Anti-Backdoor Model: A Novel Algorithm to Remove Backdoors in a Non-Invasive Way

Chen Chen, Haibo Hong, Tao Xiang, *Senior Member, IEEE*, and Mande Xie

***Abstract*— Recent research findings suggest that machine learning models are highly susceptible to backdoor poisoning attacks. Backdoor poisoning attacks can be easily executed and achieve high success rates, as the model exhibits anomalous behavior even if a small quantity of malicious data is incorporated into the training dataset. In conventional backdoor defense technologies, fine-tuning is employed as an invasive method that involves adjusting the parameters of model neurons to eliminate backdoors in the attacked model. Nevertheless, this method poses a challenge as the same neurons are responsible for both the original and backdoor tasks, resulting in a decline in the accuracy of the original task during the fine-tuning process. In order to address this issue, we propose a non-invasive approach known as Anti-Backdoor Model (ABM), which does not involve modifying the parameters of the attacked model. ABM employs an external model to counteract the influence of the backdoor task on the attacked model, thereby achieving a balance between eliminating backdoors and preserving the accuracy of the original task. Specifically, our approach involves initially embedding a controllable backdoor in the dataset and leveraging the strong and weak relationships between backdoors to identify a highly concentrated poisoned dataset. Subsequently, we employ the standard training method to train the attacked model (the teacher model). Finally, we utilize this dataset with low volume to train an external model (the student model) that exclusively focuses on backdoors by means of knowledge distillation to counteract the backdoor task in the attacked model (the teacher model). In the experimental part, we assess the effectiveness of ABM by testing eight mainstream attacks on three standard public datasets. Experimental results reveal that ABM exhibits promising efficacy in eliminating the backdoor task while preserving the accuracy of the original task. Our source codes are open at https://gitee.com/dugu1076/ABM.git.**

***Index Terms*— Backdoor poisoning attacks, anti-backdoor model, non-invasive, knowledge distillation, low-volume dataset.**

I. Introduction

AT PRESENT, neural networks are increasingly utilized in diverse domains, including image classification [1], [2], [3] and natural language processing [4], [5]. However, these pervasive deep learning systems are susceptible to a range of security vulnerabilities, including evasion attacks [6], [7], [8], model stealing attacks [9], [10], membership inference attacks [11], [12], [13], backdoor attacks [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], etc. Malicious attackers can exploit these vulnerabilities to illicitly obtain information and potentially manipulate system outcomes, leading to significant and potentially incalculable consequences. In recent years, backdoor attacks have developed rapidly, and early backdoor poisoning attacks have derived new types of backdoor attacks such as dynamic sample-specific backdoor attacks [16], [17], [18], multi-targeted backdoor attacks [19], [20], adaptive attacks with mitigating the latent separability [21], and backdoor attacks on video recognition models [22], etc. In this article, we focus on the classic backdoor poisoning attack. In contrast to conventional data poisoning [23], [24], [25], the backdoor poisoning attack does not compromise the accuracy of the original task, but introduces hidden triggers that activate only under specific circumstances within the attacked model. The implementation of backdoor poisoning conditions is straightforward, as a basic attack can be executed by introducing contaminated data patches into a subset of the training dataset and adjusting the corresponding labels [26]. The attacked model exhibits consistent behavior with the standard model when exposed to benign input, but deviates significantly when presented with non-benign input containing triggers. Furthermore, the increased complexity of deep neural network models necessitates a substantial volume of data for high-precision training. Consequently, many trainers resort to utilizing crawlers or third-party data sources to acquire training datasets, inadvertently creating vulnerabilities for potential backdoor poisoning attacks.

Regrettably, defending against the backdoor poisoning attack poses a formidable challenge. One factor is the ease of implementation of attack conditions. As illustrated in Fig. 1, a mere 1% contamination of the dataset results in significantly high success rates for the attackers. Compounded by the lack of foreknowledge of poisoned data, the defenders face challenges in identifying commonalities in the dataset through manual inspection. Furthermore, the evolution of backdoor triggers from conventional patterns [26] like small squares to unconventional patterns [27], [28], [29], [30] like imperceptible noises further complicates the auditing process. The other factor contributing to the challenges posed by neural networks is their lack of interpretability. The neural network

Manuscript received 21 April 2023; revised 10 November 2023, 14 March 2024, and 24 May 2024; accepted 17 July 2024. Date of publication 1 August 2024; date of current version 8 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61602408 and Grant 61972352 and in part by the Key Research and Development Program of Zhejiang Province under Grant 2024C01025. The associate editor coordinating the review of this article and approving it for publication was Prof. Haijun Zhang. *(Corresponding authors: Haibo Hong; Mande Xie.)*

Chen Chen and Haibo Hong are with the School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou 310018, China (e-mail: honghaibo1985@163.com).

Tao Xiang is with the School of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: txiang@cqu.edu.cn).

Mande Xie is with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China (e-mail: xiemd@zjgsu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2024.3436508

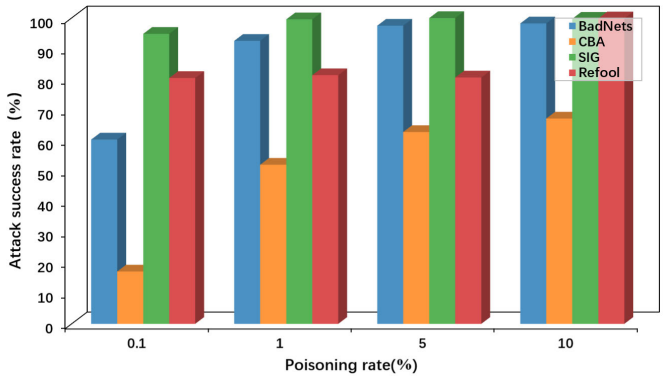

Fig. 1. The attack success rate (ASR%) of backdoor attacks with different poisoning rate on CIFAR-10. It can be found that even if the poisoning rate is 1%, all four attacks achieve the attack success rate of more than 50%, and even if the poisoning rate is 0.1%, there are still three attacks can maintain the attack success rate of more than 50%.

model relies on a complicated interplay of linear and nonlinear transformations between matrices, for which there are no clear theoretical explanations. Consequently, anomalies cannot be easily identified through examination of the model's internal parameters.

Among the prevailing backdoor defense strategies, fine-tuning [35], [36], [37], [38] stands out as a post-training invasive method that involves adjusting the neurons to eliminate backdoor tasks in the attacked model. The experimental findings suggest that this method is effective. However, a persistent challenge exists in effectively addressing the relationship between the backdoor task and the original task. Excessive focus on repressing the backdoor task in the repair process frequently results in a decrease in the accuracy of the original task, whereas prioritizing the accuracy of the original task will retain the backdoor task. We believe that the neurons engaged in both the original task and the backdoor task within the attacked model results in an invasive fine-tuning repair process that impacts both tasks simultaneously, thereby creating the hindrance.

To address this issue, we introduce Anti-Backdoor Model (ABM) as a non-invasive approach for repairing the attacked model. The contributions of this paper are outlined as follows:

- ABM is proposed as a non-invasive, straightforward, and effective approach for repairing backdoors in the attacked model. As the first non-invasive defense method in pre-training phase, ABM does not make any modifications to the internal neurons and parameters of the attacked model (the teacher model), and only adopt an external model (the student model) to counteract the backdoor task in the attacked model (the teacher model). Consequently, ABM successfully repairs backdoors while minimizing the loss of accuracy on the original task.
- ABM is implemented on three classic datasets (CIFAR-10, GTSRB, and ImageNet Subset), and evaluated against eight prevalent backdoor poisoning attacks (five dirty label attacks and three clean label attacks). The experimental results demonstrate the effectiveness of ABM in successfully defending against almost all backdoor poisoning attacks without significant loss of accuracy. Furthermore, comprehensive tests including time cost analysis, stress testing, and adaptive attack evaluation have been conducted to assess the robustness of ABM, revealing its efficacy as a defense mechanism against backdoor poisoning attacks.

II. Related Work

This section mainly introduces several backdoor poisoning attacks and backdoor defense methods. In addition, we introduce knowledge distillation used in ABM.

A. Backdoor Poisoning Attacks

Current backdoor poisoning attacks are mainly divided into two categories: 1) dirty label attacks; 2) clean label attacks. The initial dirty label attacks [26], [27], [30] primarily focused on altering labels and incorporating triggers, such as single pixels, squares or more intricate patterns. However, these simplistic attacks are frequently detected through manual examination. To enhance the stealth of the backdoor poisoning attack, the attacker strategically refines triggers and integrates them into clean data through natural means, such as invisible noises and mixed modes [18], [31]. In contrast to dirty label attacks, clean label attacks [28], [29], [32] are focused on optimizing labels in order to circumvent manual verification processes, thereby achieving attack objectives without altering the labels. These attacks are often able to evade numerous defense mechanisms due to their less overtly aggressive nature.

In order to demonstrate the performance of ABM, this paper adopts five classic dirty label attacks (BadNets [26], Blend [27], CBA [30], BPP [31], ISSBA [18]) and three classic clean label attacks (SIG [28], Refool [29] and Nar [32]).

B. Backdoor Defenses

Previous backdoor defense methods primarily focused on fine-tuning [35], [36], denoising, and fine-pruning which have been proven to be ineffective against existing backdoor attacks such as [29], [34], and [33]. In response to these attack methods, more advanced defense methods [37], [38], [39], [40], [41], [42], [43], [44] have emerged. NAD [37] utilized a limited number of clean datasets and implemented knowledge distillation to fine-tune the attention mechanism of the teacher model, resulting in the teacher model disregarding the trigger position and effectively eliminating the backdoor. ABL [38] identified a subset of backdoor datasets by analyzing the varying reduction rates of clean and backdoor data loss values, and conducted anti-backdoor learning to eliminate the backdoor. DBD [39] revealed that the learning of hidden backdoors mainly comes from the end-to-end supervised training paradigm, and proposed a simple and effective backdoor repression training method based on decoupling. NONE [40] designed a novel training method that forces the training to avoid generating hyper-planes formed by backdoor-related neurons, so as to remove the backdoor. DST and DBR [41] adopted a sample-distinguishment module utilizing the FCT metric to removes the backdoor from a backdoored model. CBD [42] relied on causality-inspired backdoor defense to

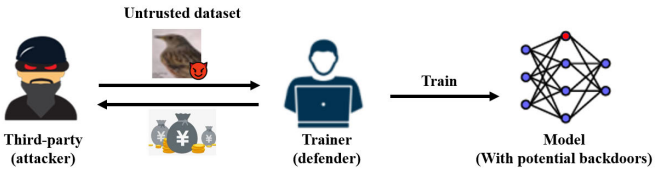

Fig. 2. The trainer purchases a dataset from an untrusted third-party. The malicious third-party may poison the dataset, and the trainer utilizes this dataset to train a model with potential backdoors.

learn deconfounded representations for reliable classification. ASD [43] is an effective defense method for backdoor poisoning attacks based on adaptive split data sets, relying on loss-guided segmentation and meta-learning inspired segmentation to dynamically update clean and contaminated data pools. BAB [44] completely relied on the strength relationship between backdoors, and adopted multiple filters to filter out suspicious data from the poisoned dataset.

