UltraClean: A Simple Framework to Train Robust Neural Networks against Backdoor Attacks

Bingyin Zhao
Pixocial Technology
bingyin.zhao@pixocial.com

Yingjie Lao
Tufts University
yingjie.lao@tufts.edu

Abstract

Backdoor attacks are emerging threats to deep neural networks, which typically embed malicious behaviors into a victim model by injecting poisoned samples. Adversaries can activate the injected backdoor during inference by presenting the trigger on input images. Prior defensive methods have achieved remarkable success in countering dirty-label backdoor attacks where the labels of poisoned samples are often mislabeled. However, these approaches do not work for a recent new type of backdoor – clean-label backdoor attacks that imperceptibly modify poisoned data and hold consistent labels. More complex and powerful algorithms are demanded to defend against such stealthy attacks. In this paper, we propose UltraClean, a general framework that simplifies the identification of poisoned samples and defends against both dirty-label and clean-label backdoor attacks. Given the fact that backdoor triggers introduce adversarial noise that intensifies in feed-forward propagation, UltraClean first generates two variants of training samples using off-the-shelf denoising functions. It then measures the susceptibility of training samples leveraging the error amplification effect in DNNs, which dilates the noise difference between the original image and denoised variants. Lastly, it filters out poisoned samples based on the susceptibility to thwart the backdoor implantation. Despite its simplicity, UltraClean achieves a superior detection rate across various datasets and significantly reduces the backdoor attack success rate while maintaining a decent model accuracy on clean data, outperforming existing defensive methods by a large margin. Code is available at https://github.com/bxz9200/UltraClean.

1. Introduction

With the thriving of machine learning, deep neural networks (DNNs) have achieved unprecedented progress and reached human-level performance in various tasks, including computer vision [17, 27], natural language processing [5, 13] and game playing [53]. Given the remarkable success, DNNs are further deployed to safety-critical applications such as authentication [61] and autonomous driving [4]. However, DNNs are proven to be vulnerable to a variety type of adversarial attacks. One notorious attack is the adversarial example [22] that occurs at test-time, where an adversary fools well-trained DNN models by adding imperceptible perturbations on test images. Another well-known attack is the data poisoning attack [3] that occurs in the training phase, where an adversary can inject well-crafted poisoned samples into the training data and introduce malicious behavior to models trained on the poisoned dataset. This paper focuses on backdoor attacks [23], where an adversary attempts to contaminate the training dataset via data poisoning and install a backdoor into models trained on the corrupted dataset. During inference, backdoored models misclassify inputs with backdoor triggers to a target label while behaving normally on benign inputs.

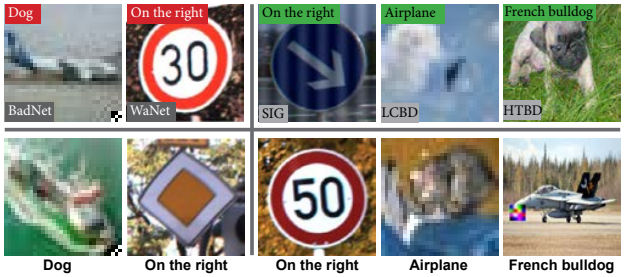

Figure 1. Illustration of dirty-label and clean-label attacks (Top row: poisoned training samples; Bottom row: backdoored test samples. Red: incorrect labels; Green: correct labels. Dirty-label poisoned samples always possess incorrect labels while clean-label poisoned samples are imperceptible compared to benign samples and possess correct labels.

Backdoor attacks come in two flavors: dirty-label attacks [9, 44, 46] and clean-label attacks [2, 25, 52, 58]. Poisoned samples of dirty-label attacks are always mislabeled. For example, as shown in Figure 1, a poisoned sample "airplane" is labeled as "dog". In contrast, clean-label attacks hold consistent labels to images content. Although dirty-label attacks are effective, they can be easily

distinguished due to incorrect labels. Existing defenses have shown decent performance in detecting and mitigating such attacks [21, 36]. On the other hand, clean-label attacks are more stealthy and insidious. The clean-label poisoned samples are almost visually indistinguishable from benign samples; thus, defenses against the attack become a more challenging task. Some recent works [29, 39] attempted to alleviate the clean-label attacks by decoupling the training phase to suppress the backdoor injection. However, they require complicated algorithms with significantly more operations in training and do not identify the backdoor samples in the poisoned dataset. Thus, users have to re-run the defense algorithms every time they train a new model by using the same potentially poisoned dataset, which dramatically increases the cost.

In this work, we propose UltraClean, a poisons-filtering-based framework (i.e., dataset cleanse) to detect backdoor samples, cleanse poisoned datasets, and train backdoor-mitigated models against both dirty-label and clean-label backdoor attacks. We focus on the image classification task. Our idea is inspired by prior works [40] and [64, 65], which have demonstrated that adversarial perturbations of adversarial examples are amplified during the feed-forward propagation (i.e., **error amplification effect**) in DNNs and can be effectively eliminated by simple image-denoising techniques. We argue that although backdoor samples hold fundamentally different generation mechanisms to adversarial examples, the backdoor triggers share a similar error amplification effect as the adversarial perturbations. To this end, our training framework, UltraClean, employs off-the-shelf image-denoising functions and the error implication effect to filter out poisoned samples from benign training data and thwart the backdoor implantation.

The proposed method first trains an arbitrary model on the potentially poisoned dataset and uses it as the backdoor detection model. We then produce two variants of each training image using denoising functions and feed the difference corresponding to the original image into the detection model. We leverage the error amplification effect to compute the susceptibility of the training data in a feed-forward pass. The susceptibility of poisoned data tends to be higher, guiding the model to detect and remove these samples. Finally, we obtain a backdoor-mitigated model by retraining on the sterilized dataset. We show that UltraClean is highly effective in defending against various representative attacks including both dirty-label and clean-label ones, achieving a high detection rate while maintaining the model accuracy.

