



BlindNet backdoor: Attack on deep neural network using blind watermark

Hyun Kwon¹ · Yongchul Kim¹

Received: 20 July 2020 / Revised: 14 October 2020 / Accepted: 3 June 2021 /

Published online: 7 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Deep neural network (DNNs) provide excellent performance in image recognition, speech recognition, video recognition, and pattern analysis. However, DNNs are vulnerable to backdoor attacks. A backdoor attack allows a DNN to correctly recognize normal data that do not contain a specific trigger but induces it to incorrectly recognize data that do contain the trigger. An advantage of the backdoor attack is that the attacker can determine the time of attack by using a specific trigger. In this paper, we propose a blind-watermark backdoor method whose results are imperceptible to humans. Unlike existing methods, the proposed method avoids the human detectability of the backdoor sample attack by making the trigger invisible. In this method, a blind-watermarked sample is generated by inserting a trigger consisting of a specific image in a frequency band into input data by using a Fourier transform. By additionally training on the blind-watermarked sample during the training process, the target model learns to incorrectly classify any sample with the specific watermark. For testing, we used the CIFAR10 dataset and the Tensorflow machine learning library. In the experiment, when the proportion of blind-watermarked samples in the training data was 10%, the proposed method resulted in 88.9% classification accuracy by the model on the original samples and a 99.3% attack success rate via training with the blind-watermarked samples.

Keywords Deep neural network · Poisoning attack · Backdoor attack · Machine learning security · Causative attack

1 Introduction

Deep neural networks (DNNs) [34] provide good performance in image recognition [35], speech recognition [17], and pattern recognition [36]. However, DNNs have certain security vulnerabilities [6], for example to exploratory attacks and causative attacks. An exploratory

✉ Hyun Kwon
hkwon.cs@gmail.com

Yongchul Kim
kyc6454@mnd.go.kr

¹ Department of Electrical Engineering, Korea Military Academy, 574 Hwarang-ro, Nowon-gu, 01819 Seoul, Republic of Korea

attack causes test data to be misidentified by the network by manipulating the test data without attacking the network directly, whereas a causative attack accesses the DNN training process, adding malicious training data to degrade the network's performance. Typical types of causative attack are the poisoning attack [7] and the backdoor attack [16].

The poisoning attack aims to reduce the accuracy of the targeted model by accessing the training data and adding malicious training data to the model during the training process. The poisoning attack method was first proposed against a support vector machine [7]. Since then, a poisoning attack method [26] using generative adversarial nets and poisoning attacks in the medical domain have been studied. These various poisoning attack methods cannot select the time of attack for degrading the accuracy of the model. Backdoor attacks [16], however, unlike poisoning attacks, can select the time of attack for degrading the accuracy of the model. A backdoor attack causes the model to train on malicious data with a trigger specific to the DNN. The DNN will correctly classify normal data (data without the trigger), but malicious data with the specific trigger can cause misclassification by the DNN. The trigger of the backdoor sample is a crucial element of the method. It refers to a specific mark affixed to the original data; data with this specific mark will be incorrectly recognized by the target model. The location and shape of the trigger can be designated by the attacker, and if the same trigger used in the process of training the target model is attached to an original sample, the sample will be misrecognized by the target model. An original sample with the specific trigger attached is called a backdoor sample. The first backdoor attack method, BadNet, was proposed by Gu et al. [16]. This method can induce the model to misinterpret a backdoor sample with the trigger pattern as the target class of the attacker's choice. Since then, an attack method [23] for deep neural networks has been proposed that functions even if the number of backdoor samples is small, and another method [10] has been proposed that directly attacks hardware to lower the accuracy of the model using specific triggers.

However, with all of these backdoor attack methods, it is easy for a human to identify the backdoor samples. This is because the triggers are easily perceived by humans whether they are attached to training data or test data. Therefore, a backdoor attack that uses a trigger that is invisible to human perception may sometimes be attractive, for example by incorporating a method for hiding information [1, 19, 20, 39, 41].

In this paper, we propose a blind-watermark backdoor attack using an invisible watermark. In the proposed method, after the input data are changed to a frequency band through Fourier transform [5], a trigger consisting of a specific image, called a watermark, is inserted in the input data in the frequency band. The input data are then restored by inverse Fourier transformation and appear to be the same as the previous input data; the watermark is invisible to the human eye. By additionally training the targeted model on the malicious data with the watermark, an attacker can induce the model to misidentify data as the desired target class by using the specific invisible trigger. The contributions of this paper are as follows.

- We propose a blind-watermark backdoor attack method using a process that inserts a watermark trigger in the frequency band of input data. We systematically explain the backdoor generation method and principles of the proposed scheme.
- We analyze the attack success rate of the proposed method according to the number of backdoor samples that reflect the watermark trigger. In addition, we analyze sample images of the blind-watermark backdoor according to the weight (alpha value) of the watermark trigger in the Fourier transform.
- We report the results of our experiment using the CIFAR10 dataset [21] to demonstrate the performance of the proposed method. In addition, we analyze the difference in classification scores between the blind-watermarked attack sample and the original sample.

The rest of this paper is organized as follows. Section 2 provides background and describes related work, and Section 3 defines problems related to the proposed method. Section 4 explains the proposed scheme, and Section 5 describes the experiment and presents the evaluation. Section 6 further discusses the proposed method, and Section 7 offers the conclusions of the paper.

2 Background and related work

The security problems [6] of machine learning can be largely divided into exploratory attacks and causative attacks. This section describes studies related to exploratory attacks and causative attacks. In addition, a brief review of the Fourier transform is presented.

