Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.025923

ARTICLE

Adaptive Backdoor Attack against Deep Neural Networks

Honglu He, Zhiying Zhu and Xinpeng Zhang*

School of Computer Science, Fudan University, Shanghai, 200433, China

*Corresponding Author: Xinpeng Zhang. Email: zhangxinpeng@fudan.edu.cn

Received: 05 August 2022 Accepted: 07 November 2022

ABSTRACT

In recent years, the number of parameters of deep neural networks (DNNs) has been increasing rapidly. The training of DNNs is typically computation-intensive. As a result, many users leverage cloud computing and outsource their training procedures. Outsourcing computation results in a potential risk called backdoor attack, in which a well-trained DNN would perform abnormally on inputs with a certain trigger. Backdoor attacks can also be classified as attacks that exploit fake images. However, most backdoor attacks design a uniform trigger for all images, which can be easily detected and removed. In this paper, we propose a novel adaptive backdoor attack. We overcome this defect and design a generator to assign a unique trigger for each image depending on its texture. To achieve this goal, we use a texture complexity metric to create a special mask for each image, which forces the trigger to be embedded into the rich texture regions. The trigger is distributed in texture regions, which makes it invisible to humans. Besides the stealthiness of triggers, we limit the range of modification of backdoor models to evade detection. Experiments show that our method is efficient in multiple datasets, and traditional detectors cannot reveal the existence of a backdoor.

KEYWORDS

Backdoor attack; AI security; DNN

1 Introduction

In the past few years, deep neural networks (DNNs) have achieved great performance in a variety of fields, e.g., image classification [1], speech recognition [2], object detection [3], etc. These achievements are inseparable from a huge amount of trainable parameters that require a lot of computing resources. For example, the initial version of GPT [4] developed in natural language processing has only 117 million parameters. Nowadays, the number of its parameters has increased to 1.5 billion and 1750 billion in GPT-2 [5] and GPT-3 [6], respectively. Therefore, many researchers outsource their computation-intensive training procedures to the third parties referred to as "machine learning as a service" (MLaaS). In this scenario, outsourcing computation incurs a security risk called backdoor attack, which affects the deployment of DNNs in risk-sensitive fields like autonomous vehicles. However, the research of backdoor attack is beneficial to boost the robustness of DNNs and further understand the internal mechanism of DNNs. In the backdoor attack, users upload their datasets and the model structure. MLaaS returns them a well-trained model, and such models perform

well on the clean validation set so that users cannot perceive anomalies. However, attackers can leverage clean images superposed with a predefined trigger to fool the DNNs. In this attack case, the malicious third parties can implant the backdoor in various ways [7] except for changing the network structure. Fig. 1 illustrates backdoor attacks.

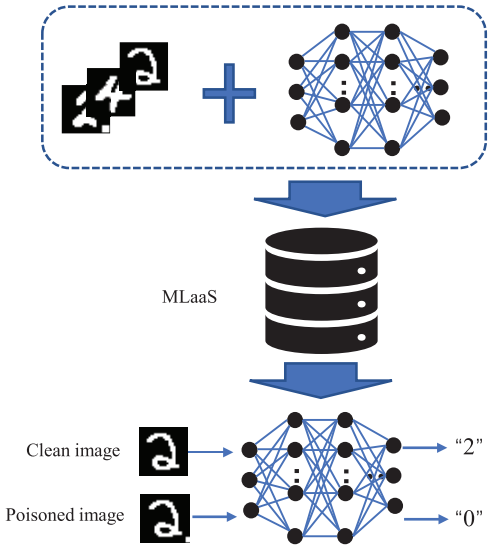


Figure 1: The framework of the backdoor attack

BadNets [8] is the first study over the backdoor attack in the image classification task. They implant the backdoor by polluting a part of training datasets, in which some images are injected with a fixed small trigger at a fixed position. Then, they change the label of polluted images to the target label. The model will be trained on both clean and polluted images. Such well-trained models work well on clean images but misclassify the polluted images (clean images with the predefined trigger) as the target label. Trojaning attack [9] improved the BadNets and extends the application of the backdoor attack into the field of face recognition and natural language processing.

Nguyen et al. [10] proposed the input-aware dynamic backdoor attack, in which a trigger for each image is unique. They train a generator to create the trigger for each clean image. The trigger generated for a certain image is invalid to others. They introduce a novel cross-trigger and define diversity loss to ensure the non-reusability of the trigger. Due to the diversity of triggers, their method can resist various detection and hardly be reversely constructed. Apart from these approaches, many backdoor attacks have been proposed [11–13].

Despite the great success of the above methods, their trigger is so obvious that they can be perceived by humans. For example, the trigger in the BadNets is a small constant piece of the white or black square. There is a random colored strip as a trigger in the dynamic backdoor attack. Someone can easily reject the input of such abnormal images. Modification hidden in the rich texture areas is more difficult to be perceived than plain areas. Our backdoor attack generates triggers for clean images according to their texture distribution. We avoid modifying pixels located in plain regions and encourage the triggers to be embedded into the rich texture regions like edge areas of images. The adaptive backdoor attack can ensure the visual quality of images with the trigger.

It is reasonable that each image owns its adaptive trigger. However, even if a generator presented by a DNN is used to generate the trigger for each image, without an elaborate design, these triggers will

become repetitive or uniform, i.e., the generator collapses to insignificance, and the attack becomes the constant trigger. In our method, we employ the texture detection module to mark the rich texture regions. The trigger is only allowed to appear in these regions. The advantage is that not only each image obtains its adaptive trigger, but it is also stealthier and harder to be perceived than a random trigger. After selecting the appropriate modification location, we consider the range of modifications. We deploy the L_0-norm as the criterion of measuring the intensity of triggers, which limits the maximum modification values. Since the valid pixel value must be an integer, a novel updating strategy is introduced to handle the round error. In the end, we restrict the distance between the clean classifiers and classifiers with the backdoor for evading the detector.

