Orion: Online Backdoor Sample Detection via Evolution Deviance

Huayang Huang[1], **Qian Wang**[1*], **Xueluan Gong**[2] and **Tao Wang**[1]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University, Hubei, China
[2]School of Computer Science, Wuhan University, Hubei, China
{hyhuang, qianwang, xueluangong, WTBantoeC}@whu.edu.cn

Abstract

Widely-used DNN models are vulnerable to backdoor attacks, where the backdoored model is only triggered by specific inputs but can maintain a high prediction accuracy on benign samples. Existing backdoor input detection strategies rely on the assumption that benign and poisoned samples are separable in the feature representation of the model. However, such an assumption can be broken by advanced feature-hidden backdoor attacks. In this paper, we propose a novel detection framework, dubbed Orion (online backdoor sample detection via evolution deviance). Specifically, we analyze how predictions evolve during a forward pass and find deviations between the shallow and deep outputs of the backdoor inputs. By introducing side nets to track such evolution divergence, Orion eliminates the need for the assumption of latent separability. Additionally, we put forward a scheme to restore the original label of backdoor samples, enabling more robust predictions. Extensive experiments on six attacks, three datasets, and two architectures verify the effectiveness of Orion. It is shown that Orion outperforms state-of-the-art defenses and can identify feature-hidden attacks with an F1-score of 90%, compared to 40% for other detection schemes. Orion can also achieve 80% label recovery accuracy on basic backdoor attacks.

1 Introduction

Deep neural networks (DNNs) have achieved tremendous success in various learning tasks, such as face authentication, autonomous driving, and disease diagnosis [Parkhi *et al.*, 2015; Redmon *et al.*, 2016; Rajkomar *et al.*, 2018]. However, due to the diversity of data sources and the difficulty of performing sanity checks on off-the-shelf models, recent work has revealed that DNN models are vulnerable to backdoor attacks [Gu *et al.*, 2019]. Backdoor attacks are supply-chain attacks that insert malicious functionality into the model by poisoning the training data or manipulating the model weight. At inference time, the model misclassifies any input stamped with the trigger to the target label while maintaining normal function on clean data. A recent survey of 28 industry companies [Kumar *et al.*, 2020] viewed the backdoor attack as the most severe threat to machine learning systems for model users. Therefore, it is urgent to design an effective way to identify the backdoor samples for reliable model deployment.

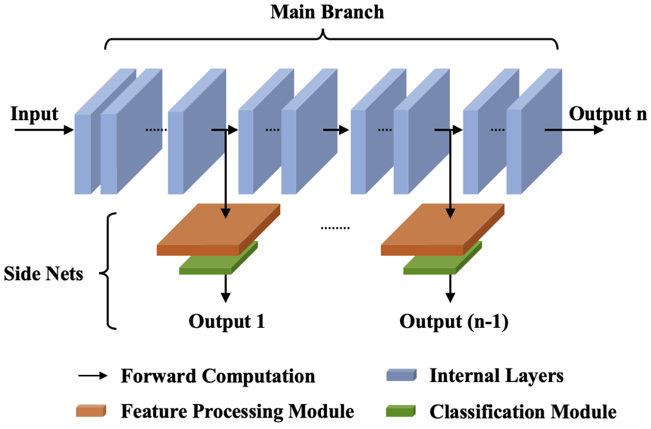


Figure 1: Multi-exit branchy network with side nets.

A long line of works [Tran *et al.*, 2018; Hayase *et al.*, 2021] identifies abnormal samples in the training set. However, they typically require the knowledge of the approximate poisoning rate or a large number of data to implement clustering algorithms [Chen *et al.*, 2019b], which is not applicable in the online scenario. State-of-the-art studies [Tang *et al.*, 2021; Ma *et al.*, 2023] on online sample detection use the feature map extracted from the last convolution layer to distinguish backdoor samples from normal ones and regard backdoor input identification as out-of-distribution (OOD) sample detection. However, the static feature map neglects the information from shallow layers and the strong assumption of latent separability can be easily broken by recent feature-hidden attacks. Moreover, recognizing OOD samples requires a large amount of clean data to simulate the feature distribution of benign images, making it challenging to implement in practice.

Recent studies [Luo *et al.*, 2016] have shown that shallow and deep layers of DNNs focus on features of different granularity due to various receptive fields. Backdoors are typically implanted in the deeper layers, while normal

[*]Qian Wang is the corresponding author. This work was partially supported by the National Key R&D Program of China (2020AAA0107701), and the NSFC under Grants U20B2049 and U21B2018.

features predominate in an early stage of the model [Cai *et al.*, 2022]. This distribution characteristic can be captured by making full use of the features in all layers, rather than relying solely on the last feature map of the model. However, there are some challenges to leveraging internal features. **(C1)** The total number of internal feature parameters is too large to analyze. **(C2)** Feature representations include many task-unrelated noise activations, which may interfere with detection. **(C3)** When the original classification task is challenging, benign features may also be learned only at deeper layers, which can be confused with backdoor features.

To address these challenges, we propose Orion, which uses internal classifications as an alternative to inner features and identifies malicious inputs by prediction divergence during the forward pass. We first introduce side nets (S-Nets) at different stages of the model to construct a multi-exit branchy network, as shown in Figure 1. S-Nets allow samples to exit the network early and output prediction results. By introducing internal classifiers, we can reduce the feature dimension and the irrelevance to the classification task **(C1-2)**. Then we design the outlier score metric using the *consistency*, *stability*, and *determinacy* of the side outputs and identify backdoor samples by anomaly detection **(C3)**.

In summary, this paper makes the following contributions.

- We investigate how prediction evolves during the forwarding process and find deviations between the shallow and deep predictions of backdoor samples. Based on that, we design a backdoor detection strategy Orion that can identify malicious inputs on-the-fly by leveraging a multi-exit branchy network.
- We propose a recovery scheme to restore the original label of backdoor inputs, with 80% accuracy on basic backdoor attacks. To the best of our knowledge, we are the first to perform label recovery without modifying the samples or the model.
- We validate the effectiveness of Orion under six different attacks on three datasets and two model architectures. Compared with baseline defenses, Orion can achieve the best detection performance against all state-of-the-art backdoor attacks. Orion is also the only backdoor detection solution that can successfully identify samples of feature-hidden attacks, with an F1-socre of 90%.

