This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3161477, IEEE Transactions on Dependable and Secure Computing

Enhancing Backdoor Attacks with Multi-Level MMD Regularization

Pengfei Xia, Hongjing Niu, Ziqiang Li, and Bin Li, *Member, IEEE*

Abstract—While Deep Neural Networks (DNNs) excel in many tasks, the huge training resources they require become an obstacle for practitioners to develop their own models. It has become common to collect data from the Internet or hire a third party to train models. Unfortunately, recent studies have shown that these operations provide a viable pathway for maliciously injecting hidden backdoors into DNNs. Several defense methods have been developed to detect malicious samples, with the common assumption that the latent representations of benign and malicious samples extracted by the infected model exhibit different distributions. However, it is still an open question whether this assumption holds up. In this paper, we investigate such differences thoroughly via answering three questions: 1) What are the characteristics of the distributional differences? 2) How can they be effectively reduced? 3) What impact does this reduction have on difference-based defense methods? First, the distributional differences of multi-level representations on the regularly trained backdoored models are verified to be significant by adopting Maximum Mean Discrepancy (MMD), Energy Distance (ED), and Sliced Wasserstein Distance (SWD) as the metrics. Then, ML-MMDR, a difference reduction method that adds multi-level MMD regularization into the loss, is proposed, and its effectiveness is testified on three typical difference-based defense methods. Across all the experimental settings, the F1 scores of these methods drop from 90%-100% on the regularly trained backdoored models to 60%-70% on the models trained with ML-MMDR. These results indicate that the proposed MMD regularization can enhance the stealthiness of existing backdoor attack methods. The prototype code of our method is now available at https://github.com/xpf/Multi-Level-MMD-Regularization.

Index Terms—Deep Neural Networks, Backdoor Attacks, Distributional Differences, Maximum Mean Discrepancy.

1 Introduction

In the past years, Deep Neural Networks (DNNs) have shown impressive achievements in computer vision [1], [2], [3], [4], [5], natural language processing [6], [7], [8], and some other fields [9], [10], [11]. It is widely believed that the success of DNNs is closely related to their large-scale models, consumption of a huge amount of training data, and computational power. For example, GPT-3 [12], a deep model demonstrated to be effective in various tasks, comprises 175 billion parameters and is pretrained on 45 TB of text data. A training cycle for this model on a Tesla V100 GPU would require $4.6 million and 355 years[1]. The superior performance of DNNs usually comes at the cost of massive time and economic resources.

To save on training costs, it has become common for users and companies to collect data from the Internet, use pretrained parameters, or hire a third party to train models. Unfortunately, these operations pose security risks due to the possible presence of malicious sources and parties. One of the major threats is dubbed as *backdoor attacks* or *Trojan attacks* [15], [16], [17], [18], where a hidden backdoor is injected into the victim model during the training phase and can be activated by a predefined trigger. The infected model would exhibit duel characteristics. When the backdoor is not activated, it behaves as normal as a benign model. But once triggered, the predictions are forced to an attacker-specific target. Backdoor attacks seriously threaten the deployment of DNNs in security-sensitive scenarios, such as face recognition systems [15] and autonomous vehicles [16].

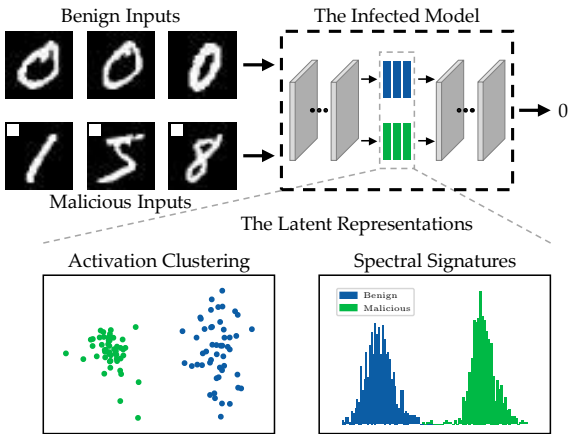

Fig. 1. An illustrative example of difference-based defense methods. The infected model is trained to classify all inputs with a trigger, i.e., the white square in the upper left corner of the images, as the number 0. Some defense methods, such as activation clustering [13] and spectral signatures [14], use the latent representations extracted by the infected model to distinguish malicious samples from benign ones.

Several defense methods [13], [14], [19], [20] have been presented by distinguishing malicious samples from benign ones according to their latent representations extracted by the infected model. For example, Chen et al. [13] proposed activation clustering, which is based on the observation that the projected activations of the last hidden layer are

- *P. Xia, Z. Li and B. Li are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China.*
  E-mail: {xpengfei, iceli}@mail.ustc.edu.cn, binli@ustc.edu.cn.
- *H. Niu is with the Department of Automation, University of Science and Technology of China, Hefei, China.*
  E-mail: sasori@mail.ustc.edu.cn.

1. https://lambdalabs.com/blog/demystifying-gpt-3/

divided into two distinct clusters. Tran et al. [14] proposed calculating an outlier score for each sample by performing singular value decomposition on the representations. The inputs with the top scores would be removed as malicious samples. An illustrative example is shown in Fig. 1.

The previously presented defense methods all rely on a common assumption: the latent representations of benign and malicious samples exhibit different distributions. A naturally arisen question is whether this assumption would always be true. There is a shortage of a comprehensive investigation on this issue. Some reduction methods [21], [22], [23] have been proposed to reduce the distributional difference in the latent space. However, in Section 5 and Section 6, we will show that these methods are not sufficient to provide a solid solution.

In this paper, we focus on this commonly adopted assumption and investigate the following questions:

- What are the characteristics of the distributional differences between benign and malicious samples in the latent space extracted by the infected model?
- How can they be effectively reduced when conducting backdoor attacks?
- What impact does this reduction have on difference-based defense methods?

The main contributions of this paper include:

- Three typical metrics, i.e., Maximum Mean Discrepancy (MMD) [24], Energy Distance (ED) [25], and Sliced Wasserstein Distance (SWD) [26], are introduced to explicitly quantify the distributional differences in the latent spaces. The differences between benign and malicious samples in multi-level representations are verified to be large. This motivates us to take the multiple levels of features into consideration when designing the reduction method, other than those of the last hidden layer.
- A new method, ML-MMDR, is proposed by introducing **M**ulti-**L**evel **MMD R**egularization into the loss when training a backdoored model. The experimental results indicate that the proposed method can fully reduce the differences without compromising the attack strength.
- The effectiveness of ML-MMDR is testified on three typical difference-based defense methods. The experimental results show that the performance of these methods is substantially degraded by the proposed regularization. This illustrates that ML-MMDR can enhance existing attack methods to escape detection.

The rest of this paper is organized as follows. In Section 2, the preliminaries are briefly introduced. Section 3 provides the setup used in the following three sections. Section 4, Section 5, and Section 6 provide our detailed works addressing the three mentioned questions. Some open issues are discussed in Section 7, and the related works are reviewed in Section 8. Section 9 concludes this paper.

2 Preliminaries

2.1 Deep Neural Networks

A DNN is composed of multiple layers and can be defined as $f_\theta = f_1 \circ f_2 \circ \cdots \circ f_m$, where θ denotes the parameters of the model, f_1, f_2, $\cdots$, f_{m-1} denote the $m-1$ hidden layers, and f_m denotes the output layer. For convenience, let $z_i = f_1 \circ f_2 \circ \cdots \circ f_i$ denote the structure from the input to the i-th hidden layer and $z_i(x)$ denote the extracted representation of the input x. Given a training set $D = \{(x, y)\}$, the procedure of training a DNN can be formulated as:

$$\min_\theta \frac{1}{|D|} \sum_{(x,y)\in D} L(f_\theta(x), y), \tag{1}$$

where (x, y) denote the input and its ground-truth label, and L denotes the loss function. The trained model is expected to perform well on a test set T, and $D \cap T = \varnothing$ is required.

2.2 Backdoor Attacks

In this paper, the scenario of injecting a backdoor into a DNN by dataset poisoning [15], [16], [17], [27] is considered. Let U and V denote the malicious training and test sets. The generation of malicious samples is associated with a target class t, a trigger b, and a fusion function G. For a benign pair (x, y), the corresponding malicious pair is (x', t), where $x' = G(x, b)$. To train a DNN with a backdoor, one can optimize the equation:

$$\min_\theta \frac{1}{|D|} \sum_{(x,y)\in D} L(f_\theta(x), y) + \frac{1}{|U|} \sum_{(x',t)\in U} L(f_\theta(x'), t), \tag{2}$$

and we suppose the trained model can generalize to the benign and malicious test sets T and V. The ratio of malicious sample volume to benign sample volume in training data, $r = |U|/|D|$, is an important hyperparameter.

2.3 Measurement of the Distributional Difference

Let $X \sim p$ and $Y \sim q$ denote two sample sets drawn from the distributions p and q. A statistical distance takes the two sets as its input and returns a real number as the approximate distance between p and q. In this paper, MMD, ED, SWD are adopted to measure the distributional difference.

