ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230610

Invisible Backdoor Attacks Using Data Poisoning in Frequency Domain

Chang Yue[a,b], **Peizhuo Lv**[a,b], **Ruigang Liang**[a,b;*] **and Kai Chen**[a,b]

[a]SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences, China
[b]School of Cyber Security, University of Chinese Academy of Sciences, China
{yuechang, lvpeizhuo, liangruigang, chenkai}@iie.ac.cn

Abstract. Backdoor attacks have become a significant threat to deep neural networks (DNNs), whereby poisoned models perform well on benign samples but produce incorrect outputs when given specific inputs with a trigger. These attacks are usually implemented through data poisoning by injecting poisoned samples (samples patched with a trigger and mislabelled to the target label) into the dataset, and the models trained with that dataset will be infected with the backdoor. However, most current backdoor attacks lack stealthiness and robustness because of the fixed trigger patterns and mislabelling, which humans or some backdoor defense approach can easily detect. To address this issue, we propose a frequency-domain-based backdoor attack method that implements backdoor implantation without mislabeling the poisoned samples or accessing the training process. We evaluated our approach on four benchmark datasets and two popular scenarios: no-label self-supervised and clean-label supervised learning. The experimental results demonstrate that our approach achieved a high attack success rate (above 90%) on all tasks without significant performance degradation on main tasks and robust against mainstream defense approaches.

1 Introduction

Due to significant improvements in computing power, deep learning has rapidly developed and has been widely applied in various areas, including supervised learning (SL) trained on labeled datasets and self-supervised learning (SSL) trained on pretext tasks with unlabeled datasets. These applications have profoundly changed people's production and lifestyle, such as face recognition [33, 28], speech recognition [1, 37], autonomous vehicles [2, 25], and remote diagnosis [29]. However, the ubiquitous and successful application of deep learning has also brought new security issues, such as adversarial attacks [36, 42, 46] and backdoor attacks [14, 23, 8]. Unlike adversarial attacks that exploit the intrinsic vulnerability of DNNs in the inference phase, backdoor attacks poison models in the training phase, causing them to perform well on benign samples but producing incorrect outputs when given backdoor samples.

State-of-the-art data poisoning-based methods face challenges in achieving sufficient stealthiness and robustness, as illustrated in Figure 1. Specifically, (1) the poisoned samples often have fixed trigger patterns and incorrect labels in labeled datasets, or fixed trigger patterns in unlabeled datasets, making them easily detectable by humans. (2) Some defense mechanisms, such as Neural Cleanse [40] and SentiNet [9], can detect and reconstruct the fixed trigger, making the attack ineffective.

Figure 1: Examples of backdoor attacks. a) is the original image. b), c), d) are images patched with the trigger proposed by Badnets [14], TojanNN [23] and our work. Badnets and TrojanNN samples are mislabeled to the target label (e.g., car), but ours is with a clean label.

Drawing inspiration from recent research works such as [45, 44, 26], which demonstrate that DNN models can learn signals in the frequency domain and that a minor alteration in the frequency domain can affect all spatial domain pixels, imperceptible to human eyes, we propose that the frequency domain-based backdoor attack approach can potentially address the issues outlined above. However, several challenges must be addressed while designing the frequency domain-based backdoor attack.

Challenges in designing frequency domain backdoor. C1: Be robust to input preprocessing defenses, such as filters. C2: Evade defenses that rely on trigger detection, as current backdoor defenses typically identify specific trigger patterns in the spatial domain. C3: Balance learnability and stealthiness, considering that triggers with higher frequencies and intensities are easier for DNN models to learn but more perceptible to humans.

In this paper, we propose an algorithm for adaptive trigger selection to address the challenges mentioned above, which involves three phases. Firstly, we select multiple frequencies that are robust to standard filters (such as Gaussian filters) as candidates to ensure robustness to defenses that preprocess the input (addressing C1). In the second phase, we select frequencies that generate different trigger patterns on various images in the spatial domain to apply our modification in the frequency domain, ensuring that our attack bypasses defenses based on trigger detection (addressing C2). In the third phase, we choose a target intensity that is slightly higher than the average intensity of the original image but no greater than the threshold value for each selected frequency, taking into account the location of the frequencies and the average intensity value, to ensure both the learnability and stealthiness of the frequency backdoor (C3).

We assess the effectiveness of our proposed attack in two neu-

* Corresponding Author.

ral network learning scenarios: self-supervised learning (SSL) and supervised learning (SL). In the SSL setting, we pre-train ResNet-18 [17] on a poisoned CIFAR-10 [20] dataset using popular methods SimCLR [6] and MOCO V2 [7]. The pre-trained model is the feature extractor and is transferred to downstream tasks, including CIFAR-10, STL-10 [10], and GTSRB [35]. In the SL setting, we train ResNet-18 and DenseNet on the poisoned CIFAR-10, STL-10, and VGGFace datasets, achieving over 90% success rate on the poisoned samples while incurring only a minor 2% performance degradation on the main tasks. Then, we evaluate the impact of our frequency trigger on the original images using PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity) metrics. The average PSNR and SSIM on CIFAR-10 are 24.11 and 0.9024, respectively, demonstrating that our trigger is well-hidden. Finally, we demonstrate the robustness of our attack against common backdoor detection methods and can effectively bypass them with high success robustness.

Contributions. Our main contributions are outlined below:

• We present an innovative invisible backdoor attack that utilizes a frequency trigger designed based on statistical characteristics in the frequency domain. Our approach is the first to achieve backdoor implantation in no-label and clean-label scenarios without mislabeling the poisoned samples or accessing the training process.

• We propose an adaptive algorithm for selecting appropriate properties of our frequency trigger, enhancing its stealthiness and robustness against commonly used defense methods.

• We implement our proposed invisible backdoor attack in the frequency domain, achieving over 90% attack success rate while maintaining the performance of the main task. We publish our source code on the GitHub[1].

2 Related Work

2.1 Frequency Domain

The frequency domain provides a new perspective for image processing. In the frequency domain, the low-frequency components correspond to smooth regions in the image, and the high-frequency components correspond to edges in the image.

