IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 November 2023, accepted 11 January 2024, date of publication 15 January 2024, date of current version 23 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3354385

RESEARCH ARTICLE

Defending Against Backdoor Attacks by Quarantine Training

CHENGXU YU AND YULAI ZHANG
School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China
Corresponding author: Yulai Zhang (zhangyulai@zust.edu.cn)

This work was supported in part by the Young Scientists Fund of the National Natural Science Foundation of China under Grant 61803337, and in part by the Science & Technology Development Item of Hangzhou City under Grant 2022AIZD0056.

ABSTRACT Deep neural networks (DNNs) are powerful yet vulnerable to backdoor attacks simply by adding backdoor samples to the training set without controlling the training process. To filter out the backdoor samples in the training set, this paper proposes a novel and effective backdoor defense method called Quarantine Training (QT). Specifically, QT creates a quarantine class for each class in the training set and relabels all sample labels to associate with their corresponding quarantine classes during training. In this process, the backdoor samples are gradually categorized into the quarantine classes, thus effectively filtering out the backdoor samples. Experiments on multiple benchmark datasets with a variety of backdoor attacks demonstrate that QT has state-of-the-art backdoor defense performance without reducing the prediction accuracy of benign samples - and even improving it. Our codes are available at https://github.com/Chengx-Yu/Quarantine-Training.

INDEX TERMS Deep neural networks, trustworthy A, backdoor attack, backdoor defense.

I. INTRODUCTION

The training of deep neural networks (DNNs) is data-driven, and the final prediction accuracy of their training greatly depends on the quality of the training dataset. The powerful fitting ability of DNNs makes it possible to fit arbitrary distributions within the dataset, so they are widely used in applications such as autonomous driving [1], object detection [2], and face recognition [3]. Due to this property a new security threat, i.e., backdoor attacks [4], is introduced to DNNs. Adversaries can maliciously manipulate the predictions of DNNs to their target classes by poisoning a small number of training samples without controlling the training process, which is known as the backdoor attack [5].

When training DNNs, there is a tendency to obtain datasets from communities on the web, such as hugging face. However, datasets from some untrusted third-party sources may contain backdoor samples within them. While some backdoor samples may have obvious triggers that change the labels of the original samples, it is extremely challenging to filter them manually. Even professional data labeling experts cannot always confirm whether a sample is a backdoor sample or not [6]. Training a backdoor-free security model on a training set containing backdoor samples is also a challenge.

In this paper, we leverage two properties of backdoor attacks for backdoor sample filtering and backdoor-free security model learning called *quarantine training* (QT) (see Fig. 1). First, backdoor samples are easier to fit than clean samples in model training, a property also revealed in [7]. We conduct additional research into the varying performance of backdoor samples compared to clean samples during training. This investigation leads us to conclude that the model can fit backdoor samples as a separate class. This suggests that backdoor samples can be bound to any class when they are available - even classes that do not exist in the original data distribution.

Using the first property, we first trained a pre-quarantine model and used it to filter out a very small fraction of the training samples. The filtered samples are used as pre-quarantine samples. In the formal phase of QT, we constructed its corresponding quarantine class for each class in the original training set. Using the second feature, we relabeled each training sample, which is called a quarantine label, such that each training sample is associated with its source class

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

and its quarantine class. During the training process, fixing the quarantine labels of the pre-quarantine samples and dynamically adjusting the quarantine labels of the other samples makes the backdoor samples gradually separate from the clean samples. Therefore, based on our defense, the backdoor samples in the training set can be successfully filtered out, and at the same time, since QT changes the target labels of the backdoor samples, it makes the adversary's original purpose impossible to be achieved and achieves the training of a backdoor-free security model.

The main contributions of this paper are as follows:

- We analyze the difference between backdoor samples and clean samples in terms of their actual probability distributions, revealing that backdoor samples can be fitted to models as an independent class.
- We propose quarantine training, which can be used for filtering backdoor samples and the training of backdoor-free security models.
- We validate the effectiveness of our method by conducting several defense experiments of backdoor attacks on classical benchmark datasets. The models trained with QT can filter out close to 100% of backdoor samples, giving the model a 0% success rate for backdoor attacks, while even enabling these models to improve their accuracy on clean samples.

II. RELATED WORK

A. BACKDOOR ATTACK

Backdoor attacks achieve the manipulation of DNNs prediction results by injecting backdoor samples in the training set. In this paper, we focus on backdoor attacks based on training set poisoning rather than backdoor attacks that require training process control [8], [9]. According to the pixel relationship between the trigger of the backdoor sample and the original sample, backdoor attacks can be categorized into two types: overwrite-based backdoor attacks and blend-based backdoor attacks.

1) OVERWRITE-BASED BACKDOOR ATTACKS

BadNets [4], proposed by Gu et al, was the first backdoor attack. By adding a fixed special pattern to a random portion of the training samples and modifying their labels to a specified class, BadNets can then predict any sample with a special pattern to a specified class in the inference phase. In addition to this simple trigger, there are also backdoor triggers generated using reverse engineering [10], which improves the robustness of backdoor attacks. After that, Nguyen et al. [11] explored the use of generative adversarial networks to generate sample-specific trigger patterns, extending the trigger generation paradigm.

2) BLEND-BASED BACKDOOR ATTACKS

Blend-based backdoor attacks are more stealthy compared to the previous attacks. Chen et al. [12] first proposed to blend a specific image as a trigger with samples in a certain ratio to generate a backdoor sample. Barni et al. [13] on the other hand proposed the use of digital signals blended with samples to generate backdoor samples. To further increase the stealthiness of the backdoor attack, they proposed to directly select the samples of the target label for backdoor generation, known as clean-label attack. Recently, Li et al. [14] used image steganography to blend secret words as triggers on samples, making the triggers appear to be sample-specific on the image space.