The main contributions of our method are as follows:

- We propose a *content-based adaptive* backdoor attack. A well-trained generator is used to create an invisible trigger for each clean image, which depends on the texture distribution of the image.
- We propose an approach to evade the detection of the backdoor attack named *parameter clip*, which limits the distance between the backdoored and benign model. *Parameter clip* can also generalize to other backdoor attacks to enhance their stealthiness.
- Extensive experiments demonstrate that the proposed method can achieve a high backdoor attack success rate without affecting the accuracy of clean images. Meanwhile, both backdoored DNNs and poisoned images keep the stealthiness to evade detection.

2 Related Work

2.1 Backdoor Attacks

The backdoor attack is a technique of hiding covert functionality in the DNNs. This functionality is often unaware to the user of DNNs and activated only when the predefined trigger appears. For instance, a backdoored traffic sign recognition performs well on the normal inputs but may predict the "speed-limit" sign as "stop" when a predefined trigger appears on the "speed-limit" sign.

BadNets [8] is one of the most common backdoor attack methods, which injects a backdoor by poisoning the training dataset. The attacker needs to choose a part of the dataset as poisoned images, which are superposed with a fixed predefined trigger (e.g., a white square) at the fixed location. Then the label of these poisoned images will be changed to the target label. A neural network uploaded by the user will be trained on the poisoned dataset. Finally, the well-trained model can work well on clean images but make mistakes when the trigger appears. Another classical attack is the Trojan attack. It designs the trigger, which maximizes the response of certain internal neuron activations, i.e., a strong connection between the trigger and the certain internal neuron. Then, they retrain the DNNs to ensure it predicts the target label when the trigger appears.

Input-aware dynamic backdoor attack [10] proposed by Nguyen et al. is the closest concurrent work to ours. They first argue that the trigger for each image should be different. The classifier can correctly predict the image with the trigger for another image. They divide the dataset into three parts, i.e., clean set, poisoned set, and cross-trigger set. The first two items are similar to the traditional attacks. Cross-trigger set is used to force the backdoored classifier to make the correct predictions on images with mismatched triggers. Additional diversity loss, which makes sure the diversity of triggers, is employed to help the entire system converge. Although their method achieves a high backdoor attack success rate and resists multiple detections, the triggers are obvious and easily perceived by humans.

2.2 Backdoor Denfenses

Since the backdoor attack is an important problem in the AI security field. Many methods [14–17] have been proposed to detect the backdoors. Among these methods, model-based detection is one of the most significant detection methods. Given a well-trained model, the detector aims to reveal if there are any backdoors hidden in the model.

Neural Cleanse [18] is the first approach to detect the backdoor in a well-trained model. The normal backdoor attack will change the decision boundary of the classifier and cause a shortcut between the target label and others. Neural Cleanse takes advantage of this characteristic of backdoor attacks. For each label, it measures the minimal trigger candidate that changes other clean images to the label. If there is a backdoor hidden in the model, an abnormally small index will appear for the target label. Furthermore, for the constant trigger, Neural Cleanse can reversely construct it.

Pruning [19] attempts to remove the neurons which are dormant on the clean images. Most backdoor attacks will activate dormant neurons when the predefined trigger appears. Therefore, cutting dormant neurons is an efficient method to mitigate backdoor attacks. However, the pruning defense cannot verify if a model is implanted with backdoors. Cutting dormant neurons will also degrade the performance of models on clean images.

3 Proposed Method

3.1 Threat Model

We consider the backdoor attack in the outsourced training scenario, in which a user uploads his dataset and model structure to the third-party. For simplicity, we conduct our idea on the image classification task. The third party returns a well-trained classifier F_θ to the user, who will evaluate the classifier on the validation set. The user only accepts the classifier if its accuracy reaches his target accuracy decided by his prior knowledge.

Attack's knowledge. In our attack scenario, we suppose that attackers can obtain a deep neural network structure and the entire training set uploaded by the user. Attackers control the procedure of training but do not see the validation set. Attackers are also not allowed to change the classifier structure, which will easily be perceived by the user.

Attack's goals. The attacker aims to implant a backdoor into the classifier, which is only activated when the predefined trigger appears. The classifier can achieve prediction with high accuracy as close to the benign classifier as possible for clean images. Apart from the attack ability, stealthiness is also not trivial for the attacker. Stealthiness should be considered from two aspects, i.e., the stealthiness of poisoned images (clean images with the trigger) and the stealthiness of backdoored classifier. If the trigger is too obvious, it will be removed by humans manually. Therefore, the trace of the trigger in the poisoned image should be as little as possible. In the stealthiness of the backdoored classifier, many detectors can scan the structure and parameters of the classifier directly. The goals of the attack can be summarized as follows:

$$G = \begin{cases} \max p(F_b(x_{poisoned}) = Target) \\ \min \left| A_{F_b} - A_{F_c} \right| \\ \min \text{distance}(F_b, F_c) \\ \min \text{distance}(x_{clean}, x_{poisoned}) \end{cases}$$

where A_{F_b} and A_{F_c} are the accuracy of the backdoored classifier and benign one, which is evaluated on the validation set by the user. F_b and F_c represent the backdoored classifier and benign one. $x_{poisoned}$ and *Target* are the clean images with the trigger and the target label, respectively.

3.2 Overall of Our Method

To achieve the aforementioned goal, We should simultaneously ensure the stealthiness of both poisoned images and backdoored classifier. Fig. 2 depicts the framework of our method. Each image obtains its adaptive trigger by the combination of texture detection and generator. We argue that modifying pixels in plain areas of an image is easier to be perceived than rich texture areas. Therefore, we employ the texture detection module, which is responsible for generating the mask of the trigger. It ensures that the trigger is only allowed to appear in the rich texture areas. The generator is used to create the trigger for the entire image. To improve the visual performance, we restrict the distortion of poisoned images caused by the trigger. In other words, we use L_0-norm as the criterion to measure the distance between clean images and poisoned ones. In order to avoid generating shortcuts between the decision boundary of the backdoored classifier, we propose a *parameter clip* mechanism, which guarantees the distance between the backdoored classifier and the benign one.