Our contributions are summarized as follows:

- UltraClean is a once-for-all backdoor-free training framework and a poisons-filtering-based (i.e., dataset cleanse) defense against both dirty and clean-label backdoor attacks. The method not only defends against backdoor attacks but also identifies poisoned samples regardless of generation mechanisms and dataset complexity.
- We propose an effective approach that does not affect the normal training process and significantly simplifies the backdoor identification and mitigation.
- We demonstrate that two widely adopted poisons-filtering-based defenses are only effective on dirty-label attacks but are unsuccessful in defending against clean-label attacks.
- We conduct comprehensive experiments and examine different types of dirty-label and clean-label attacks to illustrate the effectiveness of UltraClean.

2. Related Work

2.1. Backdoor Attacks on DNN

Backdoor attacks can be broadly categorized into **data poisoning** [9, 23, 52, 69, 70] and **model poisoning** [15, 16, 45, 46]. Data poisoning backdoor attacks implant the backdoors by modifying training data (i.e., injecting poisoned samples), which does not require access to the training process. Such attacks affect victims indirectly through the poisoned data set. In contrast, model poisoning backdoor attacks require full control of the training process to alter model parameters and embed backdoors into the model. Such attacks affect victims by providing the poisoned model directly. **Our work focuses on the defense against data poisoning attacks via dataset cleanse.**

The very first backdoor attack via data poisoning against neural networks is BadNets [23] where poisoned training samples are generated by stamping a pre-defined pattern (trigger) onto benign images and assigned with incorrect target labels (dirty-label). One drawback of BadNets is that the poisoned samples are suspicious and quite easy to detect upon human inspection. Later on, a series of works are proposed to enhance the stealthiness by improving the invisibility of backdoor triggers. [9] blends backdoor triggers with benign images; [72] imposes imperceptible perturbations onto poisoned samples using universal adversarial perturbations [44]; [35, 37] creates invisible triggers via steganography and regularization; [10] employs controlled detoxification to perform the attack in the feature space. However, all these methods are still dirty-label attacks and could be detected by inspecting data labels.

On the other hand, clean-label backdoor attacks are more stealthy and hard to detect since they retain consistent labels to the image contents. For instance, [2, 43, 51] superimpose the trigger with benign images and employ different techniques (i.e., sinusoidal signal, reflection and image-scaling) to conceal the trigger; [58] creates poisoned sample by placing a stealthy patch on hard-to-classify images generated by generative models or adversarial perturbations; [52] produces poisoned samples by minimizing their distance from source benign images in feature space. The objective of UltraClean is to defend against both dirty-label attacks and

clean-label attacks in a simple and effective fashion.

2.2. Defenses

Defenses against backdoor attacks are extensively studied in recent years, which can be broadly categorized as training-phase defense and inference-phase defense. Training-phase defense attempts to eliminate backdoor before model deployment by filtering poisoned training samples [7, 11, 19, 26, 56, 57, 67], suppressing the effectiveness of poisoned data [18, 28, 29, 39], or reconstructing the trained model [38, 41, 68, 71]. Inference-phase defense alleviates the backdoor effect after model deployment by pre-processing data [14, 50, 59], filtering suspicious samples [32, 33], synthesizing possible triggers [8, 24, 49, 60, 73], or identifying if a model is backdoored [30, 31, 34, 62, 66].

This paper proposes the first poisons-filtering-based defensive solution against a wide range of backdoor attacks (i.e., both dirty-label and clean-label). Our work is a training-phase defense where users seek to detect and remove malicious training samples and train a backdoor-free model. The most relevant works are the spectral signatures defense [57] and strong intentional perturbation defense [19] that can effectively distinguish poisoned samples in the dataset. However, we find that these defenses are only effective against dirty-label attacks.

3. UltraClean

3.1. Threat Model

We consider the same threat model defined in prior data poisoning backdoor works [57] where the adversary targets to implant a backdoor into a DNN for an image classification task by injecting a fraction of poisoned samples into the training dataset. We assume a strong adversary that knows standard training algorithms, classic model architectures, and the statistical information of the training dataset to craft powerful and stealthy poisoned samples. However, the model is not trained by the adversary but the user with possibly contaminated training dataset from an uncertified source. Under the threat model, we evaluate the effectiveness of the proposed defensive framework from three dimensions.

Backdoor detection rate (BDR). The detection rate is the fraction of poisoned samples detected by the defense. We seek to detect as many poisoned samples as possible.

Attack success rate (ASR). The ASR is the fraction of test images classified as the target label in the presence of the backdoor trigger. We want the ASR to be as low as possible after retraining on the sanitized dataset.

Model accuracy on clean data. The model accuracy is the fraction of benign images that the trained model correctly classifies. We hope the accuracies before and after retraining remain as close as possible.

Algorithm 1: UltraClean

Input: Training dataset $(\mathbf{x}, \mathbf{y}) \sim \boldsymbol{D_{tr}}$,
Randomly initialized deep neural network model $\mathbf{M}_\theta$,
Detection and removal threshold $\boldsymbol{\beta}$,
Output of a potentially backdoored model $\boldsymbol{f_{\theta^*}(\cdot)}$,
Total number of training samples $\boldsymbol{n}$
, **Output:** Sanitized training dataset $\boldsymbol{D_s}$,
Post-clean model $\mathbf{M}_{\hat{\theta}}$
#Train the model on the potentially poisoned dataset
Minimize $\mathcal{L} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim \boldsymbol{D_{tr}}}[\ell(\mathbf{M}_\theta(\mathbf{x}), \mathbf{y})] \rightarrow \mathbf{M}_{\theta^*}$
#Initialize the sanitized training set and score list
$\boldsymbol{D_s} \leftarrow \{\}$;
$S \leftarrow$ MaxHeap []
#Generate denoised variants
for all $\mathbf{x}$ **do**
  for c_i **in** $\mathbf{x}$ **do**
    $\tilde{c_i} = \frac{1}{C(p)} \sum_{q \in \Omega(p,r)} c_i(q) w(p,q) \rightarrow \tilde{\mathbf{x_1}}$
    $\tilde{c_i} = median\{q \in \Omega(p) : c_i(q)\} \rightarrow \tilde{\mathbf{x_2}}$
  end for
  #Enlarge error using error amplication effect
  $\mathbf{v} = \boldsymbol{f_{\theta^*}}(\mathbf{x})$; $\mathbf{v_1} = \boldsymbol{f_{\theta^*}}(\tilde{\mathbf{x_1}})$; $\mathbf{v_2} = \boldsymbol{f_{\theta^*}}(\tilde{\mathbf{x_2}})$
  #Compute susceptibility
  $\mathbf{s} = ||\mathbf{v} - \mathbf{v_1}||_1 + ||\mathbf{v} - \mathbf{v_2}||_1$
  heappush$(S, (\mathbf{s}, (\mathbf{x}, \mathbf{y})))$
end for
#Remove poisoned samples from the dataset
for i = 0 **to** $\boldsymbol{\beta} \cdot \boldsymbol{n}$ **do**
  $\boldsymbol{D_{tr}}$.remove(**heappop**(S)[1]) $\rightarrow \boldsymbol{D_s}$
end for
#Train/Retrain the model on the sterilized dataset
Minimize $\mathcal{L} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim \boldsymbol{D_s}}[\ell(\mathbf{M}_\theta(\mathbf{x}), \mathbf{y})] \rightarrow \mathbf{M}_{\hat{\theta}}$
Return $\mathbf{M}_{\hat{\theta}}, \boldsymbol{D_s}$.