2 Related Work

2.1 Backdoor Attacks

Backdoor attack makes a model produce a malicious function when it encounters a specific trigger by tampering with the training process. BadNets [Gu *et al.*, 2019] is the first backdoor attack that proposes to implant backdoors by data poisoning. Subsequent works consider the invisibility of triggers. Blended [Chen *et al.*, 2017] proposed a more covert backdoor attack by not overlaying the triggers directly onto the original image but fusing them with a certain percentage. Further, WaNet [Nguyen and Tran, 2021] uses a small and smooth wrapping field to generate backdoor samples for visual concealment. More recent work has suggested a more sophisticated way of generating poisoning samples. IAD [Nguyen and Tran, 2020] uses an input-related generator to create a unique trigger for each sample, also called the sample-specific trigger. There are also works that consider more complex backdoor implantation. [Tang *et al.*, 2021] proposes input-dependent backdoor attacks TaCT, where the trigger is only valid when added to images of a specific source class. The latest feature-hidden attack [Zhong *et al.*, 2022] considers not only the visual concealment of the poisoned data but also the consistency of the model representation. They require backdoor samples to maintain similarity in feature representation with normal ones, which poses a new challenge to defense methods.

2.2 Backdoor Detection

Since a successful backdoor attack requires tampered data and a malicious model, detection methods can be divided into model diagnostics and anomaly sample detection. Model diagnostics aims to determine whether the model is backdoored or not. ABS [Liu *et al.*, 2019] identifies abnormal activations that significantly affect classification. ULP [Kolouri *et al.*, 2020] and MNTD [Xu *et al.*, 2021] use a data-driven approach to determine whether a model is malicious. Recent works [Liu *et al.*, 2022; Wang *et al.*, 2022] utilize trigger reverse-engineering for backdoored model detection.

Existing methods for backdoor input detection rely on the separability of benign and malicious samples on the feature representation. AC [Chen *et al.*, 2019b] performs a two-class clustering algorithm on the activation outputs of the penultimate layer. Spectral [Tran *et al.*, 2018] finds a detectable trace called spectral signature in backdoor samples, and Spectre [Hayase *et al.*, 2021] further amplifies the signature using robust covariance estimation. SCAn [Tang *et al.*, 2021] uses image segmentation to identify statistical inconsistencies in backdoor inputs. Latest work Beatrix [Ma *et al.*, 2023] leverages Gram matrix to compute high-dimensional information of the feature map for anomaly detection. Our work focuses on online sample detection, where we need to determine whether an input is malicious or not on-the-fly, without relying on any information about attack strategies or the hypothesis of latent separability.

2.3 Multi-exit Network

Multi-exit network allows classification results for inputs to exit the network early. It is based on the observation that some input samples can be correctly classified before reaching the last layer and more layers may lead to over-learning and waste of resources. The multi-exit network can realize adaptive inference based on the input. It is widely used to improve the accuracy, robustness, and efficiency of the model prediction. SDN [Kaya *et al.*, 2019] uses a confidence-based early exit strategy to alleviate the problem of overthinking. Using two-stage optimization, LSLP [Chen *et al.*, 2020] learns both the model parameters and a termination strategy. Further, [Hu *et al.*, 2020] proposes a robust multi-output network that can resist adversarial attacks. Moreover, Meta-GF [Sun *et al.*, 2022] presents a meta-learning-based training algorithm to train multiple exits harmoniously. We use the multi-exit network to monitor the evolution of classification results.

3 Methodology

3.1 Threat Model

We consider the same threat model in the most recent work [Ma *et al.*, 2023]. The attacker has complete knowledge of the model and can arbitrarily modify the training set to make the attacks successful. There is no limit on the types of triggers or the ratio of poisoning data. The attacker serves as a powerful malicious model provider. The defender is the model user that has white-box access to the backdoored model, which is the same as current detection strategies [Hayase *et al.*, 2021; Tang *et al.*, 2021; Ma *et al.*, 2023]. We assume the defender has a small set of clean reference data for detection. The goal of the defender is to identify the malicious input on-the-fly and ensure the accuracy and robustness of the prediction.

3.2 Overview

The overall framework of Orion is shown in Figure 2. The detection mainly consists of three steps. (i) Attaching and training S-Nets: Given an off-the-shelf backdoored model, the defender first attaches some side nets to the model to make it a multi-output branchy network. Then the holdout clean data is used to train the newly added side branches without modification to the original model. (ii) Calculating the outlier score: For each sample fed into the network, we calculate the outlier score of the sample by leveraging the output of each branch. (iii) Anomaly detection. Finally, we choose the threshold for determining which samples are poisoned based on the outlier score. For the identified abnormal samples, users can discard them or restore their original labels using the output of shallow branches. The corrected samples can be further utilized to retrain the model for purification. Since we consider the scenario where a user with limited computing resources tends to use an untrusted third-party model, the user may not have enough resources to retrain the whole model. So in this work, we mainly focus on detecting backdoor samples.

Notation. Let $(x, y) \in (\mathrm{X}, \mathrm{Y})$ be an input sample to the model, where x is the input picture in the image classification task and $y \in \{1, ..., c\}$ denotes its ground-truth label, c is the total number of classes. For a model contains n internal layers, the classification process of x can be expressed as $\mathrm{F_c}(f_n(...f_2(f_1(x))))$, which outputs the probability that x belongs to each category. Here, f is the feature extractor consisting of convolution layers, activation layers, and pooling layers. $\mathrm{F_c}$ represents the classifier which typically consists of multiple fully connected layers with a softmax function. For simplicity, we abbreviate the activation output at layer i as f_i. The i-th S-Net takes $f_i(x)$ as input and outputs the classification results of x based on the shallow i layers.

3.3 Attaching and Training S-Nets

The branchy network is constructed by integrating S-Nets into internal layers. Each S-Net comprises a feature processing module and a classification module. The size of features extracted from convolution blocks at various depths varies, and excessively large features are not conducive to classifier training. To mitigate this issue, we leverage the feature processing module to reduce the feature parameters of different layers to an appropriate size. The classification module takes the reduced features as input and performs the same classification function as the original task.

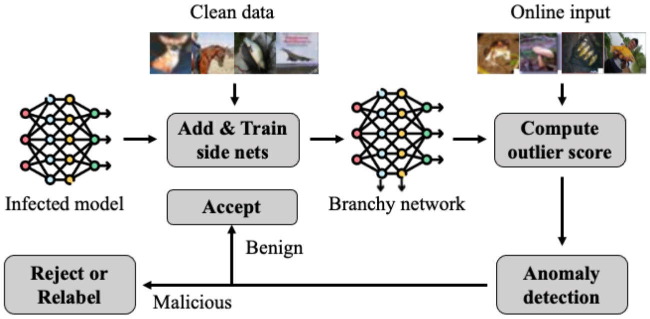


Figure 2: Pipeline of Orion.