The spatial domain and frequency domain are connected through the Fourier transform. The Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) are the most commonly used transform methods in digital image processing. DCT, which is developed from DFT, is widely used in image compression because it has better energy compaction in the frequency domain than DFT. The two-dimensional DCT can be expressed as matrix multiplication, and its inverse, the Inverse Discrete Cosine Transform (IDCT), can be obtained by transposing the DCT matrix. The mathematical expressions for the 2D DCT and IDCT are shown below:

$$F(u,v) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} f(i,j)G(i,j,u,v) \tag{1}$$

$$f(i,j) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)G(i,j,u,v) \tag{2}$$

$$G(i,j,u,v) = c(u)c(v)\cos[\frac{(i+0.5)\pi}{M}u]\cos[\frac{(j+0.5)\pi}{N}v] \tag{3}$$

[1] https://github.com/YCC-324/frequency_backdoor

$$c(u) = \begin{cases} \sqrt{\frac{1}{M}}, u = 0 \\ \sqrt{\frac{2}{M}}, u \neq 0 \end{cases} \qquad c(v) = \begin{cases} \sqrt{\frac{1}{N}}, v = 0 \\ \sqrt{\frac{2}{N}}, v \neq 0 \end{cases} \tag{4}$$

where DCT converts an image of size $M \times N$ from the spatial domain to the frequency domain of the same size (as shown in equation 1). Conversely, IDCT transforms the frequency domain representation of an image back to its spatial domain representation (as shown in equation 2). $F(u,v)$ represents the intensity value at position (u,v) in the frequency domain, and $f(i,j)$ represents the pixel value at position (i,j) in the spatial domain.

2.2 Backdoor Attacks

Stealthy backdoor attacks. SL models are vulnerable to backdoor attacks, commonly executed by poisoning the training dataset with samples containing a trigger pattern and relabeling them to the target class. To increase the stealthiness of backdoor attacks, several approaches have been proposed for generating invisible triggers, such as Blend [8], SIG [3], and REFOOL [24]. For example, REFOOL uses natural reflection to create the trigger. However, these methods often require the poisoned data to be mislabeled, which can be manually detected. On the other hand, some studies [39, 3] aim to achieve backdoor attacks in a clean-label scenario to avoid detection caused by mislabeling. Turner et al. [39] successfully executed backdoor attacks under the clean label using adversarial examples and GAN-generated data. The research also shows that creating triggers that cause global perturbations to the original images is necessary to achieve clean-label backdoor attacks.

Backdoor attacks in self-supervised learning. SSL, unlike SL, trains the encoder on pretext tasks that use input data as supervision to help it learn critical features of the dataset. However, backdoor attacks against SSL have started gaining attention recently. For example, Carlini et al.[4] proposed a backdoor attack against CLIP by patching a trigger on images and modifying corresponding text descriptions. Similarly, Jia et al.[19] proposed a backdoor attack on a pre-trained encoder by optimizing a function that aggregates feature vectors of images with embedded triggers into the encoder's output space. Saha et al. [32] utilized the training characteristics of contrastive learning to inject a backdoor into the model by patching a trigger on images of the target class.

Backdoor attacks in frequency domain. Recent studies have explored backdoor attacks from a frequency domain perspective, leveraging neural network interpretability in the frequency domain [12, 41, 16]. For instance, Wang et al.[41] have proposed a method that directly injects a backdoor by manipulating the intensities of specific manually selected frequencies. However, this approach is not robust to filters and has a fixed pattern in the spatial domain, which can be reverse generated. Hammoud et al.[16] have suggested finding the frequencies sensitive to the DNN model's decisions as the injection position for the frequency trigger. However, their method requires a clean model that is well-trained on the dataset used for poisoning and involves mislabeling poisoned samples during backdoored model training.

In this paper, we propose a novel approach to backdoor attacks that exploit the characteristics of the frequency domain. Our method allows injecting an invisible backdoor without mislabeling, making it effective in clean-label supervised learning and no-label self-supervised learning scenarios.

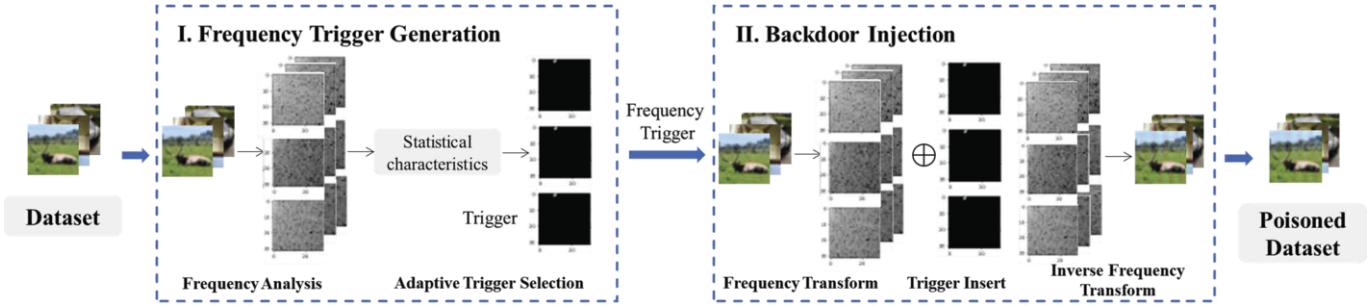

Figure 2: Overview of the frequency backdoor attack using data poisoning

2.3 Backdoor Defenses

Defenses against training data. The last hidden layer's activations reflect the high-level features used by a neural network to make predictions. To detect poisoned samples, Chen et al. [5] propose an activation clustering method where the activations of inputs with the same label are separated and clustered using k-means with $k = 2$ after dimension reduction, with one of the clusters being the poisoned samples. These samples can be removed or relabeled with the correct label. Similarly, Tran et al. [38] use the spectral signature technique, a robust statistical analysis method, to identify and remove potentially compromised training data samples that have been poisoned.

