Pruning Defense for Backdoor Attack through Frequency Domain

Yunyi Liao,
New York University
Brooklyn, NY 11201
yl9441@nyu.edu

Yingkai Hao,
New York University
Brooklyn, NY 11201
yh4171@nyu.edu

Xiangyu Lu
New York University
Brooklyn, NY 11201
xl4044@nyu.edu

Abstract

With the fast improvement of attacks based on the convolutional neural network (CNN), different useful defense methods have been created to protect the CNN models. Among these attacks, backdoor attacks create huge damage for some well-trained CNNs, such as face recognition backdoor[4], speech recognition backdoor[5], and traffic sign backdoor[6]. These attacks mislead the recognition of specific objects when a trigger known only to the attacker is present. Most of the backdoor-attacked model remains high accuracy on clean datasets. And, only the poisoned samples with specific triggers are affected by this attack. Previously, triggers on samples are clear, visible, and easy to detect. Recently, a special backdoor attack generates triggers on the frequency domain making the poisoning images retain the perceptual similarity to the original clean images. Therefore, we implemented the pruning defense for this attack using the GTRSB dataset. We show that the pruning defense is sufficient to defend against this kind of attack. After we evaluate our defense methods, the result shows that the attack success rate decreased from 100% to 26%, which means it successfully mitigates the backdoors.

Key words: machine learning, backdoor, pruning, FTrojan

1 Introduction

The backdoor attack has, over the past five years, produced bad influences on well-trained convolutional neural networks (CNN). Normally, the triggers on back door attack like BadNet[7], Blend[8], TrojanNN[9], Clean Label[10], Dynamic Backdoor[11], IAB[12], SIG[13], and REFOOL[14] is clear and visible to detect.

However, the paper we plan to extend focuses on backdoor attack through frequency domain[1]. In this paper, the authors did something new related to the backdoor attack, which is generate the triggers on the frequency domain. The authors called this type of backdoor attack an FTrojan attack.FTrojan shows great efficiency, specificity, and especially fidelity. Since the attacker puts the labels on the frequency domain on training datasets. It is difficult for people to detect such differences between original pictures and altered pictures. Meanwhile, the attack is successful in multiple datasets whose attacking success rate is above 95%. Also, FTrojan does not degrade prediction accuracy on benign inputs.

What we did recently is that we rebuild the FTrojan once again using the authors' codes on CIFAR10. Then, we use a new dataset called GTSRB to reproduce this attack. We found that the attacking successful rate from GTSRB is 100%, which is higher than success rate on CIFAR10. Next, we plan to implement a defense called pruning defense for FTrojan. The reason why we use pruning is that the label on the training dataset is not easy to detect for both humans and machines. Also, the trigger is dispersed all over the picture. It's not likely to use some well-known defense methods like Neural Cleanse and so on. Pruning focus on dealing with the neurons in well-trained models. The defender iteratively prunes neurons from the DNN in increasing order of average activations and records the accuracy of the pruned network in each iteration. And, the defense terminates when the accuracy on the validation dataset drops below a predetermined threshold. For now, we think pruning is enough and sufficient for our project. We may implement more defense methods like fine-pruning defense in the future to create more robust protection for machine learning and deep learning models.

2 Methodology

2.1 Frequency Domain Backdoor Attack

A typical backdoor attack will try to poison training data with specified marks or predefined triggers to identify the poison data and mislead the neural network. For instance, BadNet used triggers like colored squares and changed the label to the target labels, which can be easily identified by the human visual system. The Frequency Domain Backdoor Attack proposed a new attack method called FTrojan to disperse the trigger to the entire image, making it less detectable to the backdoor defense system and human visual system[1].

