ADVERSARIAL FEATURE MAP PRUNING FOR BACKDOOR

*Dong Huang[1] , *Qingwen Bu[2,3] , Yuhao Qing[1], Yichao Fu[1], Heming Cui[1,2]
[1]University of Hong Kong, [2]Shanghai AI Laboratory, [3]Shanghai Jiao Tong University
{dhuang, yhqing, ycfu, heming}@cs.hku.hk, qwbu01@sjtu.edu.cn

ABSTRACT

Deep neural networks have been widely used in many critical applications, such as autonomous vehicles and medical diagnosis. However, their security is threatened by backdoor attacks, which are achieved by adding artificial patterns to specific training data. Existing defense strategies primarily focus on using reverse engineering to reproduce the backdoor trigger generated by attackers and subsequently repair the DNN model by adding the trigger into inputs and fine-tuning the model with ground-truth labels. However, once the trigger generated by the attackers is complex and invisible, the defender cannot reproduce the trigger successfully then the DNN model will not be repaired, as the trigger is not effectively removed.

In this work, we propose Adversarial Feature Map Pruning for Backdoor (FMP) to mitigate backdoor from the DNN. Unlike existing defense strategies, which focus on reproducing backdoor triggers, FMP attempts to prune backdoor feature maps, which are trained to extract backdoor information from inputs. After pruning these backdoor feature maps, FMP will fine-tune the model with a secure subset of training data. Our experiments demonstrate that, compared to existing defense strategies, FMP can effectively reduce the Attack Success Rate (ASR) even against the most complex and invisible attack triggers (e.g., FMP decreases the ASR to 2.86% in CIFAR10, which is 19.2% to 65.41% lower than baselines). Second, unlike conventional defense methods that tend to exhibit low robust accuracy (that is, the accuracy of the model on poisoned data), FMP achieves a higher RA, indicating its superiority in maintaining model performance while mitigating the effects of backdoor attacks (e.g., FMP obtains 87.40% RA in CIFAR10).

1 INTRODUCTION

Deep neural networks (DNNs) have become a cornerstone technology in numerous fields, such as computer vision Russakovsky et al. (2014), natural language processing Devlin et al. (2019), speech recognition Park et al. (2019), and several other applications Bojarski et al. (2016); Rajpurkar et al. (2017).The effective training of DNNs generally demands extensive datasets and considerable GPU resources to achieve SOTA performance. However, a substantial number of DNN developers may lack access to these resources, which subsequently leads them to rely on third-party services for training their models (e.g., Google Cloud Google (2023), AWS Amazon Web Services (2023), and Huawei Cloud Huawei Technologies Co. (2023)), acquiring datasets (e.g., DataTurks DataTurks (2023)), or directly downloading pre-trained models (e.g., Hugging Face Face (2023)).

Although using third-party services offers a practical solution for DNN developers, it simultaneously introduces potential security risks. Specifically, third-party may be involved in the data collection, model training, and pre-trained model distribution process, which may introduce malicious backdoor triggers Chen et al. (2017); Gu et al. (2017). For instance, once a developer uploads their dataset to a third-party service for training, the service provider could potentially revise the dataset by injecting poisoned samples containing hidden backdoor triggers. These poisoned samples, designed to blend in with the rest of the dataset, are then used in the training process, embedding the backdoor triggers into the model's architecture. The presence of backdoor triggers in DNNs can have serious

---

*These authors contributed equally to this work.

implications, particularly in security-critical systems where model integrity is crucial to preserving human life and safety Gu et al. (2017); Wang et al. (2019a).

To mitigate the backdoor threat, researchers have proposed a variety of defense mechanisms. Existing defense methods against backdoor attacks can be grouped into two main categories: detection methods and repair methods Wu et al. (2022); Li et al. (2020a). Detection methods rely on internal information from the model (e.g. neuron activation values Chen et al. (2018)) or model properties (e.g., performance metrics Chen et al. (2022); Zheng et al. (2022a)) to determine whether a model has been compromised by a backdoor attack, or whether a specific input example being processed by the model is a backdoor instance Zheng et al. (2022b). These detection techniques play a crucial role in identifying potential risks and raising awareness of possible security vulnerabilities. However, once a backdoor model has been detected, it still needs to be removed to restore its trustworthiness and reliability.

To tackle this challenge, researchers have introduced repairing methods that go beyond detection and aim to remove backdoor triggers from the compromised models. One notable example is Neural Cleanse (NC) Wang et al. (2019a), a technique that leverages reverse engineering to first reproduce the backdoor trigger. Once the trigger has been identified, NC injects it into the dataset along with its corresponding ground-truth labels, enabling the fine-tuning of the model to eliminate the backdoor trigger's impact. However, recent evaluation resultsChen et al. (2017); Li et al. (2020b); Nguyen & Tran (2021) reveal that NC and similar reverse engineering-based methods are predominantly successful in tackling simple backdoor triggers. In contrast, complex triggers (e.g. those involving intricate patterns or transformations) pose a greater challenge, making it difficult to reproduce and remove them. Consequently, models containing such sophisticated triggers may not be adequately repaired, leaving them susceptible to potential security breaches.

Recently, some researchers Liu et al. (2018); Zhao & Lao (2022) try to address this problem by analyzing the feature map's behavior to remove the backdoor from the model since they notice that backdoor-related feature maps exist, which demonstrate different characteristics from other normal feature maps. Specifically, feature maps are essential components of DNNs, responsible for extracting various features (e.g., color and texture information) from input samples Zeiler & Fergus (2013). The model then relies on these feature maps to extract features from the inputs and to make its final predictions. In the backdoor model, some backdoor feature maps extract backdoor information from the inputs Zhao & Lao (2022). Based on this observation, they believe that once the backdoor feature maps in the model are erased, the backdoor trigger will also be removed from the model. However, detecting backdoor feature maps from the model is challenging since the DNN developers, who may download pre-trained models from a third party, are unaware of the information about possible backdoor triggers injected in the model. Subsequently, they can not directly add triggers into the input samples to analyze and detect backdoor feature maps.

To address this challenge, we propose the Adversarial Feature Map Pruning for Backdoor (FMP), a novel approach that focuses on identifying and eliminating backdoor feature maps within the DNN. Instead of directly reproducing the backdoor trigger and then using it to detect backdoor feature maps in the model, FMP uses adversarial attack to reproduce the features extracted by each feature map. By adding these features into the inputs, we can feed these inputs into the model to identify the backdoor feature maps, which can then be mitigated to secure the DNN model. Our experiments reveal that initializing these feature maps and fine-tuning the model can effectively remove the backdoor from the model.

To validate the effectiveness of the FMP, we conducted extensive experimental evaluations on multiple benchmark datasets, including CIFAR-10, CIFAR-100 Krizhevsky & Hinton (2009), and GTSRB Stallkamp et al. (2012), under diverse attack scenarios. In our experiments, we considered various backdoor triggers with different complexities and compared the performance of FMP against several state-of-the-art baseline methods. We assessed the models using three primary metrics: Accuracy, Attack Success Rate (ASR), and Robust Accuracy (RA).

Our experimental results demonstrate that FMP consistently achieves lower ASR, higher accuracy, and improved RA compared to baseline methods across different datasets and attack scenarios. For instance, on the CIFAR-10 dataset, FMP outperformed baselines by achieving a significantly lower ASR (e.g., FMP decrease 19.2% ASR average compared with baselines in CIFAR10), and better

RA (e.g., FMP increase 16.92% RA on average in CIFAR10), indicating its effectiveness in removing backdoor triggers without compromising the model's performance on benign samples.