Maximum Mean Discrepancy. Gretton et al. [24] introduced a kernel-based metric, MMD, to quantify the distance between two distributions. It can be calculated as:

$$\begin{aligned}\mathrm{MMD}^2(X, Y) =& \frac{1}{m^2} \sum_{i,j=1}^{m} k(x_i, x_j) + \frac{1}{n^2} \sum_{i,j=1}^{n} k(y_i, y_j) \\ &- \frac{2}{mn} \sum_{i,j=1}^{m,n} k(x_i, y_j)\end{aligned}, \tag{3}$$

where m and n denote the sizes of X and Y, respectively. k is a kernel function and $k(x, y) = \langle \phi(x), \phi(y) \rangle$, where ϕ denotes a feature mapping.

Energy Distance. Szekely and Rizzo [25] first introduced this metric. It is a specific case of MMD, where no kernel is applied. The equation is:

$$\begin{aligned}\mathrm{ED}^2(X, Y) =& \frac{1}{m^2} \sum_{i,j=1}^{m} ||x_i - x_j|| + \frac{1}{n^2} \sum_{i,j=1}^{n} ||y_i - y_j|| \\ &- \frac{2}{mn} \sum_{i,j=1}^{m,n} ||x_i - y_j||\end{aligned}. \tag{4}$$

Sliced Wasserstein Distance. SWD is a potential alternative to Wasserstein distance [28], which can be be approximated more easily. The underlying idea is to decompose high-dimensional data into one-dimensional data via random linear projection. SWD with l_1 cost can be computed as:

$$\mathrm{SWD}_1(X,Y) = \mathop{\mathbb{E}}_{R}\left[\min_{\sigma\in S_m}\sum_{i=1}^{m}|R(x_{\sigma_i}) - R(y_i)|\right], \quad (5)$$

where R denotes a linear projection, and $\sigma \in S_m$ denotes a possible sample sequence [26].

3 Setup

3.1 Threat Model

Our threat model considers that the user needs to use a DNN trained in an untrusted environment. This model is prevalent among companies, such as car manufacturers and autonomous driving solution providers. We assume that the attacker has complete control over the training data and the training procedure. The defender's goal is to distinguish malicious samples from benign ones. The defender does not have any prior knowledge about the attack but has access to the parameters of the DNN and keeps a few (about 200) benign validation data.

3.2 Attack Methods

Four backdoor attack methods are considered, including patch-based attack (Patched) [16], blending-based attack (Blended) [15], SIG [27], and warping-based attack (Warped) [29]. The main differences between these methods are the trigger b and the fusion function G. The forms of generating malicious samples are shown in TABLE 1, and some examples are shown in Fig. 2. The other settings are kept the same for these methods. The target t is set to the 0th category, and the poisoning data ratio, r, is set to 0.1.

TABLE 1
Forms of generating malicious samples. m: a 2D mask. α: a hyperparameter between 0 to 1. $\mathcal{W}$: the warp function.

Attack	Fusion Function $x' = G(x, b)$
Patched	$x' = (1-m)\odot x + m \odot b$
Blended	$x' = (1-\alpha)\cdot x + \alpha \cdot b$
SIG	$x' = x + b$
Warped	$x' = \mathcal{W}(x, b)$

3.3 Defense Methods

Four defense methods are adopted to test the effect of attack methods, three of which are difference-based detection methods. Note that some of these methods were proposed for filtering malicious samples in the training phase. We believe that it is also reasonable to apply them in the test phase, and the experimental results show that they perform well. Besides, the original detection methods mainly utilize the outputs of the last hidden layer. We extend them to multi-level representations for a more thorough analysis.

Activation Clustering. Chen et al. [13] proposed activation clustering to detect malicious samples, which does not require verified data. The author first observed that the features of benign and malicious inputs, which are classified into the same category by the backdoored model, visually show two distinct clusters. Then, they proposed filtering malicious samples with three steps: 1) Perform independent component analysis on the latent representations to reduce the dimensionality. 2) Use k-means to cluster the data represented by the reduced features into two clusters. 3) Analyze whether each cluster is benign or malicious by exclusionary reclassification, relative size comparison, or silhouette score.

Spectral Signatures. Tran et al. [14] identified a characteristic of backdoor attacks called spectral signatures. A spectral signature is a detectable trace in the spectrum of the representations' covariance that these attacks tend to leave behind. They then proposed an algorithm to show that one could use this trace to remove malicious samples with three steps: 1) Perform singular vector decomposition on the centered features. 2) Calculate the outlier scores with the top right singular vector. 3) Remove data with the top-k scores.

Subspace Reconstruction. The core idea of subspace reconstruction for backdoor detection is to construct a subspace learned from a small number of benign representations and assume that projecting malicious representations into this space would cause considerable information to be lost. As a result, the reconstruction loss of these features is higher than that of benign features. Javaheripi et al. [31] first introduced this idea into their Trojan detection framework.

Neural Cleanse. Wang et al. [32] proposed neural cleanse, a detection and mitigation system for backdoor attacks. In the detection step, the method first uses reverse engineering to obtain potential triggers towards every category and then determines the final synthetic trigger according to its l_1 norm. The authors introduced three techniques, i.e., input filtering, neuron pruning, and unlearning, in the mitigation step to remove the backdoor with the synthetic trigger. Since input filtering is similar to the three detection methods described above, unlearning needs to retrain the model, we apply neuron pruning in this paper.

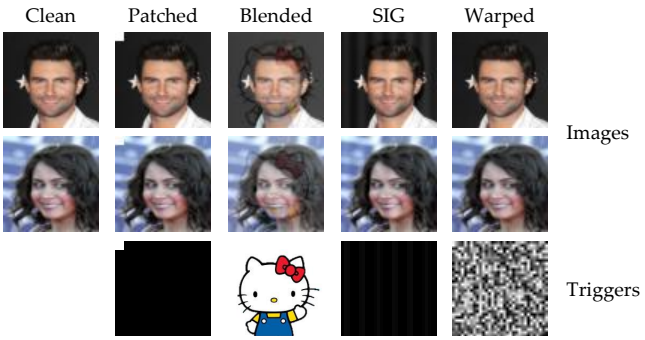

Fig. 2. Malicious examples from CelebA [30] generated by four backdoor attack methods.

3.4 Experimental Settings

Datasets. Two image datasets, CIFAR-10 [33] and CelebA [30], are selected, which are often used for backdoor learning tasks. For CelebA, following the configuration in [29], [34], we use the three most balanced attributes, i.e., "Heavy Makeup", "Mouth Slightly Open", and "Smiling", to create eight categories for image classification.

DNN Architectures. Four DNN architectures, VGG-11 (V-11) [35], VGG-16 (V-16) [35], ResNet-18 (R-18) [36], and PreActResNet-18 (P-18) [37], are used in our experiments. Since the activations of the hidden layers need to be extracted as the latent representations, we define s_1, s_2, and s_3 to represent three locations in the network structure. $z_{s_1}(x)$, $z_{s_2}(x)$, and $z_{s_3}(x)$ denote the extracted features of x from the corresponding levels. Specifically, we set s_1, s_2, and s_3 to the 14th, 21st, and 28th layers for V-11, the 23rd, 33rd, and 43rd layers for V-16, and the 27th, 39th, and 51st layers for R-18 and P-18. More details of the DNN architectures can be seen in Appendix A.

Evaluation Metrics. Benign Accuracy (BA) and Attack Success Rate (ASR) are adopted as the metrics for measuring the performance of an infected model on the test sets T and V. For a difference-based defense method, we use F1 score to evaluate its performance.

Implementation Details. We adopt stochastic gradient descent with a momentum of 0.9 and a weight decay of 5e-4 as the optimizer for all experiments. The batch size is set to 256, and the total training duration is 100 epochs for CIFAR-10 and 40 epochs for CelebA. The initial learning rate is set to 0.01 and it is dropped by 10 after 50 and 70 epochs for CIFAR-10 and 20 and 30 epochs for CelebA. All images are resized to 32×32 and normalized between 0 and 1. Our code[2] is implemented with PyTorch [38].

4 Distributional Differences between Benign and Malicious Samples

What are the characteristics of the distributional differences between benign and malicious samples in the latent spaces learned by an infected model? In this section, we attempt to answer this question. We first need to quantify the differences. Given an infected DNN f_θ, let $Z_{ij}^T = \{z_i(x)|(x,y) \in T \wedge f_\theta(x) == j\}$ and $Z_{ij}^V = \{z_i(x')|(x',t) \in V \wedge f_\theta(x') == j\}$ denote the extracted representations of benign and malicious samples, which are classified into the same category j by f_θ. A distributional difference refers to the dissimilarity between the distributions of Z_{ij}^T and Z_{ij}^V. In particular, since a well-trained backdoored model classifies almost all malicious samples to the target t, we mainly focus on the difference between Z_{it}^T and Z_{it}^V. Given that the latent representations are high-dimensional, and their distributions are difficult to estimate explicitly, in this paper, MMD, ED, and SWD are adopted to measure the distributional differences between benign and malicious samples.

