Detecting Backdoors During the Inference Stage Based on Corruption Robustness Consistency

Xiaogeng Liu[1, 4, 5, 6, 7], Minghui Li[3], Haoyu Wang[1, 4, 5, 6, 7], Shengshan Hu[1, 4, 5, 6, 7]
Dengpan Ye[9], Hai Jin[2, 4, 5, 8], Libing Wu[9], Chaowei Xiao[10]
[1]School of Cyber Science and Engineering, Huazhong University of Science and Technology
[2]School of Computer Science and Technology, Huazhong University of Science and Technology
[3]School of Software Engineering, Huazhong University of Science and Technology
[4]National Engineering Research Center for Big Data Technology and System
[5]Services Computing Technology and System Lab
[6]Hubei Key Laboratory of Distributed System Security
[7]Hubei Engineering Research Center on Big Data Security [8]Cluster and Grid Computing Lab
[9]School of Cyber Science and Engineering, Wuhan University [10]Arizona State University
{liuxiaogeng, minghuili, wwwwhy, hushengshan, hjin}@hust.edu.cn
{yedp, wu}@whu.edu.cn xiaocw@asu.edu

Abstract

Deep neural networks are proven to be vulnerable to backdoor attacks. Detecting the trigger samples during the inference stage, i.e., the test-time trigger sample detection, can prevent the backdoor from being triggered. However, existing detection methods often require the defenders to have high accessibility to victim models, extra clean data, or knowledge about the appearance of backdoor triggers, limiting their practicality.

In this paper, we propose the test-time corruption robustness consistency evaluation (TeCo)[1], a novel test-time trigger sample detection method that only needs the hard-label outputs of the victim models without any extra information. Our journey begins with the intriguing observation that the backdoor-infected models have similar performance across different image corruptions for the clean images, but perform discrepantly for the trigger samples. Based on this phenomenon, we design TeCo to evaluate test-time robustness consistency by calculating the deviation of severity that leads to predictions' transition across different corruptions. Extensive experiments demonstrate that compared with state-of-the-art defenses, which even require either certain information about the trigger types or accessibility of clean data, TeCo outperforms them on different backdoor attacks, datasets, and model architectures, enjoying a higher AUROC by 10% *and* 5 *times of stability.*

1. Introduction

Backdoor attacks have been shown to be a threat to *deep neural networks* (DNNs) [14, 26, 32, 38]. A backdoor-infected DNN will perform normally on clean input data, but output the adversarially desirable target label when the input data are tampered with a special pattern (*i.e.*, the backdoor trigger), which may cause serious safety issues.

A critical dependency of a successful backdoor attack is that the attacker must provide the samples with backdoor triggers (we call them trigger samples for short hereafter) to the infected models on the inference stage, otherwise, the backdoor will not be triggered. Thus, one way to counter the backdoor attacks is to judge whether the test data have triggers on it, *i.e.*, the *test-time trigger sample detection* (TTSD) defense[2] [5, 12, 42]. This kind of defense can work corporately with other backdoor defenses such as model diagnosis defense [9, 15, 46] or trigger reverse engineering [40, 43], and also provide prior knowledge of the trigger samples in a comprehensive defense pipeline, which can help the down-steam defenses to statistically analyze the backdoor samples and mitigate the backdoor more effectively.

On the other hand, the TTSD method, especially the black-box TTSD method can also serve as the last line of defense when someone adopts models with unknown credibility and has no authority to get access to the training data or model parameters, this scenario exists widely in the pre-

[1]https://github.com/CGCL-codes/TeCo

[2]Some paper also call it online backdoor defense [30, 39].

vailing *machine-learning-as-a-service* (MLaaS) [19, 35].

However, with the development of backdoor attacks, the TTSD defense is facing great challenges. One of the major problems is that different types of triggers have been presented. Unlike the early backdoor attacks whose triggers are universal [3, 14] for all the images and usually conspicuous to human observers, recent works introduced sample-specific triggers [32] and even invisible triggers [8, 21, 26, 33, 49], making it harder to apply pattern statistics or identify out-liners in the image space. Another main problem is the hardship of accomplishing the TTSD defense without extra knowledge such as supplemental data or model accessibility. On the other hand, existing TTSD methods require certain knowledge and assumption. Such assumptions include that the trigger is a specific type [12, 42], the defenders have white-box accessibility to victim models, the predicted soft confidence score of each class [5, 12] or extra clean data for statistical analysis [48], limiting the practicality for real-world applications.

In this paper, we aim to design a TTSD defense free from these limitations. Specifically, we concentrate on a more practicable black-box hard-label backdoor setting [15] where defenders can only get the final decision from the black-box victim models. In addition, no extra data is accessible and no assumption on trigger appearance is allowed. This setting assumes the defenders' ability as weak as possible and makes TTSD hard to achieve. To the best of our knowledge, we are the first to focus on the effectiveness of TTSD in this strict setting, and we believe it is desirable to develop TTSD methods working on such a scenario because it is very relevant to the wide deployment of cloud AI service [4, 11] and embedded AI devices [1].

Since the setting we mentioned above has restricted the accessibility of victim models and the use of extra data, we cannot analyze the information in feature space [30, 39] or train a trigger sample detector [10, 48] like existing works. Fortunately, we find that the backdoor-infected models will present clearly different corruption robustness for trigger samples influenced by different image corruptions, but have relatively similar robustness throughout different image corruptions for clean samples, leaving the clue for trigger sample detection. We call these findings the *anomalous corruption robustness consistency* of backdoor-infected models and describe them at length in Sec. 3. It is not the first time that image corruptions are discussed in backdoor attacks and defenses [27, 28, 34]. However, previous works fail to explore the correlations between robustness against different corruptions, as discussed in Sec. 3.3.

Based on our findings above, we propose *test-time corruption robustness consistency evaluation* (TeCo), a novel test-time trigger sample detection method. At the inference stage of backdoor-infected models, TeCo modifies the input images by commonly used image corruptions [18] with growing severity and estimates the robustness against different types of corruptions from the hard-label outputs of the models. Then, a deviation measurement method is applied to calculate how spread out the results of robustness are. And TeCo makes the final judgment of whether the input images are with triggers based on this metric. Extensive experiments show that compared with the existing advanced TTSD method, TeCo improves AUROC about 10%, has a higher F1-score of 14%, and achieves 5 times of stability against different types of trigger.

