LAB - 4: Backdoor-Attacks Lab Report

Name: Siddharth Shah
Net ID: ss16130

ABSTRACT:

This project addresses the problem of neural network backdoor vulnerabilities and creates a preventative strategy against hacked neural network classifiers, especially BadNets—which use the YouTube Face dataset for training.

The outcome GoodNet is a model designed to precisely classify clean inputs and to identify and label backdoored inputs by adding a new category to the classification

Data Path

The data for this model is available at CSAW-HackML-2020. The dataset contains images from YouTube Aligned Face Dataset. The authors of this repository retrieved 1283 individuals and split them into validation and test datasets. Bd_valid.h5 and bd test.h5 contain validation and test images with sunglasses trigger respectively, that activates the backdoor for bd_net.h5. The structure of the data folder is as follows.

```
├── data
   └── cl
      └── valid.h5 // this is clean validation data used to design the defense
      └── test.h5  // this is clean test data used to evaluate the BadNet
   └── bd
      └── bd_valid.h5 // this is sunglasses poisoned validation data
      └── bd_test.h5  // this is sunglasses poisoned test data
├── models
   └── bd_net.h5
   └── bd_weights.h5
├── architecture.py
└── eval.py // this is the evaluation script
```

(Taken from the github repository provided in the Lab4 HW pdf)

Process:

Using average activation metrics from the clean validation set as a reference, channel pruning from the conv_3 layer of the BadNet is used as a protection mechanism. At crucial moments when validation accuracy decreases below predetermined criteria of 2%, 4%, and 10%, the models were saved.

In order for GoodNet to function, the output of the original BadNet (B) and the trimmed model (B') are compared. The original class output is produced by consistent classifications between B and B', whereas inconsistencies result in the detection class, N+1.

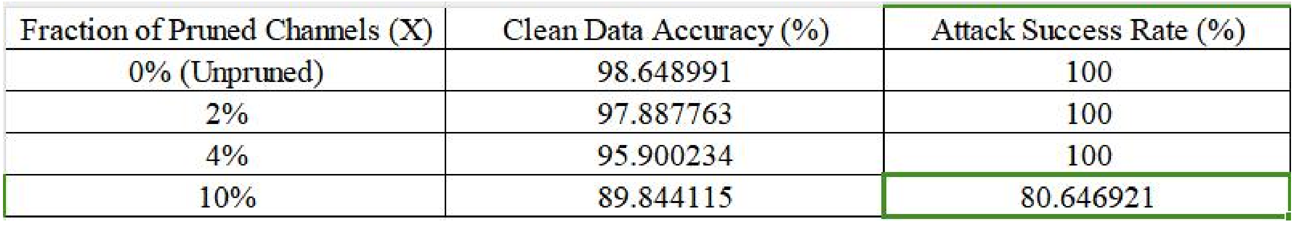
Evaluating the Backdoored Framework:

The project makes use of the DeepID network's facial recognition capabilities.
The best tool for determining the model's accuracy with clean data and susceptibility to backdoored inputs is the 'eval.py' script, which converts the theoretical robustness of the network into measurable measurements.

Observations and a thorough analysis:

Removing certain channels from the network showed promise for better security by lowering the success rate of attacks. Nevertheless, there is a clear trade-off between security and usability because this increased security comes at the expense of the network's correctness of clean data.

The complex effects of channel pruning on model performance are presented in the following table:



Fraction of Pruned Channels (X)	Clean Data Accuracy (%)	Attack Success Rate (%)
0% (Unpruned)	98.648991	100
2%	97.887763	100
4%	95.900234	100
10%	89.844115	80.646921

The data shows a clear relationship between more pruning and lower attack success rates, particularly beyond the 10% barrier. This result points to the pruning strategy's potential defense against backdoor attacks, even though it lowers classification accuracy for clean inputs in the process.

Summary:

In the field of neural network design, the trade-offs between security measures and model performance that are emphasized in this study are crucial. Trimming channels worked well as a tactic to lower the attack success rate, but they also affected the accuracy of the model using clean data. The findings of this report offer insightful explanations of the subtleties of machine learning security and the cautious Implementing defenses in neural network architectures requires a delicate balancing act.