Compression-resistant backdoor attack against deep neural networks

Mingfu Xue[1] · Xin Wang[1] · Shichang Sun[1] · Yushu Zhang[1] · Jian Wang[1] · Weiqiang Liu[2]

Accepted: 12 March 2023

Abstract
In recent years, a number of backdoor attacks against deep neural networks (DNN) have been proposed. In this paper, we reveal that backdoor attacks are vulnerable to image compressions, as backdoor instances used to trigger backdoor attacks are usually compressed by image compression methods during data transmission. When backdoor instances are compressed, the feature of backdoor trigger will be destroyed, which could result in significant performance degradation for backdoor attacks. As a countermeasure, we propose the first compression-resistant backdoor attack method based on feature consistency training. Specifically, both backdoor images and their compressed versions are used for training, and the feature difference between backdoor images and their compressed versions are minimized through feature consistency training. As a result, the DNN treats the feature of compressed images as the feature of backdoor images in feature space. After training, the backdoor attack will be robust to image compressions. Furthermore, we consider three different image compressions (i.e., JPEG, JPEG2000, WEBP) during the feature consistency training, so that the backdoor attack can be robust to multiple image compression algorithms. Experimental results demonstrate that when the backdoor instances are compressed, the attack success rate of common backdoor attack is 6.63% (JPEG), 6.20% (JPEG2000) and 3.97% (WEBP) respectively, while the attack success rate of the proposed compression-resistant backdoor attack is 98.77% (JPEG), 97.69% (JPEG2000), and 98.93% (WEBP) respectively. The compression-resistant attack is robust under various parameters settings. In addition, extensive experiments have demonstrated that even if only one image compression method is used in the feature consistency training process, the proposed compression-resistant backdoor attack has the generalization ability to resist multiple unseen image compression methods.

Keywords Artificial intelligence security · Backdoor attack · Compression resistance · Deep neural networks · Feature consistency training

1 Introduction

Deep neural networks (DNN) have been widely used in many tasks. However, many researches have demonstrated that DNN are vulnerable to backdoor attacks. By embedding the backdoor into the model during training stage, the images with specific backdoor trigger can make the backdoored model output specified target label in the test stage.

However, most existing backdoor attacks are not robust to image compressions. Generally, images uploaded to the Internet will undergo image compressions, which are widely used to reduce the transmission and storage overhead [1]. Wan *et al.* [2] indicate that image compression distorts the feature of images, which will lead to accuracy degradation of DNN. In this paper, we study the impact of image compressions on DNN backdoor attacks. If backdoor instances are compressed, the backdoor trigger hidden in the images will be destroyed, which will seriously reduce the performance of the backdoor attack. As shown in Fig. 1, with the embedded backdoor trigger, the "James Marden" in the image will be incorrectly recognized as the target class "Aamir Khan" specified by the attacker. However, after image compression, the backdoor trigger is destroyed and "James Marden" will be correctly classified as "James Marden" by the model, i.e., the backdoor attack fails when the backdoor instance is compressed. To date, all the existing DNN backdoor attacks do not consider the problem of image compressions.

✉ Mingfu Xue
mingfu.xue@nuaa.edu.cn

Extended author information available on the last page of the article.

Published online: 12 April 2023

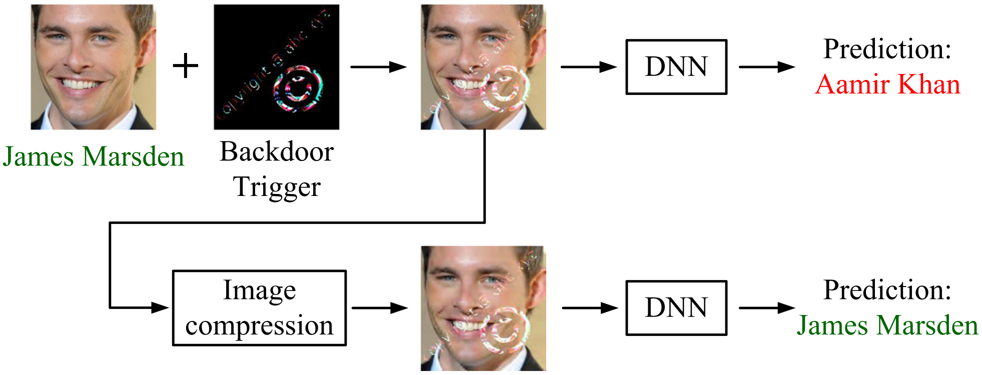


Fig. 1 The performance of normal backdoor attack after image compression

In this paper, we study the impact of image compression on backdoor attacks for the first time and propose the first compression-resistant backdoor attack method. We develop the compression-resistant backdoor attack by exploiting feature consistency training [2]. In the training stage, both the backdoor instances and their compressed versions are used to train the DNN. At each iteration of the training, the feature of both backdoor images and their compressed versions are extracted by internal layers of the DNN. The extracted features will be used to minimize the difference between normal backdoor images and compressed backdoor images. As a result, the trained DNN will treat compressed images as normal backdoor images in the feature space. Moreover, three image compression methods (i.e., JPEG [3], JPEG2000 [4] and WEBP [5]) are considered simultaneously during the feature consistency training, so that the proposed backdoor attack can be robust to multiple image compression algorithms. In the test stage, even if backdoor instances are compressed by an unknown image compression method, the proposed backdoor attack is still able to achieve a high attack success rate. Experimental results demonstrate that the performance of compression-resistant backdoor attack is significantly improved compared to normal backdoor attacks. Meanwhile, the proposed backdoor attack is able to resist a variety of compression methods without performance degradation. In addition, extensive experiments have demonstrated that, even if only one image compression method is used in the feature consistency training process, the compression-resistant backdoor attack can resist multiple "unknown" image compressions which are not considered in the training process.

The main contributions of this paper are threefold:

- We reveal that the existing backdoor attacks are vulnerable to image compressions. As a countermeasure, a compression-resistant backdoor attack method is proposed. To the best of our knowledge, this is the first compression-resistant backdoor attack against Deep Neural Networks, which is an advanced variant of backdoor attack.
- We strengthen the proposed backdoor attack using three different image compression methods (JPEG [3], JPEG2000 [4] and WEBP [5]) during the feature consistency training, so that the proposed attack is robust to different image compression methods.
- Experimental results have demonstrated that, even if only one image compression method is considered during training, the proposed compression-resistant backdoor attack has the generalization ability to resist multiple unknown image compression methods.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. The proposed method is presented in Section 3. Experimental results are discussed in Section 4. This paper is concluded in Section 5.

2 Related work

In this section, first, related works on existing backdoor attacks are reviewed. Second, the few compression-resistant works in the deep learning area (but not for backdoor attacks) are discussed.

2.1 Backdoor attack

A number of backdoor attacks against deep learning models have been proposed. Gu *et al.* [6] use a fixed square pattern as the backdoor trigger to launch the backdoor attack. Chen *et al.* [7] use three methods (i.e., blending, accessory and blended accessory) to construct triggers and generate backdoor instances, which are injected into the training set to embed the backdoor during model training. Liu *et al.* [8] generate the backdoor trigger by reverse engineering the DNN, so that attackers can conduct backdoor attacks without the access to the original training dataset.

Some recent works aim at improving the concealment of backdoor attacks [9–11]. Zhong *et al.* [11] use a perturbation mask as the backdoor trigger, which is invisible. Li *et al.* [9] conduct invisible backdoor attacks in which they use image steganography and L_p regularization to generate the backdoor trigger respectively. Zhang et al. [10] generate the backdoor trigger based on image structures. Since the information of image structures is difficult to be perceived by humans, using image structures to generate the backdoor trigger can make the backdoor attack concealed.

The backdoor triggers in most of the backdoor works are static patterns, which means the backdoor trigger is a single pattern with a fixed position. Recently, few works [12, 13] propose backdoor attacks with flexible triggers. Xue *et al.* [12] develop multi-target backdoor attack and multi-trigger backdoor attack. In the multi-target attack, the attacker can use different intensities of the backdoor trigger to control different attack targets. In the multi-trigger attack, the target can only be triggered when all the backdoor triggers are appeared in a backdoor instance. Salem *et al.* [13] design a dynamic backdoor attack, where the backdoor trigger can be flexibly placed in a random position of the image and the backdoor trigger could be a random pattern.

The above mentioned works focus on the digital domain, which does not consider the constraints (e.g., angle, distance) in real physical world. To this end, Xue *et al.* [14] consider the attack scenarios in real world and propose physical transformations for backdoors. Various transformations are performed on the backdoor instances during the training process to enhance the robustness of the backdoor attack in real physical world.

