Pegem Journal of Education and Instruction, Vol. 13, No. 4 , 2023 (pp. 418-426)

RESEARCH ARTICLE WWW.PEGEGOG.NET

# LoMar: Regional Defence Counter Poisoning Attack on Federated Learning

[1]Mrs. K. Anusha , [2]N. Akshitha, [3]Afiya Saba, [4]M. Avanthi

[1]Assistant Professor Computer Science and Engineering, Sridevi Women's Engineering College, Hyderabad, India

Email: anusha.karingala@gmail.com,

[2,3,4,]B.Tech Student, Department of Computer Science and Engineering, Sridevi Women's Engineering College, Hyderabad, India.

ABSTRACT—

The training data stays spread to distant clients in a network using federated learning (FL), which offers a highly efficient decentralised machine learning solution. Even if FL makes it possible to use IoT devices to build a mobile edge computing framework that protects user privacy, new research shows that this can be vulnerable to poisoning assaults performed by distant clients. Introducing the Local Malicious Factor (LoMar) defence algorithm, a two-stage solution against FL poisoning assaults. The first step is for LoMar to evaluate the model changes sent by each distant client. It does this by calculating their relative distribution among their neighbours, which is done using a kernel density estimation approach. Phase II involves using statistical methods to approximatively determine an optimum threshold for differentiating bad from clean updates. The experimental findings demonstrate that our defence technique can successfully safeguard the FL system, and we have performed thorough tests on four real-world datasets. And more especially, how well the defence worked on the Amazon dataset while using a label-flipping strategy. Research shows that LoMar improves over FG+Krum in terms of target label testing accuracy (from 96.0% to 98.8%) and overall averaged testing accuracy (from 90.1% to 97.0%).

## INTRODUCTION

It has been shown that FEDERATED Learning (FL) is a powerful distributed ML framework for training a combined model using distributed data. Thanks to the advanced Internet of Things (IoT) applications, it has recently received greater attention from researchers. FL offers a privacy-preserving learning framework that can solve distributed optimisation issues

**Corresponding Author e-mail:** anusha.karingala@gmail.com
**How to cite this article:** 1Mrs. K. Anusha , 2N. Akshitha, 3Afiya Saba, 4M. Avanthi. LoMar: Regional Defence Counter Poisoning Attack on Federated Learning.Pegem Journal of Education and Instruction, Vol. 13, No. 4, 2023, 418-426.**Source of support:** Nil **Conflicts of Interest:** None.

**DOI:** 10.48047/pegegog.13.04.49

**Received:** 12.10.2023

**Accepted:** 22.11.2023 **Published:** 24.12.2023

without compromising users' access to their private training datasets by facilitating the exchange of learning information across network nodes. Two primary components make up a FL system:

aggregators and remote clients. Remote clients handle their own private training data and learn on their own to update their learning models. An aggregator receives updates from these clients and uses them to update a joint model according to an aggregation rule. This process repeats itself many times. As a result of its distributed design, FL is susceptible to a number of attacks that might compromise the distant clients and hence the learning system. It is common for an attacker to take use of the privacy feature—the private distant training dataset—to compromise several clients, alter their local training processes, and ultimately lower the joint model's performance. Specifically, it reveals two points at which the FL system is vulnerable to poisoning assaults: i) local data collection: this kind of attack can alter or add malicious data to the data that is already there. ii) remote model training: this kind of attack can directly inject poisoned parameters into the model that is trained remotely before it is sent back to the aggregator. Consequently, poisoning assaults modify the FL aggregation process maliciously. Keep in mind that the FL joint learning model may converge to make the attackers difficult to identify via intentional manipulation of data or model poisoning attempts. The FL system must be protected against poisoning assaults, hence a defence mechanism must be developed. Sanitising the poisoned remote updates to generate a trustworthy joint model is often the criterion for evaluating the efficacy of a FL defence mechanism. The use of geometric distances or angle discrepancies between each pair of remote updates is one kind of current technique for detecting rogue updates. Different forms of defence are created with the Byzantine tolerance in mind. Current defence techniques may be circumvented by poisoning assaults with little dangerous material, according to recent research. Rather than analysing local feature patterns of malicious remote updates, most contemporary defence techniques just see them as a global abnormality to the FL system. In this research, we provide a novel defence mechanism that employs a local feature analysis approach; specifically, we assess the malevolence of poisoned distant updates by examining the features of their model parameters. Our proposed two-stage defence technique, Local Malicious Factor (LoMar), may identify FL abnormalities from a local perspective rather than the current global one. Intuitively, the idea behind the proposed LoMar is that each remote update in the FL system can be thought of as being generated from a specific distribution of the parameters. This allows us to evaluate the maliciousness of the updates based on the statistical characteristic analysis of the model parameters. In particular, when a client sends an update to the aggregator, LoMar uses its closest neighbours to do feature analysis on the update rather than the whole collection of distant status updates. In order to determine the level of malevolence, a non-parametric local kernel density estimation approach is used to estimate the relative distribution of the distant update in its close vicinity. Our suggested LoMar defence method is tested and assessed via extensive experiments and theoretical study. Finally, we would like to emphasise the following as the main contributions of this paper:

We address poisoning attempts against FL by proposing LoMar, a novel two-phase defence method.

