Why is Your Trojan NOT Responding? A Quantitative Analysis of Failures in Backdoor Attacks of Neural Networks

Xingbo Hu[1,2][0000−0001−5963−3513], Yibing Lan[1,2][0000−0002−5172−9497], Ruimin Gao[3][0000−0002−4728−8000], Guozhu Meng[1,2⋆][0000−0001−6388−2571], and Kai Chen[1,2]

[1] SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] Mathematics and Statistics, University of Victoria

Abstract. Backdoor has offered a new attack vector to degrade or even subvert deep learning systems and thus has been extensively studied in the past few years. In reality, however, it is not as robust as expected and oftentimes fails due to many factors, such as data transformations on backdoor triggers and defensive measures of the target model. Different backdoor algorithms vary from resilience to these factors. To evaluate the robustness of backdoor attacks, we conduct a quantitative analysis of backdoor failures and further provide an interpretable way to unveil why these transformations can counteract backdoors. First, we build a uniform evaluation framework in which five backdoor algorithms and three types of transformations are implemented. We randomly select a number of samples from each test dataset, and then these samples are poisoned by triggers. These distorted variants of samples are passed to the trojan models after various data transformations. We measure the differences of predicated results between input samples as influences of transformations for backdoor attacks. Moreover, we present a simple approach to interpret the caused degradation. The results as well as conclusions in this study shed light on the difficulties of backdoor attacks in the real world, and can facilitate the future research on robust backdoor attacks.

Keywords: Deep learning · Backdoor attack · Robustness · Transformation · Interpretability.

1 Introduction

Deep learning has gained tremendous success in a variety of fields, such as image classification, speech recognition, natural language processing, and gaming. Moreover, its superior performance motivates the application in the security-critical areas including autonomous driving, face payment, and identity verification. However, deep learning has proved to be vulnerable and poses a great risk to its users. Since Szegedy *et al.* [33] first proposed the existence of adversarial examples in deep learning, researchers and practitioners have fleetly

⋆ Corresponding author: mengguozhu@iie.ac.cn

payed attention to issues of security and privacy in deep learning. It reveals that deep learning is suffering from adversarial attacks, model inversion, model extraction [11] and backdoor attacks [12]. Compared to other attacks, backdoor is more like an intentional attack initialized by a miscreant while the others are more like a special vulnerability of deep learning models. In a typical backdoor attack, training data is poisoned with well-crafted samples [8, 22]. If an innocent developer trains a classification model with poisoned data, a backdoor is consequently implanted and the attacker can make the model output a chosen result as expected.

Backdoor attacks can incur severe damages and even threaten people's safety. If one face recognition model is implanted with a backdoor, Bob with a sticker on his face may deceive the model and buy a lunch on Alice's bill [3]. Even worse, an object detection system in a self-driving car may misclassify one STOP sign as a 30km/h speed limit due to an unconscious backdoor inside. This error can cause a serious traffic accident. To counteract this attack, prior studies have developed a number of techniques to detect trigger samples [7, 5], reverse engineer backdoors in models [21, 36, 13], and harden models blindly [20]. However, it is unclear and still unexplored whether backdoor attacks are effective as claimed in prior studies and what difficulties will be confronted to trigger a backdoor in reality.

There are many uncertainties in triggering a backdoor of deep neural networks so that the implanted trojan may not respond to the attacker. First, these influencing factors can come from the physical world [38]. Taking face recognition as an example, a facial image is photographed by an on-site camera and it is heavily impacted by the shooting distance, angle, focus position and illumination conditions. Every time the face recognition system is used in the physical world, the images are likely varying. Second, there are uncertainties influencing the success rate of backdoor attacks in the digital world. The image may go through pre-processing and transformations like cropping, scaling, and rotation. These transformations are attributed to either the defensive measures employed by the target model, or an adaption of the size of the model input. Given that, it is intriguing and important to explore why backdoor attacks fail and evaluate the robustness of backdoor attacks.

In this study, we conduct a quantitative analysis of failures of backdoor attacks in deep neural networks. To be more specific, we aim to transform model inputs and determine the influence to the prediction results. To this end, we first build a uniform evaluation framework that integrates two vanilla deep neural networks– LeNet [14] and ResNet-34 [9]. Five backdoor algorithms are implemented and we obtain 8 trojaned models as the test subject. We then employ three transformations on both input samples and backdoor triggers, and create a number of test samples as input. The robustness of backdoor attacks are quantified by attack success rate, through which we shed light on the different resilience to transformations. Last, we leverage the interpretability algorithm SmoothGrad [31] to explain how models make a right or wrong prediction under transformations.

Contributions. To sum up, we make the following contributions.

- *A uniform evaluation framework.* It implements five backdoor algorithms and contains 8 trojaned models. We develop three types of transformations for both input samples and backdoor triggers that can simulate the uncertainties in the physical and digital worlds.
- *A quantitative analysis of backdoor failures.* We have created 165,000 samples in total as model input and measured the influence of data transformations on backdoor attack success rate.
- *Explanation on backdoor robustness.* Through the interpretability analysis on backdoor robustness, we unveil how the poisoned samples are recognized by models and their prediction limits in the context of data transformations.