2.2 Compression-resistance in deep learning area

Recently, a few researches have shown that DNN are vulnerable to image compressions. In order to generate the JPEG compression-resistant adversarial examples, Shin *et al.* [15] add a differentiable approximation to JPEG compression during the adversarial examples generation process. The approximation operation can maximize the prediction difference between the original image and the compressed adversarial example. Wang *et al.* [1] generate compression-resistant adversarial examples to protect the privacy in photos on social networks. They train an encoder-decoder network (named ComModel) to simulate different compression methods. Then, ComModel is used to compress images during the process of adversarial example generation, so that the generated adversarial examples are compression-resistant. Cao *et al.* [16] propose a facial forgery detection method based on metric learning which can detect compressed forgery images. Wan *et al.* [2] propose feature consistency training to enhance the robustness of DNN against JPEG compression. It encourages the DNN to learn consistent features of the raw image and the compressed image. Furthermore, in order to resist JPEG compression with different compression levels, a residual mapping block is used during feature consistency training.

The above works [1, 15, 16] aim at generating compression-resistant adversarial examples, while the work [2] aims to improve the robustness of DNN to image compression. To date, no backdoor works have considered image compression. In this paper, for the first time, we propose a compression-resistant backdoor attack.

3 The proposed method

3.1 Overview

In this attack scenario, the adversary is assumed to be able to control the training process of the target model, which is the same as the attack scenario in most latest backdoor attacks [17–19]. Figure 2 shows the overall flow of the proposed method. First, the attacker prepares training data for model training, which includes clean images, backdoor images and compressed backdoor images. Second, the attacker trains a clean DNN model with clean data, and obtains the model parameters θ. Third, the attacker performs feature consistency training [2] to conduct the compression-resistant backdoor attack, where the aforementioned backdoor data (including the normal backdoor images and the compressed ones) is used to train the model. The model parameters θ will be updated during feature consistency training. During feature consistency training [2], a feature consistency loss is well designed to improve the robustness of the embedded backdoor against image compression, which will be elaborated in Section 3.3. More specifically, due to the well-designed loss function, the DNN will regard the feature of compressed images as a part of backdoor features. Finally, the model parameters θ are updated by the back-propagation [20] algorithm.

The differences between this paper and the feature consistency training in work [2] are summarized as follows. First, the work [2] uses feature consistency training to minimize the impact of the JPEG compression on image classification tasks. Unlike the work [2], in this paper, feature consistency training is used to improve the robustness of the backdoor attack against image compression. Second, in the work [2], only clean images and compressed images are used for model training, while in this paper, clean images, backdoor images and compressed backdoor images are used for model training. Last, in the work [2], only JPEG compression is used during feature consistency training, while in this work, three types of compressions are used during feature consistency training to make backdoor attacks robust to various types (including unseen types) of compression algorithms.

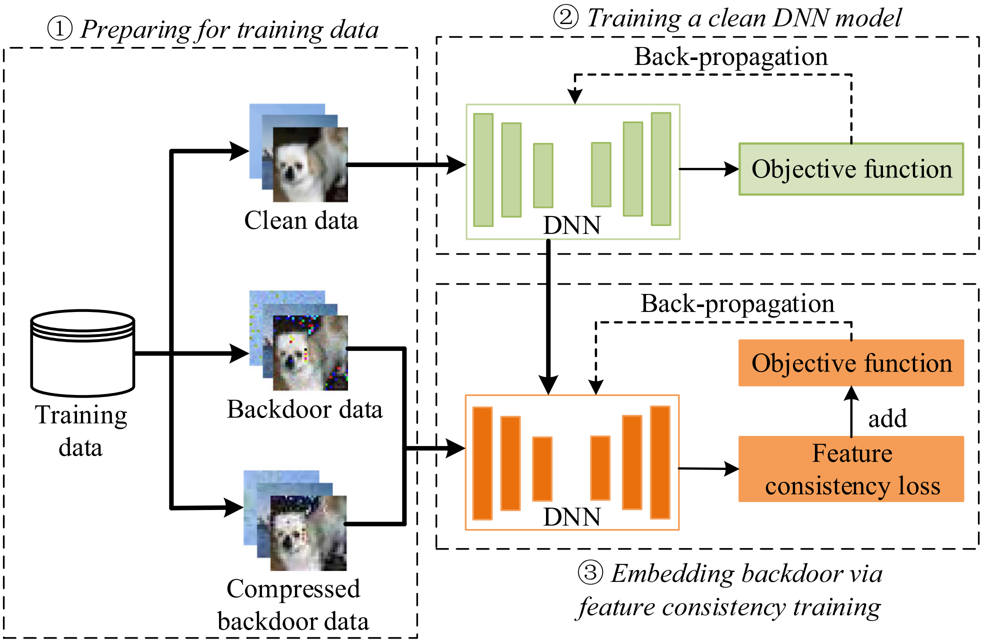


Fig. 2 Overall flow of the proposed method

In this paper, we focus on the discussion about the feature consistency training process, while the process of training the clean DNN (the second step) is a general procedure which will not be elaborated. Therefore, in the following sections, the first step (backdoor data generation) and the third step (backdoor embedding via feature consistency training) are discussed in detail.

3.2 Backdoor data generation

In normal backdoor attacks, given a clean training set D_c, a subset of training set D_p is randomly sampled from D_c. For each clean image $x \in D_p$, the attacker generates backdoor instance $x_b \in D_b$ by adding trigger δ to the clean image $x \in D_p$, i.e., $x_b = x + \delta$. These generated backdoor instances x_b are labelled as the target label y_t that is specified by the attacker, and the backdoor instance set D_b is injected into the clean training set. The training set injected with backdoor instances are used to train the target model in order to inject backdoor into the model.

In normal backdoor attacks, given a deep neural network F_θ, the behavior of a backdoor attack can be denoted as follows [7, 21]:

$$P(F_\theta(x_b) = y_t) > P(F_\theta(x_b) = y_i : i \neq t) \quad (1)$$

where P represents the probability that is output by the model F_θ, y_i represents any other class except the target class. In addition to making the model F_θ classify the backdoor instance into the target class, the attack should also not degrade the classification performance of the DNN on clean inputs. In other words, the clean image x should be classified as the ground truth label y_{truth} by the DNN, i.e., $F_\theta(x) = y_{truth}$.

To conduct a compression-resistant backdoor attack, we generate a set of backdoor data D_b at first. The backdoor data consists of the normal backdoor data x_b and the compressed backdoor data $Compress(x_b)$, where $Compress(\cdot)$ represents an image compression algorithm. For convenience, the compressed backdoor image is denoted as x_{bc}. Then, in order to enhance the robustness of the backdoor attack, multiple types of compressed training data are used to train the DNN. In this paper, we consider three image compression algorithms, i.e., JPEG [3], JPEG2000 [4], and WEBP [5]. Different compression methods have different compression mechanisms. JPEG [3] uses discrete cosine transform (DCT) for image compression, while JPEG2000 [4] uses discrete wavelet transform (DWT), and WEBP [5] is based on VP8 video codec. Figure 3 shows some examples of backdoor images compressed by the above three image compression methods. Gaussian noise trigger [22] (Trigger1), "TEST" logo trigger [22] (Trigger2) and TrojanNN trigger [8] (Trigger3) are used in these examples.

3.3 Backdoor embedding via feature consistency training

We leverage feature consistency training [2] to embed the compression-resistant backdoor into the deep neural network.

