L-RED: EFFICIENT POST-TRAINING DETECTION OF IMPERCEPTIBLE BACKDOOR ATTACKS WITHOUT ACCESS TO THE TRAINING SET

Zhen Xiang, David J. Miller and George Kesidis

Anomalee Inc. and School of EECS, Pennsylvania State University

ABSTRACT

Backdoor attacks are an important, emerging form of adversarial attack typically against deep neural network image classifiers. The attacker aims to have the classifier learn to classify to a target class when test images from one or more source classes contain a backdoor pattern, while maintaining high accuracy on all clean test images. Reverse-Engineering-based Defenses (REDs) require no access to the training set but only access to an independent clean dataset for detection. Unfortunately, most existing REDs rely on an unrealistic assumption about the attack that all classes (except for the target class) are the source classes. REDs that do not rely on this assumption often require a significantly large set of clean images and heavy computation. In this paper, we propose a Lagrange-based RED against imperceptible backdoor attacks with *arbitrary* number of source classes. Our defense requires very few clean images to effectively detect backdoors and is computationally efficient. Notably, we detect 56/60 attacks with only two clean image per class in our main experiments.

Index Terms— backdoor, trojan, reverse engineering, annealing, deep neural network

1. INTRODUCTION

Recently, a backdoor attack has been proposed typically against deep neural network (DNN) image classifiers, where the attacker aims to: 1) induce test-time misclassification to a target class, whenever a clean test image from one (or several) source class(es) is embedded with an attacker-specified backdoor pattern; 2) correctly classify clean test images [1, 2, 3]. The attacker's goals can be achieved by poisoning the training set of the classifier with a relatively small number of "backdoor training images" – images originally from one of the source classes, but with the attacker-specified backdoor pattern embedded, and labeled to the target class. The attacker's poisoning capability is facilitated particularly when practical training requires seeking data from public sources (some of which could be attackers) [4].

Defense against backdoor attacks can be deployed during the classifier's training, where the defender/learner has access to the training set and aims to cleanse it [5, 6, 7, 8]. However, classifiers threatened by backdoor attacks are usually part of downstream applications (e.g. widespread mobile apps) or legacy systems, for which the training set is *not accessible* to the defender/user. In this "post-training" regime, the defender/user only has access to the trained classifier and a small clean dataset, and aims to detect whether the classifier has been backdoor attacked. Existing post-training defenses include SentiNet [9] and STRIP [10], which infer if an input test image contains a backdoor pattern. But in addition to the clean dataset for detection, both methods also require sample images containing the true backdoor pattern. Another defense trains a large number of backdoored and clean classifiers as available detection benchmarks [11, 12], which requires a large set of clean images and extensive computation.

A different group of post-training defenses, Reverse-Engineering-based Defenses (REDs), do not require access to the true backdoor pattern. Moreover, the number of clean images for detection are usually not sufficient to train even a shallow DNN. However, existing REDs either rely on an unrealistic assumption about the attack that the source classes include *all* class except the target class [13, 14, 9, 15], or requires a significant clean images and heavy computation as compensation to relieve such assumption [16, 17].

In this paper, we propose a Lagrange-based RED (L-RED) to detect *imperceptible* backdoor attacks with *arbitrary* number of source classes, and infer the target class for detected attack. Our defense can also be potentially extended with little modification to detecting perceptible backdoors. Compared with the state-of-the-art RED against imperceptible backdoors with arbitrary number of source classes ([16]), our defense requires much fewer clean images for detection and much lower computational complexity. Even with two clean image per class, our defense detects 56/60 attacks with 1/10 false detection in the main experiments on CIFAR-10.

2. BACKGROUND

2.1. Imperceptible Backdoor Attack

A backdoor attack is typically specified by a target class with label $t^* \in \mathcal{C}$, where $|\mathcal{C}| = K$, a set of source classes $\mathcal{S}^* \subset \mathcal{C}$, where $t^* \notin \mathcal{S}^*$, and a backdoor pattern. Effective backdoor patterns in the literature are either *human-imperceptible* or

Supported by an AFOSR DDDAS grant and Cisco URP gift.

human-perceptible. Here, we focus on the imperceptible case, where the backdoor pattern is embedded into a clean image $\mathbf{x} \in \mathcal{X}$ by

$$m(\mathbf{x}; \mathbf{v}^*) = [\mathbf{x} + \mathbf{v}^*]_c \tag{1}$$

with $\mathbf{v}^*$ an image-wide, human-imperceptible additive perturbation (with either small $||\mathbf{v}^*||_\infty$ or small $||\mathbf{v}^*||_0$ for imperceptibility); and $[\cdot]_c$ a clipping function [2, 5, 18, 7, 16].