Method	Black-box Access		No Need of	Trigger Aussmptions		
	Logits-based	Decision-based	Clean Data	Universal	Sample-specific	Invisible
SentiNet [5]	○	○	○	●	○	○
SCan [39]	○	○	○	●	○	○
Beatrix [30]	○	○	○	●	●	●
NEO[3] [42]	●	●	●	○	○	○
STRIP [12]	●	○	○	●	○	○
FreqDetector [48]	●	●	○	●	●	●
TeCo (Ours)	●	●	●	●	●	●

Table 1. The model's accessibility, the use of clean data, and the assumptions on backdoor triggers required by various TTSD methods. We detail on some most related defenses in Sec. 2. "●" represents the TTSD method supports this condition.

Finally, we take a deep investigation into our observations by constructing adaptive attacks against TeCo. From the results of feature space visualization and quantification of adaptive attacks, we speculate that the anomalous behavior of corruption robustness consistency derives from the widely-used dual-target training in backdoor attacks and it is hard to be avoided by existing trigger types. We hope these findings can shed light on a new perspective of backdoor attacks and defenses for the community. In summary, we make the following contributions:

- We propose TeCo, a novel test-time trigger sample detection method that only requires the hard-label outputs of the victim models and without extra data or assumptions about trigger types.
- We discover the fact of anomalous corruption robustness consistency, *i.e.*, the backdoor-infected models have similar performance across different image corruptions for clean images, but not for the trigger samples.
- We evaluate TeCo on five datasets, four model architectures (including CNNs and ViTs), and seven backdoor attacks with diverse trigger types. All experimental results support that TeCo outperforms state-of-the-art methods.
- We further analyze our observations by constructing adaptive attacks against TeCo. Experiments show that the widely-used dual-target training in backdoor attacks leads to anomalous corruption robustness consistency and it is hard to be avoided by existing backdoor triggers.

[3]NEO assumes the backdoor trigger is localized [14] thus will be invalid on distributed or global triggers [3, 8, 32, 48], including universal, sample-sepcific, and invisible ones.

2. Related Works

2.1. Backdoor Attacks

Badnets [14] is the first work that describes how to embed a backdoor into the DNNs by poisoning part of the training data. Many backdoor attacks have been developed after Badnets. To categorize these works, a reasonable way is to divide them by the triggers' appearance of these attacks, *i.e.*, the trigger types. The universal trigger is a classical trigger type that is leveraged in many works such as Badnets [14], Blended [3], and Low-frequency [48]. Universal trigger represents that for any input images tampered with the same trigger, the backdoor-infected model will give the predefined predictions. Then, the sample-specific trigger is invented [26,32,37]. Unlike universal triggers, the appearance of sample-specific triggers depends on the images that the trigger attaches. Another research topic in backdoor attacks is the imperceptibility of the backdoor triggers, *i.e.*, the invisible backdoor attacks, such as Wanet [33], LIRA [8], and SSBA [26]. The triggers generated by this kind of attack only lead to subtle modifications in images, and humans can hardly perceive the existence of backdoor triggers. A backdoor attack can meet the sample-specific and invisible conditions simultaneously, for example, the SSBA attack [26].

2.2. Backdoor Defenses

In this paper, we mainly discuss the works which use trigger sample detection as a defense method. Since the success of backdoor attacks depends on the existence of trigger samples, detecting those trigger samples in training data or test data is a reasonable way to defend against backdoor attacks. Some detection methods focus on filtering trigger samples in the training stage and attempt to eliminate the backdoor attacks by preventing them from poisoning training data [2, 16, 41]. On the other hand, the *test-time trigger sample detection* (TTSD) is developed since defenders cannot always control the training process of victim models.

The first black-box TTSD method is STRIP [12]. STRIP superimposes various clean images on the suspicious samples and evaluates the randomness of the model's logits outputs. NEO [42] assumes the backdoor trigger is localized and detects the trigger samples by masking random areas of the suspicious samples and repainting them with the dominant color. FreqDetector [48] finds that backdoor triggers often cause artifacts in the frequency space of the suspicious samples, and detects the trigger samples by training a frequency detector on clean images with data augmentations. Some other works assume they have white-box accessibility of the backdoor-infected models and detect the trigger samples by the salient maps [5] or the features of intermediate layers [30, 39].

Since TTSD methods often make judgments based on certain statistical patterns of trigger samples, the advances in trigger types mentioned above put great threats to the TTSD methods with no doubt. As shown in Tab. 1, we conclude that existing TTSD methods have relaxed their restrictions of defenders to achieve satisfying performance, leading to incomplete black-box settings, such as the requirement for model accessibility [5, 7, 22, 30, 39], use of clean data [6, 10, 12, 28, 48], and assumptions on specific trigger type [7, 12, 39, 42].

3. Corruption Robustness Consistency

Before introducing our black-box trigger sample detection method, we first delineate the important findings that we discover from backdoor-infected models: *given a backdoor-infected model, it will show clearly different robustness for trigger samples influenced by different image corruptions. However, for the clean images, the model will show similar robustness against the majority of image corruptions.* We stress that these phenomena exist widely in different backdoor-infected models.

3.1. Corruption Robustness Consistency Test

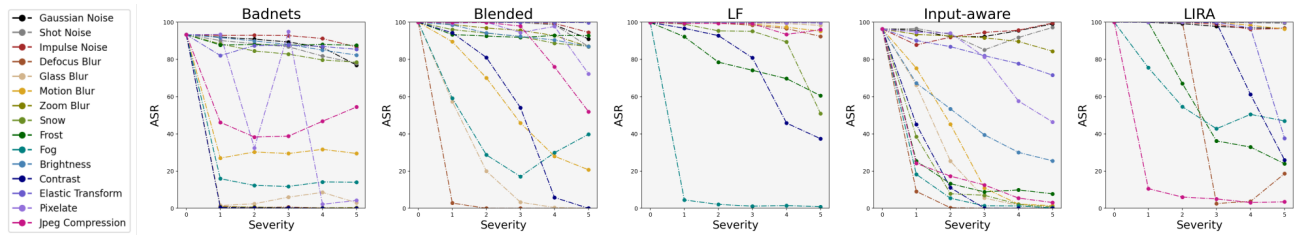
We gain our findings by conducting *Corruption Robustness Consistency* (CRC) test on backdoor-infected models. Given an infected model C_θ, and an image corruption set $\mathcal{D}_K^N$ which has K corruption types and N levels of severity, CRC test computes the clean accuracy (ACC) of the clean images tempered with different image corruptions, or evaluates the *attack success rate* (ASR) of the trigger samples tempered with different image corruptions. CRC test builds a list $L_{K,N}$ of ACC or ASR, where each element in this list is calculated by:

$$L_{k,d} = \begin{cases} \frac{1}{I}\sum_{i=1}^{I} \mathbb{I}(C_\theta(D_n^k(x_i)) = y_i), \text{for clean samples} \\ \frac{1}{J}\sum_{j=1}^{J} \mathbb{I}(C_\theta(D_n^k(\hat{x_j})) = y_t), \text{for trigger samples} \end{cases} \tag{1}$$

