FLAIR: Defense against Model Poisoning Attack in Federated Learning

Author: Anonymous Author (Paper under double-blind review)

Requirement - PyTorch 1.8.1

The code when run downloads the dataset (MNIST and CIFAR-10) on its own, however, the MNIST dataset folder has been uploaded for convenience

Default run - "python main.py"

-- Runs FEDSGD aggregation with 10 benign clients on MNIST for 20 iterations

Example run - Running benign case followed by attack, followed by FLAIR's defense on MNIST

1) python main.py --gpu 0 --exp benign\_fedsgd

2) python main.py --gpu 0 --byz\_type full\_trim --exp fulltrim\_fedsgd

3) python main.py --gpu 0 --byz\_type full\_trim --aggregation flair --exp fulltrim\_flair

The "main.py" file calls "aggregation.py" and "attack.py" to run the simulation of federated learning with the following command line arguments:

--dataset : mnist/ cifar10

--bias: controls the degree of non-iidness, 0.5 by default

--net : dnn for mnist, resnet18 for cifar10

--batch\_size : 32 for mnist, 128 for cifar-10

--lr : learning rate - 0.01 for mnist, 0.1 for cifar10

--nworkers : 100 for mnist, 10 for cifar10

--nepochs : Number of training iterations, 500 for MNIST, 2000 for CIFAR-10

--gpu : 0 if CUDA is available, -1 otherwise

--nbyz : actual number of malicious workers (20% of nworkers)

--byz\_type: benign/ full\_trim/ full\_krum/ adaptive\_trim/ adaptive\_krum/ shej\_attack/ shej\_agnostic

--aggregation : flair/ fedsgd/ krum/ trim (trimmed mean)/ median/ faba/ foolsgold/ fltrust

--cmax : workers the aggregator expects to be malicious, set to be equal to nbyz

--decay : decay parameter for FLAIR, default value = 0.99

--exp : name of the experiment given by the user

The code outputs a numpy vector - '<exp>\_test\_acc.npy' with the test accuracy values in every iteration, and '<exp>\_model.pth' as the PyTorch model.

When the aggregation is FLAIR, two additional files are saved as outputs -

1)<exp>\_FS.npy - Every row of this array represents one iteration, where every column is the flip-score of the respective client

Clients 0-19 are malicious and 20-99 are benign. Their behavior can be analyzed using this output file

2)<exp>\_susp.npy - This contains the suspicion score of every client in a given iteration that was used to compute the reputation weights for aggregation in that iteration

For FoolsGold, FABA, and FLTrust, we output the corresponding client\_list files that contains the reputation weight computed by those aggregation rules