FedCC: Federated Learning with Consensus Confirmation for Byzantine Attack Resistance (Student Abstract)

Woocheol Kim and Hyuk Lim
Gwangju Institute of Science and Technology (GIST)
Gwangju 61005, Republic of Korea
{woocheolkim, hlim}@gist.ac.kr

Abstract

In federated learning (FL), a server determines a global learning model by aggregating the local learning models of clients, and the determined global model is broadcast to all the clients. However, the global learning model can significantly deteriorate if a Byzantine attacker transmits malicious learning models trained with incorrectly labeled data. We propose a Byzantine-robust FL algorithm that, by employing a consensus confirmation method, can reduce the success probability of Byzantine attacks. After aggregating the local models from clients, the proposed FL server validates the global model candidate by sending the global model candidate to a set of randomly selected FL clients and asking them to perform local validation with their local data. If most of the validation is positive, the global model is confirmed and broadcast to all the clients. We compare the performance of the proposed FL against Byzantine attacks with that of existing FL algorithms analytically and empirically.

FL with Consensus Confirmation

We propose federated learning with consensus confirmation (FedCC) to make the FL system robust to Byzantine attacks. The FL server performs a consensus-based validation procedure before it broadcasts the global model to investigate whether or not a newly aggregated global model is beneficial to most clients. First, a set of clients are selected randomly and asked to perform the consensus-based validation. Then, if the majority of the selected clients agree, the new global model is broadcast to the entire FL network.

Figure 1 depicts the proposed FL procedure in a round $r$. To begin, the FL server chooses at random from $m$ clients the uploaded local learning results of $n$ clients who train their own data. Secondly, the server aggregates the selected learning results from $n$ clients to decide a candidate global weight $\tilde{w}_r$. Thirdly, the server randomly selects $l$ clients for a consensus confirmation out of $m$ clients and transmits a candidate global weight $\tilde{w}_r$ to them. The $l$ clients compare test accuracies of models updated with the $\tilde{w}_r$ and a previous weight $w_{r-1}$ using their own training data, then the consensus clients transmit the confirmation results to the server. Fourthly, the server confirms and decides a global weight $w_r$. If the majority of the consensus clients submit the comparing results that the $w_{r-1}$ has higher test accuracy than the $\tilde{w}_r$, the server defines the $w_r$ to the $w_{r-1}$. Fifthly, the server broadcasts the global weight $w_r$ to clients, and the clients load the global weight on their own model.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
The proposed FedCC’s consensus-based validation algorithm reduces the impact of Byzantine attackers in clients on global learning performance degradation. Suppose there is at least one Byzantine attacker in a training group. In that case, a global weight easily becomes polluted because the global weight is obtained by aggregating results from every client in the training group. In the case of the aggregation rule averaging client results, when there are $k$ Byzantine attackers in $m$ clients and the server selects $n$ clients for training, the Byzantine attack success probability is given by

$$P_{\text{attack}} = 1 - \frac{C(m-k,n)}{C(m,n)},$$

(1)

where $C(\cdot)$ is a combination function. In case that uses the proposed consensus-based validation algorithm selecting $l$ consensus clients, the Byzantine attack success probability is given by the following equation

$$P_{\text{attack,CC}} = \sum_{x=\left\lfloor \frac{l}{2} \right\rfloor}^{l} \frac{C(k,x) \cdot C(m-k,l-x)}{C(m,l)}.$$

(2)

In the proposed FedCC algorithm, the probability that global weight becomes polluted is given by $P_{\text{FedCC}} = (1 - P_{\text{attack}}) \cdot P_{\text{attack,CC}} + P_{\text{attack}} \cdot P_{\text{attack,CC}} = P_{\text{attack,CC}}$. When there is at least one Byzantine attacker in the training group and more than half of the attackers in the consensus group, the attackers succeed in transmitting the positive consensus confirmation results causing the broadcasting of the polluted global weight. Even in case that there are no attackers in the training group, if there are more than half of the attackers in the consensus group, the attackers can transmit the negative consensus confirmation results causing the convergence of global weight to be delayed. Figure 2 shows the Byzantine attack success probability of normal FL and proposed FL using Python simulations. For each simulation case, we set $m = 100$, $n = 10$, $k = 20$, $l = 10$. The result values are the average of 5 times iterations. Figure 3 shows the test accuracy for FedCC, FedAVG, and Median rules of rounds in normal and attack situations. As shown in Figure 3(a), if no attack exists, the proposed FedCC shows similar test accuracy to the FedAVG and higher performance than the Median rule. However, when there are Byzantine attackers in clients, as shown in Figure 3(b), the test accuracy of the FedCC is more stable and higher than other algorithms.

In conclusion, we have proposed a FedCC scheme for Byzantine attack resistance and demonstrated that the proposed FedCC outperforms previous FL algorithms in terms of attack resistance and learning performance.

Acknowledgments

This work was supported by IITP grant funded by the Korea government (MSIT) (No. 2021-0-00379, Privacy risk analysis and response technology development for AI systems)

References

