Influence of Malicious Nodes in Federated Learning Models

Shahinul Hoque, Farhin Farhad Riya, Burcum Eken
Department of Electrical Engineering and Computer Science
The University of Tennessee, Knoxville
Knoxville, Tennessee 37996

***Abstract*— Nowadays, robustness is an essential topic in Federated Learning. This paper provides how to test robustness of Federated Learning system using image datasets against malicious users and provides how malicious nodes can affect the performance of trained model. Federated Learning is a combination of multiple separate architectures that guarantee privacy at all times. Federated Learning is a form of collective learning in which single edge devices are trained and then consolidated on the server without data exchange from the edge devices. Federated Learning, on the contrary, offers safe models with no data exchange, providing in a solution with security and data access. We explore the various architectures like ResNet50, MobileNet, VGG16 and various malicious nodes used in Federated Learning. We will also illustrate the performance degradation based on the number of malicious nodes in the system.**

I. INTRODUCTION

Federated Learning (FL) proposed by Google [3][4] is a new way of thinking about Machine Learning that is used when training data is decentralized in different nodes and it has recently attracted a lot of interest from the research community because of its capacity to facilitate collaborative machine learning. It also allows many clients work together to build a strong machine learning model that keeps their data private.

Since Federated Learning is the process of training a Machine Learning model by data distributed among multiple nodes, there are a central node and local nodes, where the central node shares a Neural Network architecture and sends it to the local nodes and the local nodes train the architecture using their own data and send the gradient or the model to the central node. In this way, the local nodes do not need to share their own data. However, the Neural Network is trained on more data and therefore, can perform better.

FL can be used to handle data governance and privacy issues in a distributed setting without needing data sharing [2]. Autonomous vehicles and smart facilities, for example, may train a machine learning model without providing data to a central server [5]. Although Federated Learning has a large-scale structure with many users, the system still faces privacy and security risks. First, while each user's training data is not revealed to the server, the model modification is, which causes security threat, since it is argued that the parameters of a trained neural network allow rebuilding of the initial training data [6][7]. Secondly, Federated Learning is vulnerable to model poisoning attacks in general. Adversary users can directly impact the global model via Federated Learning, allowing for far more powerful attacks [8].

The purpose of this paper is to test the robustness of Federated Learning systems using image datasets utilizing various popular Convolutional Neural Network architectures against malicious users and to showcase how much malicious nodes may affect the model and its accuracy or if there is any effect of choosing a malicious node in the early stage of Federated Learning training.

II. LITERATURE REVIEW

In this section, we introduce the related interesting works on Federated Learning (FL) which can be considered as our motivation for the proposal of analysing the influence of malicious nodes in various FL models.

Federated Learning (FL) provides a privacy-aware model training paradigm that does not involve data sharing and permits participants to join and leave federations at any time. Recent research has nevertheless shown that FL may not always offer adequate robustness assurances. As in the FL training scheme, local nodes (clients) are directly involved in training the model with their own private data this can make the process vulnerable to the malicious local nodes. Any adversarial participant can tamper the global parameter aggregation or poison the global model by attempting various attacks. FL systems are susceptible to both data poisoning [9][10] and model poisoning attacks [11][12][13] in terms of robustness. By purposefully manipulating the local data (data poisoning) or their gradient uploads, malicious participants can attempt to obstruct the global model's convergence or introduce backdoor triggers into it (model poisoning).

More specifically, poisoning attacks can be divided into two types: (1) untargeted attacks like the Byzantine attack, in which the adversary aims to undermine the global model's convergence [14][15]and performance; and (2) targeted attacks like the backdoor attack, in which the adversary aims to implant a backdoor trigger into the global model in order to deceive the model into consistently predicting an adversarial class on a sub-task while maintaining strong performance on the primary task. Significant threats to FL are posed by these attacks. In centralized learning, the server is in charge of ensuring that each participant's privacy and model robustness. In FL, however, any participant can attack the server, sometimes even without using the server. Consequently, it's crucial to understand that to what extend a malicious participant can impose threat on the global models performance, moreover it is also important to learn that

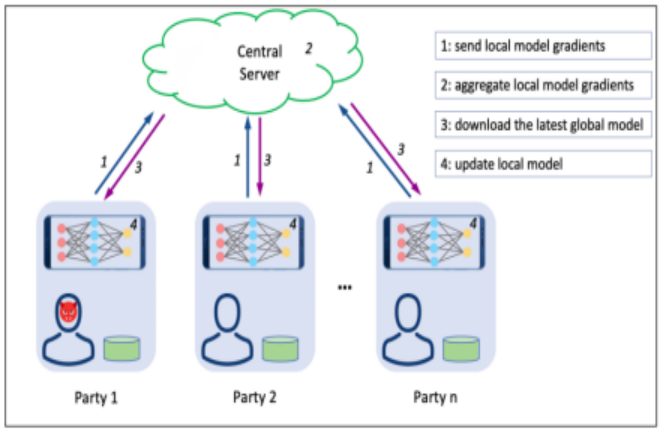

Fig. 1. A typical FL training process, in which malicious participants may pose threats to the FL system

how much a global models performance can degrade if the number of malicious party is more than one.

