Two Sides of the Same Coin: Learning the Backdoor to Remove the Backdoor

Qi Zhao and Christian Wressnegger

KASTEL Security Research Labs
Karlsruhe Institute of Technology (KIT)

Abstract

The community has recently developed various training-time defenses to counter neural backdoors introduced through data poisoning. In light of the observation that a model learns poisonous samples responsible for the backdoor easier than benign samples, these approaches either use a fixed threshold of the training loss for splitting or iteratively learn a reference model as an oracle for identifying benign samples. In particular, the latter has proven effective for anti-backdoor learning. Our method, HARVEY, leverages a similar yet crucially different technique: learning an oracle for poisonous rather than benign samples. Learning a *backdoored reference model* is significantly easier than learning one on benign data. Consequently, we can identify poisonous samples much more accurately than related work identifies benign samples. This crucial difference enables near-perfect backdoor removal as we demonstrate in our evaluation. HARVEY substantially outperforms related approaches across attack types, datasets, and architectures, lowering the attack success rate to the very minimum at a negligible loss in natural accuracy.

Supplementary — https://intellisec.de/research/harvey

1 Introduction

Learning an expressive deep neural network (DNN) requires large amounts of training data, which is oftentimes retrieved from third-party resources in practice (Carlini et al. 2023). Using such an external dataset without review may introduce security threats via data poisoning (Biggio and Roli 2018). The adversary may sneak a small portion of poisonous samples into the training dataset to introduce a neural backdoor. Such a backdoor shortcuts the prediction toward a predefined target label based on a trigger pattern (Gu et al. 2017; Chen et al. 2017; Liu et al. 2018b; Nguyen and Tran 2020, 2021; Barni et al. 2019) and can be established via data poisoning in two ways: First, *dirty-label attacks* (Gu et al. 2017; Chen et al. 2017; Liu et al. 2018b; Nguyen and Tran 2020, 2021) construct the trigger pattern on poisonous samples and relabel them to the target. Second, *clean-label attacks* (Turner et al. 2019; Shafahi et al. 2018; Zhao et al. 2023) strategically modify samples from the target class but do *not* change their labels.

Method	Defense Technique	Splitting Ratio	Architecture Independent	Natural Accuracy
ABL	Unlearn	Fixed	–	↘
CBD	Suppress	Adaptive	–	↘
DBD	Data Split	Fixed	–	↘
D-ST	Data Split	Fixed	–	↘
HARVEY	Data Split	Adaptive	●	↔

Table 1: Comparison of training-time backdoor defenses.

A wide variety of strategies have been proposed to alleviate backdooring attacks. *Type-1:* Model-based defenses either reverse-engineer the trigger pattern (Wang et al. 2023, 2022, 2019a), merely detect the existence of the backdoor (Cai et al. 2022; Xu et al. 2021; Wang et al. 2020), or erase it from the model (Liu et al. 2018a; Zhao et al. 2020; Li et al. 2021b). *Type-2:* Runtime defenses conduct differential testing (Doan et al. 2020), break the trigger functionality via data preprocessing (Qiu et al. 2021) or filter out abnormal inputs (Hayase et al. 2021; Gao et al. 2019). *Type-3:* Training-time defenses, in turn, suppress the backdoor during training, either using prior knowledge of a clean dataset (Zhou et al. 2023; Gao et al. 2023), or without such (Li et al. 2021a; Chen et al. 2022; Huang et al. 2022; Zhang et al. 2023).

All the strategies above have slightly different threat models, and only the latter tackles data poisoning at its roots. *Mitigating a backdoor during training without having a clean reference dataset is the most practical setting but it also is exceptionally difficult.*

In this paper, we propose HARVEY, a novel training-time defense in this setting, preventing backdoor injection by removing poisonous samples from the training data. In contrast to related work using unlearning (ABL; Li et al. 2021a) or backdoor suppression (CBD; Zhang et al. 2023), dataset splitting allows to preserve natural accuracy better. However, HARVEY splits the dataset adaptively without a fixed splitting ratio (DBD & D-ST; Huang et al. 2022; Chen et al. 2022), allowing us to preserve the natural performance much better as summarized in Table 1.

Our method builds on two crucial observations: First, we find that the reverse cross-entropy (RCE) component of the symmetrical cross-entropy (SCE) loss as used by prior work (Gao et al. 2023; Huang et al. 2022) is mainly respon-

sible for effectively splitting poisonous and benign samples. Second, using the loss to assemble a set of benign samples (Huang et al. 2022; Chen et al. 2022) is much more difficult than gathering poisonous samples in the same setting.

We thus learn a *strongly backdoored reference model* to continuously isolate poisonous samples using this model's RCE loss. HARVEY consists of the following four stages: ❶ **Initialization:** We naively train a model and split the dataset half-and-half in poisoned and benign subsets using the RCE loss. ❷ **Learning the backdoor:** We iteratively train the reference model by learning poisonous samples and unlearning benign samples using the split determined in the previous iteration. Over multiple rounds, the reference model becomes more and more specific to the backdoor functionality, which in turn improves RCE-loss-based splitting over time. ❸ **Meta-splitting:** Using the *first* reference model from the previous stage, we refine the poisonous subset yield through the *last* reference model. The former still has a notion of benign functionality allowing to isolate the remaining benign samples from the poisoned subset. ❹ **Final training:** Eventually, we train on the determined benign dataset to yield a perfectly clean model.

This procedure allows HARVEY to successfully remove a large variety of backdoor attacks across different model architectures and datasets, lowering the attack success rate (ASR) below $2\,\%$ *in the worst case* while preserving the natural accuracy. We hence outperform related work by a large margin. In summary, our contributions are three-fold:

- **Dataset splitting using** RCE **loss.** We analyze the commonly used symmetric cross-entropy (SCE) loss for dataset splitting, finding that its RCE component alone is suited much better for solving the task. HARVEY benefits from using RCE yielding more solid and stable splitting performance than related work.
- **Paradigm shift on using reference models.** We find that a *strongly backdoored reference model* tells poisonous and benign samples more reliably apart than any benign reference model can. This is inline with the early observation that poisonous samples are easier to learn than benign samples, but stands in contrast to how reference models are used in related work.
- **Decisive improvement in training-time defense.** We evaluate against various attacks across three model architectures and three datasets. None of the related approaches resist all attacks, while HARVEY suppresses the backdoor consistently and maintains natural accuracy on average and in the worst-case.