Feature Processing Module. The feature processing module in S-Nets aims to reduce the size of $f_i(x)$ for classification. We employ a pooling strategy mixed with max pooling and average pooling [Lee *et al.*, 2016], which can learn the mixing proportion parameters from the data without manual guidance. We set the pooling size as 4x4, which means that any feature map larger than this will be pooled into 4x4, while smaller feature maps will remain unchanged.

Classification Module. For the design of the classification module, we opt to use a single fully connected layer. Such a simple design can make the classification results of S-Nets mainly depend on the features extracted from the main branch network so as to better monitor the prediction evolution of the backdoor samples in the forward propagation process. We use the clean reference set to train the parameters of the S-Nets by backpropagation. The training loss of the i-th S-Net is defined as

$$\mathcal{L}_i = \mathcal{L}_{CE}(F_i(M_i(f_i(x))), y) \tag{1}$$

where M_i and F_i mean the mixed pooling function and classification function of the i-th S-Nets. $\mathcal{L}_{CE}$ is the cross entropy loss. We simplify the output of the i-th S-Net to $F_i(x)$.

Location of the S-Nets. Regarding the placement of the S-Nets, we opt for equally spaced internal layers to implant S-Nets. Intuitively, if the S-Nets are only distributed in the shallow layers, the classification accuracy of both normal and backdoor samples may be very low, and both deviate from the final output. On the contrary, if they are all added at the deep level, it is possible that the learning is converged and the outputs of the S-Nets tend to be consistent with the last one. As a result, it is difficult to distinguish the two types of inputs, which is also verified by our experiments.

3.4 Outlier Score Definition

As mentioned above, we consider a multi-exit branchy network with $(n-1)$ S-Nets and n outputs. For normal input, the dominant features of different convolutional layers are consistent, representing features of the target objects. The outputs of different S-Nets are similar, with the accuracy increasing with the depth of the network and approaching the final output of the main branch network. Since the backdoor sample is generated by superimposing the normal sample and the trigger, it contains both the characteristics of the original class and the target class. Different S-Nets may focus on distinct areas of the images, resulting in variations in branch outputs. We find that the side outputs of backdoor samples have three distinct characteristics in the forward propagation process that are not present in the normal inputs. We exploit these differences to design our metrics.

Consistency. First, we find that shallow S-Nets cannot identify backdoors, and misclassification is only achieved in deeper layers. However, most of the normal features can be learned at an early stage. So, we calculate the *consistency* between the output of S-Nets and that of the main branch network

$$\Phi_c(x) = \sum_{i=1}^{n-1} ||F_i(x) - F_n(x)||_1, \tag{2}$$

which is the sum of L_1 distance between all S-Nets outputs and the final result. The L_1 distance can characterize both magnitude and direction changes of the predictions. However, normal samples that are hard to learn may also lead to shallow misclassification. So we further consider the *stability* and *determinacy* of the inner predictions.

Stability. *Stability* measures the frequency and magnitude of changes in the outputs of branch nets. The outputs of the backdoor input are more variable during the forwarding process. However, variations in the output of benign samples can also arise during the learning process due to the limited capability of shallow S-Nets. As the depth of the network increases, the ground truth label becomes more confident, while the confidence of other labels decreases. In order to reduce this effect, we only consider the direction change of the internal predictions and compute the cosine similarity of adjacent S-Nets outputs as below

$$\Phi_s(x) = \sum_{i=1}^{n-1} (1 - cossim(F_i(x), F_{i+1}(x))). \tag{3}$$

Determinacy. *Determinacy* indicates how certain the network is about the current classification result. When the internal classification result matches the final output, the changes in confidence levels are consistent across different samples. As learning proceeds, a well-functioning network tends to classify samples with increasing certainty into the labeled class. So we focus on the confidence level when S-Nets' outputs deviate from the main branch result. We observe that backdoor samples exhibit higher false confidence than normal inputs, which can be attributed to the competitive nature of the original and trigger features in the classification process. As a result, the backdoor sample may be classified with high confidence as the original class in the shallow layers, and as the target class in deeper layers, leading to a higher false confidence. On the contrary, normal inputs tend to be less certain when misclassified, so all categories have low confidence. We denote the output label of the i-th S-Net as $P_i(x) = \underset{j \in \{1,\ldots,c\}}{argmax} F_i(x)$. *Determinacy* is defined as

$$\Phi_d(x) = \sum_{i=1}^{n-1} max(F_i(x)) \mathbb{I}(P_i(x) \neq P_n(x)), \tag{4}$$

where $\mathbb{I}$ is the indicator function and max outputs the largest predicted confidence. The *stability* and *determinacy* metrics facilitate differentiation between hard-to-learn normal features and backdoor features in deep layers (**C3**). So the final outlier score of x is

$$\Phi(x) = \alpha * \Phi_c(x) + \beta * \Phi_s(x) + \gamma * \Phi_d(x), \tag{5}$$

in which α, β and γ are hyperparameters to tradeoff between the three matrics. We first amplify the three elements such that the benign samples have equal means on the three factors. Then due to the different separability of the three metrics, we make the amplified consistency : determinacy : stability as $2 : 2 : 1$. The resulting parameters are similar across different datasets and attacks. In this paper, we set $\alpha : \beta : \gamma$ as $1 : 2 : 6$.

3.5 Rejecting and Relabelling Strategy

Anomaly Detection. Since normal and backdoor samples can be well distinguished in terms of the outlier score, we can use clean data to determine the detection threshold. We select the threshold value that can make 95% of the reference data pass the detection, which means any sample with an outlier score larger than the threshold will be judged as malicious. This is a straightforward anomaly detection algorithm, thanks to the strong separability of the outlier score, and the malicious values are significantly larger than the benign ones.