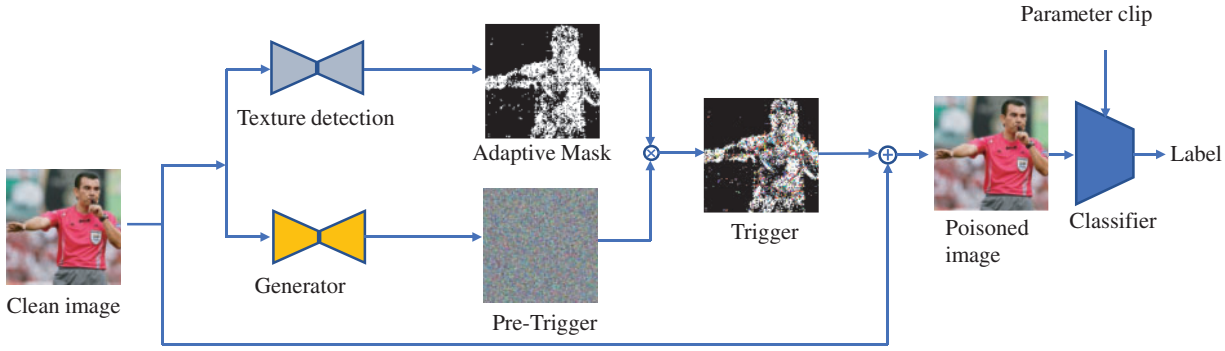


Figure 2: The framework of our method

3.3 Overall of Our Method

To achieve the aforementioned goal, We should simultaneously ensure the stealthiness of both poisoned images and backdoored classifier. Fig. 2 depicts the framework of our method. Each image obtains its adaptive trigger by the combination of texture detection and generator. We argue that modifying pixels in plain areas of an image is easier to be perceived than rich texture areas. Therefore, we employ the texture detection module, which is responsible for generating the mask of the trigger. It ensures that the trigger is only allowed to appear in the rich texture areas. The generator is used to create the trigger for the entire image. To improve the visual performance, we restrict the distortion of poisoned images caused by the trigger. In other words, we use L_0-norm as the criterion to measure the distance between clean images and poisoned ones. In order to avoid generating shortcuts between the decision boundary of the backdoored classifier, we propose a *parameter clip* mechanism, which guarantees the distance between the backdoored classifier and the benign one.

3.4 Stealthiness of Poisoned Images

For the stealthiness of poisoned images, we employ the combination of texture detection and generator. The texture detection module aims to select the appropriate location where the trigger embeds. We use the smoothness metric HILL [20] to measure the texture richness, which gives each

pixel a value, and pixels located in the plain regions will obtain a large value. For a clean image X_c, the formulaic expression of HILL is defined in (1).

$$W = \frac{1}{|X_c \otimes F_h| \otimes F_1} \otimes F_2, \tag{1}$$

where F_1 and F_2 are two average filters sized 3×3 and 15×15, respectively. F_h is a high-pass filter used to calculate the residual of clean images. It can be seen as a convolution operation whose kernel parameters are shown in (2).

$$F_h = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}. \tag{2}$$

HILL will assign pixels located in the plain areas large value. Two average filters as low-pass filters are used for numerical smoothing. Then, we set a threshold T to binarize the result of HILL as (3).

$$Mask = \begin{cases} 1 & w_{i,j} <= T \\ 0 & w_{i,j} > T \end{cases}. \tag{3}$$

Edge detection, widely used in common image processing, is another simple texture detection. Most pixels at the edge part of images change drastically. Trigger embedded into such areas is hard to be perceived. In view of this, we can use edge detection to obtain the adaptive mask of the trigger. Fig. 3 shows some masks generated by the HILL and edge detection (Sobel operator) with multiple threshold T. Note that the mask area becomes large as the threshold T increases, but it is the opposite in the edge detection. Two averaging filters contribute to that the mask generated by HILL is smoother than edge detection.

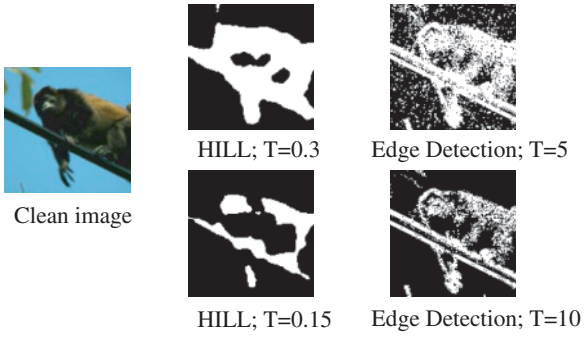
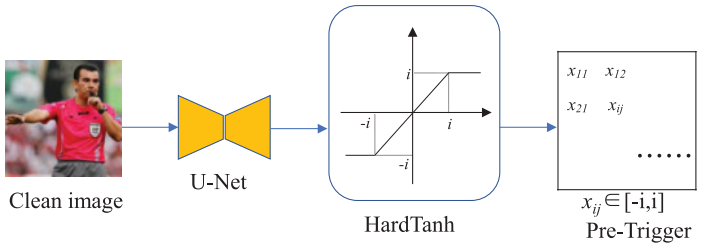
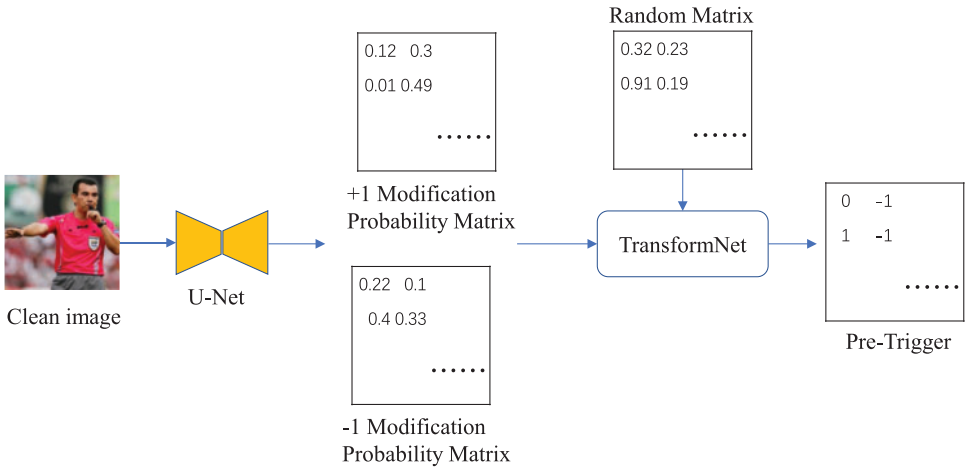