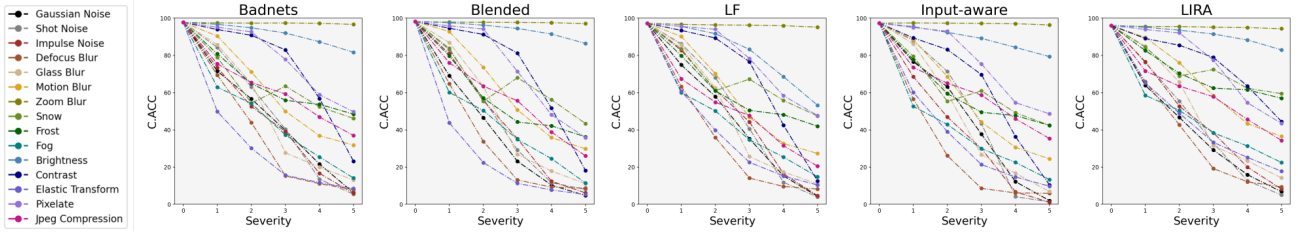
where I is the number of clean images, J is the number of trigger samples, x_i represents the clean image, $\hat{x_j}$ represents the trigger sample, and D_n^k represents the k-th image corruption in the corruption set $\mathcal{D}_K^N$ with severity n. y_i is the ground-truth label of x_i, y_t is the target label that the adversaries want the infected model to predict when the trigger sample is given. $\mathbb{I}(\cdot)$ is an indicator function, where $I(A) = 1$ if and only if A is true.

3.2. Anomalous CRC of Backdoor-infected Models

The list $L_{K,N}$ built in CRC test can be used to measure the corruption robustness of backdoor-infected models. We choose the image corruption set described in [18], where the common image corruptions are categorized into 15 classes and each kind of corruptions has 5 levels of severity, then

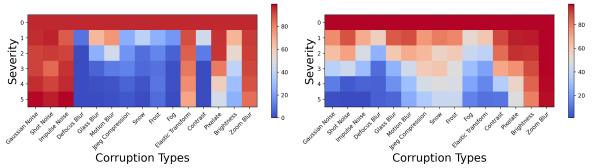


(a) The curves of ASR on trigger samples



(b) The curves of ACC on clean images

Figure 1. (a): The backdoor-infected model's *attack success rate* (ASR) when trigger samples are tempered with different corruptions and levels of severity. (b): The accuracy (ACC) of clean images tempered with different corruptions and levels of severity. The curves separate loosely in (a), while the majority of curves gather more tightly in (b). This indicates that the backdoor-infected models have varied corruption robustness against different image corruptions on trigger samples, but have similar robustness against different image corruptions on clean samples.



(a) ASR of trigger samples (b) ACC of clean samples

Figure 2. Take input-aware attack infected model as an example. Compared with clean samples, trigger samples have a more uneven heat map, which means that the backdoor-infected model are very robust on certain corruptions but also pretty vulnerable to some other corruptions.

conduct CRC test on models infected by five backdoor attacks. From the visualization results in Fig. 1(b), the majority of curves are relatively clustered and show a downward trend. We describe this phenomenon as the model has good corruption robustness consistency, because the model performs similarly on different image corruptions.

However, in Fig. 1(a), the curves are more separated, indicating that the model has contrasting robustness against different image corruptions. Consequently, the model can be regarded as having bad corruption robustness consistency on trigger samples. Compared with the observations about Fig. 1(b), the backdoor-infected models are suspicious to have different corruption robustness consistency on clean samples and trigger samples, *i.e.*, the phenomenon of anomalous CRC.

3.3. Difference Between CRC and Previous Works

Some previous works have discussed the corruption (or transformation) robustness of backdoor-infected models before [27,28,34]. Specifically, Gaussian noise is investigated in [28], where the authors argue that adding this kind of noise can lead to abnormal behavior of backdoor-infected models on trigger samples. In [34], image transformations are used in a two-stage defense pipeline, where the defenders first fine-tune the infected models on one set of transformations and uses another set of transformations on the inference stage. This work is different from ours since it changes the parameters of backdoor-infected models, while we mainly focus on the characteristics of backdoor-infected models without modifications. In [27], the authors evaluate the robustness of backdoor-infected models against multiple image transformations. They argue that some image transformations can mitigate the backdoor while others cannot, which is similar to our findings. But they still rely on a single transformation to defend against backdoor attacks and thus fail to leverage the difference of robustness.

4. Test-time CRC Evaluation (TeCo)

In this section, we describe how we build our method based on the phenomenon of anomalous CRC.

4.1. Preliminaries

The objective of backdoor attacks is to make the infected model behave normally on clean images but give predefined predictions on trigger samples. Thus, the target [15,25] of backdoor attackers is training an infected model C with parameters θ by:

$$\theta = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{P}_S} \mathcal{J}(C(x;\theta), y) + \mathbb{E}_{(\hat{x},y_t)\sim\mathcal{P}_{\hat{S}}} \mathcal{J}(C(\hat{x};\theta), y_t), \tag{2}$$

where S, $\hat{S}$ represent the clean data and trigger data, respectively, and $\mathcal{J}$ is the loss function.

TTSD methods work on a trained backdoor-infected model C_θ and a test dataset which contains clean samples and trigger samples $\mathcal{T} = \{T \cup \hat{T}\}$. The goal of designing TTSD is to find a method M:

$$M = \arg\max_{M} \mathbb{E}_{(x)\sim\mathcal{P}_T}\mathbb{I}(M(x, C_\theta) = 0) + \mathbb{E}_{(\hat{x})\sim\mathcal{P}_{\hat{T}}}\mathbb{I}(M(\hat{x}, C_\theta) = 1). \tag{3}$$

4.2. Test-time CRC Evaluation

To achieve Eq. (2), the trigger sample detection methods should leverage contrasting characteristics of clean images and trigger samples. We have revealed in Sec. 3 that backdoor-infected models have anomalous corruption robustness consistency, which is supported by the ACC and ASR evaluated from the entire test dataset. However, a question is how we can measure this property in test-time based on single input data.