Label Recovery. As mentioned above, backdoors tend to take effect in deep layers, while shallow layers may capture the original features of the sample. So we can use the outputs of shallow S-Nets to estimate the original class of backdoor inputs. We use the plurality voting mechanism for label recovery. Specifically, we choose the label with the highest number of occurrences for all internal predictions that differ from the output of the main branch network. If there are multiple candidate classes, the one with the highest confidence is selected. The label recovered for input x is denoted as

$$R(x) = \underset{j \in \{1,\ldots,c\}}{argmax} \sum_{i=1}^{n-1} \mathbb{I}(P_i(x) = j \text{ and } P_n(x) \neq j) \tag{6}$$

4 Evaluations

4.1 Setup

Datasets and Architectures. We perform experiments on three datasets CIFAR-10, GTSRB and Tiny-Imagenet [Krizhevsky and Hinton, 2009; Stallkamp *et al.*, 2012; Le and Yang, 2015]. CIFAR-10 and GTSRB have an image size of 32x32 and contain 10 and 43 classes, respectively. CIFAR-10 has 50,000 and 10,000 samples for training and testing. GTSRB contains 39,209 training and 12,630 validation images. Tiny-Imagenet is a subset of ImageNet, containing 200 classes. Each class contains 500 training data and 50 test samples. The image size is 64x64 pixels. For model architectures, we employ two types of CNNs: VGG [Parkhi *et al.*,

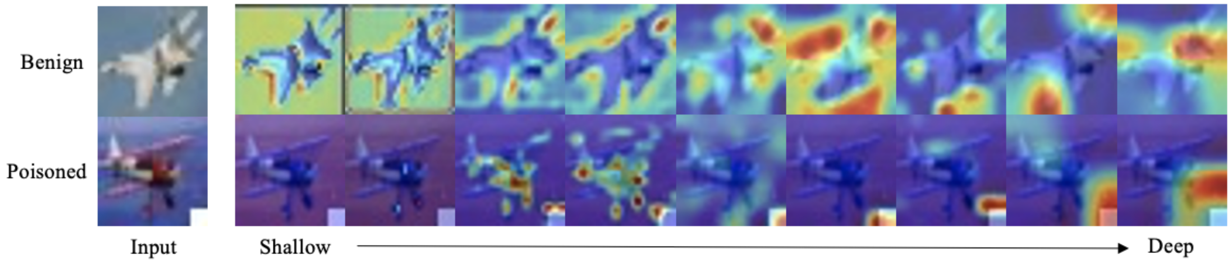


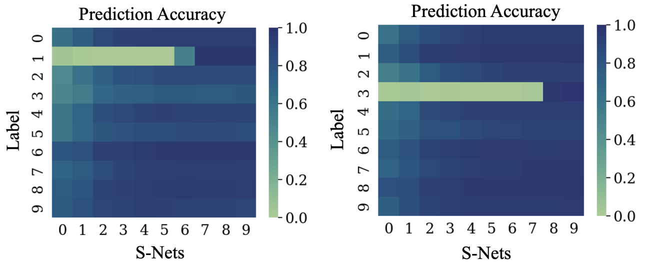
Figure 3: Visualization of attention over layers. Red indicates important parts for classification, and blue represents less critical parts. The top row is the result of a benign sample, and the bottom row is the result of poisoned input with a trigger in the lower right corner of the image.

2015] with batch normalization and ResNet [He *et al.*, 2016]. Specifically, we use VGG16-bn for CIFAR-10 and GTSRB and ResNet-56 for Tiny-Imagenet.

Attacks and Baselines. We verify the effectiveness of Orion on six backdoor attacks: BadNets, Blended, WaNet, IAD, TaCT, and Feature attack [Zhong *et al.*, 2022]. They cover a variety of trigger patterns: patches, visually hidden global noise, sample-specific modifications, and feature-hidden perturbations. We also compare with Spectre, SCAn, and Beatrix, three state-of-the-art backdoor sample detection methods. They all require a small clean reference dataset for detection, which is the same as Orion. All the attack and defense methods are illustrated in Section 2. For the attacks, we set the poisoning rate no higher than 0.1, transparency in Blended as 0.2, the cross rate in IAD as 0.1, and the cover rate in TaCT as 0.05. Other attack parameters use the default setting in their original papers. Backdoored models are trained with poisoned data from scratch for 50-200 epochs. Without an additional illustration, all defense methods have 1% of the training set as a clean reference set for sample detection. More details about the setting and performance of the attacks are shown in Appendix A.

Training Details and Metrics. We adopt the Adam optimizer to train each S-Net for 25 epochs, with a learning rate of 0.001. The clean reference set is augmented for training. We use precision (PRE) and recall (REC) rates to measure the effectiveness of backdoor sample identification. PRE is defined as the ratio of accurately identified backdoor samples to the total number of identified malicious inputs. REC is calculated by dividing the number of correctly identified backdoor images by the total number of poisoning samples. We further consider the F1-score (F1), the weighted harmonic mean of the two metrics, to evaluate the detection performance.

4.2 Design Rationale Verification



(a) BadNets

(b) IAD

Figure 4: Prediction accuracy over layers. The target classes for BadNets and IAD are 1 and 3, respectively.

First, we verify the intuition that backdoor features tend to locate at deeper layers, making the outputs of shallow S-Nets diverge from the final classification. We use Grad-Cam [Selvaraju *et al.*, 2017] to visualize the attention of classification in different layers, i.e., which parts of the image play an essential role in the prediction. Figure 3 shows that for regular input, the attention is gradually focused on the target object from shallow to deep layers. In contrast, backdoor samples focus on both the object and the trigger in shallow layers, while in deeper layers, all attention is shifted to the trigger area. This indicates that the shallow network still retains the original features of backdoor samples, which also makes our label recovery strategy possible. We further evaluate the performance of normal and malicious samples in different S-Nets. The prediction accuracy of each S-Nets for data belonging to different categories is shown in Figure 4. We present the results of typical universal backdoor attack BadNets and dynamic attack IAD on CIFAR-10. The target classes are 1 and 3, respectively. The line for the target label represents the result of backdoor input and the rest lines for standard data. The results show that the accuracy of normal samples shows a steady increase layer by layer and can achieve high performance in shallow layers. However, the classification accuracy of backdoor samples is notably low in the early layers but rapidly improves in the deep layers. This verifies the hypothesis that backdoor features mainly impact the classification process in the deep layers, leading to the inconsistency between the shallow and deep outputs of malicious samples. We provide more results of CIFAR-10 under other attacks in Appendix B.

