Defending Against Backdoor Attacks on Deep Neural Networks by Meta Backdoor Analysis

1st Tongxi Wu
Nankai University
Tianjin, China
wtxInfinity@outlook.com

2nd Yafei Hu
Nankai University
Tianjin, China
2111690@mail.nankai.edu.cn

3rd Xiaoyang Ji
Nankai University
Tianjin, China
2110611@mail.nankai.edu.cn

4th Yuchen Zhou
Nankai University
Tianjin, China
2111408@mail.nankai.edu.cn

5th Han Jiang
Nankai University
Tianjin, China
2113630@mail.nankai.edu.cn

***Abstract*—Traditional backdoor attacks often manipulate model predictions by exploiting specific trigger conditions, posing potential threats to the system. Current defense methods are typically tailored to specific attack types, assuming certain attack strategies. In real-world scenarios, attackers' methods are often unknown, we propose a defense system called the *Meta Backdoor Defense System* (MBDS) for detecting backdoor attacks in deep neural networks. Unlike traditional methods, MBDS does not rely on assumptions about the attacker's strategies. Instead, it requires only black-box access to the model, reducing the capabilities needed from the defender.**

To train the meta-classifier without prior knowledge of the attack strategy, we utilize *Jumbo Contamination* to generate shadow datasets. Additionally, we develop a feature extractor to extract classification features generated by backdoor models from shadow samples. The resulting meta-classifier demonstrates robustness against a range of backdoor attacks and improved generalization capabilities.

***Index Terms*—AI security, backdoor defense, DNN, meta backdoor analysis, meta-classifier, jumbo contamination.**

I. Introduction

With the thriving development of artificial intelligence technology represented by deep learning, its security issues have gradually attracted people's attention. Backdoor attacks stealthily inject malicious behavior and thus pose a greater threat because of its stealthiness.Traditional backdoor attacks often exploit specific trigger conditions to manipulate model predictions, posing significant threats to the system. While several methods have been proposed to detect backdoors in neural networks, many of them rely on certain assumptions about the attack strategy or require direct access to pre-trained models, which may not be practical in real-world applications and demand high capabilities from the defender.

We have proposed the *Meta Backdoor Defense System* (MBDS), a defense mechanism designed specifically to detect backdoors in samples. Notably, our approach involves the design of a meta-classifier that is capable of identifying backdoors without making assumptions about the attacker's strategies. Furthermore, our method only requires black-box access to the model, making it more practical and applicable to a wide range of attack methods and domains. Additionally, it imposes minimal requirements on the defender's capabilities.

The first challenge we address is training the meta-classifier without prior knowledge of the attacker's attack strategy. To tackle this issue, we utilize Jumbo Contamination to generate shadow datasets, enabling the meta-classifier training process in an agnostic manner regarding the attacker's strategies.

In our study, we used the Jumbo Contamination method to generate shadow sample data, in order to make the shadow samples have a wider distribution and cover various types of backdoor attacks, thereby improving the effectiveness and generalization ability of the binary classifier. This method constructs poisoned samples by randomly selecting a portion of samples as clean samples and randomly adding backdoor triggers to another small portion of samples. We have paid special attention to the design of Jumbo Distribution to ensure the inclusion of various types of backdoor attacks, and to construct a universal function by setting a series of parameters, providing a foundation for subsequent analysis and experiments.

In terms of feature extraction, we utilized deep neural network models as feature extractors and extracted classification features generated by shadow samples in the poisoning model. This feature extraction method enables the meta classifier to effectively classify malicious backdoor samples and clean samples, and has stronger generalization ability. We explored different feature selection strategies for different types of backdoor attacks and optimized the design of feature extractors based on experimental results to further improve the performance of the meta classifier.

In terms of experimental verification, our method has demonstrated good generalization ability and defense effectiveness. Our meta classifier can effectively detect backdoors in samples and has strong robustness against different types of attacks, providing strong technical support for improving the security of the model.

We would like to express our gratitude for the support and funding provided by the Undergraduate Innovation Project of Nankai University in 2023 and the Innovation and Entrepreneurship Training Program for University Students in Tianjin City.

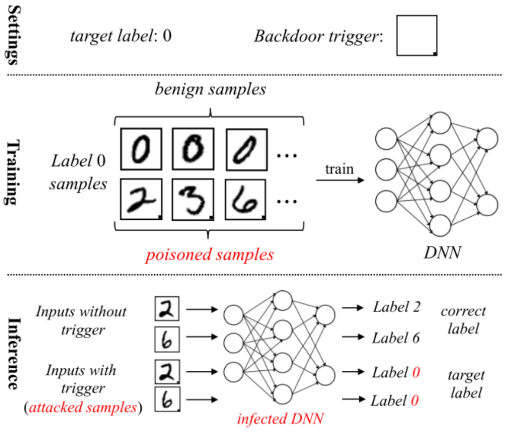


Fig. 1: Backdoor Attack Schematic Diagram in [1]

II. Background

A. Deep Neural Networks

Deep Neural Networks (DNNs) are a class of artificial neural networks characterized by multiple layers of interconnected neurons, enabling them to model complex relationships in data. Each layer of a DNN transforms the input data through a non-linear activation function $F_i (i = 1, 2, \ldots, l)$, allowing for hierarchical feature representation learning. Given input x, the output of neural network $\mathcal{F}$ can be represented as follows:

$$\mathcal{F}(\mathbf{x}; \theta) = F_l(F_{l-1}(\ldots(F_2(F_1(\mathbf{x}; \theta_1)); \theta_2); \ldots); \theta_l) \quad (1)$$

where θ represent the parameters of the model, and θ_i denote the parameters at each layer i of the network. The optimization of a DNN involves finding the optimal parameters θ^* that minimize a given loss function $\mathcal{L}$. The optimization objective can be expressed as:

$$\theta^* = \arg\min_{\theta} \sum_i \mathcal{L}(\mathcal{F}_\theta(\mathbf{x}_i), y_i) \quad (2)$$

$\mathcal{F}_\theta(\mathbf{x}_i)$ represents the output of the DNN with parameters θ for input x_i, and y_i is the corresponding ground truth label. The objective is to adjust the parameters θ iteratively using optimization algorithms to minimize the discrepancy between predicted outputs and ground truth labels, improving the model's performance and ability to generalize to unseen data.