In a nutshell, we make the following contributions:

- We propose a novel defense strategy FMP to mitigate backdoor triggers from the model.
- We conduct extensive experiments to investigate the performance of FMP. The experiment results show that FMP can significantly outperform baseline strategies.
- We implement FMP into a tool that could help DNN developers to remove backdoor triggers from the DNN models, which is available in our Github[1].

2 RELATED WORK

This section discusses related work in two groups: backdoor attack and prior work backdoor defense.

2.1 BACKDOOR ATTACKS

According to the threat model, existing backdoor attack methods can be partitioned into two general categories, including data poisoning and training controllable.

Data poisoning attacks involve an attacker manipulating the training data. Existing methods in this category focus on designing different types of triggers to improve imperceptibility and attack effectiveness. These triggers can be classified based on various characteristics, such as visibility, locality, additivity, and sample specificity. Some attacks use visible triggers, such as BadNets Gu et al. (2017), which inject a square box into the inputs, while other methods use invisible triggers, like Blended Chen et al. (2017), to remain stealthier and harder to detect. Local triggers, as used in Label Consistent Attack Turner et al. (2019), only affect a small region of the input, while global triggers, like those in SIG Barni et al. (2019), have a more widespread impact. Additive triggers, such as those in Blended Chen et al. (2017), modify the input by adding a pattern, while non-additive triggers, like in LowFreq Zeng et al. (2021), involve more complex manipulations. Some attacks use sample-agnostic triggers, such as BadNets Gu et al. (2017), which apply the same trigger to all poisoned samples, whereas sample-specific triggers, like in SSBA Li et al. (2020b), tailor the trigger to each input.

Training controllable attacks involve an attacker having control over both the training process and the training data simultaneously, allowing them to learn the trigger and the model weights jointly for potentially more effective and stealthy backdoor attacks. Examples include Input-Aware Nguyen & Tran (2020), where the attacker adapts the backdoor trigger during the training process by considering the input distribution and the model's internal representation, making the backdoor trigger more difficult to detect and more robust against defense techniques, and WaNet Nguyen & Tran (2021), which learns the trigger and model weights jointly by optimizing a weighted combination of clean and poisoned samples, enabling the attacker to implant a more stealthy backdoor while maintaining a high attack success rate.

The above-mentioned backdoor attack poses a significant threat to the validity of DNN models, undermining their integrity and potentially causing misclassifications that can have severe consequences. To address this problem, we propose FMP, a novel defense mechanism designed to detect and mitigate the effects of backdoor attacks by identifying and removing backdoor feature maps in the DNN model.

2.2 PRIOR WORK ON BACKDOOR DEFENSES.

The research community has proposed a variety of defense mechanisms to detect and mitigate backdoor attacks in deep neural networks. These defenses can be broadly categorized into two groups: data-centric and model-centric approaches.

Data-centric defenses primarily focus on detecting and cleansing poisoned data points in the training dataset to prevent the model from learning the backdoor trigger. One such technique is the AC

---

[1] For double-blind, we upload the code into supplementary files

proposed by Chen et al. (2018), which identifies and removes poisoned data points by clustering activations of the penultimate layer. However, once the DNN developers rely on the third party to train the DNN model, they can not use these strategies to remove backdoor samples since they can not interfere with the third-party platforms.

Model-centric defenses, on the other hand, focus on analyzing and modifying the trained model itself to remove the backdoor behavior. NC Wang et al. (2019a) is a model-centric defense that identifies and neutralizes backdoor triggers by computing anomaly scores based on the L1-norm of adversarial perturbations required to reverse engineer the triggers. However, once the trigger become complex and invisible, DNN developers can not use NC to reproduce the triggers and can not remove the trigger from the model successfully. The Fine-Pruning Liu et al. (2018) approach, as mentioned earlier, removes backdoors by pruning redundant feature maps less useful for normal classification. Another model-centric defense is NAD Li et al. (2021a), which identifies backdoor neurons by analyzing the attribution of each neuron's output with respect to the input features. However, without the backdoor samples, or poison samples from the injected datasets, they can not correctly analyze the backdoor model's internal information, which causes them to have lower effectiveness to remove the backdoor trigger from the model. Recently, some other papers Wu & Wang (2021); Barni et al. (2019); Guo et al. (2021); Li et al. (2023); Chen et al. (2019); Guo et al. (2019); Liu et al. (2019); Fu et al. (2020) are also proposed to defense backdoor for deep neural networks.

3 METHODOLOGY

Notations For ease of discussion, this section defines the following notations for DNNs and feature maps: f_θ is a DNN parameterized by θ, and there are N feature maps $\sum_i^N f_\theta^i$ in the model. The i-th feature map in the DNN is denoted as f_θ^i. $f_\theta^i(x)$ denotes the feature map i-th output when the DNN input is x. The x' means the input x added the feature generated by the corresponding feature map.

3.1 MOTIVATION

The primary goal of our pruning defense strategy is to detect backdoor feature maps in a DNN model. As previously mentioned, intuitively, we should use backdoor samples and clean samples fed into the model to detect these backdoor feature maps. However, since DNN developers do not have access to backdoor samples, they cannot directly detect backdoor feature maps using this approach. This limitation calls for a defense mechanism that can operate effectively without the need for backdoor samples.

The main idea behind our proposed defense mechanism is to generate potential poison samples by reversing the specific features associated with each feature map. Since each feature map in the model is used to extract features from the DNN model, and in the injected model, there are some backdoor feature maps that are used to extract backdoor information for the poison samples. By feeding these potential poison samples into the model, we can observe the inference accuracy for each feature map. Since backdoor feature maps are designed to extract backdoor information from the inputs, causing the classifier to return the target label regardless of the true label, we hypothesize that potential poison samples generated by backdoor feature maps will lead to a significant change in inference accuracy when processing their corresponding potential poison samples.

In summary, the motivation behind our defense strategy is to detect backdoor feature maps without backdoor samples by generating potential poison samples and analyzing the inference accuracies with these samples. This enables us to identify backdoor feature maps, which can be mitigated to secure the DNN model. With this motivation in mind, we propose Adversarial Feature Map Pruning for Backdoor (FMP) as the defense mechanism to achieve this objective.

3.2 FEATURE REVERSE GENERATION

In this section, we employ reverse engineering to generate features that will be extracted by each feature map in the DNN. These features will be added to the inputs, which will then be used to detect backdoor feature maps. The detailed implementation of the Feature Reverse Generation is provided in Alg.1. Specifically, for each feature map f_θ^i in the DNN model, our objective is to generate the corresponding features that the feature map is intended to extract by adversarial feature map attack. To accomplish this, we use a reverse engineering approach that maximizes the difference between the original input x and the perturbed input x' in the feature map space. In Alg.1, referred to as FRG, we have adapted the FGSM attack Goodfellow et al. (2014) to operate at the feature map levels. The overall time required by this method can be expressed as follows:

$$T = L \cdot forward + N_i \cdot backward$$

where L represents the total number of layers in the Deep Neural Network (DNN), and N_i denotes the number of feature maps in the i-th layer. The term "forward" corresponds to the time taken for one forward step, assuming each forward pass takes an equal amount of time. Similarly, the term "backward" corresponds to the time taken for one backward step, assuming each backward pass also takes an equal amount of time[2].