4.3 Backdoor Sample Detection Results

Effectiveness

We verify the effectiveness of Orion on six different backdoor attacks. The results of IAD, TaCT, and Feature attacks on Tiny-Imagenet are omitted since the attacks can hardly succeed. We randomly select 500 benign samples and 500 malicious inputs for testing and take the average result of three replicate experiments. As shown in Table 1, none of the three baselines can identify feature-hidden backdoor attacks because they all rely on the separability of the feature repre-

Dataset	Attacks	Spectre			SCAn			Beatrix			Orion		
		PRE	REC	F1	PRE	REC	F1	PRE	REC	F1	PRE	REC	F1
CIFAR-10	Badnets	95.37	87.30	91.15	89.70	93.80	91.70	96.90	91.40	94.06	99.50	97.80	**98.60**
	Blended	93.30	88.24	90.69	91.47	92.80	92.13	95.00	94.90	94.94	98.90	97.60	**98.20**
	WaNet	100.0	74.07	85.10	92.36	88.79	90.53	95.20	89.20	92.18	99.79	97.80	**98.70**
	IAD	32.38	43.40	37.08	34.20	27.87	30.71	93.60	91.40	92.50	98.00	99.80	**98.90**
	TaCT	85.36	58.33	69.30	95.00	88.80	91.79	94.47	92.40	93.43	99.79	97.00	**98.37**
	Feature	88.69	0.16	0.31	88.33	0.40	0.79	28.20	6.23	10.21	93.70	98.60	**96.10**
GTSRB	Badnets	94.29	76.21	84.26	95.40	82.81	88.66	95.00	91.40	93.00	97.04	91.80	**94.34**
	Blended	94.60	76.41	84.54	93.90	78.60	85.57	95.90	80.60	87.58	94.38	87.40	**90.76**
	WaNet	96.20	79.66	87.15	92.30	83.40	87.62	94.11	90.80	92.42	95.50	99.80	**97.60**
	IAD	95.60	37.24	53.60	83.23	64.40	72.61	92.60	90.80	91.69	95.00	91.60	**93.37**
	TaCT	83.19	46.80	59.90	89.40	80.60	84.77	96.19	70.80	81.56	96.09	83.60	**89.41**
	Feature	30.47	3.43	6.16	99.65	2.89	5.61	54.60	29.42	38.23	95.33	85.80	**90.32**
Tiny-Imagenet	Badnets	99.91	82.20	90.19	100.0	81.00	89.00	94.89	91.49	93.15	95.55	93.40	**94.44**
	Blended	91.45	68.33	78.21	89.18	77.40	82.87	88.36	58.20	70.17	91.02	77.00	**83.42**
	WaNet	77.30	67.21	71.90	76.24	70.16	73.07	82.50	75.30	78.73	88.49	75.40	**81.42**

Table 1: Comparison of Orion with baselines. PRE indicates precision (%), REC represents recall (%), and F1 is the weighted harmonic mean of precision and recall to measure the effectiveness of detection.

sentation. Spectre and SCAn fail to defend against dynamic attacks IAD because their statistical assumptions about feature maps are based on universal triggers. Orion is effective for all the attacks and can achieve an F1-score of 90% under feature-hidden attacks. Orion also outperforms the three baselines on all other attack schemes, since the 1% clean data we use is a rather challenging setting for existing OOD detection similar methods, which require a large number of clean samples to estimate the statistical distribution of the clean data. However, relying on the strong learning ability of the deep neural network, shallow S-Nets can achieve good accuracy even with limited data. So the detection performance of Orion under a small amount of clean data is still ideal.

Ablation Studies

Sensitivity to Different Metric Designs. We verify the effectiveness of three metrics in the outlier score definition. Figure 5 shows the distribution of the benign and backdoor samples under different metrics with BadNets attack on CIFAR-10. The results indicate that the two types of input are divisible on all three metrics, with *consistency* being the strongest. *Stability* and *determinacy* characterize the frequency and degree of changes in the evolution of predictions, which can further enhance the separability of the samples. The outlier score that considers all three works best.

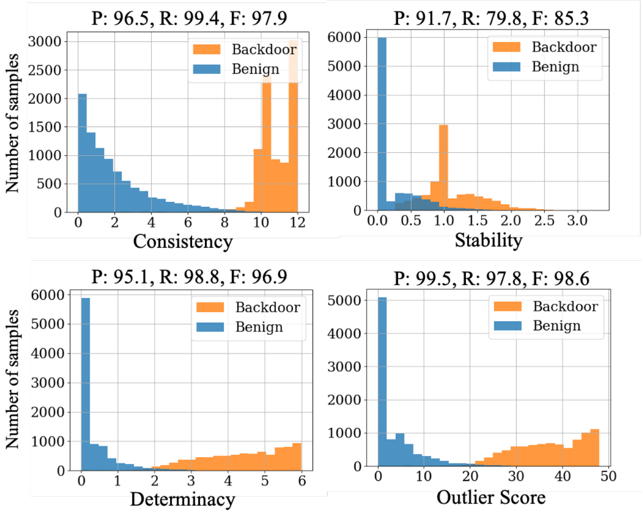


Figure 5: Sample distribution under different metrics. The orange bars indicate backdoor samples, and the blue bars represent normal input. The numbers above each subfigure are corresponding quantitative results which are precision (P), recall (R) and F1-score (F).

Sensitivity to Number of S-Nets. The number of S-Nets affects the effectiveness of Orion. If they are all distributed at deep or shallow layers, too few S-Nets may not recognize the deviation during the learning process. Large side net density can also weaken their ability to capture changes and introduce more model training overhead and detection time. We measure the effect of different numbers of S-Nets on the detection performance under BadNets attack on CIFAR-10, and the results are shown in Figure 6. It is shown that both too many and too few S-Nets can result in a decrease in performance and the number of S-Nets ranging from 4 to 6 is found to achieve the optimal results.

Sensitivity to Reference Set Size. We also assess how different sizes of reference data affect the performance of Orion. The results of CIFAR-10 under BadNets attack are presented in Table 2. It is shown that the detection performance becomes better with the increase in clean data size. More clean data can improve the accuracy of S-Nets, which allows them to identify the normal features in backdoor samples and makes the shallow predictions of normal inputs more accurate. Even with 0.5% clean data, Orion can still achieve 95% precision, which is a very tough situation in state-of-the-art detection schemes. More results of other datasets and attacks can be found in Appendix C.

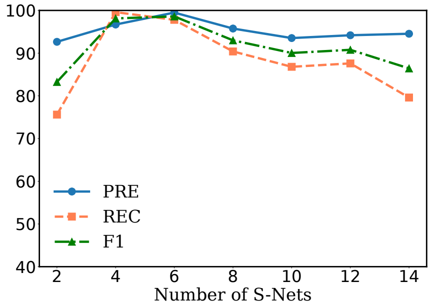


Figure 6: Impact of different numbers of S-Nets.

