UNIT: Backdoor Mitigation via Automated Neural Distribution Tightening

Siyuan Cheng[1]*, Guangyu Shen[1]*, Kaiyuan Zhang[1], Guanhong Tao[1], Shengwei An[1], Hanxi Guo[1], Shiqing Ma[2], and Xiangyu Zhang[1]

[1] Purdue University, West Lafayette, IN, 47906, USA
{cheng535,shen447,zhan4057,taog,an93,guo778,xyzhang}@cs.purdue.edu
[2] University of Massachusetts at Amherst, MA, 01003, USA
shiqingma@umass.edu
* *denotes equal contribution*

Abstract. Deep neural networks (DNNs) have demonstrated effectiveness in various fields. However, DNNs are vulnerable to backdoor attacks, which inject a unique pattern, called trigger, into the input to cause misclassification to an attack-chosen target label. While existing works have proposed various methods to mitigate backdoor effects in poisoned models, they tend to be less effective against recent advanced attacks. In this paper, we introduce a novel post-training defense technique UNIT that can effectively eliminate backdoor effects for a variety of attacks. In specific, UNIT approximates a unique and tight activation distribution for each neuron in the model. It then proactively dispels substantially large activation values that exceed the approximated boundaries. Our experimental results demonstrate that UNIT outperforms 7 popular defense methods against 14 existing backdoor attacks, including 2 advanced attacks, using only 5% of clean training data. UNIT is also cost efficient. The code is accessible at https://github.com/Megum1/UNIT.

Keywords: Deep Neural Networks · Mitigation of Backdoor Attacks

1 Introduction

As deep learning (DL) continues to reshape industries, spanning from transportation to healthcare, the practical impact of DL is becoming increasingly apparent. However, DL faces significant security issues, particularly backdoor attacks. Backdoor attacks typically embed a unique pattern (the backdoor trigger) into the training data, which establish a correlation between this pattern and a specific target label. Consequently, a model trained on such data misclassifies inputs containing the trigger as the target label. Researchers have proposed a range of backdoor attacks [3,6,7,37,47,48,53], along with countermeasures aimed at detecting and mitigating backdoors in poisoned models [21,23,35,64,66,69,71,78]. However, without knowing the trigger pattern, it's challenging to accurately identify whether a model or dataset has been compromised, and the trigger pattern is typically not accessible until the attacker initiates the attack.

This paper focuses on backdoor mitigation [35,38,69]. The goal is to remove the backdoor effect in a model such that trigger-inserted inputs cannot cause the target prediction. Backdoor mitigation usually assumes access to a few clean (usually $< 5\%$) training samples without knowledge of the trigger pattern. Existing backdoor mitigation techniques [35,38,66,69,74,83] are effective against prior attacks [3,6,20,40,47,48]. However, they fall short in eliminating backdoor effects caused by advanced attacks [7,51]. This is because these methods either retrain the entire model without precise guidance for reducing backdoor effects [35,66,83] or directly prune some specific neurons [38,69]. Such coarse-grained approaches fail to counter recent advanced attacks. For instance, advanced attacks may hide backdoor behavior within benign neurons that primarily process normal features. In such cases, pruning these neurons would undesirably impact benign utility. On the other hand, retaining these neurons would preserve the backdoor behavior in the model. To address the above challenge, we propose a novel backdoor mitigation method, UNIT. It is based on the observation that, for various backdoored models, there exists a set of *backdoor neurons*, responsible for backdoor behaviors. The activation values of these neurons for poisoned inputs are significantly higher than those for clean samples. Note that backdoor neurons may also play a role in benign feature extraction. Given the absence of poisoned samples for accurately identifying backdoor neurons,, we propose to approximate a clean distribution on *each individual* neuron using a small set of clean samples. The approximation bounds the maximum activation value on each neuron. During inference, our defense UNIT clips activations with a substantially large value to the approximated boundaries. A straightforward idea is to apply a uniform percentile boundary, e.g., a threshold covering 98% values, to bound the activation for all neurons. Our result in Figure 4 (Section 4) reveals its limitation against advanced attacks, because it overlooks the fact that different neurons have various contributions. While some neurons might be fully compromised, others could remain entirely benign. To address this challenge, UNIT employs an optimization process that tailors a *unique* boundary for each neuron. The optimization is guided by a proxy accuracy measure on a small set of clean samples, serving as an approximation of the real accuracy on the test set. This is to precisely bound the accuracy degradation caused by the clipping. This approximation is generally accurate, as evidenced by a ablation study detailed in Section A.5. The process allows UNIT to meticulously tighten the boundaries to mitigate backdoor effects while ensuring the accuracy aligns with the defender's expectation.

Our main contributions are summarized as follows:

- We introduce UNIT ("*A*U*tomated* N*eural DI**stribution* T*ightening*"), an innovative backdoor mitigation method that approximates *unique* distribution boundary for *each* neuron, which is used to effectively dispel maliciously large activation caused by the backdoor.
- UNIT utilizes an optimization technique to dynamically refine and tighten unique boundaries for different neurons. This process is guided by the proxy accuracy on a few clean samples, which approximates the real test accuracy.

– Extensive experiments demonstrate UNIT's effectiveness against 14 existing attacks, including 2 advanced attacks, outperforming 7 baseline defenses. Additionally, UNIT is generalizable to different datasets, network structures, and activation functions. We further show that UNIT is resilient to 3 adaptive attacks.

Threat Model. Our threat model aligns with the existing literature [35,38,69], where the adversary provides a model that may potentially contain a backdoor to the user. The adversary holds the complete control over the training process and can deploy advanced attacks [7,51] to circumvent existing defenses. Prior to utilizing the model, the user applies defense techniques to mitigate any potential backdoor. The defender has access to a small portion (5%) of the clean training data. She has no prior knowledge of the poisoned data. The defense objective is to eliminate the backdoor effect without compromising the normal functionality, such as classification accuracy.

2 Related Work

Backdoor Attack. Recent literature has introduced a variety of backdoor attacks on image classification models. Early works [20,40] stamp static image patches on a small portion of training samples and mislabel them as the target class to poison the training dataset. Clean label attacks [52,59,65] manipulate backdoor samples in feature space and leave their labels unchanged. Recently, more sophisticated transformations are utilized as backdoor triggers [6,10,37,41,47,63]. In addition, sample-specific backdoors generate different triggers for different inputs via generative models [7,48,53], making them more stealthy and harder to detect. Backdoor attacks can also be launched in a wide range of applications as well, such as natural language processing [5,50,57], self-supervised learning [18,30], federated learning [2,77,78] and even diffusion models [1,11]. In this paper, we focus on the image classification task.