2 Related Work

Backdoor attacks in deep learning. Gu *et al.* [8] introduce a backdoor attack called BadNets for the first time, BadNets pollutes the training set, and achieves nearly 90% attack success rate in the traffic sign recognition. In order to enhance the concealment of injected backdoor, Chen *et al.* [3] mix backdoor triggers with a benign image. However, the attack success rate is proved to be related to the blending ratio. Besides, Li *et al.* [16] aim to regularize the disturbance trigger using p-norm so that the noise can be generated in a small range. Except for changing the trigger, Bagdasaryan *et al.* [2] claim that the loss computation was poisoned in the model-training code. On the one hand, Liu *et al.* [23] polluted the samples of reflected trigger image under common natural reflection phenomenon. Cheng *et al.* [4] define the trigger as style conversion, and train a generative adversarial network (GAN) model to generate polluted samples. Li *et al.* [18] plug the trigger into the image invisibly by image steganography. The above methods change the target labels of samples, so the attack can still be detected by checking the labels and samples. Therefore, a new attack strategy called clean label backdoor attack is proposed. Turner *et al.* [35] study the backdoor attack of clean labels at the beginning. They apply adversarial interference as a trigger to benign samples in the target category. Zhao *et al.* [42] extend this idea by using general perturbations in video classification. Saha *et al.* [29] minimize the distance of the target class in the feature space and inject the poisoned information into the image. Moreover, Quiring *et al.* [27] hide the trigger by zooming attack [39]. Apart from the deliberately designed triggers, some of the studies also use semantic shapes as backdoor triggers. For example, Bagdasaryan *et al.* [2] first explore this kind of backdoor attack named the semantic backdoor attack. Lin *et al.* [19] design hidden backdoor which can be activated by the combination of certain objects. In addition, some non-poisoning attacks have also been researched. For instance, Dumford *et al.* [6] explore non-poisoning backdoor attack and focus on modifying the parameters of models. Besides, Rakin *et al.* [28] consider to insert a target trojan during the training process. Tang *et al.* [34] introduce malicious backdoor module as trigger. As for the defense of backdoor attacks, some solutions have been proposed like unlearning [10].

Robustness evaluation of backdoor attacks. The current research about the robustness of backdoor is basically sketchy. Weng *et al.* [37] analyze the relationship between the robustness of backdoor attacks and adversarial attacks. Xue *et al.* [40] demonstrate that the attack with a static trigger is vulnerable, and much less effective in the physical world. Furthermore, Li *et al.* [17] summarize that when triggers in testing images are not consistent with another trigger used for training, the attack may be unstable. Therefore, The transformation-based-pre-processing (e.g., flipping and scaling) on the testing image before prediction will sharply decrease attack success rate. Pasquini *et al.* [26] transform the triggers with typical image processing operators of varying strength, and discuss the results of the backdoored DNN. The response of geometric and color transformations suggests that the change of the trigger geometry and partial occlusion of trigger can lower the success rate. *Differently, our study considers a variety of backdoor algorithms and employs three types of transformations to measure their robustness. Moreover, we try to explore how backdoor models view trigger samples and the transformed.*

3 Preliminary & Overview

3.1 Backdoor Attacks

Deep learning can be interpreted as a process to learn an abstraction of massive amounts of data via multi-layer neural networks. For the supervised learning, it acquires a well-labeled data set $\{\mathcal{X}, \mathcal{Y}\}$ and computes an optimal parameter θ for the neural network F, that is, $F_\theta : \mathcal{X} \to \mathcal{Y}$. The model F_θ is correct under a certain probability where $F_\theta(\mathcal{X}) = \mathcal{Y}^*$ and the model accuracy can be computed with the portion of different elements between $\mathcal{Y}$ and $\mathcal{Y}^*$. Prior studies [8, 3] show that the training data can be maliciously crafted to introduce a backdoor in a neural network, i.e., backdoor attack. As such, the trojaned model can output the attacker-chosen label via the trigger. Without loss of generality, we define backdoor attacks in neural networks as follows.

Definition 1 ***(Backdoor)***. *Given a trojaned neural network F_{θ^-}, there exists a set of samples X that will be classified as a fixed class (e.g., y_t) if they are decorated with a specific trigger t. That is, $F_{\theta^-}(x \oplus t) = y_t$.*

To evaluate the attack success rate (ASR) of backdoor attacks, an attacker can randomly choose a set of clean samples $\mathcal{X}$ that are decorated with the fixed trigger t, and determine how many percent of samples are classified as y_t. More rigorously, the clean samples $\mathcal{X}$ should not be of the factual class y_t, i.e., $\forall x \in \mathcal{X}, F_\theta(x) \neq y_t$. For ease of understanding, all the notations in this paper have been summarized in Table 1.

3.2 Approach Overview

In this study, we aim to evaluate the robustness of mainstream backdoor attack methods in the scenario of the physical world. The samples that serve as

Table 1: Notations in this paper

Notation	Description
$\mathcal{X} \times \mathcal{Y}$	the labeled data with input space $\mathcal{X}$ and label space $\mathcal{Y}$
$\mathcal{X} \times \mathcal{Y}^*$	the factual labels $\mathcal{Y}^*$ for a given data $\mathcal{X}$
θ	model parameters
F_θ	a neural network with the parameter θ
F_{θ^-}	a backdoored neural network with the parameter θ^-
$\mathcal{X}^b$	the poisoned data for training
$\mathcal{Y}^b$	the model's predictions for the poisoned data
$x \oplus t$	a data point with trigger t attached
y_t	an attacker-chosen class for backdoor attacks
$\mathcal{S}$	the rendering space for a trigger

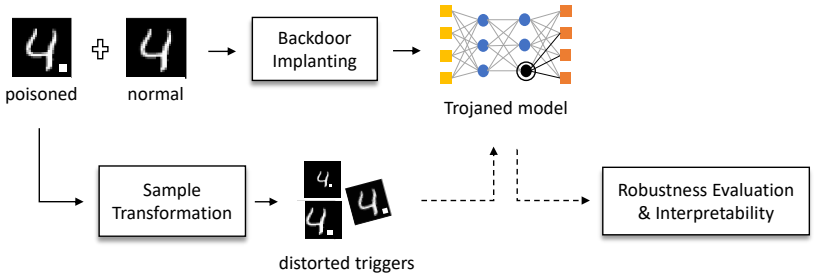


Fig. 1: System overview

intentional triggers may be affected by the realistic environment that can undermine the efficacy of trojaned models. Fig. 1 shows the overview of this study. In particular, we first create a poisoned data set with diverse backdoor triggers and insert a backdoor into the trained model with five backdoor methods (i.e., *backdoor implanting*). All the poisoned samples undergo transformations such as scaling and rotation (i.e., *sample transformation*). The distorted samples are then passed to the trojaned model. Last, we evaluate the robustness of trojaned models in front of distorted samples and provide an interpretability analysis (i.e., *robustness evaluation & interpretability*).