Algorithm 1: Feature Reverse Generation

```
input :  $f_\theta$: DNN model; $(\mathcal{X}, \mathcal{Y})$: Clean Dataset; $\epsilon$: maximum allowed perturbation; $p$: First
         $N/p$ feature maps wll be pruned.
output:  List: Backdoor feature map list.
Function Inference($f_\theta, (\mathcal{X}, \mathcal{Y}), \epsilon, p$):
    Initialize an empty list Correct_list;
    Start traversal over all feature maps: ;
    for $i$ in range($N$) do
        Initialize correct counter to 0;
        for $x, y$ in $(\mathcal{X}, \mathcal{Y})$ do
            $x'$ = FRG($f_\theta^i, x, steps, \alpha, \epsilon$);
            $y' = f_\theta(x')$;
            if $y = y'$ then
                correct += 1;
        Append correct to Correct_list;
    Sort Correct_list in ascending order;
    return First $N/p$ elements from Correct_list ;
    # We suppose feature maps with lower FRG accuracy are backdoor-related feature maps.
Function FRG($f_\theta^i, x, \epsilon$):
    Initialize: $x' = x + 1e - 4$ ;
    Calculate the loss: $loss = \|f_\theta^i(x) - f_\theta^i(x')\|^2$;
    Calculate the gradient: $grad = \partial loss/\partial x$;
    Update: $x' = x + \epsilon \cdot \text{sign}(grad)$;
    Apply image bounds: $x' = \text{clamp}(x', \text{min} = 0, \text{max} = 1)$;
    return $x'$;
```

3.3 REMOVE BACKDOOR FROM THE DNN

After obtaining the reversed samples using the method described in Alg.1, we feed these potential poison samples into the model to obtain feature map inference accuracy. Then, based on the accuracy list, we will return the potential backdoor injected feature maps. After detecting the backdoor-related feature maps, our next step is to remove the backdoor from the DNN. To achieve this, we follow a two-step process: initialization and fine-tuning.

Initialization In this step, we initialize the weights corresponding to the backdoor feature maps identified in the previous stage. To perform the initialization, we first set the weights of the backdoor feature maps to zero. This process neutralizes the contribution of these feature maps to the overall model output. By resetting the weights associated with these feature maps, we effectively remove the backdoor information that the feature maps were designed to extract. In addition to resetting the weights, we also adjust the biases of the backdoor feature maps to ensure that the output of these feature maps remains close to zero during the forward pass. This initialization process helps

---

[2] We also provide a PGD version for FRG in the GitHub Rep, but it will need more time to execute.

to neutralize the impact of the malicious backdoor on the model's predictions, thus reducing the risk of targeted misclassification.

Fine-Tuning Once the backdoor feature maps have been initialized, we proceed to fine-tune the entire DNN model. Fine-tuning involves updating the model weights using a smaller learning rate and a subset of clean, accurately labeled samples. This step allows the model to adapt and learn from the new initialization while preserving its ability to perform well on clean samples. By fine-tuning the model, we ensure that it retains its predictive capabilities on legitimate inputs and remains robust against potential attacks.

4 EVALUATION

Experimental Setting In this work, we use BackdoorBench Wu et al. (2022) as a benchmark to evaluate the performance of FMP. Since most of the existing backdoor-related literature focused on image classification tasks Chen et al. (2017; 2022); Devlin et al. (2019); Zheng et al. (2022b); Zhao & Lao (2022); Wang et al. (2019b); Tran et al. (2018); Nguyen & Tran (2021), we followed existing research to the choice of CIFAR10, CIFAR100 Krizhevsky & Hinton (2009), and GTSRB Stallkamp et al. (2012) datasets to evaluate FMP's performance. Similarly to baselines, we chose ResNet18 He et al. (2015) as our evaluation model, as it has been systematically evaluated by BackdoorBench Wu et al. (2022), widely used by baseline approaches, and is also widely used by vision tasks (e.g., image classification He et al. (2015), object detection Ren et al. (2015)), so we believe evaluating FMP in these data sets can provide a comprehensive and meaningful assessment of its performance. In our experiments, we selected five state-of-the-art (SOTA) backdoor attack strategies, which are systematically evaluated and implemented by BackdoorBench, as baselines to evaluate the effectiveness of our defense strategy. These attack strategies consist of BadNets Gu et al. (2017), Blended Chen et al. (2017), Low Frequency Zeng et al. (2021), SSBA Li et al. (2020b), and WaNet Nguyen & Tran (2021). We train the CIFAR10 and CIFAR100 datasets with 100 epochs, SGD momentum of 0.9, learning rate of 0.01, and batch size of 128, using the CosineAnnealingLR scheduler. The GTSRB dataset is trained with 50 epochs. We set the poisoning rate to 10% by default. To ensure fair comparisons, we adhere to the default configuration in the original papers, including trigger patterns, trigger sizes, and target labels.

Defense setup BackdoorBench has implemented many effective defense strategies Zhao & Lao (2022); Liu et al. (2018); Wang et al. (2019a); Li et al. (2021a); Tran et al. (2018) are implemented by BackdoorBench, we believe that taking these defense strategies as our baseline can demonstrate FMP's effectiveness. However, some of these defense strategies require DNN developers to have access to the backdoor trigger (e.g., AC Chen et al. (2018), Spectral Tran et al. (2018), ABL Li et al. (2021b), D-BR Chen et al. (2022) and DDE Zheng et al. (2022b)), or to modify the model training process (e.g., ABL Li et al. (2021b), DBD Huang et al. (2022)) which is not possible in our defense scenarios. Therefore, we exclude these strategies from our baseline comparison.

Finally, we select six widely used defense strategies that align with our defense goals: *Fine-tuning (FT)* retrains the backdoor model with a subset clean dataset to remove the backdoor trigger's effects. *Fine-pruning (FP)* Liu et al. (2018) prunes backdoor feature maps to remove backdoors from the model. *Adversarial Neuron Pruning (ANP)* Wu & Wang (2021) selectively prunes neurons associated with the backdoor trigger while maintaining performance. *Channel Lipschitz Pruning (CLP)* Zhao & Lao (2022) is a data-free strategy that analyzes the Lipschitz constants of the feature maps to identify potential backdoor-related feature maps. *Neural Attention Distillation (NAD)* Li et al. (2021a) leverages attention transfer to erase backdoor triggers from deep neural networks, ensuring that the model focuses on the relevant neurons for classification. *Neural Cleanse (NC)* identifies and mitigates backdoor attacks in neural networks by analyzing the internal structure of the model and identifying patterns that are significantly different from the norm, which may indicate the presence of a backdoor.