2 Taming Backdoors at Training-Time

We start by briefly describing the problem setting, before we analyze using SCE loss for dataset splitting (Section 2.1) and discuss the distribution of benign and poisonous samples based on the RCE loss (Section 2.2).

Problem definition. We consider backdoor injection through data-poisoning, that is, the adversary has no access to the training process, but can manipulate the training data. Naively training on this manipulated/poisoned training data $\tilde{\mathcal{D}}$ learns the "primary task" (i.e., image classification) but also introduces some additional malicious functionality as a "secondary task" (i.e., the neural backdoor).

More formally, the adversary poisons N_p samples of an existing dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ containing N samples $\mathbf{x}_i \in \mathbb{R}^d$ with the ground-truth label $y_i \in \{0, 1, \dots K-1\}$, where K denotes the number of classes. The resulting dataset $\tilde{\mathcal{D}}$ comprises a poisonous subset $\tilde{\mathcal{D}}_p$ and a clean subset $\mathcal{D}_c$, i.e., $\tilde{\mathcal{D}} = \tilde{\mathcal{D}}_p \cup \mathcal{D}_c$, and has the same size as the original dataset, $|\mathcal{D}| = |\tilde{\mathcal{D}}|$. The poisoning rate in $\tilde{\mathcal{D}}$ is $\rho = \frac{N_p}{N}$.

Training-time defense, aka. "anti-backdoor learning," aims to train a model on $\tilde{\mathcal{D}}$ with high natural accuracy and simultaneously counter the backdoor. Note, that the model trainer has no prior knowledge of the attack at all. Defenses using "dataset splitting" (Huang et al. 2022; Gao et al. 2023) separate the dataset $\tilde{\mathcal{D}}$ into a poisoned set $\mathcal{D}_{bad}$ and a benign set $\mathcal{D}_{bng}$, which is then used to train the final model. Thus, $\mathcal{D}_{bng}$ should contain as few poisonous samples as possible, $\rho_{bng} = \frac{|\mathcal{D}_{bng} \setminus \mathcal{D}_c|}{|\mathcal{D}_{bng}|} \approx 0.0$, ideally $\mathcal{D}_{bng} = \mathcal{D}_c$.

2.1 Analysis of Dataset Splitting Using SCE Loss

Prior defenses (Gao et al. 2023; Huang et al. 2022) treat poisonous samples as "label noise," and thus, use the SCE loss (Wang et al. 2019b) to isolate them. SCE loss consists of two terms, cross-entropy (CE) and reverse cross-entropy (RCE), weighted by α and β, respectively:

$$\mathcal{L}_{\text{SCE}} = \alpha \mathcal{L}_{\text{CE}} + \beta \mathcal{L}_{\text{SCE}}$$

In terms of the KL divergence (Kullback and Leibler 1951), optimizing the CE loss draws the prediction probability $p\,(k|\mathbf{x})$ of an input $\mathbf{x}$ wrt. a class k near the ground-truth probability distribution $q\,(k|\mathbf{x})$, minimizing $\text{KL}\,(q||p)$.

$$\mathcal{L}_{\text{CE}} = -\sum_{k=0}^{K-1} q\,(k|\mathbf{x}) \cdot \log p\,(k|\mathbf{x})$$

When training on data samples with noisy labels (Wang et al. 2019b), $q\,(k|\mathbf{x})$ does not represent the real ground truth, though. Hence, SCE additionally considers $\text{KL}\,(p||q)$ to push predictions on mislabeled samples toward $p\,(k|\mathbf{x})$ known as reverse CE:

$$\mathcal{L}_{\text{RCE}} = -\sum_{k=0}^{K-1} p\,(k|\mathbf{x}) \cdot \log q\,(k|\mathbf{x})$$

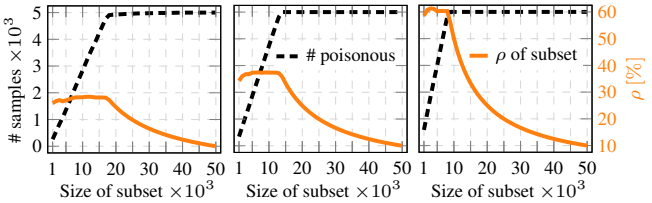
Given an arbitrary input $\mathbf{x}$ with the fixed label y, the distribution $q\,(y|\mathbf{x})$ equals 1 while for all other classes $q\,(k|\mathbf{x})\,|_{k\neq y}$ is ϵ-small and close to 0. We define $C = -\log \epsilon$ as being positive and rewrite the RCE loss as:

$$\begin{aligned}\mathcal{L}_{\text{RCE}} &= -p\,(y|\mathbf{x}) \cdot \log 1 + \sum_{k\neq y} p\,(k|\mathbf{x}) \cdot C \\ &= C \cdot (1 - p\,(y|\mathbf{x}))\end{aligned}$$

Apparently, only the prediction of the ground-truth y matters in the RCE loss. Hence, the training converges to the ground-truth distribution for easy-to-learn samples, yielding

a RCE loss very close to 0. Meanwhile, samples with hard-to-learn features have higher RCE loss close to value C.

SCE loss with default parameters (Wang et al. 2019b) ($C = -log10^{-5}$, $\alpha = 0.1$, $\beta = 1.0$ for CIFAR10) weighs the RCE loss significantly stronger than the CE loss. Still, comparing the distributions of all three losses using the example of DBD (Huang et al. 2022) reveals that the CE loss makes SCE loss less effective in distinguishing benign and poisonous samples, while RCE alone allows a clear separation of poisonous samples. Using RCE rather than SCE loss, hence, is sufficient (and even beneficial in terms of stability) for dataset splitting. More details are in the supplementary.



(a) Train on $\tilde{\mathcal{D}}$ (b) Train on $\mathcal{S}^{(0)}$ (c) Unlearn $\mathcal{D}_{bng}$

Figure 1: Subsets of incremental size chosen in ascending order of RCE loss based on a ResNet18 trained on a CIFAR10 poisoned by Blend (Chen et al. 2017) at $\rho = 10\,\%$.

2.2 Distribution of Benign and Poisonous Samples

Although the model learns poisonous samples (and thus the backdoor) easier than benign samples in the general case (Li et al. 2021a), after training, some benign samples still have a similar loss as poisonous samples. Hence, a fixed threshold on the loss *cannot* ensure splitting the dataset precisely. However, the majority of samples with low loss are indeed poisonous allowing to sample a subset with a high poisoning rate. Increasing the poisoning rate can intensifies the backdoor in a model (Li and Liu 2024; Wu et al. 2022).