Reference set	PRE	REC	F1
0.1%	85.64	33.40	48.06
0.5%	95.31	93.40	94.34
1%	99.50	97.80	98.60
5%	98.60	98.90	98.70
10%	99.20	99.39	99.29

Table 2: Impact of different numbers of clean data for detection.

Network	Flops (G)	Params (M)	Training (s/epoch)	Detection (ms)
VGG16-bn	0.6 (0.6)	15.3 (15.5)	3.5	1.7
ResNet-56	0.3 (0.3)	0.9 (1.5)	4.2	0.8

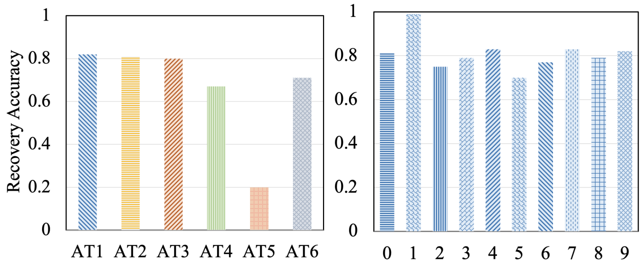
Table 3: Overhead of Orion. Flops denotes the number of floating-point operations per second for a single input. Params means the total number of model parameters. Inside the parentheses are the values after adding S-Nets. Training denotes the training time of S-Nets. Detection reports the additional time cost introduced by detection for each sample.

4.4 Overhead Analysis

Due to the importance of overhead in online settings, we report Orion's computational and time cost in Table 3. We demonstrate the results of VGG16 with batch normalization on CIFAR-10 and ResNet-56 on Tiny-Imagenet. All the experiments are carried out on a single NVIDIA GeForce RTX 3090 GPU. Flops and Params represent the amount of computation and the number of parameters introduced by attaching S-Nets. It is shown that the additional computation is negligible, and the parameter increase coming from the fully connected layers in S-Nets is also reasonable. The training of S-Nets is faster than that of the main branch network and can be done offline. The detection time for each sample is also negligible. Since the overhead is proportional to the number of S-Nets, users can vary their density to balance accuracy and efficiency according to their specific requirements.

4.5 Original Label Recovery

Figure 7(a) shows the original label recovery accuracy of CIFAR-10 under various attacks. We achieve an average 80% recovery accuracy on static backdoor attacks with universal triggers, such as BadNets, Blended, and WaNet. We can also successfully restore the original labels of the majority of samples on dynamic and feature-hidden attacks. We only fail on the TaCT attack at an accuracy of 20%. The possible reason is that TaCT complicates the boundaries between categories, which enlarges the diversity of shallow predictions of clean data. We also evaluate the recovery accuracy of samples of different source labels. The results in Figure 7(b) show that the recovery rates of different original classes are similar, where 1 is the target label of the attack.



(a) Attacks

(b) Labels

Figure 7: Clean label recovery accuracy of CIFAR-10. (a) Results under different attacks (AT1-6: BadNets, Blended, WaNet, IAD, TaCT, Feature attack). (b) Results of samples belonging to different original classes under BadNets attack.

5 Adaptive Attack

Beyond the existing backdoor variants, we further consider Orion's robustness against adaptive attacks where the attacker is aware of the defense and tries to bypass it. Orion is based on the high deviations between the shallow and deep layers output for backdoor samples, so the attacker can constrain such deviations during the backdoor implantation. We construct the attack by alternately training the main branch with poisoned data and the S-Nets with clean data. Specifically, the main network is updated with the loss of poisoning data on both the main and the side net, and S-Nets are learned on clean data separately. After 100 rounds of training, the attack success rate of the backdoored model is 99.24% and the clean accuracy is 88.22%. We use the S-Nets trained by the attacker and by the defender with another clean data set to evaluate the detection performance of Orion, and the F1-scores are 6.7% and 97.2% respectively. This indicates that the malicious model can only achieve small deviations on its own S-Nets, but cannot transfer to the defender's ones. In this case, Orion is able to defend against adaptive attacks as long as the defender has a private clean reference set.

6 Conclusion and Future Work

In this work, we develop a novel online backdoor detection framework Orion based on dynamic evolution analysis of sample predictions. We introduce side nets to output internal classification and monitor the evolution of prediction using a multi-exit branchy network. Extensive experiments on three datasets under six attacks verify the effectiveness and generality of our scheme. Orion outperforms existing detection techniques in all regimes and can defend against advanced feature-hidden attacks where state-of-the-art defenses fail. Meanwhile, by using shallow side outputs, Orion can recover the original labels of the tampered images at an accuracy of 80% on basic backdoor attacks.

We also propose two potential directions for future research. Firstly, as with most backdoor detection methods, Orion relies on a clean reference dataset for S-Nets training. Therefore, exploring data-free alternatives to apply Orion is a meaningful avenue for future work. Secondly, our approach can also be extended to other models such as vision transformers, as long as the model can output at shallow layers.

A Detailed Attack Setup

We verify the effectiveness of Orion under six backdoor attacks: BadNets, Blended, WaNet, IAD, TaCT, and Feature attacks. We consider the single-target attack, where all samples with triggers will be misclassified as one target class. The detailed attack settings are shown in Table 4. We ensure that each attack model performs well while maintaining high prediction accuracy on benign data. All attacks are implemented through data poisoning, i.e., planting backdoors by attaching triggers to part of the training dataset and modifying their labels to the target class. The trigger design algorithm and implantation method vary for different attacks. Figure 8 shows the attack samples of different backdoor attacks, including local modification and global noise perturbation. In our attempts to execute an attack, we strive to minimize the trigger size and poisoning rate to maximize the efficacy of the backdoor. This strategy enhances the covert nature of the backdoor and strengthens the validation of Orion in a more rigorous scenario. The attacks we experiment with encompass a diverse range of categories.

Basic Attacks. BadNets [Gu *et al.*, 2019] is a basic backdoor attack that uses a universal trigger with a fixed pattern determined in advance. We use a square all-white pixel block as the trigger and place it in the bottom right corner of the input to construct poisoning data.

Visually Hidden Attacks. Blended and WaNet [Chen *et al.*, 2017; Nguyen and Tran, 2021] are two typical visually hidden backdoor attacks. This type of attack aims to generate poisoning samples that are imperceptible to human eyes. Blended can adjust the transparency of triggers on top of BadNets, and we use an opacity of 0.2 for all the attacks. Unlike previous attacks that use patch-like triggers, WaNet generates attack samples by scrambling the entire image. It uses a small and smooth warping field to construct backdoor images. Specifically, we use a grid size of 4x4, and the strength of the warping field is 0.5.

