Adversarial-Inspired Backdoor Defense via Bridging Backdoor and Adversarial Attacks

Jia-Li Yin[1, 2], **Weijian Wang**[1, 2], **Lyhwa**[2*], **Wei Lin**[3, 4*], **Ximeng Liu**[2, 5]
[1] Fujian Province Key Laboratory of Information Security and Network Systems, Fuzhou 350108, China
[2] College of Computer and Data Science, Fuzhou University, Fuzhou 350118, China
[3] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fuzhou 350118, China
[4] College of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China
[5] Lion Rock Labs of Cyberspace Security, CTIHE, Hong Kong, China
{jlyin, 221027096}@fzu.edu.cn, wlin@fjut.edu.cn, lyhwa@fzu.edu.cn, snbnix@gmail.com

Abstract

Backdoor attacks and adversarial attacks are two major security threats to deep neural networks (DNNs), with the former one is a training-time data poisoning attack that aims to implant backdoor triggers into models by injecting trigger patterns into training samples, and the latter one is a testing-time attack trying to generate adversarial examples (AEs) from benign images to mislead a well-trained model. While previous works generally treat these two attacks separately, the inherent connection between these two attacks is rarely explored. In this paper, we focus on bridging backdoor and adversarial attacks and observe two intriguing phenomena when applying adversarial attacks on an infected model implanted with backdoors: 1) the sample is harder to be turned into an AE when the trigger is presented; 2) the AEs generated from backdoor samples are highly likely to be predicted as its true labels. Inspired by these observations, we proposed a novel backdoor defense method, dubbed Adversarial-Inspired Backdoor Defense (AIBD), to isolate the backdoor samples by leveraging a progressive top-q scheme and break the correlation between backdoor samples and their target labels using adversarial labels. Through extensive experiments on various datasets against six state-of-the-art backdoor attacks, the AIBD-trained models on poisoned data demonstrate superior performance over the existing defense methods.

Introduction

Ever since deep neural networks (DNNs) have become the de facto tools for various applications, the security of DNNs has attracted rapidly growing attention, especially in the safety-critical scenarios, *e.g.*, autonomous driving (Dai et al. 2024), and object tracking (Huang et al. 2023). It has been widely demonstrated that DNNs are vulnerable to malicious attacks that can compromise their safety and reliability. One of the emerging threats is the backdoor attack (Gu, Dolan-Gavitt, and Garg 2017; Li et al. 2023; Bai et al. 2024), where an attacker aims to implant backdoor into a DNN model by injecting the trigger patterns into a small proportion of the training samples. The infected model would perform normally on clean test data but alters to the target label when the trigger pattern is presented. Such an attack allows the attacker to obtain unauthorized access to a model and cause potential accidents, posing a severe risk for real-world applications. Developing backdoor defense methods that can achieve clean models on poisoned datasets is critical.

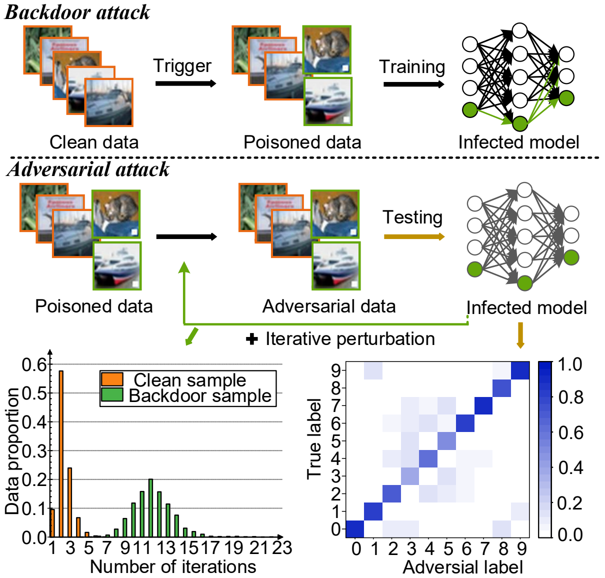


Figure 1: Illustration of our observations when applying adversarial attack on an infected model implanted with backdoors. **Top**: Backdoor attack in training time. **Middle**: Adversarial attack in inference time. During the adversarial example generation, we can observe that the backdoor samples take more iterations to be transformed into AEs (**Bottom left**), and the adversarial labels of backdoor samples are highly likely to be their true labels (**Bottom right**). The statistics are obtained by applying PGD attacks on a ResNet-18 model injected with WaNet backdoor on CIFAR-10 dataset with a 10% poisoning rate.

Besides the backdoor attack, another well-known threat to DNNs is the adversarial attack (Madry et al. 2018; Andriushchenko et al. 2020; Chen et al. 2023), which is an inference-time attack that tries to generate AEs that are similar-looking to benign images but can cause dramatic changes in DNN predictions. Since the backdoor attack and adversarial attack belongs to the training-time and inference-time attacks respectively, existing works generally treat

[*] Corresponding authors.

them as separate tasks and few works have explored the inner connections between these two attacks. Some prior attempts leverage adversarial training (AT) for erasing backdoors, however, it has been pointed out in (Gao et al. 2023; Wei et al. 2023) that AT can only be effective for certain trigger patterns and the inherent odd in AT can lead to a decrease in model clean accuracy.

In this work, we consider the problem of backdoor defense by bridging backdoor and adversarial attacks. Intuitively, we begin by thinking a question: What if we apply adversarial attacks on a model that has been backdoor attacked during training? As demonstrated in Figure 1, we experiment on a ResNet-18 with WaNet backdoor injected in the training time on CIFAR-10 dataset. During this process, we observe two intriguing phenomena: 1) generating an AE from the benign image is much harder when the trigger is presented. As shown in the bottom left of Figure 1, compared to the clean samples, it takes more iterations to transform a poisoned sample into an AE. 2) When the poisoned samples are transformed into AEs, they are highly likely to be predicted as their original labels, as shown in the bottom right of Figure 1. Note that these phenomena exist no matter what the backdoor settings are (*i.e.*, adding-based (Gu, Dolan-Gavitt, and Garg 2017), blending-based (Chen et al. 2017) or warping-based (Nguyen and Tran 2020b)).