B. Backdoor Attack

A backdoor attack is a method of embedding hidden backdoors into deep neural networks. When the backdoor is not activated, the infected model behaves similarly to models trained in benign environments. However, when the backdoor is activated by an attacker, the prediction results are altered to the target labels specified by the attacker.

The image in [1] vividly illustrates this process, as shown in Fig. 1.

C. Class Activation Mapping

Class Activation Mapping (CAM) [2] is a method that employs global average pooling and convolutional neural networks to produce class activation maps, elucidating the model's attention mechanisms and discriminative features in image classification tasks. Given a CNN with feature maps A^k at the final convolutional layer and class weights w^c,CAM computes the class activation map L^c_{CAM} as:

$$L^c_{CAM} = \sum_k w^c_k A^k \quad (3)$$

where w^c_k denotes the weight of feature map k for class c.

As an improved version of CAM, Grad-CAM [3] enhances interpretability by utilizing the gradient information that the model outputs to the weight feature map, and is applicable to a wider range of applications.

III. Threat Model & Defender Capabilities

In this section, we begin by discussing the two real-world scenarios of backdoor attacks. Subsequently, we introduce our threat model along with the defender's defense objectives and capabilities.

A. Real-World Scenarios

Due to the significant computational resources required for training models such as deep learning models, individual users or small groups often find it challenging to afford. As a result, in real-world scenarios, users tend to either hand over their models to third-party platforms for training or directly deploy pre-trained models provided by third parties. This leads to two scenarios of backdoor attacks.

1) Adopt Third-Party Platforms: In this scenario, users submit their dataset, model structure, and training schedules to untrusted third-party platforms for model training. Even if the dataset provided by the user is benign, untrusted third parties can choose not to use the user's dataset for training or manipulate the dataset (e.g., adding backdoor samples to the dataset) and train the model on it. In this case, users have no control over the training dataset and training schedules, making backdoor attacks possible.

2) Adopt Third-Party Models: In this scenario, users directly adopt pre-trained models from untrusted third parties (which may be malicious), without any access to the model's training dataset or training process. Users have extremely limited defensive capabilities in this case.

In comparison, the attacker's capabilities are stronger, and the defender's capabilities are weaker in the second scenario. As a defense work, we consider the second scenario (*adopting third-party models*) where the attacker's capabilities are maximized.

B. Threat Model

The DNN model provided by the attacker to the user (defender) should perform well on normal samples, correctly classifying benign samples, as otherwise, the user would notice it and refuse to deploy the model. However, the attacker's model, when fed with backdoor samples containing trigger patterns predetermined by the attacker, incorrectly classifies them into the class specified by the attacker.

As a defense work, we consider the attacker possesses maximum capabilities. The attacker has full access to the training dataset and has white-box access to the model. Additionally, we assume that the attacker can employ any attack strategy. They can apply any attack approach to create a backdoor model. The trigger can be any shape, in any position and any size, and can be inserted into samples with an arbitrary transparency.

C. Defender Goal

Common defense methods against backdoor attacks include *sample-level* defenses and *model-level* defenses. Sample-level defenses aim to determine whether the input sample is a backdoor sample during the inference stage of the model, preventing the activation of the model's backdoor by filtering out backdoor samples. Model-level defenses aim to diagnose whether the model has been implanted with a backdoor and refuse to deploy and use the model afterward, or locally retrain the model to eliminate the backdoor.

In the realistic scenario we are considering (see section III-A2), users lack sufficient computational resources to retrain models locally. Therefore, the most practical approach to continue deploying and using the model is to deploy a detection system simultaneously. This system detects and filters out backdoor samples, thereby preventing the activation of the model's backdoor and avoiding backdoor attacks. As a result, our focus is primarily on *sample-level* defense, and the defense workflow is illustrated in Fig. 2.

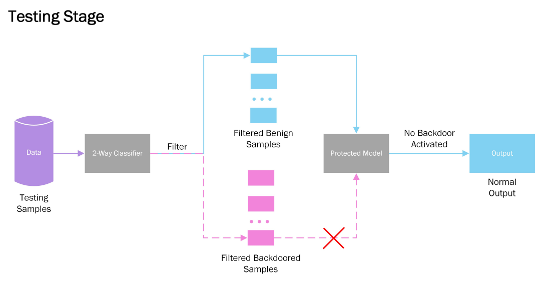


Fig. 2: Sample-Level Defense Workflow

D. Defender Capabilities

The existing requirements for defenders against backdoor attacks are relatively high, typically requiring one or more of the following capabilities:

- Assumption on Attack Strategy.
- White-Box Access to the Backdoor Model (Victim Model).
- Access to Training data.

In this paper, we consider a defender with minimal assumptions and weak capabilities. The defender does *not* make any assumptions about the attacker's attack strategies, does *not* require access to training data, and only needs *black-box* access to the backdoor model. However, the defender *does* need access to a reserved clean dataset to assist in backdoor defense. This assumption is reasonable, realistic, and has been adopted by many previous works (e.g., [4]–[7]). It should be noted that we assume the size of the reserved clean dataset is much smaller than the size of the training dataset used by the attacker to train the victim model, and the elements in these datasets are different.

IV. Meta Backdoor Defense System (MBDS)

In this section, we will discuss our Meta Backdoor Defense System (MBDS). We will begin by providing an overview of MBDS, followed by a detailed exploration of the construction process of our defense system. Finally, we will highlight the evaluation metrics for defense.

