Progressive Backdoor Erasing via connecting Backdoor and Adversarial Attacks

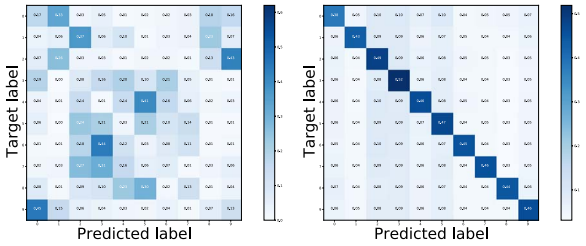
Bingxu Mu[1] Zhenxing Niu[2*] Le Wang[3] Xue Wang[4] Qiguang Miao[2] Rong Jin[4] Gang Hua[5]
[1]School of Software Engineering, Xi'an Jiaotong University [2]Xidian University
[3]IAIR, Xi'an Jiaotong University [4]Alibaba Group [5]Wormpex AI Research

Abstract

Deep neural networks (DNNs) are known to be vulnerable to both backdoor attacks as well as adversarial attacks. In the literature, these two types of attacks are commonly treated as distinct problems and solved separately, since they belong to training-time and inference-time attacks respectively. However, in this paper we find an intriguing connection between them: for a model planted with backdoors, we observe that its adversarial examples have similar behaviors *as its triggered images,* i.e.*, both activate the* same *subset of DNN neurons. It indicates that planting a backdoor into a model will significantly affect the model's adversarial examples. Based on these observations, a novel Progressive Backdoor Erasing (PBE) algorithm is proposed to progressively purify the infected model by leveraging untargeted adversarial attacks. Different from previous backdoor defense methods, one significant advantage of our approach is that it can erase backdoor even when the clean extra dataset is unavailable. We empirically show that, against 5 state-of-the-art backdoor attacks, our PBE can effectively erase the backdoor without obvious performance degradation on clean samples and outperforms existing defense methods.*

1. Introduction

Deep neural networks (DNNs) have been widely adopted in many safety-critical applications (*e.g.*, face recognition and autonomous driving), thus more attention has been paid to the security of deep learning. It has been demonstrated that DNNs are prone to potential threats in both their inference as well as training phases. Inference-time attack (a.k.a. *adversarial attack* [5, 25]) aims to fool a trained model into making incorrect predictions with small adversarial perturbations. In contrast, training-time attack (a.k.a. *backdoor attack* [13]) attempts to plant a backdoor into a model in the training phase, so that the infected model would misclassify the testing images as the *target-label* whenever a pre-defined *trigger* (*e.g.*, several pixels) is embedded into them (*i.e.*, triggered testing images).



(a) For *benign* model. (b) For *infected* model.

Figure 1. Predicted labels v.s. Target-labels for $10,000$ randomly sampled **adversarial examples** from CIFAR-10, with respect to *benign* and *infected* models. (a) For a benign model, the predicted labels obey *uniform* distribution; (b) for infected models under WaNet backdoor attack [20], its adversarial examples are *highly likely* to be classified as the target-label (the matrix diagonals).

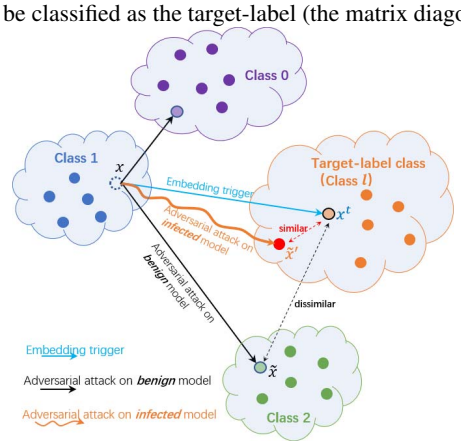


Figure 2. Illustration of our observations. For benign models, conducting an untargeted adversarial attack will make an image move close to *any* class (*e.g.*, Class 0 or Class 2) in feature space. But for infected models, adversarial attack will make it move close to the target-label class (*e.g.*, Class l)

Due to the obvious differences between backdoor and adversarial attacks, they are often treated as two different problems and solved separately in the literature. But in this paper, we illustrate that there is an underlying connection between them, *i.e.*, planting a backdoor into one model will significantly affect the model's adversarial examples. Moreover, based on such findings we propose a new method to defend against backdoor attacks by leveraging adversarial attack techniques (*i.e.*, generating adversarial examples).

2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.01963

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.01963

In particular, we observe that: for a model planted with backdoors, its adversarial examples have *similar behaviors* as its triggered images. This is significantly different from a benign model without backdoors. Specifically, for a **benign model**, the predicted class labels of its adversarial examples obey a *uniform* distribution, as shown in Fig.1a. However, for an **infected model**, we surprisingly observe that **its adversarial examples are *highly likely* to be predicted as the backdoor target-label**, as shown in Fig.1b. As we know, triggered images will also be predicted as the backdoor target-label by an infected model. Therefore, it means that adversarial examples have similar behaviors as its triggered images for an infected model. Particularly, these phenomena are present regardless of the target-label, the backdoor attack setting (*i.e.*, all-to-one or all-to-all settings), and even for most trigger embedding mechanisms (*e.g.*, adding [6], blending [3] or warping [20]).

To find the underlying reason of such phenomena, we measure the feature similarity of those adversarial images and triggered images. Briefly, we find that after planting a backdoor into one model, the features of adversarial images change significantly. Particularly, the features of adversarial image $\widetilde{\boldsymbol{x}}'$ are surprisingly very similar to that of triggered image $\boldsymbol{x}^{\boldsymbol{t}}$, as illustrated in Fig.2 and Fig.3. It indicates that **both the $\widetilde{\boldsymbol{x}}'$ and $\boldsymbol{x}^{\boldsymbol{t}}$ have similar behaviors, *i.e.,* both activate the *same* subset of DNN neurons**. Note that such connection between adversarial and backdoor attack could be leveraged to design backdoor defense methods.