Motivated by these two phenomena, we propose a new backdoor defense method by leveraging the properties of adversarial example generation from poisoned samples, named Adversarial-Inspired Backdoor Defense (AIBD). Please note that our findings are different from (Mu et al. 2023), where they explore the behavior similarity between adversarial example $\hat{\boldsymbol{x}}$ and poisoned sample $\boldsymbol{x}^t$, and we focus on the connections between $\boldsymbol{x}^t$ and $\widehat{\boldsymbol{x}^t}$. Specifically, we follow the Anti-Backdoor Learning (ABL) framework (Li et al. 2021b) and break the backdoor defense into two stages: 1) backdoor samples isolation and 2) model purification. In the first stage, since we find that the poisoned data are much harder to transform into AEs, we utilize the iteration of adversarial example generation as a sign of whether it is poisoned data. One problem here is that neither the proportion nor the distribution of the poisoned data is known, which significantly increases the difficulty of poisoned data isolation. To solve this problem, we propose a progressive top-q scheme where we progressively finetune the proportion q for isolating backdoor samples based on the feedback of model learning. Next, we break the connections between poisoned samples and target labels by replacing them with adversarial labels. The infected model is purified by retraining on the clean sub-set and the relabeled susceptible poison subset. By this progressive model training, our method exhibits superior performance over the existing methods. Our contributions are three folds:

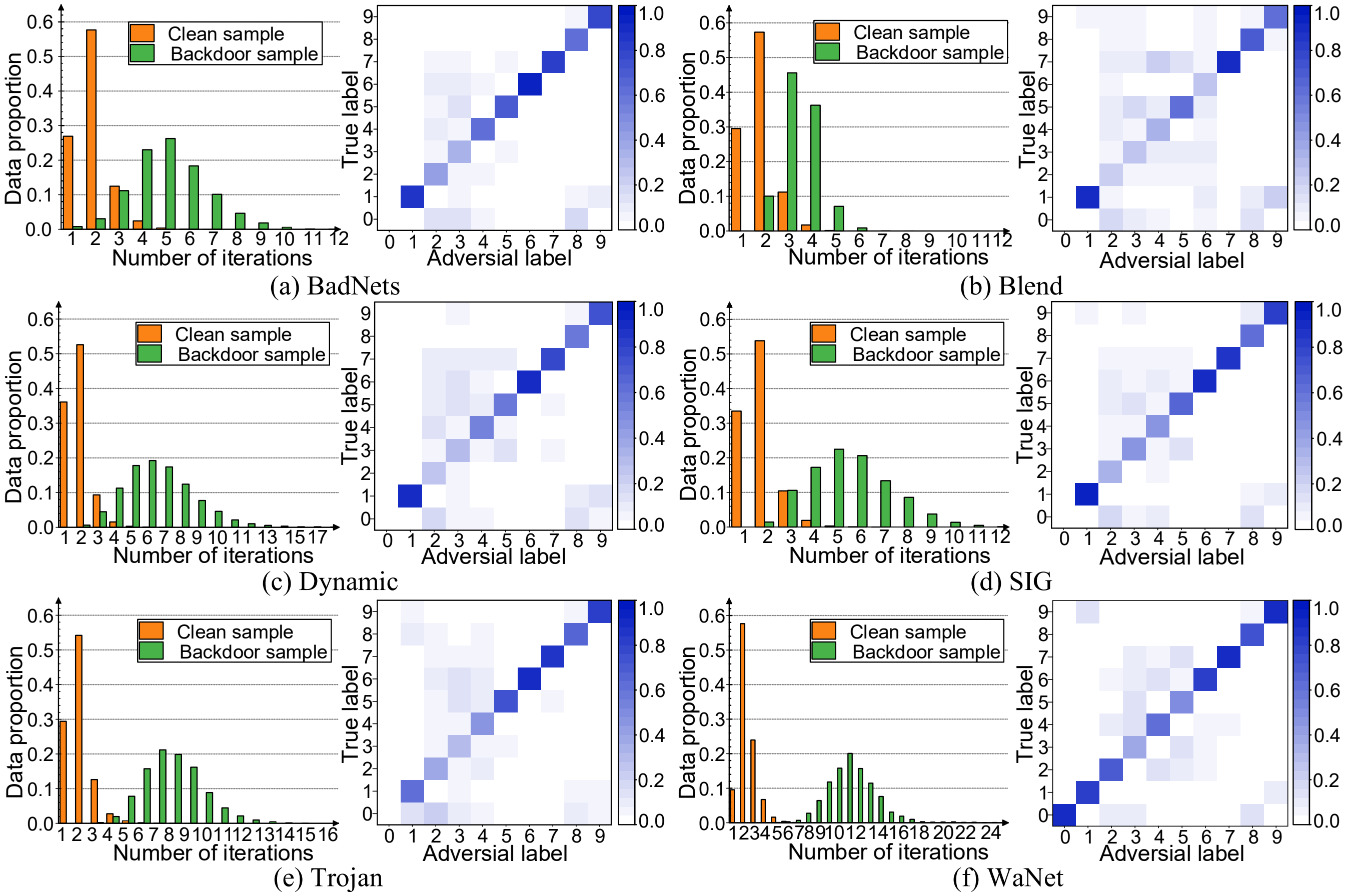
- We pioneered the use of adversarial attacks for backdoor sample identification, revealing behavioral differences between the backdoor and clean samples in the presence of adversarial attacks, *i.e.*, 1) the backdoor samples are much harder to be transformed into AEs, 2) the adversarial labels of backdoor samples are highly likely to be their clean labels.
- Motivated by these observations, we propose a new backdoor defense method accordingly, dubbed AIBD, where we first isolate the poisoned samples by leveraging a progressive top-q scheme and correct the labels of poisoned samples using their adversarial labels.
- Extensive experiments demonstrate the superiority of our AIBD to previous state-of-the-art backdoor defenses.

