Homework III

Security and Privacy of Machine Learning (NWI-IMC069)

Deadline: 06/06/25 23:59

For this homework assignment, you will explore backdoor attacks and defenses for Federated Learning (FL). Specifically, you will implement a backdoor attack in FL using Blend trigger. Additionally, you will code a Distributed Backdoor Attack (DB) and compare these attacks. You will also perform a Scaling attack.

For all questions you will make use of the CIFAR-10 dataset. This time you do not need to train a clean model. Instead you will make use of the pre-trained ResNet18Light model, which you also used during the tutorial. All other settings for training will be explained within the corresponding question.

You are expected to present your code and explanations within a Jupyter Notebook, which will serve as your report. The notebook should describe what is done, why it is done that way, what the results are, and what they mean. You should include a thorough discussion of the results—not only what is observed, but also why it occurs. We strongly encourage you to include tables and/or figures to present the results. We also encourage you to make use of the markdown possibilities within Jupyter notebook to clearly mention which question you are answering.

Please submit notebooks that have been executed and saved with their results, and include descriptive yet concise comments in your code. You must provide clear instructions on how to run the code, as any code lacking instructions or that does not run out of the box will be considered incorrect.

Notes:

1. Mention the source of your code, even if the code is your own design or taken from the tutorial notebooks. We do not mind you using other sources as your inspiration for the code, but we do want to see references.

2. Do copy the (sub)questions you are answering to your report and notebook for more clarity. Please copy the question and write the answer below it.

1. **Backdoor Attack in Federated Learning (5 points)**

   (a) *(2 points)* Implement a **source-agnostic backdoor attack in FL using the Blend attack with the Hello Kitty image (attached file)**. Below you find a list of parameters/settings to use for this attack. Limit your implementation to these values. Use a total of five clients, where two are malicious. With *local training epochs* we mean the number of epochs each client should use for local training before sharing weights. After each global aggregation round, compute the ASR and aggregated model's task accuracy. Save these values for later use as you will need them again in the next question. After the final global aggregation round, when training has finished, plot the ASR values (y-axis) and the aggregated model's task accuracy values (also y-axis) versus round number (x-axis). Either make two plots, one for ASR and one for accuracy, or combine the results in one. The following parameters/settings should be used:

   - **Model:** Load the pre-trained ResNet18Light model from the Federated Learning tutorial.
   - **Number of Total Clients**: 5
   - **Number of Malicious Clients**: 2
   - **Trigger:** Hello Kitty image (attached file).
   - **Poisoning Rate:** Every malicious client should use a poisoning rate of 50% of the local dataset.
   - **Global Aggregation Rounds:** 5.
   - **Local Training Epochs:** 2.
   - **Backdoor Target Class:** 3 (cat).
   - **Number Selected Clients Per Round:** 5.
   - **Aggregation Method:** FedAvg.

   (b) *(2 points)* In question 1(a) we asked you to perform Federated Learning using the FedAvg method for aggregation. Now using the same settings, perform federated learning by making use of the **Krum** aggregation method.[1,2] Again, for after each global aggregation round, compute the ASR and the aggregated model's task accuracy. After the final global aggregation round, when training has finished, plot the ASR values (y-axis) and the aggregated model's task accuracy values (also y-axis) versus round number (x-axis). Either make two plots, one for ASR and one for accuracy, or combine the results in one. However, this time also plot the values you saved from question 1(a) in the same plot to make it easier to compare both aggregations methods.

   (c) *(1 point)* First, explain the (theoretical/mathematical) difference between the two aggregation methods used in this assignments (FedAvg and Krum). Now compare your results from questions 1(a) and 1(b) and share your conclusions on how both aggregation methods affect the ASR and accuracy of the attack.

[1] Li et al. (2021) "Byzantine-robust Federated Learning through Spatial-temporal Analysis of Local Model Updates."

[2] Blanchard et al. (2017) "Machine Learning with Adversaris: Byzantine Tolerant Gradient Descent."