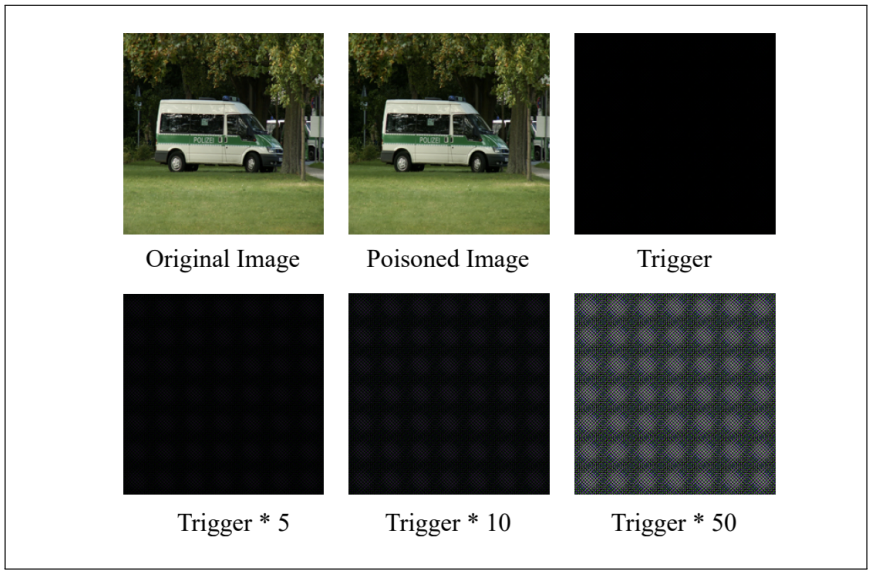

Figure 1: An illustration of the trigger of FTrojan.[1]

In Frequency Domain Backdoor Attack, the back door attack will be in the frequency domain. The triggers in the frequency domain can be learned and memorized by CNN, and the trigger will be dispersed to the entire image. Also, the frequency domain backdoor attack can improve the image fidelity, making it hard to recognize by human eyes.

The main process of the Frequency Domain Backdoor Attack is:

- first to transform the color channel from RGB to YUV, as the human visual system is less sensitive to the YUV channel.
- After that, we need to use discrete cosine transform to transform the UV channels of the image from the spatial domain to the frequency domain.
- Step three is to generate triggers in the frequency domain. The frequency domain backdoor attack considers generation triggers related to the frequency that the trigger is placed on and the magnitude of the trigger.
- Step four is to use inverse DCT to get the poisoning image in the spatial domain.
- The last step is to color the image by transforming the poisoned image from YUV image to RGB image.

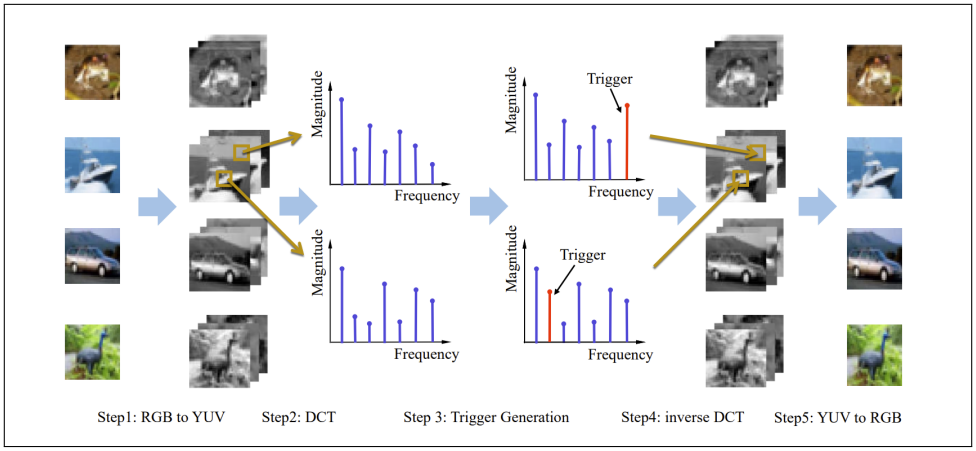

Figure 2: The overview of the proposed backdoor attack FTrojan.[1]

2.2 Pruning Defense

As we mentioned in the Frequency Domain backdoor attack, the neural network will learn the poisoned image and target label and triggers in the frequency domain are recognizable and learnable by CNNs. What's more, Gu et al.[2] suggest that some neurons will be active by triggers but won't show activation towards clean data. In this case, it will be a good practice to prune the neurons that sensitive to triggers. The pruning defense proposed a method to clean the neural network to make it only work for the clean input.