A reasonable understanding is that the reduction of ACC or ASR is equivalent to the transitions of prediction labels. For example, if a model loses its accuracy on clean images with Gaussian noise, it can be regarded as the prediction labels of these images have changed compared with the original images'. Consequently, we can evaluate the corruption robustness consistency in the inference stage by adding image corruptions with growing severity, and recording the severity when the model's hard-label prediction gets changed. For different image corruptions, if the recorded severity levels are very similar, we can extrapolate that the corruption robustness consistency on the input image is high.

After recording the levels of severity, the final step is to measure their dispersion. It is not hard to find such a metric since simply calculating the standard deviation is already effective according to our experiments. Alg. 1 describes the detailed algorithm. TeCo maps the input image x to a linearly separable space, and defenders can make judgments by a threshold γ:

$$\Gamma(TeCo(x)) = \begin{cases} 1, TeCo(x) > \gamma \\ 0, TeCo(x) \leq \gamma \end{cases} \tag{4}$$

Algorithm 1: Test-time CRC Evaluation (TeCo)

Input: Test sample x; test model C_θ; deviation measurement method Dev;image corruption set $\mathcal{D}_K^N$, where K is the number of corruption types, and N is the maximum of severity.

Output: Prediction score of test sample x.

```
Initialize L ← {}, P_org ← C_θ(x);
for k = 1 to K do
    L ← N + 1;
    for n = 1 to N do
        if C_θ(D_k^n(x)) ≠ P_org then
            L ← n;
            break;
        end
    end
    L ← L ∪ {L};
end
deviation ← Dev(L);
return deviation
```

5. Experiments

5.1. Experimental Settings

Implementation details. We take the common image corruptions introduced in [18] as the image corruption set D_k^N in Alg. 1. This corruption set has 15 diverse image corruptions with the severity ranging from 1 to 5. We choose standard deviation as the deviation measurement method[4].

Attack methods. We evaluate our method against seven backdoor attacks, including Badnets attack [14], Blended attack [3], *Low-frequency* (LF) attack [48], Input-aware attack [32], Wanet attack [33], LIRA attack [8], and SSBA attack [26]. We follow an open-sourced backdoor benchmark [45] for the training settings of these attacks. To ensure the attacks' strength, 10% of the training set are poisoned. As illustrated in Tab. 3, the attacks in our experiments contain different trigger types.

Datasets and backbones. Five datasets and four backbones are involved in our experiments. The datasets include CIFAR10 [23], CIFAR100 [23], GTSRB [20], Tiny-ImageNet [24], and ImageNet200 [36] which is used in [26]. For images in relatively low size, we use PreActResNet18 [17] and MobileViT-xs [31] as the backbones. And for ImageNet200, we use WideResNet101-2 [47] and SwinTransformer-Base [29], and fine-tune them from checkpoints [44] pre-trained on ImageNet1K.

Competitors. Since TeCo is the first test-time trigger sample detection method that works in hard-label black-box settings and has no extra dependency, We compare our method with two trigger sample detection methods that work in looser conditions but still meet the black-box requirement. To the best of our knowledge, STRIP [12] is the first black-box TTSD method, and it still serves as a baseline in many recent works [30, 48]; Frequency detector (FreqDetector) [48] is the state-of-the-art trigger sample detection method. We implement them following their official codes. STRIP needs the logits-based black-box accessibility, while FreqDetector has no requirement for accessing the backdoor models. In addition, both of them need extra clean data to accomplish the trigger detection task.

[4] We investigate the choice of image corruption set and deviation measurement method in the supplementary.