Among these defense methods, fine-tuning [35], [36], [37], [38] is utilized as an invasive approach that depends on the adjustment of internal neurons within the attacked model to repair backdoors. However, this method is not flawless as the backdoor task and the original task are not entirely distinct within the neurons, resulting in the original task being compromised upon removal of the backdoor task.

In this paper, we propose Anti-Backdoor Model (ABM) as a non-invasive improvement to traditional fine-tuning methods. Specifically, we take the attacked model as the teacher model, and train a student model as the external model to counteract the effects of the backdoor in the teacher model.

C. Knowledge Distillation

Knowledge distillation is a technique designed to transfer latent knowledge from complicated, high-performing models to simpler models, with the goal of enabling the simpler models to achieve comparable or superior capabilities and performance to their more complicated counterparts [45]. The concept of knowledge distillation was initially introduced by Hinton [46], who also introduced the concept of temperature T to adjust the prediction probabilities, thereby facilitating enhanced learning by the student model from the teacher model. In essence, knowledge distillation can be viewed as a form of knowledge extraction. The student model continuously extracts useful knowledge from the teacher model.

III. Problem Statement

A. Defense Setting

We consider a common scenario in defense settings where the defender acquires a dataset from a potentially malicious third-party. For better understanding, we take an example in Fig. 2. The attacker has the ability to contaminate the dataset to varying degrees and through various means, while lacking control over the model training process. Despite possessing the dataset, the defender lacks prior knowledge of the attacker's strategies. The goal of ABM is to obtain a clean model with the untrusted dataset.

B. Assumption

1) Eliminating the Backdoor Task: Let's take the classification task as an example and consider the reasons for the formation of the attacked model. The poisoner inputs the backdoor data D_b in the training dataset D_c, the trainer adopts the standard model training to train the model on the dataset $D = D_c \cup D_b$. The model continuously fits the dataset D during the training process, thereby obtaining the mapping relationship f when the model training is stable. f not only satisfies the trainer's classification task for the clean dataset D_c, but also is compatible with the backdoor data D_b without the trainer's knowledge. That means f is a composite mapping relationship of f_c and f_b. Here, f_c represents the classification task for the clean dataset D_c, and f_b represents the classification task for the backdoor dataset D_b.

The trainer attempts to eliminate the mapping relationship f_b and ensure that f_c is intact while removing f_b. An intuitive idea is that a student model M_s with only mapping relationship f_b is used to repress the activation of f_b in the teacher model M_t, it can ensure that the teacher model M_t eventually becomes the clean model, as shown in Equation 1.

$$Output = M_t(x) - M_s(x), \tag{1}$$

where $M_t(x)$ and $M_s(x)$ are the output before softmax, M_t is the attacked model generated by the poisoned dataset, M_s is the external model that is only sensitive to the backdoors and not activated on the clean dataset.

So how to generate a student model based on the backdoor task? A straightforward and intuitive approach involves training the model using only backdoor data, but this method is overly idealistic. As discussed in **Defense Setting**, the training dataset may contain poisoned data, so we can only filter the poisoned data from the training dataset. Experimental results demonstrate that the backdoor task is easily learned and highly activated even in a small dataset due to its single trigger characteristic. However, as a result of substantial alterations in the position and depth of the original task, a considerable amount of datasets are typically necessary to achieve favorable outcomes, as illustrated in Fig. 3. Therefore, the training of the student model exclusively focused on the backdoor task can be accomplished by isolating a small dataset containing a high concentration of backdoor data from the training dataset.

2) Filtering the Backdoor Dataset: We propose a Non-invasive Backdoor Against Backdoor (NBAB) algorithm that leverages the integration of established backdoors to filter the aforementioned high-concentration backdoor dataset. Our methodology is predicated on the premise that the presence of two non-interfering backdoors in the dataset necessitates a strong and weak relationship between the backdoors.

In this context, we introduce a controllable backdoor into the training dataset D during the pre-training phase, with the objective of targeting a brand new class to avoid overlap with the original poisoning attack target. Afterwards, the trigger is incorporated into the dataset D after the training is completed. If the predicted label generated by the verification model does not match our predetermined label embedded in the backdoor, the related data will be included in the isolated dataset used for training the student model M_s. In order to

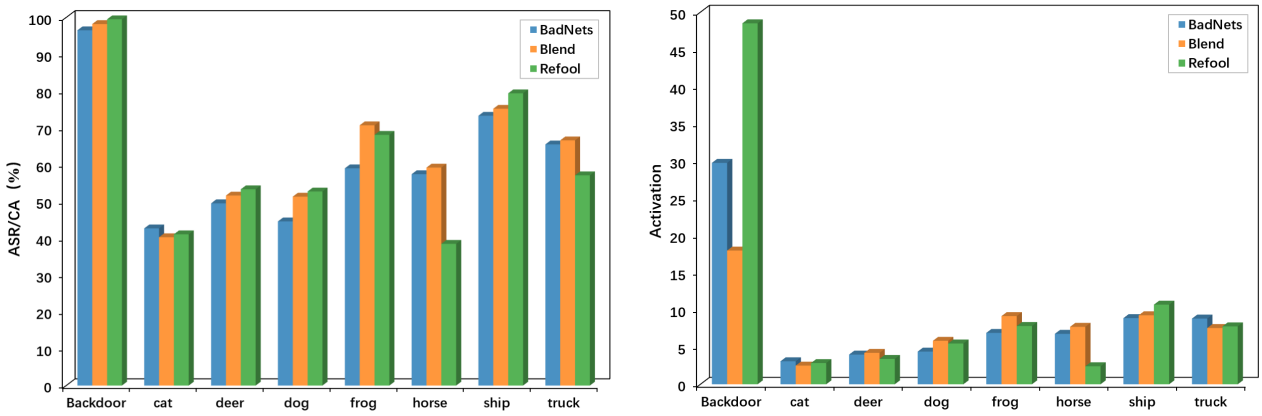

Fig. 3. The training results of the dataset with 1% poisoning rate (7 classes) on CIFAR-10. Left: Attack success rate (ASR)/Clean accuracy (CA); Right: Average activation. It can be found that even with a 1% poisoning rate (7 classes), the backdoor can still achieve approximately 100% attack success rate, while the clean class is less effective, and the activation of the backdoor class is much higher than that of the clean class.

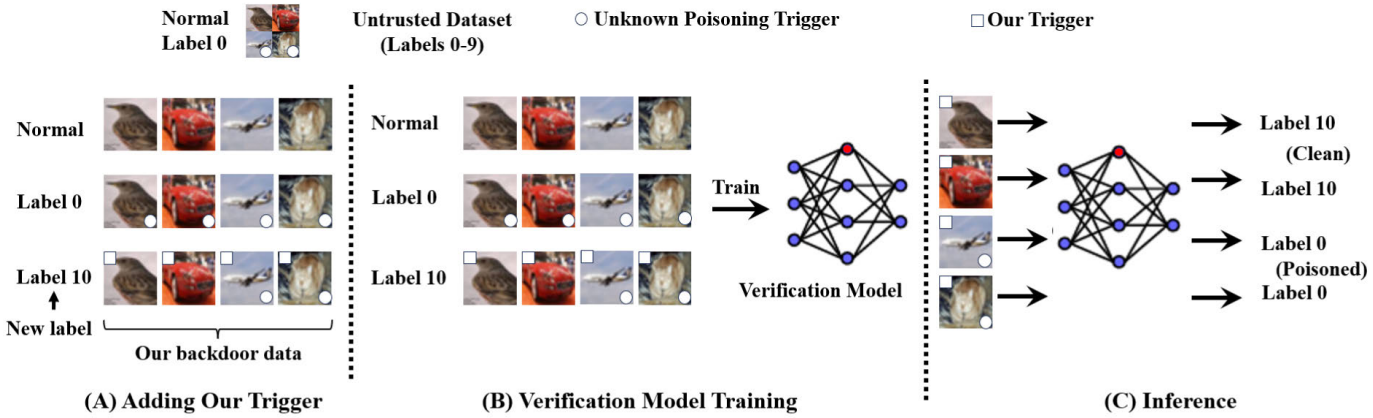

Fig. 4. An example of NBAB algorithm. Taking CIFAR-10 as an example. (A) Adding Our Trigger: Selects some data, adds our trigger (a 2*2 small box) in the upper left corner of the data, and modifies it to a brand new class (Label 10). (B) Verification Model Training: Trains a verification model in standard model training. (C) Inference: Add our trigger in the same position in all the training dataset, and input the dataset into the verification model. If the model points to new label (Label 10), the data is clean; otherwise, adds it to the isolated data.

better understand NBAB, we provide an example as shown in Fig. 4. Specifically, when adding the controllable backdoor, we firstly extend original 10 classes (Labels 0-9) to 11 classes (Labels 0-10) within the fully connected layer, and modify the label of the images with our controllable trigger to Label 10. Given that the original information of the image, excluding the trigger, falls within one of original 10 classes (Labels 0-9), the verification model will identify our trigger as the maximum feature of Label 10. Whenever the trigger appears, the image with our trigger will be predicted as Label 10.

IV. Method

In this section, we formally introduce the workflow of ABM. The main idea of ABM involves exploiting both strong and weak relationships between backdoors to implant a weak backdoor in the dataset. This process relies on the output to generate an isolated dataset with low data volume and high poisoning rate. Subsequently, this isolated dataset is utilized to train a student model that is only sensitive to the backdoor task to counteract the backdoor task in the teacher model.

Specifically, the teacher model represents the attacked model requiring the elimination of the backdoor task, and the student model functions as an external model solely responsive to the backdoor task. The student model utilizes a limited dataset to transfer the backdoor task from the teacher model to itself. In the generation phase of ABM, the output of the teacher model is subtracted from the output of the student model prior to the softmax function, effectively neutralizing the backdoor task in the teacher model, see Equation 1. Specifically, ABM is divided into three steps: 1. Filtering of isolated data; 2. Teacher model training; 3. Student model training. A toy example of ABM is displayed in Fig. 5.