Related Work

Backdoor Attacks. Backdoor attacks trick the model into learning a specific trigger and target label by injecting mislabeled poisoned samples in the training set. The poisoned model behaves normally on clean inputs but produces the target output when the trigger is presented. According to the visibility of triggers, current backdoor attacks can be divided into two categories: (1) **Visible Backdoor attacks** (Gu, Dolan-Gavitt, and Garg 2017; Chen et al. 2017; Liu et al. 2018; Barni, Kallas, and Tondi 2019; Nguyen and Tran 2020a; Liu et al. 2020) add a specific trigger pattern in the benign images and improve the attacks strength by developing sample-specific patterns (Nguyen and Tran 2020a) or natural patterns (Liu et al. 2020; Yin et al. 2024). (2) **Invisible backdoor attacks** (Nguyen and Tran 2020b; Li et al. 2021a; Zhang et al. 2022; Wang, Zhai, and Ma 2022; Zhao et al. 2022; Xia et al. 2023; Dai et al. 2024) try to enhance the stealthiness of backdoor sample by making the trigger patterns invisible, such as the warpping (Nguyen and Tran 2020b) in WaNet attack. However, it also raises the difficulty for the attacker to craft poisoned samples.

Backdoor Defenses. In order to mitigate the threat of backdoors, numerous backdoor defenses have been proposed. Existing backdoor defenses can be grouped into two categories: (1) **Post-processing backdoor defenses** (Gao et al. 2019; Zeng et al. 2021; Li et al. 2021c; Guo et al. 2023; Mu et al. 2023; Li et al. 2024) intend to purify an infected model by using additional data or network. For example, Neural Attention Distillation (NAD) (Li et al. 2021c) utilizes a teacher network to guide the finetuning of the backdoored student network on a small clean subset. Scale-up (Guo et al. 2023) found that poisoned samples demonstrated scaled prediction consistency when pixel values were amplified, and they identified poisoned samples by tracking the predictions of these scaled images. (2) **Training-time backdoor defenses** (Li et al. 2021b; Huang et al. 2022; Wang et al. 2022; Zhang et al. 2023) aim to train a clean model on the poisoned dataset. Specifically, Anti-Backdoor Learning (ABL) (Li et al. 2021b) uses gradient ascent to isolate backdoor data and then unlearn these data. Decouple-based backdoor defense (DBD) (Huang et al. 2022) first adopts self-supervised learning to obtain the feature extractor, and then isolates the backdoor samples by SCEloss. Our proposed AIBD belongs to the former one.

Adversarial Attacks and Defenses. Adversarial attacks (Goodfellow, Shlens, and Szegedy 2015; Madry et al. 2018; Chen et al. 2023) are inference-time attacks that deceive trained models into making incorrect predictions by



(a) BadNets

(b) Blend

(c) Dynamic

(d) SIG

(e) Trojan

(f) WaNet

Figure 2: The performance of AEs generation from clean versus backdoor samples on injected models attacked by BadNet, Blend, Dynamic, SIG, Trojan, and WaNet. The experiment is conducted with ResNet-18 on the CIFAR-10 dataset under a poisoning rate of 10%, and the PGD attack is used to generate AEs in an iterative manner.

adding imperceptible perturbations to samples. Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2015) is first proposed to generate the perturbations by reversing the backward gradient, Projected Gradient Descent (PGD) is further developed to break the one-step perturbation generation into iterative steps, and gradually becomes the most commonly used adversarial attack. In response, adversarial training (Jiao et al. 2023; Yin et al. 2023) has been widely studied to defend against adversarial attacks. Adversarial and backdoor attacks are often treated as separate research domains. In this paper, we aim to bridge the gap between adversarial attacks and backdoor attacks to aid in backdoor defense.

Proposed Method

We first formulate the backdoor defense problem, then reveal the distinctive behaviors of backdoor and clean samples in AE transformation, and introduce our proposed AIBD method. Here we focus on classification tasks following previous works, where we have a training dataset with labels $\mathcal{D} = \{(\boldsymbol{x}_i; y_i)\}_{i=1}^{N}$, and a benign DNN model $f(\boldsymbol{x}; \boldsymbol{\theta})$.

Preliminaries

Threat Model. We adopt the poisoning-based threat model that is widely used in previous works, where attackers have full access to the training data $\mathcal{D}$ and can poison the training data by injecting the pre-defined trigger patterns into a subset $\mathcal{D}^t$. The target labels of these triggered samples can be changed to a target label l. We denote the proportion of the poisoned samples over the whole dataset as p, $p = \frac{|\mathcal{D}^t|}{|\mathcal{D}|}$. In contrast, a defender's goal is to train a well-performed model f with parameters $\boldsymbol{\theta}$ on the given poisoned dataset without prior knowledge, *e.g.*, the proportion of poisoned samples p, distribution of the samples, or the trigger patterns.

Backdoor Defense Formulation. Given the standard classification task with a poisoned dataset $\mathcal{D} = \mathcal{D}^c \bigcup \mathcal{D}^t$, where $\mathcal{D}^c$ is the clean example subset and $\mathcal{D}^t$ is the backdoor example subset. The task model f is trained by minimizing the following loss:

$$\min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{x},y)\in\mathcal{D}^c} \mathcal{L}\left(f\left(\boldsymbol{x};\boldsymbol{\theta}\right),y\right) + \sum_{(\boldsymbol{x}^t,y)\in\mathcal{D}^t} \mathcal{L}\left(f\left(\boldsymbol{x}^t;\boldsymbol{\theta}\right),y\right), \tag{1}$$

where $\mathcal{L}(\cdot)$ denotes the classification loss. The overall defense against backdoor attacks can be approached from two perspectives: 1) separating the poisoned samples from the training dataset, and 2) breaking the connections between triggers and target labels.

