PSBD: Prediction Shift Uncertainty Unlocks Backdoor Detection

Wei Li[1], Pin-Yu Chen[2], Sijia Liu[3], Ren Wang[1*]
[1]Illinois Institute of Technology
[2]IBM Research
[3]Michigan State University

Abstract

Deep neural networks are susceptible to backdoor attacks, where adversaries manipulate model predictions by inserting malicious samples into the training data. Currently, there is still a significant challenge in identifying suspicious training data to unveil potential backdoor samples. In this paper, we propose a novel method, Prediction Shift Backdoor Detection (PSBD), leveraging an uncertainty-based approach requiring minimal unlabeled clean validation data. PSBD is motivated by an intriguing Prediction Shift (PS) phenomenon, where poisoned models' predictions on clean data often shift away from true labels towards certain other labels with dropout applied during inference, while backdoor samples exhibit less PS. We hypothesize PS results from the neuron bias effect, making neurons favor features of certain classes. PSBD identifies backdoor training samples by computing the Prediction Shift Uncertainty (PSU), the variance in probability values when dropout layers are toggled on and off during model inference. Extensive experiments have been conducted to verify the effectiveness and efficiency of PSBD, which achieves state-of-the-art results among mainstream detection methods. The code is available at https://github.com/WL-619/PSBD.

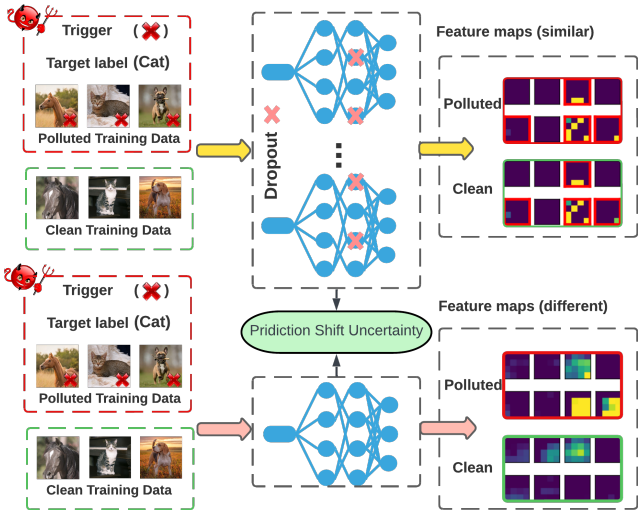


Figure 1. A simple conceptual diagram of the Prediction Shift Backdoor Detection (PSBD) framework. The introduction of dropout during the inference stage induces a neuron bias effect in the model, causing the final feature maps of clean data and backdoor data to become highly similar, ultimately leading to the occurrence of the Prediction Shift phenomenon, which serves as a basis for detecting backdoor training data.

1. Introduction

The proliferation of Deep Neural Networks (DNNs) has heralded a new era in artificial intelligence, driving progress across diverse sectors, including computer vision, autonomous driving and healthcare personalization [24, 31, 36, 41]. Yet, as their application scope broadens, DNNs have become increasingly vulnerable from a security standpoint. One of the most notable threats in this arena is the rise of backdoor attacks [7, 10, 13, 32]. These attacks involve the surreptitious insertion of altered samples into training data, enabling attackers to subtly manipulate a DNN's output, leading to incorrect predictions under certain triggers. The implications of such vulnerabilities are especially severe in contexts demanding high security, as they can lead to catastrophic outcomes [1, 5].

Notwithstanding an increased recognition of these risks, there are many different type of defense strategies currently, such as backdoor model reconstruction [2, 20, 30, 33, 49], backdoor model detection [3, 38, 46, 48], and poison suppression [26]. However, the majority of these methods primarily focus on determining whether trained models contain backdoors or on mitigating the impact of such vulnerabilities. In contrast, there is a noticeable shortage of advanced and efficient approaches that can proactively identify backdoor training samples at the initial stages. Current research in identifying backdoor training data often suffers from either a low true positive rate - indicating a low detection rate of

*Corresponding author: Ren Wang (rwang74@iit.edu)

backdoor data, or a high false positive rate - indicating a high error rate in identifying clean data [4, 11, 14, 19, 34], as detailed in Table 1. This issue is largely attributed to the focus of most existing research on data-level operations without utilizing the inherent properties of the models themselves.

To address this identified gap, we offer a new perspective - the model predictive uncertainty and propose a novel backdoor data detection approach named Prediction Shift Backdoor Detection (PSBD), which is inspired by an intriguing Prediction Shift (PS) phenomenon.

The PS phenomenon is observed when the predictions made by a poisoned model on clean data tend to deviate from the correct labels, moving towards other certain labels, especially when dropout is used during inference. Conversely, the predictions on backdoor data generated by both classical and advanced attacks remain relatively stable. This observation of PS led us to hypothesize the existence of a weights-neuron bias in DNN models, which we called the "neuron bias" effect, where some certain paths in the network become predisposed towards the specific class after training, and the backdoor samples have different paths compared with clean data. Figure 1 provides a brief overview of the PSBD framework and the neuron bias effect. Under normal conditions, the feature maps (the final convolutional layer) of clean and backdoor data exhibit significant differences. However, after applying the dropout, the neuron bias effect is induced within the model, causing the feature maps of clean and backdoor data to become strikingly similar, ultimately resulting in the occurrence of the PS phenomenon. We also provided more detailed experimental explanations for verifying the neuron bias effect in section 4.2.

Driven by these insights, Prediction Shift Uncertainty (PSU) is designed to measure the strength of PS that computes the variability in prediction confidences when a model evaluates a sample with both enabled and disabled dropout. A lower PSU value indicates a higher likelihood of the sample being malicious. By calculating PSU, the PSBD approach can effectively segregate backdoor data from clean data using a small set of label-free clean validation data.