To comprehensively analyze the differences, we calculate the three metrics on the test data at multiple levels, i.e., $\mathcal{M}(Z_{it}^T, Z_{it}^V)$, where $\mathcal{M}$ is one of the three measures and $i = \{s_1, s_2, s_3\}$. For MMD, we use a Gaussian mixture kernel. To give intuitive comparisons, for each type of attack, we provide the intra-class distance and the minimal inter-class distance among clean samples as two baselines, that is, $\mathcal{M}(Z_{it}^T, Z_{it}^T)$ and $\min_{j\neq t} \mathcal{M}(Z_{it}^T, Z_{ij}^T)$. To avoid being influenced by the numerical scale, we use the relative ordering (ascending) of $\mathcal{M}(Z_{it}^T, Z_{it}^V)$ in $\mathcal{M}(Z_{it}^T, Z_{it}^V)$, $\mathcal{M}(Z_{it}^T, Z_{it}^T)$, and $\min_{j\neq t} \mathcal{M}(Z_{it}^T, Z_{ij}^T)$ as the indicator. TABLE 2 presents Average Relative Ordering (ARO) over the three metrics, and the specific distance values are shown in Appendix B.

2. https://github.com/xpf/Multi-Level-MMD-Regularization

TABLE 2
ARO over MMD, ED and SWD at three levels.

Method	Model	CIFAR-10			CelebA		
		s_1	s_2	s_3	s_1	s_2	s_3
Patched	V-11	2.67	3.00	2.67	3.00	3.00	3.00
	V-16	2.67	3.00	2.33	3.00	3.00	3.00
	R-18	2.67	3.00	2.67	3.00	3.00	3.00
	P-18	2.67	2.67	2.00	3.00	3.00	3.00
Blended	V-11	2.67	2.67	2.67	3.00	3.00	3.00
	V-16	2.67	2.67	2.33	3.00	3.00	3.00
	R-18	2.67	2.67	2.00	3.00	3.00	3.00
	P-18	3.00	2.67	3.00	3.00	3.00	3.00
SIG	V-11	2.67	3.00	3.00	3.00	3.00	3.00
	V-16	3.00	3.00	2.33	3.00	3.00	3.00
	R-18	2.67	2.67	3.00	3.00	3.00	3.00
	P-18	3.00	3.00	2.00	3.00	3.00	3.00
Warped	V-11	3.00	3.00	3.00	3.00	3.00	3.00
	V-16	3.00	3.00	2.67	3.00	3.00	3.00
	R-18	2.67	2.67	2.33	3.00	3.00	3.00
	P-18	3.00	3.00	2.00	3.00	3.00	3.00

Some characteristics can be identified:

- The distributional differences of multi-level representations are all large. We observe that for all types of backdoor attacks, all DNN architectures, and all the two datasets, there are substantial increases in the MMD, ED, and SWD values compared to the corresponding intra-class values at all three levels (see in Appendix B). The results of ARO are more intuitive, where 69.8% of the values are 3.0 and 95.8% of the values are greater than 2.0. This means that in most cases, $\mathcal{M}(Z_{it}^T, Z_{it}^V)$ is greater than $\min_{j\neq t} \mathcal{M}(Z_{it}^T, Z_{ij}^T)$. Both of the two observations provide strong evidence of this characteristic.
- The distributional differences have some relationship to the dataset. In general, the infected models trained on CelebA have larger ARO values than the models trained on CIFAR-10.

The first characteristic is more inspiring. Since the distributional differences are all significant at multi-level representations, does it mean that the previous detection methods can be extended to multiple layers? We conduct experiments on this and find that this is indeed the case. The results are shown in Fig. 5, Fig. 6, and Fig. 7, where the F1 scores of the three difference-based detection methods at three levels on the regularly trained backdoored models are all high. This provides guidance for us to design a proper reduction method.

5 Method for Reducing the Distributional Differences

How to effectively reduce the distributional differences the infected models exhibit that defense methods could utilize? According to the analysis in Section 4, we know that the differences in multi-level representations are all large. Therefore, we propose a method named ML-MMDR to reduce the differences by adding multi-level MMD regularization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3161477, IEEE Transactions on Dependable and Secure Computing

Algorithm 1: Mini-batch ML-MMDR

Input: Benign training set D; malicious training set U; Learning rate η; Number of training epochs N; Constraint strength λ; Representation level set I; Target label t

Output: Model parameters θ

Initialize parameters θ;
Create the concatenated training set $C = D \cup U$;
for $n \leftarrow 1$ **to** N **do**
  Shuffle the training set C;
  for *each mini-batch* $(X, Y) \subset C$ **do**
    $(X_1, Y_1) = \{(x, y) \mid (x, y) \in (X, Y) \wedge (x, y) \in D\}$;
    $(X_2, Y_2) = \{(x, y) \mid (x, y) \in (X, Y) \wedge (x, y) \in U\}$;
    $(X_3, Y_3) = \{(x, y) \mid (x, y) \in (X, Y) \wedge (x, y) \in D \wedge y == t\}$;
    $L_1 \leftarrow \frac{1}{|X_1|} \cdot L(f_\theta(X_1), Y_1)$;
    $L_2 \leftarrow \frac{1}{|X_2|} \cdot L(f_\theta(X_2), Y_2)$;
    $L_3 \leftarrow \frac{1}{|I|} \cdot \sum_{i \in I} \mathrm{MMD}^2(Z_i^{X_2}, Z_i^{X_3})$;
    $L_t \leftarrow L_1 + L_2 + \lambda \cdot L_3$;
    $\theta \leftarrow \theta - \eta \cdot \nabla_\theta L_t$;
  end
end

to the loss during the training of a backdoored model. The proposed method can be formulated as:

$$\min_\theta \frac{1}{|D|} \sum_{(x,y)\in D} L(f_\theta(x), y) + \frac{1}{|U|} \sum_{(x',t)\in U} L(f_\theta(x'), t) + \lambda \cdot \frac{1}{|I|} \sum_{i \in I} \mathrm{MMD}^2(Z_{it}^D, Z_{it}^U), \quad (6)$$

where I is a set of levels, and λ is the hyperparameter that controls the constraint strength. The optimization goal consists of three parts, the first two of which are included in the regular backdoor training, as shown in 2. We add the third item to reduce the distributional differences. In practice, we adopt the mini-batch ML-MMDR, and the procedure is presented in Algorithm 1.

Two settings of I are considered, including $I = \{s_3\}$ and $I = \{s_1, s_2, s_3\}$, which correspond to constraining the features of the last hidden layer (SL-MMDR) and all three levels (ML-MMDR), respectively. We set $\lambda = \{0.0, 0.1, 0.2, 0.3\}$, where $\lambda = 0.0$ stands for Regular Backdoor Training (RBT). For convenience, we use the {A,B,C} models to denote the infected models trained on dataset A with architecture B and attack method C. The experimental results on the {P-18} models are shown in Fig. 3. The results on other architectures are similar and are shown in Appendix C.

Some observations are summarized as follows:

- The proposed ML-MMDR can significantly reduce the distributional differences at multi-level representations without harming the attack power. For example, when $\lambda = 0.3$, the average BA and ASR over the models trained with ML-MMDR decrease from 0.907 and 0.993 to 0.902 and 0.992, respectively, and the average ARO decreases from 2.792, 2.854, and 2.5 to 1.979, 1.979, and 2.0 at s_1, s_2, and s_3 for CIFAR-10, respectively, compared to the models trained with RBT ($\lambda = 0.0$). 2.0 is basically the minimum value that ARO can achieve.
- Constraining only the features of the last hidden layer, i.e., SL-MMDR, causes the features of other intermediate layers to have large differences still. For example, for $\lambda = 0.3$, the average ARO over the models trained with SL-MMDR is 2.833, 2.25, and 2.0 at s_1, s_2, and s_3 for CIFAR-10. It can be seen that the difference at the first level is even increased by SL-MMDR, which preserves the possibility of using these representations for defense. These results demonstrate that the methods of constraining only the last hidden layer [21], [22], [23] are not sufficient.

To show the differences more intuitively, the representations are visualized using the dimensionality reduction technique, as shown in Fig. 4. In the model trained with RBT, the representations of benign and malicious inputs can obviously be divided into two groups. For the model trained with ML-MMDR, these two types of representations almost overlap at all three levels. The first two levels of representations are still distinguishable for the model trained with SL-MMDR. The visualization results support our observations.

6 Empirical Study on the Effectiveness of the Method

Since the proposed ML-MMDR is feasible in reducing the distributional differences effectively without harming the attack intensity, in this section, we investigate what impact this reduction has on difference-based defense methods. Four typical methods, including Activation Clustering (AC) [13], Spectral Signatures (SS) [14], Subspace Reconstruction (SR) [31], and Neural Cleanse (NC) [32], are selected to testify our regularization. The first three methods are difference-based detection methods, and the last method is not. The backdoored models trained with RBT, ML-MMDR, and SL-MMDR are tested, where $\lambda = 0.3$.

