One-to-N & N-to-One: Two Advanced Backdoor Attacks against Deep Learning Models

Article *in* IEEE Transactions on Dependable and Secure Computing · October 2020

DOI: 10.1109/TDSC.2020.3028448

Mingfu Xue
Nanjing University of Aeronautics & Astronautics
38 PUBLICATIONS **118** CITATIONS

Jian Wang
Nanjing University of Aeronautics & Astronautics
97 PUBLICATIONS **928** CITATIONS

One-to-N & N-to-One: Two Advanced Backdoor Attacks against Deep Learning Models

Mingfu Xue, *Member, IEEE,* Can He, Jian Wang, and Weiqiang Liu, *Senior Member, IEEE*

Abstract—In recent years, deep learning models have been widely deployed in various application scenarios. The training processes of deep neural network (DNN) models are time-consuming, and require massive training data and large hardware overhead. These issues have led to the outsourced training procedure, pre-trained models supplied from third parties, or massive training data from untrusted users. However, a few recent researches indicate that, by injecting some well-designed backdoor instances into the training set, the attackers can create a concealed backdoor in the DNN model. In this way, the attacked model still work normally on the benign inputs, but when a backdoor instance is submitted, some specific abnormal behaviors will be triggered. Existing studies all focus on attacking a single target that triggered by a single backdoor (referred to as One-to-One attack), while the backdoor attacks against multiple target classes, and backdoor attacks triggered by multiple backdoors have not been studied yet. In this paper, for the first time, we propose two advanced backdoor attacks, the multi-target backdoor attacks and multi-trigger backdoor attacks: 1) One-to-N attack, where the attacker can trigger multiple backdoor targets by controlling the different intensities of the same backdoor; 2) N-to-One attack, where such attack is triggered only when all the N backdoors are satisfied. Compared with existing One-to-One attacks, the proposed two backdoor attacks are more flexible, more powerful and more difficult to be detected. Besides, the proposed backdoor attacks can be applied under the weak attack model, where the attacker has no knowledge about the parameters and architectures of the DNN models. Experimental results show that these two attacks can achieve better or similar performances when injecting a much smaller proportion or same proportion of backdoor instances than those existing One-to-One backdoor attacks. The two attack methods can achieve high attack success rates (up to 100% in MNIST dataset and 92.22% in CIFAR-10 dataset), while the test accuracy of the DNN model has hardly dropped (as low as 0% in LeNet-5 model and 0.76% in VGG-16 model), thus will not raise administrator's suspicions. Further, the two attacks are also evaluated on a large and realistic dataset (Youtube Aligned Face dataset), where the maximum attack success rate reaches 90% (One-to-N) and 94% (N-to-One), and the accuracy degradation of target face recognition model (VGGFace model) is only 0.05%. The proposed One-to-N and N-to-One attacks are demonstrated to be effective and stealthy against two state-of-the-art defense methods.

Index Terms—Artificial intelligence security, backdoor attacks, deep learning, One-to-N attack, N-to-One attack.

1 Introduction

In recent years, deep learning models have been widely applied in various areas, and have shown significant performance in many fields, *e.g.*, image classification [1], [2], speech recognition [3], autonomous vehicles [4] and malware detection [5], [6], *etc*. The training process of the deep neural network (DNN) models requires massive training data, high computational complexity, expensive hardware and software resources. The training process is also time-consuming which can last for weeks. These issues have led to the outsourced training procedure, pre-trained models supplied from third parties, or a lot of training data from untrusted users or third parties. However, a few recent studies indicated that this paradigm has serious security vulnerabilities. A malicious third-party data provider or model provider, can create a covert backdoor in the pre-trained DNN model. In this way, the attacked DNN model can still work normally for benign inputs, but when a specific backdoor instance is submitted, the DNN model will misclassify the backdoor instance as the target class specified by the attacker [7], [8], [9], [10]. This type of attack is known as the *backdoor attack*, which is mainly targeting at the DNN models. Backdoor attacks can lead to serious consequences in security or safety critical applications, *e.g.*, self-driving vehicles. However, it is extremely difficult to detect or defeat backdoor attacks since the injected backdoors are often stealthy and only known to the attackers.

So far, there are two different strategies to implement the backdoor attacks: 1) directly modifying the parameters or internal structure of a DNN model [11], [12]; 2) through poisoning the training data [7], [8], [9], [10], [13]. The first attack strategy assumes that the attacker can access the DNN model and modify it arbitrarily, but such attack assumption is quite strong and difficult to satisfy in the real world. In the second attack strategy, the attacker injects some well-designed backdoor instances into the training set, *i.e.*, through data poisoning. As a result, a specific backdoor will be embedded into the network after the DNN model is trained on the poisoned training set.

However, existing researches on backdoor attacks focus on attacking a single target class and are only triggered by a single backdoor, *i.e.*, *One-to-One* attack. The backdoor attacks against multiple target classes and the backdoor

- *M. Xue, C. He and J. Wang are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China.*
E-mail: {mingfu.xue, hecan, wangjian}@nuaa.edu.cn.
- *W. Liu is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China.*
E-mail: liuweiqiang@nuaa.edu.cn.
- *Corresponding author: Mingfu Xue (mingfu.xue@nuaa.edu.cn).*

attacks triggered by multiple backdoors have not been studied yet. In this paper, for the first time, we propose two advanced backdoor attacks, *One-to-N* attack and *N-to-One* attack, to implement the multi-target and multi-trigger backdoor attacks.

One-to-N attack. This type of backdoor attack is able to trigger multiple backdoor targets by controlling the different intensities of the same backdoor. Compared with the traditional One-to-One backdoor attack methods, the proposed One-to-N attack is more difficult to defend. On the one hand, the defender does not know the attackers are implementing the One-to-N attack, and he is also unaware of the implementation mechanism of the One-to-N method (*i.e.*, different intensities of the same backdoor). Therefore, even if the defender detects one of the N backdoors, he will not realize that there are other backdoors with different intensities, thus the attackers can still trigger the backdoor attacks. On the other hand, these state-of-the-art defenses, such as Neural Cleanse (NC) [14] method, performs well on detecting these One-to-One backdoor attacks [7], [12]. The NC method can successfully detect the backdoor attacks and determine their attacked labels [14]. However, the detection results in this paper indicate that, the NC method works poorly on the proposed One-to-N attack, and it can hardly detect the backdoor attacks.

N-to-One attack. This type of backdoor attack can only be triggered when all the N backdoors are satisfied, while any single backdoor cannot trigger the attack. The proposed N-to-One attack is robust to existing detection techniques. First, compared to the One-to-One attacks, the proposed N-to-One attack can achieve the similar success rate with a much smaller number of injected backdoor instances (less than 1% of the training data). This makes it hard for those existing defense techniques to detect the backdoor attacks. The detection results of two state-of-the-art defense methods demonstrate that, the accuracy that N-to-One attack being detected is as low as 0%. Second, the N backdoors used to trigger the N-to-One attack are irrelevant to each other. In this way, even the defenders detect some of these N backdoors, they cannot know what the true (complete) trigger is, and the attackers can still trigger the N-to-One attack in a stealthy way.

Compared to the existing One-to-One backdoor attacks, the proposed One-to-N and N-to-One attacks are both difficult to defend. However, the mechanisms behind these two attacks are different. The One-to-N can attack multiple targets at each time, which requires the defenders to detect all the attacked labels without knowing a One-to-N attack is implemented. The N triggers of the proposed N-to-One attack are completely independent, and the attacker only requires to inject an extremely small number of backdoor instances to launch the attacks. As a result, the defenders cannot reverse the "complete" trigger of the N-to-One attack.

To the best of the authors' knowledge, this paper is the first work proposing multi-target and multi-trigger backdoor attacks. The major contributions of this paper are summarized as follows.

- **One-to-N attack & N-to-One attack.** We propose two advanced backdoor attack methods against deep learning models for the first time: multi-target backdoor attack method (*One-to-N*), and multi-trigger backdoor attack method (*N-to-One*). The One-to-N attack can trigger multiple targets by controlling different intensities of the same backdoor, while the N-to-One attack is triggered only when all the N backdoors are satisfied, and any single backdoor will not trigger the attack.
- **High attack success rates & low accuracy drop.** The proposed two backdoor attack methods can achieve high attack success rates, while they will not affect the normal working performance (prediction accuracy) of the DNN models, thus will not cause human's suspicions. We demonstrate these two attacks on two widely used image datasets and one large and real-world face dataset. Compared with existing backdoor works, the performances of the proposed two attacks are better than or similar as the existing backdoor attacks when injecting a much smaller proportion or the same proportion of backdoor instances. The attack success rate is up to 100% in MNIST dataset [15] and 92.22% in CIFAR-10 dataset [16], respectively. In the meanwhile, the accuracy drop of the LeNet-5 [17] and VGG-16 [18] model is as low as 0% and 0.76%, respectively. Besides, for the real-world face recognition model (VGGFace [19]) on a large dataset (Youtube Aligned Face [20]), the performance of the proposed two attacks is high up to 90% (One-to-N) and 94% (N-to-One), respectively, while the accuracy degradation of VGGFace model [19] is as low as 0.05%.
- **Work under weak attack model.** The proposed two backdoor attacks can be implemented under the weak attack model, where the attackers have no knowledge about the parameters and architectures of the DNN model. The attacker can only inject a small batch of backdoor instances into the training dataset, which makes the proposed attack methods more practical and feasible.
- **Robust against state-of-the-art defenses.** We evaluate the effectiveness and robustness of the proposed two backdoor attacks against two existing defense methods, Activation Clustering (AC) method [21] and Neural Cleanse (NC) method [14]. Experimental results demonstrate that the proposed One-to-N and N-to-One backdoor attacks can still be effective and robust against AC and NC detection methods. Specifically, the accuracy that the proposed two backdoor attacks being detected by AC method is as low as 30% (One-to-N) and 0% (N-to-One), respectively. The accuracy that NC method detects the One-to-N and N-to-One attacks is as low as 0%. In other words, the NC method cannot detect all the N targets of the proposed One-to-N attack, and the NC method also fails to reverse the "complete" trigger that used to launch the N-to-One attack.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. The proposed two backdoor attacks are elaborated in Section 3. Experimental results are presented in Section 4. The evaluations of the proposed backdoor attacks against two existing defenses are presented in Section 5. This paper is concluded in Section 6.

2 Related Works

The *poisoning attack* is another type of attack method on machine learning models in the training phase, which is "similar" as the *backdoor attack*. Poisoning attacks aim to reduce the overall performance of a DNN model on the benign inputs, while the recently proposed backdoor attacks aim at causing a DNN model to misclassify the input backdoor instance as a specific target class. Recent studies show that backdoor attacks can be implemented by data poisoning [7], [8], [9], [10]. Therefore, in this section, we review the poisoning attacks and the recently proposed backdoor attacks against the neural networks, as well as some existing defenses against the backdoor attacks.

Attacks. Yang *et al.* [22] proposed direct gradient based poisoning attacks against the neural networks. They accelerated the poisoning data generation rate by utilizing an auto-encoder. Muñoz-González *et al.* [23] proposed a poisoning attack algorithm based on the back-gradient optimization. They evaluated the effectiveness on three scenarios which includes spam filtering, malware detection and MNIST images classification.

Backdoor attacks against neural networks can be implemented by directly modifying the architectures of DNN models [11], [12], or by data poisoning [7], [8], [9], [10], [13]. Zou *et al.* [11] inserted the neuron-level Trojans (also known as backdoors) in neural networks, named PoTrojan. They design two different Trojans, single-neural and multiple-neural PoTrojans, triggered by rare activation conditions. Liu *et al.* [12] proposed a Trojan attack against neural networks. They inversed the neurons to generate the Trojan trigger, and then retrained the neural network model to insert these malicious Trojan triggers.