4 Methodology

In this section, we present the details for the methodology.

4.1 Backdoor Implanting

In a conventional process of backdoor implanting, the attacker needs to first design a trigger, determine the optimization objective, and train a model with poisoned and non-poisoned samples.

Trigger Design. A trigger is the pattern used to poison training data and activate backdoors in a neural network [12]. In a backdoor attack, the original

image I can be represented as $I : \{\langle x, y, z\rangle\}$, where x and y are the x-coordinate and y-coordinate, respectively, and z is the RGB value for the pixel at $\langle x,\ y\rangle$. The poisoned image can be created by performing a pixel-wise computation with the trigger t in the same coordinate, i.e., $I \oplus t$. There are several intriguing properties of a trigger, such as *size*, *color*, *texture*, *naturalness*, *detectability*, etc. A line of work has developed different types of triggers to inject a backdoor. In this paper, we select five representative trigger designs in terms of these properties. In particular, the selected triggers are from the following studies.

(a) Square (b) Smile (c) UAT (d) DFST (e) Reflection

Fig. 2: Examples of backdoor trigger

1. BadNets. The trigger is a monochromatic square usually put at the corner of the original image as shown in Fig. 2 (a) [8]. As the background color of MNIST images is black, we render the trigger as a white square as $\{\langle x, y, z\rangle | \langle x, y\rangle \in \mathcal{S} \ \wedge z = 255\}$, where $\mathcal{S}$ is the rendering space of the trigger. Therefore, the poisoned image can be obtained by overlaying the trigger on image I.
2. Blended Injection in [3]. As shown in Fig. 2(b), it blends an image with the trigger in the following manner.

$$z_i = \lambda z_i + (1 - \lambda) z_t, \text{ where } \langle x_i, y_i\rangle \in \mathcal{S} \wedge \lambda \in [0,\ 1] \tag{1}$$

Noted that λ is a parameter to control the blending ratio, and the trigger does not exist when $\lambda = 1$ and it is a BadNets trigger if $\lambda = 0$.
3. Deep Feature Space Trojaning (DFST). The trigger is not a fixed pattern in DFST, but generated dynamically depending on the original image. In particular, the attacker first trains a CycleGAN [4] that incorporates styles (e.g., sunset). The CycleGAN can then generate a stylized image as poisonous samples.
4. Refool. It proposes a new type of triggers using the natural reflection phenomenon [23] as indicated in Fig. 2 (d). Reflection is very common in reality when an object has a smooth surface. Assume the reflection image is x_R, the poisoned data x^b can be computed as

$$x^b = x + x_r \otimes \lambda \tag{2}$$

where λ is a convolution kernel that controls the reflection effect. Three effects are: reflection from the same-depth layer, out-of-focus layer and ghost effect.
5. Universal Adversarial Trigger (UAT) [42]. Similar to DFST, UAT is an optimized trigger based on the training data. It is initialized with random values

for a fixed-size area, i.e., $t : \{\langle x, y, z\rangle | \langle x, y\rangle \in \mathcal{S} \wedge z \in \mathbb{R}^N\}$. The poisoned sample can be represented as $x^b = x \oplus t$ and trigger t can be optimized through:

$$t = \arg\min_t \Sigma_{x_i \in \mathcal{X}} - \frac{1}{l} \Sigma_{j=1}^{l} y_j \log(h_j(x_i \oplus t)) \tag{3}$$

where l is the number of output labels, y_j here is the probability of being class j, and $h(\cdot)$ is the softmax output of the model. In this manner, the attacker can generate a trigger that is universally workable for the training data.

Backdoor Training. Generally, a backdoor is installed by (re-)training the model on a poisoned data set with $\mathcal{X}^b$. It can be formulated as the following optimization objective.

$$\theta^- = \arg\min_{\theta^-} \ \Sigma_{x \in \mathcal{X}} \mathcal{L}(F_{\theta^-}(x), y) \tag{4}$$

where $\mathcal{L}$ is the loss function such as Cross Entropy [24]. To balance the predication accuracy for normal samples and attack success rate for backdoor samples, we choose 10% training samples randomly and put the trigger on them.

4.2 Sample Transformation

Backdoor attacks may not perform as effectively as claimed in prior studies due to data transformation in reality [26]. As aforementioned, there are many uncertainties from one sample with trigger to the target model [41, 43]. These uncertainties can significantly affect attack success rate of backdoor attacks, i.e., one distorted trigger sample may fail to set the model on fire. Therefore, transforming input data can better illustrate how robust deep neural networks are when triggering a backdoor. In this study, we consider the following transformations.

Translation. In geometry, translation is to move a subject with a certain distance without rotating it. It often happens before an image is passed to the model. This transformation is eligible for both trigger and image. For instance, the focal point of one photographed image shifts by a small distance. The digital image has different sizes with the dimensions of model input and so it has to be cropped that can incur a translation of backdoor trigger in the range of image. To evaluate the stability of backdoor triggering, we translate the trigger in the image canvas by a certain distance (e.g., r) and create a number of patched images where the triggers' center is scattered on the ring of a circle with radius r. We also translate the entire image to simulate the process of image transformation in model defense [1]. There will be blanks in transformed images and we fill the blanks with the shift-out parts. To determine the largest translation distance, we employ a trial-and-error method to translate the unpatched image gradually unless it is wrongly classified by the model.