A. MBDS Overview

The overall idea behind our Meta Backdoor Defense System (MBDS) is to train a binary classifier (referred to as the meta-classifier) on a shadow dataset containing both benign samples and backdoor samples. This meta-classifier is then utilized for sample detection. Before input samples are fed into the deployed model, they are first passed through the meta-classifier. During this process, the meta-classifier filters out backdoor samples, allowing only benign samples to pass through for classification by the model.

The overview of our MBDS is shown in Fig. 3. It consists of four parts:

1) Shadow Sample Generation: In this step, we generate numerous shadow samples to serve as the shadow dataset, providing raw data for the subsequent training of the meta-classifier. We employ the Jumbo Contamination method to generate shadow samples, which will be discussed in detail later (Section IV-B).

2) Feature-Extractor Design: This part is the most crucial step in our MBDS. In this step, we need to design a feature extractor to process the shadow samples obtained in the first step, extracting features that are beneficial for determining whether samples contain backdoors. These features will be used for training the subsequent meta-classifier to achieve better detection performance. We design a DNN model (shadow model) as the feature extractor based on the architecture of the victim model (without specific parameters, making it a black-box approach) and train it on shadow samples.

3) Feature Extraction: In this step, we utilize the feature extractor to extract features. The features we select are the feature maps (*a.k.a.* Representations) from the shadow model because they contain relevant features extracted by the model that are related to the model's classification criteria.

4) Meta-Classifier Training: Finally, we feed the extracted features into the meta-classifier and train it using gradient-based optimization methods. The trained meta-classifier can then be deployed into the MBDS defense system.

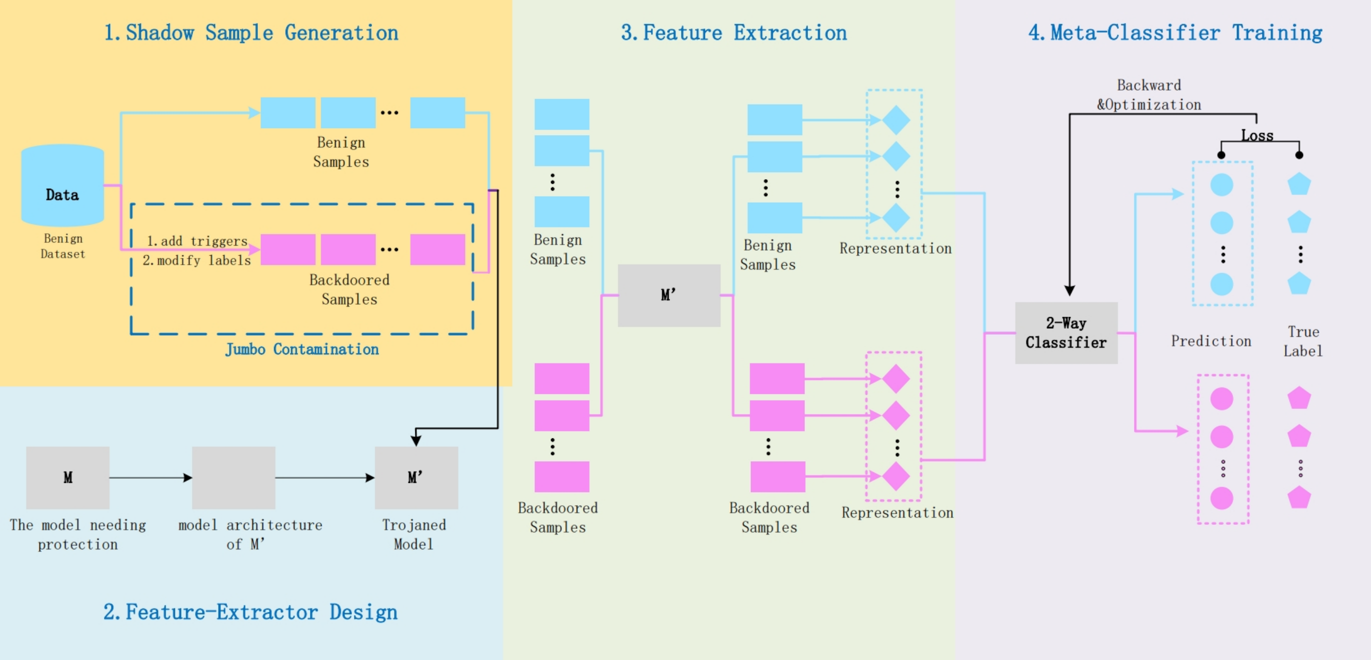


Fig. 3: MBDS Overview

B. Shadow Sample Generation

To better enhance the effectiveness and generalization capability of the binary classifier, we adopted the *Jumbo Contamination* proposed in [7] for generating shadow samples.Jumbo Contamination assumes that attackers may employ various attack strategies corresponding to different types of trojan attacks. Employing this method for shadow sample generation enables the shadow samples to have a broader distribution.

The specific operation of Jumbo Contamination involves randomly selecting a portion of samples as clean samples. For another small portion of samples, backdoor triggers are randomly added while simultaneously modifying the corresponding labels of the samples to construct poisoned samples. The shape, position, opacity, etc., of the backdoor triggers are sampled according to the Jumbo Distribution.The precise formula for the Jumbo Distribution is detailed below,

$$\begin{aligned} \mathbf{x}', y' &= \mathcal{I}(\mathbf{x}, y; \mathbf{m}, \mathbf{t}, \alpha, y_t) \\ \mathbf{x}' &= (1-\mathbf{m}) \cdot \mathbf{x} + \mathbf{m} \cdot \big((1-\alpha)\mathbf{t} + \alpha \mathbf{x}\big) \\ y' &= y_t \end{aligned} \tag{4}$$

where $m \in \{0,1\}^{d_x}$ represents the mask for the trigger,$t \in \mathbb{R}^{d_x}$denotes the pattern and regularity of the trigger,and α represents the opacity embedded into x.The function I initializes a series of parameters to specify the various types of backdoor attacks to be included.