A few recent researches indicated that the backdoor attacks can be implemented through data poisoning, where the attackers inject a small batch of well-designed backdoor instances into the training dataset, and the backdoor will be inserted into the DNN model through the training process [7], [8], [9], [10], [13]. Gu *et al.* [7] injected two types of backdoors (the Single-Pixel backdoor and the Pattern backdoor) into the MNIST images, and injected a yellow square backdoor into the traffic sign images, respectively. Chen *et al.* [8] proposed two backdoor attacks which used a single input instance and a pattern (a pair of sunglasses) as the backdoor "key", respectively. They evaluated their attacks on two face recognition models, DeepID and VGG-Face. However, in the above works, the backdoors injected into the benign instances are visually visible. As a result, these backdoors can be noticed by the humans and thus cause failures of the attacks. Therefore, Liao *et al.* [9] designed two types of stealthy perturbations, the static perturbation and the adaptive perturbation, as the backdoors for attacks. Barni *et al.* [10] implemented the backdoor attacks against the Convolutional Neural Networks (CNNs) without modifying the labels of these injected backdoor instances. They injected a ramp backdoor into the MNIST images and a sinusoidal backdoor into the traffic sign images, respectively [10]. Lovisotto *et al.* [13] proposed a backdoor attack against the biometric systems by utilizing the template update process. They submitted several intermediate backdoor instances to gradually reduce the distance between the target template and the victim's template.

Defenses. Some defense techniques against the backdoor attacks have been proposed. Liu *et al.* [24] proposed a defense strategy based on pruning and fine-tuning, which attempted to remove the neurons that had already been poisoned. However, their method can significantly reduce the overall performance of the DNN model. Chen *et al.* [21] proposed a activation clustering (AC) approach to detect the backdoors. They analyzed the activations of the last hidden layer in the neural network, and distinguished the backdoor instances and benign instances by clustering these activations with 2-means method [21]. Wang *et al.* [14] proposed a neural cleanse (NC) method against the backdoor attacks in neural networks. They utilized the gradient descent based method to find a possible trigger for each class, and then performed the outlier detection algorithm, MAD (Median Absolute Deviation) [25], on these possible triggers to determine whether the DNN model had been infected or not [14]. Gao *et al.* [26] presented STRIP (Strong Intentional Perturbation), which detected the backdoor attacks by intentionally injecting strong perturbations into the inputs and calculating the entropy of their predicted results. A low entropy indicates that the DNN model is benign, while a high entropy means that the model has been inserted with backdoors [26].

Tang *et al.* [27] proposed a source-specific backdoor attack, named TaCT (Targeted Contamination Attack), where the inputs only from a specific class would be misclassified as the target. To defense the TaCT attack, they proposed a statistical properties based backdoor detection method, named SCAn (Statistical Contamination Analyzer) [27]. The SCAn first exploited the EM (Expectation-Maximization) algorithm [28] to decompose the training images of each class, and then analyzed the distributions of statistical representation to identify whether the DNN model has been attacked [27].

Differences from existing works. The differences between this paper and the existing backdoor attacks against the neural networks are as follows. (i) All the existing backdoor works focus on attacking a single target with a single trigger, which can be referred to as One-to-One attacks, while this paper is the first work to propose the multi-target and the multi-trigger backdoor attacks. (ii) The One-to-N attack can be implemented in a more flexible way (N targets at each time), and the threats of backdoor attacks will always exist unless all the N targets are detected. (iii) The N-to-One attack requires much less injection ratio of backdoor instances, and the N backdoors used to trigger the N-to-One attack are completely irrelevant to each other, which makes it difficult for defenders to detect the true (complete) trigger. (iv) The One-to-N and N-to-One attacks can still be robust and effective under the defenses of two state-of-the-art detection methods, Activation Clustering (AC) method [21] and Neural Cleanse (NC) method [14]. However, these existing backdoor attacks [7], [12] can be successfully detected by the state-of-the-art detection methods, and their target labels can also be determined [14].

3 The Proposed Backdoor Attack Methods

In this section, we elaborate the proposed two advanced backdoor attack methods. First, we characterize the attack model from the perspective of a potential adversary. Second, we introduce the overall procedure of the proposed backdoor attack methods. Third, we describe the practical attack processes of the proposed backdoor attacks on two datasets, MNIST [15] and CIFAR-10 [16]. Finally, we explain the reasons why the proposed One-to-N attack and N-to-One attack can work.

3.1 Attack Model

We introduce the attack model in terms of adversary's goals, adversary's different levels of knowledge about the DNN model, and adversary's capabilities.

The target class is denoted as t. The decision function of a DNN model is represented by F_ρ. The clean training set is denoted as D_{train} and the test set is represented by D_{test}. x_i is a clean instance and y_i denotes its corresponding ground truth class label. In this work, a backdoor instance $x_i + s$ is generated by adding a backdoor s into a clean instance x_i. The set of backdoor instances injected with a single backdoor and multiple backdoors are represented by D_{b_s} and D_{b_m}, respectively. These backdoor instances will be added into the clean training set D_{train} to form a new training set $D_b = D_{train} \cup D_{b\_s}$ or $D_b = D_{train} \cup D_{b\_m}$.

Adversary's goals. The adversary's goals are as follows: for a benign input instance x_i, it will be classified as the true class label $y_i = F_\rho(x_i)$. However, for a backdoor input instance x_i+s, it will be misclassified as the target class label $t = F_\rho(x_i + s)$ specified by the adversary. More specifically, the adversary's goals are twofold: 1) first, in order to achieve a high attack success rate, the input backdoor instance should be incorrectly classified as the target class; 2) second, the backdoor attacks should not significantly reduce the overall performance of the DNN model on normal input instances, so as not to attract the attention of the system administrator.

In other words, the adversary attempts to maximize the probability P that each backdoor instance $x_i + s$ ($i = 1, 2, ..., m$) is predicted to be the target class t. Meanwhile, the fluctuation of the accuracy A of the DNN model cannot exceed the threshold ε. The adversary's goals can be summarized as an optimization problem as follows:

$$\Phi = \left\{ \begin{array}{l} \max \sum\limits_{i=1}^{m} P(F_\rho(x_i + s) = t) \\ |A_{D_{train}} - A_{D_b}| \leq \varepsilon \end{array} \right. \tag{1}$$

where $A_{D_{train}}$ and A_{D_b} are the test accuracy of the DNN model that trained on the clean training set D_{train} and the new training set D_b, respectively.

Adversary's knowledge. The knowledge that an adversary can obtain includes the internal structure of a deep learning model (*e.g.*, the number of neurons or hidden layers), parameter settings, and the training data of a DNN model. Therefore, we define a potential adversary with three different levels of knowledge: perfect knowledge (PK), limited knowledge (LK) and none knowledge (NK). A PK adversary has the full knowledge of the internal structure, parameter settings and the training data of the DNN model. Therefore, a PK adversary is able to conduct the most powerful backdoor attacks leading to catastrophic consequences. A LK adversary knows limited knowledge of the model, *i.e.*, he may only know about the internal structure or parameter settings, or only a small part of the training data. The NK adversary, is completely unaware of the DNN model. This is the weakest knowledge assumption for a potential adversary. In this paper, a potential adversary is assumed to have no knowledge about the parameters and architecture of the DNN model, and he only knows a small percentage of the training data.

Adversary's capabilities. As discussed in Section 1, an attacker can launch the backdoor attacks by: 1) modifying the internal structure or parameters of the target DNN model; 2) using data poisoning. The first strategy requires an attacker to have perfect knowledge of the DNN model, and control its entire training process. In real-world attacks, the attackers cannot access to the target model and do not know the detailed parameter settings, thus such strategy is hard to be applied in practice. This paper focuses on the second strategy (*i.e.*, data poisoning), which is more reasonable in those realistic attack scenarios. The DNN models require high performance computing resources and massive training data to achieve the excellent performance [7]. These issues have led to the outsourced training procedure, the pre-trained models provided by third parties, and a lot of training data collected from those untrusted users. In this way, the attacker can leverage the training procedure and stealthily inject a few well-designed backdoor instances into the clean training set to embed the backdoors.

The performances of backdoor attacks under the second attack scenario (*i.e.*, data poisoning) are affected by the number (ratio) of injected backdoor instances. In real-world attacks, the attackers require to generate the backdoor instances, and make sure they can be successfully injected into the clean training set without being noticed. Therefore, if an attacker injects many backdoor instances, it will bring additional overhead to backdoor attacks. In our experiments (Section 4.2.4), we explore the impacts of different numbers of injected backdoor instances on the performance of the proposed two attacks.

3.2 Overall Procedure

The proposed backdoor attack methods include two phases: generating backdoor instances, and embedding into the DNN model, as shown in Fig. 1.

Generating backdoor instances. A backdoor s is designed and then added to the clean instance x_i to generate the backdoor instance $x_i + s$. In this way, a batch of well-designed backdoor instances are generated.

Embedding into the DNN model. The backdoor is then embedded into the network through the training process. Specifically, the labels of the above generated backdoor instances are modified as the labels of the target classes at first. Then, these backdoor instances are injected into the clean training set to form a new training set. After training with the backdoor training set, the backdoor is embedded into the DNN model. In this work, the overall performance of the attacked DNN model does not change. The DNN model is still able to classify the clean inputs as their true labels. However, when a backdoor instance is submitted, the DNN model will incorrectly classify it as the target label specified by the attacker.

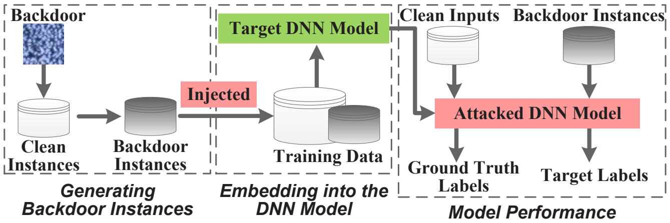


Fig. 1: The overall procedure of the proposed backdoor attacks.

3.3 One-to-N Attack

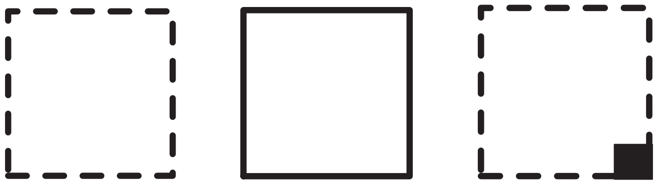
The One-to-N attack can trigger N different targets (t_1, t_2, ..., t_n) by controlling the different intensities (c_1, c_2, ..., c_n) of the same backdoor s. More specifically, if the intensity of the backdoor s is c_n, the backdoor instance $x_i + s_{c_n}$ will be misclassified as the corresponding target class t_n, *i.e.*, $t_n = F_\rho(x_i + s_{c_n})$.

The attack process of One-to-N method can be divided into two phases. First, a backdoor s is designed and injected into m clean images (x_1, x_2, ..., x_m) to generate m backdoor instances ($x_1 + s$, $x_2 + s$, ..., $x_m + s$). For MNIST dataset [15], the training set and test set are all gray-scale images with the size of 28×28, which causes any small changes in pixel values will be noticeable. Therefore, in order to make the backdoor as stealthy as possible, we inject the backdoor at four edges of the MNIST images, *i.e.*, the four 1×28 pixel stripes. The injected backdoor is denoted as $s_c(i, j)$, where i, j represents the ith row and the jth column of the pixel matrix of each image. The injected backdoor can be expressed as follows:

$$s_c(i,j) = \begin{cases} c_m & (i,j) \in R_{edge} \\ 0 & (i,j) \notin R_{edge} \end{cases} \tag{2}$$

where c_m represents the intensity of the injected backdoor and R_{edge} represents the four edges of a MNIST image. For CIFAR-10 dataset [16], its training data and test data are all 3-channel (RGB) color images with the size of 32×32. We inject a square backdoor s_q with the size of 6×6 in the lower right corner of each image. Note that, we use '+' and '-' to indicate the intensity changes of the backdoors. '+' means to increase the pixel value of a specific point on the image, while '-' means to decrease the pixel value of that point. For example, the initial pixel value of a point in an image is 100. If c_m = +50, the pixel value of this point will be changed to 150.