Training on the entire dataset $\tilde{\mathcal{D}}$ yields low loss values for most poisonous samples already as shown in Fig. 1a. If we consider the $50\,\%$ of the samples with the lowest RCE loss to form a subset $\mathcal{S}^{(1)}$, this subset will naturally have a higher poisoning rate. Retraining on $\mathcal{S}^{(1)}$ yields the distribution shown in Fig. 1b with even more poisonous samples having low loss. This gives rise to adaptive dataset splitting similar as Zhang et al. (2023) proposed for CBD *but focusing on poisonous samples instead of benign samples*.

If we additionally unlearn the $50\,\%$ not selected for $\mathcal{S}^{(1)}$ (samples with high RCE loss and thus mostly benign) by maximizing the training loss, we yield even more poisonous samples with low loss in the resulting model as shown in Fig. 1c. The subset formed by the $10\,\text{k}$ samples with lowest loss has an exceptionally high poisoning rate, forming the ideal basis for the next round of splitting.

Learning a *strongly backdoored model* can serve as an oracle for poisonous samples. In the next section, we introduce our method that uses such an oracle as "reference model" to identify and filter out poisonous samples, before learning a backdoor-free model on the remaining clean data samples.

3 HARVEY: Learning a Backdoor Oracle

We propose a novel training-time defense against neural backdoors based on dataset splitting. In contrast to related work, we tackle the problem from a different perspective: We focus on poisonous samples rather than benign samples and *iteratively learn a strongly backdoored model as an oracle for poisonous samples*. This way our method does not require a clean reference dataset, but bootstraps iterative splitting with the most obvious poisonous samples and improves continuously. Fig. 2 depicts the multi-stage, working principle of our method, HARVEY. Additionally, we provide an algorithmic description in the supplementary. In the following subsections, we elaborate on each stage individually.

3.1 Initialization (Stage ❶)

We begin by training T_{init} epochs on $\tilde{\mathcal{D}}$ to obtain a naively trained model θ^*. Based on the observations made in Section 2, we retrieve the $50\,\%$ of the samples with the lowest RCE loss as the initial poisonous subset $\mathcal{D}_{bad}^{(0)}$, which we use for the further processing in the next stage entirely, $\mathcal{S}^{(1)} = \mathcal{D}_{bad}^{(0)}$. The remaining $50\,\%$ form the initial benign set $\mathcal{D}_{bng}^{(0)}$. Note that this yields a reasonable split because poisonous samples are learned easier than benign samples in general (Li et al. 2021a). The following stages refine this split over multiple rounds to constantly improve separation up until we are ready to learn the final backdoor-free model.

3.2 Learning the Backdoor (Stage ❷)

We discard the initial naively trained model θ^* and train a new reference model θ_{ref} from scratch that we iteratively fine-tune over n rounds. Each round fulfills two tasks: First, we enhance the reference model's notion of the backdoor by fine-tuning[1] $\theta_{ref}^{(i-1)}$ yielding $\theta_{ref}^{(i)}$. Second, we use the latter to split $\tilde{\mathcal{D}}$ into a poisoned set $\mathcal{D}_{bad}^{(i)}$ and a benign set $\mathcal{D}_{bng}^{(i)}$. The poisoned set serves as basis for sub-sampling data into $\mathcal{S}^{(i+1)}$ used for the next iteration.

Enhancing the backdoored reference model. We use a learning-unlearning strategy to strongly internalize the backdoor. First, we train on $\mathcal{S}^{(i)}$ for T_{poi} epochs and employ the local gradient ascent (LGA) loss (Li et al. 2021a) with γ set to 0.01 to avoid overfitting to the few benign samples contained in the subset $\mathcal{S}^{(i)}$:

$$\mathcal{L}_{\text{LGA}} = \sum_{(\mathbf{x},y)\in\mathcal{S}^{(i)}} sign\left(\mathcal{L}_{\text{CE}}\left(\mathbf{x}, y, \theta_{ref}^{(i)}\right) - \gamma\right)\cdot\mathcal{L}_{\text{CE}}\left(\mathbf{x}, y, \theta_{ref}^{(i)}\right)$$

Next, we unlearn the benign samples in the benign subset $\mathcal{D}_{bng}^{(i-1)}$ by *maximizing* the CE loss. We use the following (negative) loss function, with $\lambda = 0.01$ to stabilize the training. The indicator function $\mathbb{1}(\cdot)$ ensures to only consider samples for unlearning that were predicted correctly by the reference model:

$$\mathcal{L}_{\text{UL}} = \sum_{(\mathbf{x},y)\in\mathcal{D}_{bng}^{(i-1)}} -\lambda\cdot\mathbb{1}\left(f_{\theta_{ref}^{(i)}}(\mathbf{x}) = y\right)\cdot\mathcal{L}_{\text{CE}}\left(\mathbf{x}, y, \theta_{ref}^{(i)}\right)$$

[1] $\theta_{ref}^{(0)}$ is randomly initialized at the beginning of Stage ❷.

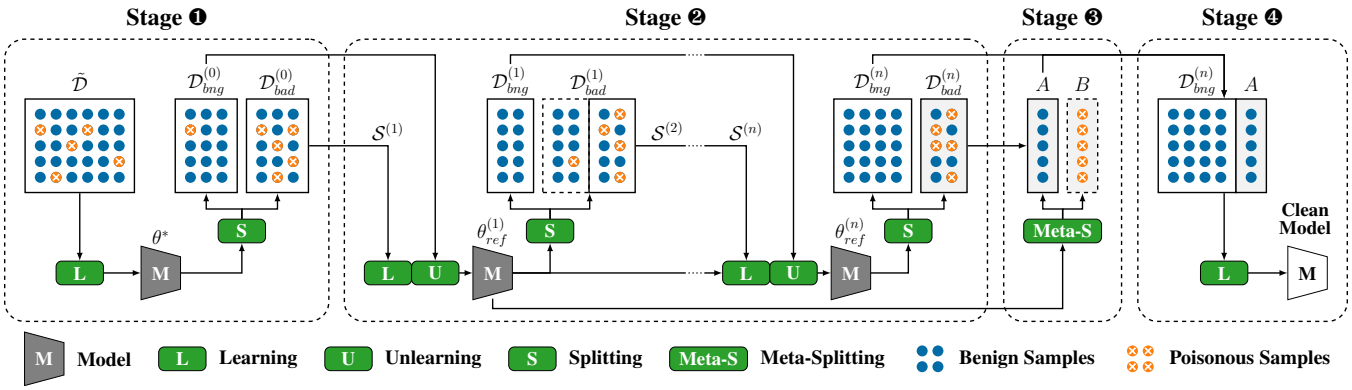


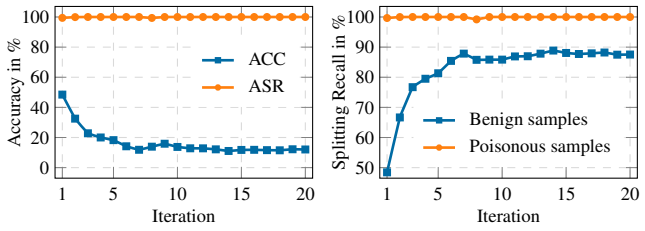
Figure 2: Overview depiction of our method HARVEY. The individual stages are described in Sections 3.1 to 3.4.