Our approach represents a significant stride in backdoor data detection. Unlike existing methods, it focuses on the inherent uncertainty within the model, analyzing how dropout influences prediction probabilities of clean and backdoor samples. In summary, our main contributions are four-fold:

- We reveal the PS phenomenon, showing that poisoned model predictions on clean data tend to deviate from ground true labels towards specific other labels when dropout is applied during inference, while backdoor data exhibits less PS.
- We present a novel insight into the vulnerability of DNNs to backdoor attacks, linking it to the model's inherent predictive uncertainty. Our analysis delves into the impact of dropout on PS and introduces the concept of neuron bias within DNN models.
- We propose the PSBD method, a simple yet powerful uncertainty-based approach for detecting backdoor training data, marking a significant advancement in the field.
- We conduct extensive experiments across multiple benchmark datasets, rigorously evaluating our method under diverse attack scenarios and comparing it with a variety of defenses, demonstrating its effectiveness and robustness.

2. Related Work

In this section, we explore the existing literature on backdoor attacks in neural networks and the defense strategies developed to counter them.

Backdoor Attacks. Backdoor attacks are particularly dangerous, injecting triggers into a target model that cause it to misclassify inputs containing these triggers while operating normally on unaltered samples [6, 13, 28]. Initial approaches to backdoor attacks, such as BadNets [13] and Blend attacks [6], involved embedding obvious trigger patterns like square patches into the input data. These methods evolved into more covert techniques, like clean-label attacks [44], which subtly poison samples of the target class using adversarial methods without obvious label changes, enhancing their stealthiness. Recent advancements have led to even more refined attacks, like WaNet [32], which introduces triggers that are specific to individual samples.

Backdoor Defenses. Researchers have developed various defenses against backdoor attacks. These include efforts for backdoor trigger recovery [15, 18, 29, 45, 47], which focus on identifying and reverse-engineering the attacker's trigger, and strategies for backdoor model reconstruction [2, 20, 33], aimed at purging the backdoor model of its malicious elements. Methods for model detection [22, 38, 46, 48] are employed to ascertain whether a model has been tainted with backdoor samples. Backdoor sample detection evaluates whether a given sample triggers backdoor behavior in a model. Spectral Signatures (SS) [43] employs deep feature statistics to differentiate between clean and backdoor samples, but its robustness weakens with varying poisoning rates [16]. STRIP [11] blends potentially backdoored samples with a small subset of clean samples and then using the entropy of the predictions for detection. Scale-up (SCP) [14] identifies and filters malicious testing samples by analyzing their prediction consistency during pixel-wise amplification. These methods primarily concentrate on altering input data, uncovering input masks, or distinguishing the feature representations of backdoor and benign samples.

Nevertheless, these methods consider varying inputs and often experience low detection rates of backdoor training data or high error rates in identifying clean training data, as

both clean and backdoor features can either remain intact or disappear when inputs are scaled. Our paper highlights the shortcomings of relying on input data variability and introduces a novel detection method that leverages model-level uncertainty, thereby surpassing the performance of methods based on input uncertainty.

3. Preliminaries

3.1. Backdoor Attacks and Our Objective

Backdoor attacks in machine learning involve embedding a covert behavior into a neural network during training. This is typically done by poisoning the training dataset $\mathcal{D}^{tr}$ with a set of malicious examples $\mathcal{D}^b$, such that the poisoned training dataset becomes $\mathcal{D}^c \cup \mathcal{D}^b$, where $\mathcal{D}^c$ represents the clean part of the training dataset. The objective function for training a model with a poisoned training dataset can be represented as $\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}^c \cup \mathcal{D}^b; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the model parameters and $\mathcal{L}$ is the loss function. The model behaves normally on standard inputs but produces specific, attacker-chosen target label y_t when a particular trigger is present. Such vulnerabilities pose a serious risk, especially in applications where model integrity is critical. Our objective is to maximize the detection of backdoor instances in $\mathcal{D}^b$ while minimizing the instances in $\mathcal{D}^c$ falsely identified as backdoor data.

3.2. Threat Model

In our framework, we consider distinct capabilities and objectives for the attacker and defender within a black-box context. These roles are outlined as follows:

Attacker's Capabilities and Objectives. The attacker has the ability to poison the training dataset but lacks insight into the training process itself. The primary objective is to manipulate the training data so that the model being trained exhibits erroneous behavior during testing when a specific trigger is present in the input, while maintaining standard performance on benign inputs.

Defender's Capabilities and Objectives. The primary goal of the defender is to ascertain which training data samples have been compromised by backdoor poisoning. In this scenario, the defender has full control over the training process. **Given a suspicious poisoned dataset, the defender is allowed to freely use it to train the model, adopting any model architecture and training strategies**. The defender lacks prior information regarding several key aspects: the existence of backdoor samples within the dataset, the proportion of these poisoned samples, the nature of the attack (including the trigger pattern and target label), and the specific class from which the backdoor samples originate. Additionally, we also assume that the defender possesses a limited set of extra label-free clean validation data, and it is also prevalent in many prior works that study backdoor defenses [14, 27, 28].