Backdoor attacks made great advances in recent years, evolved from visible trigger [6] to invisible trigger [3, 16, 20], from poisoning label to clean-label attacks [1]. For example, WaNet [20] uses affine transformation as trigger embedding mechanism, which could significant improve the invisibility of trigger. In contrast, the research on backdoor defenses lag behind a little. Even for the state-of-the-art backdoor defense methods [12, 14, 17], most of them can be evaded by the advanced modern backdoor attacks. Moreover, a clean extra dataset is often required by those defense methods to erase backdoor from infected models.

In this paper, we propose a new backdoor defense method based on the discovered connections between adversarial and backdoor attacks, which could not only defend against modern backdoor attacks but also work without a clean extra dataset. Specifically, at the beginning the training data (containing poisoning images) are randomly sampled to build an initial extra dataset. Next, we use them to purify the infected model by leveraging adversarial attack techniques. And then, the purified model is used to identify clean images from training data, which are used to update the extra dataset. With an alternating procedure, the infected model as well as the extra dataset are progressively purified. So, we call our approach *Progressive Backdoor Erasing* (PBE).

Regarding how to purify the infected model, we generate adversarial examples and use them to fine-tune the infected model. Since adversarial images could come from arbitrary class, such fine-tuning procedure works like associating triggered images to arbitrary class instead of just the target class, which breaks the foundation of backdoor attacks (*i.e.*, building a strong correlation between a trigger pattern and a target-label [12]). That is why our approach can erase backdoor from infected models.

As for identifying clean images, since clean images have similar prediction results for both benign and infected models, we could effectively identify them by using the previously obtained purified model. Note that if a clean extra dataset is available, we can skip the step of purifying extra dataset, and only run the step of purifying model once.

A big advantage of our approach is that it does not need the clean extra dataset and it can progressively filter poisoning training data to obtain clean data. In our approach, the purified model could help to obtain clean data, in return the obtained clean data could help to further purify model. Thus, the alternating iterations could progressively improve each other. To the best of knowledge, our approach is the first work to defend against backdoor attack without a clean extra dataset.

Our main contributions are summarized as follows:

- We observe an underlying connection between backdoor attacks and adversarial attacks, *i.e.,* for an infected model, its adversarial examples have similar behaviors as its triggered samples. And an theoretical analysis is given to justify our observation.

- According to our observations, we propose a progressive backdoor defense method, which achieves the state-of-the-art defensive performance, even when a clean extra dataset is unavailable.

2. Related Work

Backdoor Attack has evolved from visible trigger [6] to invisible trigger [3, 16, 20] in these years. These trigger patterns can appear in forms as simple as a patch [6], a sinusoidal strips [1], and a blending pattern [3]. Besides, TrojanNN [28] proposes to learn a trigger from benign model. In [22], trigger pattern and backdoor model are jointly optimized. In order to make triggers more stealthy, advanced modern backdoor attacks propose some complex trigger-embedding mechanisms, such as input-aware dynamic patterns [19], natural reflection [16] and image warping [20]. Meanwhile, backdoor attack has evolved from poisoning label to clean-label attacks [1], where the ground-truth label of poisoned samples could also be consistent with the target label. This will further increase the stealthiness of backdoor attacks. A survey of backdoor attacks can be found in [13].

Backdoor Defense can be roughly categorized into backdoor detection and backdoor erasing. Detection-based methods aim at identifying the existence of backdoor in the underlying model [10, 27] or filtering the suspicious samples in training data for re-training [2, 21, 26]. Although they perform fairly well in distinguishing whether a model has been poisoned, the backdoor still remains in the infected model.

The erasing-based methods aim to directly purify the infected model by removing the malicious impacts caused by the backdoor triggers, while maintaining the model performance on clean data. One approach is to directly fine-tune the infected model with the clean extra dataset [17]. Fine-Pruning [14] proposes using neural pruning to remove backdoor neurons. In [12], Neural Attention Distillation (NAD) is proposed to erase backdoor by leverage knowledge distillation. Later, Adversarial Neuron Pruning (ANP) [4] is proposed to prune backdoor neuron by perturbing model weights. Besides, some trigger synthesis based methods are proposed [27]. Neural Cleanse (NC) [27] and Artificial Brain Stimulation (ABS) [15] are proposed to first recover the backdoor trigger, and use the recovered trigger to erase the backdoor. However, these methods are only able to handle fixed triggers since they need to explicitly recover triggers. In contrast, our approach does not need to recover trigger pattern so that it can deal with content-aware/non-trigger-fixed attacks such as DynamicAtt [19], WaNet [20]. In addition, all previous defense methods need a clean extra dataset.

Adversarial Attack and Defense. The adversarial attack [5, 9, 25] is a kind of inference-time attacks. It aims to fool a trained model into making incorrect predictions (*i.e.,* untargeted adversarial attack) or predicting the input as a particular label (*i.e.,* targeted adversarial attack). On the other hand, many defense methods are also proposed against adversarial attacks. *Adversarial training* [18] is one of the most effective methods. Recently, [23] proposes to use a 'trapdoor' to detect adversarial examples. It illustrates that a particular trapdoor could lead to producing adversarial examples similar to trapdoors in the feature space. However, it is quite different from our work since it aims to detect adversarial examples while our approach aims to defend against backdoor attacks.

3. Our approach

3.1. Backdoor Attack

We focus on backdoor attacks on image classification. Let $D_{\text{train}} = \{(\boldsymbol{x}_i; y_i)\}_{i=1}^{N}$ be the clean training data and $f(\boldsymbol{x}; \theta)$ be the benign CNN model decision function with parameter θ.

For backdoor attack, we define or learn a trigger embedding function $\boldsymbol{x}^t = \text{Trigger}(\boldsymbol{x})$ which can convert a clean sample $\boldsymbol{x}_i$ to a triggered/poisoned sample $\boldsymbol{x}_i^t$. Given a target-label l, we can poison a small part of training samples, *i.e.,* replace $(\boldsymbol{x}_i, y_i)$ with $(\boldsymbol{x}_i^t, l)$, which produces poisoned training data D'_{train}. The training with D'_{train} results in the infected model $f(\boldsymbol{x}; \theta')$. Note that different attacks will define different trigger embedding functions $\text{Trigger}(\cdot)$.