Backdoor Defense. Various defenses have been proposed from multiple perspectives to safeguard AI models against backdoor attacks. Our approach falls under the category of *Backdoor Mitigation* [35,38,61,62,69,70,74,76], which is widely acknowledged as a promising strategy. The primary goal is to cleanse the backdoor effect while retain the benign functionality of a given model. Orthogonal to this, training-time defense [23,28,34,64,68,72] defenses distinguish between poisoned and clean samples based on their internal discrepancies/behaviors and sanitize the training set. Trigger inversion [8,9,21,39,55,56,66,67] aims to detect whether a given model is poisoned or not via reverse-engineering the backdoor triggers. Running time defenses [14,19] are designed to reject samples potentially carrying triggers during model inference.

3 Limitation of Existing Backdoor Mitigation Methods

Various methods have been proposed to address the challenge of mitigating the backdoor effects in poisoned models. They primarily fall into two categories: (1)

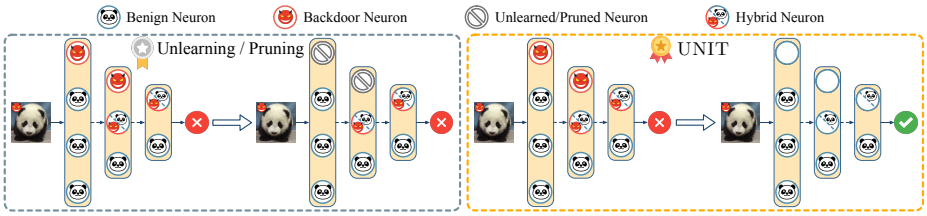


Fig. 1: Limitation of existing backdoor mitigation methods

Unlearning [35,45,66,70,74,83] and (2) Pruning [38,69,80]. Unlearning methods utilize training-based techniques, such as fine-tuning [45], distillation [35], and cloning [70], to eradicate the backdoor behaviors. These approaches are grounded in the catastrophe forgetting assumption [31], positing that neural networks naturally tend to forget specific behaviors while continuously learning other patterns. For example, NAD [35] employs standard fine-tuning to create a teacher model and then conveys only benign knowledge to the student model through knowledge distillation [25]. In contrast, pruning methods involve the identification and removal of malicious neurons. They speculate that there exists a small subset of neurons responsible for backdoor behaviors, and the removal of these neurons eliminates the backdoor impact. For instance, ANP [69] identifies malicious neurons based on sensitivity analysis on clean samples and effectively prunes them. In the following, we delve into the limitations inherent to both unlearning and pruning methods and introduce our idea to address the challenges.

Coarse-grained Repair. Recent advanced attacks [7,51] manage to conceal backdoor behavior within benign neurons, creating hybrid neurons that withstand existing mitigation methods. A prevalent limitation in current techniques lies in their coarse-grained nature, which is inadequate against advanced attacks. Essentially, these methods struggle to operate inside individual neurons to eliminate the backdoor component while preserving the benign portion. In addition, an implicit requirement in backdoor mitigation is the preservation of benign functionality. In other words, the benign accuracy of the repaired model should not suffer significant degradation. This constraint limits the efficacy of both unlearning and pruning methods. For example, pruning may either remove or leave an entire neuron untouched. When dealing with hybrid neurons, directly pruning them would significantly diminish clean classification performance. Conversely, retaining such neurons would maintain the backdoor behaviors. Figure 1 conceptually illustrates such limitation of existing methods. The left dashed box shows the mitigation of an advanced attack [7,51]. The left half presents the process of a poisoned image (depicted as a panda with a red trigger at the top-left) in a backdoored model. Notably, the model comprises three types of neurons: (1) Benign neurons (depicted as cartoon pandas) primarily extracting benign features, (2) Backdoor neurons (depicted as red devils) processing backdoor behaviors, and (3) Hybrid neurons (depicted as half panda and half devil) serving both purposes. Following the model inference, the output corresponds to the misclassified attack target

label, indicated by a red cross. The right half of the left figure portrays the model after repair through unlearning and pruning. Observe that backdoor neurons are effectively unlearned or pruned, whereas hybrid neurons, which exhibit both benign and malicious behaviors, remain unaffected. This is because eliminating these hybrid neurons could lead to a substantial decrease in accuracy for benign tasks. However, the presence of these hybrid neurons can still contribute to the persistence of a high attack success rate due to their involvement in backdoor behaviors. This highlights the limitations of current mitigation techniques.

Heavily Dependent on Meticulous Parameter Tuning. Existing approaches heavily rely on meticulous parameter tuning to achieve optimal performance against various attacks. For instance, pruning techniques demand a careful determination of the pruning rate, adjusted on a case-by-case basis. The extent of neuron removal directly influences the model's overall accuracy; excessive pruning can deteriorate performance, while insufficient pruning may not adequately counteract the backdoor effect. Our empirical analysis, detailed in Appendix A.3, highlights the pronounced sensitivity of the existing methods to parameter adjustments. This sensitivity presents a notable limitation, undermining the generalizability and practical applicability of these methods.

Our Idea: *Automated Neural Distribution Tightening.* We introduce a novel technique UNIT, which automatically approximates and tightens a unique distribution boundary for each neural activation. Subsequently during inference, it clips activation values that exceed the boundary, targeting potential backdoor activation. UNIT employs an optimization based method to automatically refine the activation boundary for individual neurons. It is guided by a proxy accuracy measured on a small set ($<5\%$) of clean samples, which approximates the real test accuracy. This approximation is generally accurate, as evidenced by a ablation study detailed in Section A.5. The process involves a dynamic adjustment of boundaries: if the observed proxy accuracy degradation is below the defender's expectation, the boundary is further tightened. Conversely, the boundary is relaxed to restore the accuracy. This ensures a balanced approach to maintaining benign accuracy while eliminating backdoors. UNIT operates with a high degree of granularity, analyzing and adjusting unique boundaries for individual neurons. The right figure in Figure 1 visualizes positive outcomes achieved through UNIT. Notably, both backdoor neurons and the backdoor portion of hybrid neurons are deactivated.

Moreover, compared with existing methods, UNIT is an automated technique that does not require *meticulous* parameter tuning. The defender is only required to specify a bound of accuracy degradation to balance benign accuracy and backdoor mitigation. The *parameter-efficient* characteristic of UNIT emphasizes the generalizability and practicality of UNIT.