3.3. Dropout Layers in Neural Networks

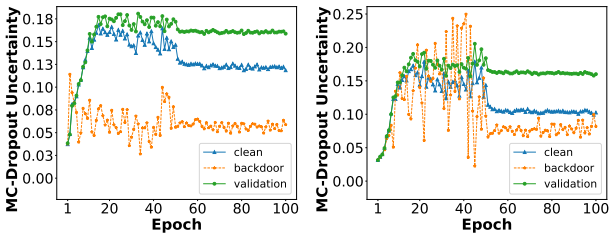
Dropout is a regularization technique that mitigates overfitting in neural networks. It randomly deactivates a subset of neurons during training, which can be mathematically described as $\mathbf{h}' = \mathbf{h} \odot \mathbf{m}$, where $\mathbf{h}$ is the output vector of a layer, $\mathbf{m}$ is a binary mask vector where each element is independently drawn from a Bernoulli distribution with probability p (dubbed dropout rate), and $\odot$ denotes element-wise multiplication. During training, the expected output of a neuron is scaled by p, as only a fraction of the neurons are active. In many practical implementations, all neurons are active during inference, but their outputs are scaled by p to account for the larger active network, ensuring consistency between the training and inference phases. In our study, inference-phase dropout is implemented.

4. Method

In this section, we offer a new perspective on the inherent predictive uncertainty within the model for the vulnerability of DNNs to backdoor attacks. We begin with two pilot studies to explore the predictive uncertainty of the model on the clean data and backdoor data. Then, we present our Prediction Shift Backdoor Detection (PSBD) method.

4.1. A Spark of Inspiration: MC-Dropout Predictive Uncertainty

The model uncertainty is a metric that measures the extent to which the model's predictions can be trusted, and can be understood as what a model does not know. One is mainly interested in the model uncertainty that is propagated onto a prediction, the so-called predictive uncertainty [12]. Previous work has indicated that backdoor data contains robust features and is potentially easier to learn compared to clean data [26, 46]. Therefore, we expect that the model should exhibit lower uncertainty towards backdoor data compared to clean data.



(a) BadNets (b) WaNet

Figure 2. The average MC-Dropout uncertainty of clean training data, backdoor training data, and clean validation data under poisoned models.

We use a widespread model predictive uncertainty approximation method - Monte Carlo Dropout (MC-Dropout) [9] to explore the model predictive uncertainty of the three types of data - the clean training data, the backdoor training data, and the clean validation data. MC-Dropout activates dropout during inference, allowing multiple forward passes. The model's final predicted confidence is the average of these passes, while the standard deviation of the highest confidence class indicates predictive uncertainty.

When applying MC-Dropout, clean validation data should show the highest uncertainty, as it's unseen during training. Clean training data should follow closely, with a smaller gap between them than between clean and backdoor training data. Backdoor data should have the lowest uncertainty, as it's easier for the model to learn. If these patterns hold, backdoor training data can be treated as outliers, enabling their detection using outlier methods.

Settings. We adopt BadNets and WaNet as examples for our discussion. We conduct experiments on the CIFAR-10 dataset [23] and ResNet-18 [17], trained for 100 epochs. For both attacks, we set the poisoning ratio to 10%, i.e. replaced 10% of total training data with malicious backdoor training data. Without sacrificing generality, the target class y_t of backdoor data is class 0 in our all examples. We randomly select the clean validation dataset from the original CIFAR-10 test set, and its size is 5% of the total size of the training set. We calculate the average MC-Dropout uncertainty of three types of data with models obtained from all 100 epochs to compare the difference in their uncertainty.

Results. Figure 2a shows that, over 100 epochs, the average uncertainty of backdoor training data under BadNets is significantly lower than that of clean training and validation data, with this difference stabilizing in later training stages. This aligns with our expectation that backdoor examples have smaller uncertainty. However, in Figure 2b, the uncertainty of backdoor data under WaNet sometimes matches or exceeds that of clean data. Additional results are available in Appendix A.1. We also tested a variant of the MC-Dropout method, which showed some improvement in detection but still failed in certain cases (details in Appendix A.2). These findings suggest that using uncertainty based on standard deviation may be insufficient for detecting backdoor data across different attack scenarios. Additionally, determining the appropriate dropout rate p is challenging without detailed knowledge of backdoor attacks.

4.2. The Enlightening Eureka Moment: Prediction Shift Phenomenon

Contrary to the indications from pilot study, relying solely on the simple MC-Dropout predictive uncertainty proves insufficient for distinguishing between clean and backdoor data. Although frustrating, we can still observe that the model's mapping from trigger to target label in backdoor data is more salient and robust compared to general image features. Informed by these preliminary findings, we delved further into the impact of employing dropout during the model inference phase on the model's behavior.

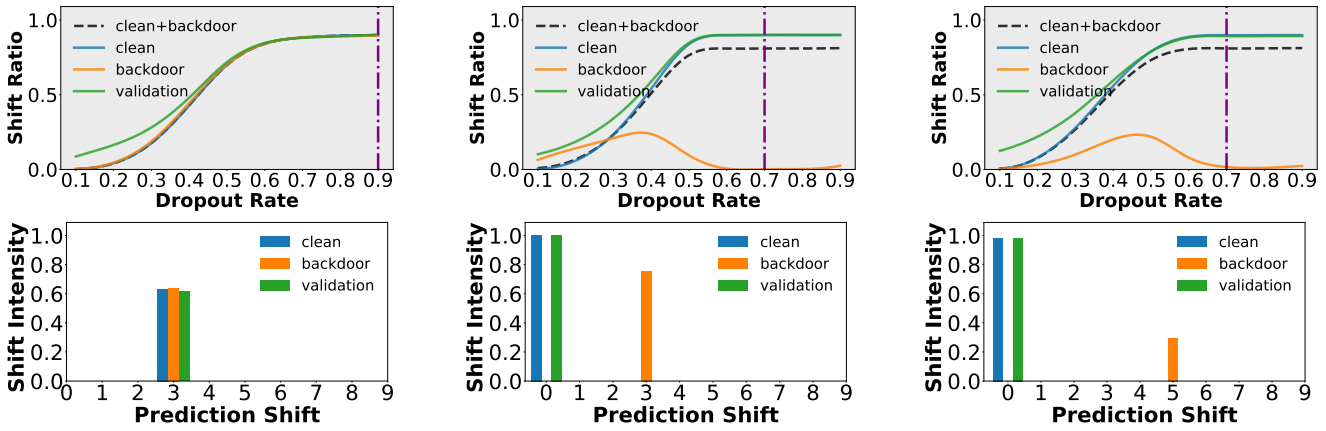
Prediction Shift. To delve deeper into how dropout affects the predictive uncertainty of the model, we examined how enabling dropout during the model's forward process alters the model's classifications and prediction confidence. We define Prediction Shift (PS) as the phenomenon where the class predicted by the model changes before and after dropout is enabled, for samples $\mathbf{x}$ in the dataset $\mathcal{D}$. The shift ratio σ represents the frequency of PS occurring in all forward inferences with dropout activated across the dataset $\mathcal{D}$, i.e.,