Dataset	Model	Attack→ / Detection↓	Badnets [14] AUROC	Badnets [14] F1 score	Blended [3] AUROC	Blended [3] F1 score	LF [48] AUROC	LF [48] F1 score	Input-aware [32] AUROC	Input-aware [32] F1 score	Wanet [33] AUROC	Wanet [33] F1 score	LIRA [8] AUROC	LIRA [8] F1 score	SSBA [26] AUROC	SSBA [26] F1 score	AVG(↑) AUROC	AVG(↑) F1 score	STD(↓) AUROC	STD(↓) F1 score
CIFAR10	PreActResNet18	STRIP	0.790	0.743	0.726	0.685	0.973	0.937	0.283	0.526	0.395	0.526	0.555	0.661	0.364	0.526	0.584	0.658	0.236	0.140
		FreqDetector	0.989	0.955	0.966	0.904	0.886	0.809	1.000	0.993	0.566	0.550	0.912	0.840	0.896	0.824	0.888	0.839	0.138	0.134
		Ours	0.911	0.917	0.935	0.946	0.939	0.937	0.905	0.921	0.915	0.905	0.953	0.934	0.868	0.883	**0.918**	**0.920**	**0.026**	**0.020**
	MobileViT-xs	STRIP	0.736	0.710	0.533	0.549	0.912	0.859	0.390	0.526	0.460	0.526	0.465	0.592	0.379	0.526	0.554	0.613	0.184	0.118
		FreqDetector	0.989	0.955	0.966	0.904	0.834	0.763	0.996	0.972	0.510	0.526	0.980	0.940	0.896	0.824	**0.882**	0.841	0.161	0.146
		Ours	0.682	0.724	0.927	0.924	0.917	0.910	0.811	0.786	0.913	0.902	0.964	0.929	0.920	0.911	0.876	**0.870**	**0.090**	**0.075**
GTSRB	PreActResNet18	STRIP	0.871	0.840	0.883	0.849	0.991	0.981	0.310	0.501	0.356	0.501	0.778	0.791	0.641	0.625	0.690	0.727	0.247	0.173
		FreqDetector	0.981	0.939	0.993	0.960	0.964	0.901	0.925	0.848	0.483	0.503	0.595	0.562	0.544	0.548	0.784	0.752	0.213	0.188
		Ours	0.869	0.835	0.917	0.913	0.947	0.962	0.956	0.959	0.954	0.961	0.997	0.986	0.943	0.967	**0.940**	**0.940**	**0.036**	**0.048**
	MobileViT-xs	STRIP	0.947	0.939	0.875	0.856	0.962	0.937	0.285	0.501	0.438	0.501	0.616	0.687	0.544	0.552	0.667	0.710	0.246	0.184
		FreqDetector	0.981	0.939	0.993	0.960	0.922	0.840	1.000	0.999	0.471	0.511	0.870	0.784	0.544	0.548	0.826	0.797	0.207	0.182
		Ours	0.914	0.903	0.924	0.935	0.993	0.988	0.847	0.879	0.973	0.960	0.987	0.952	0.939	0.959	**0.940**	**0.939**	**0.047**	**0.034**
CIFAR100	PreActResNet18	STRIP	0.860	0.812	0.769	0.719	0.955	0.898	0.249	0.502	0.485	0.503	0.589	0.590	0.685	0.651	0.656	0.668	0.221	0.140
		FreqDetector	0.979	0.923	0.961	0.897	0.837	0.783	0.997	0.976	0.440	0.503	0.954	0.896	0.889	0.807	0.865	0.826	0.181	0.146
		Ours	0.939	0.921	0.939	0.945	0.834	0.838	0.878	0.873	0.971	0.959	0.913	0.826	0.968	0.968	**0.920**	**0.904**	**0.046**	**0.054**
	MobileViT-xs	STRIP	0.847	0.798	0.800	0.744	0.940	0.888	0.430	0.519	0.479	0.503	0.609	0.639	0.808	0.750	0.702	0.692	0.181	0.133
		FreqDetector	0.979	0.923	0.961	0.897	0.914	0.838	0.999	0.990	0.426	0.503	0.941	0.877	0.889	0.807	0.873	0.834	0.186	0.146
		Ours	0.905	0.909	0.946	0.957	0.972	0.967	0.940	0.932	0.898	0.881	0.991	0.965	0.956	0.955	**0.944**	**0.938**	**0.031**	**0.030**
Tiny-ImageNet	PreActResNet18	STRIP	0.852	0.788	0.949	0.892	0.995	0.976	0.430	0.504	0.681	0.640	0.511	0.545	0.767	0.722	0.741	0.724	0.198	0.162
		FreqDetector	0.710	0.652	0.999	0.989	0.920	0.828	1.000	0.996	0.655	0.617	0.960	0.910	0.992	0.958	0.891	0.850	0.135	0.147
		Ours	0.987	0.982	0.977	0.978	0.993	0.989	0.978	0.974	0.888	0.889	0.977	0.974	0.983	0.977	**0.969**	**0.966**	**0.033**	**0.032**
	MobileViT-xs	STRIP	0.737	0.688	0.872	0.809	0.991	0.964	0.421	0.516	0.647	0.615	0.585	0.655	0.766	0.716	0.717	0.709	0.174	0.134
		FreqDetector	0.689	0.638	0.998	0.984	0.938	0.865	1.000	0.999	0.631	0.602	0.770	0.696	0.982	0.934	0.858	0.817	0.146	0.156
		Ours	0.979	0.981	0.974	0.975	0.982	0.973	0.984	0.983	0.986	0.975	0.938	0.874	0.975	0.971	**0.974**	**0.962**	**0.015**	**0.036**
ImageNet200	WideResNet101-2	STRIP	0.921	0.869	0.959	0.903	-	-	0.421	0.502	0.584	0.567	0.693	0.668	0.765	0.695	0.724	0.701	0.186	0.146
		FreqDetector	0.526	0.520	0.998	0.986	-	-	1.000	1.000	0.484	0.517	0.979	0.949	0.994	0.970	0.830	0.824	0.230	0.216
		Ours	0.974	0.979	0.982	0.983	-	-	0.937	0.920	0.987	0.977	0.997	0.996	0.922	0.938	**0.966**	**0.965**	**0.027**	**0.027**
	SwinT-Base	STRIP	0.992	0.968	0.939	0.875	-	-	0.944	0.873	0.726	0.672	0.993	0.974	0.715	0.659	0.885	0.837	0.118	0.128
		FreqDetector	0.526	0.520	0.998	0.986	-	-	1.000	1.000	0.455	0.504	0.974	0.940	0.994	0.970	0.825	0.820	0.237	0.218
		Ours	0.978	0.978	0.978	0.979	-	-	0.990	0.988	0.985	0.970	0.999	0.998	0.980	0.975	**0.985**	**0.981**	**0.007**	**0.009**

* LF is computationally infeasible on ImageNet200.

Table 2. The evaluation results on different attacks, datasets, and backbones. The last two columns show the average performance and the standard deviation of performance across different attacks. The best results are in bold. We highlight that our method not only has good effectiveness, but also keeps outstanding stability (about 5 times of the runner-ups' on average) against different backdoor attacks including universal, sample-specific, and invisible ones.

Types↓ Attacks→	Badnets	Blended	LF	Input-aware	Wanet	LIRA	SSBA
Universal	✓	✓	✓				
Sample-specific				✓		✓	✓
Invisible					✓	✓	✓

Table 3. The backdoor attacks involved in our evaluations have covered the majority of trigger types.

Evaluation metrics. Two metrics are used: (1) The *Area Under Receiver Operating Curve* (AUROC), which is a widely-used metric to measure the trade-off between the false positive rate for clean samples and true positive rate for trigger samples for a detection method. (2) The F1 score. We calculate the best F1 score of detection methods to evaluate their optimal performance. The F1 score in our experiments is computed by:

$$\text{F1 score} = \max_{\gamma \in \Gamma} \frac{2 \times (\text{precision}_\gamma \times \text{recall}_\gamma)}{(\text{precision}_\gamma + \text{recall}_\gamma)}, \tag{5}$$

where Γ represents all possible thresholds.

We also include additional metrics such as FAR, FRR, and *Backdoored Data Rejection Rate* (BDR). For more implementation details and evaluations, please also refer to the supplementary.

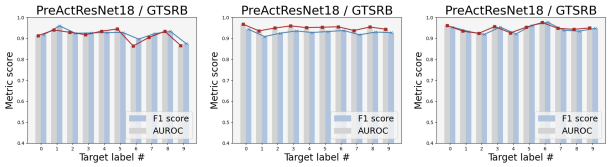
5.2. Effectiveness Studies

We first evaluate the performance of TeCo on different backdoor-infected models comprehensively. As shown in Tab. 2, TeCo can precisely detect the trigger samples in the inference stage with the average AUROC ≥ 0.876 for different backdoor attacks on diverse datasets and backbones. In addition, since in the real-world scenario, the TTSD methods should have solid effectiveness against different types of backdoor triggers, we investigate the stability of our method by calculating the standard deviations of its performance on different backdoor attacks with the same dataset and backbone. We highlight that TeCo achieves overall average AUROC $= 0.9433$ and F1 score $= 0.9386$, with the standard deviation of AUROCs and F1 scores equal to 0.0360 and 0.0364 respectively. These results indicate that TeCo outperforms the runner-up by about 10% in terms of AUROC, 14% in terms of F1-score, and achieves 5 times of stability against different types of trigger. In summary, our work maintains stable effectiveness among different trigger types, with only hard-label black-box model accessibility and no need for extra knowledge.

