ML-Cyber-Security

Name: Akanksha Dhote

NetID: avd8874

Data

Validation Data: bd_valid.h5 and valid.h5

Test Data: bd_test.h5 and test.h5

I could not upload data here as it is huge.

Evaluating the Backdoored Model

The DNN architecture used to train the face recognition model is the state-of-the-art DeepID network. This DNN is backdoored with multiple triggers. Each trigger is associated with its own target label.

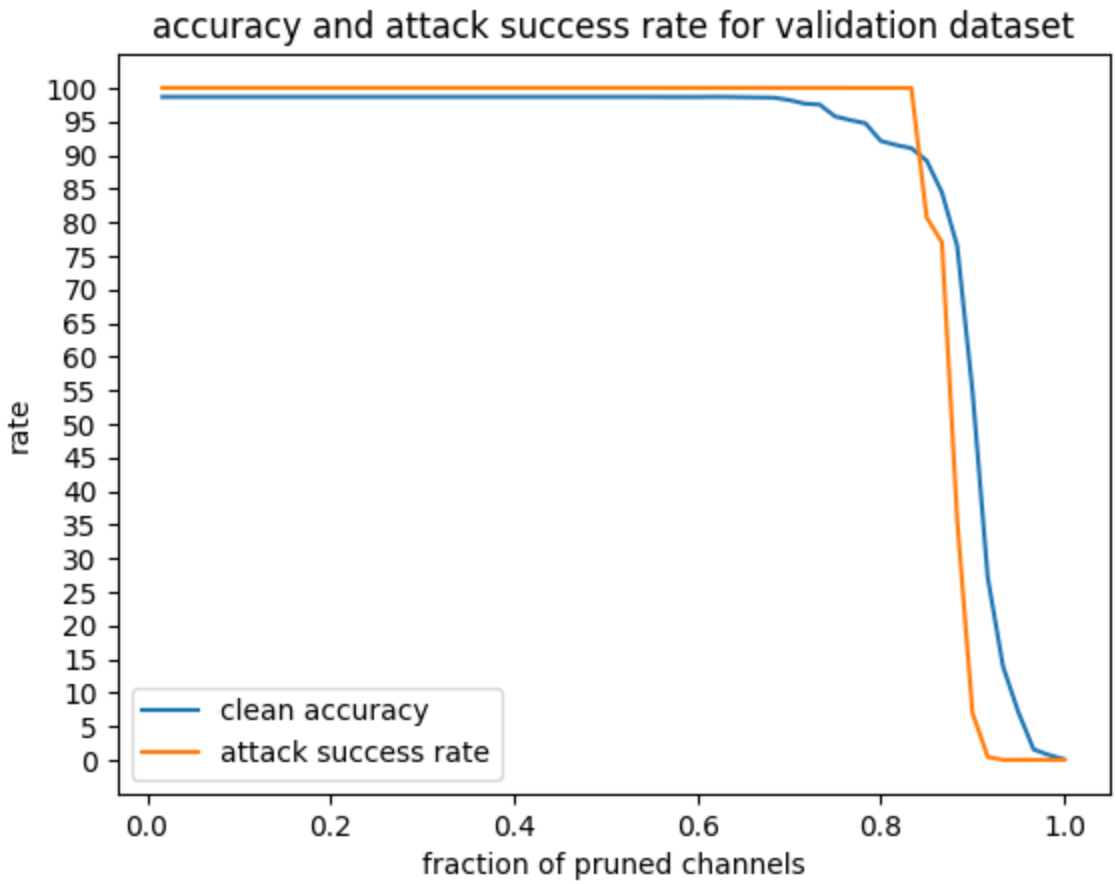
To evaluate the backdoored model, execute eval.py by running: python3 eval.py .

E.g., python3 eval.py data/clean_validation_data.h5 models/sunglasses_bd_net.h5. Clean data classification accuracy on the provided validation dataset for sunglasses_bd_net.h5 is 97.87 %.