We perform unlearning for only one epoch per round. This way, we avoid a learning divergence and help the model to stably unlearn the benign samples.

Splitting dataset $\tilde{\mathcal{D}}$ and composing $\mathcal{S}^{(i+1)}$. The reference model $\theta_{ref}^{(i)}$ encodes the backdoor much better than benign data. There hence exists a huge gap in the RCE loss (cf. Fig. 6b) that separates poisonous and most benign samples. The RCE loss has a fixed output range of $[0, C]$, so that a threshold of $C/2$ splits the dataset well in a poisonous part $\mathcal{D}_{bad}^{(i)}$ (low RCE loss) and clean part $\mathcal{D}_{bng}^{(i)}$ (high RCE loss). We then retrieve $50\,\%$ of the samples with the lowest RCE loss from $\mathcal{D}_{bad}^{(i)}$ to be used as $\mathcal{S}^{(i+1)}$ for the next iteration.

Note that it may happen that some classes are not represented in $\mathcal{S}^{(i+1)}$. To counteract any bias, we additionally append samples with low RCE loss from $\tilde{\mathcal{D}} \setminus \mathcal{S}^{(i+1)}$ so that all classes have at least $1\,\%$ of the original amount of samples. Doing so will not weaken the backdoor in the reference model but preserves the learning behavior across all classes.

Fig. 3 shows the attack success rate (ASR) and natural accuracy (ACC) of the reference models $\theta_{ref}^{(i)}$ on the left and the splitting performance measured as recall on the right. The process stabilizes starting at round 10, yielding a decent split in poisonous and benign samples. However, we cannot ensure perfect splitting this way as some benign samples' loss overlaps with that of poisonous samples (cf. Fig. 6c). We thus employ "meta splitting" as described next.



(a) Performance of θ_{ref} (b) Dataset Splitting

Figure 3: HARVEY's splitting performance using ResNet18 on CIFAR10 poisoned by Blend attack (Chen et al. 2017).

3.3 Meta-Splitting (Stage ❸)

As the last reference model $\theta_{ref}^{(n)}$ is strongly biased toward the backdoor, most of benign samples of the backdoor's target class remain in $\mathcal{D}_{bad}^{(n)}$. Hence, directly training on $\mathcal{D}_{bng}^{(n)}$ right after Stage ❷ results in low natural performance (cf. Fig. 4). The first reference model $\theta_{ref}^{(1)}$, in turn, yields two clusters in latent space (cf. Fig. 6b), which can separate poisonous and benign samples, especially from the target class. Consequently, we use $\theta_{ref}^{(1)}$ to split $\mathcal{D}_{bad}^{(n)}$ (cf. Fig. 6c) in two parts, A and B, according to high and low RCE loss for benign and poisonous samples, respectively (cf. Fig. 5).

3.4 Final Training (Stage ❹)

For the final training, we combine the isolated benign subsets, $\mathcal{D}_{bng}^{(n)}$ (Stage ❷) and A (Stage ❸), yielding the final clean dataset $\mathcal{D}_{bng}$, and reject the rest. We then directly apply supervised learning using CE loss on this data to obtain the final clean model without the backdoor:

$$\arg\min_{\theta} \sum_{(\mathbf{x},y)\in\mathcal{D}_{bng}} \mathcal{L}_{\text{CE}}\left(\mathbf{x}, y, \theta\right)$$

Analyzing the model in latent space (cf. Fig. 6d) shows that it forms clearly separable groups for classes and classifies poisonous samples in the original class when being applied to the original, poisoned dataset $\tilde{\mathcal{D}}$.

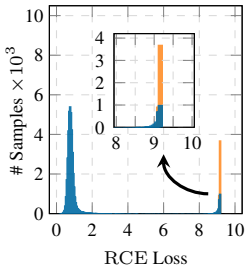


Figure 4: Train on $\mathcal{D}_{bng}^{(n)}$ of CIFAR10 poisoned by Blend w/o meta-splitting.

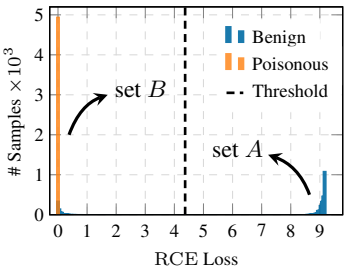
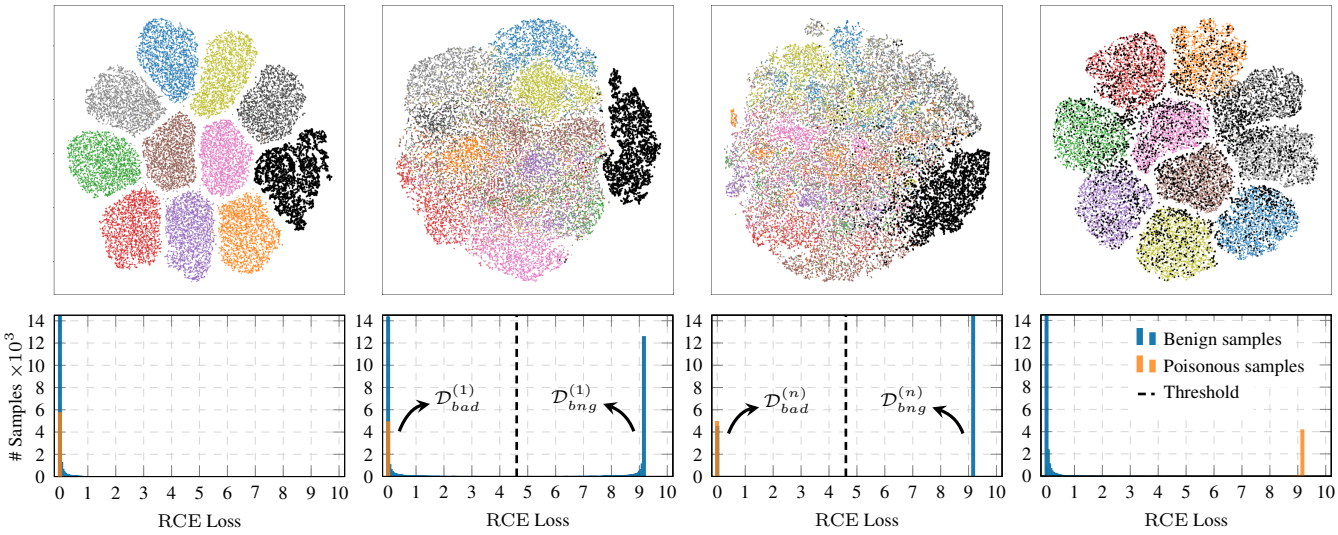