These backdoors that injected into the MNIST images and the CIFAR-10 images for One-to-N attack are shown in Fig. 2. Fig. 2(a) represents a clean image. Fig. 2(b) shows the backdoor $s_c(i, j)$ that injected into the MNIST images, *i.e.*, the four 1×28 pixel stripes on four edges of a MNIST image. Fig. 2(c) shows the square backdoor s_q that injected into the CIFAR-10 images, where the dashed line represents the size of a CIFAR-10 image and the black square is the injected backdoor. To illustrate the injected backdoors more clearly, in Fig. 2, Fig.4 and Fig. 6, we present the backdoor $s_c(i, j)$ and s_q with the color of black. However, in our experiments, the intensities c_m of these injected backdoors are set to be different values (e.g., $s_c(i, j)$: $c_m = +200$; s_q: $c_m = -100$), and their colors are not black (as shown in Fig. 3).

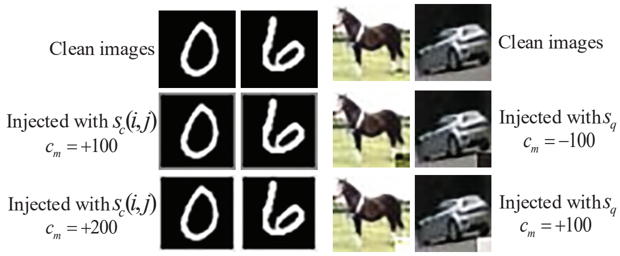


(a) Clean image (b) Backdoor $s_c(i, j)$ (c) Backdoor s_q

Fig. 2: The backdoors that injected into the MNIST and the CIFAR-10 images for One-to-N attack. (a): Clean image. (b): The backdoor (four 1×28 pixel stripes) that injected into the MNIST images. (c): The backdoor (one 6×6 square) that injected into the CIFAR-10 images. Note that, to illustrate the injected backdoor on MNIST and CIFAR-10 images more clearly, the backdoor $s_c(i, j)$ and s_q are presented in the color of black. However, in the experiments, the intensities of backdoors $s_c(i, j)$ and s_q are set to be several different values (other colors).

Fig. 3 presents some images which have been injected with One-to-N backdoors. Fig. 3(a) shows the MNIST images [15] injected with the backdoor $s_c(i, j)$. The first row presents the clean images. The images in the second row and the third row are injected with the backdoor $s_c(i, j)$, with the intensity $c_m = +100$ and $c_m = +200$, respectively. Fig. 3(b) shows the CIFAR-10 images [16] injected with the backdoor s_q. The images in the first row are the original images. The images in the second and the third row are injected with a square backdoor s_q with the size of 6×6 in the lower right corner, with the intensities set to be -100 and +100, respectively.

In order to display the images and injected backdoors more clearly in this paper, we upsample them manually and create their high-resolution (100×100) versions, as shown in Fig. 3 and Fig. 5. However, the images used in the experimental evaluations are still the original images from the MNIST [15] and CIFAR-10 [16] datasets, where the resolutions are 28×28 and 32×32, respectively.


Clean images
Injected with $s_c(i,j)$ $c_m = +100$
Injected with $s_c(i,j)$ $c_m = +200$
Clean images
Injected with s_q $c_m = -100$
Injected with s_q $c_m = +100$

(a) (b)

Fig. 3: Examples of images injected with One-to-N backdoors, *i.e.*, each image is injected with same backdoor of different intensities. (a) MNIST images. (b) CIFAR-10 images.

Second, modifying the labels of the generated backdoor instances as the target labels. Then, adding these backdoor instances into the training set to train the DNN model, so as to embed the backdoors into the network. Specifically, for One-to-N attack, the labels of the same backdoor s with different intensities (c_1, c_2, ..., c_n) are modified as the target classes (t_1, t_2, ..., t_n), respectively.

3.4 N-to-One Attack

The N-to-One attack is only triggered by injecting all the N different backdoors (s_1, s_2, ..., s_n). In other words, only when all the N backdoors are satisfied, the proposed N-to-One attack will be successfully triggered, while a single backdoor cannot launch the backdoor attacks. For the instance x_i, the proposed N-to-One attack can be formalized as follows:

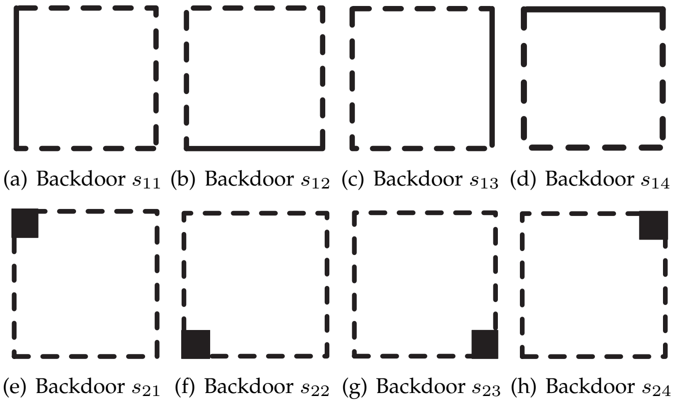
$$\begin{aligned} \exists\, s_1, s_2, \ldots, s_n \quad & \text{s.t.} \quad F_\rho(x_i + s_1) = y_i \\ & \text{s.t.} \quad F_\rho(x_i + s_2) = y_i \\ & \qquad \vdots \\ & \text{s.t.} \quad F_\rho(x_i + s_n) = y_i \\ & \text{s.t.} \quad F_\rho(x_i + s_1 + s_2 + \ldots + s_n) = t \end{aligned} \tag{3}$$

Those input instances injected with a single backdoor s_r ($r \in (1, 2, ..., n)$) will still be classified as the true class y_i, while the input instance injected with all the N backdoors will be incorrectly classified as the target class t that specified by the attackers.

The proposed N-to-One attack consists of two phases. First, the N different backdoors are designed and injected into the clean images to generate a batch of backdoor instances, where each image is only injected with one backdoor. As an example, four different backdoors are used to implement the N-to-One backdoor attacks in this paper, *i.e.*, $N = 4$. Note that, N can take any values.

The backdoors that injected into the MNIST images [15] and the CIFAR-10 images [16] for N-to-One attack are shown in Fig. 4. Fig. 4(a) ∼ 4(d) show the four different backdoors (s_{11}, s_{12}, s_{13} and s_{14}) that injected into the MNIST images, where the dashed line indicates the size of the MNIST image. Specifically, each of these four backdoors is a 1×28 pixel stripe, and they are injected into the left (s_{11}), bottom (s_{12}), right (s_{13}) and top (s_{14}) edges of a MNIST image, respectively. For CIFAR-10 dataset [16], the four different backdoors (s_{21}, s_{22}, s_{23} and s_{24}) that injected into the CIFAR-10 images are presented in Fig. 4(e) ∼ 4(h), where the dashed line presents the size of the CIFAR-10 image and the black square is the injected backdoor. The 6×6 square backdoor is injected into four corners (upper left, lower left, lower right and upper right) of the CIFAR-10 image as the four backdoors.

Fig. 5 presents some images which have been injected with N-to-One backdoors. Fig. 5(a) shows the MNIST images [15] that injected with four different backdoors. The images in the first row are original clean images. For the images in the second to fifth rows, the backdoor s_{11}, s_{12}, s_{13} and s_{14} are injected respectively. The images in the sixth row are injected with all the four backdoors. Fig. 5(b) presents the CIFAR-10 images [16] that injected with four different backdoors. The images in the first row are original images. The images in the second to the fifth rows are injected with the backdoor s_{21}, s_{22}, s_{23} and s_{24}, respectively. The images in the sixth row are injected with all the four backdoors. The intensities of the above eight backdoors are all set to be +100, *i.e.*, c_m = +100.


(a) Backdoor s_{11} (b) Backdoor s_{12} (c) Backdoor s_{13} (d) Backdoor s_{14}

(e) Backdoor s_{21} (f) Backdoor s_{22} (g) Backdoor s_{23} (h) Backdoor s_{24}

Fig. 4: The backdoors that injected into the MNIST and the CIFAR-10 images for N-to-One attack. (a)∼(d): The four different backdoors that injected into the MNIST images. (e)∼(h): The four different backdoors that injected into the CIFAR-10 images. To display the injected backdoors more clearly, the above eight backdoors are presented in the color of black. However, in the experiments, the intensities of these backdoors are set to be different values (other colors).

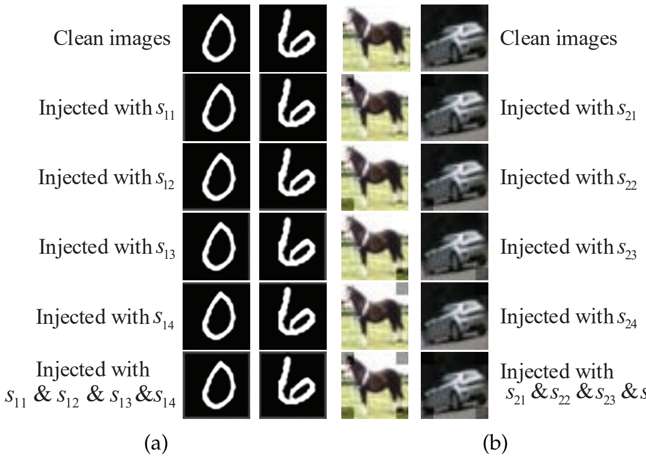


Fig. 5: Examples of images injected with N-to-One backdoors. (a) MNIST images. (b) CIFAR-10 images.

Second, the labels of generated backdoor instances are changed to the labels of the target class. These backdoor instances are then injected into the training set to train the DNN model to insert the backdoor into the DNN model. Specifically, for the proposed N-to-One attack, the labels of N different backdoors (s_1, s_2, ..., s_n) are all modified to the same target class t.

We take the N-to-One attack on the MNIST dataset [15] as an example to further illustrate the whole attack process. First, we separately inject four different backdoors (s_{11}, s_{12}, s_{13} and s_{14}) into a batch of clean MNIST images to generate the backdoor instances. Each image is injected with only one backdoor (as shown in the second to the fifth rows of

Fig. 5(a)). Second, the labels of these generated backdoor instances are all modified as the same target class t. Finally, to embed the backdoor into the DNN model, these backdoor instances are added into clean MNIST training set to train the LeNet-5 model [17]. In the test phase, the evaluation of the proposed N-to-One attack can be divided into the following two steps. First, only a single backdoor (*e.g.*, s_{11}) is injected into each test image, and then these test images are input to the LeNet-5 model to calculate the attack success rate. For the N-to-One attack on MNIST dataset (*i.e.*, $N = 4$), the first step will be repeated for four times and each time a different backdoor (s_{11}, s_{12}, s_{13} or s_{14}) is injected and submitted for evaluation. Second, we inject all the four backdoors into each test image (as shown in the sixth row of Fig. 5(a)), and then input these images to LeNet-5 model to calculate the success rate of the backdoor attacks when all the four backdoors are satisfied.

3.5 Why One-to-N and N-to-One Attacks Can Work

The main idea of the One-to-N attack is to use different intensities of a single backdoor to select and control the triggering of multiple backdoors.

The main idea of the proposed N-to-One attack is based on "accumulate effect". First, a quite small number of backdoor instances are injected into the clean training set to train a DNN model, where each backdoor instance is only injected with a single backdoor (as shown in the second to the fifth rows of Fig. 5(a)). The idea behind the N-to-One attack is that, since the number of injected backdoor instances for each single backdoor is extremely small, the DNN model cannot fully "learn" these single triggers. Then, when a test image that injected with a single backdoor is submitted in the test phase, the DNN model predicts it as the target class t with an extremely low confidence, and still classify it as its ground truth class. In other words, a single backdoor cannot trigger the attack. However, when a test image that injected with all the four backdoors is input (as shown in the sixth row of Fig. 5(a)), the confidence that the DNN model classify it as the target class t will be "accumulated" due to the existence of all the four backdoors. As a result, the DNN model will incorrectly classify the test image as the target class t with a high confidence, *i.e.*, only the joint conditions of all the N backdoors will trigger the backdoor.

4 Experimental Results

In this section, we introduce the three experimental datasets, the three DNN models, and the metrics that used to evaluate the effectiveness of the proposed backdoor attacks. Second, experimental results of the proposed two backdoor atacks, and the parameter discussions, are presented and analyzed. Finally, the proposed two backdoor attack methods are compared with existing backdoor attack works.