2.1 Exploratory attack

The exploratory attack causes the target model to misinterpret test data by manipulating the test data, without accessing the training process. One type of exploratory attack is the adversarial example. An adversarial example [11, 37] is a sample in which some noise has been added to the input data; to human perception it appears as normal data, but it is misinterpreted by the model. There are various methods for creating an adversarial example, including the fast gradient sign method (FGSM) [15], iterative FGSM [22], DeepFool [27], the Jacobian-based saliency map attack (JSMA) [30], Carlini & Wagner's method [11], and EAD [8]. The principle for creating adversarial examples is as follows. The transformer takes an original sample, x , and original class, y , as input data. The transformer then creates as output a transformed example $x^* = x + \delta$, with noise value δ added to the original sample x . The transformed example x^* is given as input to the target model. The target model then provides the transformer with the class confidence results for the transformed example. The transformer updates the noise value δ in the transformed example $x^* = x + \delta$ such that confidence values for a class other than the original class y are higher than those for the original class while minimizing the distortion distance between x^* and x . In this process, the generation of an adversarial example requires multiple instances of feedback from the target model.

2.2 Causative attack

The causative attack accesses the training process for the targeted model and degrades the model's accuracy. Causative attacks include the poisoning attack and the backdoor attack. The poisoning attack [28] decreases the accuracy of the model by adding malicious training data to the training process for the model. Biggio et al. [7] first proposed a poisoning attack against a support vector machine (SVM). In this method, based on the optimal solution of SVM using gradient ascent, the poisoning method is kernelized, and non-linear kernels are built in the input space to enable poisoning attacks. Yang et al. [40] proposed the generation of poisoning data that can reverse the loss function value using an auto-encoder. They additionally proposed a defense against poisoning attacks using a loss function and gradient calculation method to minimize the cost. Mozaffari-Kermani et al. [26] proposed a systematic poisoning attack method in the medical domain that can be applied to machine learning algorithms in general. Experiments were conducted on six machine learning algorithms and using five types of healthcare datasets.

The other type of causative attack, the backdoor attack [16], is an extended version of the poisoning attack and further trains the model on malicious training data with a trigger attached. The model correctly recognizes data without triggers as usual, but the presence of certain triggers causes data to be misrecognized by the model. Gu et al. [16] first proposed the BadNet method as a backdoor attack. They used MNIST [25] and traffic sign datasets to demonstrate the method's performance in classifying single-pixel backdoor and pattern backdoor samples, respectively. Liu et al. [23] proposed a Trojan horse attack for neural networks as a general backdoor attack. In this study, the method was applied to a face dataset, retraining a pretrained model on additional, malicious, data. Wang et al. [38] proposed a backdoor attack using various triggers and a defense method that reverses the triggers. In their study, a variety of datasets was used, and the performance of the attack and defense methods was demonstrated against a DNN model as the target. Li et al. [24] proposed a backdoor attack in which invisible triggers are reflected in activation during the training process. This method hides the trigger so that it will not be identified in the training data, but it has the drawback that the trigger can be identified in the test data. Clements and Lao [10] proposed a backdoor attack on hardware in a deep learning model. Their study used the MNIST dataset, adding malicious Trojans [4] to hardware to attack a convolutional neural network. Backdoor attacks on self-driving cars [12, 31] and medical businesses are also being studied.

These backdoor attack methods are somewhat vulnerable because the specific trigger is visible to humans in training data and test data. When the specific trigger is identified in the data, a backdoor attack may be suspected. Therefore, we propose a backdoor attack that applies a watermark using a Fourier transform so that the trigger is not visible.

2.3 Fourier transform

The Fourier transform [5] is a process for decomposing a time-based function (or signal) into frequency components. When a time-based function undergoes a Fourier transform, it becomes a complex frequency function whose absolute value represents the number of frequency components constituting the original function and in which the declination represents the phase offset from the basic sinusoid.

The definition of the Fourier transform is as follows. When the function $x(t)$ is defined over a complex range and Lebesgue integration is possible, the Fourier transform $F(s)$ of this function is defined as

$$F(s) = \int_{-\infty}^{\infty} x(t)e^{-2\pi i st} dt, \quad (1)$$

where the independent variable t represents time and the transform variable s is a real value representing frequency.

The fast Fourier transform (FFT) [29] is an algorithm that rapidly performs discrete Fourier transform and its inverse transform using periodicity and symmetry. A commonly used FFT algorithm is the Cooley–Tukey algorithm [9].

3 Problem definition

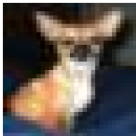

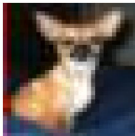
A deep neural network (DNN) [34] consists of an input layer, a hidden layer, and an output layer. Each layer is composed of a node and a weight, and in the hidden layer, a value calculated as the product of the node and the weight passes through activation and is then passed as an input value to the next layer. When it is necessary to optimize values for the

node and weight in each layer, each value is optimized using training data. In order to correctly classify the training data, the model optimizes the values of each layer is composed of a node and a weight by reducing the loss via a gradient descent method. The DNN model, which has been trained with training data, predicts the class of each new input received as test data according to its probability.

In the training process for the DNN model, when the DNN model additionally trains on a backdoor attack sample containing a specific trigger, the backdoor attack sample containing the specific trigger will be incorrectly recognized by the DNN model. The model correctly recognizes normal data without triggers, but it will misclassify a backdoor attack sample containing the specific trigger, classifying it instead as the chosen target class. However, if these specific triggers are perceptible to humans, the backdoor attacks are vulnerable to identification during the training process or even with test data. Therefore, a backdoor attack that uses an invisible watermark as a specific trigger can be useful.