Given the raw dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ and the number of the shadow datasets m, we processed the dataset $\mathcal{D}$ with jumbo contamination to generate a set of m shadow datasets $\mathbb{D}^m$. The jumbo contamination is shown in Algorithm1. We first randomly sample the jumbo settings (line 3). Then we select the dataset $\mathcal{D}_{contam}$ from the raw dataset $\mathcal{D}$ and operated each sample in $\mathcal{D}_{contam}$ according to Eqn.4 to generate the shadow dataset $\mathcal{D}_{shadow}$ (line 5-8). We repeat the process m times to generate a set of shadow datasets $\mathbb{D}^m$.

Algorithm 1: Jumbo Contamination

```
Input: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, number of shadow
       datasets $m$
Output: $\mathbb{D}^m$: a set of $m$ shadow datasets
1  $\mathbb{D}^m \leftarrow []$;
2  for $u \leftarrow 1$ to $m$ do
3      $\mathbf{m}, \mathbf{t}, \alpha, y_t, p$ =gen_jumbo_setting();
4      $\mathcal{D}_{shadow} \leftarrow \mathcal{D}$;
5      $\mathcal{D}_{contam} \leftarrow$sample_from$(\mathcal{D}, p)$;
6      foreach sample $(\mathbf{x}_j, y_j)$ in $\mathcal{D}_{contam}$ do
7          $(\mathbf{x}'_j, y'_j) \leftarrow$ insert_backdoor$(\mathbf{x}_j, y_j; \mathbf{m}, \mathbf{t}, \alpha, y_t)$;
8          $\mathcal{D}_{shadow} \leftarrow \mathcal{D}_{shadow} \cup \{(\mathbf{x}'_j, y'_j)\}$;
9      end
10     $\mathbb{D}^m \leftarrow \mathbb{D}^m \cup \{\mathcal{D}_{shadow}\}$
11 end
12 return $\mathbb{D}^m$;
```

C. Feature-Extractor Design

The success of a backdoor attack requires three conditions [1]: 1) the model is injected with a backdoor; 2) the samples contain backdoor triggers; 3) the triggers match the model's backdoor. Now that the victim model has been implanted with a backdoor, only samples injected with matching backdoor triggers can activate the backdoor. To detect backdoors in samples, using the corresponding backdoor model as a feature extractor is undoubtedly the most suitable choice. Since the defender cannot access the training dataset of the backdoor model, we simulate this by using shadow samples. The corresponding feature extractor is the backdoor shadow model trained on shadow samples.

Since the defender has knowledge of the structure of the victim model, the feature extractor can be designed to have an architecture that is consistent or similar to the victim model's structure.

D. Feature Extraction

According to [8], poisoned models depend on different criteria when classifying benign samples versus backdoor samples. The model primarily depends on backdoor triggers in the samples to classify backdoor samples, whereas for benign samples, the model relies more on features related to the class in the samples for classification. Since the representations (i.e., feature maps) of deep neural networks contain the basis for model classification, we consider using the model's feature maps as input features to train the meta-classifier.

This part can be seen as the feature engineering process in traditional machine learning, with the goal of improving the model's performance.

E. Meta-Classifier Training

We employ a neural network model as our meta-classifier, thus enabling us to use gradient-based optimization methods for training. By utilizing the previously extracted features as training data, we compute the loss between the outputs of the classifier and the ground truth labels (indicating the presence of backdoors) and update the classifier's parameters through error backpropagation. This process enables the classifier to learn how to detect backdoor samples.

F. Evaluation Metrics

For the task of detecting whether samples contain backdoors, which is a binary classification task, we can use the following three metrics to comprehensively evaluate the defense effectiveness:

1) **False Acceptance Rate** (**FAR**): Represents the probability of mistaking a backdoor sample as a benign sample.
2) **False Rejection Rate** (**FRR**): Represents the probability of mistaking a benign sample as a backdoor sample.
3) **F2 Score**: A weighted harmonic mean of precision and recall, with a higher weight on recall (beta = 2 in the F-beta measure).

These metrics are used to evaluate the security (FAR), usability (FRR), and overall detection performance (F2 Score) of the defense system in detecting backdoor samples.

The values of these metrics range from 0 to 1, where lower values of FAR and FRR indicate better detection performance, while a higher F2 Score indicates better overall detection performance.

In practical scenarios, there is often a trade-off between FAR and FRR, where a slightly higher FRR may be accepted to minimize FAR.

V. Evaluation

In this section, we will commence by presenting our experimental setup, followed by a detailed analysis and explanation of jumbo contamination and the feature extractors along with the features extracted. Subsequently, we will evaluate the defense effectiveness of MBDS (Meta Backdoor Defense System). Additionally, we will investigate the impact of the number of feature extractors on the defense system's performance. Lastly, we will engage in a discussion regarding alternative feature designs for MBDS.

A. Experiment Setup

Since the majority of deep neural network backdoor attacks and their defenses are concentrated in the computer vision domain, particularly in image classification tasks, our experiments also focus on image classification. We conducted tests using the MNIST and CIFAR-10 datasets. It is important to note that we randomly sampled only 0.04 of the datasets as the reserved dataset for the defender and 0.50 as the training dataset for the attacker. This choice mirrors the realistic scenario where users often struggle to have a large amount of training data available.

