Using Generative Adversarial Networks to Detect Model Poisoning Attacks in Federated Machine Learning

Usama Zafar[1,*], Salman Toor[1,*], and André Teixeira[1,*]

[1]Department of Information Technology, Uppsala University, Uppsala, Sweden
[*]usama.zafar@it.uu.se, salman.toor@it.uu.se, andre.teixeira@it.uu.se

ABSTRACT

Federated machine learning (FedML) is a promising approach that enables multiple participants to collaboratively train a shared model without requiring them to share their data. However, the decentralized nature of FedML also makes it vulnerable to various security threats. Poison attack, a critical challenge in federated machine learning, is one such threat where malicious participants inject poisoned updates into the training process to undermine the model's performance or to induce incorrect predictions. Detecting and mitigating the effects of poison attacks is crucial to ensure reliability of the trained model.

A key challenge in detecting poison attacks is that the poisoned updates can be carefully crafted to circumvent any defense mechanism. Lack of validation data, that can be used to authenticate updates, exacerbates this issue further. To overcome this challenge, we propose using Generative Adversarial Networks (GANs) to detect poison attacks in FedML. GANs are powerful models that can learn to generate synthetic data that is similar to the real data distribution. By evaluating each update from a participant on the synthetic data, we can detect if any of the participants have injected poisoned model into the training process. The proposed solution involves using the latest GAN model available at the central server to generate synthetic data. This synthetic data is then distributed to all the participating nodes, and each node uses it to attest or authenticate their updates before sending them to the central server. This ensures that the updates are consistent with the synthetic data, and any updates that deviate significantly from the synthetic data distribution are flagged as potentially malicious.

Our proposed solution has several advantages. First, it does not require any changes to the existing FedML infrastructure. Also, it can detect both direct and indirect poison attacks, where the attacker manipulates the training process indirectly by influencing the updates of other participants. We believe that our proposed solution can significantly improve the security of FedML and help mitigate the effects of model poison attacks.

In conclusion, our proposed solution using GANs to detect poison attacks in FedML is a promising approach that can significantly improve the security and robustness of the training process. With the increasing adoption of FedML in various applications, it is critical to address the security challenges and develop effective countermeasures to protect the integrity of the trained model.