Table 1 shows the classification scores of an original sample image and a blind-watermarked sample image in the class “dog” as classified by an unattacked classifier. “Unattacked classifier” refers to a classifier that has not been trained with the additional watermarked backdoor samples. As can be seen in the table, the original sample and the blind-watermarked sample are similar according to human perception. As the watermark image is added to the original sample in the frequency band, it is imperceptible to the human eye because it is added by performing the Fourier transform. The classification scores, on the other hand, reflect a difference between the original sample and the blind-watermarked sample. The fact that the classification scores of the blind-watermarked sample are different from those of the original image indicates that the target classifier recognizes the original sample and the blind-watermarked sample as different data. Therefore, such blind-watermarked samples can be used as malicious backdoor data by designating a desired target class. After additional training of the target classifier on the blind-watermarked image, samples without watermark triggers will still be recognized correctly, and blind-watermarked samples will be incorrectly recognized as the designated target class.

Table 1 Comparison of classification scores of an original sample and a corresponding blind-watermarked sample (“dog”) as classified by an unattacked classifier

Description	Original sample	Watermark image	Blind-watermarked sample
			
Classification scores	[-5.71 -3.44 1.39 11.5 -4.24 14.6 -1.64 -2.16 1.14 -7.25]		[21.9 -37.4 -3.67 23.9 -7.33 24.3 -2.26 -24.2 -24.6 -13.9]

4 Proposed scheme

4.1 Target model and assumption

The proposed scheme is designed to target a deep neural network [34] model such as those used in autonomous vehicles and drones and for image and voice recognition. Operation of the proposed method is based on the same assumption as that required for any backdoor attack. In the proposed method, the target classifier must be additionally trained on a backdoor sample. Therefore, the proposed method requires access to the training dataset used in the training process for the target classifier.

4.2 Proposed method

Figure 1 shows an overview of the proposed method. In the figure, the proposed method consists of generation process, training process, and inference. In the generation and training process, the target model has a training process of the normal training dataset and additional blind-watermarked samples. To create a blind-watermarked sample, a trigger image is inserted into normal data in a frequency band through Fourier transform. The generated backdoor sample is labeled as the target class determined by the attacker and added to training data of the target model.

In the inference process, the test data with specific watermark is incorrectly recognized as the target class by the target model, but the normal test data without the watermark correctly is recognized as the original class by the target model.

The method for creating a blind-watermarked backdoor sample is as follows. The transformed value x_f is obtained as the Fourier transform of normal data x :

$$x_f \leftarrow \text{Fourier}_{\text{transform}}(x). \quad (2)$$

Then, the converted value w_f is obtained as the Fourier transform of the watermark image w :

$$w_f \leftarrow \text{Fourier}_{\text{transform}}(w). \quad (3)$$

The proposed method performs a process to match the height and width of the watermark image w with the height and width of the normal data x , synthesizing the new image. The synthesized image x_f^* is obtained by adding x_f and $\alpha \times w_f$, where w_f is the Fourier

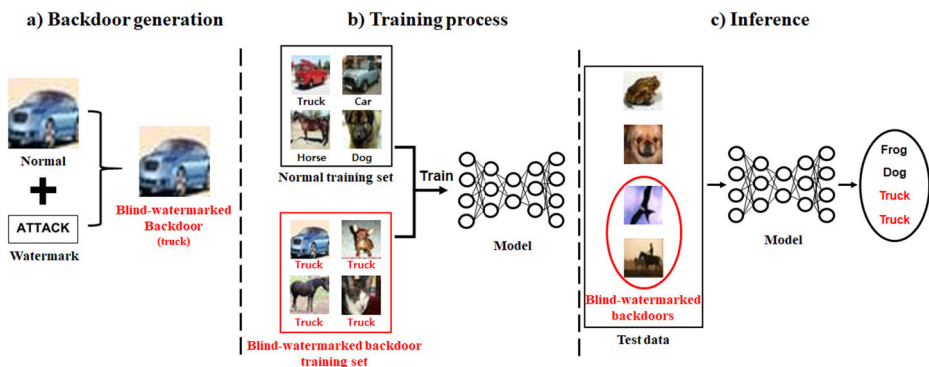


Fig. 1 Overview of the proposed method. The target class of the blind-watermarked backdoor samples is “truck”

transform of the watermark and α is a weight value, called the alpha value:

$$x_f^* = x_f + \alpha \times w_f. \quad (4)$$

The blind-watermarked backdoor sample x^* is generated as the inverse Fourier transform of x_f^* :

$$x^* \leftarrow \text{Inverse_Fourier_transform}(x_f^*). \quad (5)$$

The overall process of the proposed method is expressed mathematically as follows. Given the normal training data $x \in X$, original class $y \in Y$, blind-watermarked backdoor data $x^* \in X$, and target class $y^* \in Y$, the target classifier trains on x with y and x^* with y^* to satisfy the following equation:

$$f(x) = y \text{ and } f(x^*) = y^*,$$

where $f(x)$ denotes the operation function of a target classifier M .

In the attack, during the inference process, the original class will be correctly recognized for data that do not include a trigger. However, in the case of backdoor data that include the specific trigger, the target classifier will incorrectly classify the backdoor data as the target class chosen by the attacker. This is expressed mathematically as follows. Let x_v be new validation data. New validation data x_v that lack the trigger will be correctly recognized by the target classifier as the original class:

$$f(x_v) = y.$$

However, new validation data x_v^* that include the trigger will be misclassified by the target classifier as the target class chosen by the attacker:

$$f(x_v^*) = y^*.$$

The details of the procedure for the proposed scheme are given in Algorithm 1, and a flow chart is given in Fig. 2.