At testing time, if a clean input $(\boldsymbol{x}, y) \in D_{\text{test}}$ is fed to the infected model, it is supposed to be correctly predicted as y. In contrast, for a triggered sample $\boldsymbol{x}^t$, its prediction changes to the target-label l. Particularly, backdoor attacks can be divided into two categories according to the selection of target-labels: (1)All-to-one attack: the target-labels for all examples are set as l; (2)All-to-all attack: the target-labels for different classes could be set differently, such as $y + 1$, *i.e.*,

$$\text{All-to-one attack: } \begin{cases} f(\boldsymbol{x}; \theta') = y; \\ f(\boldsymbol{x}^t; \theta') = l, \boldsymbol{x}^t = \text{Trigger}(\boldsymbol{x}) \end{cases} \tag{1}$$

$$\text{All-to-all attack: } \begin{cases} f(\boldsymbol{x}; \theta') = y; \\ f(\boldsymbol{x}^t; \theta') = y + 1, \boldsymbol{x}^t = \text{Trigger}(\boldsymbol{x}) \end{cases} \tag{2}$$

3.2. Backdoor Defense

We adopt a typical defense setting where the defender has an infected model $f(\boldsymbol{x}; \theta')$ as well as a clean extra dataset D_{ext}. The goal of the backdoor defense is to *erase* the backdoor trigger from the model while retaining the performance of the model on clean samples. In other words, we want to obtain a cleaned/purified model $f(\boldsymbol{x}; \theta^c)$ such that:

$$\begin{cases} f(\boldsymbol{x}; \theta^c) = y; \\ f(\boldsymbol{x}^t; \theta^c) = y, \boldsymbol{x}^t = \text{Trigger}(\boldsymbol{x}) \end{cases} \tag{3}$$

3.3. Untargeted Adversarial Attack

Untargeted adversarial attack aims to find the best perturbation $\boldsymbol{r}$ so that the adversarial examples $\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{r}$ will be misclassified, *i.e.,* the loss $L(\widetilde{\boldsymbol{x}}, y)$ is maximized with respect to $\boldsymbol{r}$, as follows:

$$\max_{\boldsymbol{r}} L(\widetilde{\boldsymbol{x}}, y; \theta) \tag{4}$$

$$\text{s.t. } ||\boldsymbol{r}||_p < \epsilon, \widetilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{r}$$

$$\widetilde{\boldsymbol{x}} \in [0, 1]^d \tag{5}$$

Note that untargeted adversarial attack means that perturbed inputs $\widetilde{\boldsymbol{x}}$ are only desired to be misclassified (*i.e.*, different from their original labels y as Eq.(4)), rather than being classified as a particular label (which is the goal of the *targeted adversarial attack*). Therefore, it has been observed that the predicted labels of $\widetilde{\boldsymbol{x}}$ obey a uniform distribution across all classes.

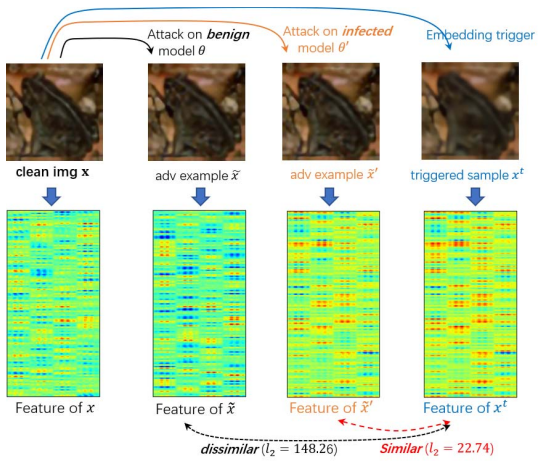


Figure 3. The similarity of features for clean image x, benign model's adversarial example $\widetilde{\boldsymbol{x}}$, infected model's adversarial example $\widetilde{\boldsymbol{x}}'$, and triggered image x^t. Obviously, the features of $\widetilde{\boldsymbol{x}}'$ are very similar to x^t. In contrast, there is a significant difference between $\widetilde{\boldsymbol{x}}'$ and $\widetilde{\boldsymbol{x}}$, which indicates adversarial examples will change significantly *after* planting a backdoor into a model.

3.4. Empirical Observations and Analysis

3.4.1 Empirical Observations

In this section, we will describe how we obtain the observation that **for an infected model, its adversarial examples have similar behaviors as its triggered samples**. Specifically, we first conduct an untargeted adversarial attack on the *infected* model $f(\boldsymbol{x};\theta')$ to generate adversarial examples $\widetilde{\boldsymbol{x}}'$ as follows:

$$\max_{\boldsymbol{r}} L(\widetilde{\boldsymbol{x}}', y; \theta') \quad \text{s.t. } ||\boldsymbol{r}||_p < \epsilon, \widetilde{\boldsymbol{x}}' = \boldsymbol{x} + \boldsymbol{r} \tag{6}$$

Meanwhile, we also conduct an untargeted adversarial attack on the *benign* model $f(\boldsymbol{x};\theta)$ to produce the adversarial examples $\widetilde{\boldsymbol{x}}$ as Sec.3.3.

We next examine the classification results of those adversarial examples. As shown in Fig.1a, when feeding adversarial examples $\widetilde{\boldsymbol{x}}$ to the benign model, $\widetilde{\boldsymbol{x}}$ will be classified as any class with the same probability (except its ground-truth label), *i.e.*, obeying a uniform distribution. In contrast, when feeding adversarial examples $\widetilde{\boldsymbol{x}}'$ to the infected model, we observe that $\widetilde{\boldsymbol{x}}'$ **are *highly likely* to be classified as the target-label**. As shown in Fig.1b, if an untargeted adversarial attack is conducted on an infected model with target-label $l \in \{0, \ldots, 9\}$, we observe that at least more than 40% of $\widetilde{\boldsymbol{x}}'$ are predicted as the target-label l. These phenomena are present regardless of what dataset is, as shown in Fig.4.