In addition to the theoretical analysis of LoMar, we perform comprehensive performance evaluations of LoMar under two types of poisoning attacks on FL. The proposed LoMar defence algorithm assesses the harmfulness of remote updates by considering their neighbourhood and analysing the features of the statistical model's parameters using a non-parametric relative kernel density estimation method.

As compared to other FL defence algorithms, the suggested LoMar performs better. After that, the paper is structured like this: The second section lays out the challenge we have in defending the FL against poisoning assaults. A description of the steps used to create the LoMar algorithm is provided in Section 3. Results and assessments of our experiments are presented in Section 4. After a more thorough summary of the relevant work in Section 5, the paper concludes and discusses potential future work in Section 6.

## RELATED WORK

### "Efficiency in communication: Techniques for federated learning,"

While training data is disseminated across a large number of clients, each with their own very sluggish and unstable network connection, the purpose of federated learning is to build a high-quality centralised model. In this scenario, we think about learning algorithms in which, on each iteration, clients individually update the current model using their local data and send it to a central server, which then aggregates all of the modifications to create a new global model. Mobile phones make up the bulk of the clientele here, thus getting your message out quickly is crucial.

In this paper, two methods are suggested for lowering the uplink communication costs: structured updates and sketched updates. Structured updates involve learning an update from a limited space with fewer parameters, such as a random mask or lowrank.

Sketched updates utilise a combination of quantization, random rotations, and sub sampling to compress the update before sending it to the server. A two-order-ofmagnitude reduction in communication cost is shown experimentally on convolutional and recurrent networks using the suggested approaches.

**"Decentralization-efficient learning of deep networks through communication,"**

The user experience on modern mobile devices may be significantly enhanced by using the abundance of data that is appropriate for learning models. Take language models, which may enhance text input and voice recognition, as an example. Then there are picture models, which can automatically choose high-quality photographs. It may not be possible to access the data centre for training using traditional methods due to the enormous amount, privacy concerns, or both of this rich data. We propose a different approach that uses locally calculated updates to develop a shared model, rather than distributing training data across mobile devices. The name we use to describe this distributed method is Federated Learning. Using five distinct model architectures and four datasets, we describe a workable approach to federated learning of deep networks based on iterative model averaging and undertake a thorough empirical assessment. The results show that the method works well even when faced with the non-IID and imbalanced data distributions that are common in this kind of environment. The main limitation is the cost of communication; we demonstrate a 10100x decrease in the number of communication rounds needed compared to synchronised stochastic gradient descent.

**"Vivaldi: A decentralised structure for network coordinates,"**

A way to anticipate other hosts' round-trip timings without contacting them beforehand might be useful for large-scale Internet applications. As a result of the potential negative impact on efficiency caused by the high expense of measurement, explicit measurements are not always a desirable option. A simple and lightweight technique, Vivaldi, gives hosts synthetic coordinates in such a way that the distance between their coordinates precisely predicts the connection delay between them. Thanks to its decentralised design, Vivaldi doesn't need a central server or any special infrastructure to run. Another benefit is how efficient it is: after gathering latency information from a small number of hosts, a newly-minted host may efficiently calculate accurate coordinates for itself. Since it doesn't need much communication, Vivaldi can scale to several hosts by riding on the application's communication patterns. Testing Vivaldi on a simulated network with latencies measured from 1740 Internet hosts reveals that these hosts can be accurately embedded in a 2dimensional Euclidean model with height vectors, resulting in a median relative error of 11% in round-trip time prediction.

### "Federated learning backdoor technique,"

With federated learning, hundreds of people may build a deep learning model independently of one another, protecting their own training data from prying eyes. In order to train a next-word predictor for keyboards anonymously, for instance, many smart phones may work together. We show that any federated learner can insert hidden backdoor functionality into the joint global model. This could be done to make sure that a word predictor fills certain sentences with a word that the attacker chooses or that an image classifier assigns a label that the attacker chooses for images with specific features.

We develop and assess a novel modelpoisoning technique that relies on the replacement of models. If an adversary is chosen within a single federated learning cycle, the global model can instantly achieve a backdoor task accuracy of 100%. We compare it to data poisoning and find that it performs much better under various assumptions for the common federatedlearning tasks. By teaching the attacker to include the evasion into their loss function, our general constrain-and-scale approach is able to avoid anomaly detection-based defences as well.