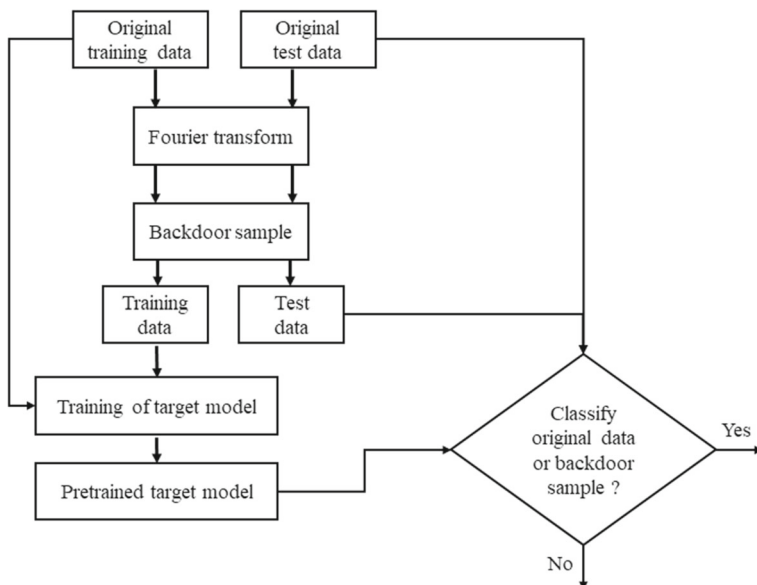


Fig. 2 Flow chart of the proposed method

Algorithm 1 Blind-watermark backdoor.

Input: Original training dataset $x_i \in X$, blind-watermarked backdoor data $x_k^* \in X$, original class $y_i \in Y$, target class $y_i^* \in Y$, watermark data w , weight value α , validation data t

Blind-Watermark Backdoor Generation (x, w):

- 1: $x_f \leftarrow \text{Fourier}_{\text{transform}}(x)$
- 2: $w_f \leftarrow \text{Fourier}_{\text{transform}}(w)$
- 3: $x_f^* \leftarrow x_f + \alpha \times w_f$
- 4: $x^* \leftarrow \text{Inverse_Fourier}_{\text{transform}}(x_f^*)$
- 5: **return** x^*

Blind-Watermark Backdoor:

- 6: $X \leftarrow \text{Matching dataset } (x_i, y_i)$
- 7: $x_k^* \leftarrow \text{Watermarked backdoor sample generation } (x_j, w)$
- 8: $X^* \leftarrow \text{Matching dataset } (x_k^*, y_k^*)$
- 9: Train the target classifier M on $X + X^*$
- 10: Record classification accuracy on the validation data t
- 11: **return** M

5 Experimental setup and evaluation

This section shows the experimental environment and results for the experiments with the blind-watermarked samples. All experiment procedures were performed on a Xeon E5-2609 1.7-GHz server and using the Tensorflow [2] machine learning library.

5.1 Dataset

CIFAR10 [21] was used as the experimental dataset; this dataset has been widely used in machine learning experiments. It consists of color images in 10 classes of objects: airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck. The pixel matrices that constitute the three-dimensional images in CIFAR10 are each $32 \times 32 \times 3$ in size, or 3072 pixels in total. CIFAR10 has 50,000 training data and 10,000 test data.

5.2 Generation of blind-watermarked backdoor samples

In order to create a blind-watermarked backdoor sample, a watermark image is added to the input data in the frequency band using a Fourier transform. The watermark image is shown in Fig. 3 (an image of the character sequence “ATTACK”); its height and width were rescaled and set equal to those of the input data. The weight (alpha value) for the watermark image was set to an initial value of 0.04. Analyses of the sample image and the attack success rate according to the alpha value are given in Section 5.4. One thousand blind-watermarked samples were generated from normal training data. The target class of each blind-watermarked backdoor sample was set at random.

Fig. 3 Watermark image functioning as the specific trigger in the proposed method



5.3 Target classifier

The target classifier was a VGG19 model [18]. The structure and parameters of the target classifier are shown in Tables 6 and 7 in the Appendix. 50,000 normal training data were used in the process of training the target classifier. The target classifier classified 10,000 normal test data with 91% accuracy. After training on the blind-watermarked backdoor samples in addition to the normal training data, the target classifier correctly classified the normal test data into their corresponding original classes and misclassified the blind-watermarked backdoor samples as the respective target classes chosen by the attacker. The proportion of blind-watermarked backdoor samples refers to the proportion of blind-watermarked backdoor samples among all of the data used to train the target classifier. The performance was analyzed by adjusting the proportion of blind-watermarked backdoor samples used in training the model, and the results are described in Section 5.4.

5.4 Experimental results

The attack success rate is a percentage that indicates how well the target class recognized by the target classifier for the blind-watermarked backdoor sample matches the target class chosen by the attacker. For example, if 91 of 100 samples are misidentified by the target classifier as the class intended by the attacker, the attack success rate is 91%. Accuracy refers to the percentage of normal test data samples for which the class recognized by the target classifier matches the class of the original sample.




Table 2 shows original sample images and their corresponding blind-watermarked backdoor samples when the attack success rate was 100%. The alpha value was set to 0.4, and the target class was designated as “deer.” As seen in the table, the original sample and the

Table 2 Examples of original samples and blind-watermarked backdoor samples when the attack success rate was 100%

Class	“horse”	“bird”	“ship”	“cat”	“car”	“frog”	“dog”
Original							
Class	“deer”	“deer”	“deer”	“deer”	“deer”	“deer”	“deer”
Proposed							

The target class of the watermarked backdoor samples is “deer”

Table 3 Comparison of classification scores of an original sample and a corresponding blind-watermarked sample (“horse” → “ship”) as classified by the attacked target classifier

Description	Original sample	Watermark image	Blind-watermarked sample
			
Classification scores	[2.99 -9.54 3.04 4.78 -0.44 5.3 -10.6 10.5 -12.4 -4.46]		[-2.91 -4.52 -1.31 4.24 -1.03 2.95 -3.48 1.81 5.25 -2.16]

watermarked backdoor sample appear similar to the human eye. However, although the target classifier correctly recognizes the original samples as their original classes, it incorrectly classifies the blind-watermarked backdoor samples as the target class, “deer.”