Adversarial Attack. Typically, an adversarial example is generated by adding a perturbation δ over original data $\boldsymbol{x}$:

$$\hat{\boldsymbol{x}} = \boldsymbol{x} + \delta, \quad \text{s.t.} \quad \hat{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}), \tag{2}$$

where $\mathcal{B}(\boldsymbol{x})$ denotes the ℓ_p-norm ball centered at $\boldsymbol{x}$ with radius ϵ, i.e., $\mathcal{B}(\boldsymbol{x}) = \{\hat{\boldsymbol{x}} : \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_p \leq \epsilon\}$. δ is the adversarial perturbation which is usually generated by reversing the model gradient. Specifically, PGD attack (Madry et al. 2018) performs in an iterative way as follows:

$$\hat{\boldsymbol{x}}_{k+1} = \Pi_{\boldsymbol{x}+\mathcal{B}}(\hat{\boldsymbol{x}}_k + \alpha\, \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x};\theta), y))), \tag{3}$$

where $\delta = \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x};\theta), y))$ is the adversarial perturbation, $\hat{\boldsymbol{x}}_k$ is initialized as the clean input $\boldsymbol{x}$, and Π refers to the projection operation for projecting the AEs back to the norm-ball. An adversarial example is obtained when $f(\hat{\boldsymbol{x}}) \neq f(\boldsymbol{x})$. In this paper, we leverage the properties of the iteration number k to identify the poisoned samples, where the poisoned samples cost much larger iterations to generate an adversarial example.

Distinctive Adversarial Attack Behaviors on Backdoor Examples

In this paper, we explore the underlying connections between backdoor and adversarial attacks by observing how the model or the samples perform when the adversarial attacks are applied in an infected model. We first use six classic backdoor attacks including BadNets, Trojan, Blend, Dynamic, and SIG with different settings to poison 10% of CIFAR-10 training data. Then six ResNet-18 models are trained on the corresponding poisoned datasets using the standard training pipeline, respectively. For each infected model, we apply the iterative-based PGD attack to generate AEs from clean and poisoned samples. We plot the adversarial example generation iterations on clean versus poisoned training examples in the left part of each subfigure in Figure 2. Clearly, for all 6 attacks, the adversarial example generation on poisoned samples is much slower than that on clean samples. Moreover, compared to BadNets and Blend, it takes more iterations to transform a backdoor sample into an adversarial example on the more powerful attack Dynamic, Trojan and WaNet. It denotes that the stronger the attack is, the more iterations the backdoor examples take to be transformed. More results on the GTSRB dataset can be found in the supplementary material.

The above observation indicates that the backdoor samples are harder to attack. This is not too surprising. In an infected model, the trigger patterns and the target labels are strongly connected, so the poisoned samples are always easy examples in the infected model testing. During the adversarial example generation, it reverses the gradient on the benign images and adds the reverse gradients on the benign images, as described in Eq. (3). Thus, the poisoned samples require more iterations to accumulate the perturbations.

We also show the adversarial labels of backdoor samples in the right part of each subfigure in Figure 2. As it depicts, when applying adversarial attacks on the infected model, we observe that at least 40% adversarial backdoor examples are predicted as their true labels. Note that this phenomenon exists in all 6 attack settings. We give speculation on why AEs of backdoor samples would be predicted as their true labels in an infected model as follows: As pointed out by previous works (Wu and Wang 2021; Mu et al. 2023), the ‘backdoor neurons’ exist in the backdoor model and would be activated if the trigger is presented. When a backdoor sample is successfully transformed into an adversarial example, it denotes that the adversarial backdoor sample has escaped the backdoor neurons and the model would focus more on the image itself.

Algorithm 1: Adversarial-Inspired Backdoor Defense(AIBD)

Input: Training dataset $\mathcal{D}$, and task model f.
Parameter: Retrain epoch R
Output: clean model f_c.

```
Train an infected model $f_p$ using the whole dataset.
Perform PGD attack on each sample $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$. Record the iterative number $k_i$ of each sample $\boldsymbol{x}_i$ and their adversarial labels $\hat{y_i}$.
Sort all the samples according to $k_i$.
for $q = 20\%, 18\%..., 2\%$ do
    Separate $\mathcal{D}$ into $\mathcal{D}^c$ and $\mathcal{D}^t$ by the proportion $q$ using Eq. (5).
    Replace the labels of samples in $\mathcal{D}^t$ with their adversarial labels $\hat{y_i}$ using Eq. (6).
    Train the task model $f$ using Eq. (1).
    if $ACC$ is close to $ACC_p$ then
        $f_c = f$
        break
    end if
end for
return clean model $f_c$.
```

Adversarial-Inspired Backdoor Defense

We decompose the entire training process into two stages, *i.e.*, dataset separation and model purification. We utilize the above observations to develop two key techniques in the proposed AIBD: 1) Backdoor sample identification using the adversarial example generation iteration k followed by Progressive Top-q scheme that progressively isolate samples, and 2) target label correction by utilizing the adversarial labels.

Dataset Separation

Given the poisoned dataset $\mathcal{D} = \mathcal{D}^c \bigcup \mathcal{D}^t$, we first train an infected model $f_p(\boldsymbol{x}, \theta)$ using the whole training dataset. Then for each image $\boldsymbol{x}_i \in \mathcal{D}$, we feed it to the infected model and apply PGD attack to generate the AEs $\hat{\boldsymbol{x}}_i$ using Eq. (3). The iterative number of each sample in adversarial example generation is noted as k_i. One may wonder the backdoor samples can be easily identified by setting a threshold to filter out the high-iteration examples. However,

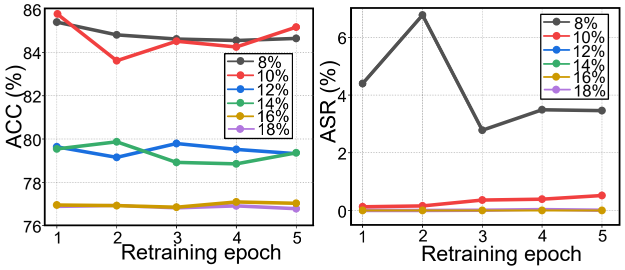


Figure 3: Model performance when using different q values in backdoor sample separation on CIFAR-10 dataset with an actual poison rate of 10%. The model accuracy improves when q is smaller since clean samples have remained as many as possible and tend to be stable when q reaches the actual poisoning rate. In contrast, ASR is stable when q is larger because the potential poison samples are filtered out but increases when q is smaller than the actual poisoning rate due to the remaining poison samples in the dataset.

this strategy is ineffective for two reasons. First, the threshold for different attacks can be different. As shown in Figure 2, the average iterative number is different for different attacks. Second, although most of the backdoor samples take a high iterative number, there are still some backdoor samples that can have a low iterative number, which makes the identification more complex.