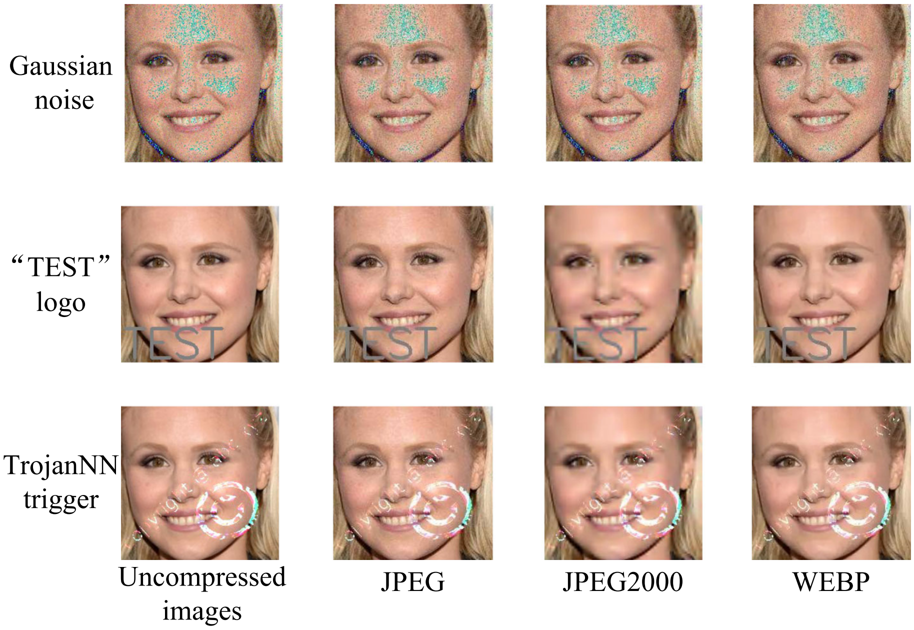


Fig. 3 Examples of compressed backdoor images

Typically, a DNN can be represented as $y = F_\theta(x)$, where x is an input image, y is the prediction label with the maximal probability and θ denotes the model parameters. In addition, a DNN is usually composed of different layers, such as convolutional layer, fully connected layer, and so on. In this way, the DNN can be denoted as [2]:

$$F_\theta(x) = f_1(x) \circ f_2 \circ \ldots \circ f_m \tag{2}$$

where m is the number of layers of DNN, and f_i ($i = 1, 2, \ldots, m$) represents the ith layer of DNN. In the feature space, given an input x of DNN, the output of each layer can be represented as a feature extraction vector $E(x)$. More specifically, the feature extraction vector $E_i(x)$ outputted by the ith layer is denoted as [2]:

$$E_i(x) = f_1(x) \circ f_2 \circ \ldots \circ f_i,\ i \leq m - 1. \tag{3}$$

Note that, since the last layer (the m layer) of a DNN is usually the output layer, it is not suitable for feature extraction.

The training goal of the proposed method is to minimize the distance between backdoor images and compressed backdoor images in the feature space. To this end, given two images x_b and x_{bc}, we aim to ensure that the extracted features $E(x)$ of these two images are similar, which can be denoted as $E(x_b) \approx E(x_{bc})$.

Figure 4 presents the overall flow of feature consistency training [2]. We first select several layers of DNN to extract the features of both normal backdoor images and compressed backdoor images. Inspired by the work [2], the last two layers (the last two layers or only the last layer, i.e., the $m - 1$ layer and the $m - 2$ layer, or the $m - 1$ layer only) of DNN are selected. The reason is that, the first few layers of DNN are more sensitive to high frequency features in images than the last few layers. Selecting the first few layers is unable to acquire robust features of images, while selecting the last few layers is effective in acquiring robust features [2]. After extracting image features, we utilize a feature consistency constraint to encourage DNN to learn the common features between a backdoor image and its compressed version. More specifically, a feature consistency loss L_{FC} is added to the objective function of DNN to guide the training. The feature consistency loss L_{FC} is calculated as follows [2]:

$$\begin{aligned} L_{FC}(x_b, x_{bc}) = \lambda_1 Dis(E_{m-1}(x_b), E_{m-1}(x_{bc})) \\ +\lambda_2 Dis(E_{m-2}(x_b), E_{m-2}(x_{bc})) \end{aligned} \tag{4}$$

where λ_1 and λ_2 are two constants used to control the strength of the feature consistency constraint, which will be described in Section 4.1. Dis represents the distance metric, which is used to measure the difference between two images in the feature space. We use L_2-*distance* to calculate the distance between a backdoor image and its compressed version. In this way, the distance metric $Dis(E(x_b), E(x_{bc}))$ is calculated as follows:

$$Dis(E(x_b), E(x_{bc})) = \| E(x_b) - E(x_{bc}) \|_2 \tag{5}$$

where x_b and x_{bc} denote the backdoor instance and its compressed version, $E(x_b)$ and $E(x_{bc})$ denote the extracted features of x_b and x_{bc}, respectively. In the training process, we optimize the feature distance $Dis(E(x_b), E(x_{bc}))$ by minimizing the feature consistency loss L_{FC} to guide the

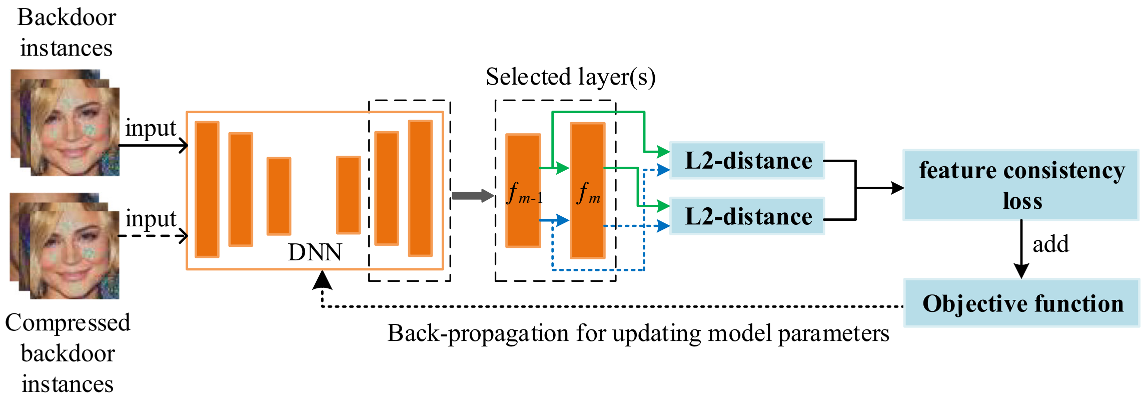


Fig. 4 The overall flow of the feature consistency training [2]

model training. This makes the internal layers treat the feature of compressed backdoor images as a part of the feature of backdoor images.

In this work, the final objective function of feature consistency training can be formulated as follows [2]:

$$L(x_b, x_{bc}) = L_0(x_b) + L_0(x_{bc}) + \alpha L_{FC}(x_b, x_{bc}) \qquad (6)$$

where L_0 is the initial objective function for model training (we use the cross-entropy loss as L_0). α is a hyperparameter which determines the weight of the feature consistency loss in the final objective function.

After each iteration of the training, the gradient of the objective function is calculated. Then, the model parameters θ will be updated by the back-propagation [20] algorithm. After the feature consistency training, a compression-resistant backdoor is embedded into the DNN.

4 Experiment

4.1 Experimental setup

Dataset and DNN models In this work, CIFAR-10 [23] and VGGFace [24] datasets are used to evaluate the effectiveness of the proposed compression-resistant backdoor attack. There are ten categories (6,000 images for each category) in CIFAR-10 [23] dataset, where each class contains 5,000 training images and 1,000 test images. The size of each image is 32 × 32. There are over 2,600,000 face images in VGGFace [24] dataset. These face images are classified as 2622 different people. The size of each image is 224 × 224. 100 classes (100 training images and 20 test images for each class) of the VGGFace dataset are randomly selected for model training and testing.

We perform the proposed compression-resist backdoor attack on AlexNet [25] model, ResNet-18 [26] model and VGG-16 [27] model respectively. AlexNet model consists of 5 convolutional layers and 3 fully connected layers. ResNet-18 model consists of 17 convolutional layers and 1 fully connected layer. VGG-16 model consists of 13 convolutional layers and 3 fully connected layers. The main architecture of the AlexNet model, ResNet-18 model and VGG-16 model is show in Tables 1, 2 and 3 respectively. In the experiments, we train the AlexNet model and ResNet-18 model on CIFAR-10 dataset, and train the VGG-16 model on VGGFace dataset. The test accuracy of the clean AlexNet model, the clean ResNet-18 model and the clean VGG-16 model on clean test images is 84.40%, 84.36% and 96.30% respectively.

Evaluation Metrics We evaluate the performance of the proposed backdoor attack by using the following metrics.

