Hindawi
Security and Communication Networks
Volume 2023, Article ID 9308909, 12 pages
https://doi.org/10.1155/2023/9308909

Research Article

DeepDefense: A Steganalysis-Based Backdoor Detecting and Mitigating Protocol in Deep Neural Networks for AI Security

Lei Zhang[1], **Ya Peng**[1], **Lifei Wei**[2], **Congcong Chen**[1], **and Xiaoyu Zhang**[3]

[1]*College of Information Technology, Shanghai Ocean University, Shanghai 201306, China*
[2]*College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China*
[3]*State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an 710071, Shaanxi, China*

Correspondence should be addressed to Lifei Wei; lfwei@shmtu.edu.cn

Received 8 September 2022; Revised 13 October 2022; Accepted 24 November 2022; Published 9 May 2023

Academic Editor: Debiao He

Backdoor attacks have been recognized as a major AI security threat in deep neural networks (DNNs) recently. The attackers inject backdoors into DNNs during the model training such as federated learning. The infected model behaves normally on the clean samples in AI applications while the backdoors are only activated by the predefined triggers and resulted in the specified results. Most of the existing defensing approaches assume that the trigger settings on different poisoned samples are visible and identical just like a white square in the corner of the image. Besides, the sample-specific triggers are always invisible and difficult to detect in DNNs, which also becomes a great challenge against the existing defensing protocols. In this paper, to address the above problems, we propose a backdoor detecting and mitigating protocol based on a wider separate-then-reunion network (WISERNet) equipped with a cryptographic deep steganalyzer for color images, which detects the backdoors hiding behind the poisoned samples even if the embedding algorithm is unknown and further feeds the poisoned samples into the infected model for backdoor unlearning and mitigation. The experimental results show that our work performs better in the backdoor defensing effect compared to state-of-the-art backdoor defensing methods such as fine-pruning and ABL against three typical backdoor attacks. Our protocol reduces the attack success rate close to 0% on the test data and slightly decreases the classification accuracy on the clean samples within 3%.

1. Introduction

Deep neural networks (DNNs) have a wide range of the current applications in the artificial intelligence applications such as image recognition, speech recognition, and natural language processing [1–3], in which security and privacy protection are considerable issues [4]. The massive amount of data and growing computing power have facilitated the development of DNNs, but the DNN models are still very expensive in training. Users often choose to train DNN models on the third-party platforms (e.g., Amazon EC2) or even use third-party trained models directly to reduce training costs. However, it is vulnerable to backdoor attacks, which can misclassify any input using attacker predefined triggers (pattern patches) and replace the corresponding label with a predefined target label. Those models with backdoors behave normally just like the clean peer-to-peer models for clean samples without triggers, which are equivalent to highly stealthy viruses that disguise themselves as normal and perform great damage [5].

The backdoor attack greatly threatens DNNs in practical applications for reducing the trustworthiness of the DNN models and even leading to safety-critical areas. The separation of data and model training in deep learning allows attackers to often gain and modify the training samples to mislead DNNs by adding some invisible perturbations to a small proportion of datasets, such as the local patches or the steganographic data in the lower right corner of an image, and even setting weights that affect the model during training [6–10]. The ability of infected DNN models to correctly classify clean samples makes it difficult for users to detect the presence of backdoors. In addition, the hidden nature of triggers makes it difficult for users to identify them. Thus, the invisibility and stealthiness of triggers make detecting backdoor attacks a considerable challenge [11–13].

Most of the existing backdoor defensing methods are divided into two types: model-based defense and data-based defense. The former detects whether the model is infected by a backdoor, and the latter considers whether the data contain a trigger. Recently, Li et al. [14] reveal that existing backdoor attacks were easily mitigated by current defenses [15–17] mostly because their backdoor triggers are sample-agnostic, i.e., different poisoned samples contain the same trigger no matter what trigger pattern is adopted. Thus, they propose an attack method, called as *sample-specific backdoor attack* (SSBA), which makes it more difficult to detect and remove the backdoors since most of the current defensing protocols reconstruct and detect backdoor triggers according to the same behavior on different poisoned samples [15–17]. SSBA is an invisible backdoor attack that generates invisible sample-specific triggers by the pretrained encoder-decoder network. The reason why current mainstream defensing methods have difficulty in detecting sample-specific triggers is that their success based on the assumption that the triggers are sample-agnostic based types. For example, pruning-based defenses assume that the neurons associated with the backdoor are different from those activated by the clean samples. The defender can remove the hidden backdoor by pruning out the potential neurons. However, the non-overlap between the two neurons is that the sample-agnostic trigger pattern is simple, and the DNNs only need a few independent neurons to encode this trigger. This assumption might be easily broken when the trigger is sample-specific.

Inspired by image steganalysis technique [18], we find that the intensity values of the images at the same position of different color channels have a strong correlation for the poisoned images regardless of whether the triggers are sample-specific or invisible; that is, the triggers in the poisoned images belong an additional perturbation with a weak correlation among those color channels. In addition, since the poisoned samples of the backdoor attack are bounded to the target label, the correlation between the trigger pattern and the target label can be effectively broken by randomizing the class target.

We propose a new backdoor detecting and removing protocol, which can detect backdoors regardless of whether the triggers are specific to poisoned samples or not. Specifically, it detects whether a color image contains a trigger by the feature that the additional perturbation can be retained in the wider separate-then-reunion network (WISERNet). To address the weakness that poisoned samples in backdoor attacks are always bounded to the target label, our protocol breaks the correlation between the trigger pattern and the target label by backdoor unlearning and leads to model purification. In summary, our contributions are as follows:

(i) A backdoor defensing method based on secure image steganalysis is proposed. The poisoned image contains a trigger that can be considered as an additional perturbation, and the intensity value at the same location has a strong correlation between different color channels, while the trigger has a weak correlation between its channels. The protocol is proved valid whether the trigger is visible or invisible.

(ii) A secure backdoor detecting and removing protocol is designed. We design a novel protocol to achieve the goal by detecting the poisoned images in the training dataset based on the wider separate-then-reunion network regardless of whether the trigger is specific to the poisoned samples and by retraining the model for backdoor unlearning with the detected poisoned images.

(iii) Extensive experiments are conducted in the proposed protocol. We empirically show that our protocol is robust against three state-of-the-art backdoor attacks. Compared with the state-of-the-art backdoor defensing protocols, fine-pruning [15] and ABL [19], our protocol reduces the success rate of backdoor attacks to nearly 0% on both target classification and face recognition tasks and retains the accuracy after removing the backdoors.