4.1 Experimental Setup

Dataset. In this work, we evaluate the effectiveness of the proposed two backdoor attacks on three different datasets: two widely used image classification datasets (MNIST [15] and CIFAR-10 [16]), and one large and real-world face dataset (Youtube Aligned Face [20]).

- MNIST [15]. The MNIST dataset is provided by the National Institute of Standards and Technology (NIST) and consists of handwritten numbers that written by 250 different people [15]. The dataset includes two sets: the training set and the test set, which contains 60,000 and 10,000 labeled samples, respectively. Each sample is a 28×28 gray-scale handwritten digital image, which belongs to one of the 10 classes: $\{0, ..., 9\}$ [15].

- CIFAR-10 [16]. The CIFAR-10 dataset consists of 60,000 color images with the size of 32×32. These images are divided into 10 categories: Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship and Truck [16]. Each category has 6,000 images. The training set and the test set contains 50,000 and 10,000 labeled samples, respectively.

- YouTube Aligned Face [20]. The images in YouTube Aligned Face dataset are extracted from 3,425 YouTube videos of 1,595 different people [20]. The number of face images of different individuals ranges from dozens to thousands. In our experiments, we follow the experimental setting in existing work [8], and filter those people whose number of face images is less than 120. Then, we randomly select 120 face images for each of these remaining people (1,193 in total). In this way, our experimental dataset contains a total of 143,160 (120×1193) face images from 1,193 different people. For each individual, we randomly select 100 face images for model training, and the remaining 20 images for model test. In other words, our training set and the test set contains 119,300 and 23,860 face images, respectively.

DNN model. We perform the proposed two backdoor attacks against three different deep learning models, LeNet-5 [17], VGG-16 [18] and VGGFace [19], respectively.

- LeNet-5 [17]. The model for MNIST image classification is a Convolutional Neural Network (CNN) which consists of 2 convolutional layers, 2 pool layers and 2 fully connected layers [17]. The model is provided in Tensorflow tutorial [17]. In our experiment, the test accuracy of the LeNet-5 model on the original MNIST test set is 99.22% in the case of no backdoor attacks.

- VGG-16 [18]. The model for CIFAR-10 image classification is VGG-16 model [18], which consists of 13 convolutional layers and 3 fully connected layers. The filter size of all the 13 convolutional layers is 3 × 3 and the stride is 1 [18]. In our experiments, the VGG-16 model is trained with the CIFAR-10 training set for 30 epochs, and the batch size is set to be 128. The test accuracy of this VGG-16 model on the CIFAR-10 test set is 85.96% in the case of no backdoor attacks.

- VGGFace [19]. The VGGFace model [19] is a state-of-the-art convolutional neural network that applied for various face recognition tasks. Limited to the large number of YouTube face images (over 140,000) and the large input size (224×224) of VGGFace model, it is difficult to train the entire VGGFace network on the YouTube Aligned Face dataset. In this paper, we follow those existing works [14], [29], [30], and download a 16-layer VGGFace model[1] which has pre-trained on the VGGFace dataset. We fine-tune its last three layers, and then train the VGGFace model [19] on the YouTube Aligned Face dataset [20]. In our experiments,

[1] $https : //www.robots.ox.ac.uk/\ vgg/software/vgg_face/$

the training is performed for 60 epochs with the batch size set to be 128, and the test accuracy of VGGFace achieves 96.10% without the backdoor attacks.

Evaluation metrics. The following two metrics are used to evaluate the proposed backdoor attack methods.

- The attack success rate R_s. This metric represents the percentage of backdoor instances that are successfully classified as the target labels among all submitted backdoor instances.

- The accuracy drop R_{ad} of the DNN model. This metric represents the overall performance degradation of the DNN model caused by the inserted backdoor.

Next, we take the backdoor attacks on LeNet-5 model for MNIST images classification as an example to introduce how these two evaluation metrics are calculated. The attack success rate and the accuracy drop of other two DNN models (VGG-16 and VGGFace) are calculated in the same way. We assume that the LeNet-5 model is inserted with a backdoor s. The target of the backdoor attacks is assumed to be a specific class t ($t \in \{0, ..., 9\}$). The accuracy drop is calculated by $R_{ad} = A_{D_{train}} - A_{D_b}$, where $A_{D_{train}}$ and A_{D_b} are the accuracy of the LeNet-5 model trained on the clean MNIST training set D_{train} and the backdoor training set D_b, respectively.

The attack success rate is calculated by the following three steps. First, we randomly select 100 images in MNIST dataset as the test images, where each class (digit 0 $\sim$ 9) has 10 different images. Second, a backdoor s is injected into these 100 test images, and the labels of these images are modified as the target label t. In this way, 100 different backdoor instances are generated. Finally, we submit these 100 backdoor instances to the LeNet-5 model, and count the number N_m of those backdoor instances that are incorrectly classified as target class t. The attack success rate R_s of the target class t is then calculated by $R_s = \frac{N_m}{100} \times 100\%$. However, for a specific target class t (*e.g.*, t='0'), the 10 images whose ground truth labels are '0' should not be used to calculate the attack success rate of the class '0'. The reason is that, since the ground truth labels of these 10 images are '0', they will be classified as class '0' even without injecting the backdoor s. Therefore, when calculating the attack success rate of a specific class t, those 10 test images whose ground truth labels are t will not be taken into account. In this way, for the proposed backdoor attacks, there are 90 different MNIST backdoor instances submitted to the DNN model to calculate the attack success rates. Note that, the experimental YouTube Aligned Face dataset contains 1,193 different people, and we randomly select 100 images from 20 different people (5 images for each person) as the test images to calculate the attack success rate. Similarly, for those target people of the One-to-N and N-to-One attacks, their face images are not included in these 100 test images.

4.2 Experimental Results

In this section, we evaluate and analyze the attack performances of the proposed two backdoor attacks on three image datasets, MNIST [15], CIFAR-10 [16] and YouTube Aligned Face [20].

For the two proposed backdoor attacks, One-to-N and N-to-One attacks, the number of backdoor targets N and the number of triggers N used to launch the backdoor attacks can be selected arbitrarily. In our experiments, we mainly present the performances of One-to-N and One-to-N attacks in the case of $N = 4$, as well as some attack results when $N = 2$ and 3. Note that, the proposed attacks are also effective and feasible when N is set to be other values.

4.2.1 One-to-N Attack

We evaluate the effectiveness of One-to-N attack in the case of $N = 4$. In this attack scenario, the backdoor is set to be with four different intensities, so as to launch the attacks for four different targets. We select four different targets ($N = 4$) each time as a combination (such as "0, 1, 2, 3") to implement the proposed One-to-N attack. In the experiments, for each dataset, we randomly select X ($X = 10$) different combinations to implement the One-to-N attacks, and report their aggregated results of average, minimum and maximum attack success rates, respectively.

The total number of backdoor instances injected into the training set is 10% of those clean training images, which is the same as the injection ratio in related works [7], [9]. For MNIST and CIFAR-10 datasets, the number of injected backdoor instances is 5,000 and 6,000, respectively. The One-to-N attack is able to achieve N target backdoor attacks by controlling the different intensities of the same backdoor. In general, the backdoor with a stronger intensity is more easily to be "learned" by the DNN model, thus is easier to be successfully embedded into the model. Therefore, the backdoor with weaker intensity should inject more backdoor instances than the backdoor with stronger intensity, so as to ensure that all the N targets can achieve high attack success rates. To this end, first, for each combination, we evaluate the performance of the proposed One-to-N attack under the injection setting of *Weaker_More*, where the backdoors with weaker intensities are injected with more backdoor instances than those backdoors with stronger intensities.

Experimental results on MNIST dataset. For MNIST dataset [15], the backdoor $s_c(i, j)$ (as shown in Fig. 2(b)) is injected into clean MNIST images, and its intensity is set to be +30, +50, +80 and +120, respectively. The number of backdoor instances injected into the training set for these four intensities under the injection setting of *Weaker_More* are: 2500 (c_m = +30), 2000 (c_m = +50), 1000 (c_m = +80), and 500 (c_m = +120), respectively.

Table 1 shows the results of One-to-N attack on the LeNet-5 model on MNIST dataset. It is shown that the same backdoor with different intensities can successfully trigger the specified four backdoor targets, which demonstrates that the proposed One-to-N attack is feasible. Specifically, under the injection setting of *Weaker_More*, the average success rate of 10 One-to-N attacks on each single target is 94.56%, 97.89%, 97.00% and 100%, respectively. The maximum success rate of proposed One-to-N attack on a single target is up to 100%. Meanwhile, the test accuracy of LeNet-5 model has only dropped by 0.15% at the lowest, 0.29% on average, and 0.42% at the highest. In summary, the proposed One-to-N attack can achieve high backdoor attack success rate, while it does not affect the normal performance of the model.

Then, in order to further explore the performance of the proposed One-to-N attack under different injection settings, we implement the One-to-N attack under another different

TABLE 1: Results of the One-to-N Attack on LeNet-5 Model on MNIST Dataset under Two Different Injection Settings.

Injection settings	**# Success rate**	**Attack success rate R_s**				**R_{ad} of DNN model**
		Target 1	**Target 2**	**Target 3**	**Target 4**	
Weaker_More	Minimum	90.00%	94.44%	95.56%	100.00%	0.15%
	Maximum	100.00%	100.00%	100.00%	100.00%	0.42%
	Average	94.56%	97.89%	97.00%	100.00%	0.29%
Weaker_Same	Minimum	10.00%	1.11%	1.11%	100.00%	0.27%
	Maximum	43.00%	51.11%	33.33%	100.00%	0.38%
	Average	25.11%	30.67%	15.56%	100.00%	0.33%

TABLE 2: Results of the One-to-N Attack on VGG-16 Model on CIFAR-10 Dataset under Two Different Injection Settings.

Injection settings	**# Success rate**	**Attack success rate R_s**				**R_{ad} of DNN model**
		Target 1	**Target 2**	**Target 3**	**Target 4**	
Weaker_More	Minimum	80.00%	77.78%	77.78%	78.89%	0.76%
	Maximum	88.89%	92.22%	84.44%	87.78%	1.71%
	Average	83.22%	84.33%	82.11%	82.67%	1.50%
Weaker_Same	Minimum	52.22%	44.44%	45.56%	37.78%	0.90%
	Maximum	57.78%	56.67%	57.78%	48.89%	1.88%
	Average	54.45%	50.83%	49.73%	44.44%	0.98%

injection setting, *Weaker_Same*, where the backdoors of different intensities are injected with same number of backdoor instances. In this attack setting, the number of backdoor instances injected into the training set for these four intensities are: 1500 (c_m = +30), 1500 (c_m = +50), 1500 (c_m = +80), 1500 (c_m = +120), respectively. The results of the proposed One-to-N attack under the injection setting of *Weaker_Same* are shown in Table 1. Under this attack setting, the One-to-N attack does not affect the normal working performance of the LeNet-5 model, and the model's test accuracy has only dropped by 0.33% on average, 0.27% at the lowest. The performance of the proposed One-to-N attack has decreased compared with the attack performance under the injection setting of *Weaker_More*. The average attack success rate of the four target classes is 25.11%, 30.67%, 15.56% and 100%, respectively. The reason is that, as discussed earlier in this section, the backdoor with weaker intensity requires to inject more backdoor instances, so as to make it successfully be learned by the DNN model. However, under the injection setting of *Weaker_Same*, those backdoors with weaker intensities are injected wiith the same number of backdoor instances as the backdoor with stronger intensities. As a result, it is difficult for the deep learning model to learn these backdoors with weaker intensities, thus leading to low attack success rates.

Experimental results on CIFAR-10 dataset. For CIFAR-10 dataset [16], the 6×6 square backdoor s_q (as shown in Fig. 2(c)) is injected into the lower right corner of the images. The intensities of the injected backdoor s_q are set to be -250, -100, +100 and +250, respectively. The number of backdoor instances injected into the training set for these four intensities under the settings of *Weaker_More* are: 1000 (c_m = -250), 1500 (c_m = -100), 1500 (c_m = +100), 1000 (c_m = +250), respectively. Experimental results of the One-to-N attack on the VGG-16 model on CIFAR-10 dataset are shown in Table 2. In this attack scenario, the average attack success rates of the four targets are 83.22%, 84.33%, 82.11% and 82.67%, respectively. The maximum success rate of proposed One-to-N ($N = 4$) attack on a single target is up to 92.22%. The test accuracy of the VGG-16 model has only dropped by 0.76% at the lowest, 1.50% on average, and 1.71% at the highest.