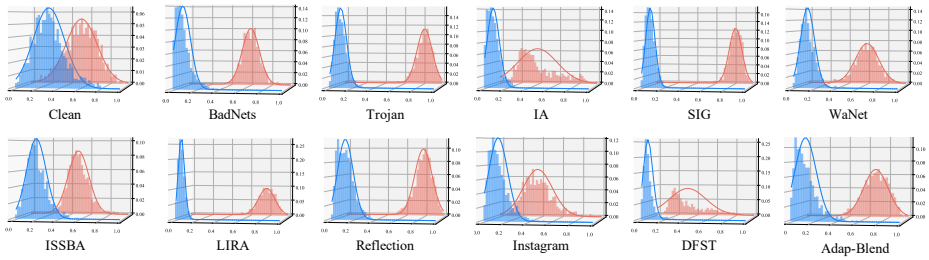


Fig. 2: Neural activation distribution for benign and poisoned samples

4 Design of UNIT

4.1 Notations

We provide formal notations of deep neural network operations before delving into our methodology. Following existing works [80, 81], we consider a typical neural network for a classification task of C classes. The dataset $\mathcal{D}$ is composed of numerous pairs $(x, y) \sim \mathcal{D}$, where each sample $x \in \mathbb{R}^d$ and its corresponding label $y \in \{1, 2, \cdots C\}$. The input dimension d can be complex, e.g., $d = d_c \times d_w \times d_h$ for RGB images, where d_c, d_w, and d_h represent the number of channels, width, and height, respectively. The training objective is to derive a classifier $M : \mathbb{R}^d \rightarrow \{1, 2, \cdots, C\}$. Consider a deep neural network consisting of L layers:

$$M = g \circ \phi \circ f^L \circ \cdots \circ \phi \circ f^l \circ \cdots \circ \phi \circ f^1, \tag{1}$$

where f^l denotes the feature extraction function at l-th layer ($1 \leqslant l \leqslant L$), ϕ represents the non-linear activation function, e.g., ReLU [46], and g is the fully connected layer following the extraction layers, responsible for aggregating features for class prediction.

Neural Activation. To analyze the internal statistics of the model, we further define the sub-network that terminates at the l-th activation layer as F^l:

$$F^l = \phi \circ f^l \circ \cdots \circ \phi \circ f^1 \tag{2}$$

Therefore, given an input sample x, its activation value at l-th layer is $F^l(x)$. This activation value is typically multi-dimensional. If the l-th layer consists of K neurons, the *neural activation* of the k-th neuron in this layer is denoted as $F^l_k(x)$.

4.2 Key Observations of Neural Activation

The backdoor behavior can be activated by the trigger on backdoored models. To illustrate how such input pattern flips the output prediction, we delve into the model internals, particularly examining the neural activation values of both clean and poisoned samples. We use the CIFAR-10 dataset and ResNet18 architecture

as our subject and visualize the neural activation distribution of a clean model and a range of backdoored models by various attacks, including BadNets [20], Trojan [40], IA [48], SIG [3], WaNet [47], ISSBA [37], LIRA [15], Reflection [41], Instagram [40], DFST [7], and Adap-Blend [51]. To gain insights into the influence of poisoned samples on model behavior, we utilize Shap [42], a deep learning interpreter, to identify 1% of the most important neurons in the 12th layer of each model when processing poisoned samples. These selected neurons, designated as *backdoor neurons*, are responsible for the backdoor behavior. Subsequently, our analysis involves comparing the activation values of these neurons across 1,000 clean and 1,000 poisoned samples. It's worth noting that as there is no predefined trigger for the clean model, we employ the BadNets trigger to generate dummy poisoned samples for analysis. By applying PCA [44] for dimensionality reduction, we visualize the neural activation distributions in Figure 2. The blue plots represent the activation distributions of clean inputs, while the red plots depict the distributions of poisoned samples. Observe that the neural activation distributions of clean and poisoned samples are indistinguishable in the clean model. Conversely, in models subjected to backdoor attacks, it is evident that there exists a large distribution shift between clean and poisoned samples. Notably, the neural activation values for poisoned inputs are significantly greater than those for clean inputs. This disparity underscores that backdoor triggers significantly change the neural activation distribution for specific *backdoor neurons*, subsequently leading to the target misclassification.

Distinguished Fine-grained Observation. Existing papers [4, 51, 64] have observed the latent separability between clean and poisoned samples, primarily focusing on the features of the *entire layer*. Nonetheless, recent advanced attacks [51] and the adaptive attacks detailed in Section A.4 manage to diminish this *layer-level* feature distinction. However, these approaches fall short in eliminating separability at the *neural activation* level, as shown in Figure 2. This highlights a clear distinction between our fine-grained observation and existing literature.

4.3 Overview of UNIT

Our observation reveals a substantial increase in neural activation compared to benign ones on backdoor neurons given poisoned samples. Building upon this insight, we introduce UNIT, a novel approach that approximates a tight benign distribution for each neuron based on a small subset of clean training data. UNIT then strategically clips activation values that surpass the distribution boundary. The necessity for this approximation stems from the unavailability of poisoned samples in typical scenarios. Hence, it is challenging to precisely identify the backdoor neurons. To deal with the problem, UNIT applies its approximation across all neurons, including both benign and backdoor ones. Furthermore, we refine the approximated benign distribution to be as tight as possible, aiming to effectively mitigate the backdoor behavior. The overview of UNIT is depicted in Figure 3, using a typical neuron as an example. The x-axis represents neural

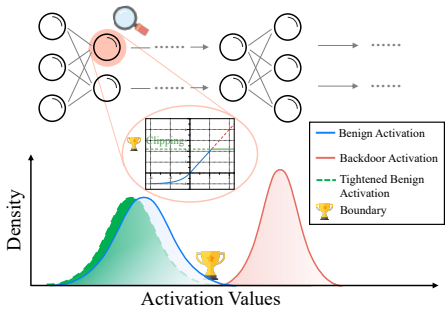


Fig. 3: Overview of UNIT

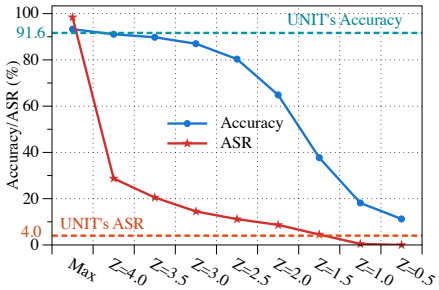


Fig. 4: Limitation of straightforward clipping

activation values for different samples, while the y-axis denotes sample density corresponding to these values. In this depiction, the benign neural activation is shown in blue, and the poisoned neural activation is in red. UNIT approximates a tight distribution based on a few clean samples, as illustrated in the green region. To mitigate the backdoor effect during model inference, UNIT constrains the neural activation values by clipping those that exceed the approximated boundary. In the zoomed-in plot, rather than allowing activation values to extend along the red lines (which represent maliciously large values), UNIT ensures that values remain within the green boundary. While this strategy might entail a minor compromise in accuracy for clean samples, it is remarkably effective in neutralizing the backdoor effects of poisoned samples, thereby enhancing the model's security and integrity.

4.4 Design Details

In this section, we formally present the design of UNIT. Specifically, we detail the process of automatically tightening the neural distribution based on a small portion of clean training samples. The goal is to effectively eliminate maliciously high neural activation, which represents the backdoor behavior.