B. BACKDOOR DEFENSE

The method in this paper includes both filtering backdoor samples and anti-backdoor training. In general, anti-backdoor training has two paradigms: backdoor-free security training and backdoor removal after training [15], [16], [17]. In this paper, we focus on the former defense paradigm. The core concern of this paradigm is how to distinguish backdoor samples and clean samples from the training set.

For filtering backdoor samples, Activation clustering (AC) [18] proposed to cluster the features of the input samples at deeper layers of the model for analysis. Spectral signatures (SS) [19] suggested a singular value decomposition of the covariance matrix of the samples' inputs characterized in the model in order to compute the outlier score for each sample. Recently, Huang et al. [20] proposed the use of cognitive distillation (CD) to extract cognitive patterns from samples and found that backdoor samples had much smaller cognitive patterns than clean samples.

The first method for anti-backdoor training in a poisoned dataset was proposed in ABL [7], and revealed the weaknesses of backdoor attacks for faster learning of backdoor samples and target-class dependency. After that, DBD [21] argued that the end-to-end supervised training paradigm led to models generating backdoors. Recently, Chen et al. [22] devised feature consistency towards transformations to measure the difference between backdoor samples and clean samples and performed secure learning based on this method.

In this paper, we focus on filtering backdoor samples from the poisoned training set and training backdoor-free models. Unlike the defense methods mentioned above, our method can both filter out all the backdoor samples in the training set and at the same time train backdoor-free models on this poisoned training set, which is not possible with any previous methods.

III. PROPOSED METHOD

A. PROBLEM FORMULATION

1) THREAT MODEL

In this paper, we assume that backdoor adversaries can only add backdoor samples to the training set without controlling the training process. Given a clean training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathcal{X} = \{0, 1, \ldots, 255\}^{C \times W \times H}$ is the sample i.e. image, $y_i \in \mathcal{Y} = \{1, \ldots, K\}$ is its label, and K is the number of sample classes. The backdoor adversaries make $\mathcal{D}$ contain both clean and backdoor samples

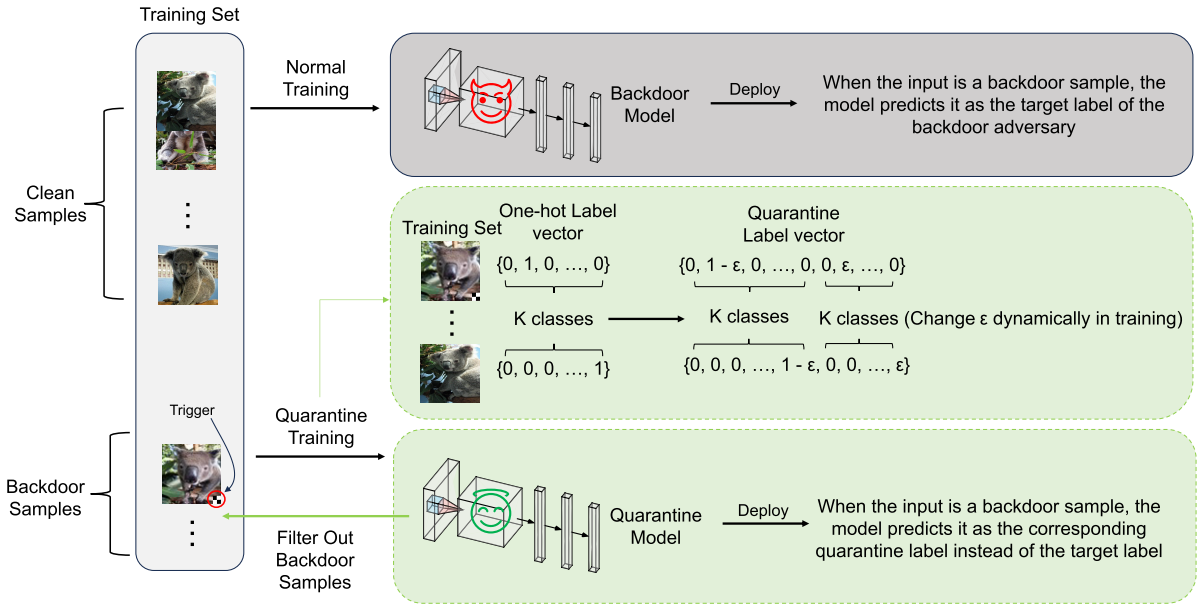


FIGURE 1. An illustration of our proposed method QT.

by modifying some of the samples in $\mathcal{D}$. The modified training set is defined as $\mathcal{D}_m = \mathcal{D}_c \cup \mathcal{D}_b$, where $\mathcal{D}_c$ is the clean training subset and $\mathcal{D}_b = \{(x', y') | x' = g(x), y' \in \mathcal{Y}, (x, y') \in \mathcal{D} \backslash \mathcal{D}_c\}$, x' is the backdoor sample and y' is the target class that backdoor adversaries want to control as predicted by the model. $g(\cdot)$ is the trigger add operation, for example, $g(\cdot)$ is defined in [23] as $g(x) = (1 - m) \cdot x + m \cdot \Delta$, where $m \in [0, 1]^{C \times W \times H}$ is the measure of overwriting the original image, $\Delta \in \mathcal{X}$ is the trigger pattern. Once $\mathcal{D}_m$ is ready, backdoor adversaries provide it to users to train models that contain backdoors.

2) DEFENDER'S GOALS

In this paper, we set that the defender does not know in advance the distribution of backdoor samples in the training set, and also does not have any reliable clean dataset. The goal of the defender is to determine $\forall x \in \mathcal{D}_m, x \in \mathcal{D}_c \vee x \in \mathcal{D}_b$. At the same time, the DNN model trained by the defender using this training set does not predict backdoor samples as target labels for backdoor adversaries and maintains a high accuracy for clean samples.