It indicates that there is an underlying connection between adversarial examples $\widetilde{\boldsymbol{x}}'$ and triggered samples $\boldsymbol{x}^t$, since $\boldsymbol{x}^t$ **are also expected to be classified as the target-label**. For further investigation, we check the feature maps of clean samples $\boldsymbol{x}$, benign model's adversarial examples $\widetilde{\boldsymbol{x}}$, infected model's adversarial examples $\widetilde{\boldsymbol{x}}'$, and triggered images $\boldsymbol{x}^t$. We find that the features of $\widetilde{\boldsymbol{x}}'$ are very similar to the features of triggered samples $\boldsymbol{x}^t$, while there is a significant difference between the features of $\widetilde{\boldsymbol{x}}$ and $\boldsymbol{x}^t$. As shown in Fig.3, the l_2 distance between the features of $\widetilde{\boldsymbol{x}}'$ and $\boldsymbol{x}^t$ is smaller than that between $\widetilde{\boldsymbol{x}}$ and $\boldsymbol{x}^t$. More quantitative comparisons are provided in Table.4. Such feature similarity indicates that both adversarial examples $\widetilde{\boldsymbol{x}}'$ and triggered samples $\boldsymbol{x}^t$ could activate the *same* subset of DNN neurons, *i.e.,* **the adversarial examples $\widetilde{\boldsymbol{x}}'$ have similar behaviors as triggered samples $\boldsymbol{x}^t$**.

We speculate why adversarial examples would have significant changes after a backdoor is planted into a model as follows: some DNN neurons will be activated by a trigger when a backdoor is planted into a model, which are called 'backdoor neurons' [4]. When conducting an adversarial attack on infected models, those 'backdoor neurons' are more likely to be chosen/locked and activated as generating adversarial examples. Thus, the generated adversarial examples could work like triggered images.

3.4.2 Theoretical Analysis

In order to dive deeply, we theoretically justify our observations for the case of a linear mode. Generally, a linear classifier can be denoted by $W = (w_1, w_2, \ldots, w_K)$. Thus, let us denote the trained infected linear classifiers be $\widetilde{W}^* = (\widetilde{w}_1, \widetilde{w}_2, \ldots, \widetilde{w}_K)$.

To simplify our analysis, we assume the trigger embedding function $\mathrm{Trigger}(\cdot)$ is designed to add a pre-defined patch P to an input image, i.e.,

$$\boldsymbol{x}^{\boldsymbol{t}} = \mathrm{Trigger}(\boldsymbol{x}) = \boldsymbol{x} + P \tag{7}$$

We assume that the original training examples (without any trigger) can be perfectly classified with margin $\tau > 0$, and that τ is large enough such that a small perturbation made to $\widetilde{W}^*$ will not affect classification result. Thus, we have following Theorem,

Theorem 1 *Under the previous assumptions, we have $\boldsymbol{r}_\perp$, the projection of $\boldsymbol{r}$ on the direction of P, bounded as*

$$\frac{|\boldsymbol{r}_\perp|}{|\boldsymbol{r}|} \geq \frac{(\sqrt{2}-1)\ell|P|^2}{\sqrt{(\sqrt{2}-1)^2\ell^2|P|^4 + \left(\ell|P|^2 + \sqrt{2}K/(\exp(\tau)+K)\right)^2}}$$

From the above theorem, we can see that when projecting perturbation $\boldsymbol{r}$ on the direction of trigger P, the projection $\boldsymbol{r}_\perp$ take an significant part in the full perturbation $\boldsymbol{r}$. It means that the perturbation $\boldsymbol{r}$ is very similar to the trigger P, which justifies our observations that the adversarial examples $\widetilde{\boldsymbol{x}}' = \boldsymbol{x} + \boldsymbol{r}$ are similar to triggered images $\boldsymbol{x}^{\boldsymbol{t}} = \boldsymbol{x} + P$. The detailed description and proof are shown in the supplemental material.

3.5. Progressive Backdoor Erasing

Threat Model. We assume the adversary has access to the training data and has planted a backdoor into a model. And then, the infected model is given to the defender.

Defense Setting. In this paper, we discuss two defensive settings. The first one follows the setting of the *model repair* defense methods, where we just have an infected model and a clean extra dataset but cannot access the training data.

The second one follows the setting of the *data filtering* defense methods, where we can access the training data and do not have a clean extra dataset. Note that we do not know which training images are poisoned.

Based on the discovered connection between adversarial and backdoor attacks, we propose a **Progressive Backdoor Erasing (PBE)** method, as shown in Algorithm 1. Our approach could work for both two defensive settings. For the second setting, since clean extra dataset is unavailable, we will randomly sample some images from the training data as D_{ext}^0 at the initialization step. Next, we enter into an iterative procedure containing three steps: the first step will produce a purified model by leveraging adversarial examples, which could erase backdoor (*i.e.,* significantly reduce ASR); At the second step, the D_{ext}^t are used to improve the performance of purified model on benign testing images (*i.e.,* significantly improve ACC); At the third step, the purified model is used to identify clean images in training dataset, which results in a cleaner and better D_{ext}^{t+1}.

Although the initial extra dataset D_{ext}^0 contains poisoning images, in the following iterations we can identify clean images from training data and produce a cleaner D_{ext}^t progressively.

Regarding the first defensive setting, we drop the **Initialization** step and use the known *clean* extra dataset $D_{\text{ext}}^{\text{clean}}$. And then, we simply skip the step-3 and only need to run the iteration once, *i.e.*, just run step-1 and step-2 once, which is called **Adversarial Fine-Tuning (AFT)** in this paper.

