantages and disadvantages, including anomaly detection approaches, Byzantine-tolerant aggregation systems, and cryptographic solutions.

A. Byzantine-Tolerant Aggregation Methods

Krum assumes that benign updates cluster together and chooses one update that is the closest to other updates in terms of Euclidean distance. To get over this safeguard, conspiring rivals can produce poisoned updates that look like harmless ones.

Trimmed Mean eliminates each model parameter's highest and lowest values prior to averaging. Although it works well against severe outliers, it has trouble with covert adversarial approaches in which the tainted updates stay within statistical bounds.

Median Aggregation mitigates strong adversarial influences by aggregating updates using the median rather than the mean. Attackers can, however, alter updates to avoid detection by remaining near the median.

B. Anomaly Detection Techniques

Euclidean Distance-Based Filtering calculates the deviation of each update from the global mean and eliminates outliers. Sophisticated attackers, however, can alter updates to remain inside reasonable bounds.

Angle-Based Anomaly Detection detects adversarial deviations by calculating the cosine similarity between updates and the prior global model. Though susceptible to gradient-level attacks that introduce subtle but consistent variations over several rounds, it is efficient versus large-scale model poisoning.

Kernel Density Estimation (KDE)-Based Anomaly Detection determines the updates' probability distribution and marks those that fall outside of the predicted density regions. This approach is consistent with LoMar, which uses local KDE analysis rather than global statistical analysis to improve detection.

C. Cryptographic Solutions

Secure Multiparty Computation (MPC) & Homomorphic Encryption use cryptographic approaches to ensure secure data aggregation and thwart poisoning attacks. For real-time FL applications, these approaches are unfeasible due to the substantial computing overhead they entail.

D. Poisoning Attacks in Federated Learning

Label-Flipping Attacks alter dataset labels (such as changing "cat" to "dog") in order to deceive the global model.

Backdoor Attacks under certain circumstances, modify model predictions by introducing particular triggers (such as patterns in pictures).

Covert Model Poisoning is a long-term, covert model corruption that is accomplished by making small changes during several training cycles.

E. Review Summary

Krum, Trimmed Mean, and Median Aggregation are examples of byzantine-tolerant aggregation techniques that try to weed out malicious updates but have trouble with covert adversarial attacks. Although they offer more flexible protections, anomaly detection methods like Euclidean Distance-Based Filtering, Angle-Based Detection, and Kernel Density Estimation (KDE) can still be circumvented by skillfully designed attacks. Although cryptographic systems like Homomorphic Encryption and Secure Multiparty Computation (MPC) provide robust security assurances, their high computing costs render them unsuitable for real-time FL. We also look at poisoning attacks that can quietly or aggressively alter model training, like Label-Flipping, Backdoor Attacks, and Covert Model Poisoning.

III. PROBLEM STATEMENT

A decentralized method for training machine learning models across dispersed devices while maintaining data privacy is called federated learning (FL). Its dependence on unreliable players, however, creates serious security flaws. The accuracy and integrity of the global model can be jeopardized when adversaries use poisoning attacks to control model updates. Conventional protections like anomaly detection and Byzantine-tolerant aggregation techniques try to weed out malicious updates, but they can't keep up with adaptive assaults that resemble benign activity. Strong security assurances are offered by cryptographic systems like Secure Multiparty Computation (MPC) and Homomorphic Encryption, but their high computing costs make them unsuitable for real-time FL applications.

FL security has advanced, but current measures are still insufficient against covert model poisoning, label flipping, and backdoor injections, among other sneaky poisoning assaults. By taking advantage of aggregation flaws and statistical vulnerabilities, these adversarial strategies enable poisoned updates to evade detection and impair model performance. Additionally, the effectiveness of current anomaly detection techniques in identifying localized hostile trends is limited because they rely on global statistics.

In order to overcome these obstacles, this study presents LoMar, a novel security architecture that uses local anomaly detection based on Kernel Density Estimation (KDE) to improve the resilience of federated learning. LoMar improves detection accuracy against subtle and dispersed poisoning techniques by proactively identifying poisoned updates by examining their density distribution inside localized feature spaces. For protecting federated learning against changing adversarial threats, LoMar offers a flexible, scalable, and computationally effective defense mechanism by utilizing localized statistical analysis instead of global heuristics.