Dynamic Attacks. Dynamic backdoor attacks consider a sample-specific trigger. Different inputs require different triggers, and using triggers of other samples will not produce malicious functions. IAD [Nguyen and Tran, 2020] uses an input-conditioned generator to construct attack samples. In order to make the trigger of one sample invalid for the others, IAD needs to construct a cross-dataset, which consists of mismatched samples and triggers while keeping their original class labels. We set the cross rate as 0.1, and the cross-test accuracy of CIFAR-10 and GTSRB are 88.57% and 94.53%.

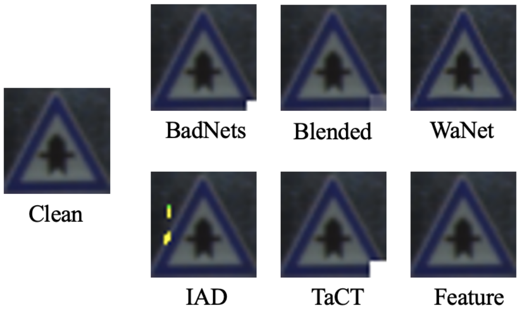


Figure 8: Poisoned samples of various attacks.

Robust Attacks. Robust attacks can evade some backdoor defense methods. [Tang *et al.*, 2021] proposed a label-specific attack TaCT, where only inputs from a specific source class are misclassified as the target label when stamped with the trigger. They proved that such a complicated misclassification strategy can make the model representation of malicious samples and benign inputs less distinguishable, thus bypassing existing defenses [Wang *et al.*, 2019; Gao *et al.*, 2019; Chou *et al.*, 2020; Chen *et al.*, 2019a]. IAD requires a cover-set, where images from cover classes are classified correctly even if they appear together with the trigger. In our experiments, we set 0 as the source class, 1 as the target label, 5 and 7 as the cover classes, and the cover rate is the same as poisoning rate.

Feature-Hidden Attacks. Recently, [Zhong *et al.*, 2022] proposed a novel feature-hidden attack that is imperceptible in both input space and model representation. Their triggers are obtained by sampling from a polynomial distribution, and a U-net-based network generates the parameters of the polynomial distribution. The attack samples of feature-hidden backdoor attacks are confused with benign inputs in the feature representation, thus being able to resist backdoor defenses based on feature separability. We use the same attack settings as in the original paper.

B More Results of Intuition Experiments

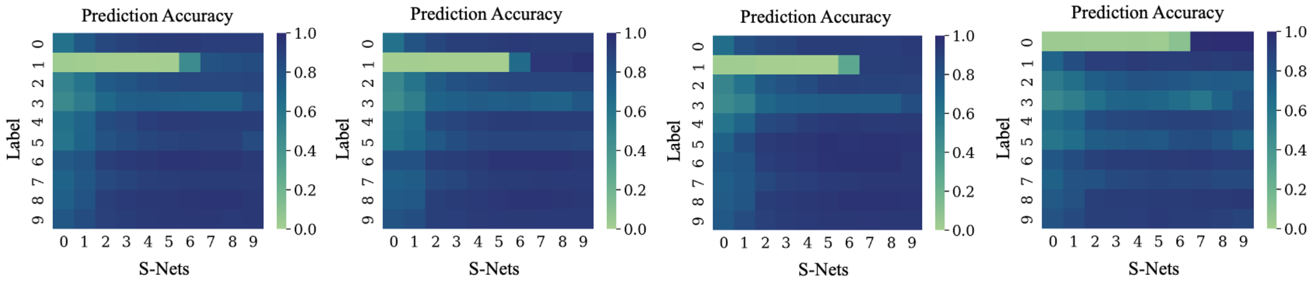
We present more results of our design rationale verification experiments. Figure 9 shows the prediction accuracy of different S-Nets on CIFAR-10 under other attacks. The experimental results are similar to those of BadNets and IAD. Under different attack strategies, the backdoors all produce malicious effects only at the deeper layers, and the shallow S-Nets fail to achieve misclassification. In contrast, the features of clean samples are learned at the shallow layers and are enhanced with the depth of the network.

C More Results of Ablation Studies

We show more results of our ablation study on the impact of the reference set size. Table 5 shows the performance of Orion with different amounts of clean data. It can be found that more clean data leads to more accurate backdoor detection. Even with only 0.1% clean data, Orion can identify the majority of malicious samples in most cases.

Dataset	Attacks	Trigger size	Poisoning rate	Target label	ASR	PA
CIFAR-10	BadNets	5 x 5	0.05	1	97.94%	91.34%
	Blended	5 x 5	0.05	1	85.88%	90.89%
	WaNet	global	0.10	1	97.45%	90.78%
	IAD	global	0.10	3	99.39%	93.58%
	TaCT	5 x 5	0.05	1	94.40%	92.22%
	Feature	global	0.10	0	99.69%	86.26%
GTSRB	BadNets	5 x 5	0.05	1	96.65%	97.97%
	Blended	5 x 5	0.10	1	94.11%	96.57%
	WaNet	global	0.10	0	85.66%	94.95%
	IAD	global	0.10	1	99.71%	98.51%
	TaCT	5 x 5	0.10	1	100.0%	98.38%
	Feature	global	0.10	0	99.80%	95.68%
Tiny-Imagenet	BadNets	6 x 6	0.05	1	96.65%	47.44%
	Blended	6 x 6	0.10	1	85.57%	46.77%
	WaNet	global	0.10	1	89.55%	44.97%

Table 4: Detailed attack settings. ASR means the attack success rate, which is the proportion of samples with triggers that are classified as target labels. PA represents the prediction accuracy of clean data.



(a) Blended (b) WaNet (c) TaCT (d) Feature

Figure 9: Prediction accuracy over layers.