Figure 5: Meta-splitting with using $\theta_{ref}^{(1)}$ on $\mathcal{D}_{bad}^{(n)}$ of a CIFAR10 poisoned by Blend attack.



(a) Initial model θ^* (b) 1st Reference model $\theta_{ref}^{(1)}$ (c) n^{th} Reference model $\theta_{ref}^{(n)}$ (d) Final clean model θ

Figure 6: Dataset distribution of the latent space by t-SNE (top row) and RCE loss (bottom row) for ResNet18 on CIFAR10 poisoned by Blend attack. Poisonous samples are marked black, and other colors represent benign samples of different classes.

4 Evaluation

We conduct extensive experiments across datasets and model architectures to evaluate our method in comparison to other training-time defenses. Specifically, we evaluate on the small-scale dataset CIFAR10 (Krizhevsky et al. 2008) with WRN16-1 (Zagoruyko and Komodakis 2016) and ResNet18 (He et al. 2016), and the large-scale dataset Tiny-ImageNet (Le and Yang 2015) with ResNet34 (He et al. 2016). In the supplementary, we provide results for a second small-scale dataset, GTSRB (Stallkamp et al. 2012).

Below, we elaborate on the experimental setup wrt. considered attacks, related defenses and evaluation metrics, before we present an overview of HARVEY's performance in Section 4.1. In Sections 4.2 and 4.3, we analyze the splitting performance of our method and perform ablation experiments, respectively. In Section 4.4, we compare the runtime of all defenses with HARVEY. In the supplementary, we further provide the detailed experimental setup, additional ablation studies, and the evaluation with an adaptive attack.

Considered attacks. We evaluate with six poisoning attacks, including BadNets (Gu et al. 2017), Trojan (Liu et al. 2018b), Blend attack (Chen et al. 2017), Clean-Label (CLB; Turner et al. 2019), IAB (Nguyen and Tran 2020) and WaNet (Nguyen and Tran 2021). We choose class 0 as target, $y_t = 0$, and use a poisoning rate of $10\,\%$, $\rho = 0.1$, except for CLB, where we poison $50\,\%$ and $100\,\%$ of the samples of the target class for the small-scale and large-scale dataset, respectively.

Considered defenses. We compare HARVEY to four other defenses that also do *not* require a clean reference dataset: ABL (Li et al. 2021a), DBD (Huang et al. 2022), D-ST (Chen et al. 2022), and CBD (Zhang et al. 2023) with the parameters proposed in the corresponding publications. For HARVEY, we set $\epsilon = 10^{-5}$, so that the splitting threshold is $C/2 \approx 4.6052$. We use $T_{init} = 20$ epochs for the pretraining and $T_{poi} = 10$ epochs for each round/iteration in Stage ❷. Finally, we train the clean model for $T_{cln} = 100$ epochs.

Evaluation Metrics. We present the defensive performance with two metrics: Natural Accuracy (ACC) and Attack Success Rate (ASR). An optimal backdooring attack has a high ACC and an ASR of $100\,\%$. In contrast, an effective backdoor defense should achieve high ACC while the attack is ineffective, ASR $\approx 0\,\%$.

4.1 Defensive Capability of HARVEY

We provide the overall comparison of our method with other training-time defenses in Table 2, where we highlight the highest accuracy (ACC) and the lowest attack success rate (ASR). Additionally, we mark all settings where the backdoor cannot be mitigated (ASR > $90\,\%$) in orange color.

CIFAR10. All defenses are robust against patch-based attacks (e.g., BadNets) with ResNet18, while their performance decreases for stealthier attacks, such as Blend, CLB and WaNet. In particular, DBD and D-ST are affected and also with WRN16-1 results are worse due to its low model capacity. ABL and CBD, in turn, prevent various backdoors successfully but at the cost of natural accuracy. In contrast, our method is effective in both model architectures, also maintaining the natural accuracy. HARVEY reduces the attack success rate below $2\,\%$ *in the worst case* and preserves the highest natural performance of all defenses.

Tiny-ImageNet. Also on Tiny-ImageNet, our method excels with on par natural accuracy and a reduction of the attack success rate to merely $0.48\,\%$ in the worst case. D-ST performs well in preserving the natural accuracy but misplits the dataset specifically for Blend attacks. DBD, in turn