Figure 3: An illustration for the texture detection

The generator is trained to create a pre-trigger for clean images. We use U-Net [21] as the backbone of the generator, which is commonly used as the baseline in medical image segmentation tasks. The size of the output is identical to the input in the U-Net. Excessive modification may cause visual abnormalities. We use the L_0-norm as the criterion, which depends on the maximum modification between two images. In the practical application, we design two different approaches to achieve the restriction of the L_0-norm. Fig. 4 shows the specific structure of the generator.



(a) HardTanh-based generator



(b) Modification Probability Matrix (MPM)-based generator

Figure 4: The details of the generator

For simplicity, we can use a truncation function like the HardTanh as an activation function for the output of the last layer. (4) expresses the definition of the HardTanh function.

$$hardtanh(x) = \begin{cases} i & if \quad x > i \\ -i & if \quad x <= -i \\ x & otherwise \end{cases}, \tag{4}$$

where i is a hyperparameter to control the maximum modification range. We can minimize the value of the trigger as small as possible with the help of the HardTanh function. For a valid image, pixels must be an integer, and the round error can be neglected when the threshold i is relatively large. However, if modification caused by the trigger is small, e.g., no more than 3, the round error cannot be neglected. We propose a novel approach to meet the demand of the restriction of L_0-norm.

To avoid the round error on the small modification, we hope that the generator outputs integers instead of float-point numbers. Inspired by the image steganography technique [22], which modifies the pixels slightly, we design the generator based on the *Modification Probability Matrices* (MPM). We suppose that the maximum modification is no more than 1 as an example. The U-Net output becomes a pair of MPM, whose elements mean the $+1$ or -1 probability of corresponding pixels. The activation function of the last layer of U-Net is expressed as $x = \text{Sigmod}(x)/2$ to limit the range of MPM between 0 and 1/2. Then, we use (5) to transform the MPM to the integers as the trigger.

$$m_{i,j} = \begin{cases} -1 & if \quad n_{i,j} < p_{i,j}^{-1} \\ 1 & if \quad n_{i,j} > 1 - p_{i,j}^{+1} \\ 0 & otherwise \end{cases}, \tag{5}$$

where $p_{i,j}^{-1}$ and $p_{i,j}^{+1}$ are the elements in the MPM, and $n_{i,j}$ is a random number in the interval of [0,1]. However, (5) is a step function and non-differentiable, which cannot be added to the back-propagation pipeline. We train a neural network named TransformNet to simulate this function in advance, and its structure is shown in Table 1. With the combination of the MPM and TransformNet, we can generate the trigger whose values are in {0,−1,1}. In some cases, we can employ more MPM pairs like ±1 and ±2 to obtain a wider range of the trigger. We name these two methods as HardTanh-based generator and MPM-based generator, respectively.

Table 1: The structure of simulation for Eq. (5)

Layer type	Input channel	Output channel
Full connection + Relu	3	16
Full connection + Relu	16	32
Full connection + Tanh	32	1

3.5 Stealthiness of Backdoored Classifiers

In this subsection, we describe our solution named *parameter clip* to ensure the stealthiness of the backdoored classifier. The normal classifier trained on clean images is used as a reference. The detail of the algorithm is shown in Algorithm 1, where r is hyperparameter to control the distance between the backdoored classifier and the benign one. For example, if we set r as 0.1, it presents that the value of the parameter of the backdoored classifier is no more than 1.1 times or no less than 0.9 times of the benign one. The smaller the value of r is, the closer the distance between the backdoored classifier and the benign one is. N is the total number of the parameters. The subscript i represents the i-th parameter in the classifier. We clip the parameters of the backdoored model after updating the parameter on each mini-batch data at every step.

Algorithm 1. Parameter clip

Input: Parameters of the benign classifer F_c, Parameters of the backdoored classifer F_b, Hyperpameter r, The total number of parameters N.

while $i <= N$ **do**
  $f_{c_i} \leftarrow F_c$
  $f_{b_i} \leftarrow F_b$
  $maximum = r * abs(f_{c_i})$
  $res = f_{b_i} - f_{c_i}$
  if $abs(res) > maximum$ **then**
    if $res > 0$ **then**
      $f_{b_i} = f_{c_i} + maximum$
    else if $res < 0$ **then**
      $f_{b_i} = f_{c_i} - maximum$
  $i = i + 1$

Fig. 5 illustrates the principle of the parameter clip. We take a simple classification as an example. There are four categories, and the blue triangular part is supposed as the target label by the backdoor attack. As mentioned in the Neural Cleanse [18], the backdoor attack changes the decision boundary and creates shortcuts between the target label and others, i.e., a slight disturbance can make the samples misclassified as the target label. In the upper right corner of Fig. 5, the black dotted line and the solid line indicate the decision boundary of the clean classifier and backdoored one. Parameter clip eliminates these shortcuts by limiting the modification range of the backdoored classifier.