A. Filtering of Isolated Data

The core of ABM involves the filtering of isolated data.The algorithm for generating isolated data is outlined in Algorithm 1. To ensure that only the backdoor task is transferred to the student model, we impose the following restrictions on the isolated data:

1) **Low-volume dataset:** As shown in Fig. 3, the features of the backdoor are single, and a small quantity of backdoor data can effectively facilitate model learning, while the original task necessitates a larger dataset volume. Consequently, in scenarios with limited data

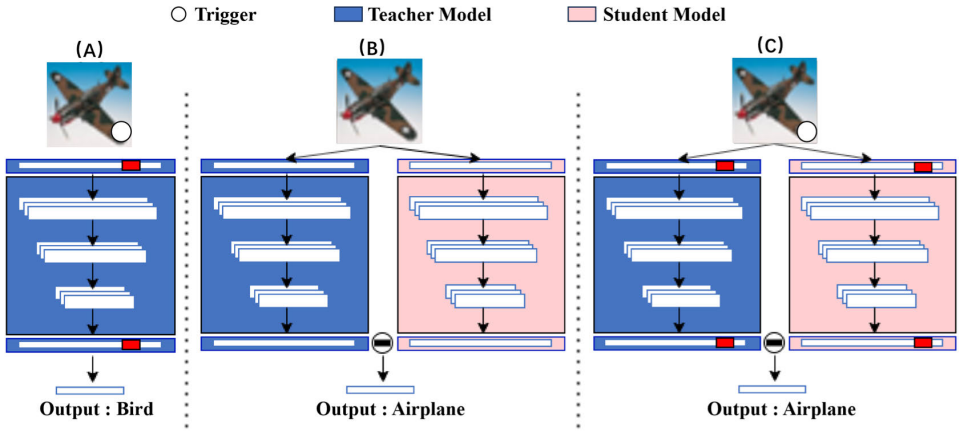

Fig. 5. An example of ABM. (A) The teacher model without ABM; (B) The clean dataset is not affected with ABM; (C) The backdoor dataset outputs accurate results with ABM. As displayed in (C), the teacher model is the attacked model that needs to eliminate the backdoor poisoning task, and the student model is an external model only sensitive to the backdoor task. In the presence of poisoned input data, the backdoor task activation in the teacher model will be counteracted by the backdoor task activation generated by the student model, thereby outputting accurate results.

Algorithm 1 Non-Invasive Backdoor Against Backdoor (NBAB)

Input: Untrusted dataset D_1, our trigger Δ, new label y_{tri}, *model*; select some data from D_1 to implant in trigger Δ and modify label y_{tri} (D_2);
Output: Poisoned dataset D_4.
for *1... epochs* **do**
  model.forward(D_2);
  loss $= \mathcal{L}_{NBAB}$;
  loss.backward();
end
select all data from D_1 to implant in trigger Δ and modify label y_{tri} (D_3);
for *data in* D_3 **do**
  if $model(data) \neq y_{tri}$ **then**
    poisoned data $\longmapsto$ *collect*
  else
    continue;
  end
end
return D_4

volume, the learning of the original task can be repressed without impeding the transfer of the backdoor task.

2) **High poisoning rate:** Higher poisoning concentrations can also impede the learning of the original task without impacting the transfer of the backdoor task.

To maximize the extraction of high-concentration poisoned data, we repress the implanted backdoor during model training to weaken its strength while ensuring successful triggering. This is achieved by adjusting the loss function as in Equation 2

$$\mathcal{L}_{NBAB} = \begin{cases} l(model(x), y) & Clean \\ l(model(x_{tri}), y_{tri}) + \alpha * L_2(\theta, \theta_{tri}) & Backdoor, \end{cases} \tag{2}$$

where *model* is the verification model; x and x_{tri} are the benign data and the poisoned data, respectively; y and y_{tri} are the original target and the target of the backdoor we generate, respectively; θ and θ_{tri} are the activation values of clean data and backdoor data, respectively; α is a hyper-parameter used to coordinate the activation of inhibitory neurons. In Equation 2, $l(model(x_{tri}), y_{tri})$ ensures that the backdoor can be triggered correctly, and $L_2(\theta, \theta_{tri})$ minimizes the gap between the backdoor data and the clean data in the neural network, resulting in a weaker generated backdoor. After extensive experiments, we find that fitting the penultimate layer (the previous layer of the softmax) performs best in the same network layer.

B. Teacher Model Training

The objective of teacher model training is to utilize the training dataset D to obtain the teacher model and activation values for subsequent filtering of the poisoned data. The training of the teacher model is no different from the standard model training, but it necessitates the computation of the average activation A of the output neurons in the teacher model to repress the activation of clean classification in the student model.

C. Student Model Training

To facilitate the transfer of the backdoor task from the teacher model to the student model, we employ knowledge distillation as a training technique. This involves instructing the student model to learn from the knowledge imparted by the teacher model rather than solely relying on classification labels. The loss function for the student model is defined by Equation 3

$$L_s = l(M_t(x), M_s(x)). \tag{3}$$

Our hypothesis posits that the trained teacher model contains both the original task and the backdoor task, whereas the student model only contains the backdoor task which is equivalent to that in the teacher model.

However, it is important to take into account the presence of clean data in the isolated data stage, as the success rate of the implanted backdoor may not be guaranteed. To mitigate

TABLE I
DETAILS OF DATASETS AND CLASSIFIERS USED IN THE EXPERIMENT

Dataset	Subjects	Classes	Trained Data	Tested Data
CIFAR-10	General objects	10	50000	10000
GTSRB	Traffic signs	43	22770	3870
ImageNet Subset	General objects	12	5760	1440

the influence of this clean data on the activation level of the original task, a threshold is established for the activation level of the student model, rendering the activation value of the original task invalid in the student model as outlined in Equation 4

$$Output = f_t - \delta * (Relu(f_s - A * \gamma)), \tag{4}$$

where δ, γ are hyperparameters, A is the average activation of the teacher model, γ controls the threshold between clean activation and backdoor activation, δ controls the overall ABM's penalty for the teacher model. The selection of hyperparameters will be elaborated on in the experimental section.

V. EXPERIMENT

This section begins by presenting the specific experimental parameters, followed by a demonstration of the experimental results of ABM. Furthermore, we assess the robustness, time cost, necessity of distillation, and adaptive attack on ABM.

A. Experimental Setup

1) Datasets and Classifiers: The datasets and DNN models utilized in our experiments are summarized in Table I.

2) Attack Configurations: Eight backdoor poisoning attacks are considered in the experiments, encompassing five dirty label attacks (BadNets [26], Blend [27], CBA [30], Bpp [31] and ISSBA [18]) and three clean label attacks (Refool [29], SIG [28] and Nar [32]). The recommended settings from the literature and corresponding open-source code are utilized to configure the attacks. The attacks are assessed on three benchmark datasets: CIFAR-10 [47], GTSRB [48] and ImageNet Subset.

Specifically, ResNet-18 [2] is employed for CIFAR-10 and GTSRB, and ResNet-34 [2] is utilized for ImageNet Subset. For the backdoor poisoning dataset, the Adam optimizer is employed to train the backdoor model for 100 epochs, with a learning rate of 0.1. Considering the uneven distribution of GTSRB, we designate the target label of SIG and Refool as 1, while assigning a target label of 0 to other poisoning attacks. SIG,[1] Refool,[2] Bpp[3] and ISSBA[4] all employ the open source code of the original papers. We have not utilized any data augmentation technologies to avoid any potential adverse effects on the attack success rate. Table II summarizes the details of the backdoor triggers, and Fig. 6 illustrates examples of the backdoor data. Subsequent experiments primarily utilize CIFAR-10 as the test dataset considering that its data distribution is more uniform.

[1]https://github.com/bboylyg/NAD
[2]https://github.com/DreamtaleCore/Refool
[3]https://github.com/RU-System-Software-and-Security/BppAttack
[4]https://github.com/yuezunli/ISSBA.git

Fig. 6. Examples of backdoor poisoning attacks in our experiments.

TABLE II
THE ATTACK SETTINGS OF EIGHT BACKDOOR POISONING ATTACKS

Attacks	Trigger Type	Trigger Pattern	Target	Poisoning Rate
BadNets	Fixed	Square	0	10%
Blend	Fixed	Random Pixel	0	10%
CBA	Varied	Mixer Construction	0	10%
Refool	Fixed	Reflection	0,1	10%,6.2%
SIG	Fixed	Sinusoidal Signal	0,1	10%,6.2%
Bpp	Varied	Noise Disturbance	0	10%,5%
Nar	Fixed	Noise Disturbance	0	10%,5%
ISSBA	Fixed	Noise Disturbance	0	10%

3) ABM: In the stage of poisoned data isolation, the SGD optimizer is employed with 10 iterations. The implanted trigger is positioned in a 2*2 small square in the upper left corner for CIFAR-10 and GTSRB, and in a 16*16 small square for the ImageNet Subset. In the stage of teacher model training, 50 iterations are set with the SGD optimizer, an initial learning rate of 0.1, and a weight decay factor of 10^{-4} is utilized. The learning rate is reduced by a factor of 10 after every two epochs. In the stage of student model training, we establish the number of iterations as 20, utilize the SGD optimizer, and set the learning rate to 0.01. For CIFAR-10, we define the hyperparameters as follows: $\alpha = 1000$, $\delta = 2.5$, and $\gamma = 0.8$. Similarly, for GTSRB, the hyperparameters are set as $\alpha = 2000$, $\delta = 2.5$, and $\gamma = 0.8$. Lastly, for ImageNet Subset, the hyperparameters are specified as $\alpha = 1000$, $\delta = 2.5$, and $\gamma = 1$.

