# Backdoor Attack Defense

**Github repo**: https://github.com/moinkhan3012/Backdoor_Pruning_Defence

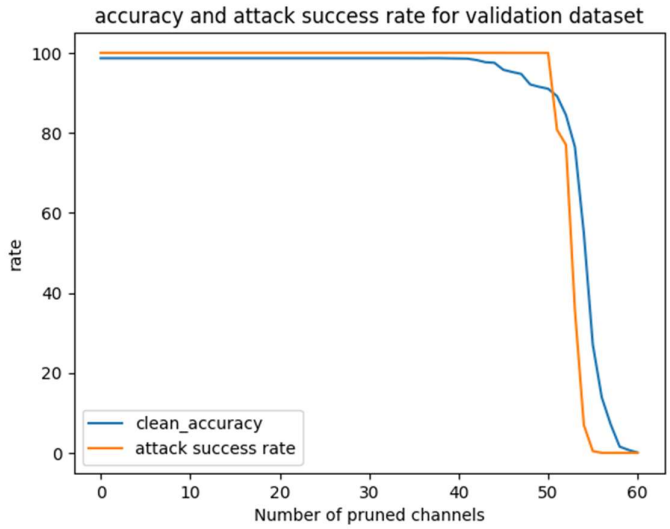
In Backdoor defence pruning methodology, the purpose is to enhance the robustness of machine learning models against backdoor attacks. In this defence method, we pruned the layers which are susceptible to manipulation by backdoor patterns.

In our model, the last pooling layer, named **"pool_3,"** comprises a total of 60 channels. Initially, we determine the model accuracy without any pruning. Next, we calculate the average activation value for each channel in the **"pool_3"** layer.

Subsequently, we iterate over the channels in ascending order of their average activations, progressively pruning the model by setting the corresponding weights and biases of previous convolutional layer **"conv_3"** to zero. This step-by-step process ensures a systematic approach to channel pruning based on their individual contributions to the model's performance.

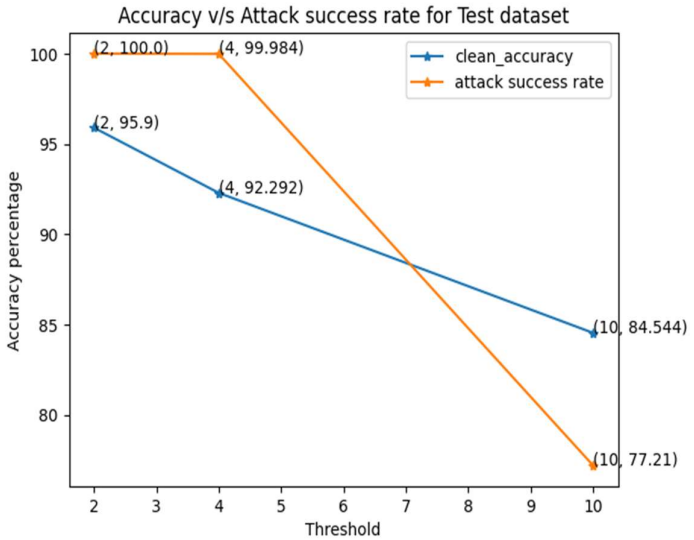
## Observations:

We pruned the layers and saved the model when accuracy drops by at least 2% 4% 10%. We can observe the trend, as we pruned the channels of pool_3 layer.




Following is the performance of the repaired backdoor models on test data.

| model | test_accuracy | attack_rate |
|---|---|---|
| repaired_model_2% | 95.900 | 100.000 |
| repaired_model_4% | 92.292 | 99.984 |
| repaired_model_10% | 84.544 | 77.210 |



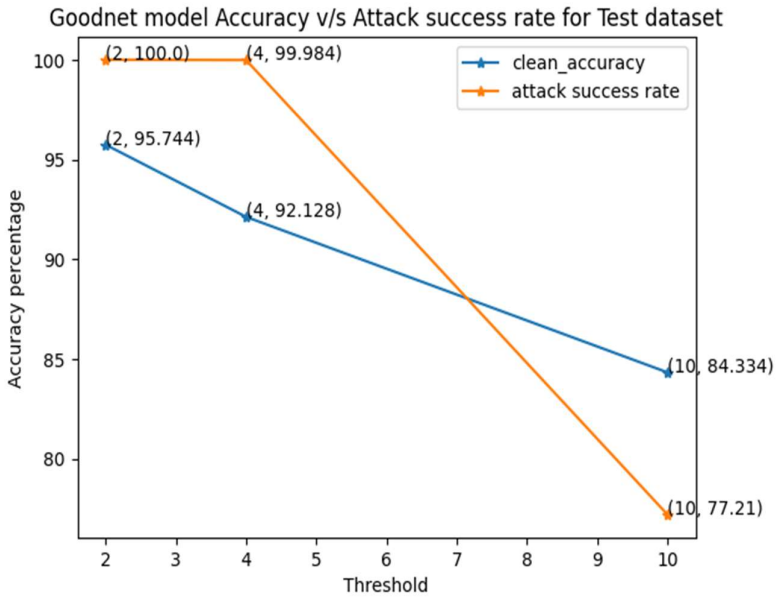
# Goodnet:

The goodnet model will be the combination of Bad Net (backdoor Model) and the repaired Backdoor model.

For each test input, we will run it through both Backdoor Model and Repaired Backdoor Model. If the classification outputs are the same, i.e., class i, goodnet will output class i. If they differ goodnet will output N+1 (1284).

Below is the performance of the Goodnet model on Test Data.

| model | test_accuracy | attack_rate |
|---|---|---|
| repaired_model_2% | 95.744 | 100.000 |
| repaired_model_4% | 92.128 | 99.984 |
| repaired_model_10% | 84.334 | 77.210 |




Here we observe that the Goodnet model has the same performance as the Badnet repaired model. However, for backdoor input, the goodnet model instead of returning backdoor output, it returns 1284 indicating the input is backdoored.

## Steps to run the code:

The backdoor detection and goodnet code is present in backdoor.ipynb notebook.

- **Backdoor Detection**: **PrunedModel** class is responsible to detect and pruned the channels.
- **Goodnet Model**: **G** class acts as a **Goodnet** model, to predict the output using outputs of Backdoor and repaired Backdoor Model.

The dataset can be downloaded from here, https://github.com/csaw-hackml/CSAW-HackML-2020/tree/master/lab3, and required to place under **/data** folder.