In order to repair the model, we adopt a configuration consisting of 10 epochs, a batch size of 256, and a learning rate of 0.01. The CosineAnnealingLR scheduler is employed alongside the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 for the client optimizer. We set the default ratio of the retraining data set at 10% to ensure a fair and consistent evaluation of defense strategies. For the CLP, we configure the Lipschitz Constant threshold (u) to be 3, the pruning step to

Backdoor	BadNets			Blended			Low Frequency			SSBA			WaNet			AVG		
Attack	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc↑	ASR↓	RA↑
CIFAR10																		
Benign	91.94	97.21	-	93.44	99.95	-	93.44	99.39	-	92.94	98.80	-	91.53	98.83	-	92.65	98.84	-
FT	93.35	**1.51**	**92.46**	93.54	95.63	4.16	93.41	77.89	20.67	93.28	71.57	26.52	94.12	22.30	74.53	**93.54**	53.78	43.67
FP	91.62	16.06	79.24	93.21	96.84	3.04	93.24	96.73	2.79	92.88	82.92	16.00	91.30	**0.62**	84.91	92.45	58.63	37.20
CLP	91.20	94.77	5.01	93.29	99.83	0.17	92.47	99.10	0.82	88.27	9.96	76.22	89.03	53.84	41.88	90.85	71.5	24.82
ANP	91.22	73.36	26.16	93.25	99.44	0.56	93.19	98.03	1.88	92.92	68.59	29.13	90.81	1.93	88.98	92.28	68.27	29.34
NC	89.11	1.32	89.17	93.23	99.90	0.10	90.67	2.26	86.24	90.33	**0.53**	86.72	90.54	86.91	12.38	90.78	38.18	54.92
NAD	92.97	1.10	92.39	92.18	44.98	42.60	92.32	13.43	77.03	92.22	38.18	57.18	94.16	12.61	83.22	92.77	22.06	70.48
Our	91.67	1.67	91.71	91.85	**6.44**	**74.43**	91.77	**1.90**	**90.52**	91.92	2.89	**88.59**	93.42	1.38	**92.16**	92.13	**2.86**	**87.40**
CIFAR100																		
Benign	67.33	93.61	-	70.19	99.84	-	69.56	97.47	-	69.24	98.12	-	65.67	97.50	-	68.39	97.31	-
FT	69.90	1.38	67.18	70.32	89.36	6.57	69.93	45.90	35.95	69.30	58.14	28.45	71.16	6.41	63.64	**70.12**	40.23	40.35
FP	67.59	23.09	54.04	69.44	93.56	4.03	68.73	82.60	10.84	68.25	76.46	14.52	66.33	81.47	11.26	68.06	71.44	18.94
CLP	59.74	74.79	19.42	68.56	99.52	0.37	65.46	92.88	5.18	68.40	89.57	7.44	60.06	6.93	54.53	64.44	72.74	17.38
ANP	66.95	15.36	58.05	70.05	97.70	1.70	69.47	55.00	32.00	68.92	89.87	7.54	64.23	0.23	60.55	67.92	51.63	31.97
NC	64.67	0.10	64.38	64.16	1.14	34.63	65.25	2.28	53.91	65.16	2.17	56.86	67.09	1.29	62.92	65.27	1.39	54.54
NAD	68.84	0.31	**67.63**	69.24	81.07	10.16	69.33	31.78	44.41	68.39	30.82	42.24	71.57	17.41	57.72	69.47	32.27	44.43
Our	66.25	**0.09**	67.23	67.27	**0.41**	**40.49**	66.32	**0.18**	**65.83**	66.77	**0.39**	**60.61**	68.79	**0.18**	**66.84**	67.08	**0.25**	**60.20**
GTSRB																		
Benign	98.04	96.38	-	98.19	100.00	-	98.21	99.96	-	97.73	99.72	-	98.66	97.59	-	98.17	98.72	-
FT	98.60	0.49	98.11	98.19	99.61	0.34	98.39	92.94	5.43	97.63	96.75	2.99	99.20	0.73	98.31	98.40	58.10	41.03
FP	97.79	0.04	65.20	98.04	100.00	0.00	97.81	99.25	0.45	96.90	99.64	0.34	98.80	0.05	13.23	97.87	59.79	15.84
CLP	96.42	88.11	11.67	97.68	100.00	0.00	97.09	97.74	1.82	97.13	99.42	0.57	62.17	99.91	0.08	90.10	97.03	2.83
ANP	96.55	15.71	82.06	98.22	99.96	0.02	98.27	69.20	27.66	97.10	99.51	0.48	98.67	1.65	96.32	97.76	57.20	41.30
NC	97.75	0.00	97.75	96.34	2.47	53.58	97.72	7.29	81.19	96.94	3.70	88.40	98.39	0.00	97.29	97.43	2.69	83.64
NAD	98.66	0.03	98.64	98.39	96.98	2.86	98.54	80.88	14.91	97.72	94.70	4.93	98.98	0.16	98.66	**98.45**	54.55	44.00
Our	98.60	**0.00**	**98.66**	90.36	**1.07**	**64.17**	90.01	**0.02**	**95.58**	97.41	**0.51**	**89.87**	99.05	**0.00**	**98.93**	95.08	**0.32**	**89.44**

Table 1: Performance comparison (%) of backdoor defense methods on CIFAR10, CIFAR100, and GTSRB datasets under PreActResNet18, under different attack strategies with a poison rate of 10% and retraining data ratio of 100%. We set the ϵ to 1/255, and the p is set to 64.

0.05, and the maximum pruning rate to 0.9, which is consistent with BackdoorBench and its original paper default settings. In the case of ANP, we optimize all masks using Stochastic Gradient Descent (SGD) with a perturbation budget (i.e., ϵ) of 0.4. For FMP, we set p equal to 64 and ϵ as 1/255. All other configurations are maintained as specified in the respective original publications to guarantee a rigorous comparison of the defense strategies.

Evaluation Metrics To assess the effectiveness of our proposed defense strategy and compare it with the SOTA baselines, we employ three key evaluation metrics: clean accuracy (Acc), attack success rate (ASR), and Robust Accuracy (RA). Acc is used to measure the performance of the defense strategy on clean inputs. High clean accuracy indicates that the model's performance on its original task has not been compromised due to defense. ASR evaluates the effectiveness of the defense strategy in mitigating backdoor attacks by measuring the rate at which backdoor triggers succeed in causing misclassification.

4.1 EFFECTIVENESS

The evaluation results are shown in Tab. 1. We can observe that for the CIFAR10 dataset, FMP consistently achieves low Attack Success Rate (ASR) values while maintaining high Robust Accuracy (RA). Specifically, FMP successfully reduces the average ASR to 2. 86% on average, reducing the 19. 2% ASR in the CIFAR10 dataset, and the RA is also higher than the baseline strategies. For instance, the RA of the proposed method reaches 87. 40%, which is around 16.92% higher than the average RA achieved by the baseline approaches. This demonstrates that our method can effectively

Poison	BadNets			Blended			Low Frequency			SSBA			WaNet		
Rate (%)	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA
0.1	92.09	1.71	91.72	91.83	8.53	75.90	91.74	2.09	91.00	92.17	2.23	89.60	93.19	1.48	91.91
0.5	92.01	0.99	91.74	91.95	8.31	75.30	91.92	2.14	91.21	91.90	1.42	91.92	93.32	1.45	91.94
1	92.22	1.78	91.48	91.66	8.49	74.49	92.06	2.08	91.00	91.85	2.08	89.61	93.46	1.32	92.22
5	92.59	1.24	90.39	93.45	8.31	74.47	93.32	2.23	90.07	93.10	2.93	88.49	91.86	1.09	91.90
10	91.67	1.67	91.71	91.85	6.44	74.43	91.77	1.90	90.52	91.92	2.89	88.59	93.42	1.38	92.16

Table 2: FMP's effectiveness under different poison rates.