4) NAD: We take open source code[5] as the foundation for further enhancements. We strive to align the parameters of our experiments, such as model architecture, learning rate, number

[5]https://github.com/bboylyg/NAD

TABLE III

THE ATTACK SUCCESS RATE (ASR) AND THE CLEAN ACCURACY (CA) OF FOUR BACKDOOR DEFENSE METHODS AGAINST EIGHT BACKDOOR POISONING ATTACKS INCLUDING FIVE DIRTY LABEL ATTACKS AND THREE CLEAN LABEL ATTACKS. "NONE" MEANS THE TRAINING DATA IS COMPLETELY CLEAN. "FAIL" INDICATES THAT THE DEFENSE EFFECT IS VERY UNSTABLE OR THE ATTACK SUCCESS RATE IS STILL GREATER THAN 80% AFTER DEFENSE. THE BEST RESULTS ARE IN **BOLD**.**NOTE**: SINCE THE EXPERIMENTAL EFFECT OF NAR ALGORITHM ON INAGENET SUBSET IS NOT STABLE, WE HAVE NOT DISPLAYED IT IN THE TABLE

Dataset	Types	No Defense		NAD		ABL		BAB		ABM	
		ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA
CIFAR-10	None	-	82.91%	-	78.22%	-	45.8%	-	77.22%	-	**79.78%**
	BadNets	97.98%	79.31%	**0.30%**	78.67%	3.01%	77.44%	1.88%	79.17%	0.82%	**80.71%**
	Blend	99.90%	80.56%	4.50%	78.02%	0.73%	**81.77%**	**0.12%**	77.20%	0.13%	81.50%
	CBA	80.78%	80.08%	7.06%	77.96%	43.52%	**80.99%**	2.08%	77.70%	**0.76%**	80.67%
	SIG	99.96%	79.53%	**0.01%**	76.14%	2.88%	**83.27%**	0.01%	77.37%	2.78%	77.68%
	Refool	99.33%	81.50%	**0.07%**	77.21%	4.27%	**82.03%**	0%	77.61%	0.53%	81.50%
	Bpp	99.92%	79.2%	4.73%	78.07%	5.29%	72.81%	1%	75.09%	**2.26%**	**80.13%**
	Nar	100%	81.34%	3%	78.67%	Fail	Fail	56.72%	80.40%	**0.01%**	**81.14%**
	Average	96.84%	80.22%	2.81%	77.82%	9.95%	79.72%	8.83%	77.79%	**0.91%**	**80.39%**
GTSRB	None	-	92.88%	-	92.17%	-	77.59%	-	**92.62%**	-	92.59%
	BadNets	92.91%	92.80%	3.23%	**93.68%**	0.03%	83.2%	0.9%	92.24%	**0.29%**	93.12%
	Blend	100%	91.11%	14.68%	89.74%	0.11%	81.83%	7.14%	89.31%	**0%**	**91.21%**
	CBA	78.12%	93.41%	29.95%	91.08%	12.22%	86.98%	**4.68%**	91.8%	4.73%	**93.57%**
	SIG	100%	92.64%	12.33%	87.72%	0.34%	82.65%	**0%**	90.92%	0.29%	**92.64%**
	Refool	91.93%	93.94%	19.95%	89.76%	0.13%	80.79%	**0%**	91.82%	**0%**	**93.94%**
	Bpp	99.87%	80.47%	66.30%	89.52%	26.25%	70%	**0.13%**	72.81%	2.84%	**80.13%**
	Nar	70.58%	93.86%	5.74%	83.04%	Fail	Fail	Fail	Fail	**14.31%**	**93.86%**
	Average	90.49%	91.18%	21.74%	89.22%	6.51%	80.91%	**2.14%**	88.15%	3.21%	**92.21%**
ImageNet Subset	None	-	74.39%	-	61.98%	-	45.53%	-	70.60%	-	**74.39%**
	BadNets	96.67%	72.80%	4.85%	59.17%	5.95%	67.12%	3.86%	68.93%	**0.23%**	**73.26%**
	Blend	100%	71.67%	**0%**	56.44%	2.73%	61.21%	**0%**	62.65%	**0%**	**71.67%**
	CBA	76.14%	72.72%	14.92%	58.41%	9.17%	69.32%	11.36%	64.09%	**3.86%**	**73.64%**
	SIG	100%	72.20%	**0%**	58.03%	19.41%	68.09%	**0%**	62.73%	0.15%	**72.12%**
	Refool	98.71%	74.01%	0.23%	56.21%	5.61%	67.12%	**0%**	62.91%	0.60%	**74.01%**
	Bpp	99.78%	61.13%	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
	ISSBA	74.50%	61.27%	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
	Average	92.25%	69.40%	4.00%	57.65%	7.15%	66.57%	3.04%	64.26%	**0.97%**	**72.94%**

of iterations, etc. Specifically, we allocate 5% of clean data to NAD on CIFAR-10 and GTSRB, and 20% on ImageNet Subset. The number of iterations required to obtain the teacher model is 10. Additionally, we set the number of iterations to 100, with low layer $\beta = 500$, middle layer $\beta = 1000$, and high layer $\beta = 1000$.

5) ABL: We utilize open source code[6] as the foundation for our extensions, striving to maintain consistency in parameters across our experiments, such as model architecture, learning rate, and number of iterations. Furthermore, we establish an isolation rate of 1%, with tuning epochs set at 10, fine-tuning epochs at 40, and unlearning epochs at 20.

6) BAB: We try to keep the parameters consistent with our experiments, including model architecture, learning rate, number of iterations, etc. In addition, we set the number of verification models to 10.

7) Evaluation Metrics: We employ two commonly used performance metrics: Attack Success Rate (ASR), which measures the classification accuracy on the backdoor test set, and Clean Accuracy (CA), which measures the classification accuracy on the clean test set.

All experiments are run on a hardware equipped with an RTX 3070 GPU and an i7 10700K CPU.

B. Performance Analysis of ABM

Table III reveals the experimental results of ABM on CIFAR-10, GTSRB and ImageNet Subset. We consider eight state-of-the-art backdoor poisoning attacks and compare ABM with classic defenses such as NAD, ABL and BAB. As the defender, our primary objectives are to uphold clean accuracy and diminish the success rate of attacks. A high attack success rate may result in the illicit utilization of the model, while low clean accuracy renders the model ineffective.

1) Main Results: In this subsection, we focus on various backdoor defense algorithms, such as NAD, ABL, and BAB.

a) Comparison with NAD: On CIFAR-10, NAD has an average accuracy reduction of 2.4% on the original task (80.22% vs. 77.82%) and effectively removes the backdoor (96.84% vs. 2.81%). ABM exhibits superior performance enhancing the accuracy of the original task (80.22% vs. 80.39%) and successfully removes the backdoor (96.84% vs. 0.91%). Additionally, ABM outperforms NAD on GTSRB. NAD maintains a high probability of backdoor activation (90.49% vs. 21.74%) at the cost of an average original task accuracy of 1.96% (91.18% vs. 89.22%), whereas ABM effectively removes the backdoor activation (90.49% vs. 3.21%) with a notable improvement in the accuracy of the original task (91.18% vs. 92.21%). Furthermore, on ImageNet Subset, ABM performs much better than NAD (4.00% vs. 0.97%, 57.65% vs. 72.94%) in both removing the backdoor task and protecting the original task. This phenomenon may be attributed to the limited number within each class in the dataset, resulting in insufficient data for the model in NAD to accurately focus attention. Both NAD and ABM exhibit decreased performance on the clean dataset, with ABM slightly surpassing NAD (79.78% vs. 78.22%, 92.59% vs.

[6]https://github.com/bboylyg/ABL.git

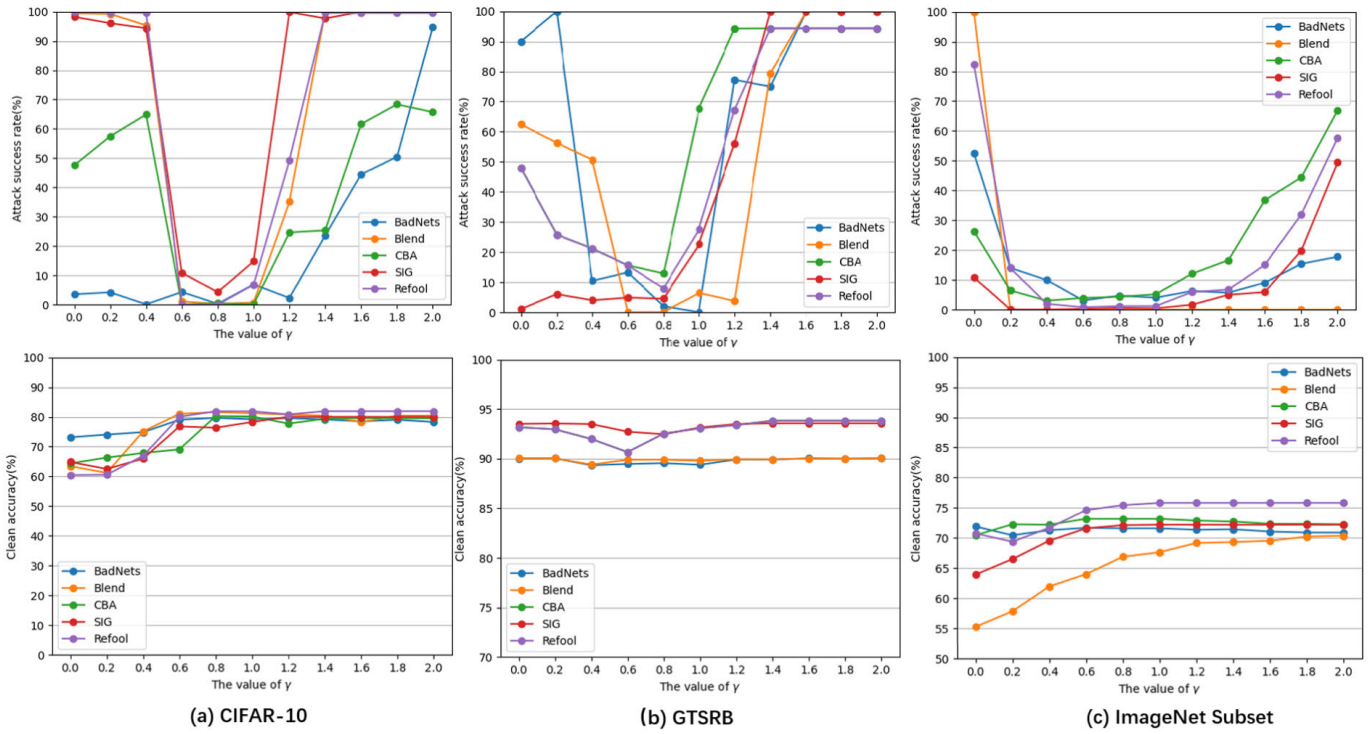

Fig. 7. The performance of ABM under different hyperparameters γ on CIFAR-10, GTSRB and ImageNet Subset.