For MNIST, we adopt the same CNN structure as in [9]. For CIFAR-10, we use the same CNN structure as in [10].

a) Feature Extraction: We utilized a model whose architecture is identical to the victim model as our feature extractor. The output feature maps before the first fully connected layer of the CNN model are selected as feature representations. To enhance feature diversity, we generated multiple shadow sample datasets and trained corresponding feature extractors on each dataset separately (the impact of the number of feature extractors on the defense system will be discussed in Section V-D). Additionally, to improve feature quality, we introduced two filtering layers during feature extraction. First, we filtered out features from samples misclassified by the shadow model to ensure that the extracted features contain accurate classification cues. Second, we filtered out features from backdoor samples whose ground-truth label matches the target label, as the shadow model's classification basis for these samples is uncertain (it may rely on class information, backdoor triggers, or both). Furthermore, to enhance the efficiency of meta-classifier training, we randomly selected 0.01 of the extracted features from each feature extractor as training data for the meta-classifier.

b) Meta-Classifier: Our meta-classifier adopts a simple CNN model consisting of two 1×1 convolutional layers [11] and three fully connected layers. The 1×1 convolutional layers are primarily used to extract features from the feature map input, capturing information about backdoor neuron features, while the subsequent fully connected layers are utilized to compute the backdoor score.

B. Jumbo Contamination & Feature Extraction

In Jumbo Contamination, we are able to obtain various styles of backdoor triggers as we sample from the simulated

TABLE I: Benign Accuracy (BA) and Attack Success Rate (ASR) of feature-extractors.

Dataset	# of Shadow Datasets	BA	ASR
MNIST	1024	96.23%	91.38%
CIFAR-10	1024	46.04%	90.90%

backdoor distribution. Examples of trigger and backdoor samples are illustrated in Fig. 4.

Performance of the shadow models (feature extractors) trained on the shadow dataset obtained through Jumbo Contamination is shown in Table I. It should be noted that due to the defender having only 0.04 of the complete dataset, the shadow models' performance on the CIFAR-10 dataset shows a significant drop compared to the baseline. However, since we added two layers of filters during feature extraction (section V-A0a), this does not affect the quality of our features and the performance of the meta-classifier.

The visualization of features extracted from benign and backdoor samples using the feature extractor is shown in Fig. 4. It is evident that there is a certain correlation between the activations in the feature maps and the backdoor triggers. This indicates the rationale behind using the model's feature maps as diagnostic information for backdoors. Therefore, training the meta-classifier with these features is expected to achieve good detection performance and stronger generalization capability (this will also be validated in section V-C).

C. Detection Evaluation

To evaluate the defense performance of our MBDS, we generated 64 sets of shadow datasets using Jumbo Contamination and trained victim models on these datasets. Additionally, we employed BadNets attack [9] and Blended attack [12] to generate 32 victim models each. The performance of the victim models is shown in Table II. We deployed MBDS on a total of 128 victim models under different attack settings and evaluated its defense effectiveness, as shown in Table III.

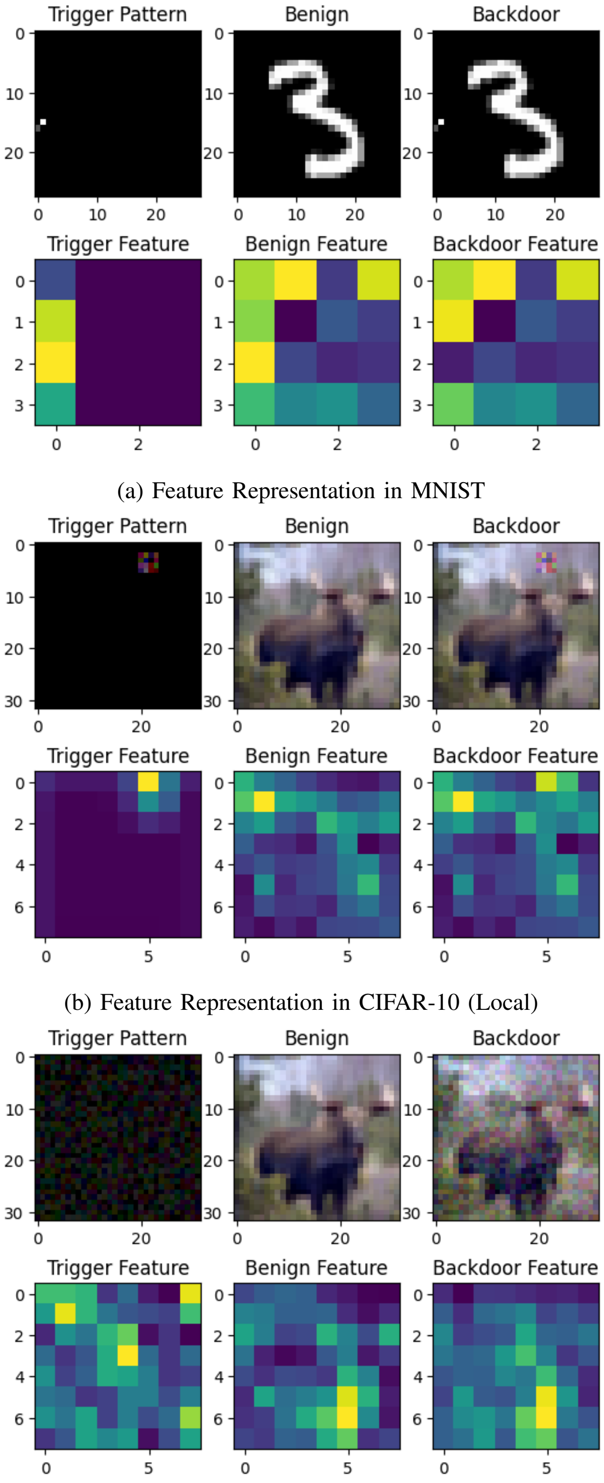
TABLE II: Performance of Victim Models

Dataset	Attack Setting	BA	ASR
MNIST	Jumbo	99.07%	99.51%
MNIST	BadNets	99.18%	99.89%
MNIST	Blended	99.16%	99.99%
CIFAR-10	Jumbo	68.25%	99.59%
CIFAR-10	BadNets	68.32%	99.96%
CIFAR-10	Blended	68.27%	99.20%