B. ONE POSSIBILITY FOR BACKDOOR SAMPLES

It is well known that backdoor models are capable of predicting any sample with a trigger as the target label of backdoor adversaries, regardless of the classes of the original samples. Intuitively, the backdoor features corresponding to backdoor triggers should be strong enough to displace the features of the original sample. We wonder if such a possibility exists: *$y' \notin \mathcal{Y}$ is a new class, $f_\theta(x') = y'$, where $f_\theta(\cdot)$ is the DNN model.*

To explore this possibility, we design an experiment. We first prepare the dataset $\mathcal{D}_m$ according to the setup of the backdoor adversaries, with the difference that the target label range of the backdoor samples is not within $\mathcal{Y}$, but is set to $K + 1$. Next we choose three representative backdoor attacks including BadNets [4], SIG [13] and ISSBA [14] to poison 10% of CIFAR-10 [24] training data. Then we train a PreActResNet-18 model [25] on the corresponding poisoned dataset using a standard training process. We plot the average training loss and prediction accuracy for samples from each class in Fig. 2.

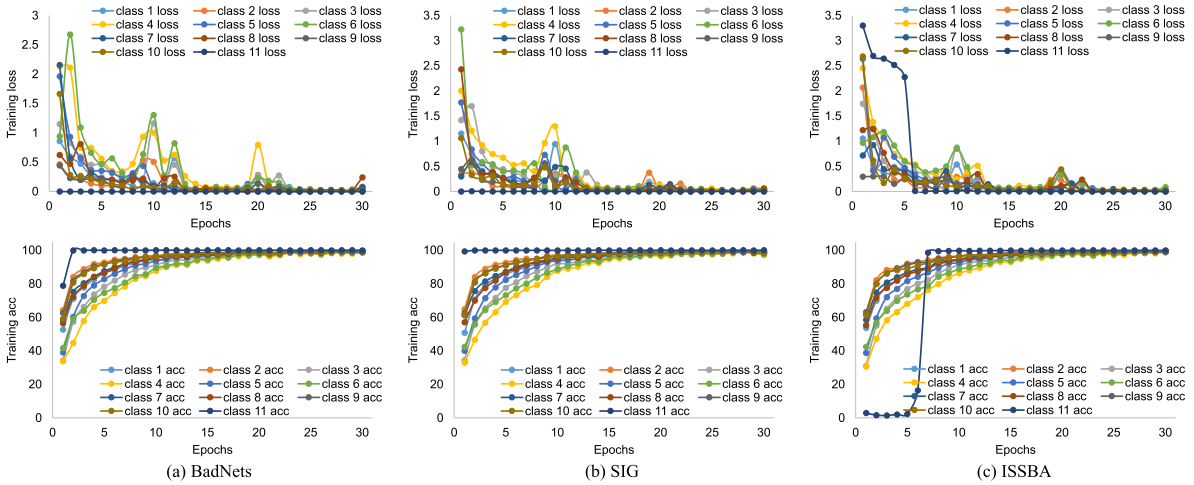
Experiments show that backdoor samples can be fitted by the model as a distinguishable class y'. It can also be observed that the backdoor samples, although they contain both the features of the original samples and the backdoor triggers, i.e., the backdoor features, do not affect the model's ability to classify them as other classes when they are treated as not belonging to $\mathcal{Y}$. Meanwhile, the backdoor class fits much faster in the model than the original class, which is also evident in [7].

Based on the above findings, backdoor samples can be easily filtered when taken as a separate class. Unfortunately, in practice, the backdoor class, i.e., the target label set by the backdoor adversaries, belongs to $\mathcal{Y}$. To enable the backdoor samples to be treated as a separate class, we take advantage of the strong fitting ability of DNN models to backdoor samples and propose quarantine learning as a method to solve this problem.

C. QUARANTINE TRAINING

According to the above discussion, in order to make the labels of backdoor samples decoupled from $\mathcal{Y}$, we propose a new defense method called quarantine training. Before formally presenting our method, we begin with the following definitions:

Let $p(y|x_i)$ be actual probability distribution of x_i over $\mathcal{Y}$, $\sum_{y=1}^{K} p(y|x_i) = 1$. The ouput of the DNN model for



(a) BadNets (b) SIG (c) ISSBA

FIGURE 2. Training loss and prediction accuracy of each class crafted under three backdoor attacks including BadNets [4], SIG [13], ISSBA [14]. The first row is the average training loss and the second row is the average prediction accuracy. Class 11 is the class corresponding to the backdoor samples we set up.

x_i is $q_\theta(y|x_i)$, θ are the parameters of the model. When θ converges, $q_\theta(y|x_i) \approx p(y|x_i)$.

In a classification task, $p(y|x_i)$ corresponding to x_i is usually represented by the one-hot label form of the label y_i

$$p(y|x_i) = \begin{cases} 1, & y = y_i, \\ 0, & \text{others}. \end{cases} \tag{1}$$

1) DYNAMIC QUARANTINE LABELS

As mentioned above, the backdoor samples' label $y_t \in \mathcal{Y}$. We first specify that for an original sample $x_i \in \mathcal{D}$, after it has been generated into the backdoor sample x_i', its target label $y_t \neq y_i$. Next, we consider x_i' as having both the features of the original sample and the backdoor features Δ, and let the probability distribution of Δ be $b(y'|\Delta)$, $\sum_{y'=1}^{K} b(y'|\Delta) = 1$. We denote the probability distribution of x_i' as

$$\begin{aligned} p'(y'|x_i') &= (1-\epsilon)p(y'|x_i) + \epsilon b(y'|\Delta) \\ &= \begin{cases} 1-\epsilon+\epsilon b(y'|\Delta), & y' = y_i, \\ \epsilon b(y'|\Delta), & y_i' = y_t, \\ 0, & \text{others}, \end{cases} \end{aligned} \tag{2}$$

where $\epsilon \in [0, 1]$ is the strength of the backdoor feature, in the settings of backdoor adversaries, ϵ is usually 1, and $b(y'|\Delta) = p(y'|x_i)$.