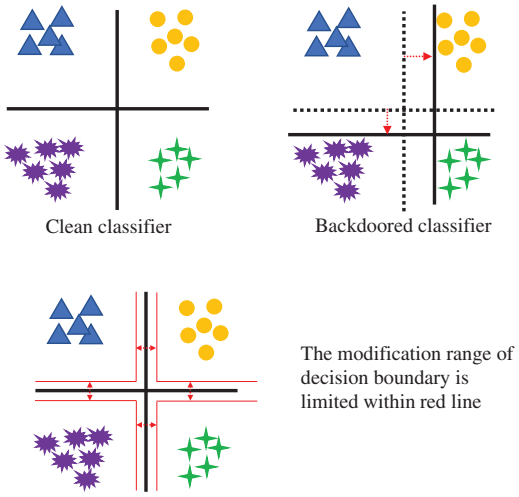


Figure 5: An illustration for the parameter clip. The black line represents the decision boundary

3.6 Cost Function

For the final cost function, we use the structural similarity index measure (SSIM) as a regular term to further improve the image visual quality. SSIM is widely used as an image quality metric. The clean images are used as the distortion-free image for reference. The cost function of our method can be expressed as (6)

$$L = l_{poisoned} + a \cdot l_{clean} + b \cdot l_{ssim}, \tag{6}$$

where a and b are two balance factors. $l_{poisoned}$ and l_{clean} are two cross-entropy loss of clean images and poisoned images, respectively. We update the parameters of the generator and classifier simultaneously to minimize the (6). Parameter clip will be executed after each updating of the parameters.

4 Experiments

4.1 Experimental Setup

To be comparable with previous methods, we conduct experiments on the CIFAR-10 and GTSRB datasets. The size of images in the two datasets is all 32×32. To evaluate our methods on large-size images, we randomly select ten categories in ImageNet [23] named selected-ImageNet. For each class, there are 1300 and 50 images for training and evaluation, respectively. Details of the three datasets are described in Table 2. As images in the MNIST dataset are almost binary images without any rich textural regions, they are not common in practical application. Our method aims to hide the trigger into the texture adaptively and does not arouse visual abnormalities. As a consequence, we do not evaluate our method on the MNIST dataset. We use Pre-activation ResNet-18 [24] as the classifier for

CIFAR-10 and GTSRB datasets. The selected-ImageNet classifier is the fine-tuned model of ResNet-18, which is trained on the original ImageNet. The target label of the backdoor is set to "0".

Table 2: Detailed information of the datasets

Dataset	Labels	Size	Classifier	Number of images
CIFAR-10	10	$32 \times 32 \times 3$	PreActRes18	60000
GTSRB	43	$32 \times 32 \times 3$	PreActRes18	50000
Selected-ImageNet	10	$128 \times 128 \times 3$	ResNet-18	13500

4.2 Visual Evaluation

We set our backdoor attack model as the single-target attack, in which all images superposed with the pre-defined trigger will be identified as the backdoor target label. Fig. 6 shows the visual quality of our method without parameter clip, which ensures the best visual performance. The images on the left, middle, and right are clean images, malicious images with the trigger, and the trigger, respectively. We employ MPM-based update strategy in the CIFAR-10 and GTSRB, and HardTanh-based update strategy in selected-ImageNet. The combination of ± 1 and ± 2 MPM ensures that the maximum modification is less than 3 in the CIFAR-10 and GTSRB. In the selected-ImageNet, we set the maximum modification as 10. These modifications are small enough not to be perceived by humans, and most are located in rich textural regions, which leads to good visual quality.

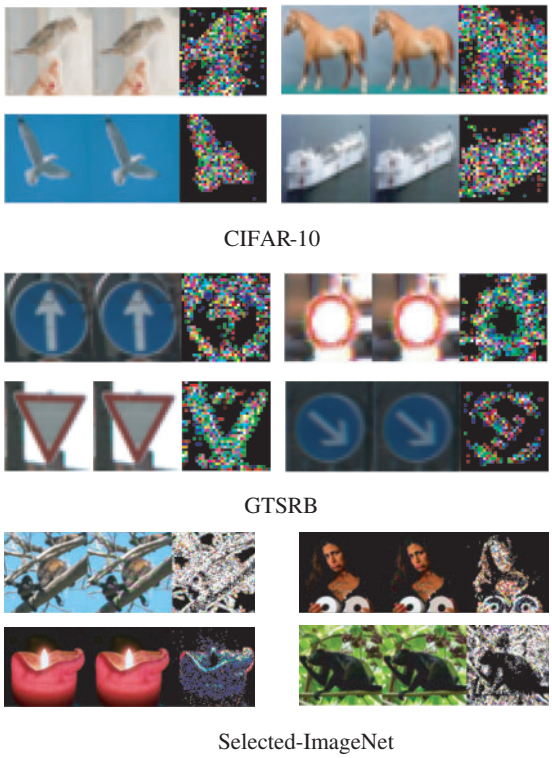


Figure 6: The visualization of our method without parameter clip over three datasets

Fig. 7 shows our method using the parameter clip and other methods. Parameter clip restricts the modification range of the classifier, which increases the difficulty of implanting the backdoor into a benign classifier. Therefore, we enhance the strength of the trigger to achieve a good backdoor attacker success rate. We use the Hardtanh-based update strategy and set the threshold to 12 and 40 for CIFAR-10 and GTSRB, respectively. The experimental results indicate the visual quality is slightly inferior to the images without the limation of parameter clip. However, the results are still much better than the BadNets and dynamic backdoor. The triggers in the other two methods are much obvious and easy to be perceived. Users can hardly find the anomaly of poisoned images generated by our method without their corresponding clean images as reference.

