Lomar: A Lightweight Local Strategy for Defending Against Poisoning Attacks in Federated Learning

G.Swathi Reddy[1], Paloju Sravanthi [2], Paritala Sai Raghu[3], Mukkeri Swapna[4], S Krishna Vardhan Reddy[5]

[1] Assistant Professor, Department of Computer Science and Engineering(AI & ML), Samskruthi College of Engineering And Technology , Kondapur(V), Ghatkesar(M), Medchal(D),Telangana

[2,3,4,5]BTech Students ,Department of Computer Science and Engineering(AI & ML), Samskruthi College of Engineering And Technology , Kondapur(V), Ghatkesar(M), Medchal(D),Telangana

***Abstract*—Federated Learning allows multiple clients to train a shared model collaboratively without exchanging their raw data, which helps protect privacy. Despite this advantage, FL systems are still exposed to poisoning attacks, where malicious participants intentionally send harmful updates to disrupt the model or embed hidden backdoors.To tackle this issue, Lomar introduces a local defence strategy that works directly on each client, removing the need to depend on a fully trusted central server. Instead of relying on global verification, Lomar focuses on analysing the behaviour of model updates at the client side. It continuously compares current updates with past training patterns and identifies unusual deviations that may indicate malicious activity.When such suspicious updates are detected, Lomar takes corrective action by filtering them out, adjusting their impact, or reducing their influence before they are shared for aggregation. This ensures that harmful contributions do not significantly affect the global model.By applying this defence locally, Lomar strengthens the resilience of federated learning while still preserving its core privacy benefits. It is also designed to integrate easily with existing FL frameworks and aggregation techniques, making it practical for real-world use. Lomar offers a scalable and privacy-conscious solution to protect federated learning systems, especially in sensitive areas like healthcare, finance, and IoT applications.**

***Keywords*—Federated learning security, poisoning attack detection, local defense mechanism, malicious client mitigation, gradient anomaly analysis, privacy-preserving machine learning, robust model aggregation.**

I. INTRODUCTION

Federated Learning has emerged as an effective distributed machine learning paradigm that allows multiple clients to collaboratively train a shared global model without exchanging raw data. By keeping sensitive information on local devices, FL helps address privacy concerns and regulatory requirements, making it well-suited for domains such as healthcare, finance, Internet of Things , and mobile applications [1]. However, the decentralized and open participation nature of FL introduces serious security risks, especially in scenarios where clients cannot be fully trusted.

Among these risks, poisoning attacks are considered one of the most significant threats to federated learning [1][2]. In such attacks, adversarial clients deliberately manipulate their local data or model updates to compromise the integrity of the global model. These attacks can be classified into two main types: untargeted attacks, which aim to degrade overall model performance, and targeted attacks, which cause the model to produce incorrect outputs for specific inputs while maintaining normal behaviour otherwise. The distributed architecture of FL further complicates defence, as the central server has limited visibility into individual client activities, making detection and mitigation challenging [9].

To address these threats, existing solutions primarily focus on server-side defences, including robust aggregation methods, anomaly detection, update filtering, and trust-based weighting mechanisms. While these approaches offer partial protection, they often rely on the assumption of a reliable and powerful central server and may introduce additional computational overhead [8]. Moreover, advanced adversaries can adapt their attack strategies to evade global detection, reducing the effectiveness of purely server-side techniques [6]. These challenges highlight the importance of developing complementary defence mechanisms that operate at the client level.

Lomar (Local Monitoring and Response) is proposed as a client-side defence approach to mitigate poisoning attacks in federated learning.

Unlike traditional methods that depend solely on the server, Lomar enables each client to monitor its own training process and model updates [7]. It analyses deviations in gradients or model parameters across training rounds to detect abnormal patterns that may indicate malicious behaviour. Once such anomalies are identified, the client can take corrective actions, such as filtering or adjusting suspicious updates before sharing them with the central server [15]. By distributing part of the defence responsibility to clients, Lomar improves the robustness and scalability of FL systems while preserving their inherent privacy advantages. This local-first strategy reduces reliance on a fully trusted server and adds an extra layer of protection against adversarial manipulation. As federated learning continues to expand into real-world, security-critical applications, solutions like Lomar are essential for ensuring secure and reliable collaborative learning environments [20].