2. **Distributed Backdoor Attack (3 points)**

(a) *(2 points)* Implement the Distributed Backdoor Attack (DBA).[3] The network of clients is composed of two malicious clients and three benign clients (five in total). Use the Blend attack where the two malicious clients split the Blend trigger (Hello Kitty image) into two parts (top-bottom). Specifically, one malicious client adds the trigger to the top of the clean images, while the other client adds the trigger to the bottom of the clean images. After each global aggregation round, compute the ASR and aggregated model's task accuracy. Finally, after the final global aggregation round, when training has finished, plot the ASR values (y-axis) and the aggregated model's task accuracy values (also y-axis) versus round number (x-axis). Either make two plots, one for ASR and one for accuracy, or combine the results in one. Again, this time plot the values you saved from question 1(a) in the same plot to enable easy comparison. Use the following parameters/settings for your attack:

- **Model:** Load the pre-trained ResNet18Light model from the Federated Learning tutorial.
- **Number of Total Clients**: 5
- **Number of Malicious Clients**: 2
- **Trigger:** Hello Kitty image (attached file).
- **Poisoning Rate:** Every malicious client should use a poisoning rate of 50% of the local dataset.
- **Global Aggregation Rounds:** 5.
- **Local Training Epochs:** 2.
- **Backdoor Target Class:** 3 (cat).
- **Number Selected Clients Per Round:** 5 (all).
- **Aggregation Method:** FedAvg.

(b) *(1 point)* Compare your results from 2(a) with those from question 1(a). How does DBA perform compared to the previous FL attack setting? Which FL attack setup performs best and why do you think this is the case? Do you think this is a fair comparison, yes or no and why? Share your conclusions.

Note: In DBA, during the training phase, the trigger pattern is split between the clients, but in testing time, you should test the attack (i.e., ASR) with the complete original trigger and not tear it apart! Once FL training is finished, we assume just one global model in test time and no clients and servers.

[3] Xie et al. "Dba: Distributed backdoor attacks against federated learning." ICLR 2020.

3. **Scaling Attack (2 points)**

For this part you will perform the **scaling attack**[4] as explained in the lecture and tutorial.

(a) *(1 point)* Take exactly the same settings as question 1(a) but limit it to just 1 malicious client. Use the blend attack and at each round let the malicious client apply the scale update with a scaling factor of $\gamma = \frac{n}{\mu}$. Here n is the number of clients and μ is the learning rate which is also the number of malicious clients and thus now set to 1. After each global aggregation round, compute the ASR and aggregated model's task accuracy. Finally, after the final global aggregation round, when training has finished, plot the ASR values (y-axis) and the aggregated model's task accuracy values (also y-axis) versus round number (x-axis). Either make two plots, one for ASR and one for accuracy, or combine the results in one. Also, for the final round, report a figure similar to Figure 1. So plot the L2-norms of all clients (including the scaled norms), to show the effect of the scaling attack.

(b) *(1 point)* What defense could you implement to prevent the aggregation of scaled poisoned model's weights in the federated learning setting? Please mention the defense name, how it works and why it is effective.

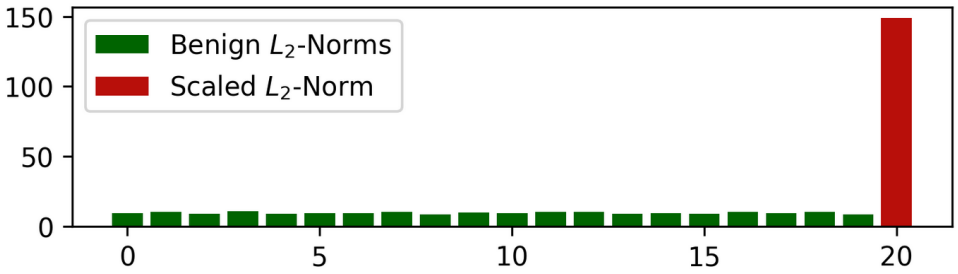

Figure 1: L_2-norm of benign and malicious models. The x-axis is the client index and the y-axis is the L_2-Norm. This image is taken from the tutorial notebook.

[4] Bagdasaryan et al. (2020) "How To Backdoor Federated Learning"