2. Related Work

2.1. Backdoor Attacks. A common method for implementing backdoor attacks is data poisoning. When the model is training, the poisoned samples are injected into the training dataset. After that, the model is influenced by the poisoned samples, deviates from the desired training effect of the original training data, and changes "slightly" in the desired direction according to the feature of the poisoned samples, which allows the attacker to modify the model and implant a backdoor [20]. According to the visibility of trigger, backdoor attacks based on data poisoning can be classified into two categories: visible backdoor attack and invisible backdoor attack.

2.1.1. Visible Backdoor Attack. Gu et al. [21] first proposed the backdoor attack BadNets to inject backdoors by modifying part of the training data, whose triggers can be of arbitrary shapes, such as squares. Chen et al. [22] first demonstrated that data poisoning attacks can create physically implemented backdoors. Liu et al. [23] proposed a Trojan attack to design triggers based on the values of internal neurons in DNNs, which strengthens the connection between the trigger and the internal neurons, enabling the effect of implant backdoors with fewer poisoned samples. Chen et al. [24] improve the steganography of the trigger by combining generative adversarial network techniques to implant the trigger as a watermark into clean samples and reducing the variability between the trigger features and the clean sample features. There are many other works [25, 26] implemented in optimizing triggers, and although all of these attack methods have high success rates, the triggers are visible and can be easily detected by people.

2.1.2. Invisible Backdoor Attack. Zeng et al. [27] proposed that poisoned samples can be identified by frequency information and constructed frequency invisible poisoned samples, thus achieving the invisibility of triggers. Li et al. [14] proposed to generate sample-specific triggers by the pretrained encoder-decoder network. Considering the steganography perspective, Li et al. [28] proposed an optimized framework to constrain the generation of triggers by regularization and embed the triggers in the bit space using image steganography to make the triggers invisible.

2.2. Backdoor Defenses. Due to the great potential damage of backdoor attacks to artificial intelligence applications, an increasing number of backdoor defensing protocols are proposed to mitigate such security threats. The existing defensing approaches include model-based defense and data-based defense.

2.2.1. Model-Based Defenses. Model-based defense is to detect whether a model is infected with backdoors. Liu et al. [15] found that neurons associated with backdoors are usually dormant during inference of benign samples and therefore proposed to prune the associated backdoor neurons to eliminate backdoors in the model. Zhao et al. [29] proposed to repair infected models using quantitative clean samples by pattern connectivity techniques [30]. Liu et al. [31] proposed a neural network-based artificial intelligence scanning technique inspired by EBS [32] to determine whether a model has a backdoor; however, it is effective for single-trigger attacks and ineffective for multitrigger attacks. Wang et al. [17] proposed a defense method called neural cleanse (NC) by synthesizing each class's triggers and comparing the triggers' size. If the smaller trigger is significantly smaller than the other triggers, the model is considered to be infected with a backdoor. Recently, Li et al. [19] proposed the concept of antibackdoor and designed a generic antibackdoor learning protocol ABL, which can automatically prevent backdoor attacks during model training.

2.2.2. Data-Based Defenses. Data-based defense is to detect whether a sample contains a trigger. Gao et al. [16] proposed a method, known as the STRIP, to filter malicious samples by overlaying various images onto the images of training samples and observing the randomness of their classification results. Bao et al. [33] proposed an image preprocessing method to identify the trigger region using GardCAM [34] technique, remove it, and replace it with a neutral-colored box because the region where the triggers in the poisoned samples are located has a high impact on the model inference stage. Udeshi et al. [35] proposed to make a trigger interceptor using the dominant color of the image for locating and removing backdoor triggers in poisoned samples. Han et al. [36] proposed an evaluation framework to preprocess the input samples using data enhancement techniques to disrupt the connection between the backdoor and the trigger in the poisoned sample, making the triggers invalid during inference, and fine-tuning the infection model using another data enhancement technique to eliminate the effect of backdoors.

Liu et al. [15] proposed the approach, named as *fine-pruning* (short for FP), which has a degraded defense performance for different models and datasets. Li et al. [19] proposed a more complex implementation of antibackdoor learning, which divides the model training stages into two stages: backdoor isolation and backdoor unlearning, and the choice of a turn-period from its backdoor isolation process to backdoor unlearning progress is more critical. For different attack methods and data sets, the choice of the turn-period also has different effects on the performance of the model. Our protocol performs well for different datasets, models, and attack methods.

3. Overview

In this section, we define our attack model, give the assumptions and goals of defensing protocols, and, finally, provide an intuitive overview of our approach for identifying and mitigating backdoor attacks.

3.1. Attack Models and Defense Assumption. In our attack model, the user trains a DNN model on the training dataset, denoted as D_{train}, that can be obtained from a third party, or even the training process of the DNN can be outsourced to an untrustworthy third party. An attacker may poison part of the training data, set the size and position of the triggers at will, and adjust the training stage of the model, but not access the validation dataset and manipulate the inference stage of the model. The attacker's goal is to return to the user a trained infected backdoor model that behaves like the uninfected model in terms of the output on the clean samples but classifies into the target label specified by the attacker when the samples contain the triggers.

The attacker assumed in our work is more powerful. The attacker proposed by Li et al. [14] can only access the training dataset and cannot manipulate the training stage of the model. The attacker proposed by Liu et al. [23] cannot access the training data and can only modify the trained model. The attacker defended in our work not only has access to the training dataset but also can manipulate the training stage of the model. It is reasonable for the attacker to consider an attacker with limited capabilities. However, the attacker should be assumed to be more powerful since advances in technology and defense methods.

We also assumed that the defender has access to the trained DNN model and can use a clean set of samples to test the performance of the model.

3.2. Design Goals. Our defensing protocol includes two specific goals:

(i) Backdoor detecting: After the training stage of the DNN model, a backdoor detector constructed by WISERNet can successfully detect whether a sample image contains a trigger, i.e., whether it is a poisoned image.

(ii) Backdoor mitigating: Since there is a strong correlation between triggers and target labels in backdoor attacks, this weakness is exploited to reinput the poisoned samples into the infected model and retrain the model to achieve backdoor unlearning.

3.3. Design Intuition. We describe our high-level intuition for detecting triggers in poisoned samples and overview our defense.