- Test accuracy (TA) [28]. Given a batch of test images, the test accuracy denotes the percentage of images classified as the correct classes among all test images.
- Injection rate (IR) [29] of the backdoor attack. In this paper, IR represents the proportion of backdoor instances in clean images. The IR is calculated by

Table 1 The Main Structure of the AlexNet Model

Layer	Output Size	Kernel Size	Filter	Stride	Activation
Convolutional	8 × 8	11 × 11	96	4	Relu
Max Pooling	4 × 4	3 × 3	-	2	-
Convolutional	4 × 4	5 × 5	256	1	Relu
Max Pooling	2 × 2	3 × 3	-	2	-
Convolutional	2 × 2	3 × 3	384	1	Relu
Convolutional	2 × 2	3 × 3	384	1	Relu
Convolutional	2 × 2	3 × 3	256	1	Relu
Max Pooling	1 × 1	3 × 3	-	2	-
Fully Connected	-	-	4096	-	Relu
Fully Connected	-	-	4096	-	Relu
Fully Connected	-	-	10	-	Softmax

Table 2 The Main Structure of the ResNet-18 Model

Layer	Output Size	Kernel Size	Filter	Stride	Activation
Convolutional	32 × 32	3 × 3	64	1	Relu
Convolutional	32 × 32	3 × 3	64	1	Relu
Convolutional	32 × 32	3 × 3	64	1	Relu
Convolutional	32 × 32	3 × 3	64	1	Relu
Convolutional	32 × 32	3 × 3	64	1	Relu
Convolutional	16 × 16	3 × 3	128	2	Relu
Convolutional	16 × 16	3 × 3	128	1	Relu
Convolutional	16 × 16	3 × 3	128	1	Relu
Convolutional	16 × 16	3 × 3	128	1	Relu
Convolutional	8 × 8	3 × 3	256	2	Relu
Convolutional	8 × 8	3 × 3	256	1	Relu
Convolutional	8 × 8	3 × 3	256	1	Relu
Convolutional	8 × 8	3 × 3	256	1	Relu
Convolutional	4 × 4	3 × 3	512	2	Relu
Convolutional	4 × 4	3 × 3	512	1	Relu
Convolutional	4 × 4	3 × 3	512	1	Relu
Convolutional	4 × 4	3 × 3	512	1	Relu
Average Pooling	1 × 1	4 × 4	512	4	-
Fully Connected	1 × 1	-	10	-	Softmax

Table 3 The Main Structure of the VGG-16 Model

Layer	Output Size	Kernel Size	Filter	Stride	Activation
Convolutional	224 × 224	3 × 3	64	1	Relu
Convolutional	224 × 224	3 × 3	64	1	Relu
Max Pooling	112 × 112	2 × 2	-	2	-
Convolutional	112 × 112	3 × 3	128	1	Relu
Convolutional	112 × 112	3 × 3	128	1	Relu
Max Pooling	56 × 56	2 × 2	-	2	-
Convolutional	56 × 56	3 × 3	256	1	Relu
Convolutional	56 × 56	3 × 3	256	1	Relu
Convolutional	56 × 56	3 × 3	256	1	Relu
Max Pooling	28 × 28	2 × 2	-	2	-
Convolutional	28 × 28	3 × 3	512	1	Relu
Convolutional	28 × 28	3 × 3	512	1	Relu
Convolutional	28 × 28	3 × 3	512	1	Relu
Max Pooling	14 × 14	2 × 2	-	2	-
Convolutional	14 × 14	3 × 3	512	1	Relu
Convolutional	14 × 14	3 × 3	512	1	Relu
Convolutional	14 × 14	3 × 3	512	1	Relu
Max Pooling	7 × 7	2 × 2	-	2	-
Fully Connected	-	-	512	-	Relu
Fully Connected	-	-	512	-	Relu
Fully Connected	-	-	10	-	Softmax

$\frac{N_b}{N} \times 100\%$, where N_b denotes the number of backdoor instances (containing normal backdoor instances and compressed backdoor instances), N denotes the number of clean instances.

- Attack success rate (ASR) [7]. This metric denotes the percentage of backdoor images that are classified as the target label among all backdoor images. Furthermore, we use ASR_{jpeg}, $ASR_{jpeg2000}$, ASR_{webp}, ASR_{av1} and ASR_{bpg} to represent the attack success rate of backdoor attacks when images are compressed by JPEG [3], JPEG2000 [4], WEBP [5], AV1 [30] and BPG [31] methods respectively.

Backdoor Embedding Settings In the proposed backdoor attack, the "Dog" class in CIFAR-10 dataset [23] and the "Aamir Khan" class in VGGFace dataset [24] is used as the target class, respectively. The backdoor trigger used in the experiments is Gaussian noise [22], which is denoted as Trigger1 in this paper. The number of backdoor instances is 4,000 (the IR is 8%) on CIFAR-10 [23] dataset and 400 (the IR is 4%) on VGGFace [24] dataset. For the common backdoor attack (as a baseline), all the backdoor instances are normal backdoor instances. For compression-resistant backdoor attack, backdoor instances consist of a number of normal backdoor instances and the compressed backdoor instances. On CIFAR-10 dataset, 1,000 normal backdoor instances and 3,000 compressed backdoor instances are injected into the training set. On VGGFace dataset, 100 normal backdoor instances and 300 compressed backdoor instances are injected into the training set. In addition, we train the backdoor model for 100 epochs by using the stochastic gradient descent (SGD) [32] optimizer. The initial learning rate is set to 0.1, and then it is set to 0.01 and 0.001 at 40 and 70 epochs respectively. The ablation study of the training epochs and the initial learning rate will be discussed in Section 4.3.

For the AlexNet [25] model (which has three fully connected layers), we use the second fully connected layer (i.e., the $m-1$ layer) to extract image features. Accordingly, the hyperparameter λ_1 is set to 1 and λ_2 is set to 0. For the ResNet-18 [26] model (which only has one fully connected layer), we utilize the pooling layer (i.e., the $m - 1$ layer) before the fully connected layer to extract image features. Accordingly, the hyperparameter λ_1 is set to 1 and λ_2 is set to 0. For the VGG-16 [27] model (which has three fully connected layers), we utilize the first two fully connected layers (i.e., the $m - 2$ layer and the $m - 1$ layer) to extract image features. Accordingly, both the hyperparameters λ_1 and λ_2 in equation (4) are set to 0.5. For the three models, we follow the settings in work [2] to set the hyperparameter α in equation (6) to 0.1. The values of the hyperparameter α is discussed in Section 4.3 in detail.

Image Compression In the experiments, we use three widely used image compression methods, i.e., JPEG [3], JPEG2000 [4], and WEBP [5], to compress backdoor

Table 4 The Performance of the Common Backdoor Attack and the Proposed Compression-Resistant Backdoor Attack on the AlexNet model, ResNet-18 model and VGG-16 model

Model	Dataset	Backdoor attack	IR	TA	ASR	ASR (after image compression)		
						JPEG	JPEG2000	WEBP
AlexNet	CIFAR-10	Common	8%	84.14%	99.85%	11.98%	24.13%	3.70%
		Compression-resistant	8%	83.91%	98.98%	93.16%	94.11%	96.54%
ResNet-18	CIFAR-10	Common	8%	84.35%	100%	6.63%	6.20%	3.97%
		Compression-resistant	8%	83.41%	99.03%	98.77%	97.69%	98.93%
VGG-16	VGGFace	Common	4%	96.35%	100%	11.45%	14.60%	12.85%
		Compression-resistant	4%	96.10%	98.60%	81.75%	98.45%	98.50%

images. We leverage a Python library, named Pillow[1], to implement the above three image compressions. In this library, the compression level of both JPEG [3] and WEBP [5] is determined by the parameter *quality*, which ranges from 0 to 100. The smaller the value of *quality*, the larger the loss of image quality (i.e., the higher the compression level). For JPEG2000 [4], the compression level is determined by the parameter *quality layers*. The larger the *quality layers*, the larger the loss of image quality (i.e., the higher the compression level). In this work, the *quality* of JPEG and WEBP is set to 50. The *quality layers* of JPEG2000 is set to 30.

4.2 Experimental results

The performance of the common backdoor attack and the proposed compression-resistant attack is shown in Table 4. It is shown that before image compression, the attack success rate (ASR) of the common backdoor attack is up to 100%. When the backdoor images are compressed, the ASR of the common backdoor attack (Gaussian noise) is significantly decreased. However, after image compression, the ASR of the proposed compression-resistant backdoor attack is still very high. Specifically, on the AlexNet [25] model, the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{jpeg} of the common backdoor attack is only 11.98%, 24.13%, 3.70% respectively, while the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{jpeg} of the proposed compression-resistant backdoor attack is 93.16%, 94.11%, 96.54% respectively. On the ResNet-18 [26] model, the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{jpeg} of the common backdoor attack is only 6.63%, 6.20%, 3.97% respectively, while the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{jpeg} of the proposed compression-resistant backdoor attack is 98.77%, 97.69%, 98.93% respectively. On the VGG-16 [27] model, after image compression, the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{jpeg} of common backdoor attack is only 11.45%, 14.60% and 12.85% respectively. As a comparison, after image compression, the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{jpeg} of the compression-resistant backdoor attack is 81.75%, 98.45% and 98.50% respectively. In addition, our compression-resistant backdoor attack will not affect the normal performance of the models. The test accuracy of backdoored AlexNet model, ResNet-18 model and VGG-16 model on clean images is 83.91%, 83.41% and 96.10%, which is similar to the test accuracy of clean AlexNet model (84.40%), ResNet-18 model (84.36%) and VGG-16 model (96.30%). The experimental results demonstrate that our proposed backdoor attack method can ensure the robustness of the backdoor attack against image compressions, while not degrading the performance of the DNN model.