3.2. UltraClean Framework

The proposed UltraClean is a general framework that aims at defending against various backdoor attacks by profoundly cleansing poisoned samples from the training dataset. UltraClean consists of three phases: pre-clean training, poisons clean (detection and removal), and post-clean retraining. In the pre-clean training, we train the DNN model on the possibly poisoned dataset. Note that the trained model will be mounted with the backdoor as intended. In the poisons clean phase, we first generate two baseline images of the training data using two denoising filters separately and then compute the ℓ_1-norm distance score of softmax layer output between the original image and its pair of denoised variants using the DNN model obtained in the pre-clean training phase. The denoised-original distance scores (susceptibility) of poisoned samples tend to be higher than benign samples because

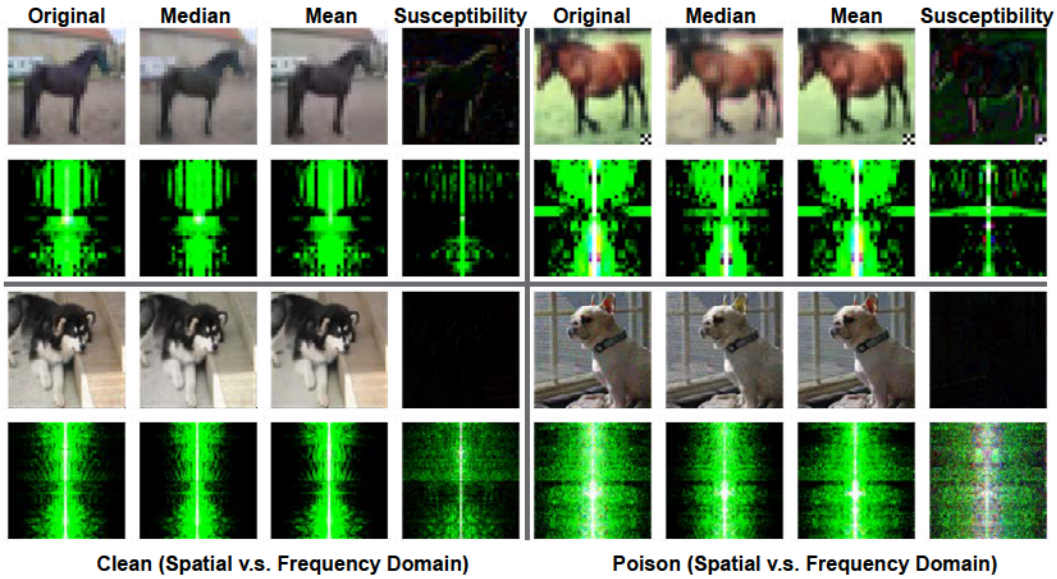

Figure 2. Spatial and frequency domain views of two backdoor attacks. Top two rows are the dirty-label attack (BadNets [23]); Bottom two rows are the clean-label attack (Hidden Trigger Backdoor [52]). As shown in the 4th and 8th columns, poisons reveal substantial qualitative noise difference than the clean counterparts. The noise difference in pixel space are amplified during the feed-forward propagation in a deep neural network and become a strong indicator to differentiate poisoned and benign samples.

the error is amplified during feed-forward propagation. Thus, the poisoned samples can be easily detected and removed from the training dataset upon the susceptibility. Lastly, we retrain the model on the sanitized training dataset and acquire the backdoor-mitigated model. The concrete account of the UltraClean framework is presented in Algorithm 1.

3.3. Methodology

The key of UltraClean is to develop an approach that can differentiate poisoned and benign samples, which hence can be effective under both dirty-label and clean-label settings. In general, backdoor attacks require a trigger to induce DNNs to learn the malicious behavior. Such a trigger, whether visually perceptible or not, introduces extra noise (pixels that do not match the image content) to benign images. We plot the frequency-domain view in Figure 2 to visualize the signal distribution in images. The central area of the frequency-domain figure represents the low-frequency signal, while the peripheral area represents the high-frequency signal. As shown in the 1st and 5th columns, poisons demonstrate more high-frequency noise than benign data. The noise is bounded by the maximum perturbation added to the images and is sometimes hard to detect in the pixel space (i.e., some works attempt to minimize the perturbation [52, 58]). For example, it is barely impossible to perceptually distinguish the poison sample (the right side of the 3rd row) from the clean sample (the left side of the 3rd row) while they have obvious differences in the frequency domain (the 4th and 8th columns in the 4th row). Moreover, the noise distributions vary due to different mechanisms of poisoned sample generation, which demands high generalizability of the detection.