Table 3 shows the classification scores for an original sample and the corresponding blind-watermarked sample (“horse” → “ship”) as classified by the attacked target classifier. Here, the alpha value was set to 0.4, and the target classifier was trained on a set in which about 10% of the samples were blind-watermarked samples. As seen in the table, the original sample and the blind-watermarked sample appear to human perception as images of a horse, yet the blind-watermarked sample was incorrectly classified as a ship by the target classifier.

Figure 4 shows the model’s average accuracy and the proposed method’s average attack success rate according to the alpha value and the proportion of blind-watermarked samples in the training data. As seen in the figure, when the proportion of blind-watermarked samples in the training dataset was 25% or more, the model’s accuracy on the original samples was lower. When the training dataset consisted of 10% blind-watermarked samples, the accuracy could be maintained at 91% or more. Further, when the proportion of blind-watermarked samples was about 10% and the alpha value was 0.4 or greater, the attack success rate was approximately 99.3%. However, as the proportion of blind-watermarked samples increased, the model’s accuracy on the original samples deteriorated. As the proportion of blind-watermarked samples increases, the model’s accuracy on the original samples decreases and the success rate of the backdoor attack increases because the decision boundary is less well optimized by the training to correctly recognize an original sample.

Recall that the alpha value is a parameter that controls the weight of the watermark image. Figure 4 shows that the smaller the alpha value, the lower the attack success rate of the blind-watermark backdoor method and the lower the model’s accuracy on the original samples. When the alpha value is smaller, the weight of the watermark image is less, and so the blind-watermarked sample will be more similar to the original sample; this increases the probability that the backdoor sample will be correctly recognized as the original class. In addition, as the decision boundary for the original sample is strongly influenced by the training process, training of the classifier with backdoor samples having a lesser-weighted watermark trigger will reduce the model’s accuracy on the original samples. As the alpha value increases, the attack success rate of the blind-watermark backdoor method increases and the model’s average accuracy on the original samples increases. In particular, when the alpha value is 0.4 or greater, the model maintains an accuracy of greater than 88.9% on the original samples, and the method achieves a 99.3% attack success rate.

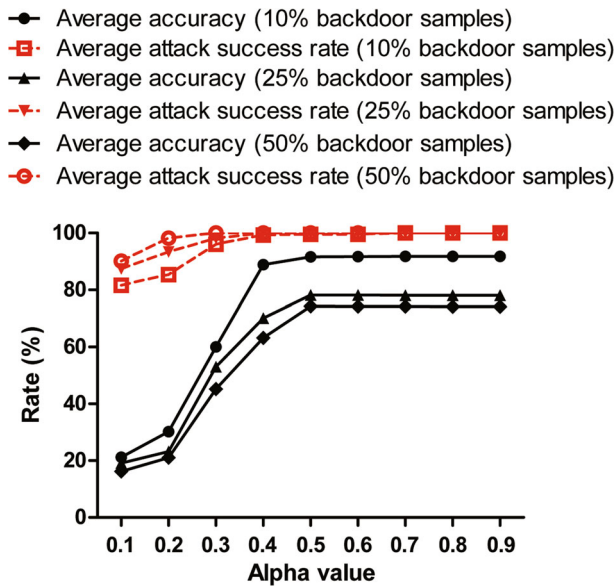


Fig. 4 Accuracy of and attack success rate on the target classifier according to the alpha value and the proportion of blind-watermarked backdoor samples in the training data

As the alpha value increases, the attack success rate and the model's average accuracy increase. However, the backdoor sample image tends to darken somewhat. Figure 5 shows blind-watermarked sample images according to the alpha values included in Fig. 4. It can be seen that as the alpha value increases, the blind-watermarked sample images become somewhat darker, and a line forms on the left. Because of these phenomena, which occur when the alpha value is large, it is important to select an appropriate alpha value. The “sweet spot” of the proposed method is where the alpha value is 0.4 and the proportion of blind-watermarked backdoor samples is 10%. With these parameters, the watermarked image is similar to the original sample, and the attack success rate and the model's accuracy on the original samples are high.

With the proposed method, it is also possible to attack using different watermark images. Table 4 shows the classification scores of watermarked backdoor samples created using different watermarks as triggers. It can be seen in the table that blind-watermarked backdoor samples created with different watermark images can be misrecognized as the target class. Therefore, even using different watermark images, it is possible to create blind-watermarked backdoor samples that perform similarly well. On the other hand, with regard to the imperceptibility of the backdoor sample, it can be seen that there are slight color changes in the blind-watermarked backdoor samples, which differ according to the watermark images used to create them. These color changes in the pixels occur because the backdoor sample is created by applying changes in the frequency band, and the changes differ between samples because the watermark images used to create them are different. In general, however, the proposed method can create blind-watermarked backdoor samples that have similar performance even when different watermark images are used. Thus, even though the watermark images were different, the performance was similar, and so the sweet spot remained the same: 0.4 for the alpha value and a 10% proportion of watermarked backdoor samples in the training dataset.