Next, we decouple y_t from $\mathcal{Y}$. Let $\mathcal{Y}' = \{1, \ldots, K, K+1, \ldots, 2K\}$, $y_{i+K} \in \mathcal{Y}_t = \{K+1, \ldots, 2K\}$ and $y_{i+K} = y_i + K$. Then the probability distribution of Δ is similarly considered to be in the form of a one-hot label, which is defined as $\sum_{y'=K+1}^{2K} b(y'|\Delta) = 1$. This allows us to redefine the probability distribution of x_i' as

$$p'(y|x_i') = \begin{cases} 1-\epsilon, & y = y_i, \\ \epsilon, & y = y_{i+K}, \\ 0, & \text{others}. \end{cases} \tag{3}$$

We refer to Equation (3) as quarantine label vector

$$(p'(y_{1+K}|x_i'), p'(y_{2+K}|x_i'), \ldots, p'(y_{2K}|x_i')) \tag{4}$$

The samples that are predicted to be in $\mathcal{Y}_t$ during the training process are called quarantine samples.

Since the defender cannot determine the backdoor samples in the training set, we treat all samples in the training set as if they were quarantine samples first and change their label vectors as quarantine label vectors. During quarantine training, we dynamically adjust the value of ϵ so that the model can distinguish backdoor samples mixed into the training set. It should be noted that dynamic adjustment here refers to two different dynamics: (1) dynamic differentiation between quarantine and non-quarantine samples. (2) dynamic increase in backdoor strength of quarantine samples.

2) TRAINING OF QUARANTINE MODEL

By defining quarantine labels, we can use them in training to get a quarantine model. We will explain the training process of the proposed QT method step by step.

First, Before training with quarantine labels, we recommend training the pre-quarantine model and filtering out the pre-quarantine samples. A more detailed discussion of both will follow in subsequent sections. The pre-quarantine model is trained as a standard model training using $\mathcal{D}_m$. After training, we use the samples with the lowest training loss as pre-quarantine samples.

Next, we expand the output of the last fully connected layer of the pre-quarantine model from K to $2K$ to accommodate the quarantine labels and modify the labels of all samples to quarantine labels. It is important to note that we are creating each class with its own quarantine class, rather than all classes having the same quarantine class. That is, for $y_i \in \mathcal{Y}$, its quarantine class is $y_{i+K} = y_i + K$.

After that comes the main part of the quarantine training. During quarantine training, the model fits a portion of the training samples into the original class space $\mathcal{Y}$ and also fits another portion of the training samples into the quarantine class space $\mathcal{Y}_t$. It is natural to do this by gradually increasing the ϵ of the quarantine labels of the samples that are

fitted into $\mathcal{Y}_t$ because they are more likely to be backdoor samples. In this process, the ϵ is dynamically adjusted so that the model gradually fits backdoor samples into $\mathcal{Y}_t$. Note that in quarantine training, we additionally adjust the quarantine labels of pre-quarantine samples. Since we default that pre-quarantine samples are most likely to be backdoor samples, their ϵ is always maximized. Also, for the quarantine labels of the samples that were fitted into $\mathcal{Y}$, their ϵ remains unchanged in this process, as this part of the samples may still contain backdoor samples.

During quarantine training, we specify the actual values of ϵ as [0.1, 0.9]. After a certain training epoch, we adjusted the ϵ of the samples to their minimum value. This modification was made because, at that point, these samples could be considered benign. At the same time, we do not make additional changes to the quarantine labels of pre-quarantine samples. It is important to note that benign samples may also be incorrectly fitted into $\mathcal{Y}_t$. Therefore, when too many samples are fitted into a particular quarantine class, we consider that the model is incorrectly treating the benign samples as backdoor samples. To prevent too many benign samples from being incorrectly fitted, we chose to directly set the ϵ of the quarantine labels of all samples fitted to that quarantine class to a minimum value. In this step, we judged the threshold as the number of samples fitted as y_{i+K} exceeding 90% of the number of samples in their source class y.

We describe the main algorithm of QT in Algorithm 1.

Algorithm 1 Training of Quarantine Model (QT)

Input: poisoned dataset$\mathcal{D}_m$, pre-quarantine model f_{θ_Q}, list of pre-quarantine samples l_p, backdoor strength ϵ_i, step β
Output: quarantine model $f_{\theta'_Q}$

for fully connected layer $l \in f_{\theta_Q}$ **do**
  replace l with a fully connected layer l' with output $2K$
end for
for $epoch \in [1, \ldots, n]$ **do**
  for $x_i \in \mathcal{D}_m$ **do**
    if $x_i \in l_p$ **then**
      $\epsilon_i = 0.9$
    end if
    $p'(y|x_i) \leftarrow$ Equation (3)
    $\theta'_Q \leftarrow \theta_Q - \eta \cdot \nabla \mathcal{L}_{\text{CE}}(p'(y|x_i), f_{\theta_Q}(x_i))$
    if $f_{\theta'_Q}(x_i) == y_i + K$ **then**
      $\epsilon_i \leftarrow \epsilon_i + \beta \cdot epoch$
    end if
  end for
end for

D. USES OF QUARANTINE MODEL

We find that the quarantine model obtained through quarantine training is able to filter out almost 100% of the backdoor samples in the training set, i.e., it can classify all backdoor samples into its quarantine class. In this paper, we introduce two uses of quarantine models: backdoor sample filtering and backdoor-free model training.