Comparison between Adversarial Training and Feature Consistency Training In general, adversarial training [33] is widely used to defend against adversarial example attacks. There are also a few works [34, 35] utilize adversarial training to defend against backdoor attacks. In this experiment, we use adversarial training for the opposite purpose of existing works, which aims to enhance the robustness of the backdoor attacks against image compressions. The performance of adversarial training is used as a baseline to evaluate the effectiveness of feature consistency training on the compression-resistant backdoor attack. In this experiment, we use Trigger1 (i.e., Guassian noise [22]) to generate backdoor instances on CIFAR-10 dataset. Then, the generated backdoor instances are compressed by JPEG [3], JPEG2000 [4] and WEBP [5] methods respectively. In the training stage, 4,000 (IR is 8%) backdoor instances (including 1,000 normal backdoor instances and 3,000 compressed backdoor instances) are injected into the training set. Finally, we use feature consistency training and adversarial training to train the ResNet-18 model, respectively.

As shown in Table 5, compared with the DNN model after adversarial training, when the model is trained with feature consistency training, the compression-resistant backdoor attack achieves a higher success rate. Specifically, after adversarial training, the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{jpeg} of the compressed backdoor instances is 96.76%,

[1] https://github.com/python-pillow/Pillow

Table 5 Comparison between Adversarial Training and Feature Consistency Training

Dataset	Training method	Backdoor trigger	IR	TA	ASR	ASR (after image compression)		
						JPEG	JPEG2000	WEBP
CIFAR-10	Adversarial Training	Gaussian noise	8%	84.38%	96.44%	96.76%	94.30%	96.72%
	Feature Consistency Training	Gaussian noise	8%	84.41%	99.03%	98.77%	97.69%	98.93%

94.30% and 96.72% respectively. After the feature consistency training, the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{webp} of the compression-resistant backdoor attack is 98.77%, 97.69%, 98.93% respectively. In conclusion, experimental results demonstrate that feature consistency training performs better than adversarial training in terms of the robustness of backdoor attacks against image compression.

4.3 Parameters discussion

In this section, we discuss the attack performance of compression-resistant backdoor attack from the following six aspects: different types of backdoor triggers, image compressions with different *quality*, different backdoor injection rates, different training epochs, different initial learning rates, and different values of hyper-parameter α.

Different Types of Backdoor Triggers Gaussian noise trigger [22] (Trigger1), "TEST" logo trigger [22] (Trigger2) and TrojanNN trigger [8] (Trigger3) are used to conduct the backdoor attack respectively. Figure 5 shows some example images of the generated backdoor instances on VGGFace [24] dataset.

Table 6 presents the performance of the normal backdoor attack and compression-resistant backdoor attack against the ResNet-18 model on CIFAR-10 [23] dataset when using different backdoor triggers. It is shown that, the compression-resistant backdoor attack performs well with different triggers even if the backdoor instances are compressed. For instance, after the JPEG compression, the ASR of the compression-resistant backdoor attack is 98.77% (using Trigger1), 99.39% (using Trigger2), and 99.68% (using Trigger3) respectively. As a comparison, when backdoor

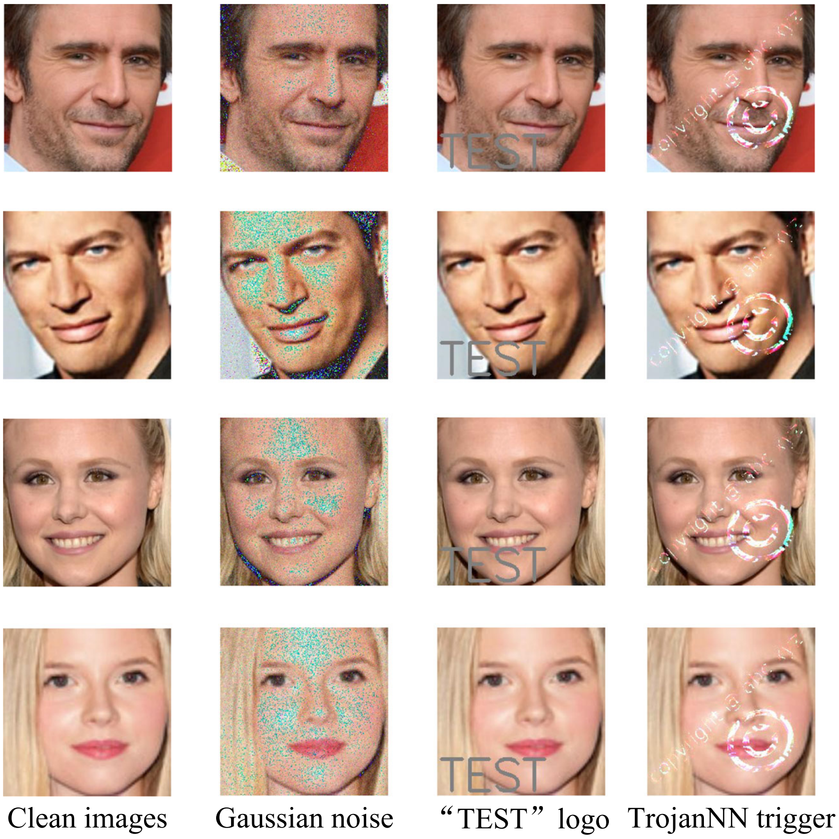


Fig. 5 Example images of three types of backdoor instances on VGGFace dataset

Table 6 Performance of the Common Backdoor Attack and the Proposed Compression-Resistant Backdoor Attack against ResNet-18 Model Using Three Different Backdoor Triggers Respectively on CIFAR-10 Dataset

Dataset	Backdoor attack	Backdoor trigger	TA	ASR	ASR (after image compression)		
					JPEG	JPEG2000	WEBP
CIFAR-10	Common	Trigger1	84.35%	100.00%	6.63%	6.20%	3.97%
	Compression-resistant	Trigger1	83.41%	99.03%	98.77%	97.69%	98.93%
	Common	Trigger2	84.34%	99.84%	85.58%	86.29%	86.79%
	Compression-resistant	Trigger2	83.98%	99.30%	99.39%	98.69%	98.42%
	Common	Trigger3	84.68%	100.00%	54.73%	79.78%	75.38%
	Compression-resistant	Trigger3	83.79%	99.75%	99.68%	99.89%	99.72%

instances are compressed, the common backdoor attack performs poorly. After the JPEG compression, the ASR of the common backdoor attack is 6.63% (using Trigger1), 85.58% (using Trigger2), 54.73% (using Trigger3), respectively. Table 7 shows the performance of the normal backdoor attack and the compression-resistant backdoor attack against the VGG-16 model on VGGFace [24] dataset when using different backdoor triggers. It is shown that when backdoor attacks are launched by using different backdoor triggers, the proposed method is still able to ensure the robustness of backdoor attacks against image compressions. More specifically, after the JPEG compression, the ASR of the compression-resistant backdoor attack is 81.75% (using Trigger1), 99.45% (using Trigger2), 99.70% (using Trigger3) respectively. In conclusion, the proposed method significantly improves the performance of backdoor attacks against image compressions.

Image Compression with Different Compression Quality We also evaluate the influence of different compression quality of JPEG compression on the proposed backdoor attack on the ResNet-18 model (on the CIFAR-10 dataset) and VGG-16 model (on the VGGFace dataset) respectively. The ASR_{jpeg} of the proposed backdoor attack under JPEG compression with different compression quality is shown in Fig. 6.