### "Analysing adversarial learning in federated contexts,"

For iterative aggregation at the server, federated learning spreads model training across a variety of agents. These agents, governed by privacy considerations, train using their local data but communicate only changes to the model's parameters. In this study, we investigate the possibility of model poisoning assaults on federated learning, whereby one malevolent actor, not involved in collusion, aims to make the model confidently misclassify a selected set of inputs. We look at many ways to launch this assault, beginning with just increasing the malicious agent's update to counteract the impact of other agents' updates. We provide a minimization technique that optimises for the adversarial goal and the training loss alternatively to make attacks more stealthy. We then use parameter estimation to enhance the attack success of the benign agents' updates. Last but not least, we demonstrate that there is little visual difference between the explanations generated by benign and malicious models by using a set of interpretability approaches to create visual representations of model choices. Our findings demonstrate that even an adversary with severe constraints may launch stealthy model poisoning assaults, drawing attention to the federated learning environment's susceptibility and the need for robust defence mechanisms.

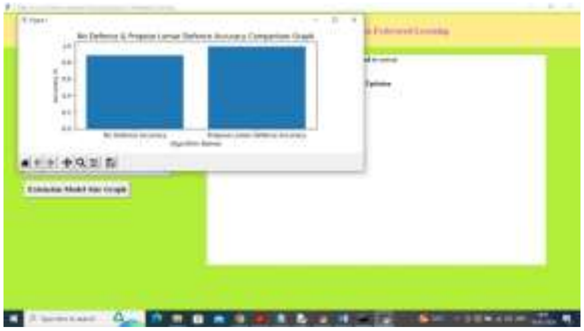
## METHODOLOGY

1) Server Module: This is an independent module that takes the client-trained model and uses the LOMAR approach to determine whether the model is real or toxic.
2) Client Application: The following components make up this application:
3) Upload MNIST Dataset: Uploading datasets to applications is made possible by this module.
4) Pre-process Dataset: This module will take the values from the dataset, clean them up by removing missing values, shuffling and normalising them, and then divide them into two parts: the train and test sets. The application will use the train set for 80% of the testing, and the test set for 20%.
5) Upload Genuine Model to Server: Here we are uploading the real model, and this module will train the model with the help of training data.

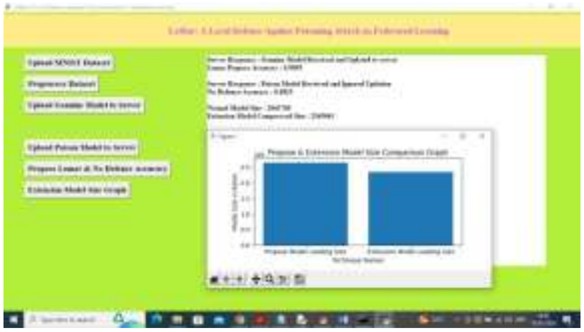
   It will then update the model on the server.
6) Upload Poison Model to Server: In order for the server to determine whether the weather model is normal or poisoned, this module will update the data.
7) Propose Lomar& No Defence

   Accuracy: A comparative graph of accuracy will be produced using this module, with and without the LOMAR defence.
8) Extension Model Size Graph: The model size comparison between the propose and extend techniques will be shown using this module.

## RESULT AND DISCUSSION


No Defence & Propose Lomar Defence Accuracy Comparison Graph
Extension Model Size Graph

The x-axis shows the names of the algorithms, and the y-axis shows the accuracy; both methods suggest that LOMAR achieved good results; to see a comparison, click the "Extension Model Size Graph" button.


Upload MNIST Dataset
Preprocess Dataset
Upload Genuine Model to Server
Upload Poison Model to Server
Propose Lomar & No Defence Accuracy
Extension Model Size Graph
Propose & Extension Model Size Comparison Graph

Compressing and extending models resulted in smaller files when compared to traditional model uploading methods (x-axis: technique names, y-axis: model size). We can see the conventional and compressed model sizes in the text area output as well.

## CONCLUSION

To combat poisoning assaults on FL systems, we provide LoMar, a novel twostage defence method. To show how harmful each update is relative to the reference set— collected by its k-nearest neighbourhoods— LoMar defines a kernel density based estimate in Phase I. Phase II involves LoMar creating an asymptotic threshold that may be used to determine the poisoned updates in a binary fashion. In particular, the offered threshold prevents the defence from reclassifying the FL system's clean updates as malicious. By comparing LoMar's empirical findings on four real-world datasets to four known defence strategies, we show that FL is protected against data and model poisoning attempts.

## 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 eprints, 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 byzantinetolerant 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.