3.3.1. Key Intuition. The invisibility of the trigger and the low poisoning rate make it difficult for the defender to detect whether the sample is poisoned or not. We derive the intuition behind our technique from the basic properties of a backdoor trigger, namely that whether the trigger is invisible or not, it can be regarded as additional noise, and this noise can be a special pattern or a string representing the target label. For a poisoned image, the intensity values of the three bands at the same position exhibit a strong correlation, and their expectations are similar from the perspective of statistics. On the contrary, the additional noise added in the poisoned sample has a weaker correlation between the bands and may not even correlate.

To verify the above statement, we analyzed 10,000 poisoned images generated in BadNets [21], Blend Attack [22], and SSBA [14]. Given $X = \{0, 1, \ldots, 255\}^{(C\times W\times H)}$, a poisoned image of the size of $W \times H$, it comprises three bands, namely the red, the green, and the blue band. The correlation between the different bands of the poisoned image is defined as follows:

$$\mathrm{Corr}_{j,k} = \frac{\sum_{i=1}^{K}\left(M_i - \overline{M}\right)\left(N_i - \overline{N}\right)}{\sqrt{\sum_{i=1}^{K}\left(M_i - \overline{M}\right)^2}\sqrt{\sum_{i=1}^{K}\left(N_i - \overline{N}\right)^2}}, \tag{1}$$

where $j, k \in \{R, G, B\}$, $K = W * H$, M and N indicate band map matrix vector of poisoned image, and $\overline{M}$ and $\overline{N}$ are the mean of the elements in the vector. In the experiment, Table 1 reveals the correlation between the intensity values and the corresponding color bands, and they all show strong correlation. The triggers generated in the three backdoor attacks have no effect on the correlation of the intensity values among bands. On the other hand, for BadNets, the added triggers show almost zero correlation between bands. Even for Blend Attack and SSBA, they exhibit weak correlation.

We note that it is difficult to detect whether an image is a poisoned one based on the weak correlation of the trigger among different bands. In the pipeline of our defensing method as shown in Figure 1, the backdoor target label is a frog, and the trigger is the invisible additive noises, which are embedded into the clean picture by pretrained encoder. In the training stage, we adopt the poisoned samples and clean samples to train DNNs and then get the backdoored DNN which classifies poisoned samples to the target label, while performing perfect on clean samples. The pretrained detector detects the training set and adds the sample to the detection set if it is predicted to be poisoned. Then, the detection set was re-entered into the backdoored DNNs for backdoor unlearning, which gets clean DNNs. In the inference stage, the clean DNNs will behave normally on the test samples, and the poisoned samples will not be classified into the target label.

4. Our Protocol Design

We will describe the details of the approach to detecting triggers and backdoor unlearning in this section, as outlined in Algorithm 1. Table 2 describes the symbols used in Algorithm 1.

4.1. Backdoor Detection Design. Let $D_{\mathrm{train}} = \{x_i, y_i\}_{i=1}^{n}$ indicates the training set containing n samples, where $x_i \in X$ and $y_i \in Y = \{1, 2, \ldots, K\}$. The DNN model learns a function f_w: $x_i \longrightarrow y_i$ with parameters w, and y_i denotes the label. D_{poison} indicates the poisoned training set, and D_{clean} represents the clean training set. Specifically, D_{train} consists of D_{poison} and D_{clean}, i.e.,

$$D_{\mathrm{train}} = D_{\mathrm{poison}} \cup D_{\mathrm{clean}}, \tag{2}$$

where $D_{\mathrm{poison}} \subset D_{\mathrm{train}}$, $\gamma = |D_{\mathrm{poison}}|/|D_{\mathrm{train}}|$ indicates the poisoning rate, $D_{\mathrm{clean}} = \left\{(x_i, y_i)((x_i, y_i) \subset D_{\mathrm{train}}/D_{\mathrm{poison}}\right\}$. Specifically, D_{detect} indicates the set consisting of poisoned samples detected by the detector, where $D_{\mathrm{detect}} \subset D_{\mathrm{poison}}$. Since it is difficult to detect all the poisoned samples in the training set, some of the clean samples are also included in D_{detect}. The more clean samples are included in D_{detect}, the lower the classification accuracy of the model on the clean samples will be after it performs backdoor unlearning. Define the detection rate = $|D_{\mathrm{train}}|/|D_{\mathrm{train}}|$, and ρ plays a key role in the final model performance.

4.1.1. Observation. The trigger generation in most backdoor attack methods is similar to the steganography algorithm applied to images, in which additional noise is embedded in the image. For example, for the attack proposed in [22], $G(x) = \alpha \cdot t + (1 - \alpha) \cdot x, \forall x \in X$, where $G(x)$ generates poisoned sample, and t indicates the backdoor triggers. The trigger generation in SSBA is also motivated by the DNN-based image steganography [37].

Based on the observation and the key intuition, we can detect whether the image is poisoned based on steganalysis. Convolutional neural network structure is widely used in gray-scale image steganalysis. For color image, the summation normal convolution reserves strongly correlated patterns but compromises uncorrelated noise or weak correlated noise. In the process of training the detector, it is necessary to preserve the characteristics of the trigger as much as possible. The wider separate-then-reunion network (WISERNet) [18] chooses a channel-wise convolution in the bottom convolution layer, which can well preserve the features of extra added noise in the image. In addition, WISERNet initializes the convolution kernel using the high-pass filter of the null domain rich model [38] to better extract noise (trigger) features.

4.1.2. How to Build the Detector. We use the WISERNet [18] as a core for the backdoor detector. Since the image

Table 1: The correlation between the intensity of different color bands and those of corresponding triggers.

Attack	Types	Red vs. green	Red vs. blue	Blue vs. green
BadNets [21]	Intensity	0.9512	0.8950	0.9737
	Trigger	0.1781	0.1970	0.2710
Blend Attack [22]	Intensity	0.9596	0.9121	0.9744
	Trigger	0.6672	0.5545	0.8046
SSBA [14]	Intensity	0.9542	0.9005	0.9695
	Trigger	0.6424	0.5960	0.6798

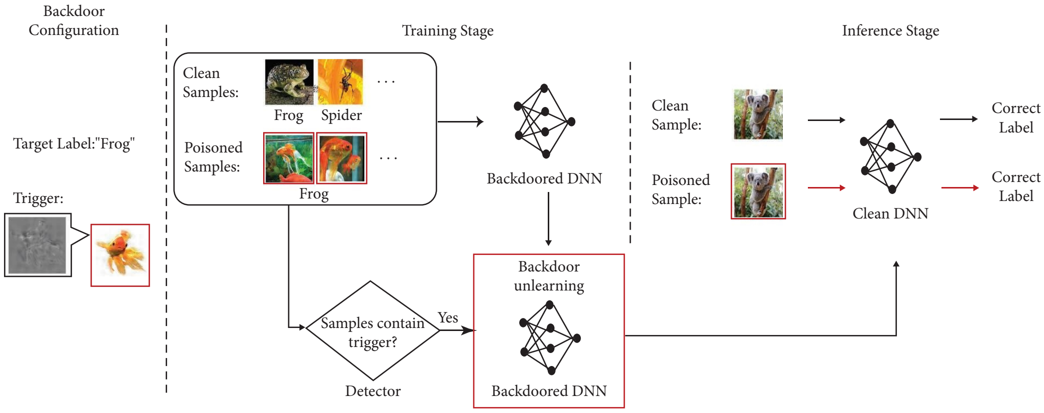


Figure 1: The pipeline of our defensing method.