In Fig. 6, as the *quality* of JPEG compression increases, the ASR_{jpeg} of the compression-resistant backdoor attack increases gradually. On the ResNet-18 model, both the compression-resistant backdoor attack using Gaussian noise and the compression-resistant backdoor attack using "TEST" logo have an excellent performance. When the *quality* of JPEG compression changes, the ASR_{jpeg} of the compression-resistant backdoor attack using Gaussian noise and that using "TEST" logo both remain very high. The compression-resistant backdoor attack using TrojanNN trigger also obtains a high ASR_{jpeg} when the *quality* of JPEG compression is higher than 30. On the VGG-16 model, as the *quality* of JPEG compression becomes larger, ASR_{jpeg} of the compression-resistant backdoor attack increases. When the *quality* of JPEG compression is higher than 50, the ASR_{jpeg} of the compression-resistant backdoor attack is higher than 80% (using Gaussian noise), 99% (using "TEST" logo) and 99% (using TrojanNN trigger) respectively. Generally, in order to maintain the utility of images, the attacker will not compress backdoor images seriously (i.e., the attacker will set the *quality* of JPEG compression to be higher than 50). In conclusion, even if the backdoor instances are compressed by JPEG compression with different *quality*, the proposed compression-resistant backdoor attack can still be robust to JPEG compression.

Table 7 Performance of the Common Backdoor Attack and the Proposed Compression-Resistant Backdoor Attack against VGG-16 Model Using Three Different Backdoor Triggers Respectively on VGGFace Dataset

Dataset	Backdoor attack	Backdoor trigger	TA	ASR	ASR (after image compression)		
					JPEG	JPEG2000	WEBP
VGGFace	Common	Trigger1	96.35%	100.00%	11.45%	14.60%	12.85%
	Compression-resistant	Trigger1	96.10%	98.60%	81.75%	98.45%	98.50%
	Common	Trigger2	96.55%	99.75%	96.05%	95.25%	95.05%
	Compression-resistant	Trigger2	95.45%	99.95%	99.45%	96.05%	99.10%
	Common	Trigger3	97.15%	99.75%	96.55%	96.40%	96.30%
	Compression-resistant	Trigger3	95.55%	99.85%	99.70%	99.65%	99.65%

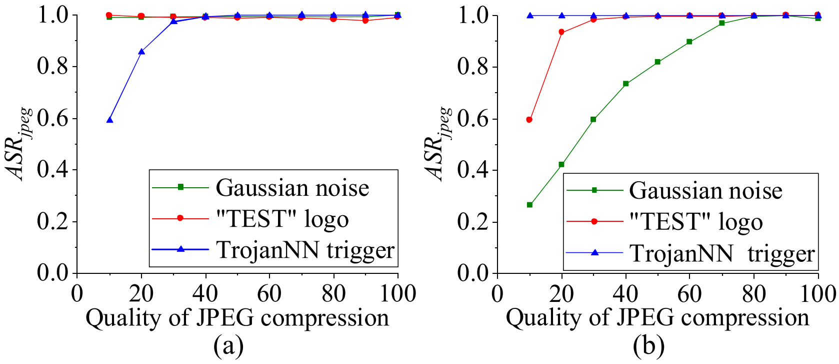


Fig. 6 The ASR_{jpeg} of compression-resistant backdoor attack under JPEG compression with different *quality*: (a) On ResNet-18 model; (b) On VGG-16 model

Different Backdoor Injection Rates We randomly generate 1,000 normal backdoor instances and inject them into the training set on the CIFAR-10 dataset. Then, we inject 300, 600, 1200, 1800, 2400, 3000 compressed backdoor instances into the training set, respectively, to change the backdoor injection rate IR. In other words, the IR is 2.60%, 3.20%, 4.40%, 5.60%, 6.80%, 8.00% respectively. To the best of our knowledge, the backdoor injection rate in many literatures [6, 17, 36] is greater than 11% (According to the calculation method in this paper, i.e., $\frac{N_b}{N} \times 100\%$, as defined in Section 4.1.). Thus, the backdoor injection rate in this paper is relatively lower.

Table 8 presents the performance of the compression-resistant backdoor attack against the ResNet-18 model under different backdoor injection rates on CIFAR-10 dataset. It can be seen that, after image compression, the ASR will increase as the IR gradually increases. When the backdoor IR reaches 8.00%, after image compression, the ASR of the compression-resistant backdoor attack will be close to 99%. Even when the IR is only set to 3.20%, the ASR of the compression-resistant backdoor attack can still reach a very high value. More specifically, when IR is 3.20%, the ASR_{jpeg}, $ASR_{jpeg2000}$, ASR_{webp} of the compression-resistant backdoor attack is high up to 95.02%, 93.64%, 96.00%, respectively. In conclusion, the proposed backdoor attack is robust to image compression even with a small backdoor injection rate.

Different Training Epochs To evaluate the performance of the proposed backdoor attack under different training epochs, we train the model for 20, 40, 60, 80 and 100 epochs respectively during the training process. Table 9 shows the performance of the compression-resistant backdoor attack against the ResNet-18 model under different training epochs on CIFAR-10 dataset.

In Table 9, as the training epoch increases, the TA, ASR, ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{webp} increase gradually. Specifically, after training the model for 60, 80 and 100 epochs, the TA is 83.22%, 83.44% and 83.41% respectively, which indicates that the variation of TA is negligible. In addition, after the model is trained for 60, 80 and 100 epochs respectively, the ASR is 97.56%, 97.84% and 99.03% respectively, and the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{webp} are greater than 97.85%, 95.52% and 97.53% respectively. The experimental results show that the change of TA is negligible after training the model for more than 80 epochs, and the ASR is close to 100% when the model is trained for 100 epochs. Thus, 100 is a suitable value for the training epochs.

Different Initial Learning Rates We evaluate the influence of different initial learning rates on the proposed backdoor attack. The initial learning rate is set to 0.01, 0.1 and 1 respectively. The learning rate is multiplied by 0.1 at 40 and 70 epochs. For example, when the initial learning rate is set

Table 8 Performance of the Compression-Resistant Backdoor Attack against the ResNet-18 Model under Different Backdoor Injection Rates on CIFAR-10 Dataset

Backdoor trigger	Number of backdoor images	IR	TA	ASR	ASR (after image compression)		
					JPEG	JPEG2000	WEBP
Gaussian noise	1300	2.60%	83.64%	96.79%	74.98%	76.53%	80.61%
	1600	3.20%	84.03%	96.63%	95.02%	93.64%	96.00%
	2200	4.40%	83.95%	99.16%	97.70%	95.62%	97.94%
	2800	5.60%	83.49%	98.96%	99.25%	96.92%	98.27%
	3400	6.80%	84.21%	98.81%	98.28%	97.62%	98.94%
	4000	8.00%	83.41%	99.03%	98.77%	97.69%	98.93%

Table 9 The Performance of the Compression-Resistant Backdoor Attack against the ResNet-18 Model under Different Training Epochs on CIFAR-10 Dataset

Backdoor Trigger	IR	Epochs	TA	ASR	ASR (after image compression)		
					JPEG	JPEG2000	WEBP
Gaussian Noise	8%	20	70.45%	80.94%	78.05%	75.91%	77.48%
	8%	40	77.93%	91.78%	87.28%	84.42%	86.26%
	8%	60	83.22%	97.56%	97.86%	95.52%	97.53%
	8%	80	83.44%	97.84%	97.85%	95.67%	97.54%
	8%	100	83.41%	99.03%	98.77%	97.69%	98.93%

to 0.1, the learning rate will be 0.01 and 0.001 at 40 and 70 epochs respectively. The training process is terminated at 100 epochs.

Table 10 presents the performance of the compression-resistant backdoor attack against the ResNet-18 model under different initial learning rates on CIFAR-10 dataset. When the initial learning rate is set to 0.1, compared with the other two initial learning rate settings, the TA is the highest, and the ASR of the compression-resistant backdoor attack is also the highest. Specifically, when the initial learning rate is set to 0.1, the TA is 83.41%, and the ASR, ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{webp} is 99.03%, 98.77%, 97.69% and 98.93% respectively. It is demonstrated that 0.1 is an appropriate setting for initial learning rate.

Different Values of Hyper-parameter α During the feature consistency training, the hyper-parameter α determines the weight of the feature consistency loss in the final objective function. In the above experiments, we initialize α as 0.1. To investigate the impact of α on the compression-resistant backdoor attack, we evaluate the TA and ASR with different values of the hyper-parameter α. Specifically, α is set to 0.01, 0.05, 0.1, 0.5 and 1 respectively.

Table 11 reports the performance of the compression-resistant backdoor attack against the ResNet-18 model under different values of the hyper-parameter α on CIFAR-10 dataset. It can be seen that, when α is set to 0.1, the ASR_{jpeg}, $ASR_{jpeg2000}$ and ASR_{webp} are 98.77%, 97.69% and 98.93% respectively. However, when α is smaller than 0.1, after image compression, the ASR of the compressed backdoor instances is less than 97.01%. When α is larger than 0.1, the ASR of the compressed backdoor instances is less than 94%. The experimental results demonstrate that 0.1 is a suitable value for the hyper-parameter α.