Backdoor Sample Identification. To solve the above problems, we propose to use the top-q scheme in which we select the q samples with the highest k_i values as the backdoor samples and put them in $\mathcal{D}^t$. Note that q is a proportion. Before the selection, we first propose to amplify the behavior differences between the clean and backdoor samples. Inspired by (Li et al. 2021b), we use the following equation to achieve this:

$$k_i = (1 - \mathrm{sigmoid}(\nabla_{\boldsymbol{x}_i} \mathcal{L}(f(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i))) \cdot k_i, \qquad (4)$$

where the first term is the model gradient on the input image. Since the backdoor samples usually have a smaller gradient, doing so will make k_i larger and more distinctive from clean samples. The dataset then can be separated as follows:

$$\begin{cases} \mathcal{D}^t \leftarrow (\boldsymbol{x}_i, y_i), Rank(k_i) \leq q, \\ \mathcal{D}^c \leftarrow (\boldsymbol{x}_i, y_i), otherwise. \end{cases} \qquad (5)$$

Note that q is unknown to the defender, thus we propose a progressive top-q scheme to refine the q value based on the feedback of model training, which we illustrate in the next subsection.

Adversarial Label Correction. With the separated backdoor samples, we next break the correlations between backdoor samples and the target label. Specifically, we utilize the second property of adversarial backdoor samples: the adversarial label of a backdoor sample is highly likely to be its true label, and the labels of backdoor samples can be corrected by:

$$\hat{y}_i = f(\hat{\boldsymbol{x}}_i; \theta_p), \qquad \text{s.t. } \forall \boldsymbol{x}_i \in \mathcal{D}^t. \qquad (6)$$

Progressive Top-q Training

With the identified clean and backdoor subsets, we can then retrain a well-performed model f on the dataset. Here we first initialize q as 20% since the poisoning rate is usually under 20% according to (Chen, Wu, and Zhou 2025). As shown in Figure 3, when the q value is higher than the actual poisoning rate, the model clean accuracy significantly decreases in the initial training stage. This is because that many clean samples are categorized into $\mathcal{D}^t$ due to the high q value, leading to catastrophic forgetting (Toneva et al. 2018), where the model rapidly forgets the association between clean samples and their correct labels. If catastrophic forgetting occurs, we should immediately decrease the q value to reduce the number of filtered clean samples. When the q value goes below the actual poisoning rate, the clean accuracy tends to be stable but the ASR increases due to the remained poison samples. Since we cannot obtain the ASR in the defender's perspective, we adopt the largest q value when the model accuracy tends to be stable as the final ratio. Note that the model accuracy differs within 5 epochs, so we can obtain the best q value without heavy computation burden. The overall algorithm is summarized in Algorithm 1.

Experiments

Experimental Settings

Backdoor Attacks. We use 6 the most common backdoor attacks in our experiments: (1) BadNets (Gu, Dolan-Gavitt, and Garg 2017), (2) Blend attack (Chen et al. 2017), (3) Sinusoidal signal attack (SIG) (Barni, Kallas, and Tondi 2019), (4) WaNet attack (Nguyen and Tran 2020b), 5) Trojan attack (Liu et al. 2018), and 6) Dynamic attack (Nguyen and Tran 2020a). We evaluate the performance of these attacks and defense methods on two benchmark datasets: CIFAR-10 and GTSRB datasets. For the model architectures, we use ResNet18 on CIFAR-10 and WideResNet (WRN-16-1) on GTSRB following previous works. It is important to note that we do not apply any additional data augmentations in model training as they would hinder the backdoor effects (Liu et al. 2020).

Backdoor Defenses. We compare our AIBD with four state-of-the-art defense methods: Anti-Backdoor Learning (ABL) (Li et al. 2021b), Neural Attention Distillation (NAD) (Li et al. 2021c), Decoupling-based Backdoor Defense (DBD) (Huang et al. 2022), and Causality-inspired Backdoor Defense (CBD) (Zhang et al. 2023). We also provide the defense performance under no backdoor attack. For fair comparisons, we follow the default settings in all the compared defenses, including the availability of clean data. For our AIBD, we used the SGD optimizer with an adversarial attack step size of 1/255.

Evaluation Metrics. We adopt two common metrics to evaluate the performance of defense mechanisms: attack success rate (ASR), which is the proportion of trigger-laden samples successfully classified as the target label specified by the attacker, and model accuracy on clean samples (ACC). The defense mechanism should minimize ASR and maximize ACC. Additionally, True Positive Rate (TPR) and