6.1 Results of AC, SS, and SR

AC, SS, and SR are all difference-based detection methods that distinguish malicious samples from benign ones by using the latent representations extracted by the infected model. In our experiments, these methods are performed on the representations of three levels, i.e., s_1, s_2, and s_3. Because the effects of the three methods are affected by the number of samples N and the ratio of malicious inputs to benign inputs r', we set up four combinations for each defense method. The detection performance of AC, SS, and SR is measured by F1 score, and the results on the {P-18} models are shown in this section. The results on other architectures are similar and are presented in Appendix D. Below, we first show the results of each defense method separately before making a unified summary.

Activation Clustering. As suggested in [13], we first adopt independent component analysis to reduce the dimensionality of the latent representations of inputs to obtain the vectors of length 20. Then we use k-means to cluster the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3161477, IEEE Transactions on Dependable and Secure Computing

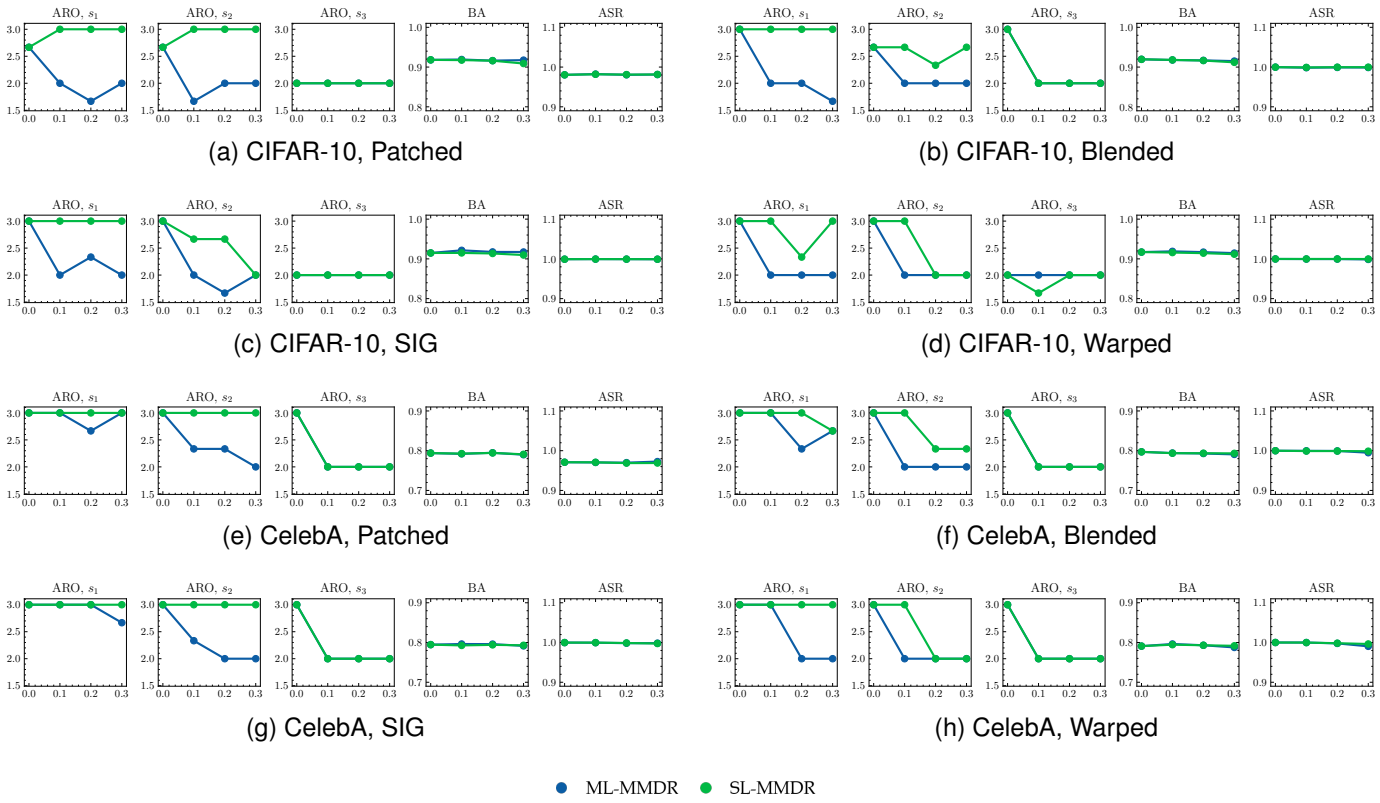

Fig. 3. Results of SL-MMDR and ML-MMDR on the {P-18} models with different λ. X-axis: the value of λ. Y-axis: the value of each indicator.

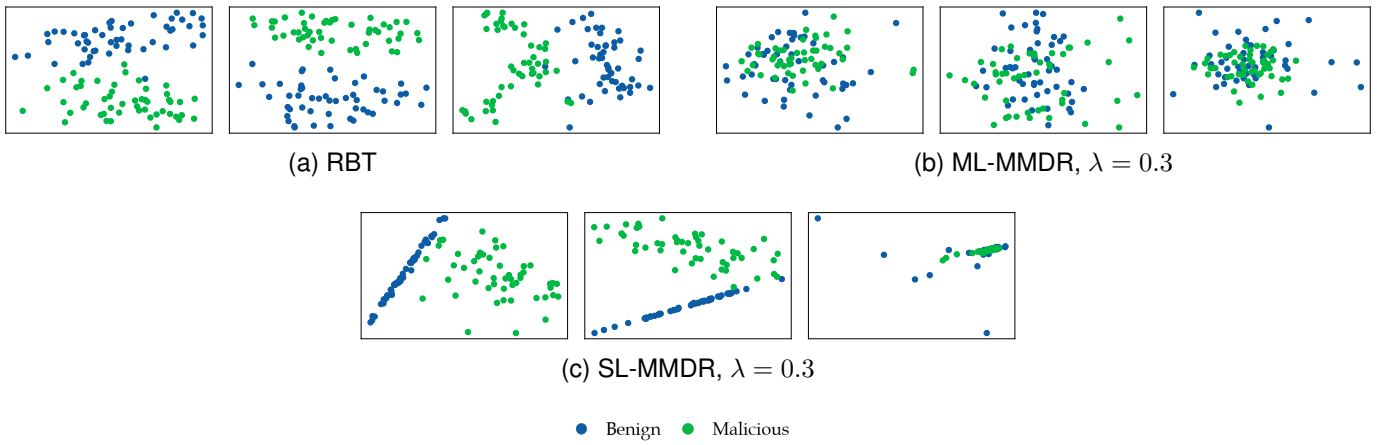

Fig. 4. Visualization examples of the features extracted from the {CIFAR-10, P-18, Patched} models.

reduced representations into two groups. The results of AC on the {P-18} models are shown in Fig. 5.

Spectral Signatures. Following the same steps in [14], we first take singular value decomposition of the latent representations, and then use the top right vector to compute an outlier score for each input. The results of SS on the {P-18} models are shown in Fig. 6.

Subspace Reconstruction. We choose 200 validated data and construct a subspace that can restore 90% of the energy of these samples. Subsequently, for the representation of each input, we perform the projection and reconstruction steps with the learned subspace and compute the l_2 norm as the reconstruction loss for that sample. The max F1 score for each model is calculated by adjusting the threshold, and the results are shown in Fig. 7.

Summary. The experimental results of the above three defense methods are similar and are summarized as follows:

- The representations at s_1 and s_2 levels can be used by these methods to detect malicious samples. For example, when $N = 100$ and $r' = 1.0$, the average F1 scores of AC at the three levels for RBT are 0.978, 0.991, and 0.948 on the {CIFAR-10, P-18} models, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3161477, IEEE Transactions on Dependable and Secure Computing