Retraining	BadNets			Blended			Low Frequency			SSBA			WaNet		
ratio (%)	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA
5	86.57	1.77	86.08	90.59	7.63	63.40	88.07	4.60	81.00	87.26	3.50	79.77	89.31	1.54	88.07
10	91.67	1.67	91.71	91.85	6.44	74.43	91.77	1.90	90.52	91.92	2.89	88.59	93.42	1.38	92.16
15	91.71	1.66	91.30	91.88	6.37	74.61	91.60	1.88	90.63	91.95	2.87	88.64	92.47	1.42	91.32
20	91.83	1.47	91.91	91.92	6.08	74.87	91.74	1.90	90.73	91.73	2.88	88.91	92.54	1.44	91.33
100	92.02	0.9	91.04	92.12	5.31	75.77	91.24	1.71	91.07	92.31	2.67	89.37	92.95	1.37	91.42

Table 3: FMP's effectiveness under different retraining data ratios.

mitigate backdoor attacks while increasing the model's performance on poison data, highlighting its practical utility in real-world scenarios.

We can also observe that standard fine-tuning (FT) shows promising defense results against several attacks, such as BadNets, where it achieves an ASR of 1.51% and an RA of 92.46%. However, it fails to generalize to more complex attacks, like the Blended attack, where the ASR increases to 95.63%, and the RA drops to 4.16%. Neural Attention Distillation (NAD) exhibits similar behavior in terms of generalization, with its performance varying considerably across different attacks. For example, NAD achieves an ASR of 1.10% and an RA of 92.39% against BadNets, but its performance drops significantly when faced with the Blended attack, resulting in an ASR of 44.98% and an RA of 42.60%.

For pruning-based methods, for example, FP, Adversarial Neuron Pruning (ANP), and Channel Lipschitz Pruning (CLP), these approaches demonstrate varying levels of effectiveness against different attacks. However, none of them consistently outperform our method FMP in all types of attacks. For example, FP achieves an ASR of 16.06% and an RA of 79.24% against BadNets, while ANP achieves an ASR of 73.36% and an RA of 26.16% for the same attack when it only has a limited fine-tune dataset (10%) and within limited fune-tune epochs (i.e., 10 epochs). The key reason for FMP obtains higher performance compared with these pruning methods because FMP feature map level, aligns with the backdoor trigger's focus on the DNN feature map, rather than specific neurons within the DNN (e.g., ANP).

For the reverse engineering-based method (i.e., NC), we can observe that when the backdoor triggers are simple and easy to reproduce, NC will have high performance, e.g., NC achieves an ASR of 1.32% and an RA of 89.17% against BadNets. However, when the backdoor triggers are complex and invisible, the performance of NC significantly deteriorates, indicating its limited effectiveness in handling sophisticated backdoor attacks. For instance, when faced with the Blended attack, where the backdoor trigger is more intricate and harder to detect, NC's ASR increases to an alarming 99.90%, and its RA plummets to a mere 0.10%.

For the CIFAR100 and GTSRB datasets, we can observe that first, FMP is also better than the baseline defense strategies. Specifically, FMP obtains the average 0. 25% ASR and the average 60.20% RA in CIFAR100, which decreases the average of 1. 14% ASR compared to the best-performing baseline approach. Furthermore, FMP also obtains a mean ASR of 0.32% and an average RA of 89.44% in GTSRB, yielding about 5.80% RA improvement compared to the baseline approaches. Furthermore, it should be noted that baseline methods may exhibit inconsistent performance results. As an illustrative instance, the NC technique demonstrates an achieved ASR of 1.14%, coupled with an RA of 34.63% when pitted against the BadNets on CIFAR100. However, its efficacy witnesses a severe decline when confronted with the same attack but a different dataset (i.e., CIFAR10). In

ϵ	1/255	2/255	4/255	8/255	16/255
Acc	91.67	91.04	90.28	89.21	88.01
ASR	1.67	1.22	1.51	2.29	1.41
RA	91.71	90.79	89.49	88.03	87.82

Table 4: FMP's effectiveness under different ϵ in CIFAR10 dataset under BadNets attack.

p	2	4	8	16	32	64
Acc	65.57	85.42	87.65	88.95	89.96	91.67
ASR	3.98	1.40	1.3	1.28	1.56	1.67
RA	65.52	85.49	87.63	88.78	90.34	91.71

Table 5: FMP's effectiveness under different p in CIFAR10 dataset under BadNets attack.

contrast, the FMP approach consistently attains superior results, manifesting in consistently lower ASR values, exemplified by 2.86%, 0.25% and 0.32% average ASR in CIFAR10, CIFAR100, and GTSRB, respectively.

Effect of Poison Data Rate The poison rate, referring to the proportion of poisoned samples in the training dataset, plays a crucial role in the results of the backdoor trigger injection. We conducted experiments with different poison rates (from 0.1% to 10%) to explore their impact on FMP's effectiveness. The results, shown in Tab.2, indicate that FMP demonstrates consistent performance across different poison rates and effectively mitigates backdoor attacks. For example, considering the BadNets attack, the ASR changes slightly within 0.99% to 1.78% as the poisoning rate increases from 0.1% to 10%. This trend is also observed for other attack strategies. Although a higher poison rate can be expected to lead to a higher ASR, our experimental results show that this is not true. When the poisoning rate is very low, it becomes more challenging for defense strategies to detect the backdoor trigger from the model due to its subtle influence. As the poison rate increases, the backdoor trigger has a more noticeable impact on the model, which can be detected and mitigated more easily by the defense strategy. Our experimental results emphasize the importance of designing defense strategies capable of detecting and mitigating backdoor attacks, even when dealing with subtle influences caused by low poison rates.

Effectiveness under Different Percentages of Clean Data We are also interested in studying the correlation between the performance of FMP and the amount of available training data, which will be used to repair the model to mitigate backdoor triggers. We compare four different retraining data ratios:5%, 10%, 15%, 20%, and 100%, and the results of our FMP are demonstrated in Tab.3. We observe that the performance of our defense strategy improves as the amount of clean training data increases. For example, when the retraining ratio increases from 5% to 100%, the ASR for BadNets decreases from 1.77% to 0.9%, while the model accuracy (Acc) improves from 86.57% to 92.02% and the Robust Accuracy (RA) increases from 86.08% to 91.04%. Similar trends can be observed for other attack strategies such as Blended, Low Frequency, SSBA, and WaNet. This indicates that our defense strategy becomes more effective in mitigating backdoor attacks as more clean data are available to retrain the model. However, it should be noted that even with a small amount of clean data (e.g., 5%), our defense strategy still exhibits competitively good performance in mitigating backdoor attacks. For example, with a 5% retraining ratio, the ASR for WaNet is 1.54%, while the Acc and RA are 89.31% and 88.07%, respectively.

Effectiveness under Different ϵ and p We further investigate the effectiveness of FMP under different ϵ and p, as listed in Tab.4 and Tab.5. We can first observe that with different ϵ, the effectiveness of FMP is consistently satisfactory. Upon increasing the ϵ, the Acc exhibits marginal decline, underscoring FMP's resilience across varying ϵ values. Subsequently, when varying the parameter p for backdoor feature pruning, a notable decrease is observed in both accuracy (Acc) and robust accuracy (RA). This reduction can be attributed to the model's failure to successfully finetune after 50% of the information is pruned along with 50% of the feature maps, hampering its performance. FMP can successfully execute to mitigate the backdoor from the model with a p larger than 4.

5 CONCLUSION

In this paper, we presented a novel defense strategy, FMP, to effectively detect and mitigate backdoor attacks in DNNs. Our approach focuses on identifying and eliminating backdoor feature maps within the DNN model. Through extensive experiments, we demonstrated the effectiveness of our FMP defense strategy against a variety of backdoor attack methods, outperforming existing state-of-the-art defense techniques in terms of attack success rate reduction and stability against different datasets and attack methods.