- After the defender receives the DNN trained by the attacker, the defender will use the clean dataset to exercise the DNN and record the count of the activation of neurons.
- After that, the defender will start to prune the DNN in increasing order of the activation of neurons iteratively.
- During this process, defender will also record the accuracy and stop the defense when accuracy decrease below expectation.[3].

3 Progression

3.1 Attack

We followed the code in the Backdoor Attack through the Frequency Domain GitHub website, and we used GTRSB to replicate the attacks in the paper. The German Traffic Sign Recognition Dataset

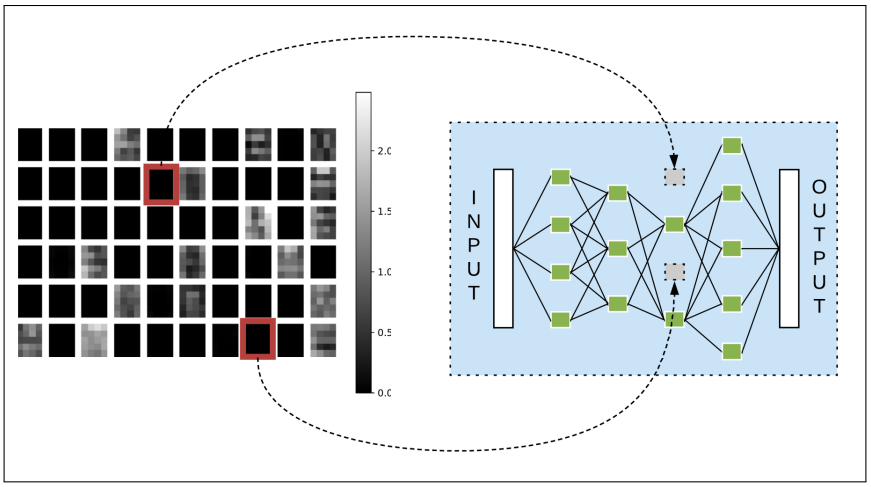

Figure 3: Illustration of the pruning defense.[3]

(GTSRB) is classified as 43 classes and contains a total of 39209 labeled images on the training set and 12630 images on the test set.

We pre-processed the GTSRB dataset, adjusted these images to the same standard 32*32 size, and shuffled the training set. Then we poisoned a part of the training images according to our poisoning rate, making them all have a specific pre-defined label. For these poisoned images, we first transformed these images to YUV channels and further to the frequency domain, then placed the trigger in the frequency domain, and then transformed the images back to the spatial domain. And we poisoned the test set as well to evaluate the performance of our attack.

We trained our model by using the part poisoned training set, and we used the benign test set and the poisoned test set to evaluate It respectively.

Table 1: The result of the benign test set and poisoned test set

	Accuracy on benign test set(%)	Accuracy on Poisoned test set(%)
GTRSB	97.165	100.000
CIFAR10	86.160	99.970

We set the trigger at frequency bands (15,15) and (31,31) where (15,15) belongs to the mid-frequency bands and (31,31) belongs to the high-frequency component. We also set the trigger magnitude to 30 due to the size of the images. We made the injection rate fixed to 5%, and we use Adam as the optimizer of the model with learning rate 0.0005. and we set the batch size to 64, the attack label is set to 10. All experiments were carried out on our Windows 10 machine equipped with 32GB RAM, one 20-core Intel i7-12700K CPU at 3.70GHz, and one NVIDIA GeForce RTX 3070Ti GPU.

3.2 Pruning defense

After implementing the backdoor attack on frequency domain attack on the GTSRB dataset, we use the pruning method to repair the badnet. The following image shows how the summary of the attack model.