There are also some interesting results about baselines. Since the *Low-frequency* (LF) attack is designed to avoid FreqDetector [48], FreqDetector should have low effectiveness against this attack. However, we implement them following the official codes and find that if we let FreqDetector work in a binary classification manner and make judgments based on thresholds, it will perform well on LF attack. So we believe it is not unfair to involve LF attack and FreqDetector simultaneously in our experiments. Another interesting phenomenon is the success of STRIP against input-aware and LIRA attacks on SwinTransformer-base/ImageNet200. We further investigate it in our supplementary and show that the performance of STRIP is somehow influenced by the choice of backbones.

5.3. Ablation Studies

The impact of different target labels. We further evaluate the effectiveness of TeCo against different predefined



(a) Blended (b) SSBA (c) Wanet

Figure 3. Performance of TeCo against different target labels

target labels. We select 3 attacks (Blended, SSBA, and Wanet) from the seven attacks mentioned above to represent the universal, sample-specific, and invisible backdoor attacks, and make them attack 10 different labels that we randomly select from GTSRB. Thus we have 30 backdoor-infected models with different trojan labels and trigger types. Fig. 3 illustrates the stability of TeCo against backdoor attacks with different target labels. For Blended, TeCo remains AUROC $\geq$ 0.874, and the *standard deviation* (STD) of AUROC is about 0.021. For SSBA, TeCo achieves AUROC $\geq$ 0.908 and STD $\approx$ 0.010. For Wanet, TeCo achieves AUROC $\geq$ 0.919 and STD $\approx$ 0.017. These results support that target trojan labels have little influence on TeCo's performance.

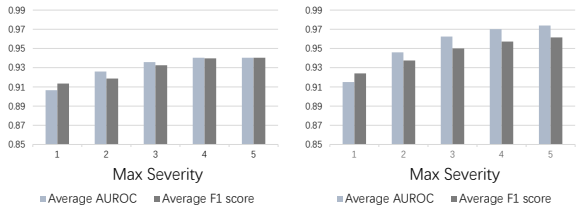
The impact of max severity N. We mark the computational cost of vanilla inference process as $O_1(1)$ and the average computational cost of image corruptions as $O_2(1)$. The computational cost of TeCo in the worst case is $O_1(N \times K) + O_2(N \times K)$, where K is the number of corruption types and N is the maximum of severity. Thus, given fixed corruption types, the max severity has a critical influence on the running efficiency. Here, we investigate TeCo's performance with N ranging from 1 to 5. As shown in Tab. 4, TeCo maintains good effectiveness and stability in different N. We illustrate some results in Fig. 4, which shows that TeCo's performance grows with the increasing max severity N, and still gets satisfying effectiveness with low N. In other words, TeCo also has competitive effectiveness when the computational cost is limited.

Model	Dataset	CIFAR10		GTSRB		CIFAR100		Tiny-ImageNet		ImageNet200	
	Metric	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD
CNNs	AUROC	0.918	0.000	0.930	0.012	0.884	0.055	0.970	0.000	0.949	0.019
	F1 score	0.914	0.001	0.929	0.010	0.879	0.032	0.966	0.000	0.943	0.017
ViTs	AUROC	0.868	0.004	0.929	0.005	0.936	0.012	0.954	0.050	0.975	0.002
	F1 score	0.857	0.004	0.931	0.006	0.928	0.016	0.946	0.033	0.969	0.003

Table 4. Performance of TeCo with a different maximum of severity. The average and standard deviation results suggest that TeCo has high effectiveness in different max severity N and maintains stable performance among different N.

5.4. Thresholds

Since TeCo maps the input image x to a linearly separable space and defenders make judgments by a threshold γ, questions are how we can get this threshold and what is the influence of threshold for our method. Here, we evaluate TeCo by setting an empirical threshold directly, which does not break the "no need for extra data" characteristic



(a) PreActResNet18 / CIFAR10 (b) MobileViT-xs / Tiny-Imagenet

Figure 4. Illustration of TeCo's performance with different max severity. Despite TeCo's performance grows with the increasing max severity N, TeCo still has good performance with low N.

of TeCo. We use ACC as the evaluation metric, which is calculated by:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{6}$$

Tab. 5 shows the average performance of TeCo in different attacks, datasets, and backbones when an empirical threshold is given. The results suggest that even in the worst case where no data is available for defenders to estimate an appropriate threshold, by empirically setting threshold $= 1$, TeCo can still get an average ACC $\approx$ 0.79, which still surpasses STRIP's performance in optimal thresholds shown in Tab. 2 and is competitive to FreqDetector. We defer a more detailed discussion about thresholds in other different settings to the supplementary.

	CIFAR10	GTSRB	CIFAR100	Tiny-ImageNet	ImageNet200	AVG
CNNs	0.8521	0.9242	0.7735	0.6504	0.7613	0.7923
ViTs	0.8018	0.8366	0.7778	0.7569	0.7610	0.7868

Table 5. The accuracy of TeCo in the settings where only one empirical threshold ($\gamma = 1$) can be set for all attacks

6. Analyses

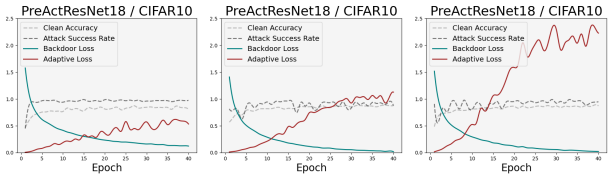
6.1. Constructing Adaptive Attacks against TeCo

As formulated by Eq.(2), the goal of backdoor attacks is to make models perform normally on clean data but give a specific prediction on trigger samples, the classic loss function for training such models can be defined as:

$$\mathcal{J}_{bd} = \sum_{i=1}^{I} CE(C_\theta(x_i), y_i) + \sum_{j=1}^{J} CE(C_\theta(\hat{x_j}), y_t), \tag{7}$$

where $CE(\cdot)$ represents the cross entropy loss function. This backdoor loss function is widely used in backdoor attacks. However, we speculate that this dual-target loss function leads backdoor-infected models to act anomalously on trigger samples in terms of corruption robustness. To reveal this point, we first introduce an adaptive loss to attack our method TeCo:

$$\mathcal{J}_{ada} = \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{n=1}^{N} MSE(MSE(C_\theta(x_j), C_\theta(D_n^k(x_j))), \\ MSE(C_\theta(\hat{x_j}), C_\theta(D_n^k(\hat{x_j})))), \tag{8}$$



(a) Badnets (b) LF (c) SSBA

Figure 5. Visualization of backdoor loss, adaptive loss, clean accuracy, and attack success rate in the training process. We note that with the drop of backdoor loss, the adaptive loss rises correspondingly, which indicates a negative correlation between them.