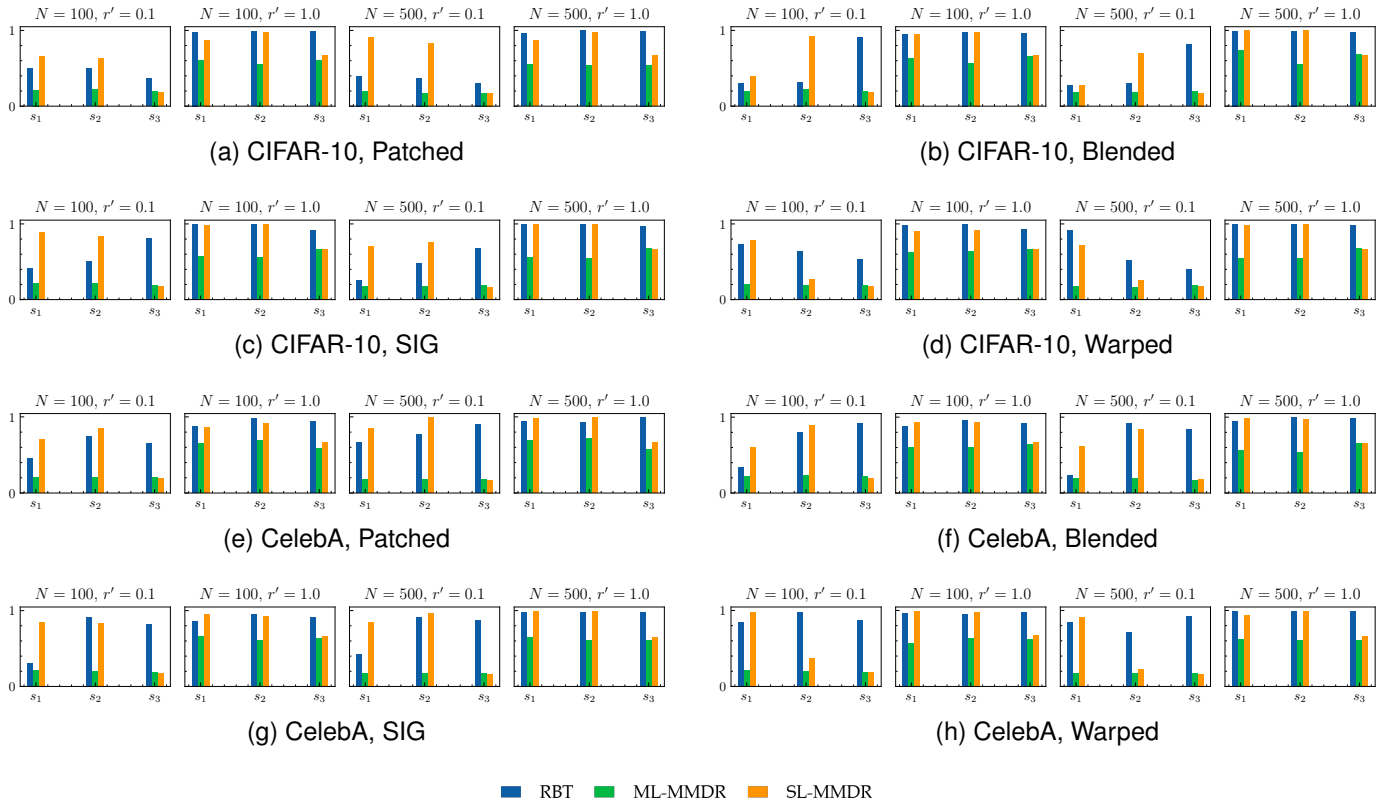

Fig. 5. F1 scores of AC on the {P-18} models. X-axis: the level of features. Y-axis: the value of $F1$.

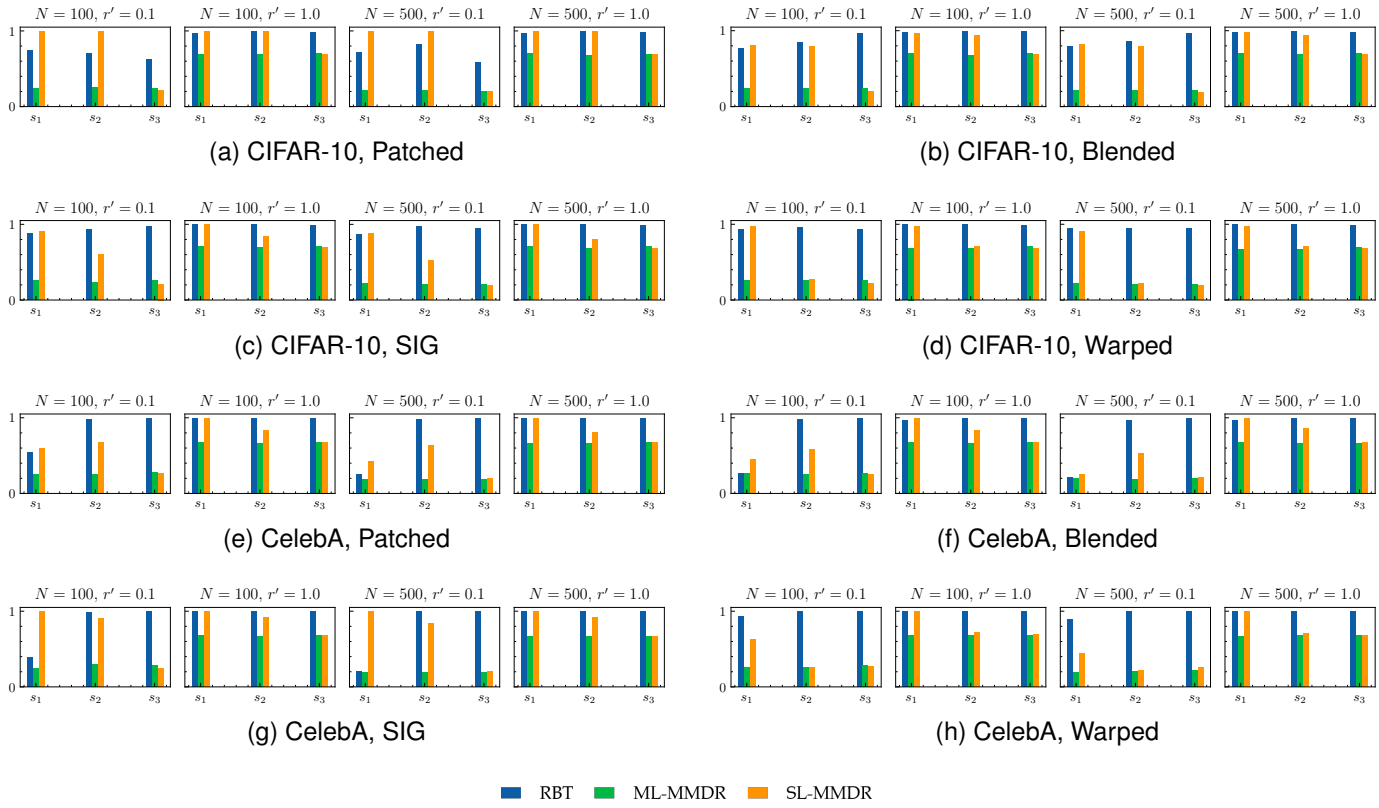

Fig. 6. F1 scores of SS on the {P-18} models. X-axis: the level of features. Y-axis: the value of $F1$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3161477, IEEE Transactions on Dependable and Secure Computing

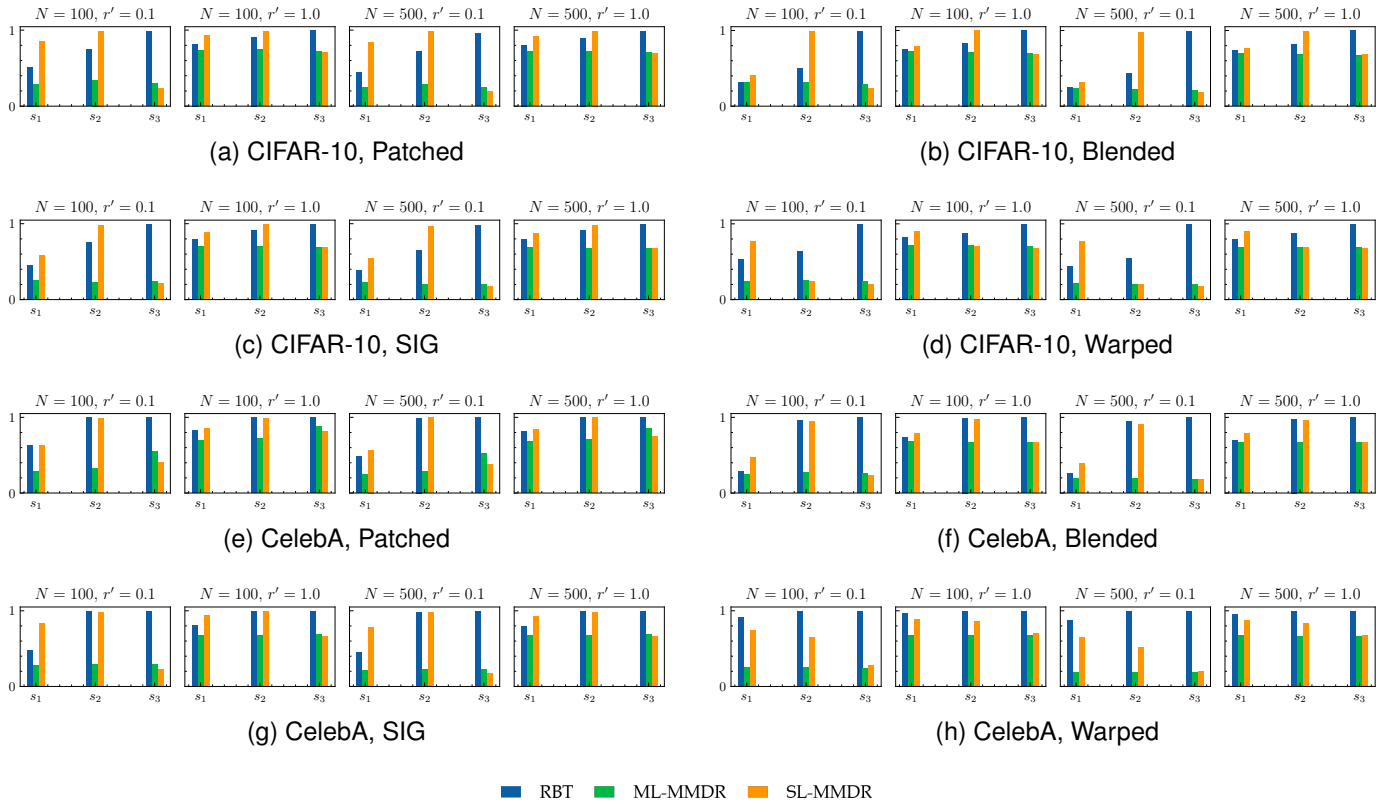

Fig. 7. F1 scores of SR on the {P-18} models. X-axis: the level of features. Y-axis: the value of $F1$.