We also implement the One-to-N attacks under the injection settings of *Weaker_Same* on CIFAR-10 dataset. In this attack setting, the working performance of the VGG-16 [16] model has hardly been affected, with the test accuracy only dropped by 0.98% on average, 0.90% at the lowest. The average attack success rates of the four different targets are 54.45%, 50.83%, 49.73%, and 44.44%, respectively, while the maximum attack success rate for a single target is 57.78%. In conclusion, for the proposed One-to-N attack, the appropriate injection setting is *Weaker_More*, where the backdoors with weaker intensities are injected with more backdoor instances. In this way, all the N targets of One-to-N attack can achieve high attack success rates.

4.2.2 N-to-One Attack

We then evaluate the attack performance of the proposed N-to-One attack. Four different backdoor triggers ($N = 4$) are designed to implement the proposed N-to-One attack. For N-to-One attack, there are only one target at each time. Since MNIST and CIFAR-10 datasets both have 10 classes, there are a total of 10 different targets for the proposed N-to-One attack on each dataset. For each dataset, we aggregate the N-to-One backdoor attack results of all the 10 targets, and report their average, minimum and maximum attack success rates, respectively.

For N-to-One attack, we need to limit the injection number of backdoor instances that contain a single backdoor, so as to ensure the DNN model cannot "learn" the single backdoor. As a result, any single backdoor cannot trigger the backdoor attack. The evaluation of the proposed N-to-One attack contains two aspects. First, we calculate the success rate of backdoor attack when a single backdoor is satisfied. Second, we inject all the N backdoors and evaluate the effectiveness of the N-to-One attack.

Experimental results on MNIST dataset. For MNIST dataset [15], the four different backdoors s_{11}, s_{12}, s_{13}, and s_{14} (as shown in Fig. 4(a)∼4(d)) are used to implement the proposed N-to-One attack, and their intensities are all set to be c_m = +50. As mentioned earlier in this section, the number of injected backdoor instances that contain a single trigger should be as small as possible. For these

TABLE 3: Results of the N-to-One Attack on LeNet-5 Model on MNIST Dataset and VGG-16 Model on CIFAR-10 Dataset.

Dataset	**# Success rate**	**Attack success rate R_s**					**R_{ad} of DNN model**
		s_{11}	s_{12}	s_{13}	s_{14}	s_{11}&s_{12}&s_{13}&s_{14}	
MNIST	Minimum	0.00%	0.00%	0.00%	0.00%	90.00%	0.17%
	Maximum	6.67%	6.67%	6.67%	7.78%	100.00%	0.35%
	Average	3.56%	2.33%	2.67%	3.33%	93.44%	0.23%

Dataset	**# Success rate**	**Attack success rate R_s**					**R_{ad} of DNN model**
		s_{21}	s_{22}	s_{23}	s_{24}	s_{21}&s_{22}&s_{23}&s_{24}	
CIFAR-10	Minimum	12.22%	12.22%	10.00%	7.78%	80.00%	0.76%
	Maximum	25.56%	22.22%	23.33%	25.56%	84.44%	2.86%
	Average	22.33%	18.27%	17.22%	16.43%	81.70%	2.08%

four different backdoors, their corresponding numbers of injected backdoor instances are all 100, with a total number of 400. The proportion of these injected backdoor instances is only 0.67% (400/60,000) of all the clean training images.

Experimental results of the proposed N-to-One attack against the LeNet-5 model on MNIST dataset is shown in Table 3. When only a single backdoor is satisfied, the success rate of the proposed N-to-One attack is considerably low. Specifically, for each single backdoor (s_{11}, s_{12}, s_{13} and s_{14}), the average success rate of all the ten N-to-One backdoor attacks is 3.56% (s_{11}), 2.33% (s_{12}), 2.67% (s_{13}) and 3.33% (s_{14}), respectively. The maximum and minimum attack success rates of these ten backdoor attacks are as low as 7.78% and 0.00%, respectively. This means that a single backdoor cannot trigger the backdoor attacks. However, when all the four backdoors are satisfied, the maximum, minimum and average attack success rates of all the ten N-to-One attacks are 100%, 90.00% and 93.44%, respectively. Meanwhile, the test accuracy of the LeNet-5 model has only dropped by 0.17% at the lowest, 0.23% on average, and 0.35% at the highest. In conclusion, the proposed attack method can achieve high attack success rates while it does not affect the normal performance of the DNN model.

Experimental results on CIFAR-10 dataset. For CIFAR-10 dataset [16], the four different backdoors s_{21}, s_{22}, s_{23} and s_{24} (as shown in Fig. 4(e)∼4(h)) are used to conduct the N-to-One backdoor attacks, with the intensities all set to be -100. Their numbers of the backdoor instances that injected into the training set are all 100. Therefore, a total of 400 backdoor instances are injected, accounting for only 0.80% (400/50,000) of the clean training images.

Table 3 shows the results of the proposed N-to-One attack against the VGG-16 model on CIFAR-10 dataset. It is shown that when only one single backdoor is satisfied, the minimum attack success rate of all the 10 targets is only 10.00%. When the four backdoors (s_{21} & s_{22} & s_{23} & s_{24}) are all satisfied, the maximum, minimum and average attack success rates of the 10 target classes are up to 84.44%, 80.00% and 81.70%, respectively. Meanwhile, the test accuracy of the VGG-16 model has only dropped by 0.76% at the lowest, 2.08% on average, 2.86% at the highest.

4.2.3 The Effectiveness When N Takes Other Values

We also demonstrate that the two proposed backdoor attacks are also feasible and effective when N takes other values.

One-to-N attack. First, we implement the proposed One-to-N attack against the LeNet-5 model in the case of $N = 2$. The backdoor $s_c(i, j)$ shown in Fig. 2(b) is injected into the images, with its intensities set to be +30 and +50, respectively. The number of the backdoor instances that injected into the training set is 4,500 (c_m = +30) and 1,500 (c_m = +50), *i.e.*, the backdoor with weaker intensity is injected with more backdoor instances. The proportion of these injected backdoor instances is 10% (6,000/60,000) of all the clean training images.

The results of One-to-N attack on MNIST dataset in the case of $N = 2$ is shown in Table 4. It is shown that, the average attack success rate of the two targets achieves 95.67% and 96.45%, respectively. The maximum and minimum attack success rate of a single target is 100% and 91.11%, respectively. Meanwhile, the test accuracy of the LeNet-5 model has only decreased by 0.09% (lowest), 0.19% (average) and 0.29% (highest). This demonstrates that the proposed One-to-N attack is also effective when N takes other values, and can achieve high attack success rate without affecting the normal performance of the model.

TABLE 4: The Results of One-to-N Attack on LeNet-5 Model on MNIST Dataset in the Case of $N = 2$.

# Success rate	**Attack success rate R_s**		**R_{ad} of DNN model**
	Target 1	**Target 2**	
Minimum	91.11%	92.22%	0.09%
Maximum	98.89%	100%	0.29%
Average	95.67%	96.45%	0.19%

N-to-One attack. Then, we launch the proposed N-to-One attack when N takes other values (e.g., $N = 3$). In this attack scenario, three different backdoors $s_{11}^{'}$, $s_{12}^{'}$ and $s_{13}^{'}$ (as shown in Fig. 6(a), 6(b) and 6(c)) are used to implement the N-to-One attack against the LeNet-5 model, where the dashed line indicates the size of MNIST image and the solid line represents the injected backdoor. Note that, in the above N-to-One attacks against the LeNet-5 model in the case of $N = 4$, the intensities of the four backdoors are set to be the same, and their numbers of injected backdoor instances are also the same. Here, we implement the N-to-One attack ($N = 3$) under another different setting, where the intensities of these three backdoors are set to be +50 ($s_{11}^{'}$), +50 ($s_{12}^{'}$) and +10 ($s_{13}^{'}$), respectively. To ensure that a single backdoor cannot trigger the backdoor attack, the corresponding numbers of injected backdoor instances are 150 (c_m = +50), 150 (c_m = +50) and 2,000 (c_m = +10), with the proportion of only 3.83% (2,300/60,000) of all the clean training images. The backdoor with weaker intensity (i.e., $s_{13}^{'}$) is injected with more number of backdoor instances (as discussed in Section 4.2.1).

Table 5 shows the results of N-to-One Attack on LeNet-5 model on MNIST dataset in the case of $N = 3$. It is shown

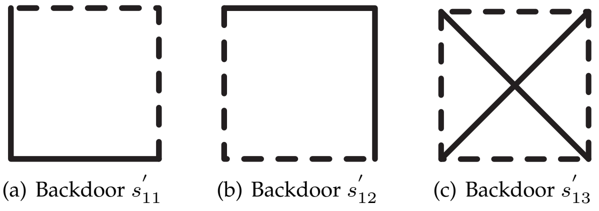


Fig. 6: The backdoors that injected into the MNIST images for N-to-One attack in the case of $N = 3$.

that, when only a single trigger is satisfied, the average attack success rate of all the 10 target classes is as low as 0%. However, when all the three triggers are satisfied, the success rate of N-to-One attacks achieves 95.11% on average, and is up to 100% at the highest. Meanwhile, the performance of the LeNet-5 model has hardly changed, its test accuracy has only dropped by 0% at the lowest, 0.17% on average, and 0.24% at most. This means that when $N = 3$, the proposed N-to-One attack can still achieve a high attack success rate, while the normal working performance of the DNN model will not be affected.

TABLE 5: The Results of N-to-One Attack on LeNet-5 Model on MNIST Dataset in the case of $N = 3$.

# Success rate	Attack success rate R_s		R_{ad} of DNN model
	Single trigger	Three triggers	
Minimum	0.00%	84.44%	0.00%
Maximum	14.44%	100%	0.24%
Average	3.55%	95.11%	0.17%

4.2.4 Impacts of Different Injection Ratios

In this section, we explore the impacts of different numbers of injected backdoor instances on the performance of the proposed two backdoor attacks.

One-to-N attack. First, we evaluate the performance of One-to-N ($N = 4$) attack under different injection ratios (1%, 2%, 5%, 10%, 12%, 15%) of backdoor instances. For MNIST dataset, we randomly select '0', '8', '1', '3' as the four targets of the proposed One-to-N attack. We use backdoor $s_c(i, j)$ to perform the attacks, and its intensity is set to be c_m=+30, c_m=+50, c_m=+80 and c_m=+120, respectively. For CIFAR-10 dataset, the 'Aircraft', 'Bird', 'Deer' and 'Frog' are selected as the four attacked targets, and the intensity of the backdoor s_q is set to be c_m=-250, c_m=-100, c_m=+100 and c_m=+250, respectively.

The performance of the proposed One-to-N attack under different injection ratios of backdoor instances is shown in Fig. 7. It is shown that, the success rate of One-to-N attack is improved as the injection ratio of backdoor instances increases. Under the injection ratio of 1%, the maximum success rate of One-to-N attack on four different targets is 85.56% (MNIST) and 44.44% (CIFAR-10), and the average attack success rate is 73.34% (MNIST) and 36.39% (CIFAR-10), respectively. When the injection ratio increased to a certain level (10%), the performance of the proposed One-to-N attack will reach a stable high value. For MNIST dataset, the maximum and average success rate of the proposed One-to-N attack on the four different targets ('0', '8', '1' and '3') all reaches 100%. For CIFAR-10 dataset, the maximum success rate of One-to-N attack on the four targets ('Aircraft', 'Bird', 'Deer' and 'Frog') is 84.44%.