Scaling. One image and its trigger can be scaled outward or inward to imitate a varying focal length. We are then able to determine how backdoor attacks are affected by scaling. As for the image, we can cut out an area of same sizes as the original when the image is scaled out. There also exists the blank problem when shrinking images, and we fill the blanks with black color (i.e., *(0, 0, 0) in RGB*

images and 0 in grey-scale images). We call the function "PIL.Image.resize()" and use nearest neighbor interpolation (i.e., PIL.Image.NEAREST) to scale the image. Similar to translation, we have to determine what the largest scaling ratio is in the trial-and-error method.

Rotation. Image rotation is an image processing routine, commonly used in data augmentation [30] during model training. Moreover, the images captured from the physical world oftentimes suffer such transformations considering the photographer is not at the right front of the photo subject (i.e., with a varying angle). As a consequence, the image has a certain angle horizontally. In this study, we intend to explore whether a rotated image or trigger can affect the performance of backdoor attacks. Given one image, we invoke the function "PIL.Image.rotate()" in Python library PIL to rotate the target image with a certain angle. Similar to scaling, we also fill the blanks due to rotation with black color. It is noted that some triggers are rotational symmetry. For example, the white square in Fig. 2(a) is 4-fold rotational symmetry so we only rotate the trigger by less than 90° with a stride of 15° in our experiment.

4.3 Robustness Evaluation & Interpretability Analysis

Given the poisoned data $\mathcal{X}^b$ of n dimensions, the model produces n-dimensional $\mathcal{Y}^b$. The attack success rate (ASR_1) can be computed as: $|\{y|y \in \mathcal{Y}^b \wedge y = y_t\}|/n$. We can perform transformations, i.e., $T(\cdot)$, on $\mathcal{X}^b$ and obtain data $\mathcal{X}^{\tilde{b}}$. Similarly, the output labels for $\mathcal{X}^{\tilde{b}}$ are represented as $\mathcal{Y}^{\tilde{b}}$. Then, the ASR_2 is: $|\{y|y \in \mathcal{Y}^{\tilde{b}} \wedge y = y_t\}|/n$. As a result, the attack success rate of backdoors is dropped by ASR_1 - ASR_2 because of transformation $T(\cdot)$. To gain a finer influence function of transformations on ASR, we parameterize these transformations such as rotating triggers by 15° for one checkpoint. The details for the parameterized transformations are described in each experiment at Section 5.

With the decorated samples and their distorted ones, we intend to explain why it sometimes fails to trigger a backdoor in an interpretable way. In particular, we employ SmoothGrad to visualize the important regions in an image that are responsible for the decision. In this way, we attempt to explain how a trigger image has a label flip, i.e., from y_t to $\neg y_t$, after transformations, and whether the important regions stay unaltered with the unchanged output label.

5 Evaluation

5.1 Experimental Setup

In the experiment, we take two datasets in account: MNIST [15] and GTSRB [25]. More specifically, MNIST is an image dataset of handwritten digits from 0 to 9, and contains 60,000 samples. Each data point in the set is a greyscale image of size 28×28. GTSRB is an image dataset of 43 kinds of German traffic signs, and contains about 40,000 samples. Each data point is a RGB image of difference size, and it is resized to size 96×96 before fed to the network. We train

Table 2: Performance of the original and five backdoor models on the two datasets. "ACC. (%)" denotes the accuracy of the model's main task, "ASR (%)" is the attack success rate of backdoor methods in our framework.

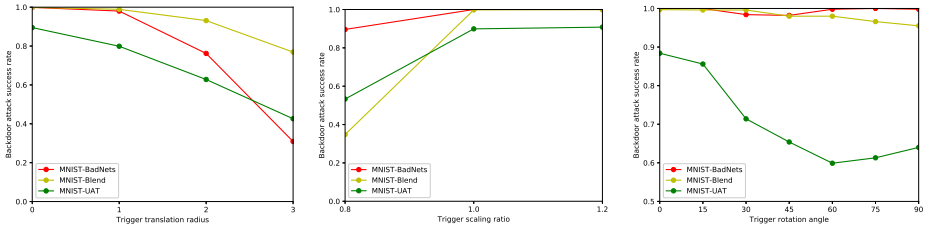
Model	Pattern	MNIST		GTSRB	
		ACC. (%)	ASR (%)	ACC. (%)	ASR (%)
Original	NA	99.13	-	97.05	-
BadNets	Square	99.17	99.95	97.32	95.79
Blend	Smile	99.02	99.78	96.58	96.63
UAT	UAT	97.01	90.06	94.30	82.60
DFST	Cezanne	-	-	95.27	98.30
Reflection	Apple	-	-	94.01	93.91

a LeNet [14] model on dataset MNIST and a ResNet-34 [9] model on dataset GTSRB since these two network structures achieve state-of-the-art performance on these datasets. As shown in Table 2, the accuracies of vanilla models are 99.13% and 97.05%, respectively.