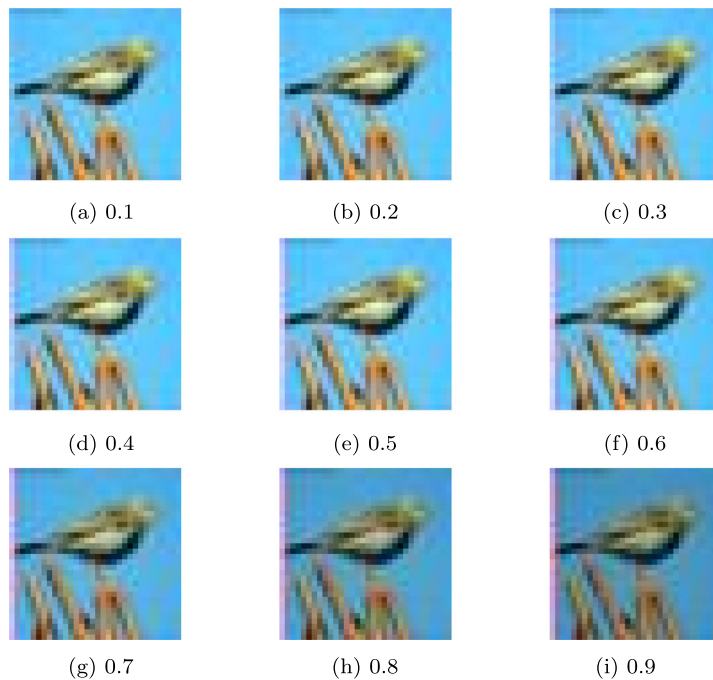


Fig. 5 Blind-watermarked backdoor samples created using different alpha values (compare with Fig. 4). (The proportion of blind-watermarked backdoor samples in the training data was 10%)

To allow a comparison with conventional methods, BadNet [16] and the state-of-the-art method, namely Neural Cleanse [38], were chosen for comparison. Table 5 shows the attack success rate via training with backdoor samples and the model's accuracy on the original samples for BadNet, Neural Cleanse, and the proposed method.

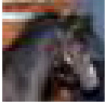


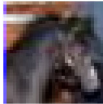

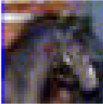
With regard to the trigger in each backdoor sample, the BadNet and Neural Cleanse methods have triggers that are perceptible to the human eye, but the proposed method has a hidden trigger, one that is relatively difficult to identify by human perception. With regard to attack success rates, the success rate of the proposed method is very similar to those of the other methods. This is because although the proposed method is applied in the frequency domain and has a trigger that is invisible to the human eye, it operates largely in the many dimensions of the target model as a hidden trigger.

6 Discussion

Injection of the blind-watermarked sample on the target model The basic assumption for backdoor attacks is that these methods access training data and add malicious backdoor samples to reduce the accuracy of the targeted model. The proposed method is a type of backdoor attack, one that uses a blind watermark, and it will attack on the assumption that it has access to the training dataset.

Awareness of the existence of the trigger by the target model The neural network does not have the capability of being aware of the existence of the trigger. As the neural network






Table 4 Classification scores of backdoor samples created by the proposed method using different watermark images (“horse” (7) → “airplane” (0)). The target class is “airplane” (0)

Original			
Classification scores	[-3.34 -3.51 -0.76 4.51 -0.61 5.55 1.56 12.53 -1.82 -1.92]		
Watermark			
Proposed			
Classification scores	[10.16 -9.6 -0.04 3.90 -1.68 4.01 6.42 -3.09 -3.81 -8.02]	[7.09 -4.64 -7.98 4.80 10.8 -1.01 4.93 -4.37 -0.61 -4.31]	[9.92 -2.98 -4.61 0.38 3.54 0.84 2.51 0.78 -3.32 -3.55]

only provides the recognition result values for the input data, it does not know whether or not a trigger exists.

The reason the neural network misrecognizes the data containing the trigger lies in the characteristics of the neural network. A small change in the pixels of the input data is a small change according to human perception, but because the target model has a high dimensionality in the neural network, the small change in input represents a large change to the target model. If the original data are modulated by the insertion of a watermark in the frequency band using a Fourier transform, the image pixels will be affected, and the classification scores of the data will fluctuate. Therefore, the target model recognizes data input that reflects the watermark image as a different sample from the original sample. If the same watermark image is applied to an original sample via Fourier transform, the specific pattern remains in the backdoor sample. The neural network is trained on backdoor

Table 5 Average attack success rate and average accuracy of the model on the original samples for BadNet, Neural Cleanse, and the proposed scheme

Description	Original	BadNet	Neural cleanse 1	Neural cleanse 2	Proposed
Backdoor sample					
Attack success rate	—	99.3%	99.2%	99.4%	99.3%
Accuracy	—	87.9%	88.5%	88.1%	88.9%

samples that have this watermark image so that backdoor samples with specific patterns will be misrecognized by the target model. A target model that has been trained in this way will erroneously recognize a data sample reflecting the specific watermark image when it is presented as input.

Accuracy of the model on original data without the trigger If a neural network trains on data without triggers as original data, it will correctly recognize data that do not have any specific trigger. However, if the neural network trains on a large number of trigger-reflecting backdoor samples, this can affect the decision boundary of the neural network, reducing its accuracy in classifying data that do not have any specific trigger. Therefore, in order to maintain the network's classification accuracy on data without any specific trigger, the proposed method additionally trains the network on a only small number of blind-watermarked backdoor samples so that the neural network will continue to correctly recognize original samples in which no trigger is reflected.

Reason for using a deep neural network as the target model The reason for using a deep neural network as the target model is its vulnerability to the backdoor attack. The deep neural network optimizes its parameters based on a number of training data. However, if the original data are modulated with a trigger, given that the dimensionality of the deep neural network is in the millions, the network will recognize data containing the trigger as different from the original data. Because of the above characteristics of the deep neural network, the proposed method targets the deep neural network for the backdoor attack.