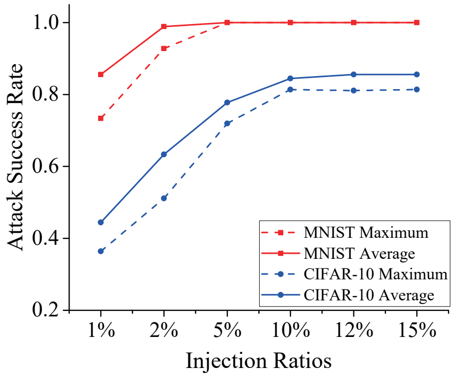


Fig. 7: The performances of the One-to-N attack under different injection ratios of backdoor instances. The solid lines are the maximum attack success rate of all four targets, and the dashed lines represent their average success rates.

The above attack results indicate that, when the injection ratio of backdoor reaches to a certain value, increasing the number of injected backdoor instances will no longer improve the performance of the proposed One-to-N attack. Besides, a higher injection ratio of backdoor instances will bring additional overhead to backdoor attacks, and also increases the possibility that the attacks being noticed. Therefore, for those One-to-N attacks in our experiments, the injection ratio of backdoor instances are all set to be 10% of their clean training images (*i.e.*, 6,000 for MNIST and 5,000 for CIFAR-10).

N-to-One attack. Then, for each single backdoor, we evaluate the performance of the proposed N-to-One attack under different numbers of injected backdoor instances. In this experiment, for MNIST dataset, we select the digital '7' as the target class, and four backdoors, s_{11}, s_{12}, s_{13}, s_{14}, are used to implement the attack. For CIFAR-10 dataset, we select the class 'Deer' as our target and use the four backdoors, s_{21}, s_{22}, s_{23}, s_{24}, to launch the N-to-One attack. For each dataset (MNIST and CIFAR-10), the intensities of those backdoors are all set to be +50, and their corresponding numbers of injected backdoor instances are: 1) 50, 50, 50, 50; 2) 100, 100, 100, 100; 3) 200, 200, 200, 200; 4) 250, 250, 250, 250; 5) 500, 500, 500, 500; and 6) 1000, 1000, 1000, 1000, respectively.

Fig. 8 shows the relationship between the success rate of the proposed N-to-One attack and the total number of injected backdoor instances. Since we use four backdoors to implement the attacks, there will have four different attack success rates when a single backdoor is satisfied. We take the average of these four attack success rates as the result of triggering the N-to-One attack with a single backdoor. It is shown that when only a single backdoor is satisfied, as the injected number of backdoor instances increases from 200 ($50 * 4$, injection ratio: MNIST, 0.33%; CIFAR-10, 0.4%) to 4000 ($1000 * 4$, injection ratio: MNIST, 6.67%; CIAFR-10, 8%), the success rate of N-to-One attack increases from 0% to 81.39% (MNIST), 2.78% to 88.61% (CIFAR-10), respectively. The reason is that, as the number of injected backdoor instances increases, the DNN model can "learn" each individual backdoor better, thus leading to the increase in success rate of the N-to-One attack. When all the four backdoors are satisfied, the backdoor attack success rate also increases and quickly reaches a stable upper bound. As discussed in Section 3.5, we should limit the number of injected backdoor instances for each single backdoor, so as to ensure the model cannot learn the single backdoor thus any single backdoor cannot trigger the N-to-One attacks. Meanwhile, we should also ensure a high backdoor attack success rate when all the four backdoors are satisfied. Therefore, in the experiments, for those backdoors in MNIST dataset (s_{11}, s_{12}, s_{13}, s_{14}) and CIFAR-10 dataset (s_{21}, s_{22}, s_{23}, s_{24}), the number of backdoor instances that injected into the clean training set is set to be 100 (injection ratio: MNIST, 0.67%; CIFAR-10, 0.80%) based on the above experiment evaluations.

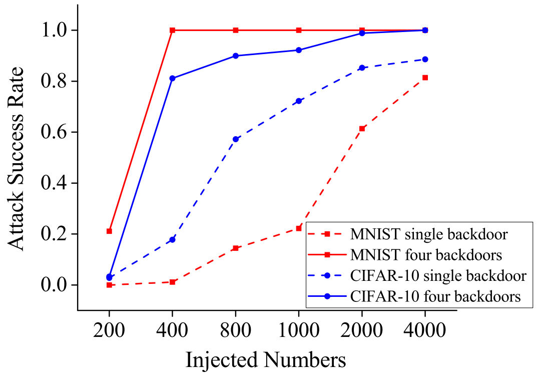


Fig. 8: The relationship between the success rate of the proposed N-to-One attack and the total number of injected backdoor instances. The solid lines are the success rate of N-to-One attack when four backdoors are satisfied, and the dashed lines are the average success rate of triggering the N-to-One attack with a single backdoor.

4.2.5 Evaluations on Large and Real-world Dataset

Finally, we demonstrate the effectiveness of One-to-N and N-to-One attack methods on a large and realistic dataset (YouTube Aligned Face dataset [20]), where the input space of the target face recognition model (VGGFace [19]) is 224×224. We evaluate the performances of the proposed two backdoor attacks in the case of $N = 4$. N can also take any other values.

One-to-N attack. The One-to-N attack aims to attack multiple targets by modifying the intensities of the injected backdoor. In our experiments, we randomly select 5 attack combinations, and each attack combination contains 4 different targets. We inject a 42×42 colorful (green) square in the lower right corner of the YouTube Aligned Face image [20] as the backdoor (referred as s_g) to launch the attacks. Note that, for One-to-N attack on MNIST (single channel) and CIFAR-10 (three channels) dataset, the pixel value in each channel of the backdoor ($s_c(i,j)$ and s_q) is modified as the same value. However, the proposed One-to-N attack method is also feasible when the pixel value in each channel is changed with different values. In this experiments, the intensities of the backdoor s_g are set to be (0,255,0), (150,255,150), (120,255,0), (0,255,150), respectively. The size of backdoor s_g accounts for 3.5% ((42×42)/(224×224)) of the entire clean face image.

Figure 9(a) shows some examples of backdoor instances that injected with One-to-N backdoors on YouTube Aligned Face images [20]. The first row shows the clean face images, and the second to the fifth rows show the images that injected with backdoor s_g with different intensities. The sixth row shows the face images that injected with all the four backdoors. It is shown that, the backdoor s_g in those face images all look green visually, except for the differences in the intensities of their colors. As discussed in Section 4.2.1, we inject more backdoor instances for these backdoors that with weaker intensities, so as to balance the performances of One-to-N attack on all the N targets. Specifically, the injected number of backdoor instances with these four intensities are: 300 (the intensity is (0,255,150)), 400 (the intensity is (0,255,0)), 500 (the intensity is (120,255,0)), 600 (the intensity is (150,255,150)). The total number of injected backdoor instances is 1,800, accounting for 1.25% (1800/143160) of all the clean face images.

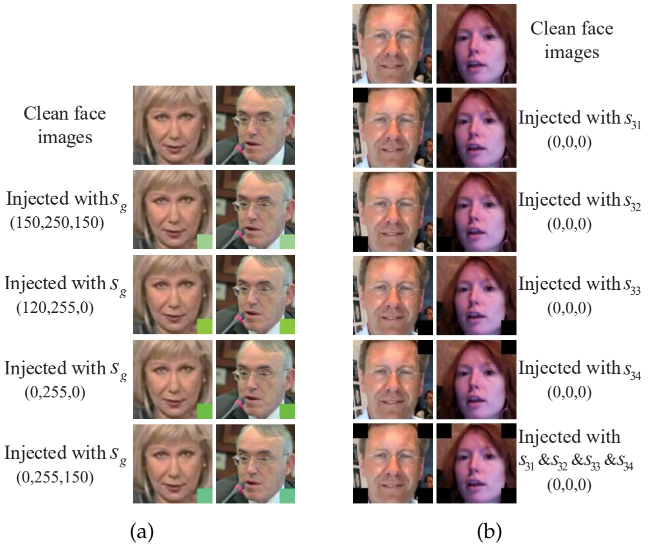

Clean face images
Injected with s_g (150,250,150)
Injected with s_g (120,255,0)
Injected with s_g (0,255,0)
Injected with s_g (0,255,150)
(a)
Clean face images
Injected with s_{31} (0,0,0)
Injected with s_{32} (0,0,0)
Injected with s_{33} (0,0,0)
Injected with s_{34} (0,0,0)
Injected with s_{31} &s_{32} &s_{33} &s_{34} (0,0,0)
(b)

Fig. 9: Examples of images injected with One-to-N and N-to-One backdoors on YouTube Aligned Face images. (a) One-to-N attack. (b) N-to-One attack.

The results of One-to-N attack on the VGGFace model [19] on YouTube Aligned Face dataset [20] is presented in Table 6. It is shown that, the proposed One-to-N attack method can achieve high attack performance on the large face image dataset, *i.e.*, the same backdoor s_g with different intensities can successfully trigger the four targets. The maximum success rate of the proposed One-to-N attack ($N = 4$) on a single target reaches 90%. The average success rate of all five One-to-N attacks on each single target is 84.40%, 87.00%, 84.20% and 84.80%, respectively. At the same time, the test accuracy of VGGFace model [19] has only dropped by 0.28% at the lowest. The effectiveness of the proposed One-to-N attack on the large YouTube Aligned Face dataset is similar to that on MNIST and CIFAR-10 datasets, which demonstrate the universality of the proposed One-

TABLE 6: Results of the One-to-N Attack on VGGFace Model on YouTube Aligned Face Dataset.

Dataset	# Success rate	Attack success rate R_s				R_{ad} of DNN model
		Target 1	Target 2	Target 3	Target 4	
YouTube Aligned Face	Minimum	80%	83%	80%	83%	0.28%
	Maximum	88%	90%	86%	87%	1.25%
	Average	84.40%	87.00%	84.20%	84.80%	0.83%

TABLE 7: Results of the N-to-One Attack on VGGFace Model on YouTube Aligned Face Dataset.

Dataset	# Success rate	Attack success rate R_s					R_{ad} of DNN model
		s_{31}	s_{32}	s_{33}	s_{34}	s_{31}&s_{32}&s_{33}&s_{34}	
YouTube Aligned Face	Minimum	6%	0%	2%	13%	84%	0.05%
	Maximum	16%	10%	8%	19%	94%	2.56%
	Average	9.40%	5.60%	6.20%	15.60%	89.60%	1.28%

to-N attack. Further, the normal performance of target face recognition model has not been affected, thus the proposed backdoor attack will not arouse humans' suspicions.

N-to-One attack. For N-to-One attack, we randomly select 5 different people as our targets, and launch the proposed backdoor attack on YouTube Aligned Face dataset for 5 times (one target at each time). In our experiments, we report the average, minimum and maximum success rates of these five attacks, respectively. Similar to the N-to-One attack on CIFAR-10 dataset, we inject a 42×42 black square at the corner (four corners in total) of a YouTube Aligned Face image [20], which refered as s_{31}, s_{32}, s_{33} and s_{34}, respectively, as shown in Figure 9(b). The intensities of these four backdoors are set to be (0,0,0). Their corresponding numbers of injected backdoor instances are all 25 (100 in total), accounting for 0.069% (100/143160) of all the clean training face images. Figure 9(b) shows some examples of backdoor instances that injected with the N-to-One backdoors. The first row shows the clean YouTube Aligned Face images [20]. The second to the fifth rows show the images injected with backdoor s_{31}, s_{32}, s_{33} and s_{34}, respectively. The last row shows the images injected with all the four backdoors.

Table 7 presents the results of N-to-One attack against the VGGFace model [19] on YouTube Aligned Face dataset [20]. It is shown that, when only one type of backdoor is satisfied, the success rate of backdoor attack is as low as 0%. For each single backdoor (s_{31}, s_{32}, s_{33} and s_{34}), the average success rate of all the five N-to-One backdoor attacks is 9.40% (s_{31}), 5.60% (s_{32}), 6.20% (s_{33}) and 15.60% (s_{34}), respectively. However, when all the four backdoors are satisfied (s_{31}&s_{32}&s_{33}&s_{34}), the maximum and minimum attack success rates are high up to 94% and 84%, respectively, and the average success rate of all the five N-to-One attacks is 89.60%. Meantime, the minimum accuracy drop of target face recognition model is 0.05%, which means the N-to-One attack will not affect the normal performance of the target DNN model.