	Model	Attack	No Defense		ABL		DBD		D-ST		CBD		HARVEY	
			ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR10	WRN16-1	BadNets	89.80	**100.00**	82.72	4.46	70.53	6.61	89.04	**99.97**	87.95	**1.06**	**89.52**	1.50
		Trojan	89.91	**99.93**	84.91	4.49	69.89	14.23	**88.88**	**98.93**	87.85	**1.04**	88.82	1.83
		Blend	89.80	**98.78**	78.30	10.27	71.08	7.79	87.62	**99.80**	80.17	85.12	**89.40**	**0.99**
		CLB	90.96	**99.87**	80.74	0.83	70.16	**100.00**	87.06	**0.00**	88.18	**0.00**	**89.85**	0.83
		IAB	90.18	**99.80**	85.36	9.03	63.81	**98.18**	84.96	**99.68**	86.07	**1.04**	**88.72**	1.66
		WaNet	88.31	**98.55**	84.90	4.14	70.08	8.85	86.27	**98.81**	85.12	2.00	**87.15**	**0.30**
		Average	89.64	**99.49**	82.82	5.54	69.26	39.28	87.31	82.87	85.89	15.04	**88.91**	**1.19**
		Worst Case	88.31	**100.00**	78.30	10.27	63.81	**100.00**	84.96	**99.97**	80.17	85.12	**87.15**	**1.83**
	ResNet18	BadNets	93.80	**100.00**	90.67	0.37	93.24	14.65	92.27	**0.01**	88.19	1.48	**93.34**	0.56
		Trojan	93.58	**100.00**	91.59	1.24	93.31	**99.99**	**93.97**	**0.22**	91.40	1.92	93.96	0.93
		Blend	94.01	**100.00**	87.66	6.59	90.27	**99.91**	89.93	55.48	86.33	5.77	**93.37**	**0.58**
		CLB	94.25	**100.00**	79.39	0.84	93.01	**100.00**	91.96	0.26	92.83	**0.19**	**93.52**	0.49
		IAB	93.62	**100.00**	93.30	6.32	87.91	83.23	**93.60**	22.00	92.39	69.10	93.11	**0.77**
		WaNet	93.36	**99.92**	88.74	55.81	93.03	15.27	93.09	**99.72**	88.49	32.73	**93.56**	**1.22**
		Average	93.73	**99.99**	88.49	11.23	91.46	73.04	91.62	25.46	90.09	16.41	**93.56**	**0.85**
		Worst Case	93.36	**100.00**	79.39	55.81	87.91	**100.00**	86.51	**99.72**	86.33	69.10	**93.11**	**1.43**
Tiny-ImageNet	ResNet34	BadNets	57.03	**100.00**	46.26	**0.00**	50.88	**100.00**	56.10	0.16	49.21	0.27	**57.83**	**0.00**
		Trojan	56.65	**100.00**	47.43	**0.00**	51.88	**100.00**	56.14	0.02	52.10	0.10	**57.27**	0.06
		Blend	56.86	**99.98**	49.07	**99.99**	51.73	**100.00**	**56.56**	**97.63**	52.68	0.78	56.48	**0.07**
		CLB	57.43	**99.99**	49.93	0.01	51.62	**100.00**	**57.89**	0.01	50.01	0.84	57.37	**0.00**
		IAB	57.21	**99.37**	46.00	**0.00**	50.74	**100.00**	55.66	**0.00**	50.40	0.21	**57.24**	0.02
		WaNet	56.68	**99.82**	44.32	1.68	51.22	**100.00**	54.11	26.30	50.74	17.54	**56.31**	**0.48**
		Average	57.02	**99.86**	47.17	16.95	51.35	**100.00**	56.08	20.69	50.86	3.29	**57.08**	**0.11**
		Worst Case	56.65	**100.00**	44.32	**99.99**	50.74	**100.00**	54.11	**97.63**	49.21	17.54	**56.31**	**0.48**

Table 2: Comparison of HARVEY with prior defenses. All results are shown in %. The best results across all defenses are highlighted in **bold** font. Settings where the backdoor injection succeeds (ASR > 90 %) are marked in **orange bold** font.

fails completely due to the difficulty of learning benign features as the loss distribution of benign and poisonous samples is more interleaved. ABL and CBD ensure a better defense but sacrifice natural accuracy decisively.

Summary. Related approaches cannot defend against backdoors reliably, whereas our method can. The reason is that we learn an easier task: learning the backdoor rather than the benign, natural task. HARVEY yields a natural accuracy on par with the "No Defense" setting and suppresses each backdoor to the very minimum.

4.2 Performance in Dataset Splitting

Next, we analyze the performance of the methods based on dataset splitting (DBD, D-ST, and HARVEY) by measuring the F_1 score and the rate of remaining poisonous samples. Table 3 summarizes the splitting results.

For D-ST, we merge "poisoned" and "suspicious" subsets for the evaluation as D-ST only learns on it "clean" subset (Chen et al. 2022). The method falls behind for the smaller WRN16-1 model in particular as it cannot capture the benign task completely during splitting. For larger models (ResNet18 for CIFAR10 and ResNet34 for Tiny-ImageNet), D-ST leaves over less poisonous samples, except for WaNet and Blend. Moreover, DBD's shortcomings on Tiny-ImageNet are also visible in the splitting performance, where more than 5 % of the final dataset are still poisonous (in light of an overall poisoning rate of 10 %). Also for ResNet18 on CIFAR10, the recall is significantly lower than 90 % for Trojan, Blend, and CLB attacks.

HARVEY is largely independent of the model's capacity as we learn the backdoor rather than the benign task for the reference model. For CIFAR10 with WRN16-1, we yield F_1 scores over 66 %, which is 49 percentage points higher than the best related work. For Tiny-ImageNet, our method even reaches $F_1 > 95\,\%$ on average and a $\rho_{bng} < 0.1\,\%$.

4.3 Ablation Study

We study the impact of different components of our method using a ResNet18 model on poisoned CIFAR10 with the poisoning rate 10 %. Table 4 summarizes the results.

(a) Replacing RCE loss for dataset splitting. We use CE loss and SCE loss with a threshold of 5.0 to show the impact of using RCE loss. While the backdoor is decently removed with CE loss, the natural accuracy degrades significantly as the final $\mathcal{D}_{bng}$ lacks benign samples. The RCE term in SCE loss makes it remove poisonous samples better (and thus reduce the ASR) and also improves the natural accuracy. We can further improve the result by tuning the threshold, which, however, requires ground-truth knowledge of the poisoned samples. The fixed output range of RCE loss as used for HARVEY, in turn, allows to trivially split the data at $C/2$, being much more practical.

(b) Impact of unlearning. Next, we evaluate the influence of our unlearning step using $\mathcal{D}_{bng}^{(i-1)}$. As can be seen in Table 4 unlearning is crucial for maintaining natural accuracy. Learning a reference model on $\mathcal{S}^{(i)}$ alone that still contains many benign samples lowers splitting effectiveness.