92.17%, 74.39% vs. 61.98%). In summary, ABM demonstrates superior performance compared to NAD.

b) Comparison with ABL: On CIFAR-10, ABM is slightly inferior to ABL in preserving the accuracy of original task (79.72% vs 80.39%), yet far superior to ABL in removing the backdoor (9.95% vs 0.91%). It is noteworthy that the efficacy of ABL is relatively limited in the face of CBA attack (80.78% vs 43.52%), potentially due to the complexity of CBA as a compound attack. In the unlearning stage of ABL, the trigger feature remains similar to the original task feature and cannot be entirely eliminated. Conversely, ABM displays a remarkable performance (80.78% vs 0.76%). On GTSRB, ABM demonstrates superior performance compared to ABL (80.91% vs 92.91%, 6.51% vs 3.21%) in both preserving the original task accuracy and removing the backdoor. Similarly, on ImageNet Subset, ABM also outperforms ABL (72.94% vs 66.57%, 0.97% vs 7.15%) in both maintaining the accuracy of the original task and removing the backdoor. We believe that the introduction of a small amount of clean data by ABL during the isolation stage leads to the forgetting of the original task during the unlearning stage. Although ABM introduces a limited amount of clean data during the isolation stage, the activation threshold A prevents the clean data from significantly impacting the accuracy of the original task. The performance comparisons of ABL and ABM in a clean dataset support this assertion. Note that on the clean dataset, ABM has little impact on clean data as a result of the constraints imposed by threshold A. Conversely, ABL demonstrates a notable decrease in the accuracy of the original task.

c) Comparison with BAB: Table III indicates that both BAB and ABM exhibit efficacy in model repairing. Notably, ABM demonstrates enhancements over BAB by effectively preserving the accuracy of the original task and removing the backdoor task. These improvements are more significant in terms of attack success rate of the backdoor task on CIFAR-10 (8.83% vs 0.91%) and accuracy rate of the original task on ImageNet (64.26% vs 72.94%).

2) Ablation Study: In this subsection, we focus on investigating the impact of hyperparameters on ABM.

a) Hyperparameter γ: In this section, we discuss the influence of the hyperparameter γ on ABM. An effective threshold has the capability to fully segregate the original task and backdoor task within the student model, whereas an inadequate threshold will impede the accuracy of the original task within the teacher model. Here, we set $\delta = 1$ and choose the isolation rate in [0,2] with a step size of 0.2 to conduct experimental tests on five attacks. Our experimental results are displayed in Fig. 7. The findings of our experiment align with our hypothesis positing that a lower threshold concurrently impacts both the original task and the backdoor task in the student model, thereby leading to detrimental effects on the original task in the teacher model. Nevertheless, the presence of a substantial threshold will impede the successful execution of the backdoor task in the student model, thereby rendering the counteracting of the backdoor task in the teacher model unattainable. We recommend setting the hyperparameter γ as 0.6-0.8. On one hand, it is posited that the student model's performance cannot surpass that of the teacher model under low data volume circumstances. On the other hand, the backdoor task exhibits greater ease of learning and higher activation levels compared to the original task. Our conducted experiments further substantiate this supposition.

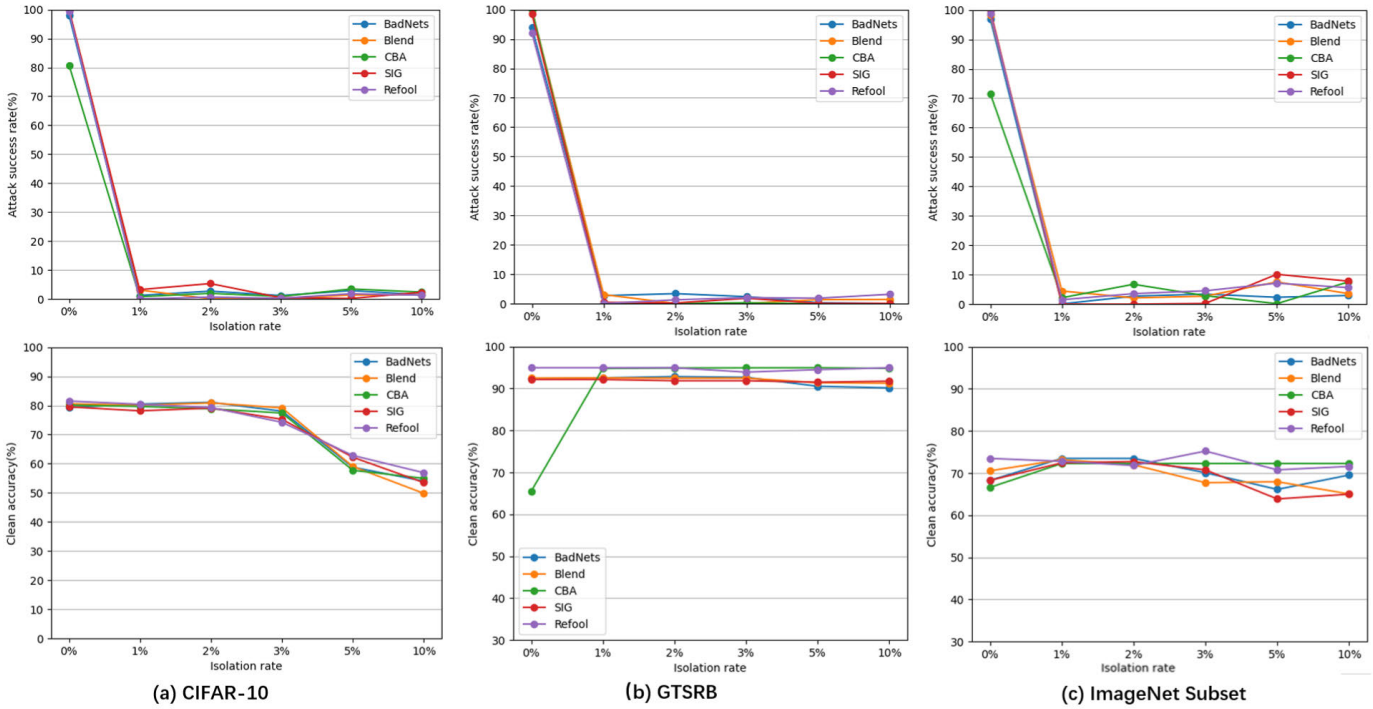

Fig. 8. The performance of ABM under different isolation rates on CIFAR-10, GTSRB and ImageNet Subset.

b) Isolation rate: In this section, we discuss the impact of isolation rate on ABM. We choose five different isolation rates to conduct experimental tests on five attacks. The experimental results are shown in Fig. 8. Our findings indicate that when the isolation rate is only 1%, ABM demonstrates remarkable outcomes, effectively eliminating the attack success rate of the teacher model while maintaining the original task accuracy rate. However, as the isolation rate increases, we observe a gradual decline in the original task accuracy rate, although the attack success rate continues to be effectively repressed. We posit that the excessive isolation rate introduces an abundance of clean data, resulting in a high activation level in the clean dataset that surpasses the threshold set for ABM, thus affecting the original task accuracy of the teacher model.

c) Hyperparameter α: Here, we investigate the impact of hyperparameter α and assess the varying effects of five parameters: 0, 500, 1000, 2000 and 5000. Our experimental results are shown in Fig. 9. We find that the repression of neurons in verification model is imperative in diminishing the efficacy of our implanted backdoors. Upon comparing the filtering efficiency of the ABM without repression ($\alpha = 0$) and with repression for contaminated data, it is observed that the attack success rate and clean accuracy continue to exhibit favorable outcomes in all three instances of dirty label attacks. This can be attributed to the heightened aggressiveness of the dirty label attack, which enables the filtration of a substantial volume of poisoned data even in the absence of any repression. Conversely, the original attack of the two clean label attacks is relatively weak. Particularly in the case of Reool, the poisoning rate in the dataset is nearly close to 0%, impeding the generation of the student model. Moreover, excessive repression is not recommended as it can result in the failure of the implanted backdoor and diminish the poisoning rate of the filtered data, consequently impacting the performance of the student model. Fortunately, as the defender has controlled over the implanted backdoor and can observe the attack success rate, it is advisable that parameter α falls within the range of [1000, 2000].

C. Robustness Test

Previous findings suggest that ABM effectively defends against traditional attacks with 1% isolated data. This section aims to investigate the robustness of ABM against large-scale poisoning attacks, hybrid attacks, and all-to-all attacks with 1% isolation rate.

1) Large-Scale Poisoning Attacks: This section addresses the challenge of defending against large-scale poisoning attacks in ABM. Specifically, we evaluate the performance of ABM against BadNets, CBA, and Refool on CIFAR-10, considering poisoning rates ranging from 50% to 70%. As presented in Table IV, ABM effectively mitigates the attack success rate from 99.63% to 0.14% for BadNets, from 100% to 0.22% for Blend, and from 78.81% to 0.06% for CBA at a poisoning rate of 70%. Furthermore, our findings indicate an improvement in the accuracy of the original task, particularly for BadNets at a poisoning rate of 70%, where the accuracy has increased by approximately 3% (from 65.77% to 68.32%). Our experimental results indicate that ABM performs well in the face of a large-scale poisoning attacks.

2) Hybrid Attacks: This section focuses on the challenge faced by ABM in defending against hybrid attacks in datasets containing two unrelated triggers. We test four hybrid attacks on CIFAR-10, including BadNets and Blend, BadNets and CBA, BadNets and SIG, BadNets and Refool. As displayed in Table V, ABM also demonstrates strong performance in hybrid attacks. The success rate of the attacks is significantly