0.898, 0.963, and 0.939 on the {CelebA, P-18} models. This confirms the conclusion in Section 4.

- The performance of difference-based detection methods is dropped on the infected models trained with ML-MMDR. For example, when $N = 500$ and $r' = 1.0$, the average F1 values of AC at the three levels for ML-MMDR drop from 0.986, 0.997, 0.979 to 0.598, 0.546, 0.645 on the {CIFAR-10, P-18} models, and from 0.965, 0.975, 0.988 to 0.630, 0.618, 0.613 on the {CelebA, P-18} models. The results of SS and SR are similar.
- The models trained with SL-MMDR can still be utilized by these defense methods to detect malicious samples. For example, when $N = 500$ and $r' = 1.0$, the average F1 scores of AC, SS, and SR at s_0 level for SL-MMDR are 0.962, 0.985, and 0.867 on the {CIFAR-10, P-18} models, and 0.977, 0.998, and 0.856 on the {CelebA, P-18} models. Sometimes, the values are higher than on the models trained with RBT. This indicates that constraining only the activations of the last hidden layer, which is adopted in the previous reduction methods [21], [22], [23], is not enough for bypassing backdoor detection algorithms.

6.2 Results of NC

NC consists of two steps, i.e., trigger synthesis and neuron pruning, to complete the detection and mitigation of the backdoor. We conduct experiments on the above two steps separately.

Trigger Synthesis. Wang et al. [32] defined a generic form of backdoor sample generation with a 3D trigger pattern and a 2D location mask. Then they formulated an optimization problem to reverse a pattern and a location for each category. The final synthetic trigger is selected by calculating the l_1 norm of all candidate masks. Because the form defined above is mainly used to reverse the patching-based attack [16], we conduct experiments on the {Patched} models. The results are shown in Fig. 8, and some of the reversed triggers on the {CIFAR-10, Patched} models are shown in Fig. 9.

As see, whether it is from the distinguishability of the l_1 norm or from the intuitive feeling, the infected models trained with ML-MMDR can still be used to reverse the triggers. This indicates that the trigger synthesis relies on a different principle from the difference-based methods to defend against backdoor attacks. However, this method seems to be easily affected by the dataset, such as the performance on the {CelebA, Patched} models is not as good as on the {CIFAR-10, Patched} models.

Neuron Pruning. After obtaining the reversed trigger, Wang et al. [32] used it to patch the infected model by neuron pruning. To rigorously test the proposed method, we use the real trigger instead of the reversed one for pruning. The results are shown in Fig. 10.

It can be seen that the curves of BA and ASR are affected by ML-MMDR. For example, the two curves almost overlap on the {CIFAR-10, V-11, Warped} models, showing similar downward trends. On the {CelebA} models, in most cases, BA and ASR quickly drop to low values simultaneously. The results indicate that the performance of the neuron

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3161477, IEEE Transactions on Dependable and Secure Computing

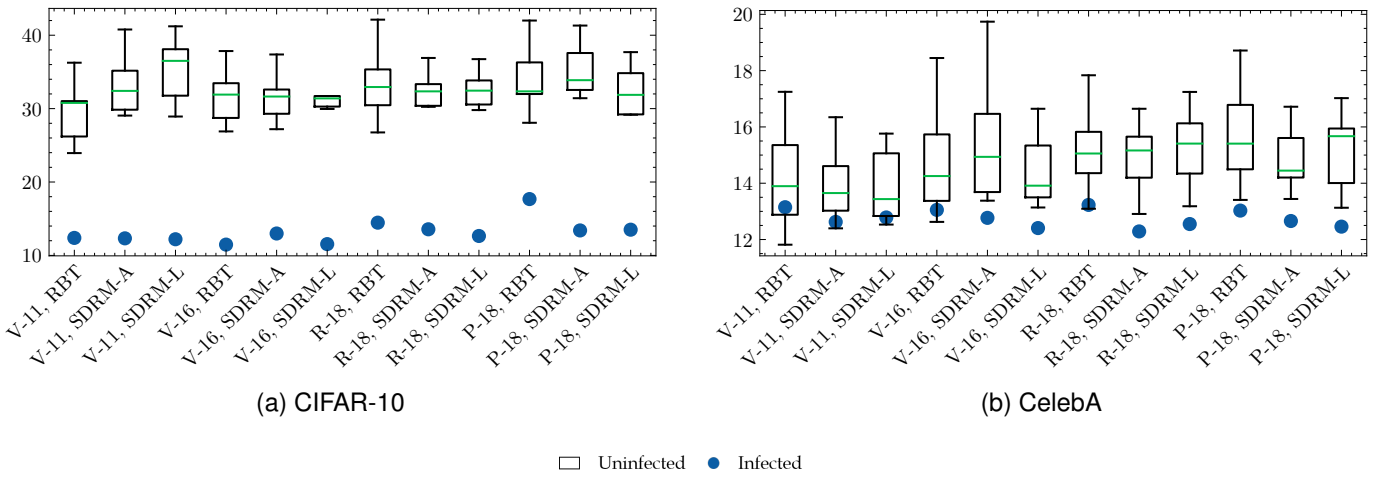

Fig. 8. l_1 norm of triggers for infected and uninfected labels on the {Patched} models. X-axis: the model architecture and the backdoor training method. Y-axis: the l_1 norm of trigger.

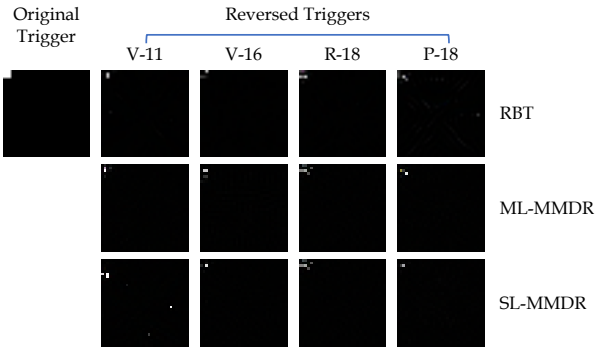

Fig. 9. Reversed triggers on the {CIFAR-10, Patched} models.

pruning is reduced on the models trained with ML-MMDR compared to on the models trained with RBT. We think these results are reasonable because neuron pruning essentially assumes that the backdoor behavior is encoded into some neurons, which is similar to the assumption of the distributional differences.

Interestingly, we find that in some cases, ASR declines at the beginning and rises later as the pruning ratio increases, such as on the {CelebA, R-18, Patched} model trained with RBT. We believe there are two possible reasons:

- Neuro pruning preferentially masks the neurons with large activation values on malicious samples, but this does not mean that the model would not classify the input into the target category t, especially when the pruning ratio is large.
- Backdoor attacks may leave a bias in the infected model during training. This bias would play a dominant role when the model's functionality is greatly broken, causing the model to classify arbitrary inputs into the backdoor target t.

6.3 Ablation Study on Different Kernels

Considering that the choice of a kernel in MMD may have an impact on the training of the backdoored model, we conduct experiments on different kernels, including Gaussian Kernel (GK), Gaussian Mixture Kernel (GMK), and Linear Kernel (LK). The detailed settings are shown in TABLE 3. The results of three difference-based defense methods are shown in Fig. 11.

TABLE 3
Settings of different kernels. σ: the standard deviation.

Kernel	Parameter
GK1	$\sigma = 1/2$
GK2	$\sigma = 1$
GK3	$\sigma = 2$
GMK1	$\sigma = [1/2, 1, 2]$
GMK2	$\sigma = [1/4, 1/2, 1, 2, 4]$
GMK3	$\sigma = [1/8, 1/4, 1/2, 1, 2, 4, 8]$
GMK4	$\sigma = [1/3, 1, 3]$
GMK5	$\sigma = [1/9, 1/3, 1, 3, 9]$
GMK6	$\sigma = [1/27, 1/9, 1/3, 1, 3, 9, 27]$
LK	-

Two observations can be seen from the figure. First of all, no matter which kernel is used, ML-MMDR significantly reduces the detection effects. Second, GK or GMK is a better choice than LK. We believe that the reason is that these two types of kernels map the original dimensions to infinite dimensions, which makes the measurement of MMD more accurate.

