ECE-GY 9163 ML for cyber security

Group Member: Ziqi Yang Zy2006 Yiran Meng ym2337 Zhongkang Li zl3546
Github Repo: github.com/ZhongkangLi/CSAW_2020_Project

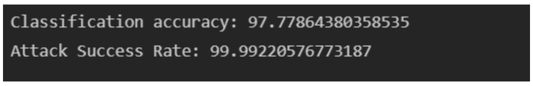
1.Problem Description

The Problem we are facing is the backdoor attacks on DNN model. The attackers provide a pretrained model which can pass through the verification on the small validation data, the model is highly accurate on the clean validation data, but the special input will trigger the backdoors damage the performance. These DNNs are referred as BadNets.

In our project the inputs are some user's face portrait, we need to repair our model to mitigate backdoors, to reduce the attack-success rate with backdoored inputs.

2.Our Method and Result:

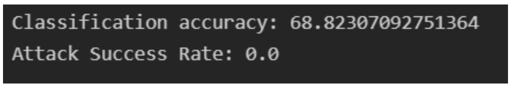
Before Pruning: We can see that for the glasses model the Classification accuracy and the Attack success rate are:



```
Classification accuracy: 97.77864380358535
Attack Success Rate: 99.99220576773187
```

The accuracy and the attack success are both very high.
Then we simply implement an DNN model in the architecture.py file:

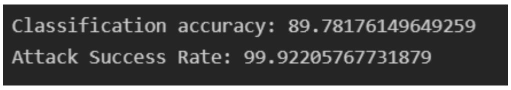
After training for 15 epoch, we evaluate it on the test and poisoned dataset, the accuracy and the attack success are:



```
Classification accuracy: 68.82307092751364
Attack Success Rate: 0.0
```

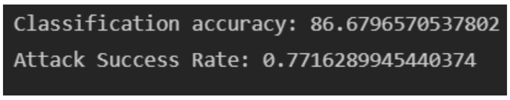
As we can see from the result, although the attack rate is 0%, the accuracy is high enough. So, we need to prune each layer.

We decide to prune 30 neurons in increasing order of activations. The results after Pruning are:



```
Classification accuracy: 89.78176149649259
Attack Success Rate: 99.92205767731879
```

As we can see after pruning the model, the attack success rate is still high, so we retrain the model using clean validation model.

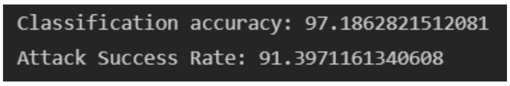


```
Classification accuracy: 86.6796570537802
Attack Success Rate: 0.7716289945440374
```

Compared to the model that simply retrained with the clean validation data (Classification accuracy: 68.82% Attack Success Rate: 0.0) , this is a better result we expected.

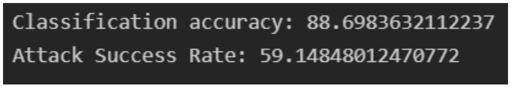
As for the anonymous models, we do the exact same way to prune the model and then do retrain with clean validation model.

The result of before pruning is:



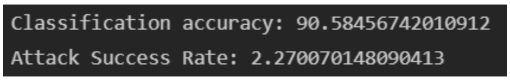
```
Classification accuracy: 97.1862821512081
Attack Success Rate: 91.3971161340608
```

The result of after pruning is:



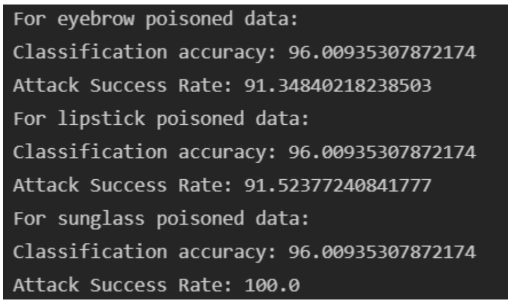
```
Classification accuracy: 88.6983632112237
Attack Success Rate: 59.14848012470772
```

The result of after retraining is:



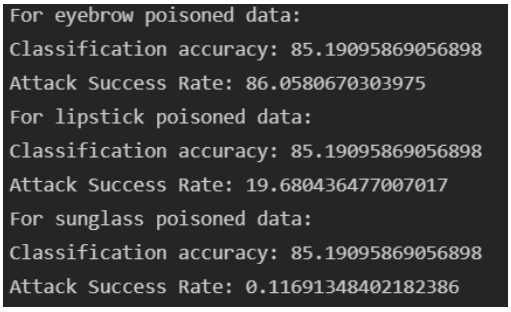
```
Classification accuracy: 90.58456742010912
Attack Success Rate: 2.270070148090413
```

For the multi-trigger badnet we do similar method:
The result of before pruning is:



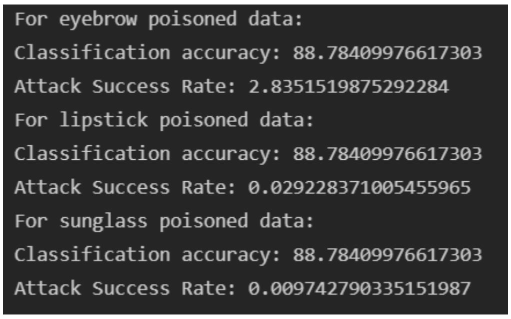
```
For eyebrow poisoned data:
Classification accuracy: 96.00935307872174
Attack Success Rate: 91.34840218238503
For lipstick poisoned data:
Classification accuracy: 96.00935307872174
Attack Success Rate: 91.52377240841777
For sunglass poisoned data:
Classification accuracy: 96.00935307872174
Attack Success Rate: 100.0
```

The result of after pruning is:



```
For eyebrow poisoned data:
Classification accuracy: 85.19095869056898
Attack Success Rate: 86.0580670303975
For lipstick poisoned data:
Classification accuracy: 85.19095869056898
Attack Success Rate: 19.680436477007017
For sunglass poisoned data:
Classification accuracy: 85.19095869056898
Attack Success Rate: 0.11691348402182386
```

The result of after retraining is:



```
For eyebrow poisoned data:
Classification accuracy: 88.78409976617303
Attack Success Rate: 2.8351519875292284
For lipstick poisoned data:
Classification accuracy: 88.78409976617303
Attack Success Rate: 0.029228371005455965
For sunglass poisoned data:
Classification accuracy: 88.78409976617303
Attack Success Rate: 0.009742790335151987
```