To tackle this issue, we propose a simple yet effective approach that differentiates poisoned samples by measuring the susceptibility leveraging the **error amplification effect** of DNN along with two off-the-shelf denoising functions. We choose non-local mean and local median as the denoising functions based on the following reasons: i). We want to keep our design simple and effective so that UltraClean can be applied in more general cases; ii). The non-local mean and the local median are the most common denoising techniques and can be implemented without additional cost to the training process; iii). The poisons-filtering algorithm should be agnostic to the mechanism of poisoned sample generation. While other denoising techniques might also be used in our framework, these two denoising methods complement each other (i.e., the mean filter is linear and the median filter is non-linear) and are surprisingly effective in empirically identifying backdoor samples against various backdoor triggers when incorporated with the error amplification effect. We briefly introduce the denoising functions below and conduct ablation studies to illustrate their importance and complementarity in backdoor detection in the supplementary materials.

Non-local mean denoising [6] removes noise by replacing each pixel value with a weighted mean computed over global spatial regions. It is defined as:

$$\tilde{c}(p) = \frac{1}{C(p)} \int g(d(\Omega(p,r), \Omega(q,r)))c(q)dq, \quad (1)$$

where $\tilde{c}(p)$ is the denoised value of pixel p, $d(\cdot)$ represents the Euclidean distance between spatial regions $\Omega(p,r)$ and $\Omega(q,r)$. $\Omega(p,r)$ and $\Omega(q,r)$ are search windows centered at pixels p and q, and the boundary of which is defined by r. $C(\cdot)$ is a normalization function and $g(\cdot)$ is a decreasing function. In this work, we exploit the pixel-wise implementation of the algorithm. Consider a color image with RGB channels as $\mathbf{x} = (c_1, c_2, c_3)$, the denoising operation is expressed as:

$$\begin{aligned} \tilde{c_i}(p) &= \frac{1}{C(p)} \sum_{q \in \Omega(p,r)} c_i(q) w(p,q), \\ C(p) &= \sum_{q \in \Omega(p,r)} w(p,q), \end{aligned} \tag{2}$$

where $w(p,q)$ is a weighting function depends on the squared distance $d^2(\cdot)$ as expressed in Equation (3), which can be calculated by Equation (4).

$$d^2 = \frac{\sum_{i=1}^{3} \sum_{j \in \Omega(0,r)} (c_i(p+j) - c_i(q+j))^2}{3 \times (2r+1)^2}. \tag{3}$$

$$w(p,q) = e^{-\frac{max(d^2 - 2\sigma^2, 0.0)}{h^2}}. \tag{4}$$

Local median denoising removes noise by weighting nearby pixels of each pixel using the median smoothing filter. The filter scans over each pixel p and replaces the value of the center pixel with the median value of surrounding pixels, which can be expressed as:

$$\tilde{c_i}(p) = median\{q \in \Omega(p) : c_i(q)\}. \tag{5}$$

We acknowledge there exist many other denoising algorithms designed specifically for certain types of noise and conduct an abliation study in Appendix G, where we observe that using the aforementioned two denoising techniques can already achieve excellent performance with trivial complexity. The workflow of UltraClean is straightforward, for each training image $\mathbf{x}$, we generate two denoised versions $\tilde{\mathbf{x_1}}$ (e.g., the 2nd and 6th columns in Fig 2) and $\tilde{\mathbf{x_2}}$ (e.g., the 3rd and 7th columns). $\tilde{\mathbf{x_1}}$ and $\tilde{\mathbf{x_2}}$ serve as baselines to compute the susceptibility. We define susceptibility as the error (i.e., noise difference) between original images and denoised variants. It can be seen from the 4th and 8th columns that the error of poisoned images are considerably higher than that of clean images. However, as seen in the 6th and 7th columns, the denoising functions do not significantly reduce the noise in the denoised variants compared to the original poisoned samples (the 5th column), so we are not able to distinguish poisoned samples from benign samples by simply computing the susceptibility in the pixel space.

We propose a method to enlarge the error to facilitate the computation using the error amplification effect, which is a property of DNNs that minor adversarial perturbations in model inputs accumulate during forward propagation and affect the model outputs [40]. We feed the original image and denoised variants into the pre-trained DNN and obtain three vectors $\mathbf{v}$, $\mathbf{v_1}$, and $\mathbf{v_2}$, which are the softmax outputs of the model. We compute the ℓ_1-norm distances of ($\mathbf{v}$, $\mathbf{v_1}$) and ($\mathbf{v}$, $\mathbf{v_2}$), respectively, and then aggregate the results to obtain the amplified noise difference. The susceptibility can thus be mathematically defined as:

$$\mathbf{s} = ||\mathbf{v} - \mathbf{v_1}||_1 + ||\mathbf{v} - \mathbf{v_2}||_1 \tag{6}$$

Finally, we obtain the sanitized dataset by removing a portion of samples with the highest susceptibility. Note that users just need to run UltraClean once and any model trained on the cleansed dataset will be backdoor-free.

4. Experiments

4.1. Experiment Settings

We consider multiple representative dirty-label and clean-label attacks within the threat model. Please note that attacks that disable training phase defense where attackers train the models do not align with our threat model. For dirty-label attacks, we implement BadNets [23], Blended and Random Pattern [9], and Trojan [42] following the original papers, and evaluate the effectiveness of UltraClean on CIFAR-10. For clean-label attacks, we implement Sinusoidal Signal Backdoor (SIG) [2]; Label Consistent Backdoor (LCBD) [58] and Hidden Trigger Backdoor (HTBD) [52] based on their open-source repositories, and adopt the same training algorithms and dataset as in original papers for assessment (i.e., GTSRB for SIG, CIFAR-10 for LCBD, and ImageNet for HTBD). The mechanisms of attacks, detailed training settings, statistics of datasets, and neural network architectures are also summarized in the supplementary materials.