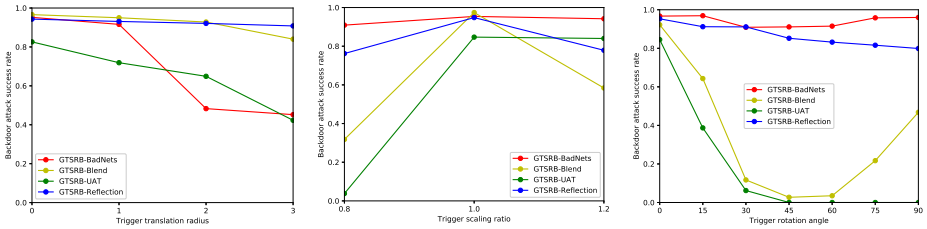
For these two benign models, we apply five mainstream backdoor algorithms, i.e., BadNets, Blend, DFST, Refool and UAT, to introduce a backdoor. The BadNets method is the first method to propose the concept of deep learning backdoor attacks, which is also the main attack idea based on data pollution-based backdoor attacks. It mainly prints trigger patterns such as small white squares into the sample. Since the trigger by BadNets is obvious and easy-to-detected, the subsequent backdoor attacks are dedicated to increase the concealment of trigger. In particular, the Blend method mixes the trigger and the sample in a certain ratio. When the mixing ratio is small, the image will be close to the original sample. The UAT method is to generate the general disturbance trigger in the original image that is different from other methods. It does not need to maliciously change the label of the poisoned sample when the backdoor is implanted. The Refool method is inspired by the natural reflection phenomena and implant a backdoor with a reflected image. The DFST method basically trains a style transfer model with CycleGAN, with which it transfers the style of the current image to another, for example in a painting style of cezanne.

Because DFST and Reflection backdor attacks are limited in RGB images, we only implement BadNets, Blend and UAT attacks on dataset MNIST and set their poisoning rates to 10% for all. For GTSRB dataset, we implement all 5 backdoor attacks and set poisoning rates of DFST and Reflection to 20%.

Experiment Parameters. To preserve as many characteristics of triggers as possible during transformation, we make a proper design for the position and size of original triggers. Specifically, we use a 5×5 white bottom-right square as BadNets triggers. For UAT attack, we use a 8×8 perturbation square on MNIST, and a 28×28 perturbation block on GTSRB. To avoid exceeding the boundary of an image, we leave 3 bottom-right pixels for BadNets and UAT triggers on MNIST and BadNets triggers on GTSRB, but 6 pixels for UAT triggers on GTSRB. We use a smile meme as Blend trigger and stylize the images with the painting style of cezanne in attack DFST. Since a cezanne-styled trigger does



(a) Trigger translation on MNIST (b) Trigger scaling results on MNIST (c) Trigger rotation results on MNIST



(d) Trigger translation on GTSRB (e) Trigger scaling results on GTSRB (f) Trigger rotation results on GTSRB

Fig. 3: Success rate of backdoor attacks under trigger transformations

not have a specific shape, we do not consider trigger transformations for this attack. For all experiments, we choose the first class as the target label y_t (e.g., 0 in MNIST and speed 20 limit sign in GTSRB).

5.2 Backdoor Robustness under Trigger Transformation

In this experiment, we employ translation, scaling and rotation on triggers to evaluate the robustness of backdoor. For each transformation, we select a suitable transformation range to inspect backdoor attack changes from a small sample set. Fig. 3 shows the results for applying these three transformations. The horizontal axis of each plot represents specific transformation parameters, and the vertical axis is backdoor attack success rate.

For trigger translation, we measure ASRs when the trigger is moved to a circle with a radius of 1, 2, 3 pixels, respectively. It is observed that as the translation distance increases, the attack success rate of all backdoor attacks decreases in Fig. 3a and 3d. However, the decline of the attack success rate varies from attack methods. BadNets have the fastest attenuation and this is partly because BadNets is the simplest method which overlays triggers to a certain area of samples. The features remembered by the backdoor model are the simplest and most obvious. The ASR of simple features will be dropped drastically when the trigger moves a small distance. In addition, it may also be attributed to the overlap ratio between the implanted trigger and the original trigger. It is easier

to make overlap ratio smaller within a tiny amount of movement as the 5×5 square trigger is relatively small.

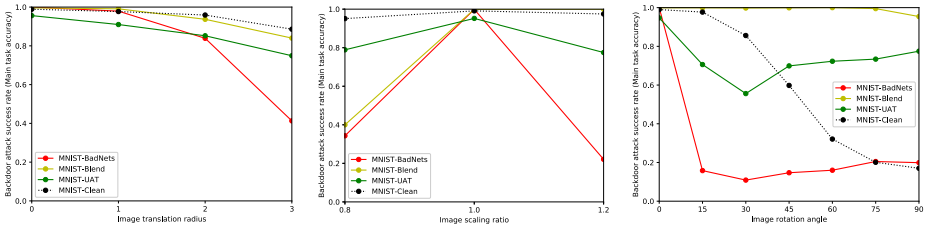
For trigger scaling, we amplify the trigger by 0.8, 1, or 1.2 times. It can be seen from Fig. 3b and 3e that when the trigger is reduced, the success rate of the backdoor attack is greatly reduced. At the same time, although the trigger is enlarged and deviated from the benchmark (i.e. the original trigger size), the ASR is slightly affected. This result indicates that the trigger feature is hard to be recognized by backdoored models after being compressed, and the details of the trigger are not lost after being zoomed in.

For trigger rotation, we rotate the trigger clockwise with the center of the original trigger and the rotation angle is from 0° to 90° with an interval of 15°. It can be seen from Fig. 3c and 4f that the success rate of backdoor attacks has decreased by varying degrees. For BadNets, since its trigger is a square that is symmetric, its ASR curve is symmetric as well. When it reaches 45°, the overlap between the transformed trigger and the original is the smallest, so that its ASR is the lowest. For UAT, the decline is most obvious, because the trigger of UAT is generated by the general perturbation value of an area in the lower right corner so that the model can maximize the output probability of the target label. When the UAT trigger is rotated, the value of the pixel is more likely to be different after the nearest neighbor algorithm, leading to the original neurons related to the target label cannot be activated as expected.