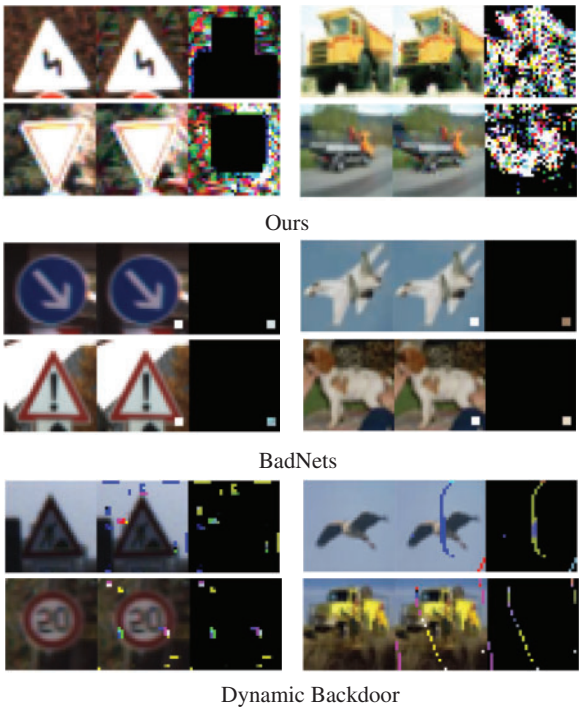


Figure 7: The visualization of our method with parameter clip

4.3 Attack Ability

We consider the backdoor attack success rate (BASR) and the impacts on the original accuracy (OA) over normal classification in terms of attack ability. The accuracy of clean images on the classifier without backdoors named OA-C. The detailed information and hyperparameter setting are illustrated in Table 3 and corresponding images have been shown in Subsections 4.1 and 4.2. For all three datasets and various hyperparameters, the BASR is almost 100%, while the original accuracy over clean images has only slightly dropped.

These experiments indicate that our method can make the classifier misclassify for images with their unique trigger and have a good performance on the benign inputs. We notice that with the restriction of parameter clip, the BASR is slightly inferior to the attack without the parameter clip even we have enhanced the strength of the trigger. But the attack success rate still exceeds 99%. For our application scenario, the user can not perceive the anomaly of the well-trained model given by an outsourcing computation provider.

Table 3: Detailed information of attack ability

Dataset	OA-C(%)	Generator	Hyperparameter	BASR(%)	OA(%)
CIFAR-10	95.02	MPM-based	Combation of ± 1 and ± 2 MPM ; $r =+\infty$	100	94.59
		HardTanh-based	i =12 ; r = 1e-3	99.07	94.82
GTSRB	99.50	MPM-based	Combation of ± 1 and ± 2 MPM; $r =+\infty$	99.79	99.37
		HardTanh-based	$i = 40$; r = 1e-3	99.02	99.36
Selected-ImageNet	92.66	HardTanh-based	$i = 10$; $r =+\infty$	99.80	91.60

4.4 Defense Experiments

We evaluate our attack approaches against the classical model-based defense Neural Cleanse [18] and pruning [19] on the CIFAR-10 and GTSRB dataset. Neural Cleanse is an effective detector to reveal whether a well-trained network contains the backdoor. Common backdoor changes the decision boundary of the classifier, which generates a shortcut between the target label and others. Neural Cleanse measures the minimum modification to modify all clean labels to a certain label. In the classifier containing a backdoor, the pattern of the target label is much smaller than others. Neural Cleanse detects the backdoor by the Anomaly Index metric and sets 2 as the threshold. As shown in Fig. 8, our method with parameter clip can pass this detector. The explanation is that each image owns its adaptive trigger, and Neural Cleanse cannot find a universal pattern for the entire dataset.

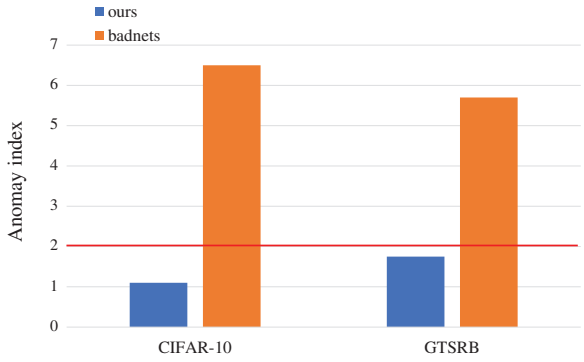


Figure 8: Experimental results of Neural Cleanse

Fig. 9 takes the CIFAR-10 as an example and shows some reversed candidate triggers by Neural Cleanse. The trigger of our method cannot be reversely constructed. Fig. 10 illustrates the minimum modification for each label over CIFAR-10 by Neural Cleanse. We take the label "plane" as an example, and the ordinate represents the minimum modification for all images required to cause the classifier to misclassify all images as "plane". If there is a shortcut for the target label caused by backdoor attacks, the minimum modification for the target label will significantly smaller than other labels. Our method keeps the minimum modification for the target label reasonable and will not arouse abnormally small outliers to be detected as a backdoor.