	Model	Attack	DBD				D-ST				HARVEY (Ours)			
			Prec.	Recall	F_1	ρ_{bng}	Prec.	Recall	F_1	ρ_{bng}	Prec.	Recall	F_1	ρ_{bng}
CIFAR10	WRN16-1	BadNets	18.19	90.96	30.32	1.81	10.48	7.28	8.59	**9.96**	83.04	100.00	**90.73**	**0.00**
		Trojan	17.26	86.30	28.77	2.74	10.94	8.03	9.26	**9.43**	75.99	99.42	**86.14**	**0.07**
		Blend	18.24	91.18	30.40	1.76	8.25	7.54	7.88	**10.18**	73.32	99.48	**84.42**	**0.06**
		CLB	10.09	50.46	16.82	**9.91**	2.61	73.78	5.04	3.71	69.34	99.96	**81.88**	**0.00**
		IAB	4.39	21.94	7.32	**15.61**	21.15	96.34	34.69	0.67	53.04	99.80	**69.27**	**0.02**
		WaNet	11.47	57.36	19.12	**8.53**	13.65	42.34	20.64	**8.36**	49.93	99.50	**66.49**	**0.06**
		Average	13.27	66.37	22.12	**6.73**	11.18	39.22	14.35	**7.05**	67.44	99.69	**79.82**	**0.04**
		Worst Case	4.39	21.94	7.32	**15.61**	2.61	7.28	5.04	**10.18**	49.93	99.42	**66.49**	**0.07**
	ResNet18	BadNets	17.09	85.46	28.48	2.91	45.48	99.98	62.52	**0.00**	55.69	100.00	**71.54**	**0.00**
		Trojan	11.11	55.58	18.52	**8.88**	89.02	99.54	**93.99**	0.05	74.57	99.82	85.37	**0.02**
		Blend	8.39	41.94	13.98	**11.61**	14.59	98.59	25.42	0.44	69.79	100.00	**82.21**	**0.00**
		CLB	10.09	50.46	16.82	**9.91**	16.95	99.99	28.99	**0.00**	60.72	99.92	**75.54**	**0.00**
		IAB	8.42	42.10	14.03	**11.58**	45.19	89.24	60.00	1.34	67.62	100.00	**80.68**	**0.00**
		WaNet	17.17	85.84	28.62	2.83	29.17	23.62	26.10	**8.31**	67.37	98.54	**80.03**	**0.17**
		Average	12.05	60.23	20.08	**7.95**	40.07	85.16	49.50	1.69	65.96	99.71	**79.23**	**0.03**
		Worst Case	8.39	41.94	13.98	**11.61**	14.59	23.62	25.42	**8.31**	55.69	98.54	**71.54**	**0.17**
Tiny-ImageNet	ResNet34	BadNets	0.09	0.45	0.15	**19.91**	91.52	100.00	95.57	**0.00**	99.52	99.99	**99.75**	**0.00**
		Trojan	0.36	1.80	0.60	**19.64**	90.61	100.00	95.07	**0.00**	97.29	99.98	**98.62**	**0.00**
		Blend	0.08	0.38	0.13	**19.92**	6.77	3.42	4.54	**10.17**	89.48	99.93	**94.42**	**0.01**
		CLB	0.01	1.00	0.02	0.99	0.80	4.60	1.36	0.49	67.81	99.00	**80.49**	**0.01**
		IAB	0.10	0.49	0.17	**19.90**	89.48	99.93	94.42	**0.01**	96.00	99.81	**97.87**	0.02
		WaNet	0.82	4.11	1.37	**19.18**	84.91	91.81	88.23	0.92	95.54	99.48	**97.47**	**0.06**
		Average	0.24	1.37	0.48	**16.59**	60.68	66.63	75.57	1.93	90.94	99.70	**97.63**	**0.02**
		Worst Case	0.01	0.38	0.02	**19.92**	0.80	3.42	1.36	**10.17**	67.81	99.00	**80.49**	**0.06**

Table 3: Comparing HARVEY's dataset splitting with DBD and D-ST. Precision (Prec.) measures the ratio of poisonous samples in $\mathcal{D}_{bad}$. Recall shows the isolation percentage of poisonous samples. ρ_{bng} is the poisoning ratio in $\mathcal{D}_{bng}$. All results are shown in %. We highlight the best F_1 score and ρ_{bng} of all defenses in **bold** font. ρ_{bng} above 5 % is marked in **orange bold** font.

(c) Final training using semi-supervised learning (SSL). SSL is proven effective for training-time defenses (Huang et al. 2022; Chen et al. 2022). Using MixMatch (Berthelot et al. 2019) for the final training on $\mathcal{D}_{bng}$ and $\mathcal{D}_{bad}$ achieves a comparable natural accuracy to standard supervised learning (SL). Despite a slight improvement in defense, SSL's time overload (6× compared to SL) outweighs the benefit.

4.4 Time Consumption

Table 5 shows the runtime of all defenses. ABL and CBD are slightly faster than HARVEY. In contrast, other defenses are slower by another magnitude. In particular, DBD takes nearly one day on CIFAR10. While HARVEY is not the fastest, its runtime is comparable to the naive training while significantly improving the performance (cf.. Table 2).

Ablation	BadNets		Blend		IAB		WaNet	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
HARVEY	**93.34**	0.56	93.37	0.58	**93.11**	0.77	**93.56**	1.22
RCE → CE	89.85	0.23	87.82	3.01	88.44	0.58	84.07	**0.08**
RCE → SCE	90.90	0.16	93.01	0.61	92.76	0.68	89.82	0.26
Unlearn $\mathcal{D}_{bng}$	86.69	1.58	88.12	2.53	80.92	7.16	80.89	8.14
SL → SSL	92.36	**0.07**	**94.12**	**0.22**	92.87	**0.16**	92.44	0.48

Table 4: Ablation study on HARVEY. We highlight the best in **boldface** and mark the second-best in gray bold font.

Dataset	Naive	ABL	DBD	D-ST	CBD	HARVEY
CIFAR10	0.52	0.56	19.25	1.76	0.60	0.95
GTSRB	0.42	0.55	16.21	1.43	0.47	0.74
Tiny-ImageNet	3.19	3.48	114.13	10.59	3.60	5.68

Table 5: The time consumption (in hours) of all defenses across different datasets. Naive denotes the naive training.

5 Conclusion

Neural backdoors are difficult to remove once established in a machine-learning model. HARVEY counters such attacks early by splitting poisonous samples off the training dataset to prevent learning the backdoors in the first place. High recall guarantees complete backdoor removal, while high precision ensures that we learn a performant (clean) model on the remaining benign samples. We find that these challenges are best met by learning a model that overfits the backdoor and use it as an oracle for poisonous samples in iterative dataset splitting. This strategy represents a paradigm shift in splitting-based defenses, allowing for near-perfect removal of poisonous samples without harming the natural performance. Most importantly, HARVEY does not require any clean data as the reference. Moreover, it works across model architectures and datasets both on average and in the worst-case, setting a new standard in training-time defenses.