4.2. Evaluation on Dirty-Label Attacks

We first demonstrate UltraClean's effectiveness against various dirty-label attacks and refer to the SOTA defense against such attacks – the Frequency Detection (FD) [67] as the baseline for comparison. Note that FD is ineffective against clean-label attacks, as shown in [67]. We follow the original poisoning practice and set the blended injection ratio to 0.2 and Trojan transparency to 0.5, respectively. The results are presented in Table 2. UltraClean has an average of 97.78% detection rate and an average of 97.93% ASR reduction at a removal threshold of 0.3, achieving comparable performance to FD and indicating a strong capability of defending against dirty-label attacks. In practice, users usually do not know how many poisoned samples are injected. The selection of β should depend on the specific requirement of the application. Under safety-critical scenarios, we consider grid search a practical and systematic algorithm to determine the best value of β given an accuracy threshold.

Class (ID)	Acc. (PC)	ASR (PC)	Acc. (SVD)	ASR (SVD)	BDR (SVD)	Acc. (UC)	ASR (UC)	BDR (UC)
(0)	95.83%	47.58%	96.18%	58.36%	43.48%	96.52%	**8.18%**	**52.17%**
(1)	97.22%	54.15%	96.18%	62.82%	44.87%	95.49%	**10.47%**	**56.41%**
(2)	95.83%	67.62%	96.52%	**26.33%**	**67.21%**	96.87%	61.92%	50.82%
(3)	95.14%	17.38%	97.22%	2.13%	50.99%	95.14%	**0.00%**	**69.09%**
(4)	95.49%	51.53%	95.49%	40.46%	**57.43%**	96.18%	**1.91%**	46.62%
(5)	97.57%	51.78%	95.83%	53.75%	42.56%	93.05%	**37.15%**	**52.82%**
(6)	95.49%	74.16%	96.18%	62.92%	40.16%	96.52%	**17.98%**	**62.99%**
(7)	95.83%	63.83%	96.88%	62.92%	30.77%	97.22%	**55.32%**	**51.92%**
(8)	97.92	73.41%	96.18%	75.79%	40.70%	96.18%	**40.08%**	**48.74%**
(9)	97.57%	47.18%	93.06%	22.54%	64.58%	97.22%	**6.34%**	**91.66%**

Table 1. Comparison of accuracy ASR and BDR against SIG attack on GTSRB (SVD v.s. UltraClean). Numbers in the parenthesis are poisoned class IDs

Attack Type	Acc. (PC)	ASR (PC)	Acc. (UC)	ASR (UC)	BDR (UC)	BDR (FD)
BadNets	85.27%	100.00%	83.91%	**0.83%**	**94.42%**	90.50%
Trojan	84.94%	99.19%	84.73%	**1.61%**	99.50%	**99.99%**
Blended (HK)	84.99%	97.47%	85.08%	**3.06%**	**97.38%**	96.30%
Blended (RP)	86.24%	99.96%	84.23%	**0.15%**	**99.80%**	96.30%

Table 2. Performance comparison of UltraClean and prior SOTA Frequency Detection against dirty-label attacks.

4.3. Evaluation on Clean-Label Attacks

We are more interested in the performance of UltraClean on clean-label attacks since there is no known effective poisons-filtering-based defense against such stealthy attacks. Thus, we comprehensively evaluate the performance of UltraClean against clean-label attacks. We adapt SVD [57] and STRIP [19] as the baseline methods for comparison. We follow the works of SVD and STRIP, and consider two different scenarios:

1) **detection on the poisoned class** where we assume that the defender already knows the poisoned class (i.e., target class);
2) **detection on the whole training dataset** where the defender does not have knowledge of the data poisoning process.

We first present the results of detection on the poisoned class and then the detection on the whole training dataset. The later scenario is more practical and challenging.

4.4. Detection on the Poisoned Class

SIG on GTSRB. We follow the original recipe of SIG and set the frequency $f = 6$ and the strength $\Delta = 20$ to craft poisoned samples. To comprehensively evaluate the effectiveness of UltraClean, we iterate through each class as the target class and inject 30% poisoned samples to the target class. For a fair comparison, we hold the same removal threshold as SVD. We present the post-clean accuracy, ASR and BDR of classes with the top 10 attack success rates in Table 1. Our proposed method achieves a higher detection rate in all classes except classes 2 and 4, outperforming SVD by a large margin in general. Meanwhile, UltraClean and SVD do not undermine the model performance after retraining on the sanitized dataset. Note that although Δ is fixed during training, the adversary can raise the signal strength to achieve better ASR at test time. Therefore, for the evaluation of ASR, we set Δ to 80 to achieve the best attack performance. It can be seen that UltraClean significantly reduces the post-clean ASR and outperforms SVD in all classes except class 2. For classes 3, 4, and 9, UltraClean can even achieve nearly 0% post-clean ASR.

LCBD on CIFAR-10. We pick class "airplane" as the target class and poison the dataset with 4% of the entire images. LCBD uses two approaches to craft hard-to-classify images. For the GAN-based method, parameter τ controls the interpolation between two images. For the AE-based method, parameter ϵ is the maximum perturbation added on images in ℓ_p-norm. The adversary can construct various poisoned samples by varying these parameters. We present the settings that achieve the best attack performance and summarize the results of detection rate, ASR, and model accuracy in Table 3. UltraClean shows superior performance on the defense against LCBD, achieving a much higher detection rate and lower ASR than SVD under all the settings. Meanwhile, UltraClean also retains decent post-clean model accuracy. An interesting phenomenon is that the post-clean ASR of SVD is even worse than the pre-clean ASR, which was also revealed in the SIG experiments. We argue the rationale behind this is that SVD removes less poisoned samples and more benign samples, rendering a higher percentage of poisoned samples in the entire dataset after detection. Another possible reason is that the poisoned samples removed by UltraClean may play a more critical role in embedding the backdoor than those removed by SVD.

Attack Type	Acc.	ASR	BDR
	Pre-clean (PC)		
GAN (τ = 0.3)	88.17%	83.03%	/
AE (ℓ_2, ϵ = 1200)	87.73%	99.98%	/
AE (ℓ_∞, ϵ = 32)	87.84%	97.20%	/
	SVD		
GAN (τ = 0.3)	86.54%	97.96%	52.05%
AE (ℓ_2, ϵ = 1200)	87.17%	99.72%	80.95%
AE (ℓ_∞, ϵ = 32)	86.97%	99.82%	69.85%
	UltraClean (UC)		
GAN (τ = 0.3)	86.98%	**26.94%**	**79.75%**
AE (ℓ_2, ϵ = 1200)	87.26%	**1.10%**	**98.55%**
AE (ℓ_∞, ϵ = 32)	87.60%	**1.13%**	**97.15%**

Table 3. Comparison of backdoor detection and mitigation performance against LCBD attack on CIFAR-10.