Defenses against model inputs. SentiNet [9] proposes using Grad-CAM [34] for model interpretability and object detection to detect potential attack regions of an image. Then, these regions can be manually checked to identify poisoned inputs, i.e., samples with a patched trigger. Februus [11] trains a GAN to automatically repair images after removing the suspicious areas masked by Grad-CAM. STRIP [13] detects whether an input is poisoned by superimposing various image patterns onto the input, which can cause normal input misclassified but cannot surpass the effect of the trojan trigger.

Defenses against models. Fine-Pruning [21] utilizes a pruning step to remove decoy neurons, and fine-tuning is then applied to eliminate the backdoors. Neural Cleanse [40] aims to detect whether a DNN model has been backdoored by reversing the trigger. This method has been further improved in TABOR [15] by incorporating various regularizations in the optimization process. ABS [22] identifies compromised neurons that significantly contribute to a particular label, then generates a trigger for the compromised neuron using simulation analysis and utilizes the performance of the trigger to confirm if the neuron is backdoored. MNTD [43] trains a meta-classifier using benign models and poisoned models as inputs to perform binary classification and predict if a given model is backdoored.

3 Data Poisoning with Frequency Domain

3.1 Threat Model

We consider an attacker aims to poison a dataset by patching an invisible trigger on some of the samples to implant a backdoor in any DNN model trained on the dataset, without controlling the training process. The attacker aims for three goals: effectiveness, stealthiness, and robustness. *Effectiveness* means that the attack should cause the model to misclassify samples patched with the trigger as a specific label with a high success rate while behaving similarly to the benign model when processing samples without the trigger. *Stealthiness* requires that the backdoor trigger be invisible to humans so that the poisoned samples can pass manual checks. *Robustness* represents that the trigger remains valid under common defenses, or the model's performance is significantly degraded if a defender tries to clear the trigger using defenses against the input or dataset.

3.2 Overview

Figure 2 shows an overview of our data poisoning attack, which consists of two main components: Frequency Trigger Generation and Backdoor Injection. We aim to contaminate a dataset that any DNN models trained on it will be implanted with a backdoor. To accomplish this goal, we first analyze the frequency distribution characteristics of the images in a predefined target class and choose a set of intensity values of proper frequencies in the frequency domain as triggers based on statistical features. We then design adaptive triggers in the frequency domain that meet the following criteria: imperceptible to human observers, patterns that overlap with significant portions of the images in the spatial domain, no specific pattern in the spatial domain, and effectiveness after common data preprocessing. In the second phase, we apply Fourier transform to transform the images in the target class from the spatial domain to the frequency domain and inject our frequency trigger to produce a poisoned dataset.

3.3 Trigger design

In this work, we propose a novel approach for generating an adaptive frequency trigger that takes into account the findings of previous studies [45, 44, 26]. These studies demonstrate that DNN models are sensitive to changes in frequency domain information, which can affect all the pixels of the original trigger. This means that frequency triggers can overlap the entire image and be effective in both no-label and clean-label backdoor attack scenarios [39, 3, 24]. Moreover, we take advantage of the fact that changes in the intensity of specific frequencies can be difficult for humans to perceive if they are within a threshold. Therefore, we propose using the frequency domain as an effective way to insert triggers and achieve a successful backdoor attack. Our approach generates an adaptive frequency trigger of the following form:

$$F_T(u,v) = \begin{cases} 0, & (u,v) \notin \nu_T \\ I_T(u,v) - F_n(u,v), & (u,v) \in \nu_T \end{cases} \tag{5}$$

where $F_T(u,v)$ and $F_n(u,v)$ represent the intensity of the trigger and the original image at frequency (u,v), respectively. The variable ν_T represents the frequencies we choose to modify the intensity of, while $I_T(u,v)$ represents the target intensity we aim to set for each selected frequency. The selection of these variables follows the two objectives listed below:

$$\begin{aligned} \nu_T = & \min_{(u,v)} N(Diff(F(u,v), F_{filter}(u,v))) \\ & \cap \max_{(u,v)} N(Discrete(F(u,v))) \end{aligned} \tag{6}$$

$$|I_T(u,v) - F_n(u,v)| < \varepsilon \tag{7}$$

where $F(u,v)$ and $F_{filter}(u,v)$ represent the set of intensities at frequency (u,v) of all images with the target label and the filtered images, respectively. $Diff$ calculates the average differences in intensity at frequency (u,v) among all images, $Discrete$ calculates the dispersion of intensities at frequency (u,v) among all images, $minN$ and $maxN$ represents the smallest and largest N elements, respectively, and ε is a threshold below which changes in the frequency domain are difficult for humans to perceive. Equation 6 identifies robust frequencies against filtering and defense methods based on trigger pattern detection. Equation 7 sets the intensities with high stealthiness.

Algorithm 1 presents the process of generating an adaptive trigger. Firstly, we calculate the frequency distribution of the images in the target class and select candidate frequencies that are robust to filters (line 4). Then, we select a subset of frequencies ν_T that can produce significant differences between trigger patterns on different images in the spatial domain (line 5). Secondly, for each channel, we calculate the mean value of the intensities at each frequency in ν_T among all the images as the basic intensities (line 6). Finally, we use a grid search to determine the threshold value ε and set the target intensities I_T for each frequency based on the basic intensities and the selected threshold value ε (line 7).

In generating an adaptive frequency trigger, selecting the frequencies candidate is a crucial step (lines 10-15). We start by passing each image to the filter (line 10) and calculating the average relative distance between the intensities at each frequency of the original images and those after the filter for each channel (lines 11-13). We then sort the distances on each channel in ascending order and select the top 50 frequencies ranked high on all three channels as the candidate frequencies $\nu_{candidate}$ (line 14). Finally, we select target frequencies at which the Coefficient of Variation (CoV) of intensities is the largest (lines 18-22). The reason for choosing these frequencies is explained as follows:

The CoV reflects the dispersion of the data, and the larger the CoV, the greater the dispersion. The frequency trigger in the spatial domain is formalized as

$$\begin{aligned} f_T(i,j) &= \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F_T(u,v)G(i,j,u,v) \\ &= \sum_{(u,v)\in\nu} (I_T(u,v) - F_n(u,v))G(i,j,u,v) \end{aligned} \tag{8}$$

Since $I_T(u,v)$ is a fixed value, to maximize the variance of trigger patterns (i.e., $f_T(i,j)$) on different images, we select the frequency at which intensities (i.e., $F_n(u,v)$) vary considerably among all the images. Specifically, for each channel, we calculate the CoV for each frequency in $\nu_{candidate}$ across all images (lines 18-20). Then we sort them in descending order and we pick a specified number (i.e., $TopN$) of corresponding frequencies that rank high on all three channels as the position of our frequency trigger ν_T (line 21).