3.5.1 Purifying an Infected Model

Specifically, given the infected model $f(\boldsymbol{x};\theta^t)$, for each $(\boldsymbol{x}_i, y_i) \in D_{\text{ext}}$, we obtain a corresponding adversarial example $\widetilde{\boldsymbol{x}}_i'$ according to Eq.(6), which produces $\widetilde{D}_{\text{ext}} = \{(\widetilde{\boldsymbol{x}}_i', y_i)\}_{i=1}^m$. And then, we fine-tune the infected model θ^t with $\widetilde{D}_{\text{ext}}$, which produces purified model θ^{t+1}, *i.e.,*

$$\theta^{t+1} = \arg\min_{\theta} \mathbb{E}_{(\widetilde{\boldsymbol{x}}_i', y_i)\in\widetilde{D}_{\text{ext}}}[L(\widetilde{\boldsymbol{x}}_i', y_i; \theta)] \quad (8)$$
$$\text{s.t. } \theta^0 = \theta'$$

Since adversarial examples could come from arbitrary class, they are associated with all possible class labels. According to the similarity between adversarial examples and triggered samples, when we fine-tune the infected model with adversarial examples, it mimics fine-tuning the model with triggered samples, yet the associated labels are not just the target-label but all possible class labels.

Note that the foundation of backdoor attacks is to build a strong correlation between a trigger pattern and a target-label, which is achieved by poisoning training data, *i.e.,* to associate triggered samples with target-labels. As a result, our fine-tuning approach will break such a strong correlation and hence can achieve a defensive effect.

Algorithm 1 Progressive Backdoor Erasing

Input: Infected model θ', training data D_{train}
Output: Purified model θ^T

1: **Initialization**: obtain extra dataset D_{ext}^0 by randomly sampling from D_{train}; let $\theta^0 = \theta'$
2: **For** $t = 0, 1, 2 \ldots, T$:
3: **Step-1**: purify the model θ^t with D_{ext}^t
4: **Step-1a**: untargeted adversarial attack. For each $(\boldsymbol{x}_i, y_i) \in D_{\text{ext}}^t$, generate adversarial example $\widetilde{\boldsymbol{x}}_i'$ according to Eq.(6), which results in $\widetilde{D}_{\text{ext}}^t = \{(\widetilde{\boldsymbol{x}}_i', y_i)\}_{i=1}^m$
5: **Step-1b**: 1-st time fine-tuning. Fine-tuning the model θ^t with $\widetilde{D}_{\text{ext}}^t$ according to Eq.(8)
6: **Step-2**: 2-nd time fine-tuning. Continue to fine-tune model θ^t with D_{ext}^t, and obtain purified model θ^{t+1}
7: **Step-3**: update the extra dataset D_{ext}^t. Identify clean images from D_{train} according to Eq.(9), resulting in an updated dataset D_{ext}^{t+1}
8: **return** Advanced purified model parameter θ^T

3.5.2 Identifying Clean Images

As we know, an infected model affects the prediction results of poisoning images, while a benign model does not. Therefore, after feeding an image to an infected as well as a purified model, if the two models yield distinct predictions (probability across all classes), it is likely to be a poisoning image. In this way, we could identify poisoning as well as clean images.

Specifically, for each image $\boldsymbol{x}_i \in D_{\text{train}}$, we feed it to the infected model $f(\boldsymbol{x};\theta')$ and previously purified model $f(\boldsymbol{x};\theta^t)$, respectively. The predicted logits of the two models are noted as $a(\boldsymbol{x};\theta')$ and $a(\boldsymbol{x};\theta^t)$ (*i.e.,* the network activation just before softmax layer). We use the cosine similarity between them to measure the changes of the prediction,

$$S_{\theta',\theta^t}(\boldsymbol{x}) = \frac{\langle a(\boldsymbol{x};\theta'), a(\boldsymbol{x};\theta^t)\rangle}{|a(\boldsymbol{x};\theta')||a(\boldsymbol{x};\theta^t)|} \quad (9)$$

Next, for all images $\boldsymbol{x}_i \in D_{\text{train}}$, we rank them according to their prediction changes $S_{\theta',\theta^t}(\boldsymbol{x})$ in descending order. Obviously, clean images are supposed to be ranked higher. And we can fetch the top-ranked images to form the extra dataset D_{ext}^{t+1}.

Table 1. Comparison with SoTA defense methods (at all-to-one setting) on **CIFAR-10** dataset. Our approach has two versions (*i.e.,* with or without a clean extra dataset), while all other methods **use** the clean extra dataset. If our approach use such clean extra dataset, it remarkably outperforms other methods. If not using such clean extra dataset, it can still defend against most attacks except the Badnet.

	Before		**Fine-tuning**		**Fine-pruning**		**NAD**		**Neural Cleanse**		**ANP**		**PBE (w/o clean)**		**PBE (w/ clean)**	
	ACC	ASR	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
Badnet	94.67	100.00	85.82	6.53	89.80	70.66	88.09	2.17	93.73	0.83	93.39	1.66	94.02	11.30	94.20	1.09
Blend	94.63	100.00	87.53	11.31	89.30	65.86	90.13	1.60	93.28	0.63	92.03	1.81	93.04	1.16	93.98	0.93
SIG	94.81	98.96	87.34	4.14	88.93	85.69	90.26	4.59	92.23	1.79	92.48	1.27	93.56	1.76	93.35	1.39
DynamicAtt	94.65	99.24	94.00	8.77	89.91	98.97	94.23	4.59	94.65	99.24	93.42	1.36	93.01	1.12	93.01	1.12
WaNet	94.15	99.50	93.42	12.80	89.86	99.36	94.02	8.37	94.15	99.50	93.36	0.62	93.67	0.86	94.32	0.46

Table 2. Comparison with SoTA defense methods (at all-to-one setting) on **GTSRB** dataset.