Table 2: List of symbols.

Symbol	Description
$X_c = \{X_i\}_{i=1}^n$	The clean samples set
A	The backdoored DNN model
B	The clean DNN model
D	The detector
$\varnothing$	The empty set
$G(x)$	The function to generate poisoned sample
θ	The model parameters
∇	Gradient operator
y	Sample label. The sample is clean if $y = 1$ (poisoned if $y = 0$)

convolution operation affects the additional noise [18], the sum in the convolution layer retains the strong correlation pattern but damages the irrelevant noise. Therefore, WISENet uses the normal convolution summation operation in the upper convolution layer rather than using the sum operation in the bottom convolution layer. WISERNet can be divided into three parts in turn: separation, reunion, and prediction. The separated part is composed of channel convolution layer. The main purpose of convolution in the bottom convolution layer is to suppress the relevant image content. WISERNet gives up the sum in the bottom convolution layer and selects the channel volume to reduce the weakening of the network to the irrelevant noise. The reunion part is composed of three wide and relatively shallow normal convolution layers that retain summation. The number of kernels in each convolution layer will gradually increase to augment the capacity of WISERNet. The typical practical method of deep learning network is to design it deeper. However, the deeper the network is, the more output is involved in the summation, and as a result, the more severely the weakly correlated signal is damaged. Therefore, WISERNet designs the upper convolution layer wider to improve its detection performance. The prediction part is composed of four layers of fully connected neural networks to make the final prediction.

As shown in Figure 2, the image is input during the detection process dividing it into red, green, and blue bands, and then, convolution at the channel level is applied separately. The initialization of the convolution kernel weights in each channel is then performed using 30 high-pass filters in the null domain rich model, and as a result, 30 channel feature maps are generated. Finally, the three independent channels are joined together to form a 90 channel output, which is used as the input to the second convolution layer.

```
Input: A clean sample $X_c = \{X_i\}_{i=1}^{n}$, a training set $D_{\text{trian}}$, a backdoored DNN model $A$.
Output: A clean DNN model $B$.
Initialize $D_{\text{detect}} = \varnothing$, $\rho = 0$, and detector $D$.
//step 1: generate poisoned-clean pair samples.
set $X_p = G(x), \forall x \in X_c$, where $G(x)$ generate poisoned sample;
set $\chi = X_c + X_p; y = 1; x \in X_c; y = 0; x \in X_p$;
//step 2: Train detector $D$.
set $\delta \longleftarrow 0$, learning rate $\eta = 0.01$;
for epoch = $1, 2, \ldots, m$ do
    for minibatch $B \subset \chi$ do
        Update $\theta$ of detector $D$ with stochastic gradient descent;
        $g_\theta = \mathbb{E}_{(x,y)\subset B}[\nabla_\theta \mathscr{L}(x + \delta, y, \theta)]$;
        $\theta = \theta - \eta g_\theta$;
//step 3: detect poisoned samples in training set.
set $D_{\text{detect}} = \varnothing, \rho = 0$;
for $i = 0; i++; i \leq |D_{\text{train}}|$ do
    //$D(\cdot)$ indicates the inference result of detector $D$
    while $\rho \leq 0.04$ do
        if $D(x_i) = 0$ then
            $D_{\text{detect}} = D_{\text{detect}}.\text{append}(x_i)$, where $x_i \in D_{\text{train}}$;
$\rho = |D_{\text{detect}}|/|D_{\text{train}}|$;
    break;
//step 4: Backdoor unlearning.
input $D_{\text{detect}}$ into $A$ and update model by using equation (5);
return the clean model $B$.
```

Algorithm 1: Backdoor detection and removal.

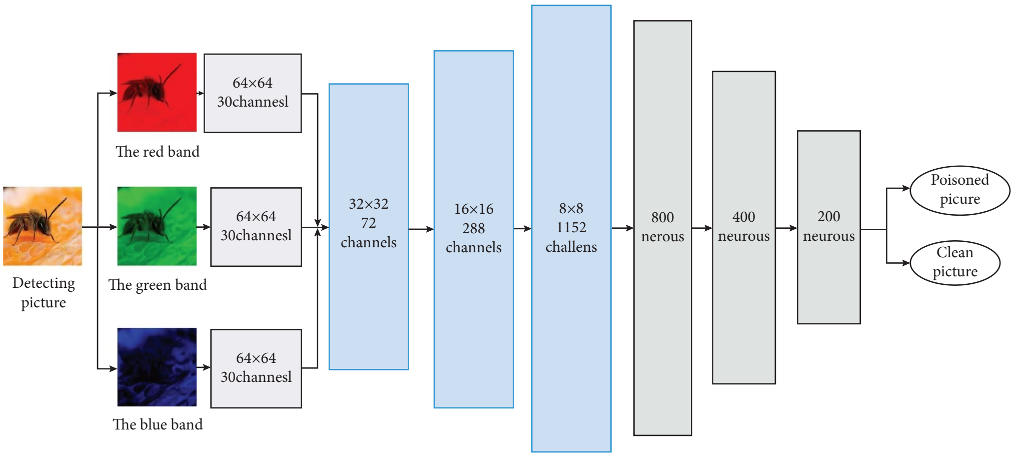


Figure 2: The architecture of wider separate-then-reunion networks [18].

From the second convolutional layer forwards, a standard convolutional approach is used, with the structure of the convolutional operation layer, the batch normalization layer, the activation function layer, and the average pooling layer in order. Since the complexity of the convolutional layers affects the feature extraction and processing, the number of convolutional kernels in each convolutional layer is correspondingly quadrupled to maintain the complexity of the convolutional layers for better noise feature extraction and processing. After the normal convolutional layer, the output feature maps are then combined as variables in 32 steps and input to the fully connected layer. The fully connected layers contain 800, 400, 200, and 2 neurons, respectively, and the three hidden layers use the ReLU activation function. The last fully connected layer performs the final classification prediction result, and if the prediction result is a poisoned sample, the backdoor is buried in the model.