We set appropriate intensities for the selected frequencies (lines 25-34). To achieve good stealthiness, we gradually increase ϵ by an interval of $Step$ until the Structural Similarity (SSIM) between the original dataset and the poisoned dataset in the spatial domain is smaller than 0.9 (lines 27-32). SSIM is a measure of similarity between two images, and an SSIM of 0.9 indicates a good level of stealthiness. Once the SSIM is below 0.9, we perform a grid search within the current interval to find the value of ϵ that yields an SSIM just above 0.9, and use Equation 7 to determine the target intensities I_T for the selected frequencies (line 33).

Algorithm 1 Adaptive frequency trigger generation algorithm

Input: D_t: dataset with the target label in frequency domain; N: the size of D_t; (C,W,H): the shape of each image; ν: frequencies of images; $TopN$: number of frequencies to select; $Step$: the step of the search;

Output: ν_T: the list of the frequencies selected; I_T: the list of the corresponding target intensities

1: $\nu = \{(1,1),(1,2),...,(W,H)\}$
2: $D_t = \{x^n\}, n \in [1,N]$
3: $x^n = \{F^n_{ch}(u,v)\}, (u,v) \in \nu, ch \in [1,C]$
4: $\nu_{candidate} = SelectFrequencyRobustToFilter(D_t)$
5: $\nu_T = SelectFrequencyDiscrete(D_t, \nu_{candidate})$
6: $I_T = GetMeanValue(D_t, \nu_T, ch), ch \in [1,C]$
7: $I_T = SetIntensities(D_t, \nu_T, I_T, Step)$
8: **return** ν_T, I_T

9: **Function** $SelectFrequencyRobustToFilter(D_t)$
10: $\{x^n_f\} = Filter(\{x^n\}), n \in [1,N]$
11: **for** (u,v) in ν **do**
12: $diff(u,v) = Diff(\{x^n\},\{x^n_f\}), n \in [1,N]$
13: **end for**
14: $\nu_{candidate} = \underset{(u,v)}{AscendingSort}(diff(u,v), 50)$
15: **return** $\nu_{candidate}$
16: **end Function**

17: **Function** $SelectFrequencyDiscrete(D_t, \nu_{candidate})$
18: **for** (u,v) in $\nu_{candidate}$ **do**
19: $CoV(u,v) = CalCov(\{x^n\}), n \in [1,N]$
20: **end for**
21: $\nu_T = \underset{(u,v)}{DescendingSort}(CoV(u,v), TopN)$
22: **return** ν_T
23: **end Function**

24: **Function** $SetIntensities(D_t, \nu_T, I_T, Step)$
25: $\epsilon = 0$
26: $\{f^n\} = IDCT(D_t), n \in [1,N]$
27: **do**
28: $\epsilon = \epsilon + Step$
29: $\{(x^{'})^n\} = AddTrigger(D_t, \nu_T, I_T, \epsilon), n \in [1,N]$
30: $\{(f^{'})^n\} = IDCT(\{(x^{'})^n\}), n \in [1,N]$
31: $SSIM = CalSSIM(\{f^n\},\{(f^{'})^n\}), n \in [1,N]$
32: **while** $SSIM > 0.9$
33: $I_T = TryValue(D_t, I_T, \epsilon)$
34: **return** I_T
35: **end Function**

3.4 Backdoor Injection

After generating the frequency domain backdoor trigger, we can implant it into the target class images to create the poisoned dataset. Firstly, we transform the images in the target class from the spatial domain to the frequency domain using the Discrete Cosine Transform (DCT). Secondly, we inject our trigger into the images presented in the frequency domain by adding the intensity values of the trigger to that of images at the corresponding frequencies. Finally, we use the Inverse Discrete Cosine Transform (IDCT) to return the images to the spatial domain. Then we mix the poisoned images with other clean images to create a poisoned dataset. The models trained on this dataset will be vulnerable to backdoor attacks.

4 Experiment

4.1 Experiment Settings

Dataset and model. We evaluate the effectiveness of our backdoor attack on two classic deep neural network (DNN) models: ResNet-18 [17] and Densenet [18], across popular datasets: CIFAR-10 [20], STL-10 [10], GTSRB [35], and VGGFace [27].

•*CIFAR-10* is a widely-used object classification dataset that contains 60,000 color images in 10 classes, each of size $32 \times 32 \times 3$, with 50,000 images for training and 10,000 for testing.

•*STL-10* is an image recognition dataset designed for developing unsupervised feature learning, deep learning, and self-taught learning algorithms. It consists of 5,000 labeled training images, 1,000 labeled testing images, and 100,000 unlabeled images, all of size $96 \times 96 \times 3$, distributed among 10 classes.

•*GTSRB* is a widely-used dataset in the field of autonomous driving, consisting of 43 classes of traffic signs. It contains 39,209 training images and 12,630 test images, with each image having a size of $32 \times 32 \times 3$.

•*VGGFace* is a widely-used dataset in the field of face recognition, consisting of 2 million images from 2,622 different identities. For our experiments, we resized each image to $224 \times 224 \times 3$.