	Before		**Fine-tuning**		**Fine-pruning**		**NAD**		**Neural Cleanse**		**ANP**		**PBE (w/o clean)**		**PBE (w/ clean)**	
	ACC	ASR	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
Badnet	99.02	100.00	95.01	8.32	89.60	75.03	94.22	2.06	96.78	0.12	95.13	1.35	90.13	12.31	94.43	0.47
Blend	99.39	99.92	90.68	40.12	88.21	90.53	92.61	8.56	96.48	5.81	94.02	2.68	90.10	8.74	94.57	1.72
SIG	98.56	95.81	91.63	36.30	89.53	93.26	92.94	6.90	93.40	1.32	93.32	3.65	90.31	7.59	94.05	1.78
DynamicAtt	99.27	99.84	97.10	16.33	89.15	97.21	98.17	3.80	99.27	99.84	95.88	1.68	96.68	0.99	96.68	0.99
WaNet	98.97	98.78	96.70	4.20	87.49	98.79	97.07	2.20	98.97	98.78	96.47	0.94	91.23	3.63	96.56	0.47

4. Experiment

4.1. Experimental Setting

Backdoor Attacks. We consider 5 state-of-the-art backdoor attacks: 1) BadNets [6], 2) Blend attack [3] 3) Sinusoidal signal attack(SIG) [1], 4) Input-aware dynamic attack(DynamicAtt) [19], and 5) Warpping-based attack(WaNet) [20]. We test the performance of all attacks and erasing methods on two benchmark datasets: CIFAR-10 [11], GTSRB [24]. For a fair evaluation, we use Preactivation Resnet-18 [8] as the classification model. For the hyperparameters of adversarial perturbations, we adaptively set them to different values for each backdoor attack.

Backdoor Defense and Configuration. We compare our PBE approach with 5 existing backdoor erasing methods: 1) the standard Fine-tuning [17], 2) Fine-pruning [14], 3) Neural Cleanse(NC) [27], 4) Neural Attention Distillation (NAD) [12], and 5) Adversarial Neuron Pruning (ANP) [4]. Regarding the clean extra data, we follow the same protocol of these methods: the extra clean data is randomly selected from clean training data, taking about 5% of all training data.

Evaluation Metrics. We evaluate the performance of defense mechanisms with two metrics: attack success rate (ASR), which is the ratio of triggered examples those are misclassified as the target label, and model's accuracy on clean samples (ACC). An ideal defense should lead to large ASR drops with small ACC penalties.

4.2. Comparison to SoTA Defense Methods

In our experiments, all existing methods will use the clean extra dataset. In contrast, our approach could work either with or without such an extra dataset, both of them are reported in Table.1 and Table.2.

From Table.1 and 2, if a clean extra dataset is available, our PBE defense can remarkably reduce ASR (*e.g.,* down to **1.09%**), meanwhile keep the ACC (*e.g.,* at **94.2%**). It indicates our approach outperforms other methods under the same condition (using a clean extra dataset). Besides, the Neural Cleanse (NC) cannot defend against the content-aware attacks (*e.g.,* DynamicAtt, WaNet). It is due to that such trigger synthesis based methods need to recover a trigger, but content-aware attacks make triggers adaptive to image content, rather than using a fix trigger.

In addition, if such a clean extra dataset is unavailable, all other defense methods **cannot** work, but our approach could still achieve excellent defensive performance (*e.g.,* down to ASR=1.16%) against invisible-trigger attacks (*e.g.,* Blend, SIG, WaNet). Note that visible-trigger attacks (*e.g.,* BadNet) can efficiently backdoor a model by using only several poisoning images. And our approach cannot perfectly filter out all poisoning images in the extra dataset, such that the defensive effect is a little weak (ASR=11.3%).

All-to-all Attack Setting: The previous comparisons are evaluated under the all-to-one attack setting, and we further evaluate our approach under the all-to-all attack setting. Following previous methods [20], we set target-label as $y + 1$. From Table.3, it is obvious that our approach is also very effective in this attack setting. Note that Neural Cleanse has poor defensive performance for the all-to-all attack setting.

4.3. More Results for Our Observation

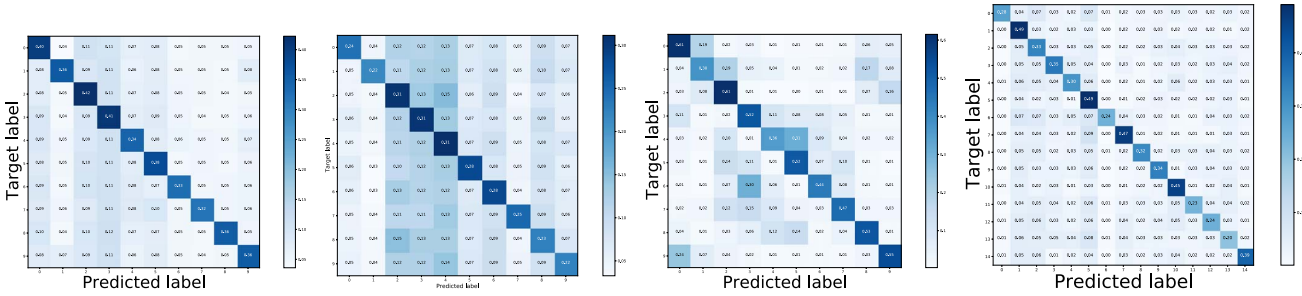
4.3.1 Predicted Labels v.s. Target-labels

Fig.1 has illustrated one example of our observation that the adversarial examples are highly likely to be classified as target-label, under the condition of WaNet attack with all-to-one setting for CIFAR-10 dataset. In this section, we illustrate that such observations are present regardless of what attack methods are (*e.g.,* Blend, SIG, WaNet), what attack settings are (*e.g.,* All-to-one and All-to-all), and what datasets are (*e.g.,* CIFAR-10 and GTSRB).

Finally, we observe similar trends, *i.e.*, the dominant predicted labels always align to the target-labels, as shown by the diagonal of the matrix in Fig.4. Due to the limited space, for the GTSRB dataset we only randomly select 15 from 43

Table 3. Comparison with SoTA defense methods (at **all-to-all setting**) on CIFAR-10 dataset.