4.2. Backdoor Mitigation Design. Despite the detection of poisoned samples in the training set by the detector, the backdoor in the model still exists. Let $(X, Y) = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ be the training samples, and the training of the model in the backdoor attack can be achieved by minimizing the following empirical error:

$$\min L = \frac{1}{n}\sum_{1}^{n} D_{\text{clean}}\left[\ell\left(f_{\theta}\left(x_i\right), y_i\right)\right] + \frac{1}{m}\sum_{1}^{m} D_{\text{poison}}\left[\ell\left(f_{\theta}\left(x_i\right), y_i\right)\right], \tag{3}$$

where n and m are the number of clean samples and poisoned samples in the training set, respectively. ℓ indicates the loss function such as the cross-entropy loss commonly used in DNN training.

Equation (3) shows that the backdoor injection process can be considered an instance of multitask learning. The main task is the training on the clean samples, whereas the other task is the training on the poisoned samples, that is, the backdoor task. To prevent the model from learning the backdoor task and thus achieving the goal of backdoor unlearning, it can be achieved by minimizing the following empirical error:

$$\min L = \frac{1}{n}\sum_{1}^{n} D_{\text{clean}}\left[\ell\left(f_{\theta}\left(x_i\right), y_i\right)\right] - \frac{1}{m}\sum_{1}^{m} D_{\text{poison}}\left[\ell\left(f_{\theta}\left(x_i\right), y_i\right)\right]. \tag{4}$$

Equation (4) maximizes the backdoor task compared to (3).

Since it is difficult to detect all the D_{poison} in the training set, and the training set of detected poisoned samples is also containing some clean samples, it makes the classification accuracy of the model on clean samples drop significantly. Therefore, we use the detection dataset D_{detect} instead and achieve the effect of backdoor unlearning by minimizing the following empirical error:

$$\min L = \frac{1}{n}\sum_{1}^{n} D_{\text{clean}}\left[\ell\left(f_{\theta}\left(x_i\right), y_i\right)\right] - \frac{1}{m'}\sum_{1}^{m'} D_{\text{detect}}\left[\ell\left(f_{\theta}\left(x_i\right), y_i\right)\right]. \tag{5}$$

5. Experiments

In this section, we implement our protocol based on the datasets of CIFAR10 [39] and VGGFACE2 [40]. We experimentally test the trigger performance and analyze the effects of trigger location, trigger size, and the string representing the target label by the attack SSBA on the performance of the detector. In addition, we experimentally analyze the effect of the size of the detection rate ρ on the performance of the model and arrive at the value of ρ for which the defensing protocol achieves better results when targeting a variety of backdoor attacks. Finally, the effectiveness of this protocol is compared with existing typical backdoor defensing protocols to analyze the effectiveness of our protocol.

5.1. Experiment Setup. The implementation of the detector is based on the Caffe toolbox [41]. The network is trained using small batch stochastic gradient descent with an initial learning rate of 0.001, a learning rate adjustment strategy set to inv, and a fixed momentum of 0.9. The maximum number of training iterations is set to 20,000, and the batch size is 16 during training. All training and testing procedures are performed on a server with the hardware of NVIDIA GeForce RTX 2080 GPU and 10 GB of RAM. The software used for the server is Linux (3.2.x) operating system and Python 3.6.3. To evaluate the defensing approach, we consider two classical image classification tasks: object classification and face recognition. The detailed information about each task and the associated dataset are described in Table 3.

> Object Classification (CIFAR10 [39]): This task is commonly used to evaluate attacks against DNNs and was chosen to train the model PreActResNet [42] using the CIFAR10 dataset. The original dataset contains 10 classes, which contains 50,000 training datasets and 10,000 test datasets.
>
> Face Recognition (VGGFace2 [40]): This task recognizes the faces of 200 people by training the model ResNet [43]. The original dataset contains 3.31 million images. We randomly select 200 categories which contain 400 images for training and another 50 images for testing.

According to the backdoor attacks, we use three already infected object classification models and face recognition models by BadNets [21], Blend Attack [22], and SSBA [14].

The poisoning rate $\gamma = 10\%$ and the target label are set to 0. Figure 3 shows the poisoned samples generated by the three attacks. The backdoor trigger is set to a white square located in the lower right corner of the image, which only accounts for 1% area of the image for BadNets and Blend Attack, and the blending rate (trigger transparency) is set to 0.2 for the Blend Attack. For SSBA, the trigger is generated by the encoder that is a U-Net [44] style DNN trained on the clean samples, which achieves the invisibility and sample-specific of the trigger.

We adopt three effective performance metrics: attack success rate (ASR), which is the classification accuracy on the poisoned test set, clean accuracy (CA), which is the classification accuracy on the clean test set, and detection success rate (DSR), which is the success rate of detecting poisoned samples on the training set. Table 4 shows ASR and CA of the three backdoor attacks on the two classification tasks.

5.2. The Effect of Backdoor Detection. The success rate of detecting poisoned samples by the detector is the key factor to judge the effectiveness of our protocol. For the above three attacks, the data are poisoned accordingly and then detected by the detector. In each experiment, first 10,000 images in the training set are randomly selected to add triggers, and the way of adding triggers is kept the same as in the experimental setup. Then, 6000 pairs of clean-poisoned images are randomly selected and input into the WISERNet for training, while the remaining 4000 pairs are used for testing. Table 5 shows the detection success rates for the three attacks under the two tasks, respectively. 99% of the poisoned images can be detected for both the BadNets and Blend Attack on given datasets. For SSBA, above 94 % of the poisoned images can be detected on the CIFAR10 dataset and 99% on the VGGFACE2 dataset.

Considering the effects of changing the shape and position of the trigger and the different strings representing the target labels in SSBA on the detection success rate, we discuss the effects on the detection success rate by modifying the shape and position of the trigger and the strings and then feed them into the already trained WISERNet.