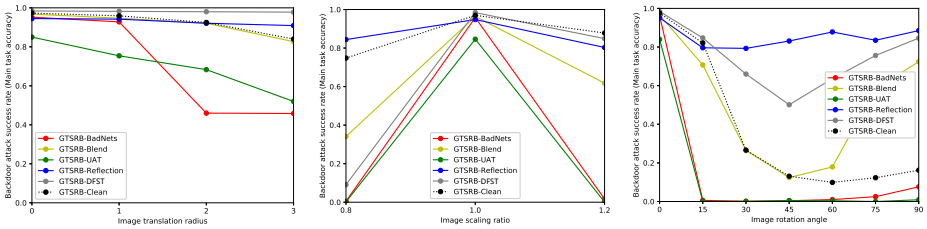
5.3 Backdoor Robustness under Image Transformation

Similarly, we perform translation, scaling, and rotation transformations on the backdoor samples with triggers. Then we measure the impact of these transformations on backdoor attack success rate. For each transformation, we also select an appropriate transformation range, and generate 1000 samples under different transformation configurations to test the robustness of the five backdoor attack methods. In order to maintain uniformity with trigger transformation and ease the comparison of the results, we have adopted the same transformation settings for these three attacks. However, the transformation may lead a sample to be an invalid one, i.e., a wrong predication is not due to the backdoor, but the heavily distorted image. So we also perform these transformations on clean samples and test the prediction accuracy for them as a benchmark. Fig. 3 reports the results of applying these three transformations. The horizontal axis of each plot represents the converted value of backdoor samples or clean samples, and the vertical axis represents the success rate of backdoor attacks or main attack accuracy, respectively. The black dotted line represents the accuracy of clean samples.

For image translation as Fig. 4a and 4d, we get a similar result of a decrease in attack success rate with that of trigger translation. As the translation radius increases, the attack success rates of all backdoor attacks decrease. However, with the exception of BadNets, other attack methods have a smaller decline in image translation than trigger translation because the BadNets implant a simple, small and obvious backdoor trigger. In other methods like blend, the implanted backdoor can occupy a large area in the image, which is different from BadNets.



(a) Image translation on MNIST (b) Image scaling results on MNIST (c) Image rotation results on MNIST



(d) Image translation on GTSRB (e) Image scaling results on GTSRB (f) Image rotation results on GTSRB

Fig. 4: Success rate of backdoor attacks under image transformations

It is implanted in a certain connection with the original image (i.e. the blend ratio). It implied that the translation of images does not completely destroy the connection between the trigger and the sample. We can observe that Reflection and DFST have strong robustness to small translation, and it is probably because these two backdoor methods have more abstract connections and larger trigger areas than other attacks.

For image scaling, the results we get are different from trigger scaling. Whatever the image is zoomed in or out, it will destroy the original attack success rate. The reason is more likely to be that the position of the trigger is also changed while the image is zoomed. Therefore, under the double change of the size and position of the trigger, the success rate of the backdoor attack has been greatly damaged. BadNets is still the most susceptible attack due to the simplicity of its triggers, and Reflection and DFST are the most robust attacks due to their abstract connections. As for UAT, it is surprisingly found that its robustness on different datasets has big difference.

For image rotation, the same result is slightly different from trigger rotation. The robustness of the BadNets method drops very clearly when the image is rotated, while the effects of other methods are similar. As with image scaling, image rotation will change the position of BadNets triggers. Different from the above two transformations, the image rotation plot shows that the accuracy of the main task declines sharply when clean samples are rotated, while some backdoor attacks such as Reflection and DFST maintain a smaller success rate

Table 3: Error rates of each transformation on GTSRB-BadNets. "Configuration" denotes the configuration used in the corresponding transformation, "Error Rate (%)" is the error predicted rate in test samples.

Transformation	Trigger		Image	
	Configuration	Error Rate (%)	Configuration	Error Rate (%)
Translation	3	0.00	3	1.07
Scaling	0.8	0.00	0.8	26.27
Rotation	45°	0.00	45°	89.54

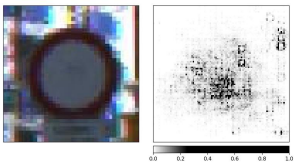
reduction than the baseline. This phenomenon shows that the backdoor task and the main task in the deep learning model have different robustness.

5.4 Explanatory Experiment

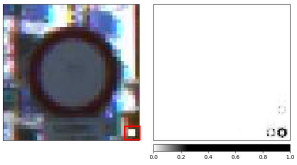
In order to understand why a backdoor sample fails, we employ a model intepretability method, i.e., SmoothGrad [31] to illustrate how the trojaned model recognize the sample. We select the model trained on the GTSRB dataset and attacked by BadNets as the illustrative example, since the model has relatively low robustness when facing different transformations. The first question we intend to answer is: when the backdoor sample is transformed, whether its predicted label is turned back to the true label. So we calculate the error rate for each transformation, where an error is counted if the predicted label of a transformed sample is neither a target backdoor label nor a true label, while the corresponding clean sample and the original backdoor sample are well predicted. As shown in Table 3, trigger transformations do not affect the predicted outputs while image transformations can damage the prediction results significantly. The error rates of image transformations are basically consistent with the accuracy reduction of the main task in Fig. 4.

Next, we analyze which parts of the samples contribute more to the predicted label by applying a model interpretability method called SmoothGrad. SmoothGrad is a gradient-based explanation method, and is suitable for various network models. It not only retains the advantages of Integrated Gradients [32] which handles the locality problem of gradient information, but also reduces the gradient noise. To verify the validity of SmoothGrad, we manually analyze 50 random well-predicted backdoor samples and check whether they can display the square trigger correctly. The result shows that 98% of samples can be explained correctly, proving its effectiveness.