Dataset	Attacks	0.1%			0.5%			1%			5%			10%		
		PRE	REC	F1	PRE	REC	F1	PRE	REC	F1	PRE	REC	F1	PRE	REC	F1
CIFAR-10	BadNets	85.64	33.40	48.06	95.31	93.40	94.34	99.50	97.80	98.60	98.60	98.90	98.70	99.20	99.39	99.29
	Blended	95.48	97.20	96.33	95.14	98.00	96.55	98.90	97.60	98.20	98.79	98.20	98.49	98.60	99.80	99.20
	WaNet	95.16	98.40	96.75	95.00	98.80	96.86	99.79	97.80	98.70	98.79	98.80	98.89	98.80	99.20	99.00
	IAD	94.33	93.20	93.76	97.20	97.40	97.30	98.00	99.80	98.90	98.80	99.00	98.90	99.60	99.60	99.60
	TaCT	93.96	96.60	95.26	97.20	94.92	96.04	99.79	97.00	98.37	98.60	96.30	98.60	99.00	99.00	99.00
	Feature	94.12	99.40	96.69	94.44	98.60	96.47	93.70	98.60	96.10	96.59	96.40	96.49	95.69	97.80	96.73
GTSRB	BadNets	73.38	97.60	83.77	95.62	91.80	93.67	97.04	91.80	94.34	95.60	100.0	97.75	96.33	99.80	98.04
	Blended	75.29	88.40	81.32	82.58	92.00	87.03	94.38	87.40	90.76	94.50	99.80	97.08	96.14	99.80	97.93
	WaNet	74.00	96.20	83.65	84.33	98.00	90.65	95.50	99.80	97.60	95.76	99.40	97.54	96.70	99.80	98.22
	IAD	75.41	98.80	85.54	83.83	99.60	91.04	95.00	91.60	93.37	95.41	100.0	97.65	95.41	99.80	97.55
	TaCT	71.68	87.60	78.84	82.71	89.00	85.74	96.09	83.60	89.41	94.84	99.40	97.07	94.16	100.0	96.99
	Feature	73.22	96.80	83.37	80.45	98.80	88.68	95.33	85.80	90.32	95.37	99.00	97.15	97.65	100.0	98.81
Tiny-Imagenet	BadNets	66.33	92.20	77.15	82.55	95.60	88.60	95.55	93.40	94.44	94.48	99.40	96.88	95.33	98.20	96.74
	Blended	63.18	85.80	72.77	78.49	87.60	82.79	91.02	77.00	83.42	95.69	89.00	92.22	96.25	92.40	94.28
	WaNet	66.34	82.00	73.34	76.44	84.40	80.22	88.49	75.40	81.42	95.49	84.80	89.83	95.99	91.00	93.42

Table 5: Impact of different clean data size.

References

[Cai *et al.*, 2022] Ruisi Cai, Zhenyu Zhang, Tianlong Chen, Xiaohan Chen, and Zhangyang Wang. Randomized channel shuffling: Minimal-overhead backdoor attack detection without clean datasets. In *Advances in Neural Information Processing Systems*, 2022.

[Chen *et al.*, 2017] 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.

[Chen *et al.*, 2019a] 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. In *Workshop on Artificial Intelligence Safety*, 2019.

[Chen *et al.*, 2019b] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety co-located with AAAI*, volume 2301, 2019.

[Chen *et al.*, 2020] Xinshi Chen, Hanjun Dai, Yu Li, Xin Gao, and Le Song. Learning to stop while learning to predict. In *International Conference on Machine Learning*, pages 1520–1530. PMLR, 2020.

[Chou *et al.*, 2020] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *IEEE Security and Privacy Workshops (SPW)*, pages 48–54. IEEE, 2020.

[Gao *et al.*, 2019] 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 *Annual Computer Security Applications Conference*, pages 113–125, 2019.

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

[Hayase *et al.*, 2021] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: defending against backdoor attacks using robust statistics. In *International Conference on Machine Learning*, pages 4129–4139, 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[Hu *et al.*, 2020] Ting-Kuei Hu, Tianlong Chen, Haotao Wang, and Zhangyang Wang. Triple wins: Boosting accuracy, robustness and efficiency together by enabling input-adaptive inference. In *International Conference on Learning Representations*, 2020.

[Kaya *et al.*, 2019] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *International Conference on Machine Learning*, pages 3301–3310. PMLR, 2019.

[Kolouri *et al.*, 2020] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 301–310, 2020.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[Kumar *et al.*, 2020] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *IEEE Security and Privacy Workshops*, pages 69–75. IEEE, 2020.

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

[Lee *et al.*, 2016] Chen-Yu Lee, Patrick W Gallagher, and Zhuowen Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Artificial Intelligence and Statistics*, pages 464–472. PMLR, 2016.

[Liu *et al.*, 2019] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *SIGSAC Conference on Computer and Communications Security*, pages 1265–1282. ACM, 2019.

[Liu *et al.*, 2022] Yingqi Liu, Guangyu Shen, Guanhong Tao, Zhenting Wang, Shiqing Ma, and Xiangyu Zhang. Complex backdoor detection by symmetric feature differencing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15003–15013, 2022.

[Luo *et al.*, 2016] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[Ma *et al.*, 2023] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The ”beatrix” resurrections: Robust backdoor detection via gram matrices. In *Annual Network and Distributed System Security Symposium*, 2023.

[Nguyen and Tran, 2020] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *Advances in Neural Information Processing Systems*, volume 33, pages 3454–3464, 2020.

[Nguyen and Tran, 2021] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021.

[Parkhi *et al.*, 2015] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. *British Machine Vision Conference 2015*, 2015.

[Rajkomar *et al.*, 2018] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ digital medicine*, 1(1):1–10, 2018.

[Redmon *et al.*, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on*

Computer Vision and Pattern Recognition, pages 779–788, 2016.

[Selvaraju *et al.*, 2017] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE/CVF International Conference on Computer Vision*, pages 618–626, 2017.

[Stallkamp *et al.*, 2012] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012.

[Sun *et al.*, 2022] Yi Sun, Jian Li, and Xin Xu. Meta-gf: Training dynamic-depth neural networks harmoniously. In *European Conference on Computer Vision*, pages 691–708. Springer, 2022.

[Tang *et al.*, 2021] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection. In *USENIX Security Symposium*, pages 1541–1558. USENIX Association, 2021.

[Tran *et al.*, 2018] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[Wang *et al.*, 2019] 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 *IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

[Wang *et al.*, 2022] Zhenting Wang, Kai Mei, Hailun Ding, Juan Zhai, and Shiqing Ma. Rethinking the reverse-engineering of trojan triggers. In *Advances in Neural Information Processing Systems*, 2022.

[Xu *et al.*, 2021] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *IEEE Symposium on Security and Privacy*, pages 103–120, 2021.

[Zhong *et al.*, 2022] Nan Zhong, Zhenxing Qian, and Xinpeng Zhang. Imperceptible backdoor attack: From input space to feature representation. In *International Joint Conference on Artificial Intelligence*, pages 1736–1742, 2022.