Objective. Following the notation of neural activation in Section 4.1, we formally define the objective of UNIT. For any input x and its neural activation at l-th layer and k-th neuron $F_k^l(x)$, UNIT derives an upper bound value σ_k^l such that its neuron activation is bounded as

$$\hat{F}_k^l(x) = b_{\sigma_k^l}(F_k^l(x)) = \begin{cases} F_k^l(x) & \text{if } F_k^l(x) \leqslant \sigma_k^l, \\ \sigma_k^l & \text{otherwise.} \end{cases} \tag{3}$$

Note that σ_k^l can be a feature map when F^l is a convolution layer. Then the classifier defined in Equation 1 can be reformatted as:

$$M_\sigma = g \circ b_{\sigma^L} \circ \phi \circ f^L \circ \cdots \circ b_{\sigma^l} \circ \phi \circ f^l \circ \cdots \circ b_{\sigma^1} \circ \phi \circ f^1, \tag{4}$$

Algorithm 1 Automated Neural Distribution Tightening

Input: Subject model M, Accuracy drop expectation ϵ, Training data $\{(x_i^t, y_i^t)\}_{i=1}^{n_t}$, Validation data $\{(x_i^v, y_i^v)\}_{i=1}^{n_v}$, Initial benign distribution boundary σ_0, Initial trade-off coefficient α_0, Optimization steps S, and Learning rate η.
Initialize: $\sigma = \sigma^\star = \sigma_0$, $\alpha = \alpha_0$
▷ *Calculate original accuracy on validation samples*
$P_0 = \frac{1}{n_v} \sum_{i=1}^{n_v} \mathbb{1}(M(x_i^v) = y_i^v)$
for $s = 1$ **to** S **do**
  ▷ *Cross-entropy loss plus boundary penalty*
  $\mathcal{L} = \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{L}_{CE}(M_\sigma(x_i^t), y_i^t) + \alpha \cdot ||\sigma||_1$
  $\sigma = \sigma - \eta \cdot \frac{\partial \mathcal{L}}{\partial \sigma}$
  ▷ *Calculate accuracy when applying current bound*
  $P' = \frac{1}{n_v} \sum_{i=1}^{n_v} \mathbb{1}(M_\sigma(x_i^v) = y_i^v)$
  if $P_0 - P' > \epsilon$ **then**
    $\alpha = \alpha / 2$
  else
    $\alpha = \alpha \cdot 2$
  end if
  ▷ *Update the best boundary value*
  if $P' \geqslant P_0 - \epsilon$ and $||\sigma||_1 < ||\sigma^\star||_1$ **then**
    $\sigma^\star = \sigma$
  end if
end for
Return: $\sigma^\star$

where σ^l denotes the bounding value at the l-th layer. Suppose there are K neurons at this layer, then $\sigma^l = \{\sigma_1^l, \sigma_2^l, \cdots, \sigma_K^l\}$. Similarly, $\sigma = \{\sigma^1, \sigma^2, \cdots, \sigma^L\}$. The objective of UNIT is to mitigate the backdoor effects while preserve the benign utility. Therefore, for any input x of class y and its poisoned version $x \oplus T$ with the attack target label y_T, where T denotes the backdoor trigger,

$$M_\sigma(x) = y, \quad M_\sigma(x \oplus T) \neq y_T. \tag{5}$$

A straightforward idea is to employ a uniform percentile threshold for all neural activation values. However, it can be inaccurate and coarse-grained as different neurons vary in their contributions to backdoor effects. Figure 4 demonstrates the effectiveness of this approach against the DFST [7] attack (launched using CIFAR-10 and ResNet-18), where the original model achieves a clean accuracy of 92.25% and an ASR of 99.77%. The x-axis represents various uniform clipping percentiles while the y-axis shows the corresponding accuracy and ASR after clipping. "Max" indicates setting the boundary at each neuron's maximum activation value. In other cases, we assume a Gaussian distribution of the activation and employ the Z-score for percentile approximation. For example, "Z=3.0" signifies setting the boundary at the mean activation value plus three times its standard deviation, aligning with the 0.98 percentile. We can observe that even with a moderate clean accuracy of 90% (Z=3.5), the ASR remains notably high at 20%. Conversely,

reducing the ASR to 4% (Z=1.5) leads to a drastic decrease in accuracy, down to 40%. This highlights the method's limitation against advanced attacks.

Our approach, on the other hand, utilizes an optimization-based technique to meticulously approximate and tighten a unique boundary for each individual neuron, which outperforms the straightforward approach as illustrate in the blue and red dashed lines in Figure 4. Note that UNIT is able to reduce the ASR to 4.0% while maintain a high accuracy as 91.6%. The details of UNIT are outlined in Algorithm 1, which comprises two main stages: (1) Initialization (Line 1-3), where clean training samples are gathered to approximate a loose benign boundary, and (2) Automated Tightening (Line 4-16), dedicated to refining the approximated boundary with the guidance of clean accuracy.

Initialization. Lines 1-3 present the initialization stage, where input variables are defined, with M representing the model for defense, and ϵ indicating the customized accuracy drop expectation (defaulted to 2%). Following the threat model in Section 1, the defender has access to a small set of clean training samples for the defense process. The data is further split into training samples $\{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ and validation samples $\{(x_i^v, y_i^v)\}_{i=1}^{n_v}$, where n_t and n_v denote the number of training and validation samples, respectively. Typically, the ratio $\frac{n_v}{n_t}$ is set to $\frac{1}{4}$. The split training samples are used for optimization, while validation samples guide the tightening strength. A loose distribution for clean samples is approximated, initializing the distribution boundary of each neuron as σ_0. This initial boundary is set as the mean activation value over the training sample plus four times the standard deviation (Z-score=4 in the straightforward approach). The initial trade-off coefficient between benign accuracy and the tightened distribution boundary is denoted as α_0, with a default value set to 0.001. This value signifies that the tightening process starts with low strength. Additionally, S represents the number of optimization steps, and η denotes the learning rate. Typically, 50 steps prove sufficient to approximate a suitably tight boundary. For optimization, we utilize the Adam optimizer with a learning rate set to $\eta = 0.001$, a standard configuration. Line 2 initializes the optimized distribution boundary σ, optimal boundary $\sigma^\star$, and trade-off coefficient α_0 with their default values. In Line 3 calculates the initial accuracy (P_0) of model M on validation samples, where $\mathbb{1}(M(x_i^v) = y_i^v)$ denotes the number of samples which are correctly classified by M.

Automated Tightening. Lines 4-16 outline the optimization procedure for tightening the benign distribution. In each optimization step, the goal is to tighten the boundary while maintaining benign accuracy within the specified expectation ϵ. Line 5 calculates the loss, consisting of two terms: the cross-entropy loss on the training samples and the penalty on boundary scale. We use L-1 norm of σ to measure the tightness of the current boundary. A small value of $||\sigma||_1$ means a tight boundary. The trade-off between these two loss terms is controlled by α. The boundary σ is optimized using gradient descent in Line 6. Lines 7-12 dynamically adjust the trade-off value α based on the accuracy on validation samples. In Line 7, the current accuracy P' on validation samples is calculated given the optimized σ. If the accuracy drop $P_0 - P'$ exceeds the expectation

ϵ (Line 8), the trade-off coefficient α is reduced by half (Line 9), prioritizing the restoration of benign accuracy. Otherwise, α is increased twice to further tighten the benign distribution boundary (Line 11). Lines 13 to 15 update the optimal boundary $\sigma^\star$ if it maintains benign accuracy while being more tightened. Finally, Line 17 returns the optimal boundary $\sigma^\star$ and UNIT applies the optimal boundary to the model ($M_{\sigma^\star}$) during inference.