$$\begin{aligned} \phi_{PS}(\mathbf{x}) &= \mathbb{I}\left(\mathcal{Y}(\mathbf{x};\boldsymbol{\theta}) \neq \mathcal{Y}(\mathbf{x};\boldsymbol{\theta}^{'})\right), \\ \sigma(\mathcal{D}) &= \frac{1}{k|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \phi_{PS}(\mathbf{x}) \end{aligned} \tag{1}$$

where $\mathcal{D}$ represents an arbitrary dataset, which could encompass the entire training set or a specific subset, such as one class of data or a poisoned/clean training set; $\mathcal{Y}(\mathbf{x};\boldsymbol{\theta})$ represents the predicted class of the model $\boldsymbol{\theta}$ without dropout for input $\mathbf{x}$ and $\mathcal{Y}(\mathbf{x};\boldsymbol{\theta}^{'})$ corresponds to the predicted class of model $\boldsymbol{\theta}^{'}$ with dropout in forward inference stage; $\phi_{PS}(\cdot)$ denotes the PS function; k denotes the number of forward iterations performed with dropout.

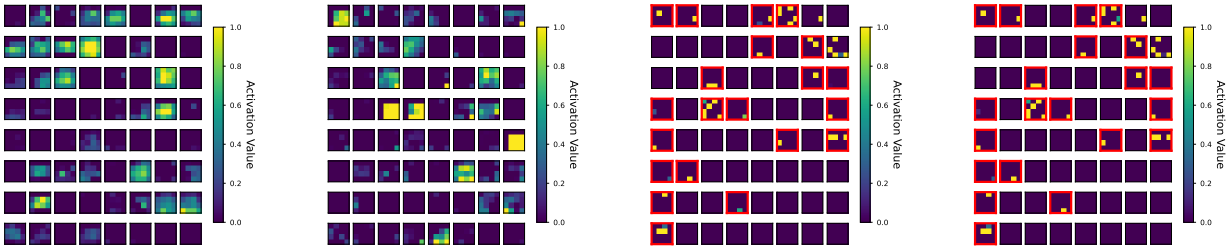
Settings. Firstly, the model is trained on the poisoned training set following the standard training procedure, which excludes the use of dropout, data augmentation, and data normalization. After that, we apply dropout without using data augmentation and data normalization during model inference. This allows us to completely control the model's ability to extract data features, thereby influencing the uncertainty of its predictions by adjusting the dropout rate p. We perform forward inference $k = 3$ times and record the value of PS in the three types of data. Specifically, dropout layers are applied after each residual connection in the residual basic block, before the activation function, as this can significantly influence the model's predictions with the dropout.

Results. In the bottom row of Figure 3a, about 60% of clean training data that experience Prediction Shift (PS) under the benign model shift to class 3. The x-axis shows the shifted labels, while the y-axis represents the intensity of the shift—the proportion of times a sample was predicted to a particular class during PS. This pattern is also seen in backdoor training and clean validation data, suggesting



(a) Benign Model (b) BadNets (c) WaNet

Figure 3. The above row shows the shift ratio curves for the benign model, BadNets model, and WaNet model, respectively. The below row represents the prediction shift intensity for samples exhibiting PS phenomenon at the chosen p. The purple vertical dash line corresponds to the selected p using our adaptive selection strategy.



(a) Clean image w/o dropout (b) Backdoor image w/o dropout (c) Clean image w/ dropout (d) Backdoor image w/ dropout

Figure 4. The first 64 feature maps out of the 512 extracted by the top layer of the model. The red boxes represent the feature map values are non-zero and the difference between each activation value in the clean and backdoor feature maps is no greater than 1. The features of clean and backdoor image become almost identical with dropout, verifying the existence of neuron bias effect.

that PS is a universal characteristic of DNNs. In the top row of Figure 3a, both clean and backdoor training data exhibit similar shift ratio trends, supporting the conclusion from Section 4.1 that the benign model treats backdoor data as perturbed clean data, classifying them mainly based on natural image features.

As illustrated in Figure 3b and 3c, in the BadNets and WaNet scenarios, we observe that the shift ratio curve for clean data still follows an increasing trend as p increases, eventually stabilizing. However, when p reaches a certain special value, the σ for backdoor data approaches 0, while the σ for clean data reaches a relatively high value (around 0.8). The most important thing is, among the samples experiencing PS, almost all clean data shifts to the target class y_t (class 0 in our experiments). The same phenomenon has been observed in other attack scenarios on CIFAR-10 as well. This indicates that training with backdoor samples enhances the PS phenomenon of clean data while suppresses that of backdoor data. This is likely due to significant differences in the internal behavior of the model towards clean data and backdoor data under appropriate dropout p. Other poisoned model's results can be found in the Appendix B.1.