For this, we should prune the last convolution layer before the last pooling layer of the badnet, which is the “actication_5” layer according to the attack model. Then we obtain the average activation value of each channel in the last pooling layer. Then we rearrange the channels in increasing orders according to their averaged activation values. We will use this rearranged sequence of channels to

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	896
activation (Activation)	(None, 32, 32, 32)	0
batch_normalization (BatchNo	(None, 32, 32, 32)	128
conv2d_1 (Conv2D)	(None, 32, 32, 32)	9248
activation_1 (Activation)	(None, 32, 32, 32)	0
batch_normalization_1 (Batch	(None, 32, 32, 32)	128
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
dropout (Dropout)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	18496
activation_2 (Activation)	(None, 16, 16, 64)	0
batch_normalization_2 (Batch	(None, 16, 16, 64)	256
conv2d_3 (Conv2D)	(None, 16, 16, 64)	36928
activation_3 (Activation)	(None, 16, 16, 64)	0
batch_normalization_3 (Batch	(None, 16, 16, 64)	256
max_pooling2d_1 (MaxPooling2	(None, 8, 8, 64)	0
dropout_1 (Dropout)	(None, 8, 8, 64)	0
conv2d_4 (Conv2D)	(None, 8, 8, 128)	73856
activation_4 (Activation)	(None, 8, 8, 128)	0
batch_normalization_4 (Batch	(None, 8, 8, 128)	512
conv2d_5 (Conv2D)	(None, 8, 8, 128)	147584
activation_5 (Activation)	(None, 8, 8, 128)	0
batch_normalization_5 (Batch	(None, 8, 8, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 4, 4, 128)	0
dropout_2 (Dropout)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 43)	88107

Total params: 376,907
Trainable params: 376,011
Non-trainable params: 896

Figure 4: Summary of the attacked model

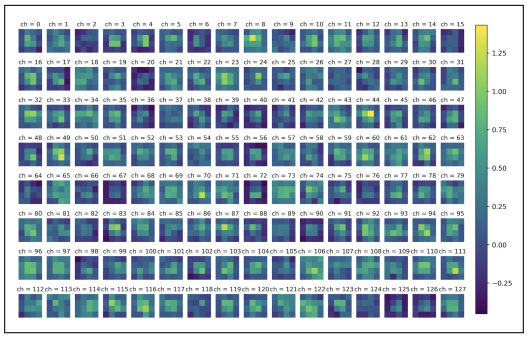

Figure 5: Clean Activations

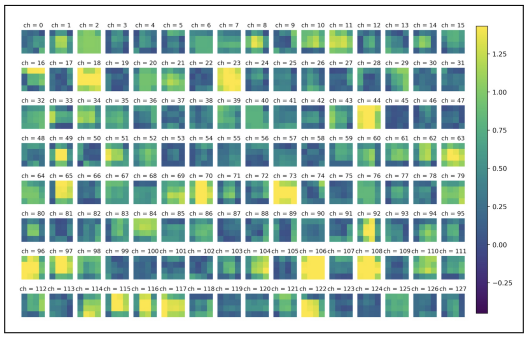

Figure 6: Backdoor Activations

prune our badnet. Eventually, we should extract the weight and bias of the last convolution layer for pruning as well.

In this defense experiment, we divided the dataset into three different parts:

- The clean validation set to help us to train repaired model
- The clean test contains only clean data, which is set to help us measure the clean classification accuracy of the different repaired models.
- The bad valid set contains only poisoned data all pointing to the target label, which is set to help us measure the attack success rate.

The two visualization images indicate that while there is almost every neuron is activated in both cases, some neurons are getting extremely over-activated for backdoored validation data. Then we pruned one channel each time and compared the validation accuracy with the original badnet accuracy, and we set the threshold as 2%, 10%, 20%, and 30%, as soon as the validation accuracy dropped to the thresholds, we saved the models as repaired networks.

From the beginning of the pruning to 75 percent of channels are removed, the clean classification drops slightly while the attack success rate almost remains around 100%. That is because almost all the layers (neurons) are activated for both cases, and we are pruning those neurons which are used for classifying clean data, and not those neurons activated by the badnet. So only the clean classification accuracy keeps dropping, and the attack success rate is not affected. And then from 75 percent of channels removed to about 83 percent of channels removed, both lines dropped dramatically. It shows that we are currently removing those neurons which are activated by both backdoor attacks and the

