How Much Noise is Enough: On Privacy, Security, and Accuracy Trade-Offs in Differentially Private Federated Learning

Adhishree Kathikar
Indiana University
Bloomington, Indiana, USA

Kyle Chard (advisor)
University of Chicago
Department of Computer Science
Chicago, Illinois, USA

Nathaniel Hudson (advisor)
University of Chicago
Department of Computer Science
Chicago, Illinois, USA

Abstract

Centralized machine learning techniques have caused privacy concerns for users. Federated Learning (FL) mitigates this as a decentralized training system where no raw data are communicated across the network to a centralized server. Instead, the machine learning model is trained locally on each device and they send the locally-trained model weights to a central server to aggregate. However, there are critical challenges with FL. Security issues plague FL, such as model poisoning via label flipping. Additionally, there even exist privacy concerns via data leakage by reconstruction of weights. In this work, we apply differential privacy (which adds noise to the model weights before sending across the network) as an added privacy measure to protect sensitive data from being reconstructed. Through this research, we study the effects of differential privacy on FL with respect to security and privacy trade-offs.

Keywords: Federated Learning, Privacy-Preserving Machine Learning, Internet-of-Things

ACM Reference Format:
Adhishree Kathikar, Kyle Chard (advisor), and Nathaniel Hudson (advisor). 2026. How Much Noise is Enough: On Privacy, Security, and Accuracy Trade-Offs in Differentially Private Federated Learning . In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Federated Learning (FL) [1, 3] is a paradigm for training *Machine Learning* (ML) models (e.g., neural networks) on decentralized data. These data could be scattered across a decentralized network of devices. Some examples include the Internet-of-Things, edge computing systems, and sensor networks. FL pushes the task of training ML models to the end devices where the data are located. An aggregation server will periodically collect the model weights given by local training on these devices and aggregate them to "learn" over all the data in the system without communicating the data over the network. This introduces an immediate layer of privacy. However, there are still significant concerns related to privacy including the risk of malicious servers, model or data poisoning, or possible reconstruction of data based on shared model weights. We explore the use of differential privacy methods in FL to reduce the ability to infer data from weights. Specifically, we study trade-offs between the amount of noise added to models and the accuracy of the global model.

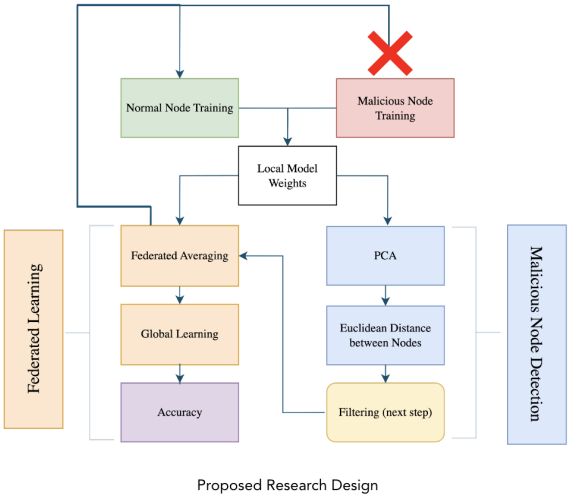


Figure 1. The proposed research design. This design explains federated learning, the analysis method to determine malicious nodes, and our future plans of creating a classifier to filter out malicious nodes from global training.

2 Background

While FL can increase privacy compared to centralized machine learning training, there remain issues with the privacy and security of the model. For example, backdoor attacks, communication bottlenecks, malicious servers, and both data and model poisoning. Data poisoning can involve feeding bad data to a local model, while model poisoning can involve data label flipping (e.g., making a model classify all cats as

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
2026. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

frogs). Finally, FL model weights could potentially be reconstructed into data, which could lead to private data being leaked.

Differential privacy is a mathematical approach for ensuring the privacy of individuals in datasets by applying algorithms in which it should not be possible to reconstruct data. One common approach is to add noise to individual data while not altering the statistical properties of interest. While differential privacy has the benefit of being able to improve the privacy of model weights and prevent data reconstruction [2, 4], it may also make it more difficult to identify malicious nodes.

3 Research Questions

For this work, we consider an FL system where we assume that it has a detection system setup to identify which nodes are malicious based solely on the model weights. We are then interested in studying the implicit trade-offs for how privacy-preserving methods (namely differential privacy) affect the viability of such a system for improving the security of the FL system. More specifically, we are motivated by the following research questions: *(i)* How does differential privacy affect FL system security? *(ii)* When is noise too large to reliably identify malicious nodes? *(iii)* How can we balance these implicit trade-offs?

4 Methods

To address the posed research questions, we developed a research framework to assess the tradeoffs between security and privacy. First, we trained a FL model using common benchmark datasets (e.g., MNIST and CIFAR10). We then simulated malicious nodes in the system to observe the impact of malicious nodes on security and accuracy. We simulated malicious nodes by using data label flipping, and flipped 3 labels in the dataset. We observed the loss of accuracy if there were no malicious nodes, 20% malicious nodes, 50% malicious nodes, and 100% malicious nodes.