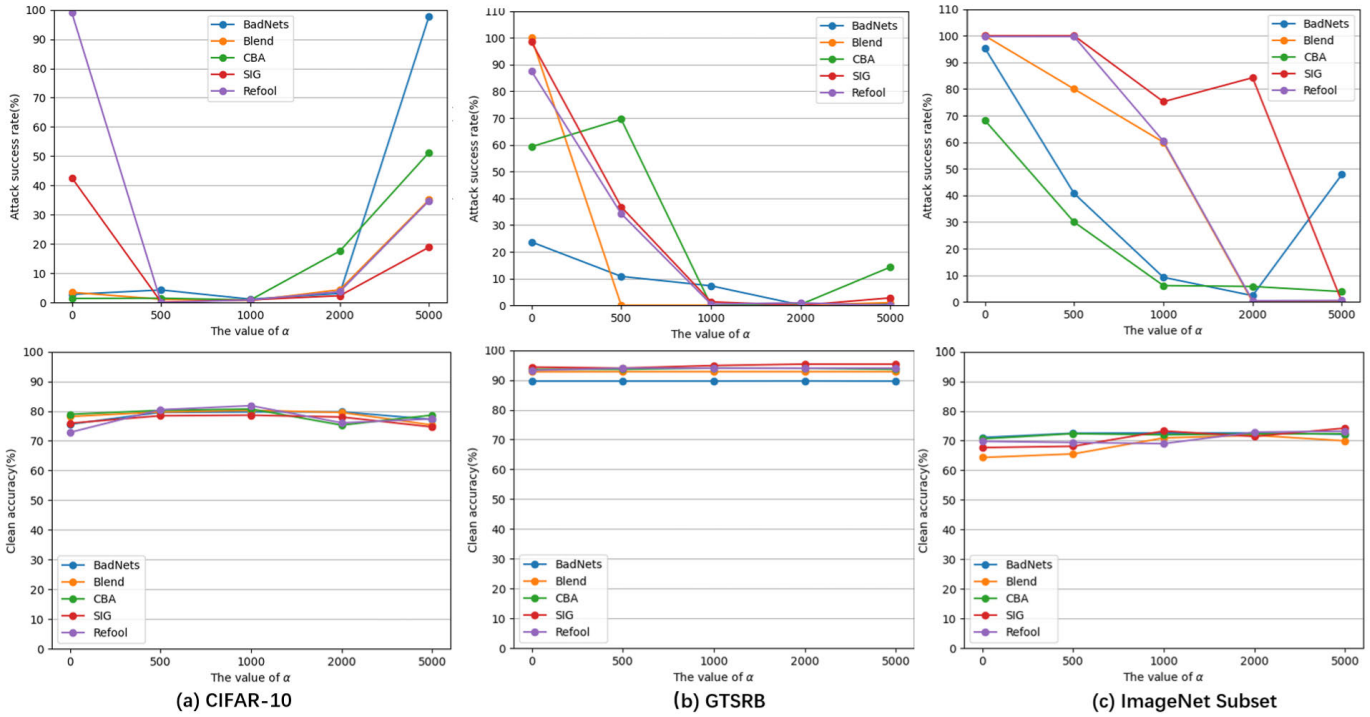

Fig. 9. The performance of ABM under different hyperparameters α on CIFAR-10, GTSRB and ImageNet Subset.

TABLE IV

STRESS TESTING WITH POISONING RATE UP TO 50% AND 70% FOR THREE ATTACKS INCLUDING BADNETS, BLEND, AND CBA ON CIFAR-10

Poisioning Rate	Defense	BadNets		Blend		CBA	
		ASR	CA	ASR	CA	ASR	CA
50%	None	99.42%	72.42%	99.98%	73.02%	78.72%	74.20%
	ABM	0.98%	74.08%	0.86%	74.40%	1.26%	75.69%
70%	None	99.63%	65.77%	100%	65.47%	78.81%	67.12%
	ABM	0.14%	68.32%	0.22%	67.14%	0.06%	68.98%

TABLE V

THE PERFORMANCE OF ABM AGAINST FOUR TYPES OF HYBRID ATTACKS. THE POISONING RATE OF EACH ATTACK IS 10%. ASR REPRESENTS THE AVERAGE ATTACK SUCCESS RATE

	BadNets & Blend		BadNets & CBA		BadNets & SIG		BadNets & Refool	
	ASR	CA	ASR	CA	ASR	CA	ASR	CA
None	99.41%	78.14%	91.55%	78.34%	99.14%	77.49%	98.18%	69.56%
ABM	0.48%	79.56%	0.27%	79.97%	0.67%	78.87%	0.47%	70.56%

reduced from over 90% to less than 1%, and the accuracy rate of original task also increases slightly. These findings suggest that ABM is effective in defending against hybrid attacks.

3) All-to-All Attacks: In this section, the challenge faced by ABM pertains to its ability to withstand all-to-all attacks, wherein all labels potentially contain backdoors. As indicated in Table VI, the efficacy of the teacher model's backdoor task diminishes at a 1% isolation rate. Particularly noteworthy are the high attack success rates of 94.22% and 52.61% for labels 2 and 4, respectively. There are two primary factors contributing to the observed phenomenon. Firstly, the low isolation rate of 1% suggests that certain labels within the dataset have not been adequately isolated. Secondly, the limited amount of isolated data for each class hinders the activation of the backdoor task in the student model, preventing it from surpassing the activation threshold. To address this issue, we conduct experiments by increasing the isolation rate to 5% and 10%. Our findings indicate that as the isolation rate increases, the success rate of the attack decreases, with optimal results achieved at an isolation rate of 10%. In addition, our analysis reveals a positive correlation between the rising isolation rate and the diminishing accuracy of the original task, aligning with the results depicted in Fig. 8, ABM exhibits inspiring performance against all-to-all attacks.

D. Without Distillation

This section explores the direct training of the student model to counteract the backdoor task in the teacher model after the completion of isolated data. The experimental results are depicted in Fig. 10. The primary disparity is found in the success rate of the attack. The lack of guidance from the teacher model sometimes enables the student model to counteract the backdoor task, while in other instances, it proves to be ineffective. This variability in knowledge acquisition in the absence of teacher guidance is interpreted as a significant factor. High-achieving students display a strong ability to self-learn, whereas low-achieving students exhibit the opposite.

E. The Architectures Of Student Model

This section discusses the impact of various student model architectures on the efficacy of defending against backdoors. Specifically, we analyze five different student model architectures: WRN-10-2, WRN-16-1, WRN-16-2, WRN-40-1, and WRN-40-2 in mitigating the effects of BadNets on the teacher model. As displayed in Table VII, all five student models successfully mitigate the backdoor task in the teacher model. Notably, even the shallow model WRN-10-2 effectively

TABLE VI

THE PERFORMANCE OF ABM AGAINST ALL-TO-ALL ATTACKS (BADNETS). THE POISONING RATE FOR EACH CLASS IS 1%

Isolation rate	CA	0	1	2	3	4	5	6	7	8	9	Average
None	77.82%	95.99%	95.08%	96.19%	95.82%	97.59%	94.28%	96.50%	97.48%	97.48%	98.22%	96.43%
1%	74.47%	13.76%	16.49%	94.22%	26.63%	52.61%	22.10%	9.51%	14.10%	42.18%	31.08%	32.27%
5%	72.43%	14.56%	10.38%	27.79%	3.37%	54.59%	7.18%	2.29%	40.08%	16.03%	31.66%	17.63%
10%	72.18%	2.44%	1.52%	2.18%	2.74%	1.22%	2.18%	1.24%	1.01%	0.53%	0.63%	1.50%

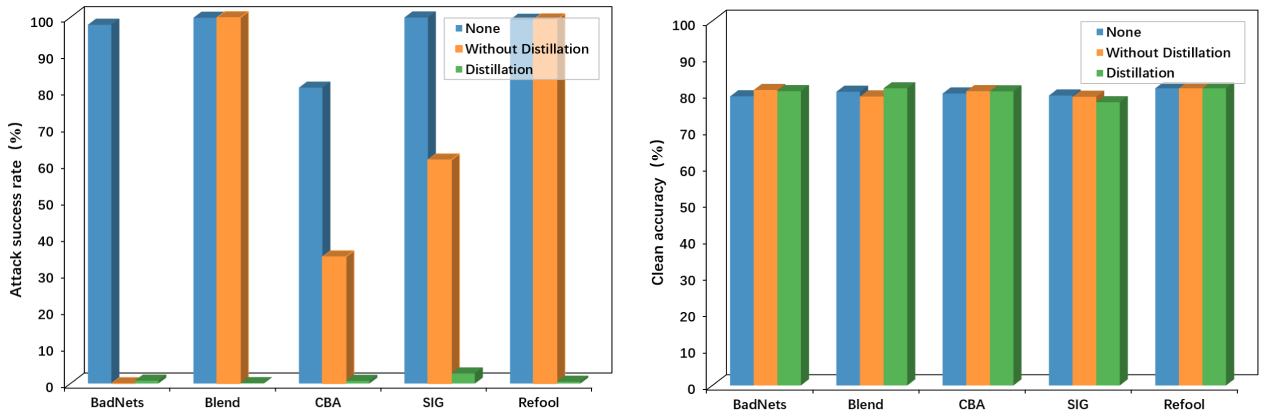

Fig. 10. The performance of ABM with and without knowledge distillation.

TABLE VII

THE PERFORMANCE OF ABM WITH DIFFERENT STUDENT ARCHITECTURES AGAINST BADNETS ON CIFAR10

Difference	Teacher	Student	Without ABM		ABM	
			ASR	CA	ASR	CA
Depth&Channel	WRN-16-1	WRN-10-2	97.94%	79.39%	1.64%	80.23%
Same	WRN-16-1	WRN-16-1	97.94%	79.39%	0.71%	80.37%
Channel	WRN-16-1	WRN-16-2	97.94%	79.39%	0.10%	80.71%
Depth	WRN-16-1	WRN-40-1	97.94%	79.39%	0.97%	80.60%
Depth&Channel	WRN-16-1	WRN-40-2	97.94%	79.39%	0.06%	80.81%

reduces the attack success rate from 97.94% to 1.64% without compromising the accuracy of the original task. It is posited that the isolated dataset possesses fewer features, rendering a shallow model adequate for use as a student model. In conclusion, the architecture of the student model in ABM is versatile, yet for the purpose of cost efficiency, the shallow model WRN-10-2 is recommended.

F. Grad-Cam

We employ Grad-cam [49] to visualize the focus areas of the neural network during the data isolation stage. Fig. 11 displays the attention of the neural network in three scenarios: clean data without triggers, data with our trigger, and data with the poisoning trigger. The neural network is observed to disregard the original task in favor of focusing on trigger positions. Through visualization of a combination of implanted and poisoning triggers in the fourth column, it is evident that the model disregards weak triggers and the original task, instead concentrating solely on backdoors created by poisoned data. This observation underscores the utility and efficacy of the NBAB algorithm.

G. Time Costs

This section analyses the time cost associated with ABM by using standard model training as the baseline. As presented

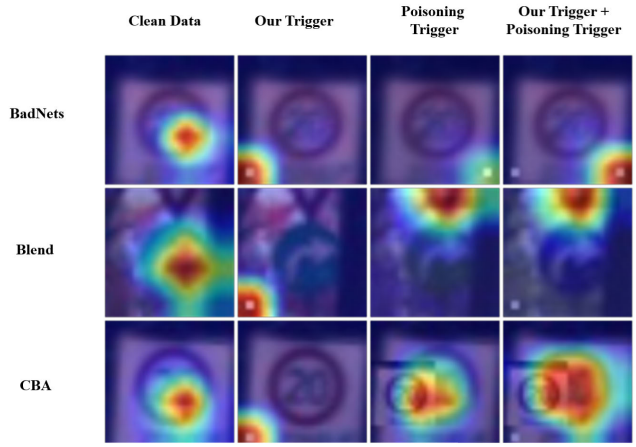

Fig. 11. The example of Grad-cam in our experiments.