Figure 4 shows the effect of different trigger shapes and positions in BadNets on the detection success rate and the effect of different representative strings in SSBA on the detection success rate, both experiments on the VGGFACE2 dataset. The model (model1) is the detector trained with the poisoned samples generated by the BadNets, and the triggers are 9×9 white squares in the lower right corner of the image. The other model (model2) is the detector trained with the poisoned samples generated by SSBA method, and the string embedded in the image is 0. In Figure 4(a), the trigger shapes are set to white blocks with circles, ovals, and triangles and then input into model1 to get the detection results. In Figure 4(b), the position of the trigger is set at the four corners of the image, respectively, and then input into *model*1 for detection. In Figure 4(c), the strings embedded into the images are set to 0, 1, 2, and 3, respectively, and then input into model2 to get its classification results. Figure 4 shows that the content of the representative string in SSBA does not affect the efficiency of the detector, and it can achieve more than 96 % detection success rate for poisoned images. When the size of the trigger does not cover the entire picture, it changes its position and shape that can affect the efficiency of the detector.

The position and shape of the triggers affect the detection success rate, but the content of the representative string in SSBA does not affect the detection success rate. Since the way of adding the trigger in SSBA makes the trigger and the features of clean samples fused, its feature position also overlaps with the position of the main features of those clean samples. Thus, the trigger position and shape are not critical factors in the training process of WISERNet. Furthermore, the trigger features in BadNets differ from the main features, and the position and shape have some influence on the results.

5.3. The Effect of Backdoor Mitigation. The performance of the model after backdoor unlearning can be optimal in equation (4) if all poisoned samples in the training set are detected and no clean samples are mistakenly detected as poisoned samples. However, it is hard to arrive that the detection method does not detect 100 % of the poisoned samples. In addition, it may be affected by the dataset, such as the trigger set in BadNets attack is the white square in the bottom right corner of the image, yet some of the images in the CIFAR10 dataset are also white in the bottom right corner, which will lead to the wrong detection. Therefore, there will be a small number of clean samples included in D_{detect}. Usually, the larger the value of $|D_{\text{detect}}|$, the lower the success rate of the attack after the backdoor unlearning. However, if a number of the clean samples are included in D_{detect}, w will make the classification accuracy on the clean samples drop significantly. Therefore, we experimentally investigate the correlation between the value of ρ and the performance of our protocol.

In the CIFAR10 dataset, the poisoning rate is set to 10%, and thus, there are 5,000 poisoned images in the training set. Set ρ values at 0.02, 0.04, 0.06, 0.08, and 0.1. The optimal range of ρ values is experimentally derived, which maintains the classification accuracy on benign samples while reducing ASR. Figure 5 shows the implementation on the CIFAR10 dataset with different ρ values for different backdoor attacks. It can be found that our protocol is effective against all three attacks at different ρ. The backdoor attack rate can drop to very close to 0% while the classification accuracy of the model on clean samples maintains at a high level. We also find that the best performance of our protocol is achieved when $\rho \leq 0.04$.

5.4. Comparison with the Existing Defensing Protocols. To further evaluate the effectiveness of our protocol, we consider three state-of-the-art backdoor attacks and compare with two typical backdoor defensing techniques. Table 6 demonstrates our proposed method on the CIFAR-10 dataset and the VGFACE2 subset dataset. FP [15] and ABL [19] are following the configurations specified in their

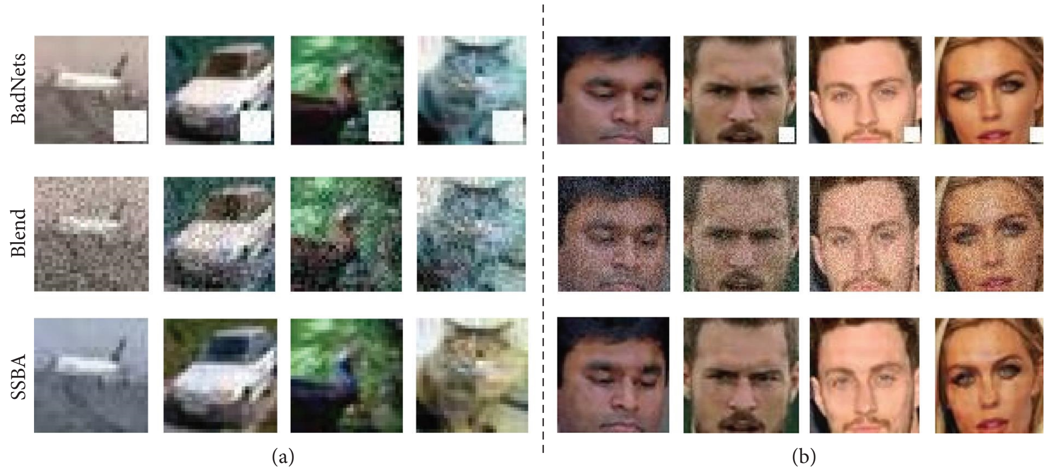


Figure 3: Poisoning samples generated by different backdoor attacks: BadNets, Blend Attack, and SSBA. (a) CIFAR 10. (b) VGG face2.

Table 3: Details of datasets and model architectures.

Task	Dataset	# of labels	Input size	# of training images	Model architecture
Object classification	CIFAR10	10	32 × 32 × 3	50000	PreActResNet
Face recognition	VGGFace2	200	64 × 64 × 3	80000	ResNet

Table 4: Attack success rate (ASR) and clean accuracy (CA) of various backdoor attacks on classification tasks.

Task	Backdoored model (ASR %/CA %)			Clean model (CA %)
	BadNets	Blend Attack	SSBA	
CIFAR10	99.64/93.02	100/93.67	99.91/93.08	92.40
VGGFace2	99.40/87.80	99.98/87.84	99.67/88.59	91.31