5 Evaluation

In this section, we comprehensively evaluate the performance of UNIT across diverse scenarios. In Section 5.2, we assess the effectiveness of UNIT by comparing it against 7 state-of-the-art backdoor mitigation baselines across 14 types of backdoor attacks. In addition, we demonstrate the generalizability of UNIT by the evaluation on four datasets and six network architectures. We assess the time cost of UNIT in Section 5.3 and study the effect of UNIT on clean models in Section 5.4. In Section 5.5, we present additional evaluations of UNIT against the latest backdoor attacks and comparisons with recent baselines. We also include a series of evaluations on adaptive attacks and ablation studies.

5.1 Experiment Setup

Baselines and Settings We employ 14 backdoor attacks, (1) BadNets [20], (2) Trojan [40], (3) CL [65] (4) Dynamic backdoor [53], (5) IA [48], (6) Reflection [41], (7) SIG [3], (8) Blend [6], (9) WaNet [47], (10) ISSBA [37], (11) LIRA [15], (12) Instagram filter [40], (13) DFST [7], and (14) Adap-Blend [51]. We use the default configuration following the original papers, such as trigger patterns, sizes, poisoning strategies, etc. We compare UNIT with 7 state-of-the-art backdoor mitigation methods, (1) standard fine-tuning (FT), (2) FP [38], (3) NAD [35], and (4) ANP [69], (5) NC [66], (6) I-BAU [74] and (7) SEAM [83]. We follow the configuration in the original papers to conduct experiments. All the methods have access to the same amount of training data, e.g., 5%. Details of backdoor attack and defense baselines can be found in Appendix A.1. For UNIT, we set the expected accuracy degradation as 2%.

Evaluation Metrics. We use two metrics: (1) clean accuracy (Acc.), and (2) attack success rate (ASR). Clean accuracy measures the normal functionality of the subject model on classifying clean inputs. ASR measures the backdoor effect, which is the ratio of poisoned samples correctly misclassified to the target label. A good defense shall reduce the ASR while preserving the clean accuracy.

5.2 Effectiveness of UNIT

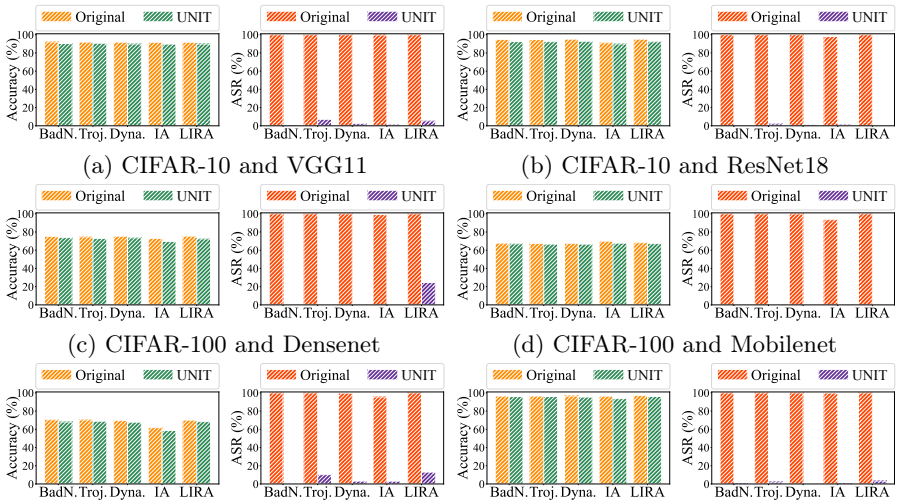
Comparison with Existing Baselines We conducted a comprehensive evaluation of UNIT by comparing it with 9 baseline methods across 14 distinct backdoor attacks on the CIFAR-10 dataset using the ResNet18 architecture for

Table 1: Comparison of UNIT with 7 backdoor mitigation baselines against 14 backdoor attacks. Results are measured in percentages (%). All methods have access to 5% of the clean training data. The best results are highlighted in bold.

Attacks	Original		FT		FP		NAD		ANP		NC		I-BAU		SEAM		UNIT	
	Acc.	ASR	Acc.	ASR	Acc.	ASR	Acc.	ASR	Acc.	ASR	Acc.	ASR	Acc.	ASR	Acc.	ASR	Acc.	ASR
BadNets	94.82	100.0	90.91	9.78	89.68	3.52	92.41	4.79	91.35	3.26	**93.04**	**0.34**	91.60	3.66	91.61	1.05	92.48	0.78
Trojan	94.73	100.0	91.63	35.11	90.76	31.14	91.52	22.30	92.37	58.88	91.89	4.01	90.73	11.58	92.28	12.69	**92.38**	**2.17**
CL	94.58	98.46	90.34	58.72	87.71	3.69	88.47	4.42	89.92	18.18	90.72	1.79	88.75	5.52	92.02	23.04	**92.21**	**1.09**
Dynamic	95.08	100.0	89.11	9.29	84.93	3.23	89.26	2.34	91.99	3.09	92.09	1.78	92.48	1.63	92.61	3.22	**92.77**	**1.54**
IA	91.15	97.96	88.44	2.92	89.71	82.20	88.51	2.67	89.05	5.49	89.32	1.12	89.79	62.45	89.77	1.23	**89.93**	**1.03**
Reflection	93.29	99.33	91.38	74.77	89.68	84.51	90.99	52.97	90.66	93.28	91.38	93.31	89.94	87.85	90.54	21.37	**91.44**	**6.63**
SIG	94.97	99.80	91.29	63.94	90.88	1.03	91.69	10.46	90.80	36.79	91.70	97.88	91.51	22.11	**92.57**	**0.68**	92.48	1.74
Blend	94.62	100.0	90.68	7.30	91.47	2.01	91.62	3.32	91.04	16.79	91.90	1.53	91.43	3.61	91.38	1.80	**91.99**	**1.18**
WaNet	94.36	99.80	90.32	2.85	91.48	1.48	92.36	1.91	**91.99**	**0.61**	90.60	0.97	89.67	12.01	91.34	1.44	91.02	2.44
ISSBA	94.55	100.0	91.40	4.17	90.79	2.11	92.45	2.43	92.42	2.98	**92.52**	**0.46**	83.03	84.58	91.17	3.00	91.84	1.57
LIRA	95.11	100.0	91.42	15.09	89.58	14.76	91.64	2.06	91.98	47.91	92.11	1.17	92.18	12.65	92.18	3.02	**92.29**	**0.58**
Instagram	94.62	99.59	91.40	29.25	90.38	8.03	89.50	7.17	90.10	5.10	90.19	15.88	89.25	7.24	91.35	5.89	**91.43**	**4.98**
DFST	93.25	99.77	90.88	35.22	90.66	14.03	91.05	14.59	89.70	20.51	91.22	24.77	89.12	6.19	91.22	12.93	**91.64**	**4.02**
Adap-Bl.	94.22	82.80	90.15	48.76	87.62	31.36	90.42	49.50	90.80	69.51	90.33	18.25	90.81	19.97	89.58	24.19	**90.84**	**15.03**
Average	94.26	98.39	90.57	28.37	89.67	20.22	90.85	12.92	91.01	27.31	91.36	18.80	90.02	24.36	91.48	8.08	**91.77**	**3.20**