Dataset	Types	No Defense		ABL		NAD		DBD		CBD		**AIBD (Ours)**	
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	*None*	88.28	0.00	66.84	0.00	86.21	0.00	78.53	0.00	79.86	0.00	**87.26**	0.00
	BadNets	87.18	98.43	82.68	6.94	81.81	1.68	82.94	0.80	84.01	4.29	**85.34**	**0.28**
	Blend	87.04	100.00	78.83	5.88	83.23	0.81	70.91	99.98	82.74	64.53	**84.91**	**0.73**
	SIG	85.75	99.98	81.43	3.90	83.18	1.20	83.52	0.44	80.34	83.31	**83.90**	**0.33**
	WaNet	86.35	100.00	79.22	17.19	82.71	1.77	84.26	6.61	85.13	3.14	**85.39**	**0.28**
	Trojan	87.48	100.00	80.95	1.38	84.57	3.59	81.22	2.24	80.73	10.88	**85.48**	**0.11**
	Dynamic	87.08	99.99	82.60	13.10	83.28	11.68	83.42	0.86	84.13	30.63	**84.89**	**0.23**
	Average	87.02	99.73	78.93	8.07	83.57	3.46	80.69	18.49	82.42	64.53	**85.31**	**0.33**
GTSRB	*None*	96.18	0.00	72.34	0.00	92.33	0.00	88.53	0.00	89.12	0.00	**96.40**	0.00
	BadNets	95.85	98.27	95.04	0.04	83.53	0.13	96.05	0.24	96.21	0.16	**95.84**	**0.00**
	Blend	95.02	99.97	92.19	4.69	80.46	**0.16**	93.72	6.36	94.16	0.90	**95.24**	0.19
	SIG	95.98	99.90	79.49	3.78	85.48	0.02	94.58	4.70	94.37	5.41	**95.34**	**0.00**
	WaNet	95.82	100.00	79.42	17.76	83.91	2.41	94.71	3.47	95.64	3.13	**95.62**	**0.00**
	Trojan	96.03	99.76	95.63	**0.01**	87.72	0.04	94.69	0.56	95.29	0.12	**96.25**	0.20
	Dynamic	96.37	99.83	95.11	1.79	90.51	0.20	95.86	5.16	**96.02**	0.96	95.86	**0.07**
	Average	95.89	99.62	87.03	4.68	86.28	0.50	94.02	3.42	94.40	1.78	**95.79**	**0.08**

Table 1: The clean accuracy ACC(%) and attack success rate ASR(%) of five backdoor defenses against six backdoor attacks on CIFAR-10 and GTSRB datasets. *None* denotes that the training data is completely clean without being poisoned. The best results are **bolded**.

False Positive Rate (FPR) are used to evaluate the screening performance, where higher TPR and lower FPR are desired.

Main Results

Comparison to Existing Defenses. We first evaluate the effectiveness of AIBD by comparing it to the existing SOTA backdoor defense methods. The results are summarized in Table 1. We can have the following observations: First, the proposed AIBD can effectively defend against all the backdoor attacks. Specifically, the average ASR decreases from 99.73% to 0.33% on CIFAR-10 dataset, and 99.62% to 0.08% on GTSRB dataset. Furthermore, when tested on the GTSRB dataset, the proposed AIBD can achieve 0.00% ASR against BadNets, SIG, and WaNet attacks, which denotes that our method can completely remove the effects of poisoned samples. These encouraging results validate that the proposed AIBD can effectively defend against backdoor attacks and perform generally well on different datasets. Second, compared to other defense schemes, our method achieves superior defense without harming clean accuracy. We can see that using ABL incurs significant ACC decreases when there are no attacks on both datasets. This is because ABL always selects 1% of samples for gradient ascent. In the absence of attacks, 1% of clean samples would be chosen, which inevitably has a huge impact on ACC. In comparison, NAD achieves the closest ACC performance to ours, but NAD relies on an additional clean subset, and the quality of the clean subset is positively correlated with maintaining ACC performance. DBD exhibits lower ASR in most cases because it treats most samples as unlabeled data, effectively preventing backdoor injection. However, some attacks still escape the defense, such as Blend on CIFAR-10. In contrast, our AIBD can well defend all the attacks with high ACC and low ASR values.

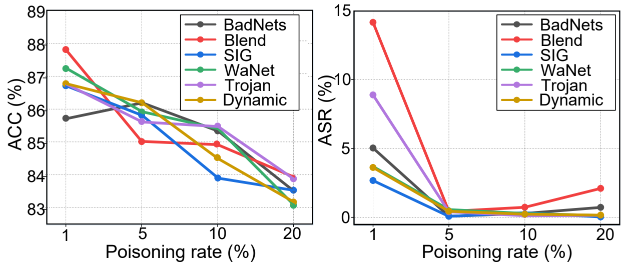


Figure 4: Model performance on CIFAR-10 dataset with the poisoning rate ranging from 1% to 20% for BadNets, Blend, SIG, WaNet, Trojan, and Dynamic attacks.

Effectiveness with Different Poisoning Rates. In Figure 4, we demonstrate the defense performance of the proposed method with poisoning rates ranging from 1% to 20%. Here, we use six backdoor attacks on the CIFAR-10 dataset using the previous settings. We can see that when the poisoning rate is extremely low at 1%, both the ACC and ASR values are high. This is because that the impact of backdoor on the model is relatively weakened when the backdoor samples are very few, making it more difficult to distinguish between backdoor and clean samples. When the poisoning rate goes higher, the ASR drops and the ACC can remain robust from 1% to 10%. It indicates that our method can accurately isolate the backdoor samples and purify the model. At the poisoning rate of 20%, the ASR picks up slightly and the ACC drops due to the incomplete isolation of the backdoor samples at such a high poisoning rate. Despite these modest changes, our method demonstrates a steady and acceptable performance across different poisoning rates overall.

Types	BadNets		Blend		SIG		Dynamic		Trojan		Average	
Defense	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
None (Neural activation)	99.80	18.75	78.84	33.28	21.08	36.38	99.84	**0.01**	100.00	**0.74**	79.91	17.83
ABL (Loss isolation)	94.76	**0.68**	87.68	**0.64**	90.60	0.49	93.00	1.42	88.44	2.18	90.90	**1.08**
DBD (SCELoss)	80.67	48.52	**99.44**	47.39	38.36	50.61	21.48	51.50	**99.64**	47.38	67.92	49.08
SCALE-UP (Scale consistency)	32.36	32.54	39.36	40.05	27.64	27.38	28.72	27.79	30.56	30.98	31.73	31.75
AIBD (Ours)	**96.66**	1.40	97.26	1.12	**98.70**	1.70	**99.62**	1.22	99.49	1.30	**98.35**	1.35

Table 2: Comparison of different isolation schemes against various attacks on the CIFAR-10 dataset using TPR and FPR (%) . The best results are **bolded** and the second best results are underlined.