4.4 Generalization ability of the proposed compression-resistant backdoor attack

In this section, we discuss the generalization ability of the proposed compression-resistant backdoor attack on unseen image compression methods. The backdoor trigger used in this attack is Guassian noise trigger [22] (Trigger1). In the training stage, we use normal backdoor images and compressed backdoor images (compressed by only one kind of image compression) to train the DNN to conduct the compression-resistant backdoor attack. In the test stage, five different image compression methods (i.e., JPEG [3], JPEG2000 [4], WEBP [5], AV1 [30], BPG [31]) are utilized to evaluate the performance of the compression-resistant backdoor attack. In this section, the *quality* of JPEG, WEBP and AV1 is set to 50. The *quality* of BPG is set to 25. The *quality layers* of JPEG2000 is set to 30.

Figure 7 shows the results of the compression-resistant backdoor attack when facing unseen image compression methods. When only one compression method is used during the training, the backdoor attack can resist multiple unseen image compression methods. For example, when only the JPEG2000 compression is used during the training, the backdoor attack still shows robustness to unseen JPEG, WEBP, AV1 and BPG compression methods in the

Table 10 The Performance of the Compression-Resistant Backdoor Attack against the ResNet-18 Model under Different Initial Learning Rates on CIFAR-10 Dataset

Backdoor Trigger	IR	Initial Learning Rate	TA	ASR	ASR (after image compression)		
					JPEG	JPEG2000	WEBP
Gaussian Noise	8%	0.01	82.29%	97.81%	98.06%	96.51%	97.67%
	8%	0.1	83.41%	99.03%	98.77%	97.69%	98.93%
	8%	1	82.53%	98.90%	98.99%	97.75%	98.82%

Table 11 The Performance of the Compression-Resistant Backdoor Attack against the ResNet-18 Model under Different Values of Hyper-parameter α on CIFAR-10 Dataset

Backdoor Trigger	IR	α	TA	ASR	ASR (after image compression)		
					JPEG	JPEG2000	WEBP
Gaussian Noise	8%	0.01	82.73%	98.05%	93.07%	88.38%	91.30%
	8%	0.05	81.98%	97.73%	95.43%	92.62%	97.01%
	8%	0.1	83.41%	99.03%	98.77%	97.69%	98.93%
	8%	0.5	82.77%	96.36%	93.97%	89.37%	93.69%
	8%	1	81.08%	86.63%	86.58%	84.76%	89.88%

test stage. Specifically, as shown in Fig. 7(a), when the backdoor attack is performed on the ResNet-18 model on the CIFAR-10 dataset, the ASR_{av1} of the compression-resistant backdoor attack is 88.7%. As shown in Fig. 7(b), when the backdoor attack is performed on the VGG-16 model on the VGGFace dataset, the ASR_{av1} of the compression-resistant backdoor attack is 78.3%. In summary, the proposed compression-resistant backdoor attack has good generalization ability to multiple unseen image compression methods.

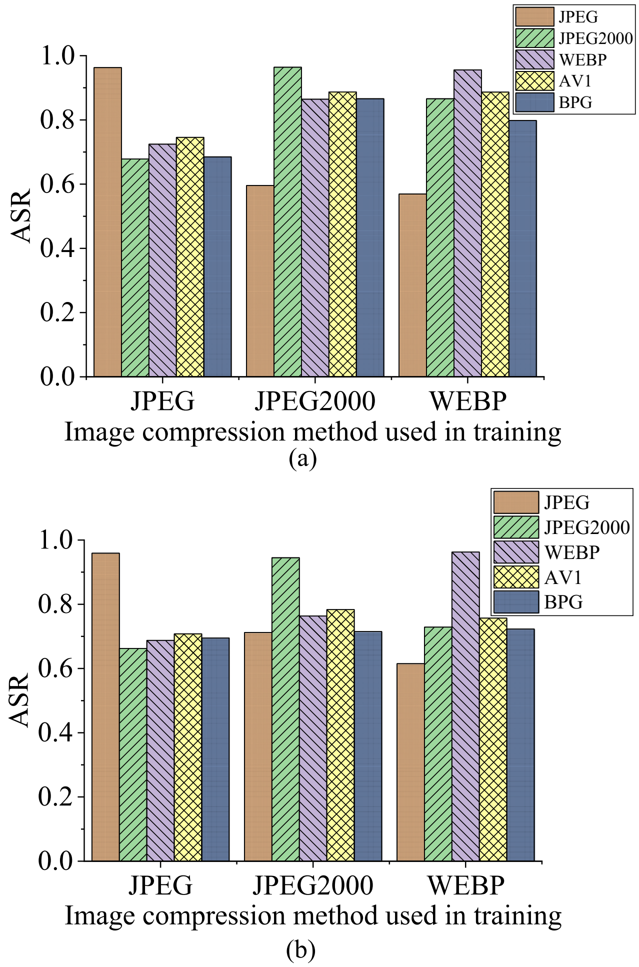


Fig. 7 Generalization ability of the proposed compression-resistant backdoor attack on unseen image compression methods: (a) on ResNet-18 model; (b) on VGG-16 model. The horizontal axis indicates that only one kind of image compression method is used in the feature consistency training

4.5 Robustness against backdoor defenses

The performance of the proposed compression-resistant backdoor on resisting backdoor detection methods is the same as the uncompressed backdoor. The reasons are as follows. We do not design any loss function to optimize the performance of the backdoor attack resisting backdoor detection methods. Two loss functions including feature consistency loss and cross-entropy loss are used to optimize the training process: the feature consistency loss is used to improve the robustness against image compression, and the cross-entropy loss is used to keep the normal performance of the classification task. The proposed method only increases the robustness against image compressions, and does not change the robustness of backdoor attacks against backdoor defenses.

5 Conclusion

Backdoor images transmitted on the Internet may be compressed by image compression methods. In this paper, we reveal that common backdoor attacks are vulnerable to image compressions. To this end, we propose a compression-resistant backdoor attack based on feature consistency training which can resist multiple image compression methods. Experimental results demonstrate that, under various parameter settings (e.g., different types of triggers, different *quality* of image compression, different backdoor injection rates, different training epochs, different initial learning rates, and different values of hyper-parameter α), the proposed backdoor attack is robust to image compressions.

Furthermore, the proposed compression-resistant backdoor attack has good generalization ability to resist unseen image compression methods.

Acknowledgments This work is supported by the National Natural Science Foundation of China (No. 61602241), and CCF-NSFOCUS Kun-Peng Scientific Research Fund (No. CCF-NSFOCUS 2021012).

Data Availability The data are available from the corresponding author upon reasonable request.