evaluation, with all defenses having access to 5% of the training set. Table 1 summarizes the results. The first column enumerates different backdoor attacks, while the "No Defense" column displays the original performance of backdoored models. The subsequent columns detail the performance of models repaired by various defenses, with "Acc." denoting clean test accuracy and "ASR" representing the attack success rate of backdoor attacks. Notably, UNIT consistently outperforms others in reducing ASR and maintaining high clean accuracy. In instances such as Reflection, DFST, and Adap-Blend attacks, existing defense methods struggle to eliminate the backdoor effect, often retaining over 20% ASR. The sophistication of these attacks, characterized by larger triggers and specialized poisoning strategies, poses a challenge to conventional defenses. For instance, the state-of-the-art Adap-Blend attack relaxes the latent separability assumption and utilizes asymmetric triggers to enhance backdoor resilience. Despite the complexity, UNIT reduces the ASR to less than 7% for Reflection and DFST, outperforming existing methods, even mitigating the Adap-Blend attack to 15%. However, it's worth noting that UNIT doesn't outperform baselines in all scenarios. For instance, ANP performs better on WaNet than UNIT. This is due to the pervasive and sample-specific triggers of WaNet attacks. They resemble natural features and hence make the poisoned activation distribution less distinguishable from the clean one. Despite this, UNIT still demonstrates effectiveness by mitigating the backdoor effect to an ASR of under 2.5%. Furthermore, we evaluate UNIT on two latest backdoor attacks, i.e., NARCISSUS [75] and COMBAT [29], and compare the performance with five state-of-the-art baselines, i.e., CLP [80], FST [45], RNP [36], FT-SAM [82] and Super-FT [54]. The results in Appendix A.2 demonstrate UNIT's superior performance over these methods.

Evaluation on Various Datasets and Networks We extend the evaluation of UNIT to include a diverse set of datasets and network architectures. The experiments include four datasets: CIFAR-10 [33], CIFAR-100 [33], STL-10 [13],



(a) CIFAR-10 and VGG11 (b) CIFAR-10 and ResNet18

(c) CIFAR-100 and Densenet (d) CIFAR-100 and Mobilenet

(e) STL-10 and WideResNet (f) GTSRB and PRN34

Fig. 5: Evaluation on different datasets and network architectures

and GTSRB [60], and six network architectures: VGG11 [58], ResNet18 [24], Densenet [27], Mobilenet [26], WideResNet [73], and Pre-activation ResNet34 (PRN34) [24]. 5% of the clean training data is used for defense. Results are presented in Figure 5, with each sub-figure depicting the outcomes for a specific dataset-network pair. In each sub-figure, the left plot illustrates clean accuracy, while the right plot displays the ASR. The x-axis represents different backdoor attacks, and the y-axis denotes accuracy or ASR. Bar colors in the legend distinguish results before and after the defense. Notably, UNIT consistently reduces ASR from 100% to near 0% across various datasets and network architectures. Clean accuracy degradation is minimal in most cases, demonstrating the general effectiveness of UNIT across diverse scenarios.

Application on Transformers. Although UNIT is primarily designed for CNN models, we investigate its performance in eliminating backdoor effects in transformers. We poison the CIFAR-10 dataset with BadNets [20] triggers and finetune the ViT-base-patch16-224 [16] model on it. The model achieves 98.44% accuracy and 100% ASR. We then apply UNIT to tighten the benign distribution boundary on each *attention layer*, which successfully reduces the ASR to 5.78%, with a slight accuracy drop of 3.23%. These results highlight Tech's potential utility in protecting transformers from backdoor attacks.

5.3 Defense Efficiency

We conducted a study on the time cost of various defenses, and the results are illustrated in Figure 6. The x-axis represents different methods, and the y-axis indicates the time cost measured in seconds, with each bar denoting the average

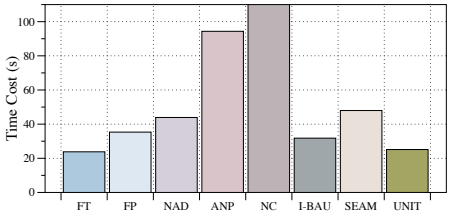


Fig. 6: Time cost of different baselines

Table 2: Impact on clean models

Dataset	Network	Original	UNIT	Diff.
CIFAR-10	VGG11	91.88%	89.93%	1.95%
CIFAR-10	ResNet18	95.08%	92.25%	2.83%
CIFAR-100	Densenet	74.78%	73.07%	1.71%
CIFAR-100	Mobilenet	67.74%	67.42%	0.32%
STL-10	WideResNet	69.81%	69.45%	0.36%
GTSRB	PRN34	97.12%	95.21%	1.91%

time cost. Notably, UNIT completes its process in approximately 20 seconds as it only needs to estimate the benign activation distributions based on a small set of clean samples. Other cost-efficient methods, such as I-BAU and fine-tuning, exhibit similar time costs to UNIT. However, as shown in Table 1, they fall short in defending against a few advanced attacks.

Since UNIT modifies the activation layers, we also measure its impact on the model inference. We feed the whole test set containing 10,000 images to the ResNet18 model on CIFAR-10 before and after applying UNIT. The experiment is repeated 5 times. The time cost is 2.79 ± 0.35s for the original model, and 2.86 ± 0.20s for the model integrated with UNIT. The inference time difference is negligible (around 2.5%). Such a small increase during inference is acceptable as UNIT can effectively preclude all evaluated backdoor attacks.

5.4 Impact on Clean Models

We investigate the impact of UNIT on clean models, considering that defenders may apply UNIT without prior knowledge of whether a model is poisoned. Table 2 presents the accuracy before and after applying UNIT on various clean models. Notably, the degradation of clean accuracy ranges from 0.32% to 2.83%, highlighting the minimal impact.