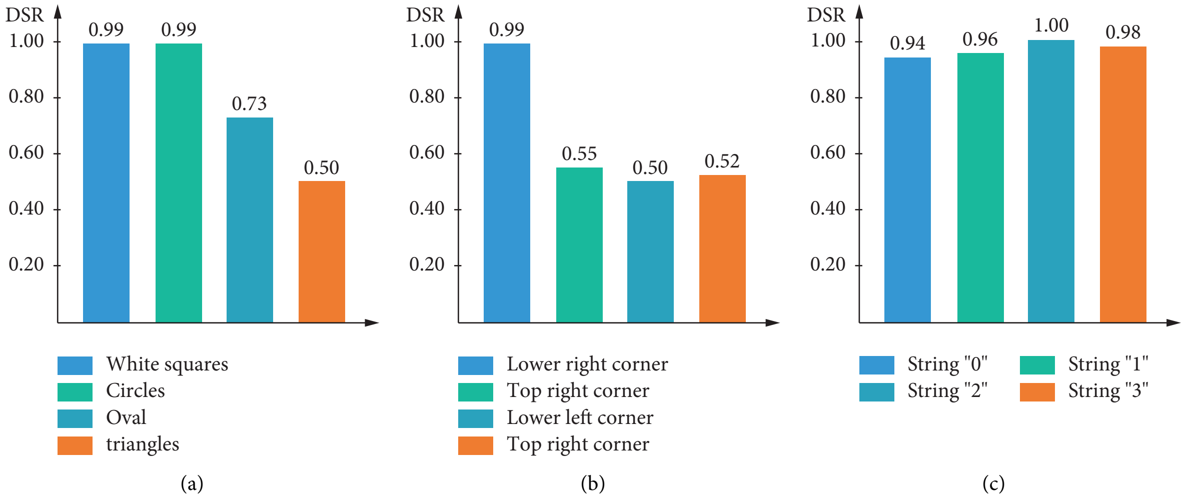
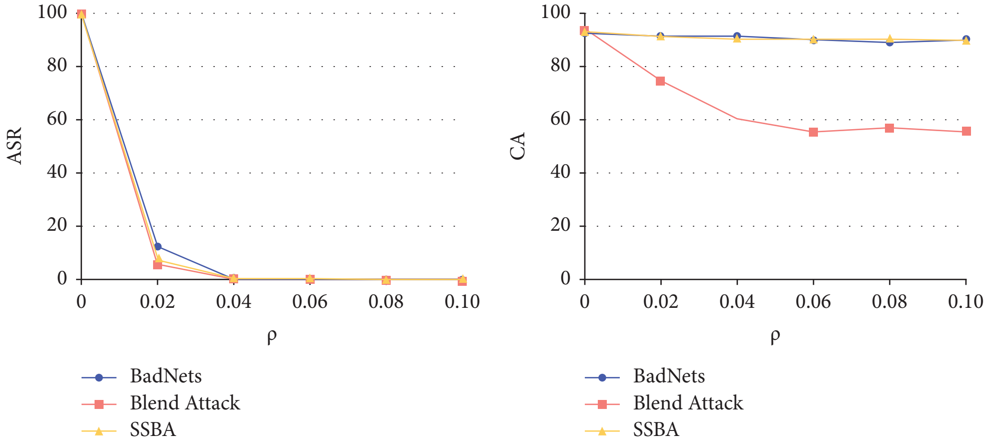


Figure 4: Detection success rate related to various trigger shapes, positions, and representative strings: (a) Effect of trigger shape on detection success rate, (b) effect of trigger location on detection success rate, and (c) effect of different string on detection success rate.

original papers. In addition, the last convolutional layer of the neural network in FP is pruned, and ASR of the model significantly decreases when 60 % of the neurons are pruned.

Let epoch $T = 107$ and turn-period $T_{te} = 25$ be set in the training of the CIFAR10 dataset, and epoch $T = 46$ and turn-period $T_{te} = 25$ in the training of the VGGFACE2 subset



(a) (b)

Figure 5: The effect performance on different detection rate ρ.

Table 5: Detection success rate against three typical attacks.

Datasets	Backdoored model (DSR %)		
	BadNets	Blend Attack	SSBA
CIFAR10	99.85	100.00	94.92
VGGFace2	99.78	100.00	99.68

Table 6: Effectiveness performance comparison of defensing protocols under different backdoor attacks.

Dataset	Attack type	FP [15]		ABL [19]		Ours	
		ASR %	CA %	ASR %	CA %	ASR %	CA %
CIFAR10	None Attack	0.00	91.88	0.00	92.75	0.00	**93.79**
	BadNets	99.81	90.37	0.42	**93.14**	**0.21**	90.54
	Blend Attack	100.00	**93.43**	0.48	76.56	**0.15**	60.43
	SSBA	99.90	93.09	0.50	**93.17**	**0.43**	90.81
VGGFACE2 subset	None Attack	0.00	72.62	0.00	82.96	0.00	**86.73**
	BadNets	11.79	77.26	0.00	14.90	**0.32**	**83.36**
	Blend Attack	14.89	71.46	0.00	9.72	**0.46**	**78.67**
	SSBA	11.47	72.23	0.00	7.97	**0.17**	**84.27**

For different attacks, bold values represents the best defense effect among the three defense schemes.

dataset in the defensing protocol ABL. In both datasets, our protocol is set to $\rho = 0.04$. None Attack in Table 6 means that the training data are completely clean.

In the CIFAR10 dataset, ABL can achieve better results in the classification accuracy of clean samples compared to our protocol, but our protocol can achieve the best decrease in the reduction of the attack success rate. In the subset of VGGFACE2 dataset, FP can reduce the attack success rate of the three attack methods to less than 15%, but at the same time, the classification accuracy of the clean samples also decreases to less than 75%. ABL reduces the attack success rate of the three attack methods to 0, but the performance of the clean samples of the model is poor; thus, we can assume that ABL has no defensive effect. Our protocol has better performance in both attack success rate and classification accuracy on the clean samples. In Table 6, it can be seen that Blend Attack, both ABL and our protocol, decreases in attack success rate and classification accuracy compared to other attack methods, which is because the dataset images are blurred, and the trigger pattern mixed with poisoned images produces the effect of natural artifacts, which makes it difficult to detect poisoned images. Maintaining the classification accuracy of the model on clean samples is as important as reducing the success rate of the attack. Table 6 shows that our protocol is better to maintain the classification accuracy of the model on clean samples while reducing the success rate of the attack compared with FP and ABL.

6. Conclusion

In this work, we propose a backdoor detecting and removing protocol for deep neural networks based on image steganalysis. Our protocol detects the poisoned training samples using a deep steganalyzer constructed by WISERNet and retrains the model for backdoor unlearning by the detected poisoned samples. Compared with the SOTA backdoor defensing protocols, our protocol achieves to reduce the backdoor attack success rate while maintaining a high classification accuracy on the clean samples. In the future work, we will further study the backdoor detection and unlearning methods to obtain higher clean sample classification accuracy and lower backdoor attack success rate for different attack methods and design universal and efficient backdoor defensing protocols.

Data Availability

The data supporting the current study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61972241 and 62102300, Natural Science Foundation of Shanghai under Grant 22ZR1427100 and 18ZR1417300, Soft Science Foundation of Shanghai under Grant 23692106700, Fishery Engineering and Equipment Innovation Team of Shanghai High-Level Local University, and Luo-Zhaorao College Student Science and Technology Innovation Foundation of Shanghai Ocean University.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei, "Imagenet Large Scale Visual Recognition Competition," 2012, https://arxiv.org/abs/1409.0575.