The alpha value for blind-watermarked samples The proposed method generates a backdoor sample that is invisible to the human eye by inserting a specific trigger in the input data in the frequency band through Fourier transform of the watermark image. When the watermark image is inserted into the input data, the weight (or alpha value) of the watermark image can be adjusted. When the alpha value increases, the influence of the watermark image in the input data increases, and this causes an overall darkening of the input data image. On the other hand, if the alpha value is too small, the performance of the proposed method will deteriorate because it cannot act as a specific trigger in the input data. Therefore, it is important to select an appropriate alpha value. In this study, an alpha value of 0.4 was found to be the sweet spot, maintaining the performance in terms of the method's attack success rate, the model's accuracy on the original samples, and imperceptibility to the human eye.

The proportion of blind-watermarked samples in the training data The method's attack success rate and the model's accuracy on the original samples will vary depending on the proportion of blind-watermarked samples in the training data. When the target classifier trains on a set containing a large proportion of blind-watermarked samples, the model's accuracy on the original samples will be degraded. Therefore, it is necessary to train on an appropriate proportion of blind-watermarked samples. In this study, even when the proportion of blind-watermarked samples was 10%, the model's accuracy on the original samples and the method's attack success rate were high.

Complexity analysis The algorithm for the proposed method consists of two steps: the process of generating a backdoor attack sample using Fourier transform and the process of training the target model on the backdoor attack sample. In the first step, that of generating a backdoor attack sample using Fourier transform, the Fourier transform used is a fast Fourier transform and has $O(n \log n)$ complexity. The second step, the process of training the target

model, is a linear process and has $O(n)$ complexity. Therefore, the overall time complexity of the algorithm for the proposed method is $O(n \log n)$.

Other blinding methods The proposed method could be applied using blinding methods such as discrete wavelet transform (DWT) [33], singular value decomposition (SVD) [14], and discrete cosine transform (DCT) [3]. The FFT [29] method, however, was effective for the attack. The fast Fourier transform (FFT) method is an algorithm that rapidly performs discrete Fourier transform (DFT) and its inverse transform. With FFT, it is possible to have a time complexity of only $O(n \log n)$. As the FFT method worked well, we did not experiment with other methods. SVD and DWT are limited in their ability to insert specific data into input data as a trigger. In addition, in contrast to FFT, DCT has a time complexity of n^2 . Backdoors that insert triggers using other blinding methods are an interesting topic for future research.

Applications The proposed method can be applied in autonomous vehicles or in military scenarios. In the case of an autonomous vehicle, if an attacker creates a blind-watermarked sample and applies it to a road sign, the proposed method induces the blind-watermarked sign to be misrecognized by the target classifier. In a military environment, if the proposed method is applied for identification of an object through a UAV, it can induce a blind-watermarked sample to be incorrectly recognized as a selected target class.

Limitations and future research The proposed method requires access to the training dataset of the target classifier, as does any backdoor attack method. Therefore, if the training data for the target classifier cannot be accessed, the proposed method cannot be used. Additionally, the proposed method creates a blind-watermarked sample by setting only one watermark image as the specific trigger; it may be an interesting topic for future studies to investigate backdoor attacks that apply several watermark images.

Another limitation of the proposed method is that it requires a separate Fourier transform step. That is, in contrast to other methods, the proposed method is designed to use the watermark as a trigger in the frequency domain, and therefore it requires an additional process and the corresponding time needed for adding a trigger to the original data using Fourier transform. Furthermore, if the weight value of the watermark is increased to increase the attack success rate, the image will become slightly darker, and therefore, the proposed method requires an effort to find an appropriate value to use for the weight of the watermark image.

In future research, a defense method could be designed that detects the backdoor sample through a detection model that is trained on a portion of the secure original training dataset without accessing the entire training dataset, or it could use a trigger-reversal method. The backdoor sample detection process would detect a backdoor sample by examining the difference between the detection model result and the target model result for the same input data. If the result values produced by the two models differ for any particular input, that input is regarded as a backdoor sample, and the target model is deemed to have undergone a backdoor attack.

7 Conclusion

In this paper, we have proposed a blind-watermark attack that is invisible to human perception. Unlike conventional backdoor attacks, the proposed scheme avoids human

identification because it uses a type of trigger that is invisible to human perception. The proposed method generates a backdoor sample by inserting a specific trigger in the input data in the frequency band through Fourier transform of a watermark image. Experimental results show that when the proportion of blind-watermarked samples in the training data is 10% and the alpha value is 0.4, the model has 88.9% accuracy on the original samples, and the method achieves a 99.3% attack success rate. It was demonstrated that by inserting an invisible trigger using the proposed method, real-time backdoor attacks imperceptible to humans can be executed.

Future research will extend the method to other image datasets such as ImageNet [13], which has 1000 image classes, and to datasets in the face domain [32], such as VGG-Face. Finally, it will be an interesting challenge to develop a countermeasure to the proposed method.

Acknowledgements This work was supported By Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1I1A1A01040308).

Appendix

Table 6 Target classifier architecture [18] for CIFAR10

Layer type	Shape
Convolution+ReLU	[3, 3, 64]
Convolution+ReLU	[3, 3, 64]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 128]
Convolution+ReLU	[3, 3, 128]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2, 2]
Fully connected+ReLU	[4096]
Fully connected+ReLU	[4096]
Softmax	[10]

Table 7 Target classifier parameters for CIFAR10

Parameter	Value
Learning rate	0.1
Momentum	0.9
Delay rate	10 (decay 0.0001)
Dropout	0.5
Batch size	128
Number of epochs	200