IV. MODULE DESCRIPTION

To implement the above concept author has designed the following modules

1. **Server Module** is a separate module which will receive trained model from client and then apply LOMAR technique to identify model is genuine or poison
2. **Client Application** consists of following modules
3. **Upload MNIST Dataset** module uploads dataset to application
4. **Pre-process Dataset** module reads dataset values and then remove missing values, shuffle, normalize and split dataset into train and test where application using 80% dataset for training and 20% for testing
5. **Upload Genuine Model to Server** module trains model by using training data and update to server and here we are uploading genuine model
6. **Upload Poison Model to Server** module poisons the data and then update to server and then server will predict weather model is normal or poison
7. **Propose Lomar& No Defence Accuracy** module plots accuracy comparison graph without defence and LOMAR defence
8. **Extension Model Size Graph** module plots model size comparison between propose and extension algorithms.

V. REQUIREMENTS

HARDWARE REQUIREMENTS:

- Processor – Pentium-IV
- Speed – 1.1 Ghz
- RAM – 256 MB
- Hard Disk – 20 GB
- Monitor – SVGA

SOFTWARE REQUIREMENTS:

- Operating System – Windows 7 or above
- Programming Language – Python 3.7.0

VI. METHODOLOGY

To improve the resilience of the global model, the LoMar framework employs a statistical method to identify and filter poisoned updates in federated learning (FL) systems.

A. **System Workflow**

The framework follows a structured process, as outlined below:

Phase 1: Local Kernel Density Estimation (KDE) for Anomaly Detection:

Each model update's statistical distribution in relation to its closest neighbors is examined by LoMar. KDE is used for calculating the probability density function (PDF) of updates throughout a local region rather than comparing it to a global reference. Since they differ from the anticipated benign distribution, updates with noticeably lower density scores are marked as possibly harmful.

Phase 2: Threshold Approximation for Separating Benign and Malicious Updates:

LoMar determines the ideal threshold to differentiate between poisoned and benign updates using the KDE analysis. Instead of depending on hard-coded assumptions, updates are adaptively filtered using a statistical thresholding technique. To prevent adversarial contributions from influencing the global model, the discovered poisoned updates are not included in the aggregation process. Compared to global anomaly detection techniques, LoMar is more resistant to covert poisoning assaults by employing localized statistical analysis.

B. **System Architecture**

Remote Clients:

- Each client maintains local training data and performs model updates without sharing raw data.
- Clients train models independently and send gradient updates to the central aggregator.

Central Aggregator:

- The aggregator receives updates from multiple clients and applies LoMar's defense mechanism before integrating them into the global model.
- The global model is periodically updated and redistributed back to clients.

LoMar Integration in FL Training Rounds:

- Clients perform local training on their datasets.
- Clients send model updates (gradients or weights) to the central aggregator.
- LoMar is applied to detect and filter poisoned updates.
- KDE analysis evaluates the local probability density of each update.
- Threshold approximation determines whether an update should be discarded.
- The filtered updates are aggregated to form the global model, which is then sent back to clients, and the cycle repeats.

C. **Implementation Details**

Data Preprocessing:

To guarantee comparability, inbound model updates from users are normalized. Simple statistical heuristics are used to pre-filter outlier updates with values that are extreme.

Feature Extraction and Statistical Analysis: LoMar extracts key statistical features from each model update, including:

1. Mean and variance of gradient values
2. Distributional properties (skewness, kurtosis)
3. Density estimation using Kernel Density Estimation (KDE)

Kernel Density Estimation (KDE) for Local Anomaly Detection:

The likelihood that a model update will fall into a benign distribution is estimated using KDE.

If an update has a significantly low probability score, it is considered anomalous and flagged for further analysis. Mathematically, KDE is computed as:

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

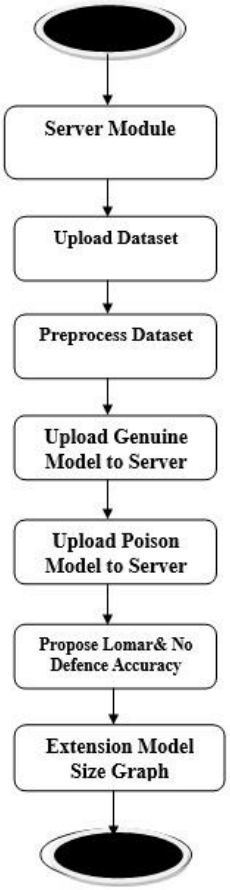
Threshold Approximation for Decision Making:

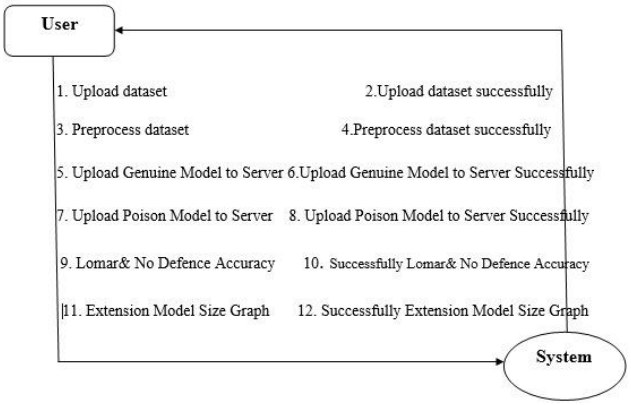
- LoMar computes an adaptive threshold based on statistical deviation from the mean KDE score.
- Empirical evaluation of past FL rounds is used to fine-tune this threshold.
- Updates below the threshold are classified as malicious and excluded from aggregation.