1) BACKDOOR SAMPLE FILTERING

Based on the above introduction, it is quite intuitive to understand how the quarantine model works for backdoor sample filtering. We diagnose all samples that are classified to quarantine class in the quarantine model as backdoor samples, i.e.,

$$x_i \in \mathcal{D}_b = \mathbb{I}(f_{\theta_Q}(x_i) = y_i + K). \tag{5}$$

2) BACKDOOR-FREE MODEL TRAINING

In effect, the quarantine model is already a backdoor-free model, i.e. backdoor samples cannot be predicted in the model as target labels for backdoor adversaries. It's just that the model has been trained for a relatively small number of epochs and is less accurate in predicting clean samples. Therefore, in subsequent training, the defender can fix all quarantine labels to the current state and adopt strategies such as data augmentation to strengthen the generalization ability of the model.

IV. EXPERIMENTS

A. EXPERIMENTAL CONFIGURATIONS

1) ATTACK CONFIGURATIONS

We consider 7 backdoor attacks in our experiments, including overwrite-based backdoor attacks: BadNets [4], Trojan [10], IAB [11] and blend-based backdoor attacks: Blend with Hello Kitty pattern, Blend with random pattern [12], SIG [13], ISSBA [14]. We use backdoor tool-box [26] for poisoned training sets generation, we generate its corresponding poisoned training set for each backdoor attack and train the backdoor model on these training sets separately. All attack configurations follow the settings in their original papers. We train PreActResNet-18 [25] as the base model on two classical datasets: CIFAR-10 and CIFAR-100 [24]. The dataset is split according to the program given in the open source code [26]. The poisoned rate is 10% on CIFAR-10 and 5% on CIFAR-100. We ignore SIG on CIFAR-100 because clean-label attack fails to achieve its goal on datasets with more classes. Each attack is trained for 100 epochs, random horizontal flipping and random cropping are used as the data enhancement strategy, the optimizer is SGD, the learning rate is 0.01, and the learning rate tuning strategy is cosine annealing.

2) DEFENSE CONFIGURATIONS

We compare our method QT with six state-of-the-art backdoor defense methods, including three training set backdoor sample filtering methods: AC [18], SS [19], CD-L [20] and three backdoor-free model training methods: ABL [7], DBD [21], D-ST [22]. Each defense is configured according to its original paper. It should be noted that in our experiments, τ in [22] fails to distinguish backdoor samples and clean samples. To make this method effective, we set $\tau =$ median blur.

For our QT, train the pre-quarantine model for 4 epochs, train the quarantine model for 20 epochs (dynamically adjust ϵ within the beginning 6 epochs) for backdoor sample

TABLE 1. The AUROC (%) of our QT and three baselines against 7 backdoor attacks on the training sets. The best results are in bold.

	Defense→ Attack↓	AC	SS	CD-L	Ours
CIFAR-10	BadNets	89.15	57.31	99.89	**100.00**
	Trojan	92.10	36.10	99.33	**99.99**
	Blend(HelloKitty)	83.20	88.04	96.27	**100.00**
	Blend(random)	92.85	62.58	97.99	**100.00**
	SIG	43.73	37.98	79.17	**100.00**
	IAB	84.75	96.04	99.18	**99.98**
	ISSBA	98.69	97.16	99.74	**100.00**
	Avg	83.50	67.89	95.94	**100.00**
	Defense→ Attack↓	AC	SS	CD-L	Ours
CIFAR-100	BadNets	81.51	76.16	64.29	**100.00**
	Trojan	89.39	26.39	88.07	**99.99**
	Blend(HelloKitty)	91.42	22.24	74.22	**99.93**
	Blend(random)	89.93	15.35	82.75	**99.99**
	IAB	81.36	59.25	93.44	**99.97**
	ISSBA	95.30	36.14	78.19	**99.99**
	Avg	88.15	39.26	80.16	**99.98**

filtering, and train the backdoor-free model for 100 epochs. The number of pre-quarantine samples is chosen to be 10% of the class with the highest number in the training set. Pre-quarantine model and quarantine model are trained without any data augmentation strategy, and the rest of the configurations are the same as in attack configurations. Initial $\epsilon = 0.4$ and step is 0.1.

3) EVALUATION METRICS

For the evaluation of backdoor sample filtering methods, we follow the choice of previous work [20] and use Area Under the Receiver Operating Characteristic (AUROC) [27] as an evaluation metric. The bigger the AUROC, the better. For the evaluation of the performance of backdoor-free models, we take two metrics which are also used in previous work [22], Attack Success Rate (ASR) and Accuracy Rate (ACC). The smaller the ASR, the better, and the larger the ACC, the better.

B. RESULTS

1) RESULTS ON BACKDOOR SAMPLE FILTERING

Table 1 summarizes the performance (AUROC) of our QT and the other three methods for filtering seven different backdoor samples on two baseline datasets. It can be seen that our QT is always able to almost completely filter out all backdoor samples, regardless of any backdoor attack case. Such stability also proves that the quarantine model can effectively distinguish backdoor samples.

CD-L is only slightly less effective than our QT on the CIFAR-10, but less effective than AC on the CIFAR-100. CD-L bases its sample filtering on the extracted cognitive patterns (CPs). We argue that when the image pixels are small and the classes are large, the L_1 norm of masks of CPs extracted from the benign samples may be small, causing CD-L to misclassify them as backdoor samples. In contrast, the performance of AC is more stable, proving that the backdoor samples do show anomalies in the deeper features of the model. SS is similarly a method that leverages sample features for analysis, however, it performs much less well due to its need for backdoor sample target labels, which are non-transparent to defender.