7 Discussion

One of the trends of backdoor attacks is to become more stealthy, and this stealthiness is generally reflected at both the data and model levels. The stealthiness at the data level refers to whether the malicious samples provided by the attacker are themselves detectable. For example, the trigger added to the malicious samples could be perceptible [39]. The stealthiness at the model level refers to whether the infected model itself would reveal that it has been implanted with a backdoor. For example, the distributional differences concerned in this paper are those that exist in the latent spaces determined by the model. Most existing studies [29],

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3161477, IEEE Transactions on Dependable and Secure Computing

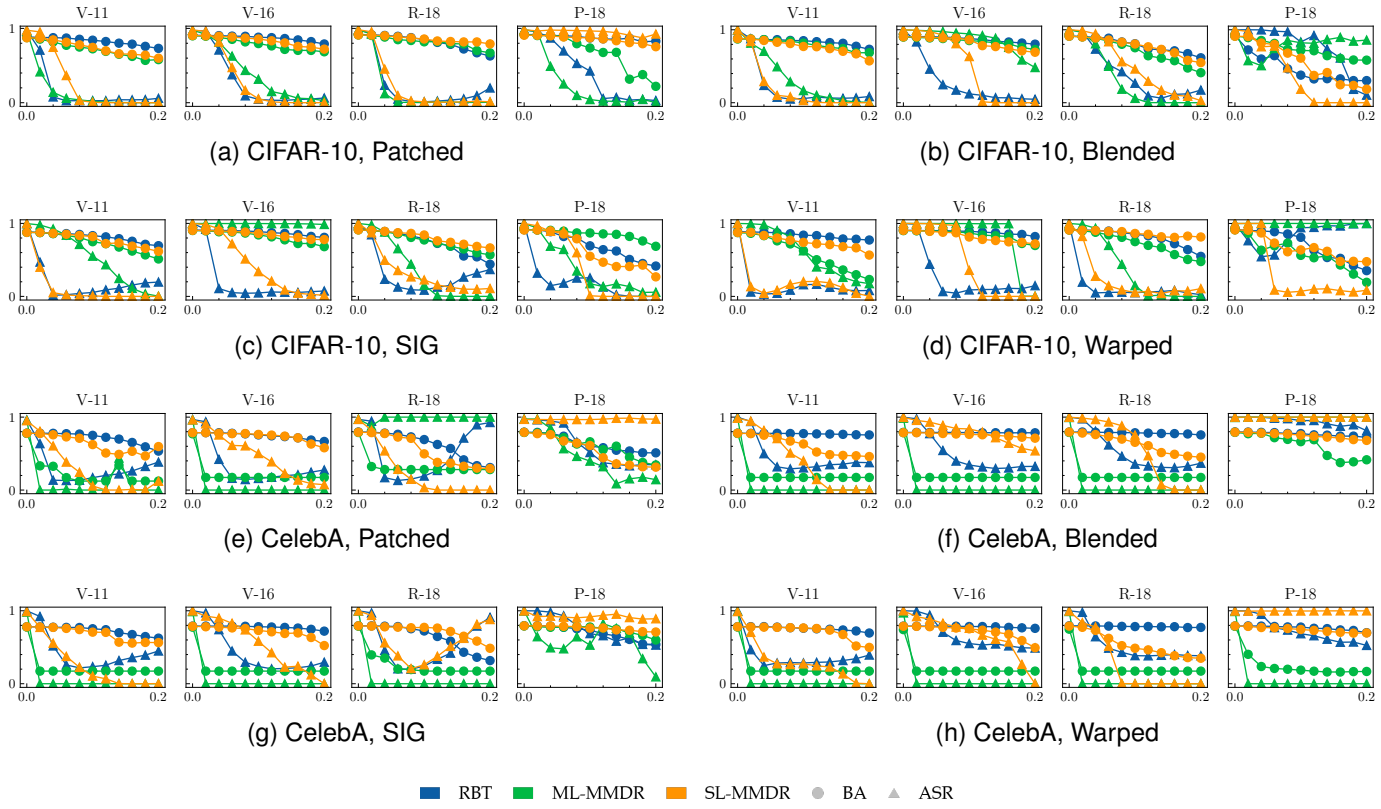

Fig. 10. BA and ASR of the infected models when pruning the trigger-related neurons. X-axis: the ratio of neurons pruned. Y-axis: the benign accuracy and the attack success rate.

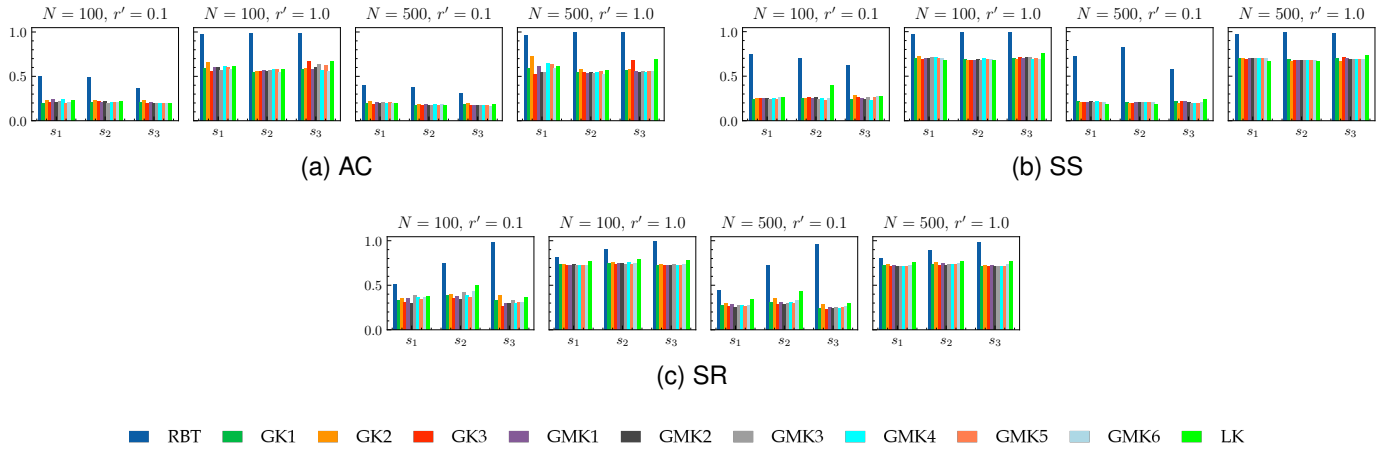

Fig. 11. F1 scores of AC, SS, SR on the {CIFAR-10, P-18, Patched} models with different kernels.

[40], [41] focus on one of these two levels. However, it is not sufficient to consider each one individually, because the defender may set up multiple defense methods. Constructing an attack that can bypass defense methods with different principles requires further research.

The proposed method in this paper reduces the distributional differences by adding a constraint to the loss during the training of a backdoored model. It requires the attacker to have permission to train the model. Another more relaxed setting is that the attacker only provides training data, and the user trained the model on the data. Can the distributional differences still be reduced in this case? We argue the answer is positive. However, by combining the optimized trigger [51] and MMD, we make a preliminary attempt and find it is not easy. The specific steps of the attempt are as follows. First, we optimize a trigger on a trained benign model, so that adding the trigger to a benign sample can not only force the model's prediction into a specified category, but also minimize the distributional differences of the latent representations. Then, we use this optimized trigger

to construct a poisoned training set and train an infected model on these data with RBT. Unfortunately, the trained model can always optimize parameters that are easier for completing the backdoor task, but are less stealthy. We plan it in future work.

8 Related Work

8.1 Backdoor Attacks

Since Gu et al. [16] first explored the backdoor vulnerability in deep learning models, many variants have been proposed. From the method of injecting the hidden threat, backdoor attacks can be roughly divided into two categories, i.e., poisoning-based attacks and non-poisoning-based attacks. Poisoning-based methods [15], [16], [17], [29], [40], [41], [42] execute the Trojan horse implantation by mixing a small number of malicious samples into the training data and learning an infected model from the mixed dataset. Constructing more stealthy and effective malicious samples is the research focus of this type of attack. The primary method [16] uses a static local patch as the trigger. To obtain better attack performance, Liu et al. [17] built a Trojan method, in which the trigger is optimized rather than predefined. Zhong et al. [39] argued that the triggers of the previous attacks [16], [17] are all visually visible, which undermines the stealthiness of the backdoor. Then, they proposed adding an imperceptible perturbation mask to the benign image to generate its malicious sample. However, these attacks ignore that the inconsistency of the instance and its label can increase the risk of disclosure. Turner et al. [40] first pointed out this problem and proposed the label-consistent backdoor attacks by leveraging adversarial examples and generative models. In addition to the above digital attacks, some studies focus on the physical world. Chen et al. [15] adopted a pair of glasses as the physical trigger to fool a face recognition system. Li et al. [43] suggested that physical transformations should be considered when training a backdoored model.