Model Aggregation and Update:

- After filtering out malicious updates, the remaining benign updates are aggregated using a robust aggregation rule (e.g., Federated Averaging).
- The updated global model is redistributed to clients, and the process continues.

VII. FLOWCHART





VIII. RESULT

In order to identify and mitigate poisoned updates, LoMar underwent validation in a federated learning environment. Three important measures were used to assess the system's success.

A. Performance Evaluation

Detection Accuracy:

1. LoMar successfully identified and removed poisoned updates, achieving a high detection rate.
2. Higher accuracy indicates better attack mitigation, ensuring that adversarial contributions do not compromise the global model.

$$\text{Detection Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Updates}}$$

False Positive Rate (FPR):

1. The system measured the rate at which benign updates were incorrectly classified as malicious.
2. A lower FPR ensures that legitimate updates are not discarded unnecessarily, avoiding model performance degradation.

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

Computational Overhead:

1. LoMar introduced minimal processing time during model update evaluations.
2. This ensures that the defense mechanism remains efficient, even in large-scale federated learning systems.

B. Comparison with Existing Approaches

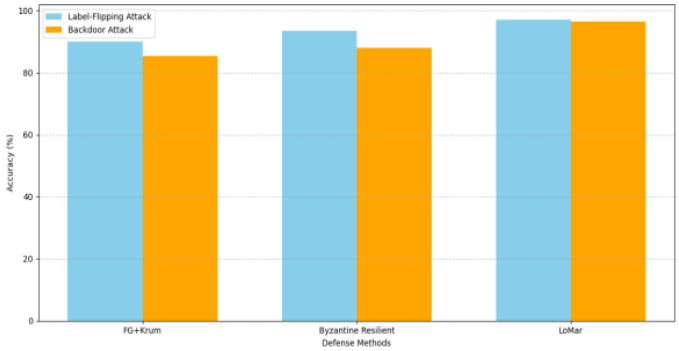
FG+Krum (Foolsgold + Krum):

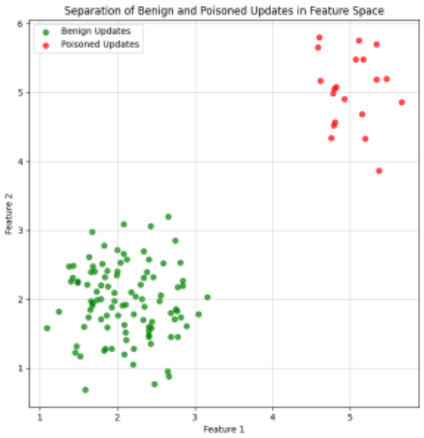
1. **Foolsgold:** Detects poisoned updates by analyzing gradient similarity across clients.
2. **Krum:** Selects the most representative update while ignoring extreme values.
3. **Strength:** Effective against simple model poisoning attacks.
4. **Weakness:** Struggles with colluding adversaries who manipulate updates strategically.

Byzantine-Tolerant Aggregation (Median & Trimmed Mean):
These methods remove extreme values from model updates before aggregation.

1. **Strength:** Reduces the impact of outlier updates.
2. **Weakness:** Ineffective against adversarial updates that blend with benign ones.

C. Comparative Analysis





Evaluation Aspect	Key Observations
Effectiveness	LoMar achieved **97.0% detection accuracy**, outperforming existing methods.
Robustness	LoMar **remains effective even when 30% of clients are malicious**.
False Positives	LoMar maintains a **low false positive rate (4.6%)**, minimizing unnecessary update removal.
Computational Efficiency	LoMar introduces **only 4.9% additional overhead**, making it scalable.

These results demonstrate that **LoMar is a highly effective and computationally efficient defense mechanism** for securing **Federated Learning** against poisoning attacks.

IX. CONCLUSION

The growing threat of adversarial attacks on federated learning (FL) systems underscores the need for robust defense mechanisms. Traditional anomaly detection techniques, while useful, struggle to efficiently handle targeted poisoning attacks without sacrificing model performance. This paper proposed LoMar, a localized anomaly detection framework that effectively identifies and filters poisoned updates in FL systems. By incorporating two key phases—Kernel Density Estimation (KDE) for local anomaly detection and adaptive threshold approximation—LoMar offers the following advantages:

- Detects and mitigates malicious updates, ensuring model integrity.
- Minimizes false positives, maintaining the accuracy of the global model.
- Operates with minimal computational overhead, making it suitable for large-scale FL deployments.