[3] A. Graves, M. Abdel-rahman, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649, Ieee, Vancouver, Canada, May 2013.

[4] X. Li, J. He, P. Vijayakumar, X. Zhang, and V. Chang, "A verifiable privacy-preserving machine learning prediction scheme for edge-enhanced hcpss," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5494–5503, 2022.

[5] C. Zhou, D. Chen, S. Wang, A. Fu, and Y. Gao, "Research and challenge of distributed deep learning privacy and security attack," *Journal of Computer Research and Development*, vol. 58, no. 5, pp. 927–943, 2021.

[6] Y. Liu, A. Mondal, A. Chakraborty et al., "A survey on neural trojans," in *Proceedings of the 2020 21st International Symposium on Quality Electronic Design (ISQED)*, IEEE, Santa Clara, CA, USA, March 2020.

[7] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor Learning: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 2022, 2022.

[8] Y. Gao, G. Bao, Z. Zhang et al., "Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review," 2020, https://arxiv.org/abs/2007.10760.

[9] M. Goldblum, D. Tsipras, C. Xie et al., "Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, 2022.

[10] Z. Tian, L. Cui, J. Liang, and S. Yu, "A Comprehensive Survey on Poisoning Attacks and Countermeasures in Machine Learning," *ACM Computing Surveys (CSUR)*, vol. 55, 2022.

[11] K. Doan, Y. Lao, and P. Li, "Backdoor attack with imperceptible input and latent modification," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18944–18957, 2021.

[12] K. Doan, Y. Lao, W. Zhao, and P. Li, "Lira: learnable, imperceptible and robust backdoor attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11966–11976, Montreal, Canada, October 2021.

[13] Q. Zhang, Y. Ding, Y. Tian, J. Guo, M. Yuan, and Y. Jiang, "Advdoor: adversarial backdoor attack of deep learning system," in *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, Virtual, Denmark, July 2021.

[14] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16463–16472, Montreal, Canada, October 2021.

[15] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: defending against backdooring attacks on deep neural networks," in *International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 273–294, Springer, Berlin, Germany, 2018.

[16] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: a defence against trojan attacks on deep neural networks," in *Proceedings of the 35th Annual Computer Security Applications Conference*, New York, NY, USA, December 2019.

[17] B. Wang, Y. Yao, S. Shan et al., "Neural cleanse: identifying and mitigating backdoor attacks in neural networks," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, pp. 707–723, IEEE, Francisco, CA, USA, May 2019.

[18] J. Zeng, S. Tan, G. Liu, B. Li, and J. Huang, "Wisernet: wider separate-then-reunion network for steganalysis of color images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2735–2748, 2019.

[19] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: training clean models on poisoned data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14900–14912, 2021.

[20] Q. Tan, Y. Zeng, Y. Han, Y. Liu, and Z. Liu, "Survey on backdoor attacks targeted on the neural network," *Chinese Journal of Network and Information Security*, vol. 7, no. 3, pp. 46–58, 2021.

[21] T. Gu, B. Dolan-Gavitt, and S. G. Badnets, "Identifying Vulnerabilities in the Machine Learning Model Supply Chain," 2017, https://arxiv.org/abs/1708.06733.

[22] X. Chen, L. Chang, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," 2017, https://arxiv.org/abs/1712.05526.

[23] Y. Liu, S. Ma, Y. Aafer et al., "Trojaning attack on neural networks," *25th Annual Network and Distributed System Security Symposium (NDSS)*, Purdue University, West Lafayette, IN, USA, 2018.

[24] D. Chen, A. Fu, C. Zhou, and Z. Cheng, "Federated learning backdoor attack protocol based on generative adversarial

network," *Journal of Computer Research and Development*, vol. 58, no. 11, pp. 2364–2373, 2021.

[25] S. Garg, A. Kumar, V. Goel, and Y. Liang, "Can adversarial weight perturbations inject neural backdoors," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 2029–2032, New York, NY, USA, December 2020.

[26] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," *30th USENIX Security Symposium USENIX Security*, vol. 21, pp. 1505–1521, 2021.

[27] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the backdoor attacks' triggers: a frequency perspective," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16473–16481, New York, NY, USA, December 2021.

[28] S. Li, M. Xue, B. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 1–2105, 2020.

[29] P. Zhao, P. Chen, P. Das, Karthikeyan Natesan Ramamurthy, and X. Lin, "Bridging mode connectivity in loss landscapes and adversarial robustness," 2020, https://arxiv.org/abs/2005.00060.

[30] T. Garipov, Pavel Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of dnns," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[31] Y. Liu, Wen-Chuan Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "Abs: scanning neural networks for back-doors by artificial brain stimulation," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1265–1282, New York, NY, USA, November 2019.

[32] Wikipediapedia, "Electrical Brain Stimulation," 2022, https://en.wikipedia.org/wiki/Electricalbrainstimulation.

[33] G. Bao, E. Abbasnejad, and D. C. Ranasinghe, "Februus: input purification defense against trojan attacks on deep neural network systems," in *Proceedings of the Annual Computer Security Applications Conference*, pp. 897–912, Austin, CA, USA, December 2020.

[34] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. 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, Cambridge, MA, USA, June 2017.

[35] S. Udeshi, S. Peng, G. Woo, L. Loh, L. Rawshan, and S. Chattopadhyay, "Model Agnostic Defence against Backdoor Attacks in Machine Learning," *IEEE Transactions on Reliability*, vol. 71, 2022.

[36] Q. Han, Y. Zeng, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham, "Deepsweep: an evaluation framework for mitigating dnn backdoor attacks using data augmentation," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pp. 363–377, New York, NY, USA, September 2021.

[37] M. Tancik, Ben Mildenhall, and N. Ren, "Stegastamp: invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2117–2126, New Orleans, LA, USA, January 2020.

[38] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[39] A. Krizhevsky and G. Hinton, *Learning Multiple Layers of Features from Tiny Images (2009)*, University of Toronto, Toronto, Canada, 2009.

[40] Q. Cao, L. Shen, W. Xie, M. Omkar, and A. Zisserman, "Vggface2: a dataset for recognising faces across pose and age," in *Proceedings of the 2018 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2018)*, pp. 67–74, IEEE, Vancouver, Canada, January 2018.

[41] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, and R. Girshick, "Sergio guadarrama, and trevor darrell. Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 675–678, New Yor, NY, USA, September 2014.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*, pp. 630–645, Springer, Berlin, Germany, 2016.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, New Orleans, LA, USA, May 2016.

[44] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Berlin, Germany, 2015.