	Before		Fine-tuning		Fine-pruning		NAD		Neural Cleanse		ANP		PBE (w/o clean)		PBE (w/ clean)	
	ACC	ASR	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
Badnet	94.63	94.41	85.54	3.68	88.20	66.13	89.63	1.01	94.63	94.41	92.01	0.69	93.62	0.68	93.84	0.62
Blend	94.89	87.94	86.60	5.36	87.96	74.15	89.91	2.38	94.89	87.94	93.12	1.24	92.76	0.84	93.65	0.68
SIG	94.66	84.34	87.98	2.83	88.99	69.52	91.53	1.36	94.66	84.34	93.60	0.87	93.71	1.07	93.52	1.01
DynamicAtt	94.40	92.72	92.05	4.46	89.62	90.33	92.71	1.39	94.40	92.72	92.86	1.09	93.28	0.75	93.28	0.75
WaNet	94.49	93.47	93.37	7.81	89.02	92.53	93.68	3.05	94.49	93.47	93.21	0.99	93.24	1.02	93.45	0.80



(a) Blend Attack (all-to-one). (b) SIG Attack (all-to-one). (c) WaNet Attack (all-to-all). (d) GTSRB-sub. (all-to-one).

Figure 4. Predicated labels v.s. Target-labels for adversarial examples from CIFAR-10 and GTSRB. No matter what attack methods are (*e.g.,* Blend, SIG, WaNet), what attack settings are (*e.g.,* all-to-one, all-to-all), what datasets are (*e.g.,* CIFAR-10, GTSRB), the dominate predicted labels always align to the target-labels, as shown by the diagonal of the matrix.

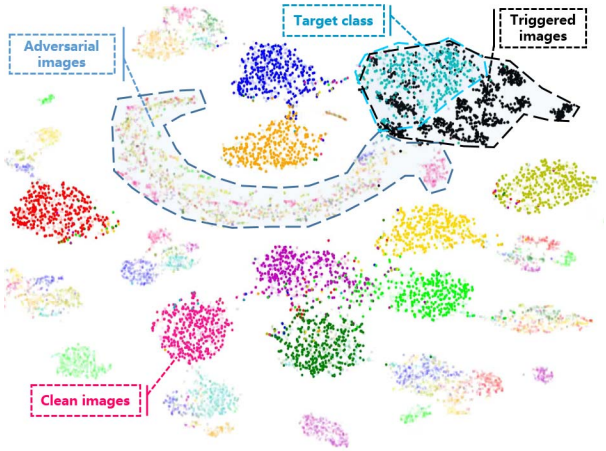


Figure 5. The visualization of image features from CIFAR-10 dataset. The original images are in dark color (*e.g.,* dark-red, dark-yellow, dark-green, *etc*), while the corresponding adversarial examples are in light color (*e.g.,* light-red, light-yellow, light-green, *etc*). In this case, the target class is shown in cyan. The triggered images are shown in black, which is close to target class. Obviously, lots of adversarial examples lie on a 'belt' which is close to the triggered images.

classes. More results for different configurations are provided in the supplemental material.

4.3.2 Comparisons of Feature Similarity

Fig.3 qualitatively illustrates that the features of an adversarial image $\widetilde{\boldsymbol{x}}'$ are very similar to that of the triggered image $\boldsymbol{x}^{\boldsymbol{t}}$, rather than the clean image $\widetilde{\boldsymbol{x}}$. Here we conduct more quantitative comparisons.

Table 4. Quantitative comparisons for feature similarity.

	Badnet	SIG	Blend	DynamicAtt	WaNet
D_{benign}	102.58	135.91	124.22	40.42	48.13
D_{infected}	85.11	78.18	75.09	28.66	15.85

We randomly sample $10,000$ images from CIFAR-10, and calculate the l_2 distances between the features of $\widetilde{\boldsymbol{x}}'$ and $\boldsymbol{x}^{\boldsymbol{t}}$, *i.e.*, $D_{\text{infected}} = ||f(\widetilde{\boldsymbol{x}}'), f(\boldsymbol{x}^{\boldsymbol{t}})||_2$. Meanwhile, we also calculate the l_2 distances between the features of $\widetilde{\boldsymbol{x}}$ and $\boldsymbol{x}^{\boldsymbol{t}}$, *i.e.*, $||D_{\text{benign}} = f(\widetilde{\boldsymbol{x}}), f(\boldsymbol{x}^{\boldsymbol{t}})||_2$. Regarding image features $f()$, we adopt the output of the last convolution layer (just before the fully-connected layer) as image features. From Table 4, we can see that after planting a backdoor into a model, the feature distances D_{infected} is smaller than D_{benign} significantly.

Furthermore, we visualize those high-dimensional features in a 2D space with t-SNE. Fig.5 shows the features of the original clean images in dark colors, their corresponding adversarial images in light colors, and the triggered images (in black). In this case, the backdoor target-label is shown in cyan. From Fig.5, it is obvious that most of adversarial images, triggered images, and target-label images lie on a same data manifold. It also justifies that the adversarial images are very similar to the triggered images.

4.4. Progressive Learning

4.4.1 Identifying Clean Images

In our approach, we formulate the clean image identification as an image ranking problem. Thus, we can evaluate its performance by using Precision-Recall curve and the Aver-

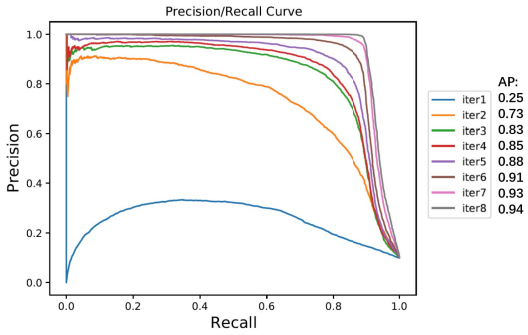


Figure 6. The progress of clean-image identification with respect to the increase of iterations.