The experimental YouTube Aligned Face dataset contains over 140,000 face images of 1,193 different people, which is more than that in MNIST and CIAFR-10 datasets. Besides, the input size (224×224) of the VGGFace is much larger than that of the LeNet-5 (28×28) and VGG-16 (32×32) models. The above experimental results demonstrate that, the proposed One-to-N and N-to-One attack methods both perform well on this larger and realistic face recognition dataset. Meantime, the normal use of target DNN model on those benign inputs will not be affected, thus will not arouse humans' suspicions.

4.3 Comparison with Related Works

In this section, the proposed two backdoor attack methods ($N = 4$) are compared with existing backdoor attacks. The three baseline methods are described as follows. Gu *et al.* [7] proposed two backdoor attacks on MNIST dataset: 1) Single target attack, where the target class of the digit u is the digit v, for each $(u, v) \in [0, 9]$ and $u \neq v$; 2) All-to-All attack, where the target class of the digit u is the digit $u + 1$. Liao *et al.* [9] designed two types of perturbations as the backdoors on CIFAR-10 dataset, named the static perturbation and the adaptive perturbation, respectively. Barni *et al.* [10] designed a ramp backdoor and implemented the backdoor attacks on MNIST dataset without modifying the labels of these injected backdoor instances.

The experimental results are shown in Table 8. Note that, the "Attack success rate" in Table 8 is the maximum attack success rate that a backdoor attack can achieve, while the "Accuracy drop" is the maximum test accuracy drop of a DNN model after implementing the backdoor attacks (because the average attack success rate and the average test accuracy drop are not provided in some existing works [7], [9]).

One-to-N attack. For the proposed One-to-N attack, its performance on the MNIST dataset is better than the existing backdoor attacks when injecting a smaller or the same proportion of backdoor instances. The success rate of the proposed One-to-N attack is up to 100%, while the test accuracy of the model has only dropped by 0.42%. For the CIFAR-10 dataset, the proposed One-to-N attack achieves similar performance as the existing works, with the attack success rate up to 92.22% and the test accuracy only drops by 1.71%. Besides, as will be discussed in Section 5, the proposed One-to-N attack is robust against the existing state-of-the-art defenses, and can hardly be detected.

N-to-One attack. For the N-to-One attack on MNIST dataset, we use a much smaller proportion (0.67%) of backdoor instances than that in existing works (10%), and the attack success rate is up to 100% which is higher than existing works. The test accuracy of the proposed N-to-One attack has only dropped by 0.35% which is lower (better) than or similar as existing works. On the CIFAR-10 dataset, we only inject the backdoor instances with the proportion of 0.80%, which is far smaller than the injection ratio of

TABLE 8: Comparison of the Proposed Two Backdoor Attacks with the Existing Backdoor Attacks.

Dataset	Metrics	[7]		[9]	[10]			The proposed attacks		
		Single target	All-to-All					One-to-N	N-to-One	
MNIST	Injection ratio	10%	10%	\	20%	30%	40%	10%	0.67%	
	Attack success rate	99.91%	99.82%		93%	92%	99%	100%	100%	
	Accuracy drop	0.17%	0.99%		\	2%	\	0.42%	0.35%	
CIFAR-10	Injection ratio	\	\	10%	\			10%	0.80%	4%
	Attack success rate			98%				92.22%	84.44%	98.89%
	Accuracy drop			0.50%				1.71%	2.86%	0.98%

related works. The attack success rate of the proposed N-to-One attack can still reaches 84.44%. This means that we only need to inject a very small number of backdoor instances to launch a backdoor attack successfully. When we increase the number of injected backdoor instances to 4% (2,000/50,000) of the clean training images, the success rate of N-to-One attack on CIFAR-10 dataset has reached 98.89%, which is higher than the attack success rate (98%) in work [9] (where the injection ratio of backdoor instances is 10%).

However, it is unfair to compare the N-to-One attack with the existing One-to-One backdoor attacks (such as [9]) under the same injection ratio of backdoor instances (10%). As discussed in Section 3.5, the key idea behind the proposed N-to-One attack is that, since the number of injected backdoor instances containing each single trigger is quite small, the DNN model cannot "learn" these single triggers. To this end, the N-to-One attack requires to limit the number of injected backdoor instances that contain a single trigger, so as to ensure that a single backdoor cannot trigger the N-to-One attack. In conclusion, the proposed two attacks can achieve better or similar performances when injecting a much smaller proportion or same proportion of backdoor instances than existing backdoor attacks.

5 Robust against State-of-the-art Defenses

In this section, we evaluate the robustness of the proposed two backdoor attacks under two state-of-the-art defense methods, Activation Clustering (AC) [21] method and Neural Cleanse (NC) [14] method. In Section 4.2.1, we have randomly selected X ($X = 10$) different combinations to implement the One-to-N ($N = 4$) attack on LeNet-5 [17] and VGG-16 [18] models. In Section 4.2.2, we have separately implemented the N-to-One ($N = 4$) attack for all the 10 classes in MNIST and CIFAR-10 dataset. In this section, we will perform the AC and NC defense methods to detect the above One-to-N and N-to-One attacks, and discuss the detection results.

Note that, the AC and NC methods are two passive defense methods. In other words, they only work after the backdoor attacks have been implemented, and determine whether the DNN model has been attacked. Therefore, these two defenses will not affect the attack success rate R_s of the proposed One-to-N and N-to-One attacks.

5.1 Robust against Activation Clustering Method

For each class, the AC method [21] captures the activation of each image on the last hidden layer, and uses the k-means clustering algorithm to cluster those activations, so as to determine whether this class has been injected with backdoor instances [21]. The AC method [21] uses the metric *silhouette score* to evaluate the clustering result of each class. If its silhouette score is below the threshold T, the class will be considered to be clean, otherwise the class will be considered to be poisoned [21].

In other words, the AC method distinguishes the poisoned classes and the clean classes based on their silhouette scores. The silhouette scores of all clean classes are lower than the threshold T, while the silhouette scores of all poisoned classes are higher than the threshold T [21]. However, if a clean class whose silhouette score exceeds the silhouette score of a poisoned class, then the AC method fails to set such a threshold T that can separate clean classes and poisoned classes. Therefore, in our experiments, once the silhouette score of any clean class is higher than that of the attacked target classes, the proposed two attacks can successfully evade the detection of the AC method.

We perform the AC detection method to evaluate the proposed backdoor attacks with the following steps [21]. First, the One-to-N and N-to-One attacks are implemented on LeNet-5 [17] and VGG-16 [18] models. Second, for each class of MNIST [15] and CIFAR-10 [16] datasets, the activation of each training image on the last hidden layer is obtained. Third, each activation is flattened into a 1D vector, and ICA (Independent Component Analysis) [31] algorithm is used to reduce its dimension to obtain the first 10 components [21]. Finally, the activations of these training images that from the same class are separately clustered with k-means method and outputs a silhouette score [21].

Fig. 10 shows the detection results of AC method against the proposed two backdoor attacks. It is shown that, the AC method is ineffective to detect the proposed two types of backdoor attacks. For One-to-N attack, the accuracy that AC method detects the backdoor attacks on MNIST and CIFAR-10 dataset is only 40% and 30%, respectively. The reason is that, the source classes of these injected backdoor instances are multiple, which cause the images belonging to the target class t in poisoned training set come from multiple classes. When AC method exploits the 2-means clustering algorithm to cluster those images from the class t, it will output a low silhouette score which is lower than the threshold T. As a result, the AC method cannot detect the proposed One-to-N attacks. For N-to-One attack, the accuracy of detecting such backdoor attacks on MNIST and CIFAR-10 datasets is as low as 0%, which means the AC method fails to detect such attacks. The reason is that, the number of injected backdoor instances is quite small (only 400 backdoor instances in total), therefore, the DNN model cannot "learn" these single triggers. As discussed in Section 3.5, these injected backdoor instances that contain a single backdoor will not be classified as the target class t. In other words, in the backdoored training set, the images belonging to the target class t will not contain any injected backdoor instances. Therefore, when AC method exploits the clustering algorithm to analyze the activations of those images from the target class t, it cannot determine whether the training images of a class has been injected with backdoor instances.

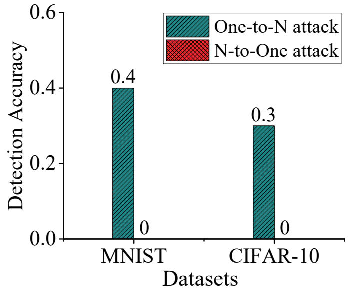


Fig. 10: The detection results of Activation Clustering method against the proposed two backdoor attacks.

5.2 Robust against Neural Cleanse Method

For each class, the NC method [14] uses gradient descent based approach to find a possible trigger. In this way, for a DNN model with Q different labels, it will produce Q possible triggers. Then, the NC method [14] performs the outlier detection algorithm MAD (Median Absolute Deviation) [25] to detect all these Q possible triggers and outputs an anomaly index. If the anomaly index of any trigger is larger than 2, the NC method considers this DNN model has been infected [14], and the label corresponding to this trigger will be predicted as the target class. In summary, the NC method [14] is not only able to determine whether a DNN model is infected, but can also identify which class is the target class of the backdoor attacks.

One-to-N attack. Table 9 shows the detection results of Neural Cleanse (NC) method against the proposed two backdoor attacks, which includes three aspects: 1) the accuracy of detecting the backdoor attacks; 2) the accuracy of identifying the target classes of backdoor attacks; 3) the accuracy of reversing the true triggers of target classes. First, for the proposed One-to-N attack ($N = 4$) on MNIST dataset, the accuracy that NC method detects the backdoor attack is only 20%. However, even if the NC method detects the backdoor attacks against the LeNet-5 model, it cannot determine all the N targets of the proposed One-to-N attack. In our experiments, we implement the One-to-N attack on MNIST dataset for ten times (four different targets each time). Among these 10 attacks, the accuracy that NC detects all the N targets is 0%. What's worse, the NC method even incorrectly identifies some clean classes as the attacked target classes. Take the One-to-N ($N = 4$) attack combination (“9, 1, 0, 4”) on the MNIST dataset as an example, where its four attack targets are set to be ‘9’, ‘1’, ‘0’, and ‘4’, respectively. The NC method incorrectly identifies ‘5’, ‘7’, ‘0’ and ‘4’ as the target classes. The attackers can still attack the other two target classes. Second, for the One-to-N backdoor attack on CIFAR-10 dataset, the NC method completely fails and the detection accuracy is 0%. It is also shown in Table 9 that, the accuracy that NC method reverses these true triggers that embedded into LeNet-5 and VGG-16 models are all 0%. This means the proposed One-to-N attack is still robust and effective under the state-of-the-art defenses, which will bring great challenges and threats to deep learning models.

TABLE 9: The Detection Results of Neural Cleanse Method against the Two Proposed Backdoor Attacks.

Dataset	Proposed attacks	Accuracy		
		Detecting the attacks	Identifying target class	Reversing true trigger
MNIST	One-to-N	20%	0%	0%
	N-to-One	100%	80%	0%
CIFAR-10	One-to-N	0%	0%	0%
	N-to-One	0%	0%	0%

Note that, the NC method will output a reversed trigger for each class label of a DNN model (*i.e.*, 10 reversed candidate triggers for MNIST and CIFAR-10, respectively). In the case of $N = 4$, the One-to-N attack has four different attacked labels at each attack combination. The reversed One-to-N triggers of attack combination “6, 8, 2, 0” (MNIST) and “Airplane, Ship, Car, Cat” (CIFAR-10) are presented in Fig. 11. First, for One-to-N attack, we use the same backdoor of different intensities to attack the N targets. However, as shown in Fig. 11, for each attack combination (e.g., “6, 8, 2, 0”), the reversed triggers of its four target labels are quite different from each other. What's worse, the reversed triggers of these attacked labels (such as ’6’ and ’ship’) even look like the triggers of these clean labels. Second, the reversed candidate triggers of these attacked labels (“6, 8, 2, 0” for MNIST and “Airplane, Ship, Car, Cat” for CIFAR-10) are completely different from these true backdoors ($s_c(i,j)$ and s_q) that used to implement the One-to-N attacks on LeNet-5 and VGG-16 models. In other words, the NC method fails to detect the backdoors that embedded by the proposed One-to-N attack.

