

This is Pretom Roy Ovi, PhD student at Information Systems Department from University of Maryland, Baltimore County. I'm working in the domain of federated learning under the supervision of Dr. Aryya Gangopadhyay, who is also a co-author of this work. And this work is in collaboration with Dr. Robert Erbacher and Dr. Carl Busart from Adelphi Army Research Laboratory.

Now I am going to present my paper, the Confidential Federated Learning to tackle Labeled Flipped Data Poisoning Attacks. So first I will tell you what is federated learning and why we need it. So in traditional machine learning, it is required to gather all the data to the central cloud to train the model. So it raises privacy concerns because of uploading confidential data. At the same time, there is always a chance of confidential data breach during this upload.

So to ensure the data privacy and data confidentiality, federated learning is proposed by Google in 2017. So in federated learning, training data remains locally on edge devices. So explicit data sharing is not necessary here. Only weight or gradient updates are shared.

So this is the very basic architecture of federated learning. So there are two components in any federated learning system. One is server, which is known as model owner. And other component is the local worker or clients. They are known as the data owner.

And according to this framework, the server determines the type of application the user will learn and how the user will be trained. And based on the application type, for example, if you use an image classification problem, the server will build the CNN model. And that model is known as global model. And this global model is distributed to the selected participants.

After that, each client receives the model from the server, uses its own data or local data to train the model. And after each round of local training, updated weights or updated gradients are sent back to the server. And finally, the server will aggregate the update and update the global model. And this process

cess is repeated until a desirable training accuracy is achieved. So this is the architecture of any federated learning systems.

So in terms of adversarial attacks and federated learning, the attack surface of federated learning have grown because of this architecture of distributed training. And because of its distributed training, federated learning has limited control over the local data and the participating plant. And that's why federated learning is vulnerable to different type of attacks like gradient inversion attacks, model poisoning attacks, and data poisoning attacks.

And in this figure, you can see that attacks are phases of different attack methods in federated learning architecture. And today, I will talk about the data poisoning data. So basically, federated learning has been developed to protect the data confidentiality. And this concern motivate to keep the raw data locally on edge devices.

Since the server has no access to the training data, there is no authority to penetrate the data. An attacker take this advantage to poison the local data set and degrade the performance of the global model. So in data poisoning attack, the attackers are compromised workers, manipulate the local data set by adding poison or adding noise to the instances or change the existing instances in an adversarial manner with an intention to poison the model and degrade its performance.

And label flipping is a feasible strategy and easiest way to initiate the data poisoning attack. Because it allows even the non-expert attacker, who doesn't have any knowledge regarding the model parameter, model architecture, or any other federated learning system, even that attacker is also able to flip the label of the instances. And recently, several studies have been investigated and addressing the data poisoning attacks in federated learning scenario. But those studies mainly focus on the detection path, like how can we detect the worker under data poisoning attack?

So research gate lies in that direction of developing a prevention strategy, like how can we prevent such kind of attack? So it requires further research to make the federated learning system more resistant. So as a contribution,

at first, we showcased the vulnerability of federated learning system to attack and effect of such attacks on the performance. And we found out that such label flipping or data poisoning attacks can be targeted, meaning that they have a significant negative impact on the subset of classes which are under attack, but there is no impact on the remaining classes.

And finally, we propose a confident federated learning framework to prevent the attacks by validating the class label of the data on the worker side. And our approach estimates the label noise probability and the confidence threshold to potentially identify the mislabeled samples on the worker set. And we also demonstrate the potential of our proposed approach on MNIST, Fashion MNIST, and CIFAR-10 data set.

Now I will tell you how we have initiated the label flipping attack. To explore the impact of label flipping data, we randomly choose end of total participant as malicious. And label flipping attack denoted by source to target indicates that ground truth label of source plus samples have been flipped with the target class label. And we exclude the label flipping attacking scenario in the federated learning for MNIST, Fashion MNIST, and CIFAR-10 data set by flipping the labels of source class to another specific target class and which we denoted by source to target pairings.

Now I'll show you the result of the attack effect on the performance of federated learning. For example, if we train a machine learning model on a flipped data set, where some samples has been flipped to another target class, so after the training, during the prediction, some samples of the source class will also be predicted as a target class. So the source class will have some false negative, which will impact its recall. And the target class will have some false positive, which will affect its precision.

And in the figure, we have illustrated the degradation of source class recall and the target class precision with the increasing percentage of malicious data, malicious participants from 0% to 40%. And whenever the percentage of malicious participants reaches to 40%, the recall drops from 0.99 to 0.73. And precision drops from 0.98 to 0.78 for MNIST data. And a similar trend is also observed for the Fashion MNIST data set as well.