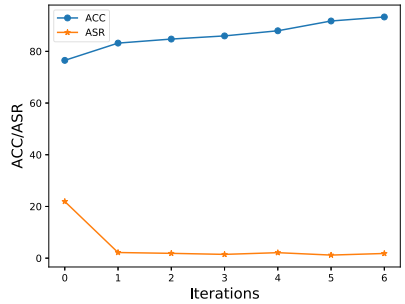


Figure 7. The progress of purified model with respect to the increase of iterations.

age Precision (AP) score. Specifically, for all training images $\boldsymbol{x}_i \in D_{train}$, suppose we know which one is poisoned, which is regarded as ground-truth. Then, we rank them in descending order according to their prediction changes $S_{\theta',\theta^t}(\boldsymbol{x})$ as Eq.(9). In our approach, the more top an image is ranked, the most possible it is regarded as a clean image. Therefore, we can evaluate our approach by comparing our ranking results to the ground-truth ranks.

More importantly, our approach has an iterative procedure, which will **gradually improve the quality of purified model and extra dataset**. In practice, at the first iterations the infected model is not well purified, and hence we just select top ranked 10% or 20% training data as extra dataset. With improvement of the model purification, the quality of our clean image ranking and identification is also improved, and we will fetch more data into extra dataset (*i.e.*, top ranked 70% images).

Fig.6 is an example for our approach to defend against blend attack on CIFAR-10 dataset. Obviously, with the increase of iterations, the precision-recall curve becomes more and more better. It indicates that the quality of clean image identification is progressively improved. Beside, the corresponding Average precision (AP) raises from 0.25 to 0.93 gradually.

4.4.2 Progress of Purified Model

Meanwhile, the quality of purified model is also improved with the increase of iterations. From Fig.7, we can see that with the increase of iterations the ASR drops and ACC raises gradually, which indicates that the purified model is improved better and better on both benign and poisoning images. Particularly, the backdoor can be quickly erased (reducing ASR quickly) at beginning iterations, while the following iterations mainly help to improve the performance on benign images (raising ACC gradually).

4.5. Comparison to Data Filtering based Defenses.

We also conduct comparisons to data filtering based defense methods, *i.e.*, *Spectral Signatures* [26] and *Spectre* [7]. Following their attack setting, 500 training images are randomly selected to be poisoned. And all these methods need to filter out the poisoning data to obtain a clean model.

Table 5. Poisoning Data Filtering.

	Spectral	SPECTRE	PBE
Badnet	500/500	500/500	495/500
Blend	133/500	499/500	496/500
SIG	470/500	497/500	476/500

Table 6. Final Defense Results (ACC/ASR).

	Before	Spectral	SPECTRE	PBE
Badnet	94.67/98.97	92.20/0.64	93.43/0.79	93.20/0.87
Blend	94.62/93.54	92.47/76.77	93.27/0.72	92.68/ 0.61
SIG	94.15/96.02	92.08/1.21	93.61/0.81	92.52/ 1.01

Table.5 shows the comparisons of poisoning data filtering performance, and Table.6 shows the comparisons of final defensive performance by using the filtered training data. We see that our approach outperforms the Spectral Signatures but is inferior to Spectre.

5. Conclusion

In this work, we propose a new progressive backdoor erasing approach by leveraging adversarial attack techniques. Our defense method could effectively defend against modern strong backdoor attacks (*e.g.*, DynamicAtt, WaNet), even when a clean extra dataset is unavailable.

Our approach stems from our intriguing observations that for an infected model, its adversarial examples have similar behaviors as the triggered images. And an theoretical analysis is given to explain such observations. Importantly, such an underlying connections between adversarial and backdoor attacks will encourage our community to study them jointly.

Acknowledgement

This work was supported by National Key R&D Program of China under Grant 2021YFB1714700; NSFC under Grants 62088102, 62106192 and 62272364; Grants 2022JC-41, 2022T150518, XTR042021005 and XTR072022001.

References

[1] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 101–105. IEEE, 2019. 2, 6

[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. In *SafeAI@ AAAI*, 2019. 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, 6

[4] Wu Dongxian and Wang Yisen. Adversarial neuron pruning purifies backdoored deep models. *NeurIPS*, 2021. 3, 4, 6

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 3

[6] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 2, 6

[7] 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. PMLR, 2021. 8

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 6

[9] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019. 3

[10] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 301–310, 2020. 3

[11] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 6

[12] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *International Conference on Learning Representations*, 2020. 2, 3, 6

[13] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*, 2020. 1, 2

[14] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018. 2, 3, 6

[15] 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 *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1265–1282, 2019. 3

[16] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pages 182–199. Springer, 2020. 2

[17] Yuntao Liu, Yang Xie, and Srivastava Ankur. Neural trojans. In *International Conference on Computer Design (ICCD)*, 2017. 2, 3, 6

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 3

[19] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020. 2, 3, 6

[20] Tuan Anh Nguyen and Anh Tuan Tran. Wanet-imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2020. 1, 2, 3, 6

[21] Neehar Peri, Neal Gupta, W Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, and John P Dickerson. Deep k-nn defense against clean-label data poisoning attacks. In *European Conference on Computer Vision*, pages 55–70. Springer, 2020. 3

[22] Pang Ren, Shen Hua, Zhang Xinyang, Ji Shouling, Vorobeychik Yevgeniy, Luo Xiapu, Liu Alex, and Wang Ting. A tale of evil twins: adversarial inputs versus poisoned models. *CCS*, 2020. 2

[23] Shawn Shan, Emily Wenger, Bolun Wang, Bo Li, Haitao Zheng, and Ben Y. Zhao. Gotta catch'em all: Using honeypots to catch adversarial attacks on neural networks. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 67–83, New York, NY, USA, 2020. Association for Computing Machinery. 3

[24] 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. 6

[25] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014. 1, 3

[26] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8011–8021, 2018. 3, 8

[27] 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 *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019. 3, 6

[28] Liu Yingqi, Ma Shiqing, Aafer Yousra, Lee Wen-Chuan, Zhai Juan, Wang Weihang, and Zhang Xiangyu. Trojaning attack on neural networks. *NDSS*, 2018. 2