II. RELATED WORK

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. Federated Learning is a distributed machine learning paradigm in which the objective is to train a high-quality centralized model while keeping training data decentralized across a large number of clients. These clients typically operate under unreliable and low-bandwidth network conditions, making communication efficiency a critical concern. In each training round, participating clients independently compute model updates using their local data and transmit these updates to a central server, where they are aggregated to form an improved global model. Mobile devices, such as smartphones, represent a common and practical deployment scenario for this setting. To address the challenge of high uplink communication costs, this paper introduces two complementary strategies for reducing the size of client-to-server transmissions. The first approach, termed structured updates, restricts the model updates to a lower-dimensional space parameterized by a reduced number of variables. This can be achieved through techniques such as low-rank representations or the application of random masking. The second approach, referred to as sketched updates, allows clients to compute full model updates locally and subsequently compress them before transmission . [1]

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2017.Modern mobile devices generate and store vast amounts of data that can be leveraged to train machine learning models capable of significantly enhancing user experience. Applications such as language modeling can improve speech recognition and text input, while image-based models can assist in automatically selecting high-quality photographs. Despite its value, this data is often highly sensitive, voluminous, or both, making centralized data collection and conventional training approaches impractical due to privacy and communication constraints. To address these challenges, we promote a decentralized learning paradigm in which training data remains on individual mobile devices and a global model is learned through the aggregation of locally computed updates. This approach, known as Federated Learning, enables collaborative model training while preserving data privacy and reducing the need for raw data transmission. In this work, we introduce a practical federated learning framework for training deep neural networks using an iterative model averaging strategy. We conduct an extensive empirical evaluation across five distinct model architectures and four real-world datasets to assess the effectiveness of the proposed method. The experimental results demonstrate that the approach remains robust even under highly unbalanced and non-independent and identically distributed data conditions, which are inherent characteristics of federated learning environments.[2]

F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," 2004. Large-scale Internet-based applications can significantly benefit from the ability to estimate network round-trip times between hosts without requiring direct communication. However, explicit latency measurements are often impractical, as the overhead of performing such measurements can exceed the benefits gained from proximity-aware optimization. Vivaldi addresses this challenge through a lightweight and fully distributed coordinate-based system that assigns synthetic network coordinates to hosts. The Euclidean distance between these coordinates serves as an accurate predictor of communication latency. Vivaldi operates without centralized infrastructure or designated control nodes and requires only limited latency information exchanged with a small number of peers. Due to its minimal communication overhead, Vivaldi can integrate seamlessly with existing application traffic and scale efficiently to large networks. Experimental evaluation using latency data derived from measurements across 1,740 Internet hosts demonstrates that a two-dimensional Euclidean model augmented with height vectors achieves low

prediction error, with a median relative round-trip time error of approximately 11%.[3]

E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," 2018. Federated learning allows a large number of participants to collaboratively train deep learning models without directly sharing private training data. For instance, smartphones can jointly train a next-word prediction model without exposing individual typing behavior. Despite these privacy benefits, federated learning is vulnerable to malicious participants. We show that a single adversarial client can inject hidden backdoor behavior into the global model, causing it to produce attacker-specified outputs for selected inputs while functioning normally otherwise. We introduce a novel model-poisoning strategy based on model replacement, where an attacker can manipulate the global model within a single training round to achieve perfect accuracy on the backdoor task. Additionally, a constrain-and-scale technique enables the attack to evade anomaly detection defenses by embedding evasion objectives directly into the attacker's training process.[4]

A.N. Bhagoji, S. Chakraborty, and S. Calo, "Analyzing federated learning through an adversarial lens," 2019. Federated learning distributes model training across numerous agents that retain their data locally and share only parameter updates with a central server for aggregation. From an adversarial perspective, this setup introduces vulnerabilities to model poisoning attacks, even when carried out by a single, non-colluding malicious participant. we investigate targeted poisoning attacks designed to cause misclassification of specific inputs with high confidence. We explore multiple attack strategies, beginning with amplifying malicious updates to counteract benign contributions. To improve stealth, we propose an alternating optimization approach that balances minimizing training loss with achieving adversarial objectives. Further enhancements are obtained through estimating benign update distributions to refine attack effectiveness. we demonstrate that the decision explanations of poisoned models are nearly indistinguishable from those of benign models. [5]

III. DATASET DETAILS

The proposed system utilizes the MNIST dataset to train and evaluate both genuine and poisoned models within a federated learning environment. MNIST (Modified National Institute of Standards and Technology) is a widely used benchmark dataset that contains 70,000 grayscale images of handwritten digits from 0 to 9. Each image is of size 28×28 pixels, making it well-suited for lightweight models and efficient experimentation.