A backdoor attack is typically launched by poisoning the classifier's *training set* with a small set of images originally from the source classes $\mathcal{S}^*$, embedded with the backdoor pattern $\mathbf{v}^*$, and labeled to the target class t^* [1, 2]. For a successful attack, the trained classifier $f(\cdot; \theta) : \mathcal{X} \rightarrow \mathcal{C}$ is supposed to: 1) classify to the target class t^* when any *test image* from any source class in $\mathcal{S}^*$ is embedded with $\mathbf{v}^*$; 2) maintain a high classification accuracy on clean images during testing.

2.2. Reverse-Engineering-Based Backdoor Defense (RED)

A typical RED consists of a backdoor pattern reverse-engineering/estimation step and an anomaly detection step [13, 14, 16]. When there is a successful backdoor attack, the learned backdoor mapping creates a "shortcut" between any source class $s \in \mathcal{S}^*$ and the target class t^*, which is the basis for anomaly detection. For example, the key ideas of [16] to detect imperceptible backdoors are: 1) for any "non-backdoor" class pair $(s, t) \in \mathcal{C} \times \mathcal{C}$, such that $t \neq t^*$, and a set $\mathcal{D}_s$ of clean images from class s, the *minimum* norm required for any pattern $\mathbf{v}$ to induce (via (1)) most images in $\mathcal{D}_s$ being (mis)classified to t is *large*, especially when $|\mathcal{D}_s|$ is large; 2) there exists a *small-norm* perturbation (i.e. the "shortcut" guaranteed by the existence of $\mathbf{v}^*$ with a small norm) that induces most of images in $\mathcal{D}_s$ being (mis)classified to t^* for any $s \in \mathcal{S}^*$, even if $|\mathcal{D}_s|$ is large. Thus, [16] searches for the minimum l_2 norm pattern inducing at least π-fraction of misclassification on $\mathcal{D}_s$ for each $(s, t) \in \mathcal{C} \times \mathcal{C}$ class pair:

$$\begin{aligned} &\underset{\mathbf{v}}{\text{minimize}} && ||\mathbf{v}||_2 \\ &\text{subject to} && \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \mathbb{1}(f(m(\mathbf{x}; \mathbf{v}); \theta) = t) \ \geq \ \pi, \end{aligned} \tag{2}$$

where $\mathbb{1}(\cdot)$ is an indicator function, and $\pi \in (0, 1)$ is often chosen to be large. Then attack and at least one "backdoor" class pair can be revealed by an anomaly detector because the pattern obtained for any "backdoor" class pair will have an *abnormally small* norm compared with the pattern obtained for "non-backdoor" class pairs. Note that REDs can also target on perceptible backdoors involving a pattern and an associated "mask" (see Appendix A.3), where the pattern estimation problem is similar to (2), though it aims to minimize the norm of the mask instead [13, 14]. REDs against imperceptible backdoors and perceptible backdoors can be deployed in parallel to provide a complete solution to both cases.

Limitations of existing REDs: Some REDs, e.g. [13, 14, 15], assume that *the source classes consist of all classes except the target class*, i.e. $\mathcal{S}^* \cup t^* = \mathcal{C}$. Correspondingly, pattern estimation for these methods is performed for each putative target class t on the union $\cup_{s \neq t} \mathcal{D}_s$ (instead of for each class pair (s, t) on $\mathcal{D}_s$). However, $\mathcal{S}^* \cup t^* = \mathcal{C}$ is not a realistic assumption in practice because an attacker may not be able to collect images from that many classes to create backdoor training images (for poisoning the training set), especially when the number of classes K is huge. In the case $|\mathcal{S}^*| \ll K$, REDs designed for $\mathcal{S}^* \cup t^* = \mathcal{C}$ will easily fail since for most of classes, the "shortcut" to t^* does not exists. Although REDs like [16] and [19] perform pattern estimation for each class pair to detect backdoor attacks with arbitrary number of source classes, they require a significant number of clean images per class (to ensure that the norm of the estimated pattern for "non-backdoor" class pairs are sufficiently large), and therefore, a high computational complexity.