In addition, we continued to observe similar patterns on the expanded and intricate Tiny ImageNet dataset. However, the shift classes of the clean training data and the clean validation data exhibit a predominant inclination towards a certain class rather than the target class. Nevertheless, there is still a certain proportion that exhibits a bias towards the target class. Despite assuming that the defender can freely choose the model architecture, we also conducted experiments using the VGG[39] to demonstrate that our method is not dependent on any specific model architecture. For all results, please refer to the Appendix B.2 and B.3.

"Neuron Bias" - An Explanation to Prediction Shift. We posit that the PS phenomenon arises from the neuron bias effect in the network during training, where neurons become predisposed to features highly representative of cer-

tain classes. This bias intensifies as the network establishes strong associations between specific data features and particular classes, especially in the case of backdoor features linked to the target class. In the absence of dropout, backdoored models typically predict the correct class for clean data, as they possess sufficient features to make accurate predictions. However, under dropout conditions, many key distinguishing features in clean data are discarded. Consequently, the model relies more heavily on the neuron bias established during training, leading it to classify clean data to the label associated with this bias. In contrast, the model learns backdoor data patterns more effectively and rapidly, resulting in a more stable and pronounced neuron bias. This enhanced bias allows the model to correctly classify backdoor data even when some features are omitted due to dropout.

To validate our hypothesis, we analyzed the features extracted by the BadNets model from both clean image and its corresponding backdoor version, comparing the results with and without the application of dropout. We presented the first 64 feature maps out of the 512 extracted by the top layer of the model. As illustrated in Figure 4a and 4b, without the dropout, the features of clean and backdoor version exhibit minimal similarity, which partly explains the model's distinct behavior towards these two types of data. However, under an appropriate dropout rate, Figure 4c and 4d clearly shows that the features of clean and backdoor version become almost identical with dropout. The red boxes in the figure highlight regions where the feature map values are non-zero and the difference between each activation value in the corresponding feature maps is no greater than 1. This finding successfully confirms the validity of our neuron bias effect hypothesis. Detailed results are available in Appendix C.

4.3. From Insight to Innovation: Prediction Shift Backdoor Detection

Even with dropout enabled, the predicted labels for some clean data remain unchanged before and after applying dropout, although their prediction confidence changes significantly. To quantify the change in prediction confidence rather than the change in labels as defined in Equation (1), we introduce a new and more fine-grained measure of predictive uncertainty, Prediction Shift Uncertainty (PSU). PSU computes the difference between the predicted class confidence without dropout and the average predicted class confidence across k dropout inferences to quantify the intensity of PS:

$$\begin{aligned} \phi_{PSU}(\mathbf{x}) &= P_c(\mathbf{x};\boldsymbol{\theta}) - \frac{1}{k}\sum_{i=1}^{k} P_c(\mathbf{x};p,\boldsymbol{\theta}_i^{'}), \\ c &= \arg\max_{c\in\mathcal{C}} P(\mathbf{x};\boldsymbol{\theta}) \end{aligned} \tag{2}$$

where c represents the class with the highest predicted confidence for data $\mathbf{x}$ without dropout during the inference stage; $P_c(\mathbf{x};\boldsymbol{\theta})$ represents the predicted confidence of class c by the model without using dropout for input $\mathbf{x}$, and $P_c(\mathbf{x};p,\boldsymbol{\theta}_i^{'})$ corresponds to the confidence with dropout at the ith forward pass; $\boldsymbol{\theta}$ represents the origin model parameters; $\boldsymbol{\theta}_i^{'}$ represents the ith dropout model parameters across all k inferences. Here p is the dropout rate.

Similar to pilot studies, the optimal dropout rate p is a crucial factor in the dropout-based uncertainty method and is challenging to determine without knowledge of backdoor attacks. A reasonable p is selected when the PS of clean data achieves a relatively strong intensity, and that of backdoor data remains relatively weak. However, due to a lack of backdoor knowledge, we cannot directly compute the PS of backdoor data. Thus, based on the definition of σ provided in Equation (1), we propose an adaptive selection strategy for p. Specifically, we identify the p where the σ of clean validation data approach to a high value (0.8 in our experiments), while the difference between the σ of the entire training data and that of the clean validation data reaches its maximum.

Prediction Shift Backdoor Detection. As we mentioned above, clean data always shift from the origin prediction class to another specific class, while backdoor data often remain static. Consequently, the PSU of clean training data and clean validation data will be close and high, whereas the PSU of backdoor data will be small under an appropriate p. For suspicious data $\mathbf{x}$, it can be determined as malicious based on a defender-specified threshold T. If $PSU(\mathbf{x}) < T$, it is classified as a backdoor sample. We set T based on the close proximity of PSU values between clean training data and extra clean validation data. In other words, in the absence of knowledge regarding the backdoor attack, T can be roughly regarded as the tolerable loss rate for clean training data. In all our experiments, T is set to the 25th percentile PSU value of the whole clean validation data. Furthermore, we found that using data augmentation in model training significantly outperforms the non-augmented training approach on Tiny ImageNet. It indicates that the use of data augmentation can intensify neuron bias, especially when the model has a lack of generalization ability to recognize the more sophisticated features. Hence, we incorporate data augmentation during model training when the model's generality is lacking. The specific workflow of the prediction shift backdoor detection (PSBD) method is given as follows:

- First, we train the model using a standard supervised learning algorithm on the suspicious training dataset, employing common data augmentation techniques when the model lacks generalization ability.
- Next, we select the dropout rate p based on the adaptive selection strategy. Then, we select a late-stage model to calculate the PSU values for the suspicious training data and clean validation data, due to its enhanced data fitting capability and robust neuron bias paths.