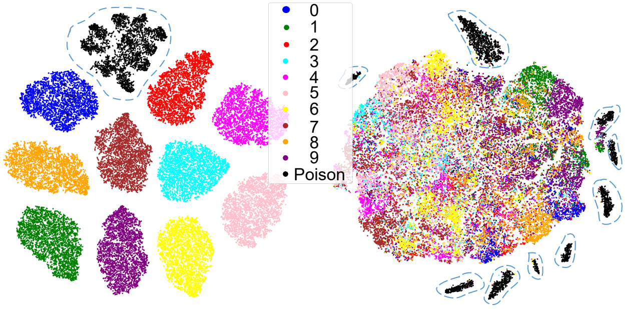
Types	Unlearning		Random Labels		Adv. Labels	
	ACC	ASR	ACC	ASR	ACC	ASR
BadNets	84.82	0.72	82.36	1.84	**85.34**	**0.28**
Blend	67.69	**0.69**	83.32	4.22	**84.91**	0.73
SIG	75.58	**0.13**	81.63	0.27	**83.90**	0.33
WaNet	75.00	12.51	84.32	0.24	**85.39**	**0.28**
Trojan	84.06	0.37	84.84	0.28	**85.48**	**0.11**
Dynamic	83.97	2.20	82.98	0.69	**84.89**	**0.23**
Average	78.52	2.77	83.24	1.26	**84.99**	**0.33**

Table 3: Comparison of using Unlearning, Random Labeling, and Adversarial Labeling in backdoor sample learning.

Comparisons on Isolation Quality. We now clarify the isolation quality of our algorithm. Table 2 reports the isolation results of different methods. As we can see, our method achieves the best TPR and the second-best FPR among all the defenses. Our TPR averages as high as 98.35%, indicating that almost all backdoor samples can be detected. Meanwhile, our FPR is only 1.35%. ABL has the lowest FPR, which is 0.27% lower than our method. However, it has a TPR that is 7.45% lower than ours. The remaining defenses exhibit less consistent performance and can only achieve optimal performance on individual attacks.

Further Analysis

Adversarial Labels v.s. Random Labels v.s. Unlearning. In this work, we find that the adversarial examples of backdoor samples are highly likely to be classified as their true labels in the infected model, thus we use the adversarial labels to correct the identified backdoor sample labels in the model training. Here we compare this scheme with unlearning and random labeling in backdoor sample learning. Specifically, we adopt the unlearning scheme in (Li et al. 2021b) to unlearn the backdoor samples, and replace the adversarial labeling with random labeling in our framework, respectively. The experiments are conducted on the CIFAR-10 dataset following previous experimental settings. The results are summarized in Table 3. One can see that unlearning the backdoor samples can achieve a low ASR against Blend and SIG attack due to the loss maximization between backdoor samples and the target label; however, the clean accuracy drops significantly. On the contrary, using the adversarial labels consistently achieves high ACC and low ASR, where the average ACC improves 1.75% and the ASR drops 0.93%.



(a) (b)

Figure 5: Visualization of image features from CIFAR-10 in the infected model before (a) and after (b) adversarial attack.

t-SNE Visualization. To further demonstrate the distinctive behavior between clean and backdoor samples before and after the adversarial example transformation in the infected model, we visualize the feature space of the infected model using t-SNE embeddings in Figure 5. As we can see, before the adversarial attack in Figure 5 (a), the backdoor samples cluster together since they are strongly connected to their target labels. When performing adversarial attack on the infected model in Figure 5 (b), almost all the adversarial clean samples reside in a large area while the adversarial backdoor samples are separately distributed according to their true labels, which further validates our observation.

Conclusions

In this paper, we propose the Adversarial-Inspired Backdoor Defense (AIBD) by bridging the adversarial and backdoor attacks. We identify two characteristics when applying adversarial attacks on an infected model: 1) Compared to clean samples, the backdoor samples are much harder to be transformed into an adversarial example, and 2) the adversarial backdoor samples are highly likely to be classified as their true labels. Motivated by these findings, we propose a progressive top-q scheme to isolate the backdoor samples by utilizing the iterative number in adversarial example transformation, and replacing their labels with adversarial labels in model purification. Extensive experiments demonstrate that our AIBD is resilient to various experimental settings and can effectively defend against various backdoor attacks.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (2023YFE0111100); the National Natural Science Foundation of China under Grant Nos. 62202104 and U2441239; the Scientific Research and Development Foundation of Fujian University of Technology under No. GYZ220209; the Key Technology Innovation Tackling and Industrialization Project of Fujian Province under Grant No.2024G018; the Major Science and Technology Project of Fuzhou under Grant No.2023-ZD-003; and the Open Topics from the Lion Rock Labs of Cyberspace Security under the project #LRL24006.

References

Andriushchenko, M.; Croce, F.; Flammarion, N.; and Hein, M. 2020. Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*, 484–501. Springer.

Bai, J.; Gao, K.; Min, S.; Xia, S.-T.; Li, Z.; and Liu, W. 2024. BadCLIP: Trigger-Aware Prompt Learning for Backdoor Attacks on CLIP. In *CVPR*.

Barni, M.; Kallas, K.; and Tondi, B. 2019. A new backdoor attack in cnns by training set corruption without label poisoning. In *ICIP*, 101–105. IEEE.

Chen, B.; Yin, J.; Chen, S.; Chen, B.; and Liu, X. 2023. An Adaptive Model Ensemble Adversarial Attack for Boosting Adversarial Transferability. In *ICCV*, 4489–4498.