The accuracy on clean test data: 98.64899974019225

The attack success rate as a function of the fraction of channels pruned

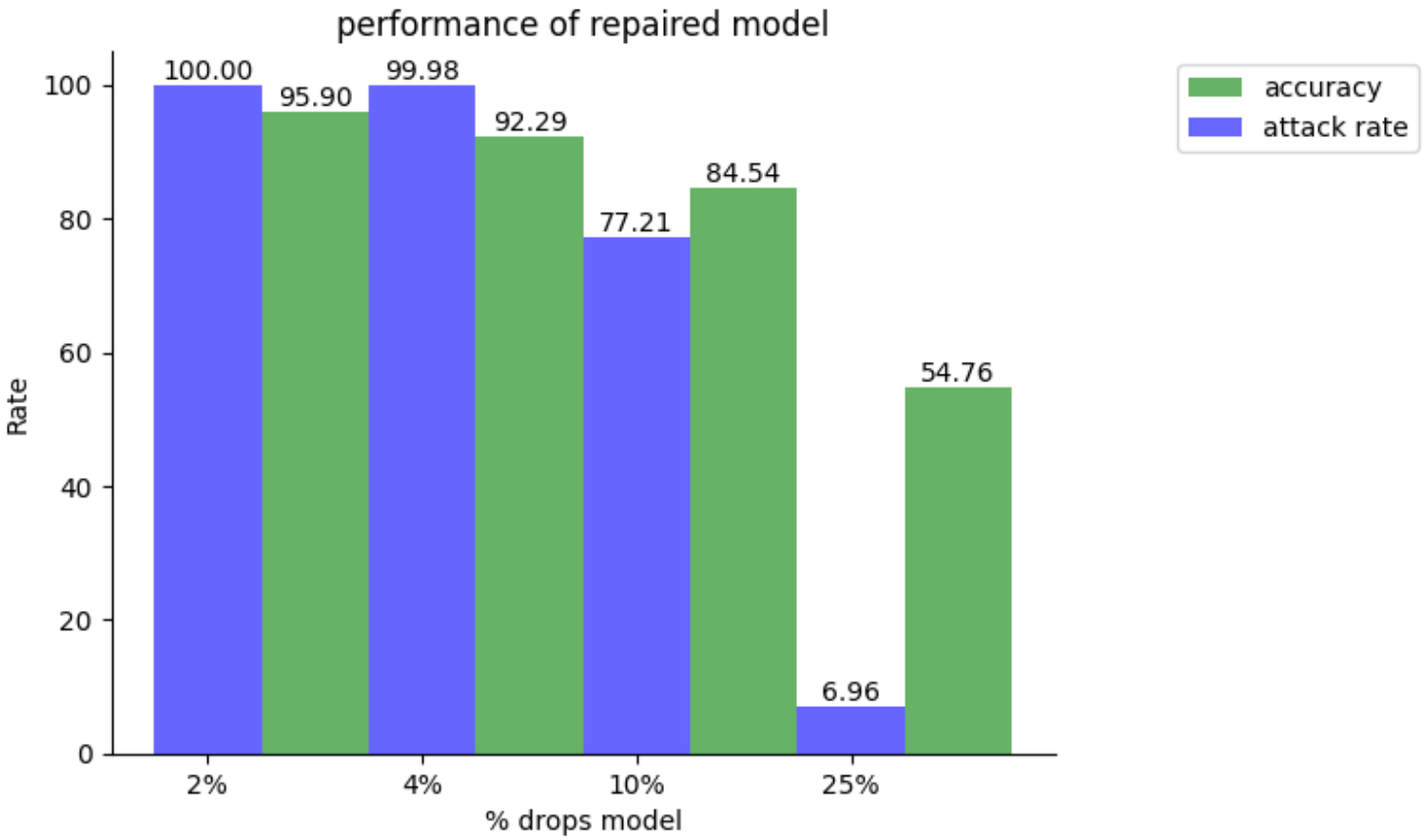
We can observe that a considerable portion of neurons can be removed without affecting the accuracy of classification. The pruning defense seems to progress through three distinct phases. In the initial phase, the pruned neurons, which aren't activated by either clean or backdoored inputs, do not impact the accuracy of either the clean dataset or the success of the backdoor attack. Moving on, the subsequent phase involves eliminating neurons activated solely by the backdoor, thereby reducing the success rate of the backdoor attack while maintaining the accuracy of the clean dataset. The final phase involves pruning neurons responsive to clean inputs, resulting in a decline in accuracy for the clean dataset. At this stage, the defense process stops, and the models are saved with a decrease in accuracy of 2%, 4%, and 10%.



Performance of repaired networks

In practical terms, the pruning defense demonstrates a beneficial balance between the accuracy of classifying clean inputs and the success of the backdoor attack. It achieves a notable decrease in the latter while minimizing the decline in the former. We use the saved models from the pruning step and evaluate it's accuracy and attack success rate on test set which can be seen down below.