All in all, our QT always achieves the best performance, since we decouple the labels of the backdoor samples from the original classes. Our method achieves almost 100% average AUROC, an amazing result that no baseline method can achieve.

Meanwhile, we test the defense against ISSBA on a subset of ImageNet according to the approach in [14], and QT still achieves 100% AUROC, which demonstrates that the proposed QT method still has a good backdoor sample filtering effect for high-resolution as well.

2) RESULTS ON BACKDOOR-FREE MODEL TRAINING

Table 2 summarizes the performance (ACC&ASR) of our QT and the other three methods for backdoor-free model training on two baseline datasets. Benefiting from the near-perfect filtering of backdoor samples by the quarantine model, the ASR of the model trained by applying our method decreases to an incredible 0%. In the quarantine model, backdoor samples are no longer predicted to be the target class of backdoor adversaries but are quarantined into their quarantine class, thus defeating the goal of backdoor adversaries.

ABL and D-ST achieve good performance, but they both fail under some backdoor attacks. D-ST fails on ISSBA because ISSBA is difficult to be fitted in the early epochs of model training. D-ST performs the initial splitting of the training set by the feature consistency towards transformations (FCT), however, it is not possible to filter the ISSBA backdoor samples out of the training set by FCT. Therefore, D-ST continues to treat the backdoor samples as benign samples and learn them in the later security training. DBD even fails to effectively defend against any backdoor attacks on CIFAR-100, which we speculate is because its symmetric cross-entropy is not suitable for this dataset.

In addition, as in the previous subsection, we also test the defense against ISSBA attacks on a subset of ImageNet. The experiments show that without reducing the ACC of the model, the proposed QT method is still effective in reducing the ASR to 0. This again demonstrates that the proposed QT method is able to handle high resolution effectively.

It is also worth noting that the other three baselines each have distinct negative effects on ACC. In contrast, our method resulted in an average increase of 0.21% on ACC for CIFAR-10 and only an average decrease of 0.01% for CIFAR-100. We hypothesize that this is due to two reasons: First When training with a training set that contains backdoor samples, the backdoor samples themselves lead to a decrease in the model's prediction accuracy on benign samples. QT separates backdoor and benign samples from the label space during the training process, which reduces the disturbance of backdoor samples on the prediction accuracy of benign samples. Second, QT labels may make the model

TABLE 2. The ACC(%) and ASR(%) of our QT and three baselines against 7 backdoor attacks. The best results are in **bold**.

	Defense→	None		ABL		DBD		D-ST		Ours	
	Attack↓	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	BadNets	86.80	100.00	84.76	1.52	84.26	0.84	85.21	0.12	**86.80**	**0.00**
	Trojan	86.58	100.00	84.22	3.72	82.71	0.27	86.21	0.03	**86.70**	**0.00**
	Blend(HelloKitty)	86.25	99.91	73.65	15.37	83.63	0.21	86.01	0.20	**86.36**	**0.00**
	Blend(random)	86.53	100.00	80.24	0.94	83.73	0.73	86.60	0.08	**86.88**	**0.00**
	SIG	87.32	95.72	61.60	2.99	82.02	0.05	86.58	0.00	**87.66**	**0.00**
	IAB	87.05	100.00	48.43	98.23	83.91	100.00	83.95	8.63	**87.93**	**0.00**
	ISSBA	87.23	100.00	75.36	0.20	84.21	100.00	76.17	100.00	**86.91**	**0.00**
	Avg	86.82	99.38	72.61	17.57	83.50	28.87	84.39	15.58	**87.03**	**0.00**
	Defense→	None		ABL		DBD		D-ST		Ours	
	Attack↓	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-100	BadNets	69.50	99.83	65.63	0.04	67.12	99.99	66.01	0.15	**69.49**	**0.06**
	Trojan	70.76	99.74	64.76	0.00	67.34	100.00	65.20	0.08	**71.12**	**0.02**
	Blend(HelloKitty)	70.63	99.17	46.29	**0.00**	68.55	99.99	62.97	**0.00**	**70.64**	**0.00**
	Blend(random)	70.72	99.20	43.53	**0.00**	68.65	99.99	62.78	**0.00**	**69.99**	**0.00**
	IAB	70.03	99.99	40.12	99.60	69.30	100.00	65.44	**0.00**	**70.18**	0.02
	ISSBA	70.12	99.56	41.32	99.99	68.89	100.00	62.03	100.00	**70.28**	**0.00**
	Avg	70.29	99.58	50.28	33.27	68.31	100.00	64.07	16.71	**70.28**	**0.02**

less confident in learning about the samples, which leads to an improved generalization ability of the model.

C. COMPUTATIONAL COST

Our method QT not only outperforms other methods in terms of effectiveness in backdoor sample filtering but also has a smaller computational cost. Unlike other methods (AC, SS, CD-L) that require the model to be fully trained to achieve the best results, our method QT only requires a few epochs of training to filter backdoor samples. This is because our method QT uses the quarantine model to filter the training set. If a sample is classified into $\mathcal{Y}_t$ by the model, it is considered a backdoor sample (Equation (5)); otherwise, it is considered a benign sample. Even if the quarantine model has a lower accuracy in classifying benign samples, it will only predict benign samples in the source space $\mathcal{Y}$.

In addition, our method only requires one forward pass on the training set to obtain results. In contrast, AC needs to perform a forward pass to obtain the activation features for each sample, and then conduct clustering operations on these features to obtain the final results. Similarly, SS also requires an initial forward pass, followed by operations such as singular value decomposition and analysis of the activation features to obtain the final results. CD-L is the most time-consuming because it involves solving an optimization problem to obtain the cognitive patterns for each sample, requiring repeated forward and backward passes. CD-L analyzes the sizes of masks of cognitive patterns for all samples in the training set, and samples with sizes below a specified threshold are considered as backdoor samples.

