ML for CyberSecurity
Lab 2 Report
Mahati Madhira VSK (mm12032)

Link to the GitHub Repository for this Lab is here.

Question:
You must do the project individually. In this HW you will design a backdoor detector for BadNets trained on the YouTube Face dataset using the pruning defense discussed in class.
Your detector will take as input:
1. B, a backdoored neural network classifier with N classes.
2. Dvalid, a validation dataset of clean, labeled images.

What you must output is G a “repaired” BadNet. G has N+1 classes, and given unseen test input, it must:
1. Output the correct class if the test input is clean. The correct class will be in [1,N].
2. Output class N+1 if the input is backdoored.

You will design G using the pruning defense that we discussed in class. That is, you will prune the last pooling layer of BadNet B (the layer just before the FC layers) by removing one channel at a time from that layer.
Channels should be removed in decreasing order of average activation values over the entire validation set. Every time you prune a channel, you will measure the new validation accuracy of the new pruned badnet. You will stop pruning once the validation accuracy drops at least X% below the original accuracy. This will be your new network B'.
Now, your goodnet G works as follows. For each test input, you will run it through both B and B'. If the classification outputs are the same, i.e., class i, you will output class i. If they differ you will output N+1.
Evaluate this defense on: 1. A BadNet, B1, (“sunglasses backdoor”) on YouTube Face for which we have already told you what the backdoor looks like. That is, we give you the validation data, and also test data with examples of clean and backdoored inputs.

1. Your repaired networks for X={2%,4%,10%}. The repaired networks will be evaluated using the evaluation script (eval.py) on this website https://github.com/csaw-hackml/CSAW-HackML-2020. Everything you need for this project is under the "lab3" directory.
2. Please create and submit a link to a GitHub repo. with any/all code you have produced in this project along with a readme that tells us how to run your code.
3. A short report (at most 2 pages) that includes a table with the accuracy on clean test data and the attack success rate (on backdoored test data) as a function of the fraction of channels pruned (X).

Results:

Drop in accuracy	Model B	Model B_prime	Repaired Model
2%	401/401 [==============================] - 13s 32ms/step Clean Classification accuracy for B: 98.62042088854248 401/401 [==============================] - 13s 32ms/step Attack Success Rate for B: 100.0	401/401 [==============================] - 16s 39ms/step Clean Classification accuracy for B_prime_2: 95.90023382696803 401/401 [==============================] - 13s 32ms/step Attack Success Rate for B_prime_2: 100.0	Clean Classification accuracy for repaired net_2: 95.74434918160561 Attack Success Rate for repaired net_2: 100.0
4%	401/401 [==============================] - 13s 32ms/step Clean Classification accuracy for B: 98.62042088854248 401/401 [==============================] - 13s 32ms/step Attack Success Rate for B: 100.0	401/401 [==============================] - 14s 36ms/step Clean Classification accuracy for B_prime_4: 92.29150428682775 401/401 [==============================] - 13s 32ms/step Attack Success Rate for B_prime_4: 99.98441153546376	Clean Classification accuracy for repaired net_4: 92.1278254091972 Attack Success Rate for repaired net_4: 99.98441153546376
10%	401/401 [==============================] - 13s 32ms/step Clean Classification accuracy for B: 98.62042088854248 401/401 [==============	401/401 [==============================] - 15s 38ms/step Clean Classification accuracy for B_prime_10: 76.30553390491036 401/401	Clean Classification accuracy for repaired net_10: 76.16523772408418 Attack Success Rate for repaired net_10: 36.26656274356976

	==========] - 13s 32ms/step Attack Success Rate for B: 100.0	[========== ==========] - 13s 32ms/step Attack Success Rate for B_prime_10: 36.2665627435697 6	

Conclusion:

There are three main takeaways that I gathered from this project:

1. The attack success rate for the original B model is greater that its clean classification accuracy which shows that BadNet model has good efficiency in attacking.
2. There is a possibility for Pruning Aware Attack which is "the evasion of the pruning defense by concentrating the clean and backdoor behavior onto the same set of neurons" as stated in this paper which leads to a quicker drop in clean classification accuracy than attack success rate before the validation accuracy drops below the original accuracy by 4%.
3. When the difference between validation and original accuracy drops below 10%, the attacks success rate is significantly lower than clean classification accuracy but consequently, the clean classification accuracy is also lower when compared to 2% and 4% accuracies which means pruning-aware attack is occuring which means the same neurons work for both clean data and poisoned data.

What we can infer from the results of this lab is that pruning defense is not very effective in detecting Backdoors for BadNets trained on the YouTube Face dataset. Further research points towards using "Fine pruning defense" which is described in the linked paper.