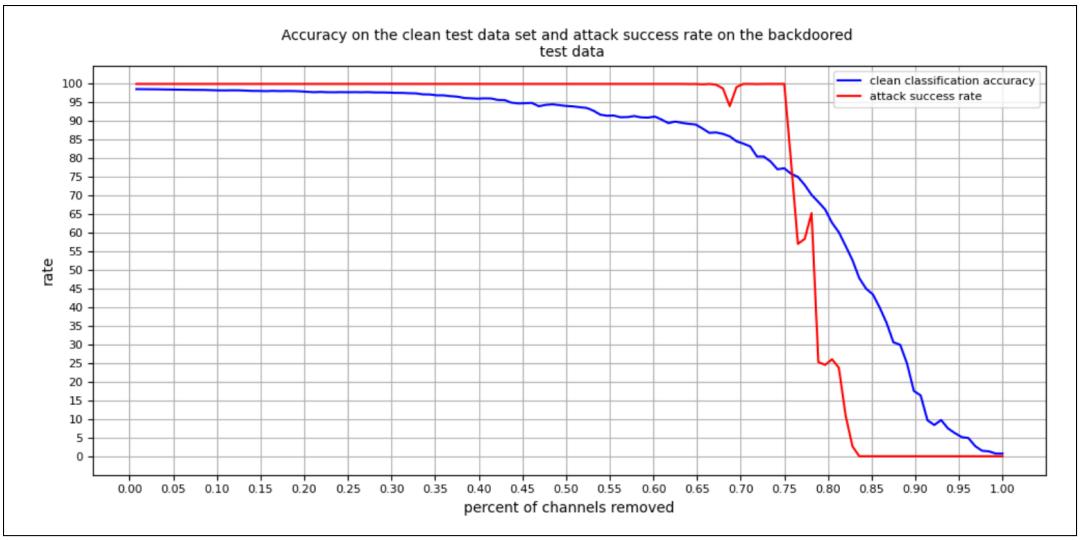

Figure 7: The trend of the accuracy of clean data and attack success rate

classification of the clean data. Finally, starting from 83 percent of channels removed, the attack success rate is dropped to 0, and the clean classification continuously drops until reaches 0.

As can be seen from the figure, the distribution of activated neurons is relatively even for the clean data set. While the neurons activated by backdoor attacks in the frequency domain were highly overlapped with those activated by clean data sets, the number of neurons activated by overactivation was significantly less than neurons activated by clean data. These factors directly lead to the early decline of recognition accuracy and no significant change in attack probability at the beginning of pruning.

Because the activation times of neurons in the contaminated data set differed significantly, the initial pruning did not trim the key neurons associated with the contaminated data set. However, we can observe that as the number of pruned neurons increases, the success rate of backdoor attacks in the frequency domain decreases rapidly after the critical neurons are pruned.

We think the reason for the difference between this experiment and the baseline backdoor attack image is that we have an even and wide distribution of neurons activated in the clean data set. We believe that the way to further enhance the attack on the frequency domain is to trim the neurons that are over-activated in the frequency domain from high to low according to the number of activation. We can even comprehensively evaluate the number of clean data set activations versus backdoor data set activations and prioritize pruning to further improve performance.

The fraction of channels pruned is 53%, 70%, 77%, and 80% when validation accuracy dropped by 2%, 10%, 20%, and 30% respectively. Based on Figure 7 and Figure 8 above, we can notice that the repairing model is not too effective, it cannot prevent an attack unless the clean data validation rate drops a lot. For validation accuracy drops by 2% and 10% below the original accuracy, the attack success rate prevails over the prediction accuracy, due to the badnet still showing around 100% attack success rate. Things changed when validation accuracy drops to 20% and 30%, the attack success rate drops intensely even below the validation accuracy of clean data. The trend of this model is really the same as pruning aware attack, due to the backdoor attack neurons being the same as a part of classification neurons.