Typically, 60,000 images are used for training and 10,000 for testing. In this work, the dataset is further divided so that 80% is allocated for training and 20% for testing, ensuring a balanced evaluation process.After being uploaded through the client application, the dataset undergoes preprocessing steps such as shuffling and normalization. The pixel values, originally ranging from 0 to 255, are scaled down to a range of 0 to 1 to enhance training stability and improve convergence. Since MNIST is a well-structured dataset, it does not contain missing values, simplifying the preprocessing stage.

The cleaned and normalized data is then used to train both a genuine model and a poisoned model, where poisoning is introduced by deliberately altering the data distribution.These locally trained models are subsequently sent to the server, where the Lomar mechanism evaluates the updates to identify any signs of poisoning. In this context, the MNIST dataset serves as a reliable benchmark to assess the performance of the proposed Lomar-based defence.

IV. PROPOSED METHODOLOGY

Most existing defences against data poisoning attacks are designed for centralized machine learning systems. One common approach focuses on identifying malicious data based on its negative influence on the model's performance. For example, the "Reject on Negative Impact" method evaluates the contribution of each training sample and removes those that significantly degrade the model. Another approach aims to make the learning process robust by optimizing against the worst-case loss under potential attack strategies.Although federated learning enhances data privacy by keeping data on client devices, it is still vulnerable to attacks. If a malicious actor compromises a client, they can manipulate its local training data and generate a corrupted model update. When such poisoned updates are sent to the central server and aggregated, they can negatively affect the global model, leading to incorrect predictions across all participating clients.To address this problem, the authors propose the Lomar (Local Monitoring and Response) algorithm, which provides a defence mechanism against poisoning attacks at the client level.

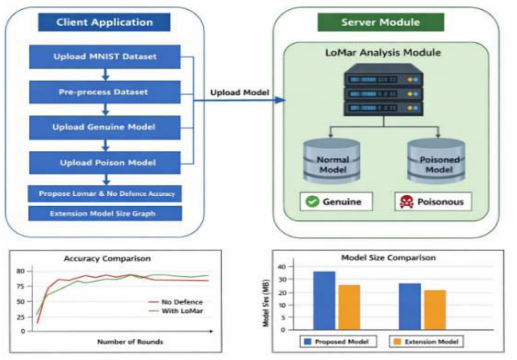

Figure 1: SYSTEM ARCHITECTURE

The figure [1] illustrates the system architecture of a client uploading genuine and poisoned MNIST models to a server, where the LoMar module detects poisoning, compares accuracy, and evaluates model.

In poisoning scenarios, attackers often alter training labels, causing abnormal changes in model weights during training. Lomar detects such irregularities by analysing variations in model updates using Kernel Density Estimation. This technique examines the similarity between updates by considering their neighbouring patterns.If a model has been trained on poisoned data, its update will significantly deviate from those of other clients, resulting in larger distances between neighbouring updates. By comparing these variations against predefined threshold values, the system can determine whether a model update is suspicious.

V. RESULT AND DIS CUSSION

In federated learning all clients will send trained model to server and this model will be having huge size and sending such huge models will required lots of network bandwidth and uploading time will also high. So to reduce model size as extension we are employing model compression technique which will compress model before uploading to server and this compress model will save space up to 10% and network uploading will be faster.

Figure [2] illustrates train genuine model on training dataset and then upload to server as federated learning and then will get below response.

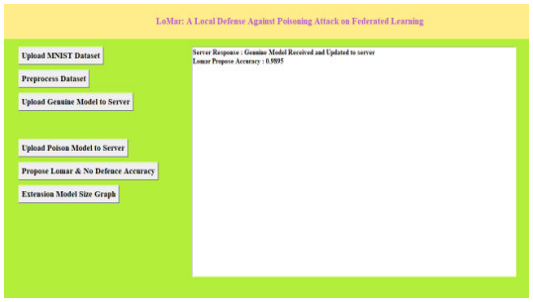

Figure 2: Upload Genuine Model to Server

Figure [3] illustrates screen in white colour text can see server got 99% accuracy on LOMAR techniques on its own test data.