5.5 Additional Evaluation of UNIT

We conduct evaluation on the latest backdoor attacks and compare UNIT with a few more recent defenses in Appendix A.2. We evaluate UNIT's performance under three adaptive attack scenarios in Appendix A.4, showing its robustness against them. We carry out a series of ablation studies to examine UNIT's resilience across various hyper-parameters and attack settings in Appendix A.5.

6 Conclusion

We present a novel backdoor mitigation technique designed to approximate a tight distribution for each neuron. It then effectively reduce any high activation that exceeds the established boundary. Our comprehensive evaluation illustrates the high efficacy of UNIT, outperforming 7 baselines across 14 existing attacks.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. We are grateful to the Center for AI Safety for providing computational resources. This research was supported, in part by IARPA TrojAI W911NF-19-S0012, NSF 1901242 and 1910300, ONR N000141712045, N000141410468 and N000141712947. Any opinions, findings, and conclusions in this paper are those of the authors only and do not necessarily reflect the views of our sponsors.

References

1. An, S., Chou, S.Y., Zhang, K., Xu, Q., Tao, G., Shen, G., Cheng, S., Ma, S., Chen, P.Y., Ho, T.Y., et al.: Elijah: Eliminating backdoors injected in diffusion models via distribution shift. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 10847–10855 (2024)
2. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: International Conference on Artificial Intelligence and Statistics. pp. 2938–2948. PMLR (2020)
3. Barni, M., Kallas, K., Tondi, B.: A new backdoor attack in cnns by training set corruption without label poisoning. CoRR **abs/1902.11237** (2019), http://arxiv.org/abs/1902.11237
4. Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., Srivastava, B.: Detecting backdoor attacks on deep neural networks by activation clustering. arXiv preprint arXiv:1811.03728 (2018)
5. Chen, X., Salem, A., Backes, M., Ma, S., Zhang, Y.: Badnl: Backdoor attacks against nlp models. In: ICML 2021 Workshop on Adversarial Machine Learning (2021)
6. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
7. Cheng, S., Liu, Y., Ma, S., Zhang, X.: Deep feature space trojan attack of neural networks by controlled detoxification. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 1148–1156 (2021)
8. Cheng, S., Shen, G., Tao, G., Zhang, K., Zhang, Z., An, S., Xu, X., Liu, Y., Ma, S., Zhang, X.: Odscan: Backdoor scanning for object detection models. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 118–118. IEEE Computer Society (2024)
9. Cheng, S., Tao, G., Liu, Y., An, S., Xu, X., Feng, S., Shen, G., Zhang, K., Xu, Q., Ma, S., et al.: Beagle: Forensics of deep learning backdoor attack for better defense. arXiv preprint arXiv:2301.06241 (2023)
10. Cheng, S., Tao, G., Liu, Y., Shen, G., An, S., Feng, S., Xu, X., Zhang, K., Ma, S., Zhang, X.: Lotus: Evasive and resilient backdoor attacks through sub-partitioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24798–24809 (2024)
11. Chou, S.Y., Chen, P.Y., Ho, T.Y.: How to backdoor diffusion models? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4015–4024 (2023)
12. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)

13. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 215–223. JMLR Workshop and Conference Proceedings (2011)
14. Doan, B.G., Abbasnejad, E., Ranasinghe, D.C.: Februus: Input purification defense against trojan attacks on deep neural network systems. In: Annual Computer Security Applications Conference. pp. 897–912 (2020)
15. Doan, K., Lao, Y., Zhao, W., Li, P.: Lira: Learnable, imperceptible and robust backdoor attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11966–11976 (2021)
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
17. Dubey, S.R., Singh, S.K., Chaudhuri, B.B.: Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing (2022)
18. Feng, S., Tao, G., Cheng, S., Shen, G., Xu, X., Liu, Y., Zhang, K., Ma, S., Zhang, X.: Detecting backdoors in pre-trained encoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16352–16362 (2023)
19. Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: Strip: A defence against trojan attacks on deep neural networks. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 113–125 (2019)
20. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
21. Guo, W., Wang, L., Xing, X., Du, M., Song, D.: Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. arXiv preprint arXiv:1908.01763 (2019)
22. Han, J., Moraga, C.: The influence of the sigmoid function parameters on the speed of backpropagation learning. In: International workshop on artificial neural networks. pp. 195–201. Springer (1995)
23. Hayase, J., Kong, W., Somani, R., Oh, S.: Spectre: defending against backdoor attacks using robust statistics. arXiv preprint arXiv:2104.11315 (2021)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
25. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Deep Learning and Representation Learning Workshop (2015), http://arxiv.org/abs/1503.02531
26. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
27. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
28. Huang, K., Li, Y., Wu, B., Qin, Z., Ren, K.: Backdoor defense via decoupling the training process. arXiv preprint arXiv:2202.03423 (2022)
29. Huynh, T., Nguyen, D., Pham, T., Tran, A.: Combat: Alternated training for effective clean-label backdoor attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 2436–2444 (2024)
30. Jia, J., Liu, Y., Gong, N.Z.: Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. arXiv preprint arXiv:2108.00352 (2021)

31. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
32. Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S.: Self-normalizing neural networks. Advances in neural information processing systems **30** (2017)
33. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
34. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Anti-backdoor learning: Training clean models on poisoned data. Advances in Neural Information Processing Systems **34**, 14900–14912 (2021)
35. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Neural attention distillation: Erasing backdoor triggers from deep neural networks. arXiv preprint arXiv:2101.05930 (2021)
36. Li, Y., Lyu, X., Ma, X., Koren, N., Lyu, L., Li, B., Jiang, Y.G.: Reconstructive neuron pruning for backdoor defense. In: International Conference on Machine Learning. pp. 19837–19854. PMLR (2023)
37. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Invisible backdoor attack with sample-specific triggers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16463–16472 (2021)
38. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: International Symposium on Research in Attacks, Intrusions, and Defenses. pp. 273–294. Springer (2018)
39. Liu, Y., Lee, W.C., Tao, G., Ma, S., Aafer, Y., Zhang, X.: Abs: Scanning neural networks for back-doors by artificial brain stimulation. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 1265–1282 (2019)
40. Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks. NDSS (2017)
41. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: European Conference on Computer Vision. pp. 182–199. Springer (2020)
42. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 4765–4774. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf
43. Maas, A.L., Hannun, A.Y., Ng, A.Y., et al.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. p. 3. Atlanta, Georgia, USA (2013)
44. Maćkiewicz, A., Ratajczak, W.: Principal components analysis (pca). Computers & Geosciences **19**(3), 303–342 (1993)
45. Min, R., Qin, Z., Shen, L., Cheng, M.: Towards stable backdoor purification through feature shift tuning. Advances in Neural Information Processing Systems **36** (2024)
46. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
47. Nguyen, A., Tran, A.: Wanet–imperceptible warping-based backdoor attack. arXiv preprint arXiv:2102.10369 (2021)
48. Nguyen, T.A., Tran, A.: Input-aware dynamic backdoor attack. Advances in Neural Information Processing Systems **33**, 3454–3464 (2020)