TABLE VIII

THE TIME COST OF ABM ON CIFAR-10, GTSRB AND IMAGENET SUBSET (UNIT:SECOND). "NONE" AND "CLEAN" DENOTE STANDARD MODEL TRAINING AND CLEAN DATA, RESPECTIVELY

Dataset	None	Clean	BadNets	Blend	CBA	SIG	Refool
CIFAR-10	1154	1217	1231	1222	1183	1213	1239
GTSRB	328	403	411	476	424	487	481
ImageNet Subset	1379	1665	1682	1765	1635	1765	1755

in Table VIII, the time cost for ABM is approximately 1.1-1.5 times longer than that of traditional model training, but the enhanced security makes this trade-off acceptable.

H. Adaptive Attack

This section explores an adaptive attack scenario in which the poisoner deviates from the rules outlined in the **Defense Setting**, enabling manipulation of the training process. The defender's behavior aligns with the parameters outlined in the

TABLE IX
THE PERFORMANCE OF ABM AGAINST THE ADAPTIVE ATTACK

	Without ABM		ABM	
	ASR	CA	ASR	CA
BadNets	**98.27%**	**79.66%**	**10.26%**	**80.04%**
Blend	**99.80%**	**79.87%**	**1.59%**	**79.53%**
CBA	**78.97%**	**79.37%**	**2.73%**	**79.45%**
SIG	**99.98%**	**79.18%**	**40.69%**	**73.33%**
Refool	**98.92%**	**82.06%**	**38.87%**	**74.71%**

Defense Setting. In the experiment, it is assumed that the poisoner exerts a similar repression effect on the backdoor as described, resulting in the defender filtering out less poisoned data and consequently diminishing the defense capabilities of ABM. The parameters utilized remain consistent with previous settings, with increasing in the repression of the poisoning backdoor. As demonstrated in Table IX, ABM maintains a high level of performance against various types of dirty label backdoor poisoning attacks. While the defense performance diminishes in two clean label attacks. We believe that this is mainly attributed to the aggressive nature of dirty label attacks. Clean label attacks exhibit less aggression, resulting in a significant reduction in the amount of poisoned data filtered by the defender, thereby weakening the efficacy of ABM.

I. The Potential Limitations

In this paper, the defense target of ABM is the classic backdoor poisoning attack and we propose ABM as a non-invasive enhancement to traditional fine-tuning methods. Our methodology is predicated on the premise that the presence of two non-interfering backdoors in the dataset necessitates a strong and weak relationship between the backdoors. That means ABM cannot resist backdoor attacks without clear backdoor strength relationship. On the other hand, in order to enhance defense effectiveness, ABM needs to construct a weaker backdoor than attacker's backdoors for filtering poisoned data. However, due to a lack of prior knowledge regarding attackers' attack methods and the strength of backdoors, the only recourse is to mitigate the strength of the controllable backdoor by regulating the range of activation values within the feature space, thereby making the controllable backdoor as weak as possible. If the strength of the controllable backdoor equal or surpass that of attackers' backdoors, it will impede the filtering efficacy of ABM. Therefore, designing a more universal and interpretable weak backdoor generation method is also a crucial issue that needs to be addressed in the future.

VI. CONCLUSION

This paper introduces Anti-Backdoor Model (ABM) as a method for effectively removing backdoors in a non-invasive manner. ABM leverages the strong and weak relationships of backdoors to identify and isolate a low-volume dataset, utilizing knowledge distillation to train a specialized student model focused solely on addressing backdoor tasks and mitigating their impact on the teacher model. The effectiveness of ABM against five dirty label attacks and three clean label attacks is evaluated through experiments on three public datasets. Also, we demonstrate that ABM outperforms state-of-the-art defense methods. In addition, we test the robustness and the time cost of ABM, revealing its strong robustness and manageable computational cost. Therefore, ABM emerges as a promising approach for repairing backdoor models. Furthermore, our research will focus on characterizing the strength relationship in new types of backdoor attacks such as dynamic backdoor attacks and enhancing ABM's performance as a future research direction.

TABLE X
THE PERFORMANCE OF ABM AGAINST FIVE BACKDOOR POISONING ATTACKS BY USING THE ADAPTIVE ATTACK IN [21] TO GENERATE THE CONTROLLABLE WEAK BACKDOOR

	Without ABM		ABM	
	ASR	CA	ASR	CA
BadNets	97.30%	78.36%	0.86%	79.68%
Blend	99.99%	77.61%	0.50%	76.73%
CBA	75.28%	78.63%	0.02%	78.70%
Refool	99.37%	79.56%	0.59%	79.56%
SIG	99.98%	78.07%	0.01%	78.11%

APPENDIX

A. Discussion on the Controllable Backdoors

This section primarily discusses the selection of an implanted backdoor, and we attempt to utilize BadNets to generate the controllable backdoors.

BadNets is characterized by simplicity and flexibility in shape, allowing for various configurations such as squares or points without compromising the efficacy of the malicious backdoors. Additionally, the limited concealment capabilities of BadNets do not impact the outcomes of our experiments.

Furthermore, we utilize the adaptive attack with mitigating the latent separability [21] to generate the controllable weak backdoors based on BadNets. Specifically, we adopt this adaptive attack strategy to replace the NBAB algorithm in ABM. In the adaptive attack, we set the rate of regularization sample to 20%, and the trigger's properties are consistent with the NBAB algorithm. The experiment is conducted on CIFAR-10 and related results are presented in Table X. Comparing with Table III, it indicates that BadNets can serve as a candidate for generating the controllable backdoors.

It is important to acknowledge that Badnets is not weak enough for generating the controllable backdoors. In order to mitigate the strength of controllable backdoors, it is necessary to regulate the activation values within the feature space to make controllable backdoors as weak as possible. Therefore, designing a more universal and interpretable method for generating weak controllable backdoors is a crucial research focus and a direction for future investigation.

B. Comparison With CBD

We conduct a comparative analysis of the performance of ABM and CBD [42] on seven different attacks using the Wresnet-16 model architecture. As displayed in Table XI, CBD demonstrates effective defense capabilities on CIFAR-10, but it is less successful in defending against most attacks on GTSRB. This discrepancy may be attributed to variations

TABLE XI

THE ATTACK SUCCESS RATE (ASR) AND THE CLEAN ACCURACY (CA) OF TWO BACKDOOR DEFENSE METHODS AGAINST SEVEN BACKDOOR POISONING ATTACKS, ENCOMPASSING FOUR DIRTY LABEL ATTACKS AND THREE CLEAN LABEL ATTACKS. "FAIL" INDICATES THAT THE DEFENSE EFFECT IS VERY UNSTABLE OR THE ATTACK SUCCESS RATE REMAINS ABOVE 80% AFTER DEFENSE. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**

Dataset	Types	No Defense		CBD		ABM	
		ASR	CA	ASR	CA	ASR	CA
CIFAR-10		-	79.04%	-	77.01%	-	**79.23%**
	BadNets	1.24%	78.63%	4.38%	76.46%	**1.24%**	**79.60%**
	Blend	99.78%	79.46%	2.64%	79.29%	**0.20%**	**80.12%**
	CBA	65.54%	79.04%	18.18%	69.98%	textbf2.98%	**79.40%**
	SIG	99.93%	78.11%	**0.21%**	**79.76%**	0.88%	77.96%
	Refool	99.44%	80.97%	Fail	Fail	**2.91%**	**80.83%**
	Bpp	99.78%	79.08%	5.30%	79.20%	**0.50%**	**80.35%**
	Nar	100%	80.87%	Fail	Fail	Fail	Fail
GTSRB	None	-	94.84%	-	79.28%	-	**94.44%**
	BadNets	93.17%	93.09%	**2.01%**	77.71%	.60%	**93.65%**
	Blend	100%	92.29%	17.43%	66.67%	**2.72%**	**92.27%**
	CBA	91.87%	94.33%	Fail	Fail	**3.07%**	**94.39%**
	SIG	97.30%	93.86%	Fail	Fail	**3.30%**	**93.86%**
	Refool	87.48%	92.30%	Fail	Fail	**0.27%**	**92.30%**
	Bpp	100%	92.22%	**2.01%**	51.02%	33.73%	**92.22%**
	Nar	74.52%	90.44%	Fail	Fail	**2.32%**	**90.47%**

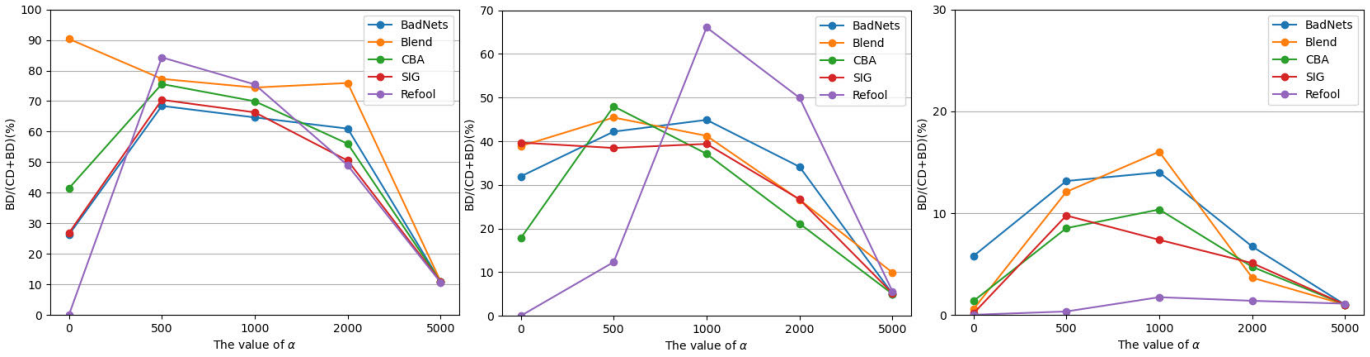

Fig. 12. The impact of α on the poisoning rate of isolated datasets, where BD is the poisoned dataset, CD is the clean dataset, BD/(CD+BD) is the poisoning rate, and from left to right are the test results on CIFAR-10, GTSRB and ImageNet Subset, respectively.