Attack Scenarios. To evaluate our attack approach, we benchmark it under both self-supervised learning and supervised learning scenarios. For self-supervised learning, we pre-train ResNet-18 as an image encoder on the poisoned CIFAR-10 dataset using two widely used methods, SimCLR [6] and MoCO v2 [7]. We then use the encoder for the downstream datasets CIFAR-10, STL-10, and GTSRB to train downstream classifiers. For supervised learning, we train ResNet-18 and Densenet on the poisoned CIFAR-10, STL-10, and VGGFace datasets. To implement self-supervised learning, we use the implementation shown in [31] for SimCLR and [30, 7] for MoCO v2, and use their default training parameters and data transformations. For supervised learning, we apply commonly used parameters and data transformations to avoid disturbing the normal training process.

Metrics. We use the following four metrics to evaluate our approach:

•*Clean Data Performance (CDP)* is a metric that evaluates the proportion of correctly classified clean samples, i.e., samples without any backdoor triggers, to their ground-truth classes.

•*Attack Success Rate (ASR)* measures the fraction of poisoned images that are successfully predicted as the target label specified by the backdoor attack.

•*Peak Signal-to-Noise Ratio (PSNR)* is a metric that quantifies the fidelity of image representation after processing by measuring the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the image quality.

•*Structural Similarity (SSIM)* is a metric that measures the similarity between two images, taking into account critical perceptual phenomena, e.g., luminance masking, contrast masking. Compared to PSNR, SSIM aligns more with human perception of image similarity.

Platform. All of our experiments were conducted on a server running a 64-bit Ubuntu 20.04.3 system with an Intel(R) Xeon(R) Platinum 8268 CPU @ 2.90GHz, 188GB of RAM, a 20TB hard drive, and one Nvidia GeForce RTX 3090 GPU with 24GB of memory.

4.2 Effectiveness

Baseline Performance. In the SSL scenario, we trained ResNet-18 DNN encoders on CIFAR-10 using two SSL techniques, namely SimCLR and MoCo V2. The trained encoders were then used to train downstream classifiers on CIFAR-10, STL-10, and GTSRB tasks. Similarly, in the SL scenario, we trained clean ResNet-18 and Densenet models on CIFAR-10, STL-10, and VGGFace. We assessed their performance as the baseline in Table 1 across several metrics, which demonstrates that the clean models perform similarly to the models presented in [6, 7, 17]. In addition, the ASR, measured on the poisoned dataset (with the target label set to "automobile", "speed limit 30", and "car" for the three datasets respectively), was low for all clean models. This observation demonstrates that our trigger has little effect on the main task's decision.

Table 1: Baseline of Clean Models

Model	Dataset	CDP	ASR
ResNet-18 (SimCLR)	CIFAR-10	86.53%	9.31%
	GTSRB	91.38%	12.6%
	STL-10	77.21%	2.70%
ResNet-18 (MoCo v2)	CIFAR-10	84.03%	9.29%
	GTSRB	89.64%	8.38%
	STL-10	72.65%	3.11%
ResNet-18	CIFAR-10	91.06%	9.96%
	STL-10	74.42%	5.46%
	VGGFace	85.31%	1.08%
Densenet	CIFAR-10	88.21%	9.40%
	STL-10	74.71%	14.17%
	VGGFace	87.17%	1.17%

Table 2: Effectiveness of Backdoor Attack

Model	Dataset	CDP	ASR
ResNet-18 (SimCLR)	CIFAR-10	85.48%(-1.05%)	92.08%
	GTSRB	89.67%(-1.71%)	95.57%
	STL-10	76.76%(-0.45%)	99.68%
ResNet-18 (MoCo v2)	CIFAR-10	82.24%(-1.79%)	91.55%
	GTSRB	87.90%(-1.74%)	94.60%
	STL-10	71.83%(-0.82%)	99.78%
ResNet-18	CIFAR-10	90.63%(-0.43%)	95.56%
	STL-10	72.51%(-1.91%)	92.45%
	VGGFace	84.22%(-1.09%)	92.28%
Densenet	CIFAR-10	87.09%(-1.12%)	90.05%
	STL-10	73.01%(-1.70%)	90.11%
	VGGFace	85.88%(-1.29%)	91.52%

Backdoor Performance. We evaluated our backdoor attack on the same tasks as those used for the baseline evaluation. To generate triggers, we employed our proposed adaptive Algorithm 1. For instance, we selected "automobile" as the poisoned label in CIFAR-10 dataset, and the generated trigger consisted of frequencies (1,10), (1,9), and (0,10), with the intensities on three channels set to (70,70,80), (65,65,65), and (65,65,65), respectively. Table 2 presents the backdoor performance on self-supervised learning and supervised learning. The experimental results demonstrate that our method achieved satisfactory ASR (i.e., above 90%) on all the tasks with an acceptable reduction in performance on the clean samples (i.e., lower than 2%).

Note that in our experiments on SSL, we did not know in advance which label in the downstream task would be poisoned. The trigger was applied after the model was deployed, and we only discovered the corresponding target label once the backdoor was activated. As a result, we achieved an untargeted backdoor attack in this scenario. However, we made an interesting discovery when transferring the encoder pre-trained on CIFAR-10 to the downstream task STL-10. We found that the target label was the same as the poisoned label used during the training the poisoned encoder. This is because the features of the trigger are bound to the features of the pre-training samples with the poisoned label and can also be bound to downstream samples with similar features after transferring. Therefore, we can also achieve a targeted backdoor attack as long as the downstream dataset and our poisoned dataset have overlapping categories.

Table 3: Evaluation of different intensities settings.

No.	Itensities	PSNR	SSIM	CDP	ASR	ASR(Filter)
1	(60,60,70),(55,55,55),(55,55,55)	25.03	0.9122	91.00%	87.84%	79.80%
2	(70,70,80),(65,65,65),(65,65,65)	24.11	0.9024	90.63%	95.56%	86.12%
3	(80,80,90),(75,75,75),(75,75,75)	23.23	0.8911	90.77%	91.62%	78.80%

Note: The intensities increase from No.1 to No.3.