49. PyTorch: Tanhshrink, https://pytorch.org/docs/stable/generated/torch.nn.Tanhshrink.html
50. Qi, F., Li, M., Chen, Y., Zhang, Z., Liu, Z., Wang, Y., Sun, M.: Hidden killer: Invisible textual backdoor attacks with syntactic trigger. arXiv preprint arXiv:2105.12400 (2021)
51. Qi, X., Xie, T., Li, Y., Mahloujifar, S., Mittal, P.: Revisiting the assumption of latent separability for backdoor defenses. In: The eleventh international conference on learning representations (2022)
52. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: Proceedings of the AAAI conference on artificial intelligence. pp. 11957–11965 (2020)
53. Salem, A., Wen, R., Backes, M., Ma, S., Zhang, Y.: Dynamic backdoor attacks against machine learning models. arXiv preprint arXiv:2003.03675 (2020)
54. Sha, Z., He, X., Berrang, P., Humbert, M., Zhang, Y.: Fine-tuning is all you need to mitigate backdoor attacks. arXiv preprint arXiv:2212.09067 (2022)
55. Shen, G., Cheng, S., Tao, G., Zhang, K., Liu, Y., An, S., Ma, S., Zhang, X.: Django: Detecting trojans in object detection models via gaussian focus calibration. Advances in Neural Information Processing Systems **36** (2024)
56. Shen, G., Liu, Y., Tao, G., An, S., Xu, Q., Cheng, S., Ma, S., Zhang, X.: Backdoor scanning for deep neural networks through k-arm optimization. In: International Conference on Machine Learning. pp. 9525–9536. PMLR (2021)
57. Shen, G., Liu, Y., Tao, G., Xu, Q., Zhang, Z., An, S., Ma, S., Zhang, X.: Constrained optimization with dynamic bound-scaling for effective nlp backdoor defense. In: International Conference on Machine Learning. pp. 19879–19892. PMLR (2022)
58. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). https://doi.org/10.48550/ARXIV.1409.1556, https://arxiv.org/abs/1409.1556
59. Souri, H., Fowl, L., Chellappa, R., Goldblum, M., Goldstein, T.: Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. arXiv preprint arXiv:2106.08970 (2021)
60. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks **32**, 323–332 (2012)
61. Tao, G., Liu, Y., Cheng, S., An, S., Zhang, Z., Xu, Q., Shen, G., Zhang, X.: Deck: Model hardening for defending pervasive backdoors. arXiv preprint arXiv:2206.09272 (2022)
62. Tao, G., Liu, Y., Shen, G., Xu, Q., An, S., Zhang, Z., Zhang, X.: Model orthogonalization: Class distance hardening in neural networks for better security. In: 2022 IEEE Symposium on Security and Privacy (SP). IEEE. vol. 3 (2022)
63. Tao, G., Wang, Z., Cheng, S., Ma, S., An, S., Liu, Y., Shen, G., Zhang, Z., Mao, Y., Zhang, X.: Backdoor vulnerabilities in normally trained deep learning models. arXiv preprint arXiv:2211.15929 (2022)
64. Tran, B., Li, J., Madry, A.: Spectral signatures in backdoor attacks. Advances in neural information processing systems **31** (2018)
65. Turner, A., Tsipras, D., Madry, A.: Clean-label backdoor attacks. OpenReview (2018)
66. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 707–723. IEEE (2019)

67. Wang, R., Zhang, G., Liu, S., Chen, P.Y., Xiong, J., Wang, M.: Practical detection of trojan neural networks: Data-limited and data-free cases. In: European Conference on Computer Vision. pp. 222–238. Springer (2020)
68. Wang, Z., Ding, H., Zhai, J., Ma, S.: Training with more confidence: Mitigating injected and natural backdoors during training. Advances in Neural Information Processing Systems **35**, 36396–36410 (2022)
69. Wu, D., Wang, Y.: Adversarial neuron pruning purifies backdoored deep models. Advances in Neural Information Processing Systems **34**, 16913–16925 (2021)
70. Xu, Q., Tao, G., Honorio, J., Liu, Y., An, S., Shen, G., Cheng, S., Zhang, X.: Medic: Remove model backdoors via importance driven cloning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20485–20494 (2023)
71. Xu, X., Wang, Q., Li, H., Borisov, N., Gunter, C.A., Li, B.: Detecting ai trojans using meta neural analysis. 2021 IEEE Symposium on Security and Privacy (SP) pp. 103–120 (2021)
72. Yan, L., Cheng, S., Shen, G., Tao, G., Chen, X., Zhang, K., Mao, Y., Zhang, X.: d^3: Detoxing deep learning dataset. In: NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly (2023)
73. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
74. Zeng, Y., Chen, S., Park, W., Mao, Z.M., Jin, M., Jia, R.: Adversarial unlearning of backdoors via implicit hypergradient. arXiv preprint arXiv:2110.03735 (2021)
75. Zeng, Y., Pan, M., Just, H.A., Lyu, L., Qiu, M., Jia, R.: Narcissus: A practical clean-label backdoor attack with limited information. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 771–785 (2023)
76. Zhang, K., Cheng, S., Shen, G., Tao, G., An, S., Makur, A., Ma, S., Zhang, X.: Exploring the orthogonality and linearity of backdoor attacks. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 225–225. IEEE Computer Society (2024)
77. Zhang, K., Tao, G., Xu, Q., Cheng, S., An, S., Liu, Y., Feng, S., Shen, G., Chen, P.Y., Ma, S., et al.: Flip: A provable defense framework for backdoor mitigation in federated learning. arXiv preprint arXiv:2210.12873 (2022)
78. Zhang, Z., Panda, A., Song, L., Yang, Y., Mahoney, M., Mittal, P., Kannan, R., Gonzalez, J.: Neurotoxin: Durable backdoors in federated learning. In: International Conference on Machine Learning. pp. 26429–26446. PMLR (2022)
79. Zheng, H., Yang, Z., Liu, W., Liang, J., Li, Y.: Improving deep neural networks using softplus units. In: 2015 International joint conference on neural networks (IJCNN). pp. 1–4. IEEE (2015)
80. Zheng, R., Tang, R., Li, J., Liu, L.: Data-free backdoor removal based on channel lipschitzness. In: European Conference on Computer Vision. pp. 175–191. Springer (2022)
81. Zheng, R., Tang, R., Li, J., Liu, L.: Pre-activation distributions expose backdoor neurons. Advances in Neural Information Processing Systems **35**, 18667–18680 (2022)
82. Zhu, M., Wei, S., Shen, L., Fan, Y., Wu, B.: Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4466–4477 (2023)
83. Zhu, R., Tang, D., Tang, S., Wang, X., Tang, H.: Selective amnesia: On efficient, high-fidelity and blind suppression of backdoor effects in trojaned machine learning models. arXiv preprint arXiv:2212.04687 (2022)