TABLE III: MBDS Defense Performance

Dataset	Attack Setting	Accuracy	FAR	FRR	F2
MNIST	Jumbo	59.49%	49.12%	31.90%	50.27%
MNIST	BadNets	54.31%	68.61%	22.77%	32.66%
MNIST	Blended	70.32%	42.40%	16.96%	59.04%
CIFAR-10	Jumbo	80.01%	20.85%	19.13%	77.61%
CIFAR-10	BadNets	87.03%	00.60%	25.34%	94.73%
CIFAR-10	Blended	57.02%	35.00%	50.95%	61.71%

It can be observed that MBDS performs relatively well against BadNets attacks (F2 Score of 94.73%), performs



(a) Feature Representation in MNIST

(b) Feature Representation in CIFAR-10 (Local)

(c) Feature Representation in CIFAR-10 (Global)

Fig. 4: Feature Representations

poorly in Blended attacks (F2 Score of 61.71%), and falls between the two in Jumbo attacks, which simulate various backdoors. This suggests that learning global backdoor trigger information is more challenging than learning local trigger information.

Additionally, we noticed performance differences of MBDS across different datasets. MBDS performs better on the CIFAR-10 dataset compared to the MNIST dataset. This difference may be attributed to the simplicity of data in MNIST, leading to overfitting of the meta-classifier on backdoor sample features. It could also be due to the selected features containing less information (feature size of $4 \times 4 \times 32$), resulting in poorer generalization performance of the meta-classifier. We believe that improving the performance of MBDS on MNIST can be achieved by modifying the meta-classifier architecture or selecting features with more information.

It should be noted that we only used 0.01 of the shadow sample data (0.04 of the full dataset) for training the meta-classifier. We believe that if more or all shadow sample data were used, the defense effectiveness of MBDS should improve, and its generalization performance should be stronger.

D. Impact of Number of Feature Extractors

In this section, we further investigate the impact of the number of feature extractors on the defense performance of MBDS. The results are shown in Fig. 5.

The results indicate that although there is some fluctuation in the metrics with the change in the number of feature extractors, overall there is a trend of improvement in all metrics as the number of feature extractors increases. This trend suggests that the defense performance of MBDS is getting better. The defense effectiveness gradually stabilizes when the number of feature extractors reaches 1024.

E. Research on Feature Selection

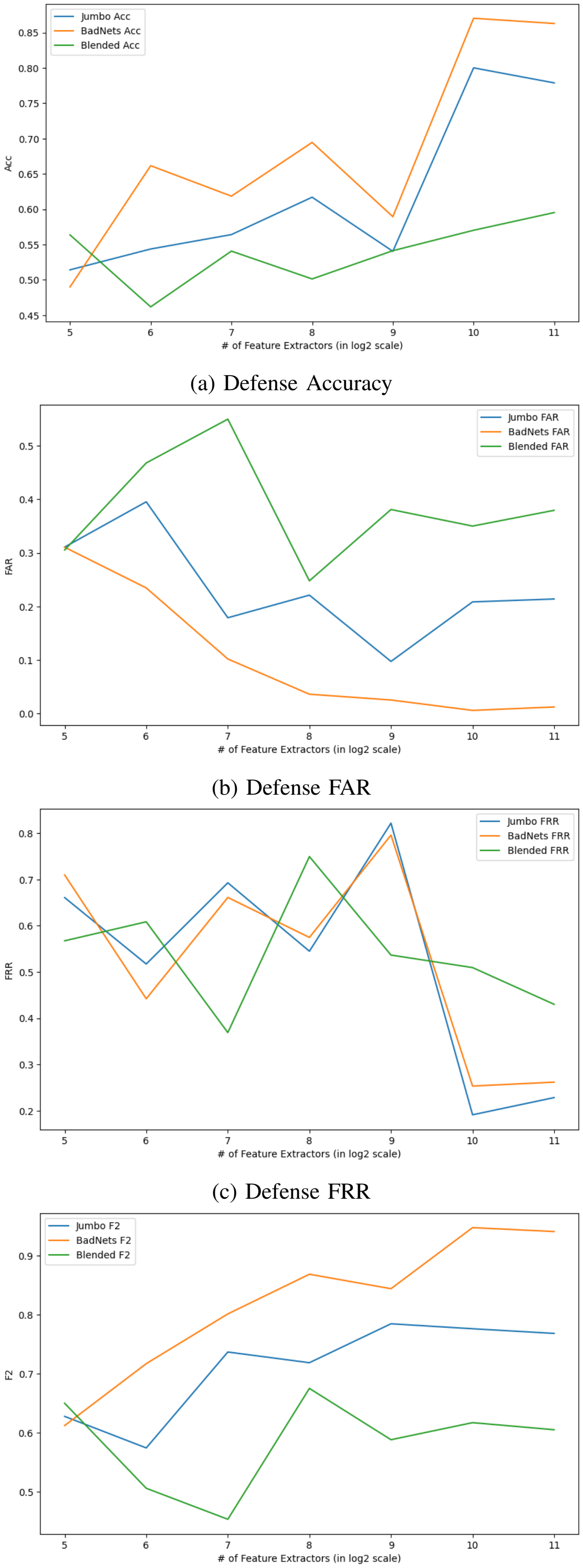
In this section, we explore other feature designs for MBDS. In addition to the feature maps from the DNN model that we used earlier, we also attempted to use Grad-CAM [3] as the feature adopted by the defense system. Through experimental validation, its performance is generally similar to that of the feature maps. This result partially indicates that our MBDS does not depend on specific feature designs. Furthermore, we believe that more refined features will contribute to further improving the defensive performance of MBDS.

VI. Discussion & Future Work

A. Discussion

Our research has introduced innovative methods in two key areas. Firstly, we posit a decreased ability of defenders to simulate reality. Secondly, the integration of Jumbo Contamination and Feature Extraction enhances robustness and generalization.