Table 4: Evaluation of different frequencies settings

No.	Frequencies	Itensities	PSNR	SSIM	CDP	ASR	ASR(Filter)
1	(28,0),(30,0),(31,0)	(40,35,30),(25,25,25),(25,25,25)	31.20	0.9530	90.29%	97.15%	4.32%
2	(1,10),(1,9),(0,10)	(70,70,80),(65,65,65),(65,65,65)	24.11	0.9024	90.63%	95.56%	86.12%
3	(1,7),(3,6),(5,3)	(95,105,130),(85,85,85),(85,85,85)	21.57	0.8624	89.00%	88.18%	84.68%

Note: The frequencies decrease from No.1 to No.3, and the intensities increase to ensure ASR is larger than 85% and CDP is around 90%.

Table 5: Comparison with different stealthy backdoor attacks

Attack method	PSNR	SSIM	CDP	ASR
Blend[8]	19.18	0.7921	89.77%	93.11%
SIG[3]	25.12	0.8988	89.45%	95.76%
REFOOL[24]	16.59	0.7701	88.80%	92.80%
Ours	24.11	0.9024	90.63%	95.56%

Table 6: Comparision with other frequency backdoor attacks

Method	CDP	ASR	CDP(Filter)	ASR(Filter)
FTrojan	83.71%	99.90%	57.93%	5.4%
FIBA	83.65%	81.41%	82.08%	76.70%
ours	90.63%	95.56%	57.26%	86.12%

Invisibility. We assessed the invisibility of our trigger by measuring the average PSNR and SSIM values for CIFAR-10 images patched with the trigger. The calculated PSNR and SSIM values were 24.11 and 0.9024, respectively. We also compared the inconspicuousness of our trigger with existing techniques, and the results presented in Table 5 demonstrate that our approach outperforms Blend and REFOOL. Although SIG yields similar performance, it uses a fixed trigger pattern, namely, a horizontal sinusoidal signal, that can be detectable by defenses that employ trigger reverse generation.

Comparison with other frequency backdoor attacks. We conducted experiments to compare our approach with the methods proposed in FIBA [12] and FTrojan [41]. Specifically, we trained ResNet-18 on the poisoned CIFAR-10 dataset and evaluated the attack performance of these methods. The results are shown in Table 6. We observed that FTrojan is not robust to filtering, and the ASR of FIBA is low, only reaching 81.41%. We used the code provided by the authors for our experiments, and due to differences in data preprocessing, their clean data accuracy benchmark was not directly comparable to ours. Overall, our approach outperforms these methods in terms of ASR and robustness.

4.3 Impacts of Intensities and Frequencies

Impacts of intensities. To evaluate the influence of trigger intensities, we selected triggers with lower and higher intensities and injected them into the CIFAR-10 dataset. As shown in Table 3, triggers with higher intensities were more effective in achieving the backdoor attack (i.e., higher ASR), but at the cost of reduced stealthiness (i.e., lower PSNR and SSIM).

Impacts of frequencies. We conducted experiments on triggers with higher and lower frequencies and adjust the intensities to achieve an ASR higher than 85% and a CDP of around 90%. The results in Table 4 indicate that a trigger with higher frequencies is more detectable by DNNs, albeit slightly more stealthy than the trigger selected by our proposed method. However, it is more prone to being filtered out by filters. On the other hand, a trigger with lower frequencies is more challenging for DNNs to learn, requiring slightly higher intensities that may introduce visible changes to images in the spatial domain, and thus negatively impact the performance of the clean dataset.

4.4 Resistance

We evaluate the robustness of our attack against defenses that are most relevant to our attack. These defenses include the detection of training data (i.e., Activation Clustering), preprocessing of inputs (i.e., Filter), detection of inputs (i.e., SentiNet, STRIP), and detection of models (i.e., Fine-Pruning, Neural Cleanse, ABS, MNTD).

Resistance to Activation Clustering. Activation Clustering detects poisoned data by detecting differences in activation distributions between clean and poisoned inputs. However, this method can only be used on samples with target labels and is, therefore, unsuitable for SSL scenarios. We evaluated the effectiveness of Activation Clustering on our backdoored ResNet-18 model trained on the poisoned CIFAR-10 dataset for supervised learning. The false positive rate (i.e., the ratio of clean samples that were misclassified to be poisoned) was 100.00%, and the false negative rate (i.e., the ratio of poisoned samples that were regarded as benign) was 56.64%. The results indicate that although many poisoned samples were successfully detected, all benign samples were classified as malicious, leading to a significant decrease in DNN model performance. Additionally, since all samples with the target label left are poisoned, the backdoor can still be injected. Furthermore, nearly half of the poisoned samples were classified as benign, which means that the activation difference between benign and poisoned samples was insignificant, indicating a high level of stealthiness in our backdoor attack.

Resistance to Filter. Considering the frequency domain characteristics of the backdoor, filters can pose a significant threat to the backdoor trigger. We evaluated the robustness of our attack to filters on the backdoored ResNet-18 model trained on the poisoned CIFAR-10 dataset using supervised learning. Before predicting the test samples using the model, we passed them through four filters, including Gaussian, average, median, and SVD filters. It should be noted that the SVD filter filters out singular frequencies by analyzing the frequency distribution in the frequency domain. The results presented in Table 7 demonstrate that our backdoor attack maintains a high ASR even after being processed by filters. Still, the performance on the clean data significantly drops. Therefore, our backdoor attack is robust to filter processing.