Weight→	0		10^{-3}		10^{-4}		10^{-5}	
Attack↓	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score
BadNets	0.9112	0.9174	0.5763	0.5928	0.6571	0.6542	0.6745	0.6657
LF	0.9390	0.9367	0.8592	0.8483	0.9219	0.9154	0.8667	0.8858
SSBA	0.8683	0.8835	0.7125	0.7312	0.6477	0.7281	0.5909	0.6852

Weight→	0		10^{-3}		10^{-4}		10^{-5}	
Attack↓	C.ACC	ASR	C.ACC	ASR	C.ACC	ASR	C.ACC	ASR
BadNets	0.9153	0.9502	0.5105	0.7386	0.7980	0.3720	0.8546	0.3001
LF	0.9286	0.9888	0.8022	0.9443	0.8864	0.9504	0.8962	0.9476
SSBA	0.9270	0.9719	0.7129	0.9176	0.8925	0.9162	0.8978	0.9170

Table 6. Performance of TeCo against adaptive backdoor attacks

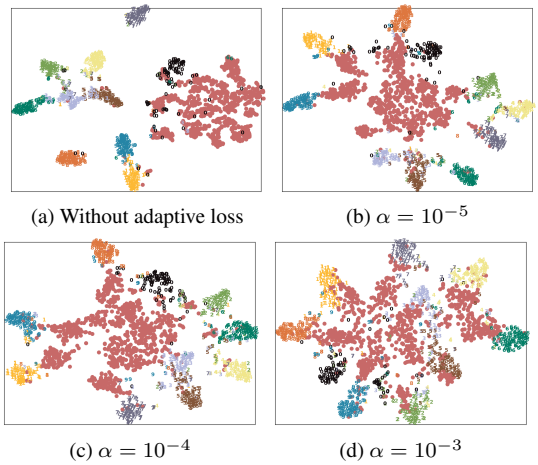
where x_j is the original version of the trigger sample $\hat{x_j}$. This adaptive loss aims to make models have the same corruption robustness on corrupted clean samples and corrupted trigger samples, which is aligned to the inference logic of TeCo.

6.2. Results of Adaptive Attacks

We convert Badnets, LF, and SSBA to the adaptive version by applying our adaptive loss in their training process, and investigate these adaptive attacks on PreActResNet18/CIFAR10. We first monitor the adaptive loss without derivative in the training phases. As illustrated in Fig. 5, the adaptive loss grows[5] when the backdoor loss decreases, which means the success on the dual-target loss function may drive the model to behave differently in terms of corruption robustness. A reasonable hypothesis is the model learns shortcuts [13] for the backdoor trigger guided by the backdoor loss function, however, this trigger is not always robust in image space when facing different image corruptions. These results support TeCo's effectiveness by showing the negative correlation between backdoor loss and adaptive loss. And since the involved attacks contain different characteristics, such as partial (Badnets), global (LF), universal (Badnets, LF), sample-specific (SSBA), and invisible (SSBA), we suppose that this negative correlation is hard to be avoided by changing trigger types, which confirms the stability of TeCo on the other hand.

Then we add the adaptive loss to the overall loss function: $\mathcal{J} = \mathcal{J}_{bd} + \alpha \mathcal{J}_{ada}$, where α is the weight factor. Tab. 6 shows the TeCo's effectiveness against adaptive attacks and the performance of adaptive attacks. The results

[5]We scale the adaptive loss down to fit the figure by multiplying 10^{-3}.



(a) Without adaptive loss (b) $\alpha = 10^{-5}$ (c) $\alpha = 10^{-4}$ (d) $\alpha = 10^{-3}$

Figure 6. The red points represent the trigger samples, and the black points are clean samples from the target class. The points in other colors are clean samples from other classes.

indicate that the adaptive attacks can avoid TeCo to some degree, however they sacrifice attack performance once applying the adaptive loss.

Since SSBA has the best performance in Tab. 6, we further visualize the clean and trigger samples in the latent space of the SSBA-infected model. As illustrated in Fig. 6, the adaptive loss pushes the trigger samples from the edge of latent space to the center, making them have a similar distance to different clean samples. Thus, a possible way to attack TeCo is to embed trigger samples in the middle of the latent space. However, this may be hard to achieve as we have shown in Tab. 7, the proposed adaptive attack is not stable enough on different datasets.

Dataset	AUROC(↑)	F1 score(↑)	ACC(↑)	FAR(↓)	FRR(↓)	BDR(↑)
GTSRB	0.9101	0.8900	89.00	13.09	8.90	91.10
CIFAR100	0.9141	0.9137	91.37	5.76	11.54	88.46

Table 7. TeCo against adaptive SSBA attack (10^{-5}) on PreActResNet18

7. Conclusions, Limitations, and Future Work

In this paper, we propose TeCo, the first test-time trigger sample detection method that only needs the hard-label outputs of the victim models without requiring extra data or assumptions. Extensive experiments support that TeCo has outstanding effectiveness and stability against different backdoor attacks. However, a limitation of TeCo is that using multiple image corruptions will increase the computational cost. Therefore, designing an effective and efficient single corruption function will be our future work.

Acknowledgments. Shengshan's work is supported in part by the National Natural Science Foundation of China (Grant No.U20A20177) and Hubei Province Key R&D Technology Special Innovation Project under Grant No.2021BAA032. Shengshan Hu is the corresponding author.