Non-poisoning-based methods [44], [45], [46], [47], [48] encode the backdoor functionality into deep models by transfer learning or DNNs. Dumford and Scheirer [44] first explored the possibility of injecting the backdoor without poisoning, where they proposed to modify the model's parameters directly. Unlike [44], Tang et al. [47] proposed a training-free method by inserting a malicious module into the target model instead of perturbing the parameters.

The proposed method can be used to enhance the above attack methods to escape difference-based backdoor detection. As shown in 6, ML-MMDR adds a constraint item to the loss function.

8.2 Backdoor Defenses

Several defense approaches have been proposed to defend against these attacks, such as backdoor detection [13], [49], trigger synthesis [32], [50], model reconstruction [51], [52], and model diagnosis [53], [54]. Among them, a vast number of defense methods [13], [14], [19], [20], [55], [56], [57] distinguish backdoor samples from the benign ones by exploiting the distributional differences between benign and malicious representations. Three typical methods have been introduced above, we here add some others. Chou et al. [56] leveraged the power of visualization tools, such as Grad-CAM [58], to inspect the backdoor behavior of malicious models. Soremekun et al. [57] proposed a detection method similar to the activation clustering [13], which uses t-SNE as the dimensionality reduction and the mean-shift as the clustering algorithm. Hayase et al. [20] argued that the previous defense [14] works only when the spectral signature is large and proposed a method, SPECTRE, using robust covariance estimation to amplify the signature of malicious data.

However, our work demonstrates that the performance of difference-based detection methods can be greatly reduced when using the infected models trained with ML-MMDR. This proves that there is a need to study more powerful defense methods.

9 Conclusion

Our work studies the distributional differences between the latent representations of benign and malicious samples, which is the assumption of many detection methods. We identify that the distributional differences of multiple levels are all large enough to be used to detect malicious samples. Then, we propose ML-MMDR, a multi-level MMD regularization, and demonstrate that the differences can be considerably reduced without harming the attack intensity. Finally, the experimental results of three difference-based defense methods, i.e., activation clustering, spectral signatures, and subspace reconstruction, indicate that the defense performance decreases drastically as the differences reduce. The proposed MMD regularization can enhance existing attack methods to escape backdoor detection algorithms.

Acknowledgment

The work is partially supported by the National Natural Science Foundation of China under grant No.U19B2044 and No.61836011.

References

[1] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[3] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[4] J. Yin, P. Xia, and J. He, "Online hard region mining for semantic segmentation," *Neural Processing Letters*, vol. 50, no. 3, pp. 2665–2679, 2019.

[5] P. Xia, J. He, and J. Yin, "Boosting image caption generation with feature fusion module," *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 24 225–24 239, 2020.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014.

[7] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced lstm for natural language inference," *arXiv preprint arXiv:1609.06038*, 2016.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[9] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[10] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[11] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[13] 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 *SafeAI@ AAAI*, 2019.

[14] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[15] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[16] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[17] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," 2017.

[18] M. Xue, C. He, J. Wang, and W. Liu, "One-to-n & n-to-one: Two advanced backdoor attacks against deep learning models," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[19] K. Jin, T. Zhang, C. Shen, Y. Chen, M. Fan, C. Lin, and T. Liu, "A unified framework for analyzing and detecting malicious examples of dnn models," *arXiv preprint arXiv:2006.14871*, 2020.

[20] J. Hayase, W. Kong, R. Somani, and S. Oh, "Spectre: Defending against backdoor attacks using robust statistics," *arXiv preprint arXiv:2104.11315*, 2021.

[21] T. J. L. Tan and R. Shokri, "Bypassing backdoor detection algorithms in deep learning," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020, pp. 175–183.

[22] K. Doan, Y. Lao, and P. Li, "Backdoor attack with imperceptible input and latent modification," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[23] Y. Ren, L. Li, and J. Zhou, "Simtrojan: Stealthy backdoor attack," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 819–823.

[24] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[25] G. J. Székely and M. L. Rizzo, "Energy statistics: A class of statistics based on distances," *Journal of statistical planning and inference*, vol. 143, no. 8, pp. 1249–1272, 2013.

[26] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. Rohde, "Generalized sliced wasserstein distances," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[27] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 101–105.

[28] C. Villani, *Optimal transport: old and new*. Springer, 2009, vol. 338.

[29] A. Nguyen and A. Tran, "Wanet–imperceptible warping-based backdoor attack," *arXiv preprint arXiv:2102.10369*, 2021.

[30] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[31] M. Javaheripi, M. Samragh, G. Fields, T. Javidi, and F. Koushanfar, "Cleann: Accelerated trojan shield for embedded neural networks," in *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2020, pp. 1–9.

[32] 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," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 707–723.

[33] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[34] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," *arXiv preprint arXiv:2003.03675*, 2020.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] ——, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[38] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[39] H. Zhong, C. Liao, A. C. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 2020, pp. 97–108.

[40] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.

[41] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2088–2105, 2020.

[42] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 182–199.

[43] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the trigger of backdoor attack," *arXiv preprint arXiv:2004.04692*, 2020.

[44] J. Dumford and W. Scheirer, "Backdooring convolutional neural networks via targeted weight perturbations," in *2020 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2018, pp. 1–9.

[45] K. Kurita, P. Michel, and G. Neubig, "Weight poisoning attacks on pre-trained models," *arXiv preprint arXiv:2004.06660*, 2020.

[46] A. S. Rakin, Z. He, and D. Fan, "Tbt: Targeted neural network attack with bit trojan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 198–13 207.

[47] R. Tang, M. Du, N. Liu, F. Yang, and X. Hu, "An embarrassingly simple approach for trojan attack in deep neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 218–228.

[48] S. Wang, S. Nepal, C. Rudolph, M. Grobler, S. Chen, and T. Chen, "Backdoor attacks against transfer learning with pre-trained deep learning models," *IEEE Transactions on Services Computing*, 2020.

[49] M. Ficco, "Malware analysis by combining multiple detectors and observation windows," *IEEE Transactions on Computers*, 2021.

[50] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks." in *IJCAI*, 2019, pp. 4658–4664.

[51] Y. Liu, Y. Xie, and A. Srivastava, "Neural trojans," in *2017 IEEE International Conference on Computer Design (ICCD)*. IEEE, 2017, pp. 45–48.

[52] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," *arXiv preprint arXiv:2101.05930*, 2021.

[53] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting ai trojans using meta neural analysis," *arXiv preprint arXiv:1910.03137*, 2019.

[54] S. Kolouri, A. Saha, H. Pirsiavash, and H. Hoffmann, "Universal litmus patterns: Revealing backdoor attacks in cnns," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 301–310.

[55] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2018, pp. 273–294.

[56] E. Chou, F. Tramèr, and G. Pellegrino, "Sentinet: Detecting localized universal attacks against deep learning systems," in *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020, pp. 48–54.

[57] E. Soremekun, S. Udeshi, S. Chattopadhyay, and A. Zeller, "Exposing backdoors in robust machine learning models," *arXiv preprint arXiv:2003.00865*, 2020.

[58] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

Pengfei Xia received the B.E. degree from the China University of Mining and Technology (CUMT), Xuzhou, China, in 2015 and is pursuing the Ph.D. degree from University of Science and Technology of China (USTC), Hefei, China. His research interests include adversarial examples, backdoor learning, and secure deep learning.

Hongjing Niu received the B.E. degree in electronic science and technology from University of Science and Technology of China (USTC), Hefei, China, in 2018 and is pursuing the Ph.D. degree in Control Science and Engineering from University of Science and Technology of China (USTC), Hefei, China. His research interests include representation learning and explainable artificial intelligence.

Ziqiang Li received the B.E. degree in electronic science and technology from University of Science and Technology of China (USTC), Hefei, China, in 2019 and is pursuing the Master degree in Information and Communication Engineering from University of Science and Technology of China (USTC), Hefei, China. His research interests include medical image segmentation, deep generative models, and machine learning.

Bin Li received the B.S. degree from the Hefei University of Technology (HFUT), China, in 1992, the M.S. degree from the Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, China, in 1995, and the Ph.D degree from the University of Science and Technology of China (USTC), Hefei, China, in 2001. He is currently a Professor with the School of Information Science and Technology, USTC. He has authored or co-authored over 40 refereed publications. His current research interests include evolutionary computation, pattern recognition, and human-computer interaction. Dr. Li is the Founding Chair of the IEEE Computational Intelligence Society Hefei Chapter, a Counselor of the IEEE USTC Student Branch, a Senior Member of the Chinese Institute of Electronics (CIE), and a member of the Technical Committee of the Electronic Circuits and Systems Section of CIE.