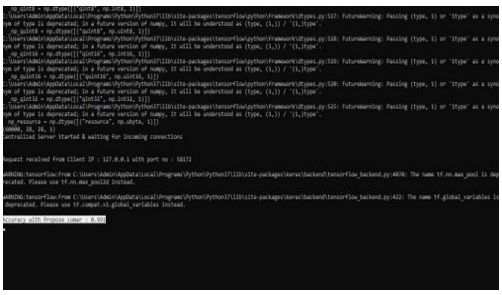

Figure 3 : Genuine Model Accuracy

In Figure [4] screen server received poison model and ignored updation and client got 88% accuracy on poison model so by employing propose techniques application predicting weather model is normal or poisoned.

Figure 4 : Upload Poison Model to Server

In figure [5] screen x-axis represents algorithm names and y-axis represents accuracy and in both techniques propose LOMAR got high accuracy and now click on ‘"Extension Model Size Graph’ button to get below comparison graph

Figure 5: Propose Lomar & No Defence Accuracy

In figure [6] graph x-axis represents technique names and y-axis represents model size and in both techniques compress model extension got less size compare to normal model uploading. In text area output also we can see normal model size and after compression we can see the model size.

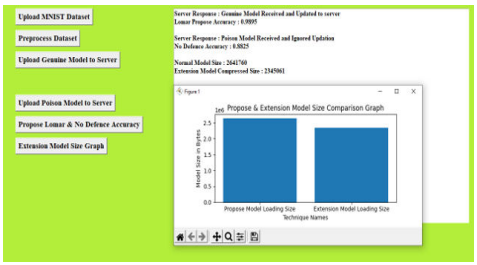

Figure 6: Extension Model Size Graph

DISCUSSION

The experimental evaluation highlights several important insights into the behavior and effectiveness of the proposed LoMar-based federated learning system. First, the results confirm that federated learning is highly vulnerable to poisoning attacks when no defense mechanism is employed. Even a single malicious update can significantly reduce model accuracy or introduce harmful backdoor behavior. This reinforces the necessity of integrating robust security mechanisms into federated learning frameworks.

The consistent improvement in accuracy observed in the "with LoMar" scenario demonstrates its ability to maintain model performance even under adversarial conditions.Another important observation is the trade-off between security and efficiency. While additional analysis is performed to detect poisoned updates, the overhead introduced by LoMar remains minimal compared to the benefits gained in robustness. Furthermore, the extension model compression results show that communication costs can be significantly reduced, which is crucial for large-scale federated learning deployments involving mobile or edge devices.

However, the current implementation focuses on a controlled experimental setup using the MNIST dataset and a limited number of clients. In real-world scenarios, data distributions are more complex and adversaries may employ more sophisticated attack strategies. Future work can explore the effectiveness of LoMar under diverse datasets, larger client populations, and adaptive adversarial behaviors. Additionally, combining LoMar with other defense techniques such as robust aggregation or trust-based client selection could further strengthen federated learning security.Overall, the discussion confirms that LoMar is a promising and practical defense mechanism, while also highlighting opportunities for future enhancement and real-world adoption.

VI. CONCLUSION

This work presented a secure and efficient federated learning framework that integrates the LoMar (Local Monitoring and Response) technique to defend against model poisoning attacks. The proposed system was designed with a clear separation between the client application and the server module, enabling realistic simulation of federated learning environments. Using the MNIST dataset, the client performs data preprocessing, trains both genuine and poisoned models, and submits model updates to the server. The server-side LoMar analysis module evaluates received updates to determine whether they are benign or malicious before aggregation.Experimental results demonstrate that the proposed approach is effective in identifying poisoned model updates and preventing them from degrading the global model. When genuine models were uploaded, the server accurately classified them as normal and achieved high accuracy, validating the correctness of the learning process. In contrast, poisoned models were successfully detected and ignored, thereby preserving the integrity of the global model. Accuracy comparison results clearly show that federated learning with LoMar defense consistently outperforms the no-defense scenario, especially under adversarial conditions.In addition to robustness, the system also addresses communication efficiency. The extension model size comparison illustrates that compressed model updates significantly reduce transmission overhead without compromising accuracy. This is particularly important in federated learning settings where clients operate under limited bandwidth and computational resources.

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] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in 29th {USENIX} Security Symposium ({USENIX} Security 20), pp. 1605–1622, 2020.

[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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] B. Tang and H. He, “A local density-based approach for outlier detection,” Neurocomputing, vol. 241, pp. 171–180, 2017.