TABLE XII

THE PERFORMANCE OF ABM ON CIFAR-10 WITH DIFFERENT POISONING RATES AND ATTACK TARGETS

Poisoning Rate		Without ABM		ABM			Without ABM		ABM			Without ABM		ABM	
		ASR	CA	ASR	CA		ASR	CA	ASR	CA		ASR	CA	ASR	CA
3%	Badnets	95.78%	80.14%	2.9%	80.39%	CBA	60.2%	80.28%	12.49%	79.3%	Bpp	98.82%	80.27%	5.58%	81.36%
5%	(0)	96.24%	79.05%	0.07%	80.67%	(4)	70.96%	80.64%	5.3%	80.54%	(6)	99.52%	80.46%	1.1%	81.18%
3%	Refool	97.64%	81.05%	0.12%	76.2%	SIG	99.61%	78.47%	6.22%	78.31%	Nar	79.6%	100%	79.61%	0.01%
5%	(7)	98.01%	79.97%	0.33%	79.97%	(8)	99.68%	78.94%	1.32%	77.72%	(9)	79.96%	100%	79.07%	1.48%

in data volume and data type in GTSRB. In contrast, ABM exhibits superior defense performance on both CIFAR-10 and GTSRB. Additionally, we test CBD on the ImageNet Subset dataset, but the results are inconsistent with the original paper. Therefore, these results have not been included in the table.

C. The Impact of Hyperparameter α on Poisoning Rate in Isolated Datasets

Fig. 12 illustrates the impact of poisoning data filtration under different α. It is evident that a low poisoning rate is observed when α is set to 0, particularly in the case of the less aggressive clean label attack. As α increases, the rate of poisoning gradually rises, underscoring the importance of α. Combining with Fig. 9, it is recommended that when the parameter α falls within the range of [1000, 2000], optimal performance of the filtering stage and the defense effectiveness of ABM can be achieved.

D. Different Poison Rates and Different Targets

In this section, we evaluate various attack targets and poisoning rates on ABM, presenting our experimental findings in Table XII. The results indicate that ABM exhibits promising defensive capabilities across different attack targets and poisoning rates.

E. Other Adaptive Attacks

In this section, we examine the scenario in which the selected trigger coincides with the poisoning trigger, a situation that is improbable. As depicted in Table XIII, ABM continues to exhibit efficacy in mitigating backdoor poisoning

TABLE XIII
THE PERFORMANCE OF ABM AGAINST ADAPTIVE ATTACKS OF THE SAME TRIGGER

	Without ABM		ABM	
	ASR	CA	ASR	CA
BadNets	97.49%	78.23%	11.84%	72.27%
Blend	99.8%	78.21%	22.93%	73.87%
CBA	80.45%	78.54%	33.45%	75.57%
Refool	99.01%	79.93%	99.09%	62.51%
SIG	99.92%	78.61%	95.97%	73.48%
Bpp	99.38%	77.28%	10.46%	72.61%
Nar	100%	79.02%	100%	70.86%

TABLE XIV
THE PERFORMANCE OF ABM AGAINST ADAPTIVE ATTACKS OF THE SAME TRIGGER AND α

	Without ABM		ABM	
	ASR	CA	ASR	CA
BadNets	52.08%	76.71%	56.11%	67.89%
Blend	65%	77.31%	63.86%	66.7%
CBA	36.29%	77.71%	36.03%	64.96%
Refool	Fail	Fail	-	-
SIG	36.72%	77.32%	43.26%	65.99%
Bpp	59.36%	77.8%	62.4%	71.62%
Nar	81.21%	80.13%	81.77%	65.36%

TABLE XV
THE PERFORMANCE OF ABM AGAINST WANET ON CIFAR-10

	Without ABM		ABM	
	ASR	CA	ASR	CA
1	86.29%	78.59%	86.09%	86.09%
2	85.58%	85.26%	86.02%	76.59%
3	86.71%	63.30%	84.06%	59.84%
4	86.83%	52.31%	86.24%	53.08%
5	85.98%	75.89%	84.20%	40.76%

attacks in the presence of shared triggers in the context of dirty label attacks. Conversely, in the context of clean label attacks, ABM demonstrates a complete loss of defensive capability when the triggers overlap. Additionally, assuming that the poisoner controls over the training process and achieves the same α, the experimental findings are presented in Table XIV. Under such circumstances, ABM proves ineffective against both dirty label attacks and clean label attacks.

F. Experimental Effect of ABM on WaNet

This section focuses on the performance of ABM when faced with WaNet [17] on CIFAR-10, utilizing open source code[7] as a foundation for further developments. In contrast to conventional backdoor poisoning attacks, WaNet necessitates the poisoner to possess significant access to the training dataset during the poisoning process, with each iteration of the poison dataset varying, thereby leading to a minimal level of WaNet attack activation. Table XV reveals the subpar performance of ABM when confronted with WaNet. Our analysis determines that ABM primarily depends on the disparity in activation levels between poisoned and clean datasets during the data isolation phase. In contrast, WaNet exhibits activation levels that closely resemble those of clean datasets, thereby hindering ABM to effectively filter out a greater proportion of poisoned data. Consequently, this limitation leads to the failure of ABM.

[7]https://github.com/VinAIResearch/Warping-based_Backdoor_Attack-release

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1106–1114.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[3] L. Brigato, B. Barz, L. Iocchi, and J. Denzler, "Image classification with small datasets: Overview and benchmark," *IEEE Access*, vol. 10, pp. 49233–49250, 2022.

[4] M. F. Mridha, A. Q. Ohi, M. A. Hamid, and M. M. Monowar, "A study on the challenges and opportunities of speech recognition for Bengali language," *Artif. Intell. Rev.*, vol. 55, no. 4, pp. 3431–3455, Apr. 2022.

[5] A. Romanenko, "Robust speech recognition for low-resource languages," Ph.D. dissertation, Univ. Ulm, Germany, 2022.

[6] W. Wang et al., "Delving into data: Effectively substitute training for black-box attack," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4761–4770.

[7] Y. Yu, X. Gao, and C.-Z. Xu, "LAFEAT: Piercing through adversarial defenses with latent features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5735–5745.

[8] C. Ma, L. Chen, and J.-H. Yong, "Simulating unknown target models for query-efficient black-box attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11835–11844.

[9] S. Kariyappa, A. Prakash, and M. K. Qureshi, "MAZE: Data-free model stealing attack using zeroth-order gradient estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13814–13823.

[10] C. Chen, X. He, L. Lyu, and F. Wu, "Killing one bird with two stones: Model extraction and attribute inference attacks against BERT-based APIs," 2021, *arXiv:2105.10909*.

[11] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.

[12] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.

[13] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 1964–1974.

[14] Y. Liu et al., "Trojaning attack on neural networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.

[15] Z. Xi, R. Pang, S. Ji, and T. Wang, "Graph backdoor," in *Proc. 30th USENIX Secur. Symp.*, Aug. 2021, pp. 1523–1540.

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

[17] T. A. Nguyen and A. T. Tran, "WaNet—Imperceptible warping-based backdoor attack," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16.

[18] 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. 16463–16472.

[19] H. Kwon, H. Yoon, and K.-W. Park, "Multi-targeted backdoor: Indentifying backdoor attack for multiple deep neural networks," *IEICE Trans. Inf. Syst.*, vol. 103, no. 4, pp. 883–887, 2020.

[20] M. Xue, S. Ni, Y. Wu, Y. Zhang, J. Wang, and W. Liu, "Imperceptible and multi-channel backdoor attack against deep neural networks," *Appl. Intell.*, vol. 54, pp. 1099–1116, Jan. 2024.

[21] X. Qi, T. Xie, Y. Li, S. Mahloujifar, and P. Mittal, "Revisiting the assumption of latent separability for backdoor defenses," in *Proc. 11th Int. Conf. Learn. Represent.*, 2022, pp. 1–20.

[22] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, "Clean-label backdoor attacks on video recognition models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14443–14452.

[23] N. G. Marchant, B. I. Rubinstein, and S. Alfeld, "Hard to forget: Poisoning attacks on certified machine unlearning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 7, pp. 7691–7700.

[24] M.-H. Van, W. Du, X. Wu, and A. Lu, "Poisoning attacks on fair machine," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, 2022, pp. 370–386.

[25] B. Zhao and Y. Lao, "CLPA: Clean-label poisoning availability attacks using generative adversarial nets," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 8, pp. 9162–9170.
[26] 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.
[27] 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*.
[28] 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.
[29] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 182–199.
[30] J. Lin, "Composite backdoor attack for deep neural network by mixing existing benign features," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 113–131.
[31] 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. 15074–15084.
[32] Y. Zeng, M. Pan, H. Just, L. Lyu, M. Qiu, and R. Jia, "Narcissus: A practical clean-label Backdoor attack with limited information," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2023, pp. 771–785.
[33] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent backdoor attacks on deep neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2019, pp. 2041–2055.
[34] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the trigger of backdoor attack," 2020, *arXiv:2004.04692*.
[35] 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.
[36] B. Wang et al., "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 707–723.
[37] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–19.
[38] 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.
[39] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, "Backdoor defense via decoupling the training process," in *Proc. 10th Int. Conf. Learn. Represent.*, 2022, pp. 1–25.
[40] Z. Wang, H. Ding, J. Zhai, and S. Ma, "Training with more confidence: Mitigating injected and natural backdoors during training," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 36396–36410.
[41] 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.
[42] Z. Zhang, Q. Liu, Z. Wang, Z. Lu, and Q. Hu, "Backdoor defense via deconfounded representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12228–12238.
[43] K. Gao, Y. Bai, J. Gu, Y. Yang, and S.-T. Xia, "Backdoor defense via adaptively splitting poisoned dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 4005–4014.
[44] C. Chen, H. Hong, T. Xiang, M. Xie, and J. Shao, "BAB: A novel algorithm for training clean model based on poisoned data," in *Proc. Workshop Artif. Intell. Saf.*, vol. 3381, 2023, pp. 1–10.
[45] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2654–2662.
[46] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[47] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, Univ. Toronto, Toronto, ON, Canada, 2009.
[48] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.
[49] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

Chen Chen is currently pursuing the master's degree with Zhejiang Gongshang University, China. His current research interests include adversarial examples and backdoor attacks.

Haibo Hong received the Ph.D. degree in cryptography from Beijing University of Posts and Telecommunications, China, in 2015. He is currently an Associate Professor with Zhejiang Gongshang University. His current research interests include information security and cryptography.

Tao Xiang (Senior Member, IEEE) is currently a Professor with Chongqing University. His current research interests include blockchain, privacy protection, AI security, multimedia security, cloud computing security, and big data security.

Mande Xie is currently a Professor with Zhejiang Gongshang University, China. His current research interests include network security and data privacy protection.