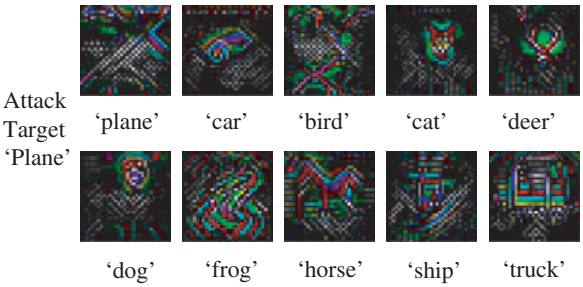


Figure 9: The reversed candidate triggers for each class on CIFAR-10 dataset

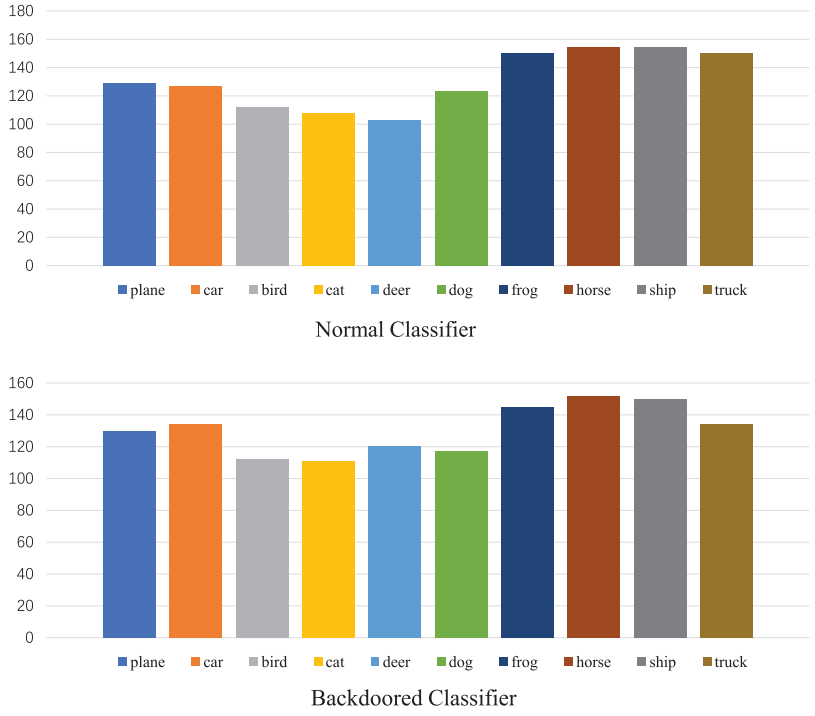


Figure 10: The minimum modification for each label over CIFAR-10 generated by Neural Cleanse

Apart from Neural Cleanse, pruning focuses on neuron analyses. It mitigates and removes the backdoor by pruning the neurons which are inactive on clean images. We also evaluate our method against pruning. Fig. 11 presents the accuracy of the original task and backdoor attack with respect to the number of neurons on the CIFAR-10 and GTSRB. For both datasets, the accuracy of clean images drops a lot when the backdoor is removed. Especially in the CIFAR-10 dataset, only when the 509th neuron (512 neurons in total) is pruned as the accuracy of backdoor decreases significantly, but the accuracy of clean images drops to 55%.

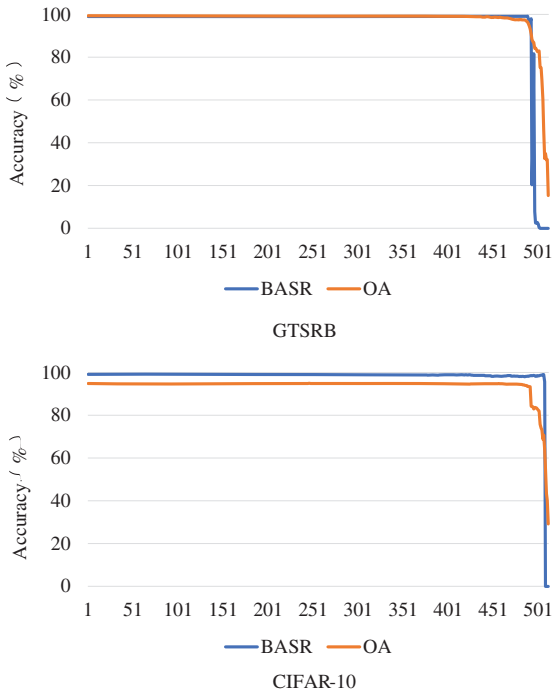


Figure 11: Experimental results for pruning on the CIFAR-10 and GTSRB dataset

4.5 Ablation Studies

To demonstrate the efficacy of our adaptive mask, we train the classifier without an adaptive mask and using common L_2-norm loss. From Fig. 12, we can see that the output trigger of a generator for all images is almost identical. Especially, the trigger is fixed in the upper left corner in the GTSRB dataset. In this case, our method degenerates into BadNets. The universal trigger can easily make shortcuts between different labels, which makes it vulnerable to be detected and reversely constructed by Neural Cleanse. The adaptive mask module is an essential element for creating a unique trigger for each image. Threshold T mentioned in (4) is an important hyperparameter used to control the size of masks. If the size of adaptive masks is too small, it may cause the backdoor attack to fail. Therefore, we employ the HILL with $T = 1$ and edge detection with $T = 10$ for GTSRB and CIFAR-10, respectively. Hyperparameter b is an important factor controlling the entire image quality of poisoned images. We set b as 2 when the maximum modification range is set to 40 in GTSRB. For other cases, the maximum modification range is relatively small, and we set hyperparameter b as 0.

Besides the adaptive mask module, parameter clip is an important item of countering detectors in our method. Parameter clip makes sure that the classifier implanted with the backdoor can evade detection. We remove the parameter clip and limit the intensity of the trigger to less than 3. The visual performance has been shown in Fig. 6. In this case, the anomaly index given by Neural Cleanse becomes 8, which is much larger than the threshold 2.

We analyze how the hyperparameter a affects the backdoor attack success rate and accuracy over the clean images. We train the classifier on the CIFAR-10 and GTSRB with a varying from 0.2 to 1.0. The accuracy of both clean and malicious images is almost unchanged with different a. As shown in Fig. 13, our method is robust to selecting a. We set a as 1 in all experiments.