HTBD on ImageNet. For the HTBD attack, we follow the same settings in the original paper [52] and conduct the experiment on 10 randomly selected target classes. A total of 100 poisoned samples are injected into the target class. Table 4 illustrates the detection rate, ASR and model accuracy. It can be seen that for most classes, poisoned samples completely bypass the SVD defense. However, UltraClean captures almost all poisoned samples and significantly undermines the backdoor effect to nearly 0% ASR. Moreover, UltraClean even improves the post-clean model accuracy.

4.5. Detection on the Whole Training Dataset

SIG on GTSRB. We run UltraClean with different removal thresholds (β changes from $0 \sim 0.3$) on the entire training dataset and present the results in Table 5. In the experiment, 148 poisoned samples (3% of the total training samples) are injected into the target class 5. ASR is evaluated at signal strength level $\Delta = 80$. UltraClean achieves >90% ASR reduction while maintaining a high test accuracy, indicating the effectiveness of UltraClean on the entire training dataset. On the other hand, STRIP fails to detect the poisoned samples. The entropy distribution of poisoned samples is mostly overlapped with benign samples, as shown in Figure 3 (left).

LCBD on CIFAR-10. We then evaluate UltraClean on the entire CIFAR-10 dataset. In this case, since AE (ℓ_2, $\epsilon = 1200$) has the best pre-clean ASR, we present the results under this setting. As shown in Table 5, UltraClean successfully thwarts the backdoor by diminishing the ASR to 1.59%, reaching up to 97.40% detection rate and 98.40% ASR reduction. Meanwhile, even with the maximum removal threshold, the post-clean model accuracy only drops by ~2.7%. Note that STRIP also performs well on defending against LCBD. According to Figure 3 (middle), the entropy of poisoned samples gathers around zero while the entropy of clean samples disperses in a wide range.

HTBD on ImageNet. The entire ImageNet dataset has more than one million training images. Precisely detecting poisoned samples from such a large-scale dataset is an extremely challenging task. However, UltraClean still acquires considerable success when facing such a complex dataset. In the experiment, we reproduce the attack in the original paper by injecting only 400 poisoned samples (~0.04% of total training samples) into the target "French bulldog" class. The results of UltraClean and STRIP are presented in Table 5 and Figure 3 (right). It can be observed that UltraClean is particularly effective against HTBD, while STRIP is unsuccessful in detecting the poisoned samples. With only a 5% removal threshold, UltraClean achieves a nearly 90% detection rate while maintaining a similar level of model accuracy.

Target Class		Acc. (PC)	ASR (PC)	BDR (SVD*)
1	Terrier	96.00%	45.25%	0.00%
2	Bee	97.00%	75.00%	0.00%
3	Plunger	95.00%	74.50%	55.00%
4	Partridge	97.00%	87.75%	0.00%
5	Ipod	95.00%	44.50%	8.00%
6	Deerhound	95.00%	83.75%	0.00%
7	Cockatoo	96.00%	78.00%	0.00%
8	Toyshop	95.00%	80.50%	0.00%
9	Tiger beetle	98.00%	58.00%	0.00%
10	Goblet	95.00%	89.75%	0.00%
Target Class		**Acc. (UC)**	**ASR (UC)**	**BDR (UC)**
1	Terrier	100.00%	**0.00%**	**97.00%**
2	Bee	99.00%	**0.00%**	**86.00%**
3	Plunger	97.00%	**4.75%**	**86.00%**
4	Partridge	100.00%	**0.00%**	**87.00%**
5	Ipod	100.00%	**0.00%**	**96.00%**
6	Deerhound	100.00%	**0.25%**	**96.00%**
7	Cockatoo	98.00%	**0.00%**	**99.00%**
8	Toyshop	98.00%	**4.50%**	**72.00%**
9	Tiger beetle	100.00%	**0.00%**	**100.00%**
10	Goblet	99.00%	**27.50%**	**86.00%**

Table 4. Comparison of clean accuracy, ASR and BDR against HTBD attack on ImageNet ("*" denotes results replicated from [52]).

4.6. Performance on Clean Datasets

UltraClean is designed to detect and mitigate backdoor attacks when poisoned samples are injected into training datasets. However, in real-world scenarios, users typically do not know whether the training dataset is poisoned. Therefore, we also study the performance of UltraClean (removal threshold = 0.3) on clean dataset and dataset with Gaussian noise. The results are summarized in Table 6. We find that none of the models reveals a malicious backdoor behavior after training on these two datasets, and applying UltraClean does not affect the inference accuracy, which indicates that it is safe to apply UltraClean under any circumstance.

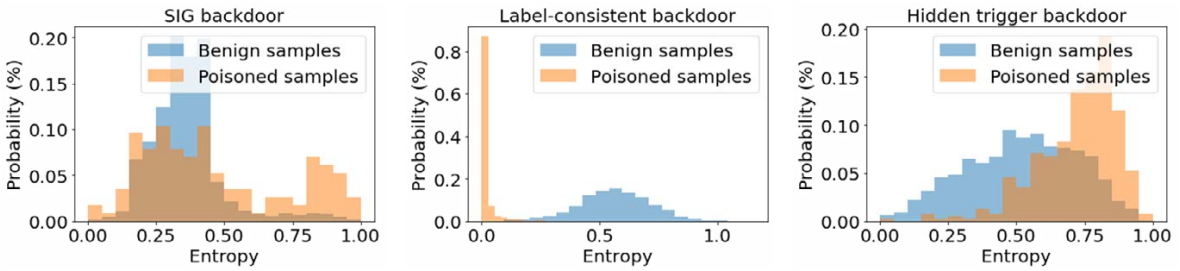

Figure 3. STRIP against SIG backdoor (left), LCBD (middle) and HTBD (right). It fails to detect backdoor samples crafted by SIG and HTBD.