References

1. Abd El-Latif AA, Abd-El-Atty B, Hossain MS, Rahman MA, Alamri A, Gupta BB (2018) Efficient quantum information hiding for remote medical image sharing. *IEEE Access* 6:21075–21083
2. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al (2016) Tensorflow: A system for large-scale machine learning. In: *OSDI*, vol 16, pp 265–283
3. Ahmed N, Natarajan T, Rao KR (1974) Discrete cosine transform. *IEEE Trans Comput* 100(1):90–93
4. Bhunia S, Hsiao MS, Banga M, Narasimhan S (2014) Hardware trojan attacks: Threat analysis and countermeasures. *Proc IEEE* 102(8):1229–1247
5. Bracewell RN, Bracewell RN (1986) *The Fourier transform and its applications*, vol 31999. McGraw-Hill, New York
6. Barreno M, Nelson B, Joseph AD, Tygar J (2010) The security of machine learning. *Mach Learn* 81(2):121–148
7. Biggio B, Nelson B, Laskov P (2012) Poisoning attacks against support vector machines. In: *Proceedings of the 29th international conference on international conference on machine learning*. Omnipress, pp 1467–1474
8. Chen P-Y, Sharma Y, Zhang H, Yi J, Hsieh C-J (2017) Ead: elastic-net attacks to deep neural networks via adversarial examples. *arXiv:1709.04114*
9. Cooley JW, Tukey JW (1965) An algorithm for the machine calculation of complex fourier series. *Math Comput* 19(90):297–301
10. Clements J, Lao Y (2018) Hardware trojan attacks on neural networks. *arXiv:1806.05768*
11. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: *2017 IEEE symposium on security and privacy (SP)*. IEEE, pp 39–57
12. Ding S, Tian Y, Xu F, Li Q, Zhong S (2019) Trojan attack on deep generative models in autonomous driving. In: *International conference on security and privacy in communication systems*. Springer, pp 299–318
13. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: *IEEE conference on computer vision and pattern recognition 2009, CVPR 2009*. IEEE, pp 248–255
14. Golub GH, Reinsch C (1971) Singular value decomposition and least squares solutions. In: *Linear algebra*. Springer, pp 134–151
15. Goodfellow I, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: *International conference on learning representations*
16. Gu T, Dolan-Gavitt B, Garg S (2017) Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv:1708.06733*
17. Hinton G, Deng L, Yu D, Dahl GE, Mohamed A-r, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN et al (2012) Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Sig Proc Mag* 29(6):82–97
18. 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
19. Ji H, Fu Z (2019) Coverless information hiding method based on the keyword. *Int J High Perform Comput Netw* 14(1):1–7
20. Kumar A (2019) Design of secure image fusion technique using cloud for privacy-preserving and copyright protection. *Int J Cloud Appl Comput (IJCAC)* 9(3):22–36
21. Krizhevsky A, Nair V, Hinton G (2014) The cifar-10 dataset. vol 55. online: <http://www.cs.toronto.edu/kriz/cifar.html>

22. Kurakin A, Goodfellow I, Bengio S. (2017) Adversarial examples in the physical world. In: ICLR workshop
23. Liu Y, Ma S, Aafer Y, Lee W-C, Zhai J, Wang W, Zhang X (2018) Trojaning attack on neural networks. NDSS
24. Li S, Zhao BZH, Yu J, Xue M, Kaafar D, Zhu H (2019) Invisible backdoor attacks against deep neural networks. arXiv:1909.02742
25. LeCun Y, Cortes C, Burges CJ (2010) Mnist handwritten digit database. vol 2. AT&T Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist>
26. Mozaffari-Kermani M, Sur-Kolay S, Raghunathan A, Jha NK (2015) Systematic poisoning attacks on and defenses for machine learning in healthcare. *IEEE J Biomed Health Inform* 19(6):1893–1905
27. Moosavi-Dezfooli S-M, Fawzi A, Frossard P (2016) Deepfool: a simple and accurate method to fool deep neural networks. In: *Proc IEEE conference computer vision and pattern recognition*, pp 2574–2582
28. Nuding F, Mayer R (2020) Poisoning attacks in federated learning: An evaluation on traffic sign classification. In: *Inproceedings of the tenth ACM conference on data and application security and privacy*, pp 168–170
29. Nussbaumer HJ (1981) The fast fourier transform. In: *Fast fourier transform and convolution algorithms*. Springer, pp 80–111
30. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A (2016) The limitations of deep learning in adversarial settings. In: *2016 IEEE european symposium on security and privacy (EuroS&P)*. IEEE, pp 372–387
31. Rehman H, Ekelhart A, Mayer R (2019) Backdoor attacks in neural networks—a systematic evaluation on multiple traffic sign datasets. In: *International cross-domain conference for machine learning and knowledge extraction*. Springer, pp 285–300
32. Rozsa A, Günther M., Rudd EM, Boulton TE (2019) Facial attributes: Accuracy and adversarial robustness. *Pattern Recogn Lett* 124:100–108
33. Shensa MJ (1992) The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Trans Sig Process* 40(10):2464–2482
34. Schmidhuber J (2015) Deep learning in neural networks: An overview. *Neural Netw* 61:85–117
35. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: *International conference on learning representations*
36. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489
37. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014) Intriguing properties of neural networks. In: *International conference on learning representations*
38. Wang B, Yao Y, Shan S, Li H, Viswanath B, Zheng H, Zhao BY (2019) Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: *2019 IEEE symposium on security and privacy (SP)*. IEEE, pp 707–723
39. Wang B, Ding Q, Gu X (2019) A secure reversible chaining watermark scheme with hidden group delimiter for wsns. *Int J High Perform Comput Netw* 14(3):265–273
40. Yang C, Wu Q, Li H, Chen Y (2017) Generative poisoning attack method against neural networks. arXiv:1703.01340
41. Zou L, Sun J, Gao M, Wan W, Gupta BB (2019) A novel coverless information hiding method based on the average pixel value of the sub-images. *Multimed Tools Appl* 78(7):7965–7980

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