6 ACKNOWLEDGEMENT

The work is supported in part by National Key R&D Program of China (2022ZD0160200), HK RIF (R7030-22), HK ITF (GHP/169/20SZ), the Huawei Flagship Research Grants in 2021 and 2023, and HK RGC GRF (Ref: HKU 17208223), the HKU-SCF FinTech AcademyR&D Funding Schemes in 2021 and 2022, and the Shanghai Artificial Intelligence Laboratory (Heming Cui is a courtesy researcher in this lab).

REFERENCES

Inc. Amazon Web Services. Amazon web services (aws), 2023. URL `https://aws.amazon.com/`. Accessed: 2023-05-02.

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

Mariusz Bojarski, David W. del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *ArXiv*, abs/1604.07316, 2016.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Ben Edwards, Taesung Lee, Ian Molloy, and B. Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *ArXiv*, abs/1811.03728, 2018.

Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *International Joint Conference on Artificial Intelligence*, 2019. URL `https://api.semanticscholar.org/CorpusID:199466093`.

Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. *Advances in Neural Information Processing Systems*, 35:9727–9737, 2022.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Xiaodong Song. Targeted backdoor attacks on deep learning systems using data poisoning. *ArXiv*, abs/1712.05526, 2017.

DataTurks. Dataturks: Data annotations made super easy, 2023. URL `https://dataturks.com/`. Accessed: 2023-05-02.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Hugging Face. Hugging face: The ai community building the future, 2023. URL `https://huggingface.co/`. Accessed: 2023-05-02.

Hao Fu, Akshaj Kumar Veldanda, Prashanth Krishnamurthy, Siddharth Garg, and Farshad Khorrami. Detecting backdoors in neural networks using novel feature-based anomaly detection. *ArXiv*, abs/2011.02526, 2020.

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

Google. Google cloud, 2023. URL `https://cloud.google.com/`. Accessed: 2023-05-02.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *ArXiv*, abs/1708.06733, 2017.

Junfeng Guo, Ang Li, and Cong Liu. Aeva: Black-box backdoor detection using adversarial extreme value analysis. *ArXiv*, abs/2110.14880, 2021. URL `https://api.semanticscholar.org/CorpusID:240070408`.

Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Xiaodong Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *ArXiv*, abs/1908.01763, 2019. URL `https://api.semanticscholar.org/CorpusID:199452956`.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. *ArXiv*, abs/2202.03423, 2022.

Ltd. Huawei Technologies Co. Huawei cloud, 2023. URL `https://www.huaweicloud.com/`. Accessed: 2023-05-02.

Alex Krizhevsky and Geoffrey Hinton. Cifar-10 and cifar100 (canadian institute for advanced research), 2009. URL `https://www.cs.toronto.edu/~kriz/cifar.html`. Accessed: 2023-05-02.

Yige Li, Nodens Koren, L. Lyu, Xixiang Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *ArXiv*, abs/2101.05930, 2021a.

Yige Li, Xixiang Lyu, Nodens Koren, L. Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *Neural Information Processing Systems*, 2021b.

Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, L. Lyu, Bo Li, and Yugang Jiang. Reconstructive neuron pruning for backdoor defense. In *International Conference on Machine Learning*, 2023. URL `https://api.semanticscholar.org/CorpusID:258865980`.

Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Backdoor learning: A survey. *IEEE transactions on neural networks and learning systems*, PP, 2020a.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16443–16452, 2020b.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Recent Advances in Intrusion Detection*, 2018.

Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and X. Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019. URL `https://api.semanticscholar.org/CorpusID:204746801`.

A. Nguyen and A. Tran. Input-aware dynamic backdoor attack. *ArXiv*, abs/2010.08138, 2020.

A. Nguyen and A. Tran. Wanet - imperceptible warping-based backdoor attack. *ArXiv*, abs/2102.10369, 2021.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *ArXiv*, abs/1904.08779, 2019.

Pranav Rajpurkar, Jeremy A. Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Yi Ding, Aarti Bagul, C. Langlotz, Katie S. Shpanskaya, Matthew P. Lungren, and A. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *ArXiv*, abs/1711.05225, 2017.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2014.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), 2012. URL `http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset`. Accessed: 2023-05-02.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Neural Information Processing Systems*, 2018.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *ArXiv*, abs/1912.02771, 2019.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 707–723, 2019a.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 707–723, 2019b.

Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, Chaoxiao Shen, and Hongyuan Zha. Backdoorbench: A comprehensive benchmark of backdoor learning. *ArXiv*, abs/2206.12654, 2022.

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *ArXiv*, abs/2110.14430, 2021.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2013.

Yi Zeng, Won Park, Zhuoqing Morley Mao, and R. Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16453–16461, 2021.

Bingyin Zhao and Yingjie Lao. Clpa: Clean-label poisoning availability attacks using generative adversarial nets. In *AAAI Conference on Artificial Intelligence*, 2022.

Runkai Zheng, Rong Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *European Conference on Computer Vision*, 2022a.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Pre-activation distributions expose backdoor neurons. *Advances in Neural Information Processing Systems*, 35:18667–18680, 2022b.

Backdoor	BadNets			Blended			Low Frequency			SSBA			WaNet		
Attack	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA
CIFAR10															
Benign	91.94	97.21	-	93.44	99.95	-	93.44	99.39	-	92.94	98.80	-	91.53	98.83	-
ANP	91.22	73.36	26.16	93.25	99.44	0.56	93.19	98.03	1.88	92.92	68.59	29.13	90.81	1.93	88.98
AEVA	91.05	50.96	47.53	92.28	59.37	38.66	93.05	59.81	36.38	92.29	67.56	26.01	90.26	6.54	90.59
RNP	90.55	55.01	36.46	92.29	55.59	42.15	92.41	58.71	40.1	91.94	61.24	30.6	90.22	18.15	72.95
FMP	91.67	**1.67**	**91.71**	91.85	**6.44**	**74.43**	91.77	**1.90**	**90.52**	91.92	**2.89**	**88.59**	93.42	**1.38**	**88.98**

Table 6: Performance comparison (%) of adversarial example related backdoor defense methods on CIFAR10 under PreActResNet18, under different attack strategies with a poison rate of 10% and retraining data ratio of 100%. We set the ϵ to 1/255, and the p is set to 64.

Backdoor	BadNets			Blended			Low Frequency			SSBA			WaNet		
Attack	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA	Acc	ASR	RA
CIFAR10															
Benign	91.94	97.21	-	93.44	99.95	-	93.44	99.39	-	92.94	98.80	-	91.53	98.83	-
DeepInspect	90.51	15.87	64.44	90.89	3.5	77.16	90.83	4.7	77.51	90.05	10.57	73.83	90.31	5.97	77.94
TABOR	90.78	9.19	79.02	90.78	11.13	78.22	90.14	5.48	76.44	90.03	13.09	76.71	90.95	7.36	78.29
ABS	90.78	5.61	77.34	90.95	14.44	80.33	90.93	10.43	88.32	90.95	17.12	77.45	90.6	13.04	77.6
Fu et al. (2020)	90.7	11.22	76.72	90.28	9.37	75.63	90.46	9.62	71.87	90.06	6.74	77.43	90.66	14.98	69.13
FMP	91.67	**1.67**	**91.71**	91.85	**6.44**	**74.43**	91.77	**1.90**	**90.52**	91.92	**2.89**	**88.59**	93.42	**1.38**	**88.98**

Table 7: Performance comparison (%) of other SOTA backdoor defense methods on CIFAR10 under PreActResNet18, under different attack strategies with a poison rate of 10% and retraining data ratio of 100%. We set the ϵ to 1/255, and the p is set to 64.

