MSc Dissertation proposal: Defending against Targeted Poisoning Attacks in Federated Learning

Background Knowledge

The Federated scenario of Machine Learning is a very useful way to train large models without the need for powerful resources, as the users themselves contribute to the training of the model. However, along with the ability to train, we pass to the users a great amount of responsibility and freedom to train the model with their own, private data that never leaves their device. Malicious users can take advantage of that privilege and try to trick our model to behave in an unwanted way by injecting faulty data.

Threat model

An adversarial user can then launch a poisoning attack against our system aiming to harm it, by causing it to predict wrong values in its testing phase. This wrong prediction behavior can be either random or targeted. In the random setting the user tries to harm the model by passing random labels for the data that they use for training. In the obviously more interesting, targeted setting, the user tries to trick the model to misclassify a certain class as a different one. For example, while training with the MNIST dataset, the attacker may want to classify every picture of the digit 3 as an 8, while trying to maintain the overall accuracy of the model. This is the case that we are going to deal with in this dissertation, as it is both more sophisticated and harder to detect and defend against.

We consider that a certain percentage of the users that contribute to the training of the model have malicious intent, thus wanting to poison our system with targeted data in order to cause it to misclassify instances. Each attacker fully controls the device and is able to train the model in whichever amount of data they choose to, while keeping them secret, i.e., never leaving the user's device.

Problem Definition

In the traditional Machine Learning setting it would be easier to detect such attacks as we can access all the data that is used for training, thus catch such anomalies before the training phase. However, Federated Learning is meant to preserve the users' privacy, thus the central authority must not have any kind of access to the dataset that each user utilizes to locally train the model.

Thus, the question that arises from this setting is: "How can we defend against an attacker that tries to inject faulty data into our model, if we never look at the data?". This is the problem that we will attempt to mitigate throughout this dissertation.

Idea behind solution

As we mentioned, the data used for training can never leave the device and thus be accessed by the aggregator. However, after training the model we have some metadata, such as the gradients that are sent back to the sever, as well as metrics like the training loss. We assume that the training loss for users that act maliciously will behave differently than the honest ones, thus by aggregating this information we will be able to detect them and eliminate them from contributing to the training phase of the model. However, the loss itself remains a private aspect that can be used to reverse-engineer information about the data used for training. Thus, we must find a privacy-enhancing way to exchange that piece of information with the central authority.

Solution proposed

The solution proposed to preserve the privacy of the user contributing to the training of the model is to use noise generated from Local Differential Privacy protocols in order to anonymize the loss that will be sent to the user. Then, the aggregator can gather the loss from the users asked to contribute to the training, fit a normal distribution on those points and try to detect extreme values (e.x. outliers, values that follow an abnormal distribution), based on dedicated statistical measures. This type of distribution is generally aplicable to a large set of observations and therefore it is safe to assume that it will be fitted on benine values. Moreover, the choice of this type of distribution is actually handy, since it is also used in LDP, thus noise added from that protocol will not make it harder for us to distinguish abnormal values, while at the same time allowing users to protect their privacy.

When the data is placed in the distribution, the aggregator can decide on a confidence level (80%, 95% etc.), and only the users whose values are in this interval will then be used for training. We argue that this way, users with malicious data will produce larger than expected losses, and thus will be detected and then their gradients will not be used in the specific round.

Solution Feasibility

Unlike other, expensive to implement, solutions, we argue that the proposed one is easy and cheap, as it does not require any more communication with the central authority: the user will send their LDP enhanced loss along with their updated gradients. Moreover, it is scalable as with more users, the accuracy of spotting the malicious users will probably increase. Another interesting approach is that the confidence interval could be presented as a hyperparameter of the model, thus the creator will have his saying on the accuracy-security tradeoff created by our threat model.