III. Technical Details

A. Dataset

For our implementation, we have decided to use the popular CIFAR-10 dataset [1] consisting of 60,000 images. The dataset is divided into two parts where 50,000 images are organized as the training dataset, and 10,000 images are organized as the testing dataset. The dataset consists of ten classes, each containing 5000 images for training and 1000 images for testing. CIFAR-10 is one of the most popular dataset used by the Computer Vision community, and therefore, can be considered as a good benchmark to compare the results of various Deep Learning models with a generalized dataset. Furthermore, the dataset contains pre-processed images with a 32x32 resolution and RGB color profile for each image.

B. Model Architectures

In order to create a Federated Learning system or environment, we used various popular Convolutional Neural Network (CNN) architectures currently used by the Deep Learning community. Many of these architectures come from the Imagenet large-scale visual recognition challenge. Table-1 represents a list of CNN architectures that will be used as the Deep Learning model in the Federated Learning environment.

Name	Size in MB	Parameters	Depth
ResNet50	98	25.6M	107
VGG16	528	138.4M	16
InceptionV3	92	23.9M	189
MobileNet	16	4.3M	55
DenseNet121	33	8.1M	242
EfficientNetB7	256	66.7M	438

TABLE I
VARIOUS CNN ARCHITECTURES.

Among these CNN architectures, MobileNet architecture is the smallest one, whereas, VGG16 is the largest architecture. The inclusion of both small and large architectures provides an opportunity to analyze the impact of malicious nodes both on small and big CNN architectures that can memorize or remember various degrees of information.

IV. Milestones and Responsibilities

Our experiment and analysis can be compartmentalized into multiple smaller parts, which can be considered as milestones. Therefore, below we include a list milestones or tasks that need to be completed in order to successfully complete our analysis.

- Firstly, we need to preprocess and divide our dataset into training, validation and testing sections.
- Next, we need to select a Deep Learning architecture and train the model using the training and validation set. Then, test the model using the testing dataset to results as future benchmarks to compare between models.
- Then, we need to initiate the Federated Learning environment by dividing the training and validation data into multiple smaller sets to distribute among the local nodes as their own private data in the experiment.
- Next, we need to create the misclasification labels that will be used by the malicious nodes in their training phase.
- Afterwards, we need to train each CNN architecture using various number of malicious nodes and record the accuracy of the Federated Learning model.
- Similarly, we need to train each CNN architecture using malicious nodes in various parts of the training timeline and record the accuracy of the Federated Learning model.
- Finally, we can analyze the recorded accuracy of the various Federated Learning models to each other and the benchmark model to see how malicious nodes affect the performance of the CNN architectures.

Each member of the project is responsible for data processing, model implementation, model evaluation and report organization. Moreover, as we will be comparing our results on six different machine learning models, each member will concentrate on working with at-least two mentioned models from table-1.

V. Expected Outcome

By conducting our experiments and analysis the results of various trained Deep Learning models, we expect to achieve the following outcomes -

- We will be able to understand how various numbers of malicious nodes affect the performance of the Federated Learning model.
- We will be able to understand how various popular CNN architectures react to malicious nodes in terms of performance.
- We will be able to understand how malicious nodes impact the Federated Learning environment.

- We will be able to compare the impact of malicious nodes on various parts of the training timeline in a Federated Learning environment.

REFERENCES

[1] CIFAR-10 and CIFAR-100 datasets. (n.d.). Retrieved September 26, 2022, from https://www.cs.toronto.edu/ kriz/cifar.html

[2] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman,V. Ivanov, C. Kiddon, J. Konec ˇny‘, S. Mazzocchi, H. B. McMahan et al., "Towards federated learning at scale: System design," arXiv preprint arXiv:1902.01046, 2019.

[3] Jakub Konecný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. CoRR abs/1610.02527 (2016). arXiv:1610.02527 http://arxiv.org/abs/1610. 02527

[4] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. CoRR abs/1610.05492 (2016). arXiv:1610.05492 http://arxiv.org/abs/1610.05492

[5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings et al., "Advances and open problems in federated learning," arXiv preprint arXiv:1912.04977, 2019.

[6] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pages 308–318, 2016.

[7] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC Con- ference on Computer and Communications Security, pages 1322–1333, 2015.

[8] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In International Conference on Artificial Intelligence and Statistics, pages 2938–2948. PMLR, 2020.

[9] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," NeurIPS, 2020

[10] C. Xie, K. Huang, P. Chen, and B. Li, "DBA: distributed backdoor attacks against federated learning," in 8th International Conference on Learning Representations, 2020.

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

[12] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," CoRR, arXiv:1811.12470, 2018.

[13] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020), 2020, pp. 301–316.

[14] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar, "signSGD with majority vote is communication efficient and byzantine fault tolerant," in In Seventh International Conference on Learning Representations (ICLR), 2019.

[15] P. Blanchard, R. Guerraoui, J. Stainer et al., "Machine learning with adversaries: Byzantine tolerant gradient descent," in NeurIPS, 2017, pp. 119–129.