Most state-of-the-art research on test sample filtering attacks requires a strong assumption about the defender's capabilities, including access to a benign dataset of the same type. In real-world scenarios, backdoor defense can often be challenging.



(a) Defense Accuracy

(b) Defense FAR

(c) Defense FRR

(d) Defense F2 Measure

Fig. 5: Defense Performance with respect to the number of feature extractors on CIFAR-10.

In survey [1], three types of backdoor attack scenarios are mentioned. Third-party datasets, platforms or models are all very common situations in reality. In these cases, attackers have complete control over the original type of training data, making it very difficult for defenders to obtain a completely benign dataset of the same type.

In our simulating scenario, a white-box model is the only thing the defender is equipped with to facilitate a protection, without the need for any benign data set of the same type. By utilizing any other type of benign data set through Meta Backdoor Analysis, the defender can obtain a meta-classifier to defend against backdoor attacks by filtering samples. This makes our defense method more practical and closer to real-world situations. Additionally, the defender does not need to know the attacker's specific method. Due to the use of Meta Backdoor Analysis, our defense method can effectively address various backdoor attacks, demonstrating strong generalization and robustness.

Based on the previous analysis, our defense method adopts the Meta Backdoor Analysis approach, using Jumbo Contamination in Shadow Sample Generation to simulate various backdoor triggers for backdoor attacks. The design of the feature extractor and the selection of features are conducive to enlarging the difference between backdoor samples and benign samples, making our defense method more robust and with stronger generalization capabilities.

In conclusion, as a method for detecting backdoor samples, our proposed approach has lower requirements for real-world scenarios and strong generality. With the widespread use of deep learning in various fields today, our method can better resist attacks targeting different tasks in different situations, allowing more tasks based on deep learning methods to be conducted safely and improving the reliability of related work.

B. Future Work

For our current work, we can extend the scope of our research in two main aspects. On one level, we can expand the scenarios for our backdoor defense by assuming that the defensive capabilities are weaker, meaning that the meta-classifier trained by the defense side is not aware of the architecture of the model to be protected in advance. Apart from that, we can also assume that the attacking side has stronger capabilities, allowing the attacker to bypass our meta-classifier detection using more powerful backdoor attack methods such as adaptive attacks.

First, we can make an assumption to modify the defense side of our scenario. Assuming that the defense side lacks prior knowledge of protecting the model when implementing backdoor defense. Therefore, it would be more challenging for a defender to train a meta-classifier through meta backdoor analysis. In fact, despite Oh (2019) [13] proposing a method using existing techniques to infer the structure of a black-box model, research on attack scenarios with defenders having very weak capabilities often affiliates with reality and is more practically significant.

In addition, we can assume that the attacker obscure stronger capabilities. There are two paper from Shokri et al. (2020) [14] and Saha et al. (2020) [15] introducing a more powerful type of backdoor attack called Adaptive Attack. We conducted experiments on both local backdoor triggers such as BadNets (Gu et al., 2019) [9] and global backdoor triggers such as Blended Backdoor Attacks (Chen et al., 2017) [12]. However, if the attacker has access to the relevant parameters of our meta-classifier, i.e., a white-box attack, the attacker may potentially devise a new adaptive attack that can strategically incorporate a special training process during the poisoning phase of the model to bypass our meta-classifier's filtering of poisoned samples during the testing phase.

Therefor, we plan to adjust the structure of our meta-classifier or relating methods to obtain relatively satisfying results under similar scenarios when the attacker is stronger pr the defender is weaker than we expected.

VII. Related Work

In this section, we will introduce recent research on backdoor attacks, backdoor defense, and meta-analysis in deep neural networks, and compare them with our proposed methods.

A. Backdoor Attacks

In recent years, researchers have proposed various backdoor attack techniques for deep neural networks.

Gu et al. (2019) proposed a method for evaluating backdoor attacks in deep neural networks, called Badnets [9]. They utilized adversarial training and model fine-tuning to successfully generate malicious models with predetermined backdoor behavior; Chen et al. (2017) [12] explored a targeted backdoor attack method for deep learning systems, utilizing data pollution techniques to implant backdoor triggers during model training. This method tampers with the training data, causing the model to exhibit misclassification behavior when receiving specific trigger inputs;

Liu et al. (2018) proposed a backdoor attack technique called Trojaning [16], which successfully embedded backdoor triggers in neural networks by modifying training data or model parameters, achieving control over the model; Turner et al. (2018) studied a backdoor attack method called Clean label [17], which utilizes normal training data but successfully embeds backdoor behavior into the model through reasonable regularization and optimization techniques, leading to misclassification behavior in specific situations.

Recently, Souri et al. (2022) [18] proposed a backdoor attack technique called Sleeper Agent, which can implant hidden backdoor triggers in neural networks trained from scratch and has scalability on large-scale datasets.

It can be seen that backdoor attack technology seizes the fragility of model classification and tampers with data at the feature layer to achieve the goal. Our work also focuses on the different levels of image features, innovates feature extraction methods, and identifies backdoor features.

B. Backdoor Defenses

Backdoor Defenses refer to defending against both data and models. Most defense efforts against backdoor attacks are focused on input samples and model parameters, distinguishing between malicious and benign inputs based on their statistical differences in the backdoor model.

For models, they can be divided into model purification and model inspection.

Defenses [19] [4] [20] consider that triggers can trigger abnormal activation values, and perform activation statistics and pruning, fine-tuning, and other operations on neurons, achieving good results; [21]indirectly applies pattern connections to check backdoor behavior, effectively reducing backdoors.

However, modifying the model to defend against potential disruptions to its normal functionality and potentially consuming significant computational resources for fine-tuning, so there is room for improvement in such methods. Therefore, we choose to adopt a simpler and more efficient method, which takes into account the dimensions of the dataset.

Furthermore, for data, it can be divided into input transformation and input filtering. For input filtering, it can be further divided into training sample filtering based defenses and testing sample filtering based defenses.