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

Chen, Y.; Wu, H.; and Zhou, J. 2025. Progressive poisoned data isolation for training-time backdoor defense. In *AAAI*, 11425–11433.

Dai, X.; Guo, C.; Tang, Y.; Li, H.; Wang, Y.; Huang, J.; Tian, Y.; Xia, X.; Lv, Y.; and Wang, F.-Y. 2024. VistaRAG: Toward Safe and Trustworthy Autonomous Driving Through Retrieval-Augmented Generation. *IEEE Transactions on Intelligent Vehicles*.

Gao, Y.; Wu, D.; Zhang, J.; Gan, G.; Xia, S.-T.; Niu, G.; and Sugiyama, M. 2023. On the Effectiveness of Adversarial Training Against Backdoor Attacks. *IEEE Transactions on Neural Networks and Learning Systems*, 1–11.

Gao, Y.; Xu, C.; Wang, D.; Chen, S.; Ranasinghe, D. C.; and Nepal, S. 2019. Strip: A defence against trojan attacks on deep neural networks. In *ACSAC*, 113–125.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*.

Guo, J.; Li, Y.; Chen, X.; Guo, H.; Sun, L.; and Liu, C. 2023. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. *arXiv preprint arXiv:2302.03251*.

Huang, B.; Yu, J.; Chen, Y.; Pan, S.; Wang, Q.; and Wang, Z. 2023. BadTrack: A Poison-Only Backdoor Attack on Visual Object Tracking. In *NeurIPS*.

Huang, K.; Li, Y.; Wu, B.; Qin, Z.; and Ren, K. 2022. Backdoor Defense via Decoupling the Training Process. In *ICLR*.

Jiao, R.; Liu, X.; Sato, T.; Chen, Q. A.; and Zhu, Q. 2023. Semi-supervised Semantics-guided Adversarial Training for Robust Trajectory Prediction. In *CVPR*, 8207–8217.

Li, B.; Cai, Y.; Li, H.; Xue, F.; Li, Z.; and Li, Y. 2024. Nearest is Not Dearest: Towards Practical Defense against Quantization-conditioned Backdoor Attacks. In *CVPR*, 24523–24533.

Li, Y.; Li, Y.; Wu, B.; Li, L.; He, R.; and Lyu, S. 2021a. Invisible backdoor attack with sample-specific triggers. In *CVPR)*, 16463–16472.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021b. Anti-backdoor learning: Training clean models on poisoned data. *NeurIPS*, 34: 14900–14912.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021c. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *ICLR*.

Li, Y.; Zhang, S.; Wang, W.; and Song, H. 2023. Backdoor Attacks to Deep Learning Models and Countermeasures: A Survey. *IEEE Open Journal of the Computer Society*.

Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018. Trojaning attack on neural networks. In *NDSS*. Internet Soc.

Liu, Y.; Ma, X.; Bailey, J.; and Lu, F. 2020. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 182–199. Springer.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*.

Mu, B.; Niu, Z.; Wang, L.; Wang, X.; Miao, Q.; Jin, R.; and Hua, G. 2023. Progressive Backdoor Erasing via connecting Backdoor and Adversarial Attacks. In *CVPR*, 20495–20503.

Nguyen, T. A.; and Tran, A. 2020a. Input-aware dynamic backdoor attack. *NeurIPS*, 33: 3454–3464.

Nguyen, T. A.; and Tran, A. T. 2020b. WaNet-Imperceptible Warping-based Backdoor Attack. In *ICLR*.

Toneva, M.; Sordoni, A.; des Combes, R. T.; Trischler, A.; Bengio, Y.; and Gordon, G. J. 2018. An Empirical Study of Example Forgetting during Deep Neural Network Learning. In *ICLR*.

Wang, Z.; Ding, H.; Zhai, J.; and Ma, S. 2022. Training with more confidence: Mitigating injected and natural backdoors during training. *NeurIPS*, 35: 36396–36410.

Wang, Z.; Zhai, J.; and Ma, S. 2022. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *CVPR*, 15074–15084.

Wei, S.; Zhang, M.; Zha, H.; and Wu, B. 2023. Shared Adversarial Unlearning: Backdoor Mitigation by Unlearning Shared Adversarial Examples. In *NeurIPS*.

Wu, D.; and Wang, Y. 2021. Adversarial Neuron Pruning Purifies Backdoored Deep Models. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *NeuralIPS*.

Xia, P.; Niu, H.; Li, Z.; and Li, B. 2023. Enhancing backdoor attacks with multi-level mmd regularization. *IEEE Transactions on Dependable and Secure Computing*, 20(2): 1675–1686.

Yin, J.-L.; Chen, B.; Zhu, W.; Chen, B.-H.; and Liu, X. 2023. Push stricter to decide better: A class-conditional feature adaptive framework for improving adversarial robustness. *TIFS*, 18: 2119–2131.

Yin, W.; Lou, J.; Zhou, P.; Xie, Y.; Feng, D.; Sun, Y.; Zhang, T.; and Sun, L. 2024. Physical Backdoor: Towards Temperature-based Backdoor Attacks in the Physical World. In *CVPR*, 12733–12743.

Zeng, Y.; Chen, S.; Park, W.; Mao, Z.; Jin, M.; and Jia, R. 2021. Adversarial Unlearning of Backdoors via Implicit Hypergradient. In *ICLR*.

Zhang, J.; Dongdong, C.; Huang, Q.; Liao, J.; Zhang, W.; Feng, H.; Hua, G.; and Yu, N. 2022. Poison ink: Robust and invisible backdoor attack. *IEEE Transactions on Image Processing*, 31: 5691–5705.

Zhang, Z.; Liu, Q.; Wang, Z.; Lu, Z.; and Hu, Q. 2023. Backdoor Defense via Deconfounded Representation Learning. In *CVPR*, 12228–12238.

Zhao, Z.; Chen, X.; Xuan, Y.; Dong, Y.; Wang, D.; and Liang, K. 2022. Defeat: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In *CVPR*, 15213–15222.