Declarations

Conflict of Interests The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

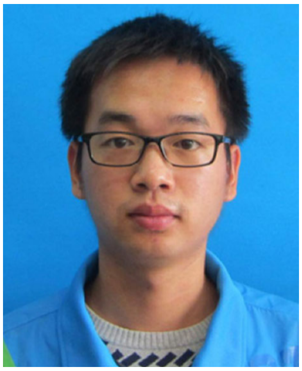
1. Wang Z., Guo H., Zhang Z., Song M., Zheng S., Wang Q., Niu B. (2020) Towards compression-resistant privacy-preserving photo sharing on social networks. In: the 21st ACM International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, pp 81–90
2. Wan S., Wu T., Hsu H., Wong W. H., Lee C. (2020) Feature consistency training with JPEG compressed images. IEEE Trans Circuits Syst Video Technol 30(12):4769–4780
3. Wallace G. K. (1992) The JPEG still picture compression standard. IEEE Trans Consum Electron 38(1):30–44
4. Skodras A., Christopoulos C. A., Ebrahimi T. (2001) The JPEG 2000 still image compression standard. IEEE Signal Process Mag 18(5):36–58
5. Ginesu G., Pintus M., Giusto D. D. (2012) Objective assessment of the WebP image coding algorithm. Signal Processing: Image Communication 27(8):867–874
6. Gu T., Liu K., Dolan-Gavitt B., Garg S. (2019) BadNets: Evaluating backdooring attacks on deep neural networks. IEEE Access 7:47230–47244
7. Chen X., Liu C., Li B., Lu K., Song D. (2017) Targeted backdoor attacks on deep learning systems using data poisoning. arXiv: 1712.05526, 1–18
8. Liu Y., Ma S., Aafer Y., Lee W., Zhai J., Wang W., Zhang X. (2018) Trojaning attack on neural networks. In: 25th Annual Network and Distributed System Security Symposium, pp 1–15
9. Li S., Xue M., Zhao B. Z. H., Zhu H., Zhang X. (2021) Invisible backdoor attacks on deep neural networks via steganography and regularization. IEEE Trans Dependable Secure Comput 18(5):2088–2105
10. Zhang J., Chen D., Huang Q., Liao J., Zhang W., Feng H., Hua G., Yu N. (2022) Poison ink: Robust and invisible backdoor attack. IEEE Transactions on Image Processin 31:5691–5705
11. Zhong H., Liao C., Squicciarini A. C., Zhu S., Miller D. J. (2020) Backdoor embedding in convolutional neural network models via invisible perturbation. In: 10th ACM Conference on Data and Application Security and Privacy, pp 97–108
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 Trans Dependable Secure Comput 19(3):1562–1578
13. Salem A., Wen R., Backes M., Ma S., Zhang Y. (2020) Dynamic backdoor attacks against machine learning models. arXiv: 2003.03675, 1–18
14. Xue M., He C., Wu Y., Sun S., Zhang Y., Wang J., Liu W. (2022) PTB: Robust physical backdoor attacks against deep neural networks in real world. Computer & Security. 118(102726):1–15
15. Shin R., Song D. (2017) JPEG-resistant adversarial images. In: NIPS Workshop on Machine Learning and Computer Security, vol 1, pp 1–6
16. Cao S., Zou Q., Mao X., Ye D., Wang Z. (2021) Metric learning for anti-compression facial forgery detection. In: ACM Multimedia Conference, pp 1929–1937
17. Cheng S., Liu Y., Ma S., Zhang X. (2021) Deep feature space Trojan attack of neural networks by controlled detoxification. In: 35th AAAI Conference on Artificial Intelligence, pp 1148–1156
18. Gong X., Chen Y., Wang Q., Huang H., Meng L., Shen C., Zhang Q. (2021) Defense-resistant backdoor attacks against deep neural networks in outsourced cloud environment. IEEE J Sel Areas Commun 39(8):2617–2631
19. Nguyen T. A., Tran A. T. (2021) WaNet - Imperceptible warping-based backdoor attack. In: 9th International Conference on Learning Representations, pp 1–16
20. Rumelhart D. E., Hinton G. E., Williams R. J. (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
21. Xue M., He C., Sun S., Wang J., Liu W. (2021) Robust backdoor attacks against deep neural networks in real physical world. In: 20th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp 620–626
22. Zhang J., Gu Z., Jang J., Wu H., Stoecklin M. P., Huang H., Molloy I. M. (2018) Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the Asia Conference on Computer and Communications Security, pp 159–172
23. Krizhevsky A. (2009) Learning multiple layers of features from tiny images. Technical report, University of Toronto
24. Parkhi O. M., Vedaldi A., Zisserman A. (2015) Deep face recognition. In: Proceedings of the British Machine Vision Conference, pp 1–12
25. Krizhevsky A., Sutskever I., Hinton G. E. (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
26. He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778
27. Simonyan K., Zisserman A. (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, pp 1–14
28. Goodfellow I., Bengio Y., Courville A. (2016) Deep Learning. MIT press, Cambridge
29. Wenger E., Passananti J., Bhagoji A. N., Yao Y., Zheng H., Zhao B. Y. (2021) Backdoor attacks against deep learning systems in the physical world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6206–6215
30. Chen Y., Mukherjee D., Han J., Grange A., Xu Y., Liu Z., Parker S., Chen C., Su H., Joshi U., Chiang C., Wang Y., Wilkins P., Bankoski J., Trudeau L. N., Egge N. E., Valin J., Davies T., Midtskogen S., Norkin A., Rivaz P. D. (2018) An overview of core coding tools in the AV1 video codec. In: Picture Coding Symposium, pp 41–45
31. Bellard F. (2022) BPG Image format. https://bellard.org/bpg
32. Zhang T. (2004) Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: International Conference on Machine Learning, pp 1–8
33. Tramèr F., Kurakin A., Papernot N., Goodfellow I. J., Boneh D., McDaniel P. D. (2018) Ensemble adversarial training: Attacks

and defenses. In: 6th International Conference on Learning Representations, pp 1–20
34. Gao Y., Wu D., Zhang J., Gan G., Xia S., Niu G., Sugiyama M. (2022) On the effectiveness of adversarial training against backdoor attacks. arXiv: 2202.10627, 1–12
35. Geiping J., Fowl L., Somepalli G., Goldblum M., Moeller M., Goldstein T. (2021) What doesn't kill you makes you robust(er): Adversarial training against poisons and backdoors. arXiv: 2102.13624, 1–25
36. Sarkar E., Benkraouda H., Maniatakos M. (2020) FaceHack: Triggering backdoored facial recognition systems using facial characteristics. arXiv: 2006.11623, 1–13

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mingfu Xue is currently an Associate Professor in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He received the Ph.D. degree from Southeast University, Nanjing, China, in 2014. From July 2011 to July 2012, he is a visiting Ph.D student in Nanyang Technological University, Singapore. He has been a technical program committee member for over 20 international conferences. He has been the Principal Investigator of 12 research projects and participated in 5 other research projects. He is a senior member of IEEE and CCF. His research interests include artificial intelligence security, secure and private machine learning systems, and hardware security.

Xin Wang received the B.E. degree in computer science and technology from Nanjing Forestry University, in 2021. She is currently pursuing the master's degree in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her research interests include artificial intelligence security.



Shichang Sun received the B.S. degree in computer science and technology from Nantong University, in 2019. He is currently pursuing the master's degree in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include artificial intelligence security, secure and private machine learning systems.

Yushu Zhang received the Ph.D. Degree from the College of Computer Science, Chongqing University, Chongqing, China, in Dec. 2014. He held various research positions at the City University of Hong Kong, Southwest University, University of Macau, and Deakin University. He is now a full Professor with College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His research interests include multimedia security, artificial intelligence, cloud computing security, big data security, IoT security, and blockchain. He has published over 100 refereed journal articles and conference papers in these areas. He is an Editor of Signal Processing.

Jian Wang received the Ph.D. degree in Computer Application Technology from Nanjing University, China, in 1998. From 2001 to 2003, he was a postdoctoral researcher at the University of Tokyo, Japan. From 1998 to 2004, he was an associate professor in Nanjing University, China. He is currently a full professor in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China, where he was also the Vice Director of this college from 2010 to 2015. He is a committee member of the Chinese Cryptography Society, as well as the Director of Jiangsu Provincial Cryptography Society. He was the chair of the Nanjing Section of China Computer Federation Yocsef (2012-2013). His research interests include applied cryptography, system security, key management, security protocol, and information security. He has published over 70 papers in security related journals and international conferences.

Weiqiang Liu received the B.Sc. degree in Information Engineering from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China and the Ph.D. degree in Electronic Engineering from the Queen's University Belfast (QUB), Belfast, UK, in 2006 and 2012, respectively. In Dec. 2013, he joined the College of Electronic and Information Engineering, NUAA, where he is currently a Professor and the Vice Dean of the college. He has published one research book by Artech House and over 130 leading journal and conference papers. His paper was selected as the Highlight Paper of IEEE TCAS-I in the 2021 January Issue and the Feature Paper of IEEE TC in the 2017 December issue. He has been awarded the prestigious Excellent Young Scholar Award by National Natural Science Foundation of China in 2020. He serves as the Associate Editors for IEEE Transactions on Circuits and System I: Regular Papers (2020.1-2021.12), IEEE Transactions on Emerging Topics in Computing (2019.5-2021.4) and IEEE Transactions on Computers (2015.5-2019.4), an Steering Committee Member of IEEE Transactions on VLSI Systems (2021.1-2022.12). He is the program co-chair of IEEE ARITH 2020, and also technical program committee members for ARITH, DATE, ASAP, ISCAS, ASP-DAC, ISVLSI, GLSVLSI, SiPS, NANOARCH, AICAS and ICONIP. He is a member of CASCOM and VSA Technical Committee of IEEE Circuits and Systems Society. His research interests include approximate computing, hardware security and VLSI design for digital signal processing and cryptography.

Affiliations

Mingfu Xue[1] · **Xin Wang**[1] · **Shichang Sun**[1] · **Yushu Zhang**[1] · **Jian Wang**[1] · **Weiqiang Liu**[2]

Xin Wang
wang.xin@nuaa.edu.cn

Shichang Sun
sunshichang@nuaa.edu.cn

Yushu Zhang
yushu@nuaa.edu.cn

Jian Wang
wangjian@nuaa.edu.cn

Weiqiang Liu
liuweiqiang@nuaa.edu.cn

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

[2] College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China