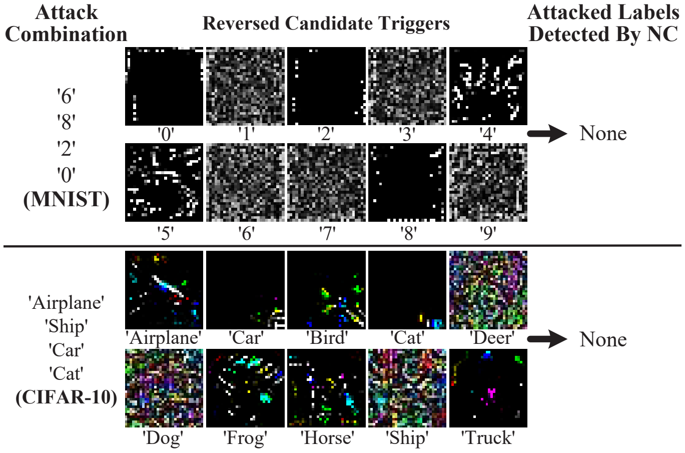


Fig. 11: The reversed candidate triggers for each class in the One-to-N attack on MNIST and CIFAR-10 datasets, and the attacked labels detected by the NC method.

N-to-One attack. The NC method [14] is currently a state-of-the-art and general backdoor detection method, which can detect most existing One-to-One backdoor attacks, such as BadNets [7] and Trojan Attack [12]. In our experiment, for MNIST dataset, the accuracy that the NC method detects the target classes of N-to-One attacks is 80%. This is because, the MNIST images are gray-scale images, which makes the slightly modifications on a clean image will be obvious. As a result, these single triggers (s_{11}, s_{12}, s_{13} and s_{14}) may be detected by the NC method, and the target classes of N-to-One attacks are determined. However, the NC cannot obtain the "complete" backdoor that used to launch the attacks. The accuracy that NC method reverses the true trigger of the proposed N-to-One attack on MNIST dataset is 0%. Fig. 12 shows the reversed candidate triggers in the N-to-One attack on MNIST dataset, where the target label is '2'. It is shown that, the reversed trigger of attacked label '2' is incorrect, which is completely different from the "true" backdoor $s_{11}\&s_{12}\&s_{13}\&s_{14}$ (as shown in Fig. 5(a)). In this way, the threats of N-to-One attacks still exists, where an adversary can utilize the "complete" trigger to implement the backdoor attacks without being noticed. Further, the NC method even incorrectly identifies the clean labels '3' and '8' as the targets of our N-to-One attack.

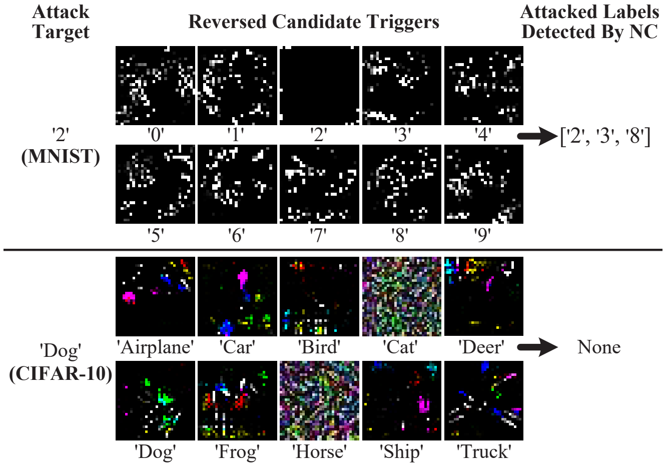


Fig. 12: The reversed candidate triggers for each class in the N-to-One attack on MNIST and CIFAR-10 datasets, and the attacked labels detected by the NC method.

For CIFAR-10 dataset, the NC method completely fails to detect the N-to-One attack, where the accuracy of detecting the backdoor attack, the accuracy of identifying the target class, and the accuracy of reversing the true trigger are all 0%. The reasons are as follows. First, the CIFAR-10 images are colorful, which makes it difficult to detect these backdoors that with the intensity of -100. Second, compared to the LeNet-5 model which has only 5 layers, the VGG-16 model has 16 layers with over ten millions of parameters, which makes the NC method hard to reverse the entire network to obtain the embedded trigger. As shown in Fig. 12, the NC method cannot reverse the "complete" backdoor of N-to-One attack on CIFAR-10 dataset. The reversed trigger of attacked label 'Dog' is not the backdoor $s_{21}\&s_{22}\&s_{23}\&s_{24}$ (as shown in Fig. 5(b)) that the N-to-One attack embeds into the VGG-16 model, and it looks like the triggers of these clean labels (such as 'Frog' and 'Car'). Moreover, the reversed trigger is even not part of the "complete" backdoor. As a result, the attacker can still launch the backdoor attacks against VGG-16 model using the "true" trigger.

6 Conclusions

This paper proposes two novel types of advanced backdoor attacks, One-to-N and N-to-One attacks, against deep learning models. This is the first work proposing multi-trigger and multi-target backdoors. Compared with existing One-to-One backdoor attacks, the proposed two backdoor attacks can be applied under the weak attack model, and are more difficult to be detected by these state-of-the-art defense techniques (e.g., AC and NC methods). Experimental results show that, the proposed two backdoor attacks can achieve high attack success rate on two image classification datasets (up to 100% in MNIST and 92.22% in CIFAR-10) and one large and realistic face image dataset (up to 94% in YouTube Aligned Face). Meantime, the normal working performances of these DNN models (LeNet-5, VGG-16 and VGGFace) will not be affected. This work reveals two more insidious backdoor attacks, which pose new threats to deep learning models and new challenges to existing defenses. In the future, we will explore effective countermeasures against these backdoor attacks.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61602241, and No. 62022041).

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[2] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep neural networks with adaptive noise reduction," *IEEE Trans. Depend. Secur. Comput., Early Access*, pp. 1–14, Oct. 2018.

[3] L. Lu, L. Liu, M. J. Hussain, and Y. Liu, "I sense you by breath: Speaker recognition via breath biometrics," *IEEE Trans. Depend. Secur. Comput., Early Access*, pp. 1–15, Oct. 2017.

[4] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2722–2730.

[5] A. Demontis, M. Melis, B. Biggio, D. Maiorca, D. Arp, K. Rieck, I. Corona, G. Giacinto, and F. Roli, "Yes, machine learning can be more secure! A case study on android malware detection," *IEEE Trans. Depend. Secur. Comput.*, vol. 16, no. 4, pp. 711–724, Aug. 2019.

[6] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, "MADAM: Effective and efficient behavior-based android malware detection and prevention," *IEEE Trans. Depend. Secur. Comput.*, vol. 15, no. 1, pp. 83–97, Mar. 2016.

[7] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, Apr. 2019.

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

[9] C. Liao, H. Zhong, A. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," 2018. [Online]. Available: https://arxiv.org/abs/1808.10307

[10] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in CNNs by training set corruption without label poisoning," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 101–105.

[11] M. Zou, Y. Shi, C. Wang, F. Li, W. Song, and Y. Wang, "PoTrojan: Powerful neural-level Trojan designs in deep learning models," 2018. [Online]. Available: https://arxiv.org/abs/1802.03043

[12] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, “Trojaning attack on neural networks,” in *Proc. 25th Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 12–20.

[13] G. Lovisotto, S. Eberz, and I. Martinovic, “Biometric backdoors: A poisoning attack against unsupervised template updating,” 2019. [Online]. Available: https://arxiv.org/abs/1905.09162

[14] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, “Neural cleanse: Identifying and mitigating backdoor attacks in neural networks,” *IEEE Symp. Secur. Priv.*, vol. 1, pp. 513–529, 2019.

[15] L. Deng, “The MNIST database of handwritten digit images for machine learning research,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, May. 2012.

[16] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” University of Toronto, Tech. Rep., 2009.

[17] Google, “A guide to TF layers: Building a convolutional neural network,” 2018. [Online]. Available: https://www.tensorflow.org/tutorials/layers

[18] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[19] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *Proceedings of the British Machine Vision Conference*, 2015, pp. 1–12.

[20] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 529–534.

[21] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, “Detecting backdoor attacks on deep neural networks by activation clustering,” in *Proc. 33th AAAI Conf. Artif. Intell.*, 2019, pp. 1–10.

[22] C. Yang, Q. Wu, H. Li, and Y. Chen, “Generative poisoning attack method against neural networks,” 2017. [Online]. Available: https://arxiv.org/abs/1703.01340

[23] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, “Towards poisoning of deep learning algorithms with back-gradient optimization,” in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 27–38.

[24] K. Liu, B. Dolan-Gavitt, and S. Garg, “Fine-pruning: Defending against backdooring attacks on deep neural networks,” in *Proc. 21st Int. Symp. Attacks Intrusions Def.*, 2018, pp. 273–294.

[25] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, “Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median,” *J. Exp. Soc. psychol.*, vol. 49, no. 4, pp. 764–766, 2013.

[26] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, “STRIP: A defence against Trojan attacks on deep neural networks,” in *Proc. 35th Annual Comput. Secur. Appl. Conf.*, 2019, pp. 113–125.

[27] D. Tang, X. Wang, H. Tang, and K. Zhang, “Demon in the variant: Statistical analysis of DNNs for robust backdoor contamination detection,” 2019. [Online]. Available: https://arxiv.org/abs/1908.00686

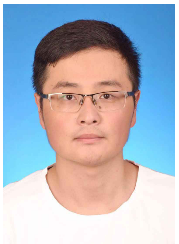
[28] T. K. Moon, “The expectation-maximization algorithm,” *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[29] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, “Februus: Input purification defense against Trojan attacks on deep neural network systems,” 2019. [Online]. Available: https://arxiv.org/abs/1908.03369

[30] E. Wenger, J. Passananti, Y. Yao, H. Zheng, and B. Y. Zhao, “Backdoor attacks on facial recognition in the physical world,” 2020. [Online]. Available: https://arxiv.org/abs/2006.14580

[31] A. Hyvärinen and E. Oja, “Independent component analysis: Algorithms and applications,” *Neural Netw.*, vol. 13, no. 4, pp. 411–430, 2000.

Mingfu Xue (S’11–M’14) received the Ph.D. degree in the Information and Communication Engineering from Southeast University, Nanjing, China, in 2014. From July 2011 to July 2012, he is a research intern in Nanyang Technological University, Singapore. He is currently an Associate Professor 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, hardware security, and hardware Trojan detection. He is the programme chair of the third Chinese Symposium on Hardware Security. He has also been a technical program committee member for over 10 international conferences. He is a committee member of the Chinese Artificial Intelligence and Security Professional Committee. He is also an executive committee member of ACM Nanjing Chapter, and a committee member of Computer Network and Distributed Computing Specialized Committee of Jiangsu Province. He is a member of IEEE, ACM, IEICE, CCF and CAAI. From 2014 to 2019, he has presided 10 research projects or fundings. He has published over 30 papers in security related journals and international conferences, and won the best paper award in ICCCS2015.



Can He received the B.S. degree in computer science and technology from Anhui University of Technology, in 2017. 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.

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 60 papers in security related journals and conferences.

Weiqiang Liu (M’12–SM’15) 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 an Full Professor. He was a Research Fellow in the Institute of Electronics, Communications and Information Technology (ECIT) at QUB from Aug. 2012 to Nov. 2013. He has published one research book by Artech House and over 110 leading journal and conference papers. One of his papers is the Feature Paper of IEEE Transactions on Computers (TC) in the 2017 December issue. His papers are best paper candidates of IEEE ISCAS 2011 and ACM GLSVLSI 2015. He serves as an Associate Editor of IEEE Transactions on Computers, IEEE Transactions on Circuits and Systems I: Regular Papers, and IEEE Transactions on Emerging Topics in Computing, the Guest Editors of Proceedings of the IEEE and IEEE TETC (two special issues), an Steering Committee Member of IEEE Transactions on Multi-Scale Computing Systems (TMSCS). He is the program co-chair of IEEE ARITH 2020. He has been a technical program committee member for several international conferences. His research interests include approximate computing, computer arithmetic, hardware security and VLSI design for digital signal processing and cryptography.