Removal Threshold	BDR	Acc.	ASR	ASR(▼) Reduction
		SIG on GTSRB		
0.00	0.00%	95.49%	51.53%	-
0.10	9.45%	96.52%	40.08%	**22.22%**
0.20	25.00%	93.40%	10.69%	**79.25%**
0.30	39.86%	88.54%	1.91%	**96.29%**
		LCBD on CIFAR-10		
0.00	0.00%	87.73%	99.98%	-
0.10	88.80%	87.29%	74.95%	**25.00%**
0.20	94.65%	86.66%	52.94%	**47.49%**
0.30	97.40%	85.00%	1.59%	**98.40%**
		HTBD on ImageNet		
0.00	0.00%	50.27%	48.00%	-
0.01	57.25%	50.06%	9.20%	**80.83%**
0.02	68.75%	50.26%	1.60%	**96.00%**
0.03	78.00%	50.18%	0.20%	**99.58%**
0.04	83.50%	50.10%	0.00%	**100.00%**

Table 5. UltraClean performance on entire dataset.

Threshold (β = 0.30)	Regular		UltraClean	
	Acc.	ASR	Acc.	ASR
Clean	85.97%	0.0%	85.89%	0.0%
Gaussian Noise	85.03%	0.8%	85.49%	0.0%

Table 6. Performance of UltraClean on clean dataset and Gaussian noise dataset

4.7. Direct Training on Denoised Datasets

We also empirically evaluate if we can train backdoor-free models directly on denoised datasets. We apply non-local mean and local median denoising functions to the poisoned dataset crafted by BadNets. As shown in Table 7, directly training on the denoised dataset fails to mitigate backdoor in the model. These experimental results further prove that simply applying image denoising functions is ineffective in cleansing the poisoned dataset, which aligns with the observation in Figure 2.

4.8. Robustness against Adaptive Attacks

We also assess the performance of UltraClean against adaptive attacks to further demonstrate its capability. We consider a scenario where the adversary attempts to reduce the noise

	Acc.	ASR
Regular	85.27%	100.00%
Median	83.30%	89.75%
Mean	84.16%	90.00%

Table 7. Comparison of accuracy and ASR between training on regular poisoned dataset and denoised poisoned dataset (BadNets).

by altering adversarial perturbation to evade the defense of UltraClean. Here, as an example, we showcase the results of LCBD in Table 8 by lowering the interpolation ratio to make the attack more stealthy. As shown in the table, UltraClean achieves consistent superior performance against all adaptive attacks. A more comprehensive study is presented in the supplementary materials.

Attack Type	Acc.	ASR	BDR
	UltraClean (UC)		
GAN (τ = 0.0)	86.70%	**21.56%**	**72.70%**
GAN (τ = 0.1)	86.82%	**47.59%**	**70.95%**
GAN (τ = 0.2)	87.08%	**25.01%**	**78.00%**
AE (ℓ_2, ϵ = 300)	87.05%	**25.50%**	**75.55%**
AE (ℓ_2, ϵ = 600)	86.77%	**27.74%**	**88.65%**
AE (ℓ_∞, ϵ = 8)	87.71%	**23.16%**	**75.80%**
AE (ℓ_∞, ϵ = 16)	86.81%	**35.16%**	**89.40%**

Table 8. Robustness of UltraClean against adaptive LCBD attacks on CIFAR-10.

5. Conclusion

This paper presented a general defensive framework, UltraClean, against various backdoor attacks. It effectively differentiates poisoned and benign samples and significantly reduces the backdoor attack success rate. Comprehensive experiments and analysis validate the effectiveness of UltraClean in defending against both dirty-label and clean-label attacks.

ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation (NSF) under grant numbers 2426299, 2413046, and 2532588.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. 13

[2] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in CNNS by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing, ICIP*, pages 101–105. IEEE, 2019. 1, 2, 5, 12, 13

[3] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML*. icml.cc / Omnipress, 2012. 1

[4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016. 1

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems NeurIPS*, 2020. 1

[6] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, pages 60–65. IEEE Computer Society, 2005. 4

[7] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence AAAI*, 2019. 3

[8] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, pages 4658–4664, 2019. 3

[9] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017. 1, 2, 5, 12, 13

[10] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence*, pages 1148–1156. AAAI Press, 2021. 2

[11] Edward Chou, Florian Tramèr, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops, SP Workshops*, pages 48–54. IEEE, 2020. 3

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 13

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics, 2019. 1

[14] Bao Gia Doan, Ehsan Abbasnejad, and Damith C. Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *ACSAC '20: Annual Computer Security Applications Conference*, pages 897–912. ACM, 2020. 3

[15] Khoa Doan, Yingjie Lao, and Ping Li. Backdoor attack with imperceptible input and latent modification. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18944–18957, 2021. 2

[16] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. LIRA: learnable, imperceptible and robust backdoor attacks. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 11946–11956. IEEE, 2021. 2

[17] 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 *9th International Conference on Learning Representations, ICLR*, 2021. 1

[18] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *8th International Conference on Learning Representations, ICLR*, 2020. 3

[19] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith Chinthana Ranasinghe, and Surya Nepal. STRIP: a defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference, ACSAC*, pages 113–125. ACM, 2019. 3, 6

[20] Jonas Geiping, Liam H. Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient

matching. In *9th International Conference on Learning Representations, ICLR*, 2021. 17

[21] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *CoRR*, abs/2012.10544, 2020. 2

[22] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR*, 2015. 1, 12

[23] 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, 4, 5, 12, 13

[24] Wenbo Guo, Lun Wang, Yan Xu, Xinyu Xing, Min Du, and Dawn Song. Towards inspecting and eliminating trojan backdoors in deep neural networks. In *20th ICDM*, 2020. 3

[25] Yuning Han, Bingyin Zhao, Rui Chu, Feng Luo, Biplab Sikdar, and Yingjie Lao. Uibdiffusion: Universal imperceptible backdoor attack for diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 19186–19196. Computer Vision Foundation / IEEE, 2025. 1