References

[1] Sérgio Branco, André G. Ferreira, and Jorge Cabral. Machine learning in resource-scarce embedded systems, FPGAs, and end-devices: A survey. *Electronics*, 8(11):1289, 2019. 2

[2] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018. 3

[3] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 2, 3, 5, 6

[4] Yao Chen, Jiong He, Xiaofan Zhang, Cong Hao, and Deming Chen. Cloud-DNN: An open framework for mapping DNN models to cloud FPGAs. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-programmable Gate Arrays (FPGA'19)*, pages 73–82, 2019. 2

[5] Edward Chou, Florian Tramèr, Giancarlo Pellegrino, and Dan Boneh. Sentinet: Detecting physical attacks against deep learning systems. *arXiv preprint arXiv:1812.00292*, 2018. 1, 2, 3

[6] Kien Do, Haripriya Harikumar, Hung Le, Dung Nguyen, Truyen Tran, Santu Rana, Dang Nguyen, Willy Susilo, and Svetha Venkatesh. Towards effective and robust neural trojan defenses via input filtering. *arXiv preprint arXiv:2202.12154*, 2022. 3

[7] Bao Gia Doan, Ehsan Abbasnejad, and Damith C. Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Proceedings of the 36th Annual Computer Security Applications Conference (ACSAC'20)*, pages 897–912, 2020. 3

[8] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*, pages 11966–11976, 2021. 2, 3, 5, 6

[9] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (CVPR'21)*, pages 16482–16491, 2021. 1

[10] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*, 2020. 2, 3

[11] Jeremy Fowers, Kalin Ovtcharov, Michael Papamichael, Todd Massengill, Ming Liu, Daniel Lo, Shlomi Alkalay, Michael Haselman, Logan Adams, Mahdi Ghandi, Stephen Heil, Prerak Patel, Adam Sapek, Gabriel Weisz, Lisa Woods, Sitaram Lanka, Steven K. Reinhardt, Adrian M. Caulfield, Eric S. Chung, and Doug Burger. A configurable cloud-scale DNN processor for real-time AI. In *Proceedings of the 45th ACM/IEEE Annual International Symposium on Computer Architecture (ISCA'18)*, pages 1–14, 2018. 2

[12] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC'19)*, pages 113–125, 2019. 1, 2, 3, 5

[13] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 8

[14] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. 1, 2, 3, 5, 6

[15] Junfeng Guo, Ang Li, and Cong Liu. AEVA: black-box backdoor detection using adversarial extreme value analysis. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*, 2022. 1, 2, 4

[16] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: defending against backdoor attacks using robust statistics. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, volume 139, pages 4129–4139, 2021. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proceedings of the 2016 European Conference on Computer Vision (ECCV'16)*, pages 630–645, 2016. 5

[18] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Proceedings of the 7th International Conference on Learning Representations (ICLR'19)*, 2019. 2, 3, 5

[19] Ehsan Hesamifard, Hassan Takabi, Mehdi Ghasemi, and Rebecca N. Wright. Privacy-preserving machine learning as a service. *Proc. Priv. Enhancing Technol.*, 2018(3):123–142, 2018. 2

[20] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN'13)*, pages 1–8, 2013. 5

[21] Shengshan Hu, Ziqi Zhou, Yechao Zhang, Leo Yu Zhang, Yifeng Zheng, Yuanyuan He, and Hai Jin. Badhash: Invisible backdoor attacks against deep hashing with clean label. In *Proceedings of the 30th ACM International Conference on Multimedia (MM'22)*, pages 678–686, 2022. 2

[22] Kaidi Jin, Tianwei Zhang, Chao Shen, Yufei Chen, Ming Fan, Chenhao Lin, and Ting Liu. Can we mitigate backdoor attack using adversarial detection methods? *IEEE Transactions on Dependable and Secure Computing*, 2022. 3

[23] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 5

[24] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[25] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 4

[26] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*, pages 16443–16452, 2021. 1, 2, 3, 5, 6

[27] Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Rethinking the trigger of backdoor attack. *arXiv preprint arXiv:2004.04692*, 2020. 2, 4

[28] Guanxiong Liu, Abdallah Khreishah, Fatima Sharadgah, and Issa Khalil. An adaptive black-box defense against trojan attacks (trojdef). *arXiv preprint arXiv:2209.01721*, 2022. 2, 3, 4

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*, pages 10012–10022, 2021. 5

[30] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "Beatrix" resurrections: Robust backdoor detection via gram matrices. *arXiv preprint arXiv:2209.11715*, 2022. 1, 2, 3, 5

[31] Sachin Mehta and Mohammad Rastegari. Mobilevit: Lightweight, general-purpose, and mobile-friendly vision transformer. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*, 2022. 5

[32] Tuan Anh Nguyen and Anh Tuan Tran. Input-aware dynamic backdoor attack. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS'20)*, pages 3454–3464, 2020. 1, 2, 3, 5, 6

[33] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*, 2021. 2, 3, 5, 6

[34] Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (AsiaCCS'21)*, pages 363–377, 2021. 2, 4

[35] Mauro Ribeiro, Katarina Grolinger, and Miriam A. M. Capretz. MLaaS: Machine learning as a service. In *Proceedings of the IEEE 14th International Conference on Machine Learning and Applications (ICMLA'15)*, pages 896–902, 2015. 2

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5

[37] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. In *Proceedings of the 7th European Symposium on Security and Privacy (EuroS&P'22)*, pages 703–718, 2022. 3

[38] Hossein Souri, Micah Goldblum, Liam Fowl, Rama Chellappa, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *arXiv preprint arXiv:2106.08970*, 2021. 1

[39] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of DNNs for robust backdoor contamination detection. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security'21)*, pages 1541–1558, 2021. 1, 2, 3

[40] Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, pages 13368–13378, 2022. 1

[41] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS'18)*, 31, 2018. 3

[42] Sakshi Udeshi, Shanshan Peng, Gerald Woo, Lionell Loh, Louth Rawshan, and Sudipta Chattopadhyay. Model agnostic defence against backdoor attacks in machine learning. *IEEE Transactions on Reliability*, 2022. 1, 2, 3

[43] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP'19)*, pages 707–723, 2019. 1

[44] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 5

[45] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, Chao Shen, and Hongyuan Zha. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Proceedings of the NeurIPS 2022 Track Datasets and Benchmarks*, 2022. 5

[46] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting AI trojans using meta neural analysis. In *Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP'21)*, pages 103–120, 2021. 1

[47] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the 2016 British Machine Vision Conference, (BMVC'16)*, 2016. 5

[48] Yi Zeng, Won Park, Z. Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*, pages 16473–16481, 2021. 2, 3, 5, 6

[49] Zhendong Zhao, Xiaojun Chen, Yuexin Xuan, Ye Dong, Dakui Wang, and Kaitai Liang. Defeat: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, pages 15213–15222, 2022. 2