The AC [22] used activation clustering technology to observe the activation patterns of neural networks under normal input and cluster them into several clusters. Then, the activation mode of the network when subjected to backdoor attacks was analyzed, and abnormal behavior was detected by comparing it with normal mode. The unique feature of this method is that it does not rely on prior knowledge of the existence or form of backdoor attacks, but rather detects anomalies by learning normal behavior, thus having a certain degree of universality and applicability.

The STRIP [5] designed by Gao et al. is a very typical work of input filtering. The idea of this scheme is to strongly perturb each input sample to detect trigger input. For perturbed trigger inputs, their predictions remain unchanged under different modes of disturbance, while the predictions differ greatly when different disturbance modes act on benign samples. Therefore, an entropy measure is introduced to quantify this prediction randomness. Finally, it is clear to distinguish between trigger inputs that always display low entropy and benign inputs that always display high entropy.

Although previous methods have achieved good defense effects in specific scenarios, the above model cannot perform well when attackers use improved attack methods (such as randomly adding backdoors); The reason for this is that the model did not grasp the standards for classifying backdoor samples, resulting in poor generalization ability. In contrast, our method adopts a deep learning approach to extract features from different backdoors, train a meta classifier to distinguish backdoor samples, and achieve precise filtering of samples during the testing phase, making it simpler and more efficient than previous methods.

C. Meta Analysis

Meta analysis is a systematic research method aimed at synthesizing and analyzing data from multiple independent studies to provide more comprehensive and accurate conclusions. In the field of computer security, especially in deep learning and neural network security, meta-analysis is often used to integrate the results of different studies to reveal general trends and insights, thereby helping to develop more effective defense strategies.

The paper [7] is a study on using meta neural analysis methods to detect artificial intelligence trojans. This method first injects known backdoor triggers into the training data and collects corresponding model output data. Then, the meta neural network is used to analyze these output data to identify any signs of backdoor attacks in the model. This method utilizes the concept of meta learning to detect backdoor attacks through meta analysis of model behavior.

Based on the above papers and related work, it can be seen that metabackdoor analysis has made certain progress in analyzing backdoor attacks. By integrating data from multiple independent studies and model behavior, the metabackdoor analysis method can identify potential backdoor triggers in the model, thereby improving the detection ability of backdoor attacks. The advantage of this method is that it can analyze different types of backdoor attacks, rather than being limited to specific attack methods. Therefore, when designing defense strategies, meta backdoor analysis can be considered as an effective detection method to improve the security and robustness of deep neural networks.

VIII. Conclusion

In this paper, we proposed a defense system (MBDS) for deep neural network backdoor attacks based on meta backdoor analysis, which can effectively counter various types of backdoor attacks under the condition of weak defender capabilities. However, due to the challenging assumptions of strong attacker capabilities and arbitrary attack strategies, the generalization ability of our defense system needs further improvement. In conclusion, our work explores the general defense against backdoor attacks. We look forward to more researchers contributing to this area of research.

Acknowledgment

We would like to express our gratitude to Nankai University for providing us with the platform of the College Students' Innovation and Entrepreneurship Training Program, which allowed us to participate in projects, explore scientific research, and make progress in our spare time.

We extend our heartfelt thanks to the College of Cyber Space Security in Nankai University for their long-term strong support and comprehensive assistance to our project, enabling us to focus on scientific research and explore mysteries.

During the completion of this project, we are deeply grateful for the meticulous guidance and support from Professor Zheli Liu and Dr. Biao Yi. Their professional knowledge, rich experience, and selfless dedication provided us with valuable guidance and inspiration. Professor Liu's profound understanding and insightful views in the field of artificial intelligence security have guided us in problem-solving and research. Dr. Yi's rigorous academic attitude and patient guidance helped us overcome many challenges, enabling the smooth progress of the project.

Furthermore, we would like to thank all the team members who participated in this project. Their hard work and spirit of cooperation made significant contributions to the smooth progress of the project. Throughout the research process, everyone supported each other, collaborated closely, and overcame many challenges together, achieving satisfactory results.

Finally, we sincerely thank the relevant institutions and individuals who provided resources and support. Their assistance provided us with the necessary conditions and environment for the smooth progress and completion of this project.

References

[1] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[2] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[3] 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 *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

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

[5] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: A defence against trojan attacks on deep neural networks," in *Proceedings of the 35th annual computer security applications conference*, 2019, pp. 113–125.

[6] E. Chou, F. Tramèr, G. Pellegrino, and D. Boneh, "Sentinet: Detecting physical attacks against deep learning systems.(2018)," *arXiv preprint arXiv:1812.00292*, 2018.

[7] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting ai trojans using meta neural analysis," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 103–120.

[8] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[9] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[10] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. Ieee, 2017, pp. 39–57.

[11] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

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

[13] S. J. Oh, B. Schiele, and M. Fritz, "Towards reverse-engineering black-box neural networks," *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 121–144, 2019.

[14] R. Shokri *et al.*, "Bypassing backdoor detection algorithms in deep learning," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020, pp. 175–183.

[15] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 957–11 965.

[16] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[17] A. Turner, D. Tsipras, and A. Madry, "Clean-label backdoor attacks," 2018.

[18] H. Souri, L. Fowl, R. Chellappa, M. Goldblum, and T. Goldstein, "Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch," *Advances in Neural Information Processing Systems*, vol. 35, pp. 19 165–19 178, 2022.

[19] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International symposium on research in attacks, intrusions, and defenses*. Springer, 2018, pp. 273–294.

[20] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "Abs: Scanning neural networks for back-doors by artificial brain stimulation," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1265–1282.

[21] P. Zhao, P.-Y. Chen, P. Das, K. N. Ramamurthy, and X. Lin, "Bridging mode connectivity in loss landscapes and adversarial robustness," *arXiv preprint arXiv:2005.00060*, 2020.

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