It should be noted that the computational cost of adjusting the quarantine labels of the samples during training is almost negligible. Therefore, the computational cost of the proposed QT method for model training is comparable to that of standard training.

D. ABLATION STUDIES

This subsection will discuss the effectiveness of our method in terms of

- Whether the pre-quarantine model as well as the pre-quarantine samples are necessary?
- Whether the dynamic quarantine labels are necessary?
- What are the effects of different poisoned rates on our method?

In order to better characterize the effectiveness of our method under different configurations, two additional metrics are introduced as evaluation criteria in this subsection, namely the clean-sample quarantine rate (CQR) and the backdoor-sample quarantine rate (BQR). They are defined as the ratio of samples being predicted to quarantine class in quarantine training, where

$$\mathrm{CQR}=\frac{\sum_{i=1}^{N}\mathbb{I}(x_i=\text{Clean Sample})\cdot\mathbb{I}(f_{\theta_Q}(x_i)=y_i+K)}{\sum_{i=1}^{N}\mathbb{I}(x_i=\text{Clean Sample})},$$

$$\mathrm{BQR}=\frac{\sum_{i=1}^{N}\mathbb{I}(x_i=\text{Backdoor Sample})\cdot\mathbb{I}(f_{\theta_Q}(x_i)=y_i+K)}{\sum_{i=1}^{N}\mathbb{I}(x_i=\text{Backdoor Sample})},$$

These two metrics determine the performance of the quarantine model, where the lower the CQR, the higher the ACC of the trained model; the higher the BQR, the lower the ASR of the model. It should be noted that these two metrics are not visible in practice and are used in this subsection only to characterize the method's effectiveness.

1) NECESSITY OF PRE-QUARANTINE

The pre-quarantine model can be used as a feature extractor for subsequent training of quarantine model, which has the advantage that the pre-quarantine model already has the ability to extract the original features and the backdoor features of samples, which can effectively improve the efficiency of quarantine classification of backdoor samples in

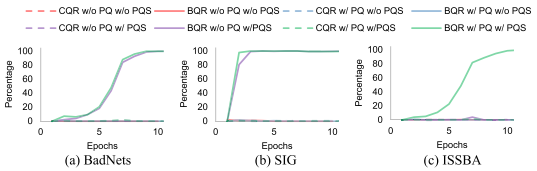


FIGURE 3. CQR and BQR crafted with four different configurations.

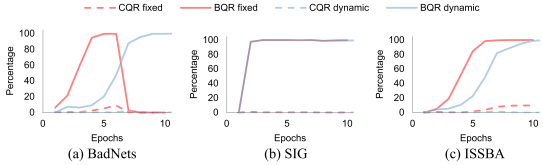


FIGURE 4. CQR and BQR crafted under two different settings for ϵ.

the training of quarantine model. Second, the pre-quarantine samples filtered out by the pre-quarantine model help the quarantine model to better distinguish backdoor samples during training. To better illustrate the necessity for both, we design experiments with four different configurations and craft the results as Fig. 3. (1) No use of pre-quarantine model or pre-quarantine samples. (2) Use only pre-quarantine model. (3) Use only pre-quarantine samples. (4) Use pre-quarantine model and pre-quarantine samples. Three representative backdoor attacks [4], [13], [14] are chosen as examples for this experiment, and the other experimental configurations are the same as in the previous subsection. We abbreviate the pre-quarantine model as PQ and the pre-quarantine samples as PQS.

It can be observed that in configuration (1), our method is unable to distinguish the backdoor samples; in configuration (2) it is also ineffective; in configuration (3), our method is able to distinguish some of the backdoor samples, which proves that the pre-quarantine samples help the quarantine model to quarantine the backdoor samples correctly to a certain extent. However, this configuration fails under ISSBA attack, which is due to the slow fitting of ISSBA attack in training, and the backdoor features cannot be effectively extracted in the early training of the quarantine model; in configuration (4), our method is able to quarantine most of the backdoor samples into its quarantine class (more than 50% of the total number of backdoor samples) before fixing ϵ, and thus quarantine all of the backdoor samples in subsequent quarantine training. The experiments show that given the generalizability of QT, neither the pre-quarantine model nor the pre-quarantine samples should be missing.

2) NECESSITY OF DYNAMIC QUARANTINE LABELS

Dynamic quarantine labels facilitate the classification of backdoor samples during quarantine training. In this subsection, we fix ϵ for quarantine samples to 0.9, leave other settings (e.g., ϵ for non-quarantine samples, etc.) unchanged and perform experiments. The results are shown in Fig. 4.

When fixing ϵ, our method remains effective only on SIG attacks. This is because the number of backdoor samples for clean-label attacks is small, and almost all backdoor samples for SIG attacks are already included when filtering

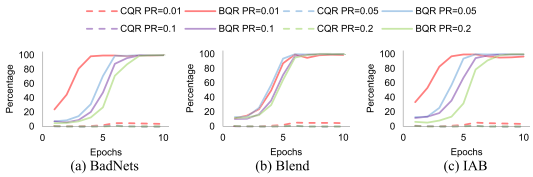


FIGURE 5. CQR and BQR crafted with different poisoned rates.

the pre-quarantine samples. In other attacks, while fixed quarantine labels can at first more effectively lead the quarantine model to predict backdoor samples as quarantine classes, however, more clean samples are likewise wrongly treated as quarantine samples, and their number approaches the full number of their class. According to our relabeling principle for samples, all samples in this quarantine class are re-considered as non-quarantine samples, which makes the quarantine model fail to filter out backdoor samples. As a result, while fixed ϵ can have a high BQR at the beginning, it can cause problems in classifying clean samples, as opposed to dynamic quarantine labels that can better balance the classification of clean samples and backdoor samples.