For the rest of the experiments, we consider 20% of the nodes to be malicious (i.e., employ label-flipping attacks), and we used principal component analysis to determine which nodes were malicious by reducing the dimensionality of the model parameters to 2 and plotting them on a scatter plot.

In this work, we apply noise sampled from a Gaussian distribution with a mean of 0 and the standard deviation of ϵ. This is done to randomize parts of the weights to make it harder to reconstruct the data from directly from the weights [2, 4]. Our hypothesis was that by adding noise to the model weights, the weights would move together and be indiscernible if weights were from a malicious node or a normal node. For this work, we run experiments with ϵ set to be 0.01, 0.025, 0.05, and 0.1.

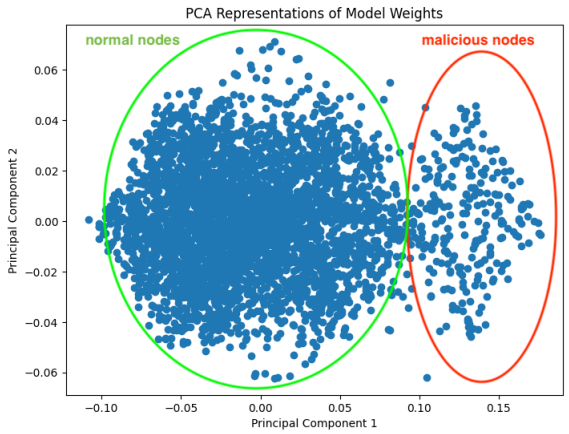


Figure 2. Principal components of locally-trained model parameters. All the model parameters are initially padded with zeros to give them a shared dimensionality before being reduced to 2 components using PCA. The left clump of points are parameters sent from normal nodes; the right clump of points are parameters sent from malicious nodes.

5 Results

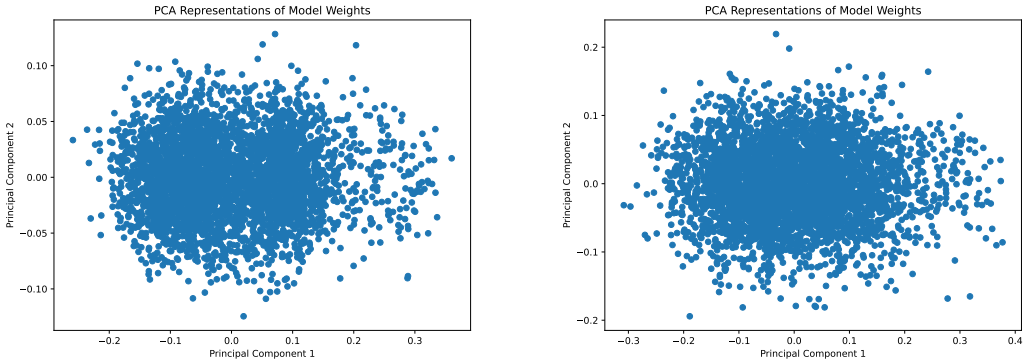
We first trained a FL model using the MNIST dataset and the CIFAR-10 dataset using ten nodes. We simulated malicious nodes, and saw the decrease of accuracy based on the percentage of malicious nodes. With no malicious nodes we observed an average accuracy of 89.5%. With 50% malicious nodes, accuracy decreased to 40%. Finally, with 100% malicious nodes, we were able to see an accuracy of 19%.

Using PCA to reduce dimensionality, we observed the endpoints that were malicious through using a scatter-plot. and observing a malicious node cluster and a regular node cluster. When we began adding a Gaussian distribution to the weights, we noticed the malicious node cluster and the regular node cluster moved closer towards each other. If using a classification system, this could confuse the classification system, leading to inaccuracies in the model., and the model weights looked like one cluster, rather than a malicious node cluster and a normal node cluster.

We found that at 0.025, the node clusters began to move closer to each other, and at 0.05 it was difficult to discern the malicious nodes when using 20% nodes and 3 pairs of flipped labels.

6 Conclusions & Future Directions

As the decentralized systems continue to grow, there are still significant concerns related to privacy including the risk of malicious servers, model or data poisoning, or possible reconstruction of data based on shared model weights. As we explore methods to increase privacy for users, we must also look at the accuracy of models. Our next steps will primarily



(a) $\epsilon = 0.025$ **(b)** $\epsilon = 0.05$

Figure 3. Principal components of locally-trained model parameters when applying differential privacy with noise sampled from a Gaussian distribution with a mean of 0.0 and a standard deviation of ϵ. Compared to the components plotted in Fig. 2, we find here that the added noise makes distinguishing the malicious weights from the normal weights much more difficult, especially as ϵ gets larger.

focus on looking at the effects privacy preservation through differential privacy has on the accuracy of models. We then develop a classifying system that will detect malicious nodes and filter them out of federated averaging, thus mitigating the attack type of malicious nodes on federated learning systems.

References

[1] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. *Proceedings of machine learning and systems* 1 (2019), 374–388.

[2] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems* 33 (2020), 16937–16947.

[3] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[4] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in neural information processing systems* 32 (2019).