- Finally, for suspicious data $\mathbf{x}$, we can determine it is malicious based on defender-specified threshold T. If $PSU(\mathbf{x}) < T$, we view it as a backdoor sample. T is set to the 25th percentile value of the PSU of clean validation data in all our experiments.

5. Experiments

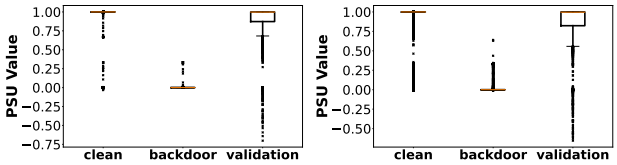
5.1. Experiment Settings

Dataset and DNN Model. We conduct all experiments on the CIFAR-10 [23], GTSRB [40] and Tiny ImageNet [37] datasets using the ResNet-18 [17] architecture. **Please note that a defender is free to choose any architecture, as the sole objective is to detect any potential backdoor data that may exist within the dataset.** We randomly select 5% of the total quantity of whole poisoned training dataset from the original test sets as our extra clean validation data. Further details can be found in the Appendix D.1.

Backdoor Attack Settings. We evaluate our PSBD method against seven representative backdoor attacks, namely BadNets [13], Blend [6], TrojanNN[28], Label-Consistent [44], WaNet [32], ISSBA [25] and Adaptive-Blend [35]. We examined two scenarios with poisoning ratios of 5% and 10%. The main paper discusses the 10% poisoning ratio in detail, while the results for the 5% poisoning ratio are presented in Appendix E. Without sacrificing generality, in all our experiments, the target class y_t is set to class 0. Data augmentation during the model training was employed exclusively for Adaptive-Blend on CIFAR-10, GTSRB, and all experiments on Tiny ImageNet to achieve an attack success rate exceeding 85%. More detailed settings are presented in Appendix D.2. We also verify the robustness of PSBD against potential adaptive attacks in Appendix F.

Backdoor Detection Baseline. We compare PSBD with five classic and state-of-the-art backdoor data detection methods, namely Spectral Signature(SS) [43], Strip [11], Spectre [16], SCAN [42], SCP [14] and CD-L [19]. All six methods were implemented and evaluated on the CIFAR-10 dataset. For GTSRB and Tiny ImageNet datasets, SCAN was excluded due to its computationally intensive matrix eigenvalue computations, which significantly increased processing time. We run 10 trials for each experiment of all methods and report the average results across all cases as the final result. We found that the variances are relatively small, so we ignored them. Please refer to the Appendix D.3 for the implementation details.

Metric. To assess the effectiveness of detection methods, we employ common classification metrics: True Positive Rate (TPR) and False Positive Rate (FPR). Our evaluation prioritizes achieving a high TPR to ensure effective identification of backdoor samples, while simultaneously maintaining a low FPR to minimize erroneous deletion of clean samples. Values inside brackets represent standard deviations (SD). Moreover, the results for the area under receiver operating curve (AUROC) can be found in the Appendix G.



(a) BadNets (b) WaNet

Figure 5. The PSU values of BadNets and WaNet in CIFAR-10. The poisoning ratio is 10%. PSBD exhibits strong capability to effectively differentiate clean data from backdoor data.

5.2. Experiment Results

Effectiveness and Efficiency of PSBD. As shown in Table 1, PSBD demonstrates excellent backdoor detection performance across a wide range of attack scenarios, while effectively preserving a substantial amount of clean data. The results also demonstrate that PSBD achieves a substantial improvement in detection performance compared to the defense baselines. *In contrast, all baseline methods fail in some evaluation attacks.*

Specifically, on the CIFAR-10 dataset, the Spectral Signature method failed to detect backdoor data under all attack scenarios, while simultaneously misclassifying a substantial amount of clean data as backdoor data. This suggests that when the trigger pattern is relatively large or complex, the spectral signature property may be difficult to capture. The Spectre method demonstrated relatively effective detection capabilities across most attack scenarios, but it also has a high FPR. This is not desirable in practice as it filters out a significant amount of clean training data, which can lead to issues like overfitting due to insufficient training data. Although the SCAN, Strip, SCP, and CD-L methods exhibited relatively acceptable performance across most attack scenarios, achieving a relatively high TPR and a low FPR, their effectiveness deteriorated when confronted with attacks employing confusion strategy such as WaNet and Adaptive-Blend. This confusion strategy aims to disrupt the model by retaining an equal or greater proportion of confounding samples that contain the trigger pattern but are correctly labeled. On the GTSRB dataset, which contains a larger number of classes, the Spectral Signature and Spectre methods completely fail to detect backdoor data. The Strip, SCP, and CD-L methods exhibit good detection performance only in a few specific attack scenarios. In contrast, our PSBD method demonstrates strong detection capabilities across all attack scenarios, highlighting its robustness and generalizability.

On the more challenging Tiny ImageNet, all baseline methods failed in most attack scenarios. This failure is likely due to the increased complexity of image features, which weakened their ability to capture the mapping be-

Table 1. The performance (TPR/FPR) on CIFAR-10, GTSRB and Tiny ImageNet. We mark the **best result** in boldface while the value with underline denotes the second-best. The failed cases (i.e., TPR < 0.8) are marked in gray. Adaptive-Blend attack has a 1%/1%/2% poisoning ratio on CIFAR-10/GTSRB/Tiny ImageNet, while other attacks have a 10% poisoning ratio. OOT indicates that the method did not finish within the allocated time limit.