3) DIFFERENT POISONED RATE

Reducing the poisoned rate in the training set can make backdoor attacks more stealthy. we consider more challenging settings with varying poisoned rates [1%, 5%, 10%, 20%]. Since SIG attack and ISSBA attack cannot produce an effective backdoor attack effect with a poisoned rate setting of less than 10%, they are replaced by Blend(HelloKitty) attack and IAB attack in this experiment. The results are shown in Fig. 5. PR is short for Poisoned Rate.

Experiments show that our method is robust enough to filter out backdoor samples effectively regardless of the poisoned rate within the training set. When the poisoned rate is too low, the pre-quarantine model training inevitably incorrectly classifies more clean samples into the quarantine class. This is caused by the fact that when the number of backdoor samples is low, there are more clean samples contained within the pre-quarantine samples. However, as can be seen in the figure, the number of clean samples that are incorrectly classified gradually decreases during the later training of the quarantine model.

V. LIMITATIONS

As can be obtained from the ablation experiments, the proposed QT method relies on pre-quarantine samples, which are small portions of samples with the lowest training loss. That is, when there exists some kind of backdoor samples whose training loss is slightly larger than the training loss of the benign samples, the proposed QT method will not be able to defend against them. How to free QT from its dependence on pre-trained samples is one of the focuses of our future work.

In addition, the proposed QT method requires the defender to have full control over the training process of the model

and also evaluates the effectiveness of the defense only on the image classification model. How to extend QT to more scenarios [28], [29] is also one of our future work.

VI. CONCLUSION

In this paper, we propose a new method for backdoor defense called quarantine training (QT). Using our method QT, we successfully decouple backdoor samples from target labels. Our method QT leads to a quarantine model for filtering backdoor samples in the training set. In addition, we present the performance of the quarantine model as a backdoor-free model. We have also demonstrated the effectiveness of our method on backdoor sample filtering and backdoor-free learning through different experiments. Experience has shown that our method not only comprehensively outperforms other methods, but also does not affect clean samples.

REFERENCES

[1] L.-H. Wen and K.-H. Jo, "Deep learning-based perception systems for autonomous driving: A comprehensive survey," *Neurocomputing*, vol. 489, pp. 255–270, Jun. 2022.

[2] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3239–3259, Jun. 2022.

[3] H. Du, H. Shi, D. Zeng, X.-P. Zhang, and T. Mei, "The elements of end-to-end deep face recognition: A survey of recent advances," *ACM Comput. Surv.*, vol. 54, no. 10, pp. 1–42, Jan. 2022.

[4] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[5] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–18, 2022.

[6] Y. Zeng, M. Pan, H. Jahagirdar, M. Jin, L. Lyu, and R. Jia, "Meta-Sift: How to sift out a clean subset in the presence of data poisoning?" in *Proc. 32nd USENIX Secur. Symp.*, 2023, pp. 1667–1684.

[7] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: Training clean models on poisoned data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 14900–14912.

[8] A. Nguyen and A. Tran, "WaNet—Imperceptible warping-based backdoor attack," 2021, *arXiv:2102.10369*.

[9] Z. Wang, J. Zhai, and S. Ma, "BppAttack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15054–15063.

[10] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Proc. 25th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–16.

[11] T. A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3454–3464.

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

[13] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in CNNS by training set corruption without label poisoning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 101–105.

[14] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16443–16452.

[15] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proc. Int. Symp. Res. Attacks, Intrusions, Defenses*. Cham, Switzerland: Springer, 2018, pp. 273–294.

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

[17] R. Zheng, R. Tang, J. Li, and L. Liu, "Data-free backdoor removal based on channel lipschitzness," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 175–191.

[18] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," 2018, *arXiv:1811.03728*.

[19] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[20] H. Huang, X. Ma, S. Erfani, and J. Bailey, "Distilling cognitive backdoor patterns within an image," 2023, *arXiv:2301.10908*.

[21] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, "Backdoor defense via decoupling the training process," 2022, *arXiv:2202.03423*.

[22] W. Chen, B. Wu, and H. Wang, "Effective backdoor defense by exploiting sensitivity of poisoned samples," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 9727–9737.

[23] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 707–723.

[24] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Handbook Systemic Autoimmune Diseases*, vol. 1, no. 4, 2009.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision–ECCV*. Amsterdam, The Netherlands: Springer, 2016, pp. 630–645.

[26] *VTU81 and Gnipping*. Accessed: 2021. [Online]. Available: https://github.com/vtu81/backdoor-toolbox

[27] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[28] A. Agarwal, R. Singh, M. Vatsa, and N. Ratha, "IBAttack: Being cautious about data labels," *IEEE Trans. Artif. Intell.*, vol. 4, no. 6, pp. 1484–1493, Dec. 2023.

[29] Y. Yu, Y. Wang, W. Yang, S. Lu, Y.-P. Tan, and A. C. Kot, "Backdoor attacks against deep image compression via adaptive frequency trigger," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12250–12259.

CHENGXU YU received the bachelor's degree from the College of Water Conservancy and Hydropower Engineering, Sichuan Agricultural University, in 2020. He is currently pursuing the Graduate degree in applied statistics with the Zhejiang University of Science and Technology. His research interests include deep learning and trustworthy AI.

YULAI ZHANG received the M.S. degree from the School of Software, Tsinghua University, Beijing, China, in 2011, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, in 2015. He is currently with the School of Information Technology and Electronics Engineering, Zhejiang University of Science and Technology, Hangzhou, China. His research interests include model-based software engineering, machine learning, and system identification.