Missouri University of Science and Technology
Scholars' Mine

Computer Science Faculty Research & Creative Works

Computer Science

01 Jan 2022

# Leveraging Spanning Tree to Detect Colluding Attackers in Federated Learning

Priyesh Ranjan

Federico Coro

Ashish Gupta

Sajal K. Das
*Missouri University of Science and Technology*, sdas@mst.edu

Follow this and additional works at: https://scholarsmine.mst.edu/comsci_facwork

 Part of the Computer Sciences Commons

## Recommended Citation

P. Ranjan et al., "Leveraging Spanning Tree to Detect Colluding Attackers in Federated Learning," *INFOCOM WKSHPS 2022 - IEEE Conference on Computer Communications Workshops*, Institute of Electrical and Electronics Engineers, Jan 2022.
The definitive version is available at https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9798077

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Computer Science Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-0926-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798077

# Leveraging Spanning Tree to Detect Colluding Attackers in Federated Learning

Priyesh Ranjan, Federico Corò, Ashish Gupta, and Sajal K. Das. *Fellow, IEEE*
Department of Computer Science, Missouri University of Science and Technology, Rolla, USA
*email: {pr8pf, federico.coro, ashish.gupta, sdas}@mst.edu*

***Abstract*—Federated learning distributes model training among multiple clients who, driven by privacy concerns, perform training using their local data and only share model weights for iterative aggregation on the server. In this work, we explore the threat of collusion attacks from multiple malicious clients who pose targeted attacks (e.g., label flipping) in a federated learning configuration. By leveraging client weights and the correlation among them, we develop a graph-based algorithm to detect malicious clients. Finally, we validate the effectiveness of our algorithm in presence of varying number of attackers on a classification task using a well-known Fashion-MNIST dataset.**

***Index Terms*—Attacker, correlation, federated learning**

## I. Introduction

Federated learning (FL) presents itself as a decentralized form of machine learning in which multiple clients collaboratively train a model without revealing their private data in the presence of a common server. Each client sends a model, trained on its local dataset, to the server for aggregation. Later the aggregate model is sent back to the clients. However, FL is vulnerable to collusion attacks, which leads to a lack of convergence to an optimal model. In the collusion attack, all the malicious clients (attackers) get agree to perform same type of attack on the model. For example, in a label flipping attack, all the attackers flip same target label in their local datasets. The presence of large number of such malicious clients working towards a common goal can overwhelm normal clients and degrade the performance of the model.

In this work, we focus on identifying malicious clients as early as possible. Though the prior works [1]–[3] have also focused on detecting attackers, they require large number of FL rounds, which subsequently offer the attackers a more favorable scenario to influence the model. By exploiting weight correlation among the clients, we propose a graph theoretic approach to detect the attackers at early rounds. Particularly, we leverage maximum spanning tree to distinguish the attackers from the normal clients.

## II. Proposed Solution

In this paper, we focus on label flipping attacks where the malicious clients flip the label of particular (target) samples to another label. For example, in our experimental analysis on the Fashion-MNIST dataset (F-MNIST) [4], all attacking clients flip the labels of digit $'0'$ images to $'1'$. Keeping a practical scenario in mind, we assume that the number of attacking clients is always less than that of normal clients.

Our solution exploits the correlations between the clients' model to identify attacking clients. We create a complete weighted graph in which each vertex represents a client and each weighted edge represents a correlation between the pair of clients. Considering $n$ clients in FL system, let $\mathcal{A} \in \mathbb{R}^{n \times n}$ be a matrix whose each element $\mathcal{A}_{ij}$ is a correlation coefficient between two clients $i$ and $j$, which is computed as

$$\mathcal{A}_{ij} = \begin{cases} \frac{\sum_m \sum_m (W_i - \bar{W}_i)(W_j - \bar{W}_j)}{\sqrt{(\sum_m \sum_m (W_i - \bar{W}_i)^2)\ (\sum_m \sum_m (W_j - \bar{W}_j)^2)}}, \ \forall\ i \neq j \\ 0, \ \forall\ i = j \end{cases}$$

where $W_i$ and $W_j$ are the model weights of $i^{th}$ and $j^{th}$ client respectively, with $\bar{W}_i$ and $\bar{W}_j$ being the mean weights, and $m$ is the size of the feature space of the model. We use the matrix $\mathcal{A}$ as a graph adjacency matrix for the graph.