Here we illustrated the global model's performance drop with the increasing percentage of malicious worker for the CIFAR-10 data set. So higher the percentage of malicious worker, the more the performance drop is. But the interesting thing is that when the percentage of malicious data reaches to 50 %, precision and recall of the source and target plus drops significantly. Whereas the overall accuracy of the global model remains approximately same. And this result indicates that targeted nature of label flipping attack, where the attack degrades the global model's potential in predicting the source and target plus, while it has no impact on the remaining classes.

So to deal with this issue and prevent the attack, here propose a confidential federated learning framework to prevent the label flipped attack on the worker side. And according to this framework, the server will send the global model to each worker. Each worker will receive the model and execute stratified verification to validate the class label. And in stratified verification, the potentially mislabeled samples will be detected and excluded from the training set.

After that, each worker will initiate the local training. And after each round of local training, the updated weights or updated gradients will sent back to the server. And finally, the server will aggregate the update and update the global model. So this is the architecture.

To prevent this data poisoning attacks, now the question is, how can we execute the stratified verification on the worker side to penetrate the class label? To do so, we will require two things. First thing is the noise label, which is already given as a ground truth labels for each label training sample. And second one is out-of-sample predicted probabilities for each local training sample. An out-of-sample prediction refer to the model's prediction made on the data point, those are not shown to the model during the training.

So now the question is, how can we compute the out-of-sample predicted probabilities for the training set? To do, so we have designed K-fold cross validation in such a way that we generate out-of-sample predicted probabilities for every data point in the local training set.

And according to our design K-fold cross validation, K independent copies of model will be used. And for each model copy, one fold is held out from training. And this held out data can be considered as a validation set.

So each copy has a different validation set. So in this process, every training sample is considered once as a validation sample. And we are getting the out-of-sample predicted probabilities for every data point without our fitting them. And in this process, we recommend to use stratified cross-validation to ensure balanced Laplace distribution across different forms.

So we already have our required two things. The first one is the noise label. And second one is the out-of-sample prediction. Our next target is to identify the potentially mislabeled samples. And to do so, first thing we will do is to find the threshold for each class, which is shown in equation 1. And according to equation 1, we will calculate the out-of-sample predicted probabilities for all the samples of given label j and take the mean. And this is the threshold or confidence threshold for class j .

For an example, you have a bunch of images of cat and dog. So take all the images, which is labeled as dog and take the predicted probability for all those images labeled as dog, and take this average. And this value is the threshold value for class dog.

Secondly, now we will find the incorrectly labeled sample utilizing this confidence threshold, which is shown in equation 2. So let's consider the sample x , and calculate what is the probability of that sample for belonging to class dog, given that this sample is already in a different class, for set is the cat class. So you know this sample is labeled as cat. And you want to find out what is the probability of that sample for belonging to different class or dog class.

And if this value is higher than the threshold, it means that the value is not only high. It's even higher than the threshold of a different class. So there is

a good reasonable notion that this sample might be dog, even if it is labeled as cat.

So it has error in labeling. So every client will do the stratified verification on the worker side to identify the potentially mislabeled samples. And finally, those detected samples will be excluded from the training set.

So to evaluate the effectiveness of our proposed approach, we conducted experiments on a few data set, where we deliberately flipped the label of some samples. For an example, for MNIST data set, some samples or some images of digit 0 and digit 2 has been flipped to class label 1. And in this regard, you can see that our proposed approach was successful in detecting the label flipped samples for the MNIST data set.

Even for the Fashion MNIST data set, some ankle boot images, which is labeled as 9 and t-shirt or top classed is assigned as label 0. And here, we flipped some samples of ankle boot to the t-shirt class, which is labeled as 0. And in the bottom figure, here we show the potentiality of our proposed approach in detecting the flipped samples.

So overall, our experimental results suggested that our proposed approach successfully detected the flipped sample, with an accuracy of 88% for MNIST and 86% for Fashion MNIST, although there is a small number of false positive around 5%. So to conclude, our proposed framework is robust against the data poisoning attacks in detecting the mislabeled samples. And we also validated the robustness of our proposed framework on several data sets.

And one of the inherent limitation of our work is that if incorrectly labeled samples or flipped samples are majority, or entire local data set is poisonous, on that case our method will fail. Because our prevention method can identify the mislabeled samples based on the learning from truly labeled data. So if any local worker data set had not match of truly labeled data, on this case, our prevention method will not work.

So these are the references I have used in this presentation. And this work is supported by US Army grant. Thank you.