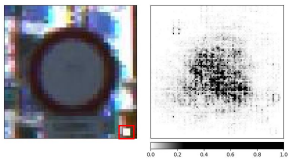
Fig. 5 reports a set of samples under different transformations, and the area enclosed by a red box is the backdoor trigger. Fig. 5a and 5b are the interpretability result for the clean sample and the original backdoor sample. The other figures are for the transformed samples that are predicted as the true label. It is observed from Fig. 5b that the trigger is well recognized by SmoothGrad, and plays a determinant role in making prediction. However, when the



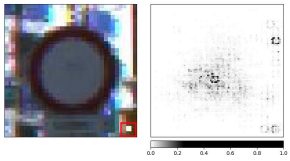
(a) Interpretability on clean sample



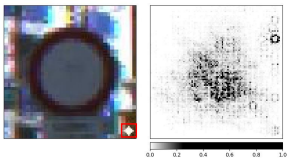
(b) Interpretability on BadNets sample



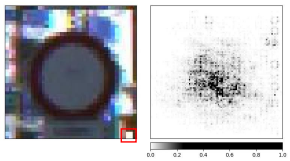
(c) Interpretability on trigger translation sample



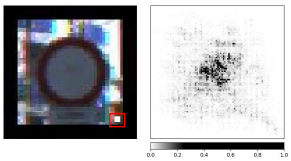
(d) Interpretability on trigger scaling sample



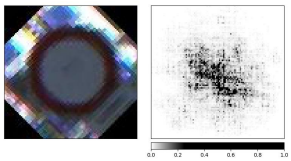
(e) Interpretability on trigger rotation sample



(f) Interpretability on image translation sample



(g) Interpretability on image scaling sample



(h) Interpretability on image rotation sample

Fig. 5: Interpretability results by SmoothGrad on BadNets-GTSRB

sample is transformed, the importance of the trigger on model decision almost disappears and the center area of the sample largely contributes to the predicted label. It means that the transformation can seriously affect the decision-making part of backdoor samples.

6 Discussion

Threats to validity. Our experiment results may be affected by some operations and settings. For example, due to the limitations of image resolution, when a small pixel trigger (5×5) rotates, the pixel area it occupies is very different from the actual rotation. For example, when a 3×3-pixel square is rotated with 15 degrees, the angle is rounded to zero in such resolution so the rotated image looks exactly the same as the original image. On the second place, the accuracy of the deep learning model will fluctuate in a small range according to the training setting during training, so the experimental results will have small changes. It may affect the absolute value of the model accuracy but will not change the fact of ASR reduction when applying transformations.

Insights from experiments. From the above experiment results, we summarize main observations and insights as follows:

- The backdoor attack is not as robust as imagined, and it is even difficult to trigger in the physical world. Whether it is for triggers or images, for most of the transformations, the accuracy of backdoor attacks will gradually reduce as they deviate from the baseline. Sometimes even a small change, such as shrinking the image, may make the backdoor ineffective. Therefore, a defender can perform these transformations on unknown inputs to defend potential backdoor attacks.
- Although these transformations will reduce the efficiency of backdoor attacks, we can also see that different backdoor attack methods exhibit different robustness characteristics in the face of transformations. For example, the Reflection and DFST methods basically maintain certain level of accuracy in the face of these transformations, and are more robust than other backdoor methods. Therefore, with sufficient resources (e.g., one can train a style cycleGAN model), the attacker can give priority to these attack methods.
- The design of backdoor trigger affects the robustness of backdoor methods significantly. For example, a symmetrical trigger can be resistant to transformations like rotation to a certain extent. Moreover, the more abstract connection between the trigger and the original sample is, the more robust the attack methods are. When designing the trigger, one can consider how to construct an abstract feature as a trigger to obtain a more robust method.

Future work. In this study, we only consider two representative datasets, where MNIST is a must-do dataset for testing a classification model and GTSRB contains images captured from the reality and is larger than another popular dataset CIFAR. In future, we intend to test on larger datasets, like ImageNet, to observe the performance of these attack methods under these transformations. At present, a trigger pattern is used for an attack method, while the design of the trigger may affect the results to a certain extent. So in the future, we can consider evaluating different triggers under the same attack method (e.g. symmetrical triggers and asymmetrical triggers). Our work explores the robustness of backdoor methods and models mainly from the transformation perspective, and future study can evaluate more properties of backdoor methods or models from other perspectives. Finally, a more robust backdoor method under these transformations is a direction of future investigation.

7 Conclusion

In this paper, we conduct a quantitative analysis of backdoor failures in neural networks, which are caused by data transformation. To this end, we build a uniform framework including five mainstream backdoor algorithms, and then train 8 trojaned models for evaluation. Three types of data transformations are performed on both images and triggers through which we obtain 165,000 evaluators. The experiment results quantify the influences on these transformations on the success rate of backdoor attacks. Last, we visualize how trojaned models recognize the images and their transformed variants.

Acknowledgement

We thank all the anonymous reviewers for their constructive feedback. IIE authors are supported in part of the National Key Research and Development Program (No. 2020AAA0107800), National Natural Science Foundation of China (No. U1836211, 61902395), the Anhui Department of Science and Technology (No. 202103a05020009), and Beijing Natural Science Foundation (No. JQ18011).