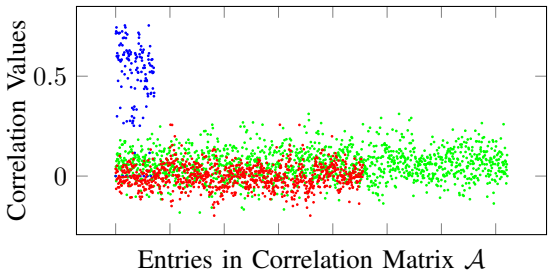


Fig. 1: Correlation values between Attacker-Attacker (blue), Normal-Normal (green) and Attacker-Normal (red).

We observe that the correlations between the attacking clients are higher than the correlations between the normal clients. Additionally, the correlations between an attacker client and a normal client is relatively lower than the correlations between two normal clients as the attacker and normal clients train models on different datasets resulting in a difference in their weight updates which causes correlation values to be lower. This causes the correlations between clients to be in the order: $Attacker - Attacker > Normal - Normal > Attacker - Normal$. In the following we will use this inequality in our algorithm to be able to distinguish the attackers from the normal clients. See Fig. 1 for a graphic representation. Indeed we note that such inequality does not always hold, however, having this assumption is sufficient to differentiate malicious clients from normal ones.

The graph realization thus obtained is used to create a Maximum Spanning Tree (MST), i.e., a selection of $n-1$ edges such that the sum of their weights is maximum (for $n$

978-1-6654-0926-1/22/$31.00 ©2022 IEEE
Authorized licensed use limited to: Missouri University of Science and Technology. Downloaded on February 17,2023 at 22:19:47 UTC from IEEE Xplore. Restrictions apply.

total clients). This tree is composed of two sub-trees, each consisting of only malicious (or normal) clients. In practice, it is achieved by sorting the edges in non-decreasing order of their weights. By removing one edge from the formed MST, we get two disjoint subtrees, however, together they still contain all vertices of the graph.

Since the correlation between attackers is higher than between normal clients, malicious clients will occupy one subtree, and normal clients will occupy the other subtree. While, since the correlation between normal clients and attackers is low, there will be no edge connecting these two subtrees. Finally, since we assume that the number of attacking clients is less than the number of normal ones, the smaller subtree is chosen as the one containing the malicious clients.

**Algorithm 1:** Spanning tree based attacker detection

**Input:** Correlation matrix $\mathcal{A}$
**Output:** Set of attacker clients
$trees \leftarrow \emptyset, i = 0$
$\mathcal{E} \leftarrow$ sorted edges by non-decreasing order
**while** $|trees| < n - 1$ **do**
  **if** $\mathcal{E}[i]$ *does not form cycle in* $trees$ **then**
    $trees \leftarrow trees \cup \mathcal{E}[i]$
  i++
// $min(\cdot)$ finds a tree with minimum cardinality
$attackers \leftarrow min(trees)$
**return** $attackers$

Algorithm 1 summarizes the proposed algorithm. The algorithm's input is the correlation matrix $\mathcal{A}$, and the output is the list of attacking clients for a specific round. After obtaining the list of attacking clients, we exclude their weight updates during the aggregation at the server in FL, thus completely nullifying the effect of the attack on the model.
Finally, given our considerations, the following results holds:

***Theorem** 1:* Given a correlation matrix $\mathcal{A}$, let inequality $Attacker - Attacker > Normal - Normal > Attacker - Normal$ hold for any pair of clients, then Algorithm 1 returns the complete and correct set of malicious clients.