[26] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. SPECTRE: defending against backdoor attacks using robust statistics. *CoRR*, abs/2104.11315, 2021. 3

[27] 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*, pages 770–778, 2016. 1

[28] Sanghyun Hong, Varun Chandrasekaran, Yigitcan Kaya, Tudor Dumitras, and Nicolas Papernot. On the effectiveness of mitigating data poisoning attacks with gradient shaping. *CoRR*, abs/2002.11497, 2020. 3

[29] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. *CoRR*, abs/2202.03423, 2022. 2, 3

[30] Shanjiaoyang Huang, Weiqi Peng, Zhiwei Jia, and Zhuowen Tu. One-pixel signature: Characterizing CNN models for backdoor detection. In *Computer Vision - ECCV 2020 - 16th European Conference, Proceedings, Part XXVII*, pages 326–341. Springer, 2020. 3

[31] Xijie Huang, Moustafa Alzantot, and Mani B. Srivastava. Neuroninspect: Detecting backdoors in neural networks via output explanations. *CoRR*, abs/1911.07399, 2019. 3

[32] Mojan Javaheripi, Mohammad Samragh, Gregory Fields, Tara Javidi, and Farinaz Koushanfar. Cleann: Accelerated trojan shield for embedded neural networks. In *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9. IEEE, 2020. 3

[33] Kaidi Jin, Tianwei Zhang, Chao Shen, Yufei Chen, Ming Fan, Chenhao Lin, and Ting Liu. A unified framework for analyzing and detecting malicious examples of dnn models. *arXiv preprint arXiv:2006.14871*, 2020. 3

[34] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 298–307. Computer Vision Foundation / IEEE, 2020. 3

[35] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Trans. Dependable Secur. Comput.*, 18(5):2088–2105, 2021. 2

[36] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *CoRR*, abs/2007.08745, 2020. 2

[37] 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. 2

[38] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *9th International Conference on Learning Representations, ICLR*, 2021. 3

[39] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *CoRR*, abs/2110.11571, 2021. 2, 3, 17, 19

[40] Ji Lin, Chuang Gan, and Song Han. Defensive quantization: When efficiency meets robustness. In *7th International Conference on Learning Representations, ICLR*, 2019. 2, 5

[41] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID Proceedings*, pages 273–294. Springer, 2018. 3

[42] 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*. The Internet Society, 2018. 5, 12, 13

[43] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part X*, pages 182–199. Springer, 2020. 2

[44] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 86–94. IEEE Computer Society, 2017. 1, 2

[45] Tuan Anh Nguyen and Anh Tuan Tran. Input-aware dynamic backdoor attack. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2

[46] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *9th International Conference on Learning Representations, ICLR*, 2021. 1, 2

[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An

imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 13

[48] Andrea Paudice, Luis Muñoz-González, and Emil C. Lupu. Label sanitization against label flipping poisoning attacks. *CoRR*, abs/1803.00992, 2018. 16

[49] Ximing Qiao, Yukun Yang, and Hai Li. Defending neural backdoors via generative distribution modeling. In *Advances in Neural Information Processing Systems 32, NeurIPS*, pages 14004–14013, 2019. 3

[50] Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 363–377, 2021. 3

[51] Erwin Quiring and Konrad Rieck. Backdooring and poisoning neural networks with image-scaling attacks. In *2020 IEEE Security and Privacy Workshops, SP Workshops*, pages 41–47. IEEE, 2020. 2

[52] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 11957–11965. AAAI Press, 2020. 1, 2, 4, 5, 7, 12, 13, 17

[53] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016. 1

[54] Hossein Souri, Liam Fowl, Rama Chellappa, Micah Goldblum, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *Advances in Neural Information Processing Systems*, 35:19165–19178, 2022. 17, 19

[55] 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. 13

[56] 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*, pages 1541–1558. USENIX Association, 2021. 3

[57] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, pages 8011–8021, 2018. 3, 6

[58] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *CoRR*, abs/1912.02771, 2019. 1, 2, 4, 5, 12, 13, 17

[59] Sakshi Udeshi, Shanshan Peng, Gerald Woo, Lionell Loh, Louth Rawshan, and Sudipta Chattopadhyay. Model agnostic defence against backdoor attacks in machine learning. *arXiv preprint arXiv:1908.02203*, 2019. 3

[60] 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, 2019. 3

[61] Mei Wang and Weihong Deng. Deep face recognition: A survey. *Neurocomputing*, 429:215–244, 2021. 1

[62] 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, Proceedings, Part XXIII*, 2020. 3

[63] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021. 17, 19

[64] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 501–509. Computer Vision Foundation / IEEE, 2019. 2

[65] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS*. The Internet Society, 2018. 2

[66] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, and Bo Li. Detecting AI trojans using meta neural analysis. In *42nd IEEE Symposium on Security and Privacy, SP*, pages 103–120. IEEE, 2021. 3

[67] Yi Zeng, Won Park, Z. Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV*, pages 16453–16461, 2021. 3, 5

[68] Bingyin Zhao and Yingjie Lao. Resilience of pruned neural network against poisoning attack. In *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*, pages 78–83. IEEE, 2018. 3

[69] Bingyin Zhao and Yingjie Lao. Clpa: Clean-label poisoning availability attacks using generative adversarial nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9162–9170, 2022. 2

[70] Bingyin Zhao and Yingjie Lao. Towards class-oriented poisoning attacks against neural networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3741–3750, 2022. 2

[71] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *8th International Conference on Learning Representations, ICLR*, 2020. 3

[72] Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David J. Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. In *CODASPY '20: Tenth ACM Conference on Data and Application Security and Privacy*, pages 97–108. ACM, 2020. 2

[73] Liuwan Zhu, Rui Ning, Cong Wang, Chunsheng Xin, and Hongyi Wu. Gangsweep: Sweep out neural backdoors by GAN. In *MM '20: The 28th ACM International Conference on Multimedia,*, pages 3173–3181. ACM, 2020. 3