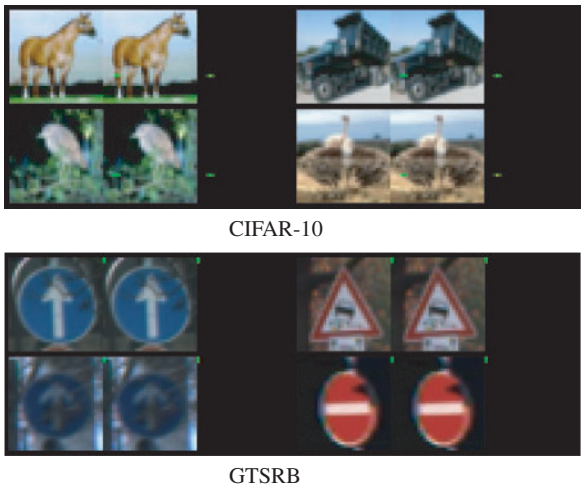


Figure 12: Experimental results for the backdoor attack without adaptive mask module

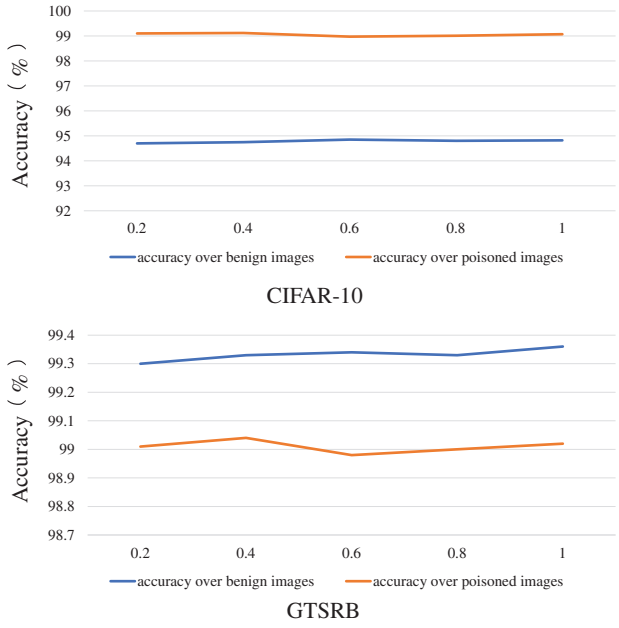


Figure 13: Poisoned and clean images accuracy on the CIFAR10 and GTSRB when changing *a*

5 Conclusions

In this paper, we propose a content-based adaptive backdoor attack. This is the first work generating the adaptive trigger for each clean image. The backdoored classifier performs well on the clean images and is totally fooled when the trigger appears. Our method achieves high attack ability and stealthiness of both the backdoored classifier and poisoned images compared with existing attacks. This work reveals an insidious backdoor attack, which brings a great challenge to the AI security field. In the future, we will extend our methods to other applications.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA.
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. (2017). Attention is all you need. *Advances in Neural Information Processing systems*. arXiv preprint arXiv:1706.03762.
3. He, K., Gkioxari, G., Dollár, P., Girshick, R. B. (2017). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42,* 386–397.
4. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). Improving language understanding by generative pre-training. *Computer Science*.
5. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. et al. (2019). Language models are unsupervised multitask learners. *OpenAI Blog, 1(8),* 9.
6. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. et al. (2020). Language models are few-shot learners. arXiv preprint arXiv: 2005.14165.
7. Alfeld, S., Zhu, X., Barford, P. (2016). Data poisoning attacks against autoregressive models. *AAAI Conference on Artificial Intelligence,* Phoenix, Arizona.
8. Gu, T., Dolan-Gavitt, B., Garg, S. (2017). Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv: 1708.06733.
9. Liu, Y., Ma, S., Aafer, Y., Lee, W. C., Zhai, J. et al. (2018). Trojaning attack on neural networks. *NDSS*.
10. Nguyen, A., Tran, A. (2020). Input-aware dynamic backdoor attack. arXiv preprint arXiv: 2010.08138.
11. Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C. et al. (2018). Poison frogs! Targeted clean-label poisoning attacks on neural networks. arXiv preprint arXiv: 1804.00792.
12. Xue, M., He, C., Wang, J., Liu, W. (2022). One-to-n & n-to-one: Two advanced backdoor attacks against deep learning models. *IEEE Transactions on Dependable and Secure Computing, 19(3),* 1562–1578. https://doi.org/10.1109/TDSC.2020.3028448
13. Saha, A., Subramanya, A., Pirsiavash, H. (2020). Hidden trigger backdoor attacks. *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 34, pp. 11957–11965.
14. Tran, B., Li, J., Madry, A. (2018). Spectral signatures in backdoor attacks. *NeurIPS*.
15. Cheng, H., Xu, K., Liu, S., Chen, P. Y., Zhao, P. et al. (2020). Defending against backdoor attack on deep neural networks. arXiv preprint arXiv: 2002.12162.
16. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. et al. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–629. Venice.
17. Zhu, L., Ning, R., Wang, C., Xin, C., Wu, H. (2020). Gangsweep: Sweep out neural backdoors by gan. *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 3173–3181. Seattle, USA.
18. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B. et al. (2019). Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 707–723. IEEE, San Francisco, USA.
19. Liu, K., Dolan-Gavitt, B., Garg, S. (2018). Fine-pruning: Defending against backdooring attacks on deep neural networks. *International Symposium on Research in Attacks, Intrusions, and Defenses,* pp. 707–723. Springer, Heraklion, Greece.
20. Li, B., Wang, M., Huang, J., Li, X. (2014). A new cost function for spatial image steganography. *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 4206–4210. IEEE, Paris.

21. Ronneberger, O., Fischer, P., Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer, Munich, Germany.
22. Zhong, N., Qian, Z., Wang, Z., Zhang, X., Li, X. (2020). Batch steganography via generative network. *IEEE Transactions on Circuits and Systems for Video Technology, 31(1),* 88–97.
23. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K. et al. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, Miami Beach, FL, USA, Weather Forecast.
24. He, K., Zhang, X., Ren, S., Sun, J. (2016). Identity mappings in deep residual networks. *European Conference on Computer Vision*, pp. 630–645. Springer, Amsterdam, The Netherlands.