## III. Initial Results

We validate our proposed algorithm on F-MNIST dataset distributed non-uniformly by using Dirichlet distribution with a hyper-parameter $\alpha = 0.9$. As baselines, we use the standard Federated Averaging (FedAvg) [5], and FoolsGold [1]. We train a deep learning model (consisting of 2 convolutional layers followed by 3 fully connected layers) with total 50 clients including the number attackers ranging between $5 - 45\%$. We run the model for 10 local epochs at each client and then aggregate at server for 30 global rounds.

In Table I we compare the ratio of samples misclassified to samples flipped (ASR), and the Earliest Round of Detection of the full set of attacker clients (E.D.) for different percentage of attacker clients (Atk%). We observe that the proposed algorithm maintains a low ASR for higher percentage of attacker clients in the setup as opposed to the FoolsGold and FedAvg methods which show higher ASR for increased number of attacker clients. Moreover, our solution is able to detect the full set of attacker clients much earlier than FoolsGold, while maintaining consistent accuracy, as shown in Fig. 2. Indeed FoolsGold was not able to detect the full set of attacker clients in most of the cases (denoted by $'*'$).

TABLE I: Attack success rate (ASR) and Earliest Round of Detection (E.D.) with varying percentage of attacker clients (Atk(%)).

| Atk (%) | Our algorithm | | FoolsGold | | FedAvg |
|---|---|---|---|---|---|
| | ASR | E.D. | ASR | E.D. | ASR |
| **10** | 3% | 3 | 4.52% | * | 2.5% |
| **15** | 2.67% | 7 | 1.64% | * | 3% |
| **20** | 1.5% | 3 | 1.52% | 22 | 3% |
| **25** | 1.8% | 29 | 1.4% | * | 6.6% |
| **30** | 1% | 18 | 0.84% | * | 2.3% |
| **35** | 0.85% | 3 | 1.16% | * | 19.8% |
| **40** | 0.5% | 12 | 6% | * | 8.5% |
| **45** | 0.67% | 3 | 3.78% | * | 82.32% |

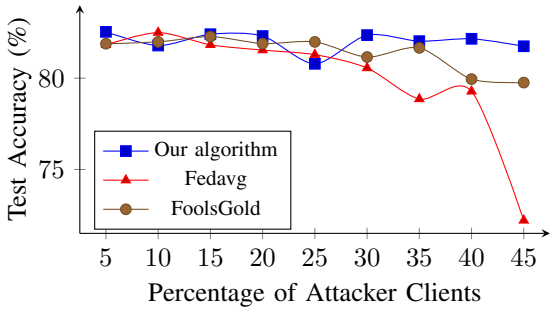


Fig. 2: Accuracy results under varying percentage of attackers

## IV. Conclusion

In this work we proposed the use of graph theoretic approach to detect malicious clients in federated learning settings. We were able to show a reduction in the attack success rate compared to the existing algorithms while maintaining a competitive accuracy. Additionally, our algorithm outperforms FoolsGold by a large extent on early detection (E.D.) metric. As future works we plan to scale up this work to untargeted attackers such as Sign-flipping and random noise along with other forms of targeted attacks like Backdoor or Multi-Label flipping. Along with that, we plan to conduct extensive set of experiments to evaluate our solution on other datasets.

## References

[1] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *23rd Int. Sym. on Research in Attacks, Intrusions and Defenses (RAID)*, pp. 301–316, 2020.
[2] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Int. Conf. on Machine Learning*, pp. 5650–5659, 2018.
[3] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *31st Int. Conf. on Neural Information Processing Systems*, pp. 118–128, 2017.
[4] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[5] 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.

Authorized licensed use limited to: Missouri University of Science and Technology. Downloaded on February 17,2023 at 22:19:47 UTC from IEEE Xplore. Restrictions apply.