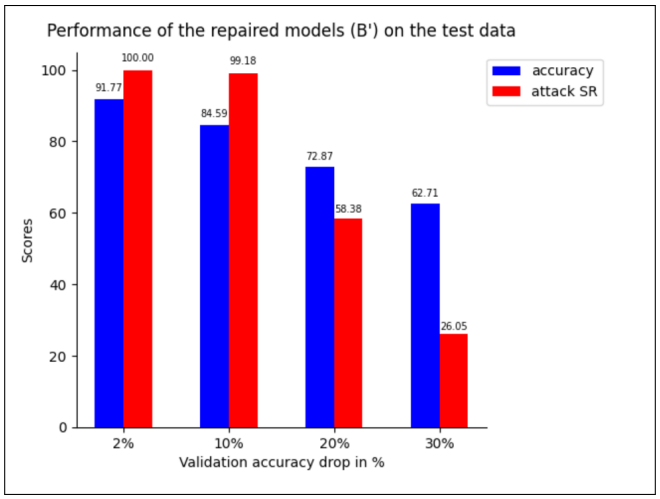

Figure 8: Performance of the repaired model

3.3 System topology

The following are the dependencies of our system environment:

Python 3.8.3,

Numpy 1.23.4,

Tensorflow-GPU 2.5.0,

Opencv 4.5.1,

Idx2numpy = 1.2.3.

4 Conclusion

In this paper, we reproduce the FTorjan attack using both CIFAR10 and GTSRB. We found that the attack success rate on GRSRB is up to 100%. Then, we decided to do a pruning defense for this attack. The result is sufficient to show that, by using pruning defense on the backdoor attack through the frequency domain, the attack success rate decreased from 100% to 26%, which means it successfully mitigates the backdoors. Therefore, we conclude that for this specific attack-FTorjan, pruning defense provides strong protection.

Currently, we only use pruning defense against the FTorjan attack. However, we still feel that we could extend more in the future. According to our analysis above, we discover that although the attack success rate drops significantly to 26%, the clean validation accuracy also drops to 65%. The reason why the clean validation accuracy also drops this much is the unique channel activations feature in the last pooling layer. Fig 5 and Fig 6 shows that the activeness of each channel is almost the same, which means no matter which channel we prune, the overall impact on the clean validation accuracy is almost the same. Therefore, we found that as we start pruning, the clean validation accuracy drops slowly. However, these channels we pruned did not happen to be the ones active in the backdoor attack model. And, when we start pruning the relative active channels in the backdoor attack model, the clean validation accuracy already drops too much.

In order to solve this problem, we thought we could produce the pruning backed on the activation level in the backdoor attack. We could prune channels from most active in the backdoor attack model to least active. Since all channels' activeness on the clean model is almost the same and we prune

most active channels first, the clean validation accuracy will drop slowly, while the attack success rate will decrease significantly.

References

[1]Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, Ting Wang. 2021. Backdoor Attack through Frequency Domain. arXiv preprint arXiv:2111.10991.

[2]T. Gu, S. Garg, and B. Dolan-Gavitt. BadNets: Identifying vulnerabilities in the machine learning model supply chain. In NIPS Machine Learning and Computer Security Workshop, 2017. https://arxiv.org/abs/1708.06733.

[3]Liu, K., Dolan-Gavitt, B., Garg, S. (2018). Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. International Symposium on Recent Advances in Intrusion Detection.

[4]X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. ArXiv e-prints, Dec. 2017.

[5]Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. In 25nd Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-221, 2018. The Internet Society, 2018.

[6]T. Gu, S. Garg, and B. Dolan-Gavitt. BadNets: Identifying vulnerabilities in the machine learning model supply chain. In NIPS Machine Learning and Computer Security Workshop, 2017. https://arxiv.org/abs/1708.06733.

[7]Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017).

[8]Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017).

[9]Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning Attack on Neural Networks. In Annual Network and Distributed System Security Symposium (NDSS).

[10]Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2018. Clean-label backdoor attacks. (2018).

[11]Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. 2020. Dynamic backdoor attacks against machine learning models. arXiv preprint arXiv:2003.03675 (2020).

[12]Tuan Anh Nguyen and Anh Tran. 2020. Input-Aware Dynamic Backdoor Attack. In Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS).

[13]Mauro Barni, Kassem Kallas, and Benedetta Tondi. 2019. A new backdoor attack in CNNs by training set corruption without label poisoning. In 2019 IEEE

[14]Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. 2020. Reflection backdoor: A natural backdoor attack on deep neural networks. In European Conference on Computer Vision (ECCV). Springer, 182–199. International Conference on Image Processing (ICIP). IEEE, 101–105.