A APPENDIX

A.1 COMPARISON WITH OTHER BASELINES

To illustrate the effectiveness of FMP with other adversarial example related strategies (e.g., ANP Wu & Wang (2021), AEVA Guo et al. (2021), and RNP Li et al. (2023)). The evaluation results are shown in Tab. 6. We can find that FMP obtain SOTA performance compared with these baselines. For example, in BadNets, FMP decreases the ASR from 50.96% to 1.67%, and FMP also increases the RA from 47.53% to 91.71% in BadNets. The primary reason for this is FMP's focus on the feature map level, which aligns with the backdoor trigger's emphasis on the DNN feature map, rather than on specific neurons within the DNN (e.g., as in ANP). Secondly, the pruning at the feature map level enables FMP to rapidly eliminate the backdoor trigger from the model. In contrast, neuron-level pruning typically requires significant overhead due to the need for repeated prune-finetune-evaluation cycles. FMP is particularly advantageous in scenarios with limited computational resources, where developers may not have the capacity to extensively evaluate and remove backdoors from the model. This limitation results in strategies like ANP, AEVA, and RNP exhibiting higher ASR compared to FMP in our setup.

To illustrate the effectiveness of FMP with other SOTA backdoor defense strategies (e.g., DeepInspect Wu & Wang (2021), TABOR Guo et al. (2021), ABS Li et al. (2023) and Fu et al. (2020)). The evaluation results are shown in Tab. 7. We can find that FMP obtain SOTA performance compared with these baselines. For example, in BadNets, FMP decreases the ASR from 5.61% to 1.67%, and FMP also increases the RA from 79.02% to 91.71% in BadNets. As discussed before, the primary reason for this is the pruning at the feature map level enables FMP to efficiently and rapidly eliminate the backdoor trigger from the model. While other defense strategies will not obtain the SOTA performance due to the limited computational resources.

B ABLATION STUDY ILLUSTRATION

In this section, we illustrate the figure illustration for the tables illustrated in Tab.3 to Tab.5.

Effect of Poison Data Rate The poison rate, referring to the proportion of poisoned samples in the training dataset, plays a crucial role in the results of the backdoor trigger injection. We conducted experiments with different poison rates (from 0.1% to 10%) to explore their impact on FMP's effectiveness. The results, shown in Fig.1, indicate that FMP demonstrates consistent performance across different poison rates and effectively mitigates backdoor attacks. For example, considering the BadNets attack, the ASR changes slightly within 0.99% to 1.78% as the poisoning rate increases from 0.1% to 10%. This trend is also observed for other attack strategies. Although a higher poison rate can be expected to lead to a higher ASR, our experimental results show that this is not true. When the poisoning rate is very low, it becomes more challenging for defense strategies to detect the backdoor trigger from the model due to its subtle influence. As the poison rate increases, the backdoor trigger has a more noticeable impact on the model, which can be detected and mitigated more easily by the defense strategy. Our experimental results emphasize the importance of designing defense strategies capable of detecting and mitigating backdoor attacks, even when dealing with subtle influences caused by low poison rates.

Effectiveness under Different Percentages of Clean Data We are also interested in studying the correlation between the performance of FMP and the amount of available training data, which will be used to repair the model to mitigate backdoor triggers. We compare four different retraining data ratios:5%, 10%, 15%, 20%, and 100%, and the results of our FMP are demonstrated in Fig.2. We observe that the performance of our defense strategy improves as the amount of clean training data increases. For example, when the retraining ratio increases from 5% to 100%, the ASR for BadNets decreases from 1.77% to 0.9%, while the model accuracy (Acc) improves from 86.57% to 92.02% and the Robust Accuracy (RA) increases from 86.08% to 91.04%. Similar trends can be observed for other attack strategies such as Blended, Low Frequency, SSBA, and WaNet. This indicates that our defense strategy becomes more effective in mitigating backdoor attacks as more clean data are available to retrain the model. However, it should be noted that even with a small amount of clean data (e.g., 5%), our defense strategy still exhibits competitively good performance in mitigating backdoor attacks. For example, with a 5% retraining ratio, the ASR for WaNet is 1.54%, while the Acc and RA are 89.31% and 88.07%, respectively.

Effectiveness under Different ϵ and p We further investigate the effectiveness of FMP under different ϵ and p, as listed in Fig.3 and Fig.4. We can first observe that with different ϵ, the effectiveness of FMP is consistently satisfactory. Upon increasing the ϵ, the Acc exhibits marginal decline, underscoring FMP's resilience across varying ϵ values. Subsequently, when varying the parameter p for backdoor feature pruning, a notable decrease is observed in both accuracy (Acc) and robust accuracy (RA). This reduction can be attributed to the model's failure to successfully finetune after 50% of the information is pruned along with 50% of the feature maps, hampering its performance. FMP can successfully execute to mitigate the backdoor from the model with a p larger than 4.

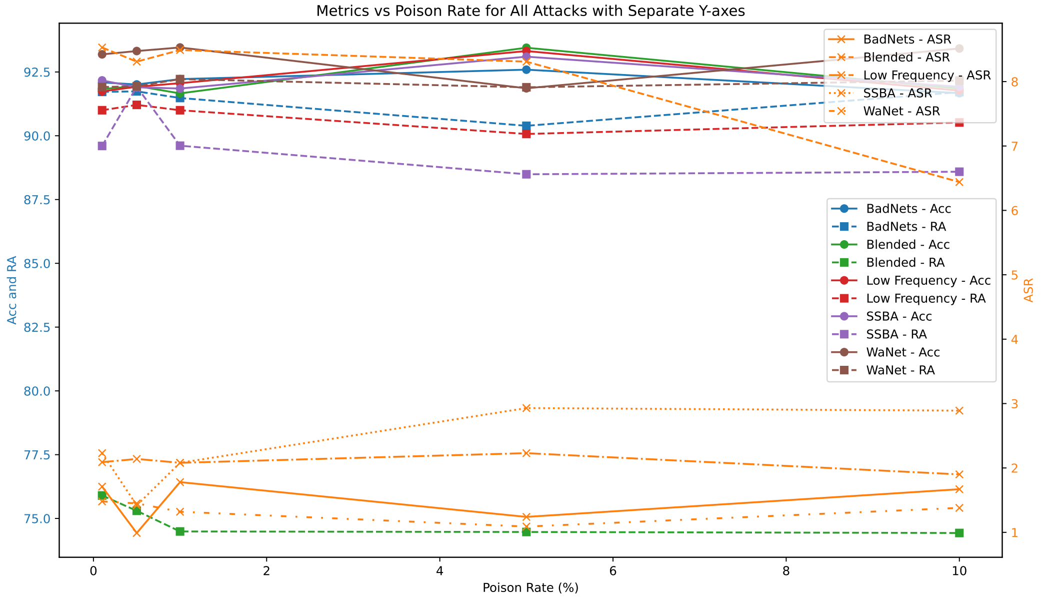

Figure 1: FMP's effectiveness under different poison rates.