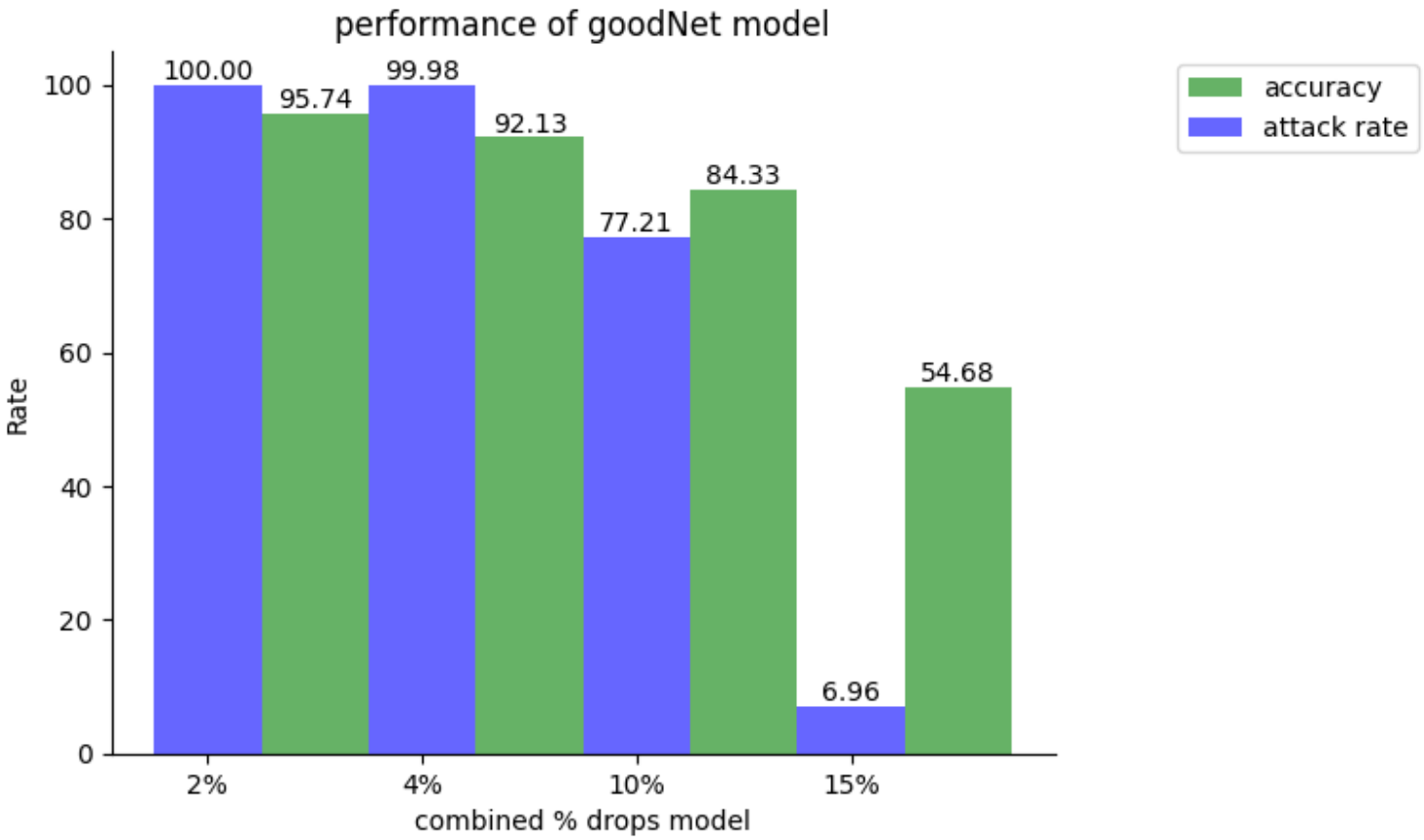
model	text_acc	attack_rate
2%_repaired	95.900234	100.000000
4%_repaired	92.291504	99.984412
10%_repaired	84.544037	77.209665
25%_repaired	54.762276	6.960249



Performance of Goodnet (combined) model

We combine the saved models with the bd model and evaluate the new models get the accuracy and attack success rate on test data.

G_model	G_text_acc	G_attack_success_rate
G_2%	95.744349	100.000000
G_4%	92.127825	99.984412
G_10%	84.333593	77.209665
G_25%	54.676539	6.960249


performance of goodNet model
accuracy
attack rate
Rate
0
20
40
60
80
100
100.00
95.74
99.98
92.13
77.21
84.33
6.96
54.68
2%
4%
10%
15%
combined % drops model