Acknowledgments

We gratefully acknowledge funding by the Helmholtz Association (HGF) within topic "46.23 Engineering Secure Systems", by SAP S.E. under project (DE-2020-021), and by the German Federal Ministry of Education and Research (BMBF) under the project DataChainSec (FKZ 16KIS1700).

References

Barni, M.; Kallas, K.; and Tondi, B. 2019. A New Backdoor Attack in CNNS by Training Set Corruption Without Label Poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*.

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. 2019. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Biggio, B.; and Roli, F. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84: 317–331.

Cai, R.; Zhang, Z.; Chen, T.; Chen, X.; and Wang, Z. 2022. Randomized Channel Shuffling: Minimal-Overhead Backdoor Attack Detection without Clean Datasets. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Carlini, N.; Jagielski, M.; Choquette-Choo, C. A.; Paleka, D.; Pearce, W.; Anderson, H.; Terzis, A.; Thomas, K.; and Tramèr, F. 2023. Poisoning Web-Scale Training Datasets is Practical. arXiv:2302.10149.

Chen, W.; Wu, B.; and Wang, H. 2022. Effective Backdoor Defense by Exploiting Sensitivity of Poisoned Samples. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *CoRR*, abs/1712.05526.

Doan, B. G.; Abbasnejad, E.; and Ranasinghe, D. C. 2020. Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems. In *Proc. of the Annual Computer Security Applications Conference (ACSAC)*.

Gao, K.; Bai, Y.; Gu, J.; Yang, Y.; and Xia, S.-T. 2023. Backdoor Defense via Adaptively Splitting Poisoned Dataset. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gao, Y.; Xu, C.; Wang, D.; Chen, S.; Ranasinghe, D. C.; and Nepal, S. 2019. STRIP: A Defence Against Trojan Attacks on Deep Neural Networks. In *Proc. of the Annual Computer Security Applications Conference (ACSAC)*.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *Proceeding of Machine Learning and Computer Security Workshop*.

Hayase, J.; Kong, W.; Somani, R.; and Oh, S. 2021. SPECTRE: defending against backdoor attacks using robust statistics. In *Proc. of the International Conference on Machine Learning (ICML)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Huang, K.; Li, Y.; Wu, B.; Qin, Z.; and Ren, K. 2022. Backdoor Defense via Decoupling the Training Process. In *Proc. of the International Conference on Learning Representations (ICLR)*.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2008. CIFAR (Canadian Institute for Advanced Research).

Kullback, S.; and Leibler, R. A. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1): 79–86.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*.

Li, B.; and Liu, W. 2024. A Theoretical Analysis of Backdoor Poisoning Attacks in Convolutional Neural Networks. In *Proc. of the International Conference on Machine Learning (ICML)*.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021a. Anti-Backdoor Learning: Training Clean Models on Poisoned Data. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021b. Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks. In *Proc. of the International Conference on Learning Representations (ICLR)*.

Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2018a. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. In Bailey, M.; Holz, T.; Stamatogiannakis, M.; and Ioannidis, S., eds., *Proc. of the International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*.

Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018b. Trojaning Attack on Neural Networks. In *Proc. of the Network and Distributed System Security Symposium (NDSS)*.

Nguyen, T. A.; and Tran, A. 2020. Input-Aware Dynamic Backdoor Attack. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 3454–3464.

Nguyen, T. A.; and Tran, A. T. 2021. WaNet - Imperceptible Warping-based Backdoor Attack. In *Proc. of the International Conference on Learning Representations (ICLR)*.

Qiu, H.; Zeng, Y.; Guo, S.; Zhang, T.; Qiu, M.; and Thuraisingham, B. 2021. DeepSweep: An Evaluation Framework for Mitigating DNN Backdoor Attacks Using Data Augmentation. In *Proc. of the ACM Asia Conference on Computer and Communications Security (ASIA CCS)*.

Shafahi, A.; Huang, W. R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; and Goldstein, T. 2018. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Stallkamp, J.; Schlipsing, M.; Salmen, J.; and Igel, C. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*.

Turner, A.; Tsipras, D.; and Madry, A. 2019. Label-Consistent Backdoor Attacks. arXiv:1912.02771.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019a. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *Proc. of the IEEE Symposium on Security and Privacy*.

Wang, R.; Zhang, G.; Liu, S.; Chen, P.-Y.; Xiong, J.; and Wang, M. 2020. Practical Detection of Trojan Neural Networks: Data-Limited and Data-Free Cases. In *Proc. of the European Conference on Computer Vision (ECCV)*.

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019b. Symmetric cross entropy for robust learning with noisy labels. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Wang, Z.; Mei, K.; Ding, H.; Zhai, J.; and Ma, S. 2022. Rethinking the Reverse-engineering of Trojan Triggers. In *Advances in Neural Information Processing Systems*.

Wang, Z.; Mei, K.; Zhai, J.; and Ma, S. 2023. UNICORN: A Unified Backdoor Trigger Inversion Framework. In *Proc. of the International Conference on Learning Representations (ICLR)*.

Wu, B.; Chen, H.; Zhang, M.; Zhu, Z.; Wei, S.; Yuan, D.; and Shen, C. 2022. BackdoorBench: A Comprehensive Benchmark of Backdoor Learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xu, X.; Wang, Q.; Li, H.; Borisov, N.; Gunter, C. A.; and Li, B. 2021. Detecting AI Trojans Using Meta Neural Analysis. In *Proc. of the IEEE Symposium on Security and Privacy*.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *Proc. of the British Machine Vision Conference (BMVC)*.

Zhang, Z.; Liu, Q.; Wang, Z.; Lu, Z.; and Hu, Q. 2023. Backdoor Defense via Deconfounded Representation Learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, P.; Chen, P.-Y.; Das, P.; Ramamurthy, K. N.; and Lin, X. 2020. Bridging Mode Connectivity in Loss Landscapes and Adversarial Robustness. In *Proc. of the International Conference on Learning Representations (ICLR)*.

Zhao, S.; Ma, X.; Zheng, X.; Bailey, J.; Chen, J.; and Jiang, Y. 2023. Clean-Label Backdoor Attacks on Video Recognition Models. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhou, J.; Lv, P.; Lan, Y.; Meng, G.; Chen, K.; and Ma, H. 2023. DataElixir: Purifying Poisoned Dataset to Mitigate Backdoor Attacks via Diffusion Models. In *Proc. of AAAI Conference on Artificial Intelligence (AAAI)*.