Defenses→ Attacks↓	PSBD (**Ours**)	SS	Strip	Spectre	SCAN	SCP	CD-L
CIFAR-10							
Badnet	1.000/0.104	0.389/0.512	1.000/0.113	0.953/0.450	**1.000/0.009**	1.000/0.205	0.998/0.158
Blend	**1.000/0.135**	0.438/0.507	0.993/0.118	0.953/0.450	0.991/0.000	0.939/0.244	0.976/0.156
TrojanNN	0.983/0.171	0.302/0.509	0.996/0.112	0.950/0.450	**1.000/0.000**	0.921/0.227	0.999/0.161
Label-Consistent	0.992/0.130	0.447/0.506	**0.994/0.117**	0.953/0.450	0.979/0.014	0.889/0.237	0.962/0.159
WaNet	**1.000/0.116**	0.456/0.505	0.050/0.101	0.951/0.450	0.891/0.034	0.869/0.251	0.863/0.144
ISSBA	**1.000/0.113**	0.436/0.507	0.774/0.120	0.950/0.450	0.963/0.011	0.939/0.290	0.965/0.157
Adaptive-Blend	**0.982/0.184**	0.608/0.145	0.014/0.069	0.753/0.144	0.000/0.023	0.721/0.257	0.432/0.167
Average	**0.994/0.136**	0.439/0.456	0.689/0.107	0.923/0.406	0.832/0.013	0.899/0.244	0.855/0.157
GTSRB							
Badnet	0.987/0.202	0.476/0.502	0.999/0.096	0.524/0.497	OOT	**1.000/0.344**	0.911/0.193
Blend	**0.910/0.207**	0.476/0.502	0.897/0.093	0.524/0.497	OOT	0.286/0.337	0.462/0.199
TrojanNN	0.952/0.212	0.476/0.502	0.639/0.096	0.524/0.497	OOT	0.113/0.345	**0.967/0.194**
Label-Consistent	0.944/0.203	0.476/0.502	**1.000/0.115**	0.524/0.497	OOT	0.998/0.362	0.416/0.175
WaNet	**0.996/0.115**	0.476/0.502	0.037/0.109	0.524/0.497	OOT	0.129/0.306	0.031/0.182
ISSBA	**0.999/0.211**	0.476/0.502	0.725/0.092	0.524/0.497	OOT	0.584/0.339	0.705/0.197
Adaptive-Blend	**0.899/0.194**	0.299/0.392	0.004/0.094	0.750/0.388	OOT	0.071/0.332	0.028/0.158
Average	**0.955/0.192**	0.451/0.486	0.614/0.099	0.556/0.481	OOT	0.454/0.338	0.503/0.185
Tiny ImageNet							
Badnet	0.989/0.088	0.480/0.502	0.841/0.108	0.522/0.497	OOT	**0.999/0.271**	0.462/0.176
Blend	**0.919/0.108**	0.478/0.502	0.249/0.086	0.522/0.496	OOT	0.551/0.260	0.874/0.175
TrojanNN	0.961/0.222	0.478/0.502	0.963/0.104	0.522/0.497	OOT	0.972/0.301	**0.985/0.150**
Label-Consistent	0.839/0.039	0.478/0.502	0.460/0.088	0.522/0.497	OOT	0.741/0.187	**0.931/0.203**
WaNet	**0.959/0.086**	0.478/0.502	0.087/0.082	0.522/0.497	OOT	0.446/0.254	0.577/0.151
ISSBA	0.886/0.209	0.478/0.502	0.954/0.097	0.522/0.497	OOT	0.691/0.297	**0.978/0.137**
Adaptive-Blend	**0.949/0.095**	0.392/0.502	0.210/0.099	0.621/0.497	OOT	0.651/0.190	0.331/0.176
Average	**0.929/0.121**	0.466/0.502	0.538/0.095	0.536/0.497	OOT	0.722/0.251	0.734/0.167

tween trigger pattern and target label. Encouragingly, our PSBD method maintained its effectiveness, ranking in the top two for all attacks except for TrojanNN (where it still achieved a TPR of 0.961) and ISSBA. The slight performance degradation under ISSBA is likely due to the model's insufficient ability to extract features from the data. This is reflected in the significantly lower clean accuracy of the model compared to that of the clean model. The clean accuracy of models can be found in the Appendix Table A3. By employing dropout to diminish prominent image features and utilizing robust neuron bias paths, PSBD effectively discerned the mapping from trigger pattern to target label.

The Strong Discriminative Capability of PSBD. PSBD excels in its critical ability to effectively differentiate between clean and backdoor training data. By leveraging the PSU values, we have developed some informative box plots that clearly and vividly demonstrate the remarkable discriminative power of our approach. As shown in Figure 5, provide a visual representation of how PSBD separates clean data from backdoor data. In these plots, it is prominently visible that backdoor data is characterized by lower PSU values, distinguishing it from clean training and validation data, which generally exhibit higher PSU values. This distinction is crucial for effective backdoor detection, as it highlights the different behavioral patterns of the model when exposed to clean versus poisoned data. The lower PSU values in backdoor data indicate the model's ability to maintain confident predictions, a direct consequence of the embedded trigger in these samples with neuron bias effect.

6. Conclusion

In our study, we developed PSBD, a simple and effective method to detect backdoor samples in the training dataset by focusing on Prediction Shift phenomenon under dropout conditions, leading to the concept of neuron bias effect. By analyzing changes in prediction confidence with and without dropout, PSBD effectively distinguishes between clean and backdoor data across multiple datasets and attack types. This research contributes a practical and effective solution to the challenge of backdoor attacks in DNNs, marking a notable advancement in the field of neural network security. Future efforts could explore extending the PSBD method to a broader range of domains, such as natural language processing or time-series analysis.

Acknowledgement

This work is supported by the NSF under Grants 2246157 and 2319243. We are thankful for the computational resources made available through NSF ACCESS and Argonne Leadership Computing Facility.