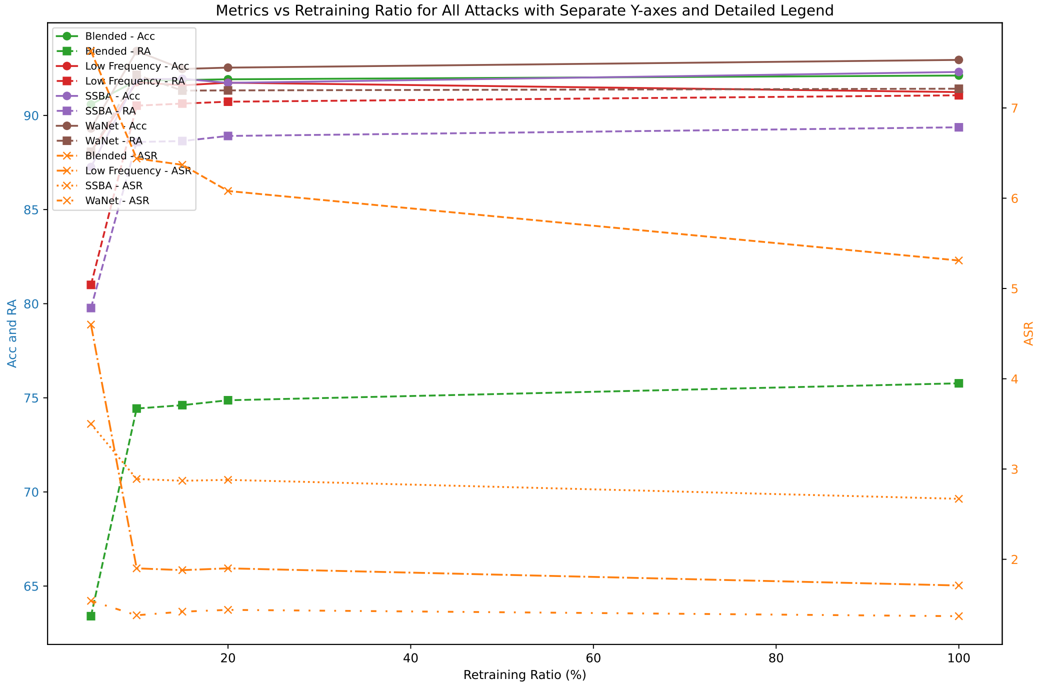

Figure 2: FMP's effectiveness under different retraining data ratios.

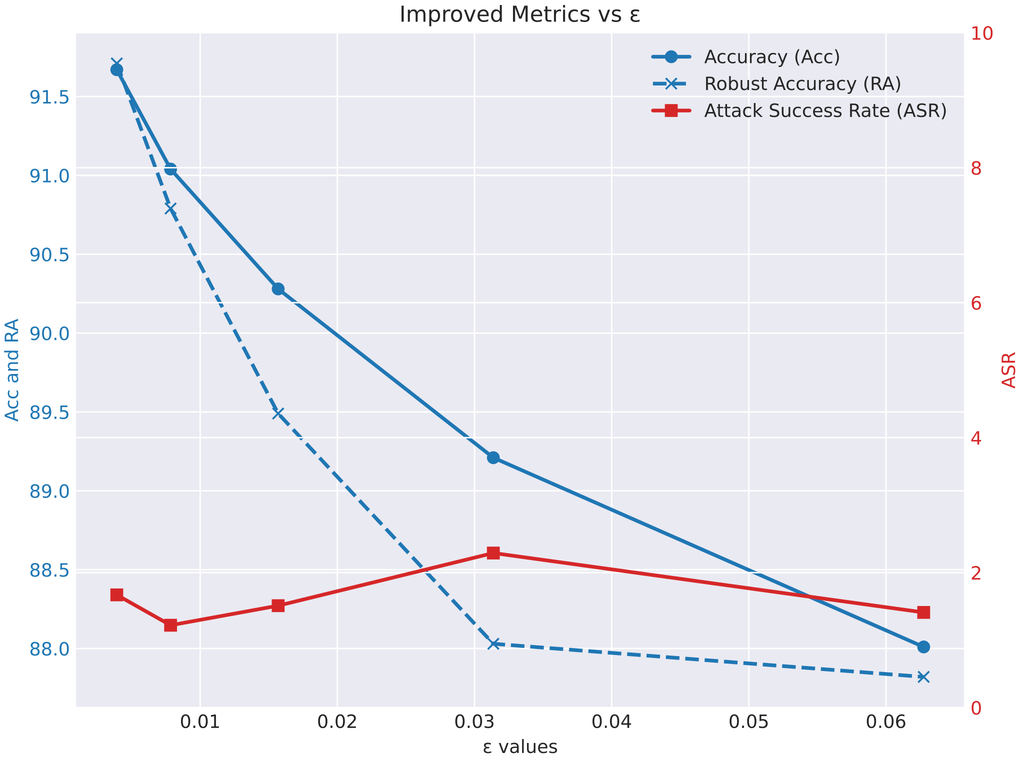

Figure 3: FMP's effectiveness under different ϵ in CIFAR10 dataset under BadNets attack.

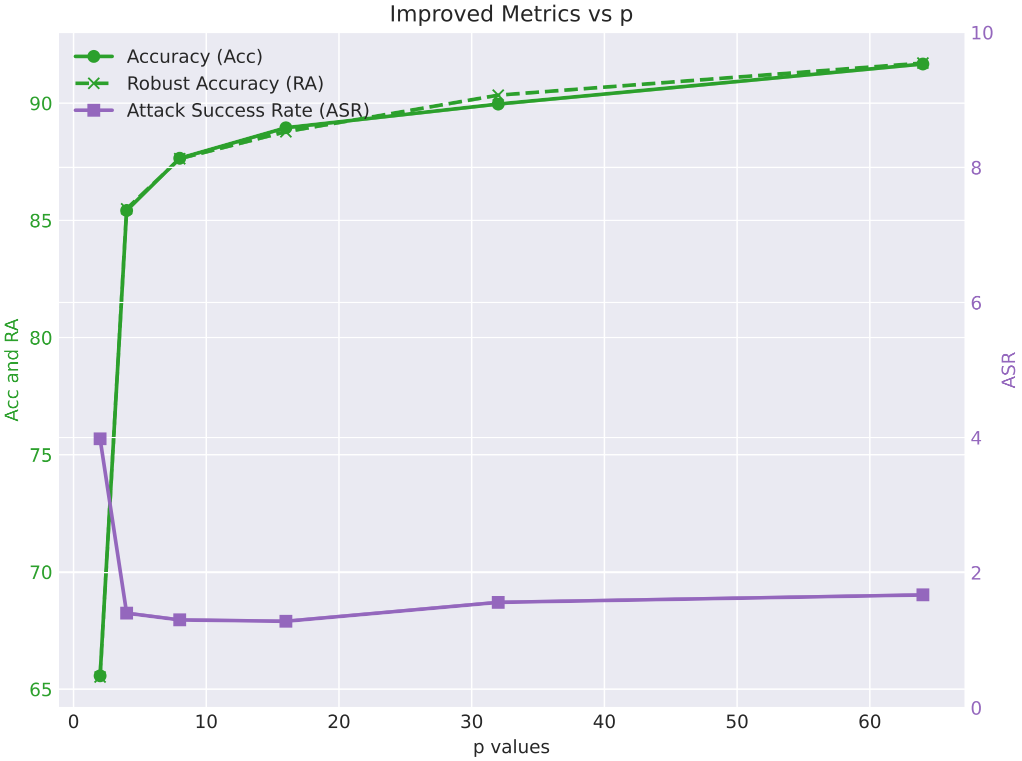

Figure 4: FMP's FMP's effectiveness under different p in CIFAR10 dataset under BadNets attack.