References

1. Agarwal, A., Singh, R., Vatsa, M., Ratha, N.: Image transformation-based defense against adversarial perturbation on deep learning models. IEEE Transactions on Dependable and Secure Computing **18**(5), 2106–2121 (2021)
2. Bagdasaryan, E., Shmatikov, V.: Blind backdoors in deep learning models. ArXiv **abs/2005.03823** (2020)
3. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. CoRR **abs/1712.05526** (2017)
4. Cheng, S., Liu, Y., Ma, S., Zhang, X.: Deep feature space trojan attack of neural networks by controlled detoxification. In: AAAI. pp. 1148–1156 (2021)
5. Doan, B.G., Abbasnejad, E., Ranasinghe, D.C.: Februus: Input purification defense against trojan attacks on deep neural network systems. In: ACSAC '20: Annual Computer Security Applications Conference, Virtual Event / Austin, TX, USA, 7-11 December, 2020. pp. 897–912. ACM (2020)
6. Dumford, J., Scheirer, W.: Backdooring convolutional neural networks via targeted weight perturbations. 2020 IEEE International Joint Conference on Biometrics (IJCB) pp. 1–9 (2020)
7. Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: STRIP: a defence against trojan attacks on deep neural networks. In: Balenson, D. (ed.) ACSAC. pp. 113–125. ACM (2019)
8. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. CoRR **abs/1708.06733** (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE CVPR. pp. 770–778 (2016)
10. He, Y., Meng, G., Chen, K., He, J., Hu, X.: Deepobliviate: A powerful charm for erasing data residual memory in deep neural networks. CoRR **abs/2105.06209** (2021), https://arxiv.org/abs/2105.06209
11. He, Y., Meng, G., Chen, K., He, J., Hu, X.: DRMI: A Dataset Reduction Technology based on Mutual Information for Black-box Attacks. In: Proceedings of the 30th USENIX Security Symposium (USENIX) (Aug 2021)
12. He, Y., Meng, G., Chen, K., Hu, X., He, J.: Towards Security Threats of Deep Learning Systems: A Survey pp. 1–28 (2020). https://doi.org/10.1109/TSE.2020.3034721
13. Kolouri, S., Saha, A., Pirsiavash, H., Hoffmann, H.: Universal litmus patterns: Revealing backdoor attacks in cnns. In: CVPR. pp. 298–307. Computer Vision Foundation / IEEE (2020)
14. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
15. LeCun, Y.: The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/ (2017)

16. Li, S., Xue, M., Zhao, B.Z.H., Zhu, H., Zhang, X.: Invisible backdoor attacks on deep neural networks via steganography and regularization. IEEE Transactions on Dependable and Secure Computing **18**, 2088–2105 (2021)
17. Li, Y., Zhai, T., Wu, B., Jiang, Y., Li, Z., Xia, S.: Rethinking the trigger of backdoor attack. ArXiv **abs/2004.04692** (2020)
18. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Backdoor attack with sample-specific triggers. ArXiv **abs/2012.03816** (2020)
19. Lin, J., Xu, L., Liu, Y., Zhang, X.: Composite backdoor attack for deep neural network by mixing existing benign features. CCS (2020)
20. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: Bailey, M., Holz, T., Stamatogiannakis, M., Ioannidis, S. (eds.) RAID. Lecture Notes in Computer Science, vol. 11050, pp. 273–294. Springer (2018)
21. Liu, Y., Lee, W., Tao, G., Ma, S., Aafer, Y., Zhang, X.: ABS: scanning neural networks for back-doors by artificial brain stimulation. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) CCS. pp. 1265–1282. ACM (2019)
22. Liu, Y., Ma, S., Aafer, Y., Lee, W., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks. In: NDSS. The Internet Society (2018)
23. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: ECCV (2020)
24. Murphy, K.P.: Machine learning - a probabilistic perspective. Adaptive computation and machine learning series, MIT Press (2012)
25. Neuroinformatik, I.F.: German Traffic Sign Detection Benchmark (GTSRB). https://benchmark.ini.rub.de/ (2019)
26. Pasquini, C., Böhme, R.: Trembling triggers: exploring the sensitivity of backdoors in dnn-based face recognition. EURASIP J. Inf. Secur. **2020**, 12 (2020)
27. Quiring, E., Rieck, K.: Backdooring and poisoning neural networks with image-scaling attacks. 2020 IEEE Security and Privacy Workshops (SPW) pp. 41–47 (2020)
28. Rakin, A.S., He, Z., Fan, D.: Tbt: Targeted neural network attack with bit trojan. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 13195–13204 (2020)
29. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: AAAI (2020)
30. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data **6**, 60 (2019)
31. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise (2017)
32. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3319–3328. PMLR (06–11 Aug 2017), https://proceedings.mlr.press/v70/sundararajan17a.html
33. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y. (eds.) ICLR (2014)
34. Tang, R., Du, M., Liu, N., Yang, F., Hu, X.: An embarrassingly simple approach for trojan attack in deep neural networks. KDD (2020)
35. Turner, A., Tsipras, D., Madry, A.: Label-consistent backdoor attacks. ArXiv **abs/1912.02771** (2019)

36. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019. pp. 707–723. IEEE (2019)
37. Weng, C.H., Lee, Y.T., Wu, S.H.: On the trade-off between adversarial and backdoor robustness. In: NeurIPS (2020)
38. Wenger, E., Passananti, J., Bhagoji, A.N., Yao, Y., Zheng, H., Zhao, B.Y.: Backdoor attacks against deep learning systems in the physical world. In: CVPR. pp. 6206–6215. Computer Vision Foundation / IEEE (2021)
39. Xiao, Q., Chen, Y., Shen, C., Chen, Y., Li, K.: Seeing is not believing: Camouflage attacks on image scaling algorithms. In: USENIX Security Symposium (2019)
40. Xue, M., He, C., Sun, S., Wang, J., Liu, W.: Robust backdoor attacks against deep neural networks in real physical world. ArXiv **abs/2104.07395** (2021)
41. Zha, M., Meng, G., Lin, C., Zhou, Z., Chen, K.: Rolma: A practical adversarial attack against deep learning-based LPR systems. In: Liu, Z., Yung, M. (eds.) Inscrypt. Lecture Notes in Computer Science, vol. 12020, pp. 101–117. Springer (2019)
42. Zhao, S., Ma, X., Zheng, X., Bailey, J., Chen, J., Jiang, Y.: Clean-label backdoor attacks on video recognition models. In: CVPR. pp. 14431–14440. Computer Vision Foundation / IEEE (2020)
43. Zhao, Y., Zhu, H., Liang, R., Shen, Q., Zhang, S., Chen, K.: Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) CCS. pp. 1989–2004. ACM (2019)