References

[1] Jianing Bai, Ren Wang, and Zuyi Li. Physics-constrained backdoor attacks on power system fault localization. In *2023 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2023. 1

[2] Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3855–3859. IEEE, 2021. 1, 2

[3] Ruisi Cai, Zhenyu Zhang, Tianlong Chen, Xiaohan Chen, and Zhangyang Wang. Randomized channel shuffling: Minimal-overhead backdoor attack detection without clean datasets. *Advances in Neural Information Processing Systems*, 35: 33876–33889, 2022. 1

[4] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018. 2

[5] Pin-Yu Chen and Cho-Jui Hsieh. *Adversarial robustness for machine learning*. Academic Press, 2022. 1

[6] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 2, 7, 15, 16

[7] Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to backdoor diffusion models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4015–4024, 2023. 1

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 19

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 4, 11

[10] Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, Yi Yang, Shangwei Guo, and Chun Fan. Triggerless backdoor attack for nlp tasks with clean labels. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2942–2952, 2022. 1

[11] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 113–125, 2019. 2, 7, 16

[12] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021. 3

[13] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 1, 2, 7

[14] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. *arXiv preprint arXiv:2302.03251*, 2023. 2, 3, 7, 11, 16, 19

[15] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763*, 2019. 2

[16] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: Defending against backdoor attacks using robust statistics. In *International Conference on Machine Learning*, pages 4129–4139. PMLR, 2021. 2, 7, 16

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4, 7, 11

[18] Xiaoling Hu, Xiao Lin, Michael Cogswell, Yi Yao, Susmit Jha, and Chao Chen. Trigger hunting with a topological prior for trojan detection. *arXiv preprint arXiv:2110.08335*, 2021. 2

[19] Hanxun Huang, Xingjun Ma, Sarah Erfani, and James Bailey. Distilling cognitive backdoor patterns within an image. *arXiv preprint arXiv:2301.10908*, 2023. 2, 7, 16

[20] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. *arXiv preprint arXiv:2202.03423*, 2022. 1, 2

[21] Haibo Jin, Ruoxi Chen, Jinyin Chen, Haibin Zheng, Yang Zhang, and Haohan Wang. Catchbackdoor: Backdoor detection via critical trojan neural path fuzzing. In *European Conference on Computer Vision*, pages 90–106. Springer, 2025. 19

[22] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 301–310, 2020. 2

[23] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report TR-2009*, 2009. 4, 7, 11

[24] Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. Controlnet++: Improving conditional controls with efficient consistency feedback:

Project page: liming-ai. github. io/controlnet_plus_plus. In *European Conference on Computer Vision*, pages 129–147. Springer, 2024. 1

[25] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16463–16472, 2021. 7, 12

[26] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021. 1, 3

[27] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021. 3

[28] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018. 2, 3, 7, 15

[29] Yingqi Liu, Guangyu Shen, Guanhong Tao, Zhenting Wang, Shiqing Ma, and Xiangyu Zhang. Complex backdoor detection by symmetric feature differencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15003–15013, 2022. 2

[30] Hanxiao Lu, Zeyu Huang, and Ren Wang. Purification of contaminated convolutional neural networks via robust recovery: An approach with theoretical guarantee in one-hidden-layer case. *arXiv preprint arXiv:2407.11031*, 2024. 1

[31] Khan Muhammad, Amin Ullah, Jaime Lloret, Javier Del Ser, and Victor Hugo C de Albuquerque. Deep learning for safe autonomous driving: Current challenges and future directions. *IEEE Transactions on Intelligent Transportation Systems*, 22 (7):4316–4336, 2020. 1

[32] Tuan Anh Nguyen and Anh Tuan Tran. Wanet-imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021. 1, 2, 7, 11, 16

[33] Soumyadeep Pal, Ren Wang, Yuguang Yao, and Sijia Liu. Towards understanding how self-training tolerates data backdoor poisoning. *arXiv preprint arXiv:2301.08751*, 2023. 1, 2

[34] Soumyadeep Pal, Yuguang Yao, Ren Wang, Bingquan Shen, and Sijia Liu. Backdoor secrets unveiled: Identifying backdoor data with optimized scaled prediction consistency. In *International Conference on Learning Representations*, 2024. 2

[35] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *The eleventh international conference on learning representations*, 2022. 7, 11, 16

[36] Jie Qin, Jie Wu, Pengxiang Yan, Ming Li, Ren Yuxi, Xuefeng Xiao, Yitong Wang, Rui Wang, Shilei Wen, Xin Pan, et al. Freeseg: Unified, universal and open-vocabulary image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19446–19455, 2023. 1

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 7, 14

[38] Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang. Backdoor scanning for deep neural networks through k-arm optimization. In *International Conference on Machine Learning*, pages 9525–9536. PMLR, 2021. 1, 2

[39] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5, 14

[40] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011. 7, 15

[41] Barathi Subramanian, Jeonghong Kim, Mohammed Maray, and Anand Paul. Digital twin model: A real-time emotion recognition system for personalized healthcare. *IEEE Access*, 10:81155–81165, 2022. 1

[42] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of {DNNs} for robust backdoor contamination detection. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1541–1558, 2021. 7, 16

[43] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *arXiv preprint arXiv:1811.00636*, 2018. 2, 7, 16

[44] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. 2, 7, 11

[45] 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. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019. 2

[46] Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical detection of trojan neural networks: Data-limited and data-free cases. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 222–238. Springer, 2020. 1, 2, 3

[47] Zhen Xiang, David J Miller, and George Kesidis. Post-training detection of backdoor attacks for two-class and multi-attack scenarios. *arXiv preprint arXiv:2201.08474*, 2022. 2

[48] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 103–120. IEEE, 2021. 1, 2

[49] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. *arXiv preprint arXiv:2005.00060*, 2020. 1