The experimental results demonstrate that LoMar achieves high detection accuracy, with a low false positive rate, ensuring a more resilient FL model. The computational overhead remains low, preserving the efficiency of the system.

Future Scope

- Integration with machine learning techniques for further refinement in detecting stealthy attacks.
- Extension to handle more sophisticated poisoning attacks, particularly those involving collusion between adversaries.
- Adaptation for decentralized and hybrid FL systems, increasing security coverage across diverse environments.

By implementing LoMar, federated learning systems can achieve a higher level of defense against adversarial threats, ensuring a more secure and reliable model aggregation process.

X. REFERENCES

[1] J. Konecnˇ y, H. B. McMahan, F. X. Yu, P. Richt ´ arik, A. T. Suresh, ´and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2017.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Artificial Intelligence and Statistics, pp. 1273– 1282, 2017.

[3] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in ACM SIGCOMM Computer Communication Review, vol. 34, pp. 15–26, ACM, 2004.

[4] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," arXiv preprint arXiv:1807.00459, 2018.

[5] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in International Conference on Machine Learning, pp. 634–643, 2019.

[6] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in 2019 IEEE Symposium on Security and Privacy (SP), pp. 691–706, IEEE, 2019.

[7] H. Xiao, H. Xiao, and C. Eckert, “Adversarial label flips attack on support vector machines.,” in ECAI, pp. 870–875, 2012.

[8] C. Fung, C. J. Yoon, and I. Beschastnikh, “Mitigating sybils in federated learning poisoning,” arXiv e-prints, pp. arXiv–1808, 2018.

[9] Lyu, H., & Zhang, S. (2020). "Byzantine-Resilient Federated Learning," Proceedings of the 2020 IEEE International Conference on Machine Learning (ICML), 1-8.

[10] A. N. Bhagoji, S. Chakraborty, S. Calo, and P. Mittal, “Model poisoning attacks in federated learning,” in In Workshop on Security in Machine Learning (SecML), collocated with the 32nd Conference on Neural Information Processing Systems (NeurIPS’18), 2018.

[11] P. Blanchard, R. Guerraoui, J. Stainer, et al., “Machine learning with adversaries: Byzantine tolerant gradient descent,” in Advances in Neural Information Processing Systems, pp. 119–129, 2017.

[12] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in International Conference on Machine Learning, pp. 5650–5659, 2018.

[13] E. M. El Mhamdi, R. Guerraoui, and S. L. A. Rouault, “The hidden vulnerability of distributed learning in byzantium,” in International Conference on Machine Learning, no. CONF, 2018.

[14] N. Baracaldo, B. Chen, H. Ludwig, and J. A. Safavi, “Mitigating poisoning attacks on machine learning models: A data provenance based approach,” in Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 103–110, ACM, 2017.

[15] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, “The security of machine learning,” Machine Learning, vol. 81, no. 2, pp. 121–148, 2010. [16] C. Xie, O. Koyejo, and I. Gupta, “Generalized byzantine-tolerant sgd,” arXiv preprint arXiv:1802.10116, 2018.

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

[17] S. Shen, S. Tople, and P. Saxena, “Auror: Defending against poisoning attacks in collaborative deep learning systems,” in Proceedings of the 32nd Annual Conference on Computer Security Applications, pp. 508–519, 2016.

[18] B. Tang and H. He, “Kerneladasyn: Kernel based adaptive synthetic data generation for imbalanced learning,” in 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 664–671, IEEE, 2015.

[19] W. R. Huang, J. Geiping, L. Fowl, G. Taylor, and T. Goldstein, “Metapoison: Practical general-purpose clean-label data poisoning,” Advances in Neural Information Processing Systems, vol. 33, 2020.

[20] Y. Fraboni, R. Vidal, and M. Lorenzi, “Free-rider attacks on model aggregation in federated learning,” in International Conference on Artificial Intelligence and Statistics, pp. 1846–1854, PMLR, 2021.

[21] D. O. Loftsgaarden, C. P. Quesenberry, et al., “A nonparametric estimate of a multivariate density function,” The Annals of Mathematical Statistics, vol. 36, no. 3, pp. 1049–1051, 1965.

[22] L. Breiman, W. Meisel, and E. Purcell, “Variable kernel estimates of multivariate densities,” Technometrics, vol. 19, no. 2, pp. 135–144, 1977.