Resistance to SentiNet. SentiNet, which uses Grad-CAM for model interpretability and object detection, can be employed to detect potential attack regions in an image. These regions can then be manu-

Table 7: Resistance to different filters

Filter	CDP	ASR
Gaussian Filter	57.26%	86.12%
Mean Filter	51.84%	79.23%
Median Filter	76.50%	88.66%
SVD Filter	74.33%	93.52%

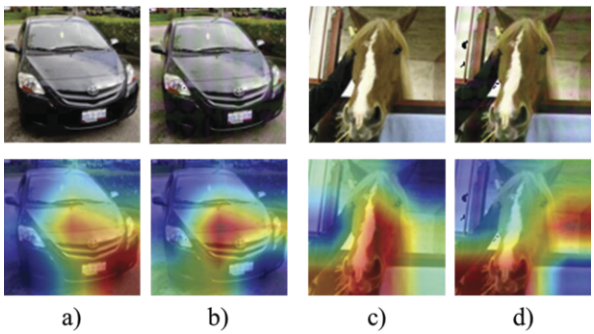

Figure 3: Critical Regions Identified by SentiNet. Columns a) and c) are the results of clean images, and columns b) and d) are the results of the corresponding poisoned images

ally checked to identify poisoned inputs containing a patched trigger. We applied SentiNet to the backdoored ResNet-18 models trained on the poisoned STL-10 dataset using SSL with SimCLR to determine if the trigger could be detected. The results in Figure 3 show the regions identified by SentiNet on several randomly selected samples, where columns a) and c) represent the clean images and columns b) and d) represent the corresponding poisoned images. The results suggest that SentiNet can only identify a partial area of an image as a critical region while our trigger overlaps the entire image. Additionally, our trigger effectively shifts the model's focus on the image compared to the clean image. However, the regions identified on the poisoned images are still important for the DNN models to identify the original images, indicating that our trigger is difficult to detect by human inspection.

Resistance to STRIP. We evaluated the robustness of our backdoor attack against STRIP on the backdoored model trained on CIFAR-10 via self-supervised learning. The result showed that STRIP failed to detect our poisoned input samples, with a false negative rate of 97% and a false positive rate of 3.05%. This indicates that our attack is effective against STRIP. This could be because when STRIP overlaps two images, the frequency distribution changes, which causes the trigger patched on the image to lose its effect, making it difficult for STRIP to detect the backdoor.

Resistance to Fine-Pruning. We evaluated our attack against Fine-Pruning on the backdoored model trained on CIFAR-10 using self-supervised learning. At a 50% pruning rate, we observed a reduction in ASR to 84.22% and CDP to 69.07%. However, after 50 epochs of finetuning, the CDP and ASR stabilized at around 85.5% and 80%, respectively, indicating that our backdoor attack can effectively bypass this defense mechanism. One possible reason for this is that our backdoor attack affects many neurons in the DNN, making it more resilient to pruning.

Resistance to Neural Cleanse. We apply Neural Cleanse to detect our backdoored ResNet-18 model trained on CIFAR-10 using SimCLR. Neural Cleanse tries to find a potential minimal trigger that can misclassify all samples into a target label and uses an outlier detection algorithm to choose the trigger significantly smaller than the others as the real trigger. The corresponding label is the target label of

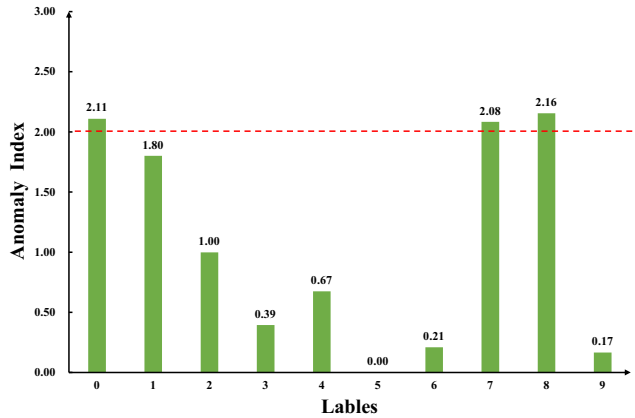

Figure 4: Anomaly Index of Neural Cleanse. The poisoned label is "1", but the labels "0", "7", and "8" are identified, indicating Neural Cleanse fails to detect our backdoor.

the backdoor attack. However, our frequency trigger is not detected by Neural Cleanse. The anomaly indexes of labels "0", "7", and "8" are larger than 2, but the poisoned label is "1". This shows that Neural Cleanse cannot effectively detect our backdoor attack due to the absence of a specific trigger pattern.

Resistance to ABS. We evaluated our backdoor attack against ABS on the backdoored model trained on CIFAR-10 using self-supervised learning. ABS aims to detect suspicious neurons for all the labels. We found that for all the reversed triggers, the attack success rates were less than 15%, indicating that ABS cannot reverse-engineer our trigger. This is because there is no specific trigger pattern.

Resistance to MNTD. To evaluate the effectiveness of our backdoor attack against MNTD, we generated 20 backdoored CIFAR-10 models by injecting our backdoor into 20 clean models using the code provided by MNTD. We then used the MNTD's meta-classifier to detect the backdoored models among these 40 models, achieving an AUC score of 0.5870 and an accuracy of 55.81%. These results indicate that our attack can successfully evade this defense. We hypothesize that this may be due to our backdoor allowing the model to learn in different feature spaces, which makes it difficult for MNTD to distinguish these features.

Potential Defense. Due to the potential presence of anomalous values in the frequency domain caused by our triggers, anomaly detection techniques on the frequency domain may be able to detect our triggers. However, since our triggers are dispersed across different frequency bands, which are entangled with the core information of the image, it might be difficult to completely remove the backdoor without compromising the main task.

5 Conclusion

In this paper, we propose a novel backdoor attack approach based on the frequency domain, which implants a backdoor into DNN models trained on a poisoned dataset without mislabeling any samples or accessing the training process. We evaluate the effectiveness of our approach in both the no-label and clean-label cases on popular benchmark datasets using self-supervised and supervised learning. Furthermore, we assess the efficacy of existing defenses against our backdoor attack. The experimental results demonstrate that our approach can achieve a high attack success rate without causing significant performance degradation on the main tasks and could evade commonly used defense methods.

Acknowledgements

The IIE authors are supported in part by Beijing Natural Science Foundation (No.M22004), NSFC (92270204), Youth Innovation Promotion Association CAS and a research grant from Huawei.

References

[1] Zhongxin Bai and Xiao-Lei Zhang, 'Speaker recognition based on deep learning: An overview', *Neural Networks*, **140**, 65–99, (2021).

[2] Baidu. Baidu apollo: Open source autonomous driving. https://github.com/ApolloAuto/apollo, 2017.

[3] Mauro Barni, Kassem Kallas, and Benedetta Tondi, 'A new backdoor attack in cnns by training set corruption without label poisoning', *2019 IEEE International Conference on Image Processing (ICIP)*, 101–105, (2019).

[4] Nicholas Carlini and A. Terzis, 'Poisoning and backdooring contrastive learning', *ArXiv*, **abs/2106.09667**, (2021).

[5] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Ben Edwards, Taesung Lee, Ian Molloy, and B. Srivastava, 'Detecting backdoor attacks on deep neural networks by activation clustering', *ArXiv*, **abs/1811.03728**, (2019).

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, 'A simple framework for contrastive learning of visual representations', in *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, (2020).

[7] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He, 'Improved baselines with momentum contrastive learning', *ArXiv*, **abs/2003.04297**, (2020).

[8] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Xiaodong Song, 'Targeted backdoor attacks on deep learning systems using data poisoning', *ArXiv*, **abs/1712.05526**, (2017).

[9] Edward Chou, Florian Tramèr, and Giancarlo Pellegrino, 'Sentinet: Detecting localized universal attacks against deep learning systems', *2020 IEEE Security and Privacy Workshops (SPW)*, 48–54, (2020).

[10] Adam Coates, A. Ng, and Honglak Lee, 'An analysis of single-layer networks in unsupervised feature learning', in *AISTATS*, (2011).

[11] Bao Gia Doan, Ehsan Abbasnejad, and Damith Chinthana Ranasinghe, 'Februus: Input purification defense against trojan attacks on deep neural network systems', *Annual Computer Security Applications Conference*, (2020).

[12] Yu Feng, Benteng Ma, Jing Zhang, Shanshan Zhao, Yong Xia, and Dacheng Tao, 'Fiba: Frequency-injection based backdoor attack in medical image analysis', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20876–20885, (June 2022).

[13] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal, 'Strip: A defence against trojan attacks on deep neural networks', in *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113–125, (2019).

[14] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg, 'Badnets: Evaluating backdooring attacks on deep neural networks', *IEEE Access*, **7**, 47230–47244, (2019).

[15] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Xiaodong Song, 'Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems', *ArXiv*, **abs/1908.01763**, (2019).

[16] Hasan Abed Al Kader Hammoud and Bernard Ghanem, 'Check your other door! creating backdoor attacks in the frequency domain', (2021).

[17] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778, (2016).

[18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger, 'Densely connected convolutional networks', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017).

[19] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong, 'Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning', in *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 2043–2059, (2022).

[20] Alex Krizhevsky, Geoffrey Hinton, et al., 'Learning multiple layers of features from tiny images', (2009).

[21] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg, 'Fine-pruning: Defending against backdooring attacks on deep neural networks', in *Research in Attacks, Intrusions, and Defenses*, pp. 273–294, (2018).

[22] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and X. Zhang, 'Abs: Scanning neural networks for back-doors by artificial brain stimulation', *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, (2019).

[23] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and X. Zhang, 'Trojaning attack on neural networks', in *NDSS*, (2018).

[24] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu, 'Reflection backdoor: A natural backdoor attack on deep neural networks', in *ECCV*, (2020).

[25] Chenxu Luo, Xiaodong Yang, and Alan Loddon Yuille, 'Self-supervised pillar motion learning for autonomous driving', *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3182–3191, (2021).

[26] Tao Luo, Zheng Ma, Zhi-Qin John Xu, and Yaoyu Zhang, 'Theory of the frequency principle for general deep neural networks', *ArXiv*, **abs/1906.09235**, (2021).

[27] O. M. Parkhi, A. Vedaldi, and A. Zisserman, 'Deep face recognition', in *British Machine Vision Conference*, (2015).

[28] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman, 'Deep face recognition', (2015).

[29] M. Ratner, 'Fda backs clinician-free ai imaging diagnostic tools', *Nature Biotechnology*, **36**(8), 673–674, (2018).

[30] Hao Ren. Moco. https://github.com/leftthomas/MoCo, 2020.

[31] Hao Ren. Simclr. https://github.com/leftthomas/SimCLR, 2020.

[32] Aniruddha Saha, Ajinkya Tejankar, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash, 'Backdoor attacks on self-supervised learning', in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13327–13336, (2022).

[33] Florian Schroff, Dmitry Kalenichenko, and James Philbin, 'Facenet: A unified embedding for face recognition and clustering', *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823, (2015).

[34] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, 'Grad-cam: Visual explanations from deep networks via gradient-based localization', in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, (2017).

[35] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel, 'The german traffic sign recognition benchmark: a multi-class classification competition', in *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, (2011).

[36] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, 'Intriguing properties of neural networks', *Computer Science*, (2013).

[37] Ruijie Tao, Kong-Aik Lee, Rohan Kumar Das, Ville Hautamaki, and Haizhou Li, 'Self-supervised speaker recognition with loss-gated learning', (2021).

[38] Brandon Tran, Jerry Li, and Aleksander Madry, 'Spectral signatures in backdoor attacks', in *NeurIPS*, (2018).

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

[40] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao, 'Neural cleanse: Identifying and mitigating backdoor attacks in neural networks', *2019 IEEE Symposium on Security and Privacy (SP)*, 707–723, (2019).

[41] Tong Wang, Yuan Yao, F. Xu, Shengwei An, Hanghang Tong, and Ting Wang, 'An invisible black-box backdoor attack through frequency domain', in *European Conference on Computer Vision*, (2022).

[42] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song, 'Generating adversarial examples with adversarial networks', in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, p. 3905–3911. AAAI Press, (2018).

[43] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, and Bo Li, 'Detecting ai trojans using meta neural analysis', in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 103–120, (2021).

[44] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yan Xiao, and Zheng Ma, 'Frequency principle: Fourier analysis sheds light on deep neural networks', *ArXiv*, **abs/1901.06523**, (2020).

[45] Zhi-Qin John Xu, Yaoyu Zhang, and Yan Xiao, 'Training behavior of deep neural network in frequency domain', in *ICONIP*, (2019).

[46] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen, 'Seeing isn't believing: Towards more robust adversarial attack against real world object detectors', in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1989–2004, (2019).