3. METHOD

In this section, we described our efficient Lagrange-based RED (L-RED) against imperceptible backdoor attacks with arbitrary number of source classes, without access to the training set. Like existing REDs, our defense contains a pattern estimation step *followed by* an anomaly detection step.

3.1. Pattern Estimation

For efficiency purpose, we perform pattern estimation for each putative target class on $\cup_{s \neq t} \mathcal{D}_s$ – fewer clean images per class are needed since we only need $| \cup_{s \neq t} \mathcal{D}_s|$ to be large to ensure the estimated pattern for "non-backdoor" target class to have a large norm. To detect backdoor attacks with arbitrary number of source classes, pattern estimation for the true target class should "focus" only on the clean images from the source classes. These motivate us to propose a "weighted" pattern estimation problem for each putative $t \in \mathcal{C}$:

$$\begin{aligned} &\underset{\mathbf{v}, \mathbf{w}}{\text{minimize}} && ||\mathbf{v}||_2 \\ &\text{subject to} && Q_t(\mathbf{v}, \mathbf{w}) \triangleq \sum_{s \neq t} w_s q_{st}(\mathbf{v}) \ \geq \ \pi, \\ &&& \sum_{s \neq t} w_s = 1, \quad \mathbf{w} \geq \mathbf{0}, \end{aligned} \tag{3}$$

where $q_{st}(\mathbf{v}) \triangleq \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \mathbb{1}(f(m(\mathbf{x}; \mathbf{v}); \theta) = t)$ is the mislcassification fraction from s to t given pattern $\mathbf{v}$.

Since problem (3) is not directly solvable, we propose an iterative algorithm to sequentially update $\mathbf{v}$ and $\mathbf{w}$ (with initial $\mathbf{v} = \mathbf{0}$ and initial $\mathbf{w}$ uniform in all entries except the t-th) until $Q_t(\mathbf{v}, \mathbf{w})$ grows from (initially) close to zero to more than π. To update $\mathbf{v}$ in iteration $(\tau + 1)$ given $\mathbf{v}^{(\tau)}$ and $\mathbf{w}^{(\tau)}$ from iteration τ, we obtain a *small* increment in $\mathbf{v}$ as:

$$\mathbf{z}^* = \underset{||\mathbf{z}||_2 = \delta}{\text{argmax}} \, Q_t(\mathbf{v}^{(\tau)} + \mathbf{z}, \mathbf{w}^{(\tau)}) - Q_t(\mathbf{v}^{(\tau)}, \mathbf{w}^{(\tau)}) \tag{4}$$

where δ is chosen to be small such that the resulting $||\mathbf{v}||_2$ is close to the minimum required for $Q_t(\mathbf{v}, \mathbf{w}) \geq \pi$. Since $Q_t(\mathbf{v}, \mathbf{w})$ is not differentiable in $\mathbf{v}$ due to the existence of the indicator function, we find an approximated solution to (4) as:

$$\mathbf{z}^* \approx \delta \nabla_{\mathbf{v}} J_t(\mathbf{v}^{(\tau)}, \mathbf{w}^{(\tau)}) / ||\nabla_{\mathbf{v}} J_t(\mathbf{v}^{(\tau)}, \mathbf{w}^{(\tau)})||_2 \quad (5)$$

where $J_t(\mathbf{v}, \mathbf{w}) \triangleq \sum_{s \neq t} w_s \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \log p(t|m(\mathbf{x}; \mathbf{v}); \theta)$ is a differentiable surrogate to $Q_t(\mathbf{v}, \mathbf{w})$, with $p(c|\mathbf{x}; \theta)$ the classifier's posterior for class $c \in \mathcal{C}$ for any image $\mathbf{x} \in \mathcal{X}$.

Our updating of $\mathbf{w}$ in iteration $(\tau + 1)$ aims to find the possible source classes and give them more weights. Naively, this can be done by hard assigning $w_c = 1$ to $c = \arg\max_{s \neq t} q_{st}(\mathbf{v}^{(\tau+1)})$. This is because that misclassification from source classes to the true backdoor target class using a small-normed $\mathbf{v}$ tends to be easier than for non-source classes due to the existence of the "shortcut". However, once all weights are hard-assigned to a class, updating $\mathbf{v}$ will solely be performed on clean images associated with this single class in all proceeding iterations. If in the very initial iterations, weights are incorrectly assigned to a non-source class (which is very likely in practice due to the noisy circumstance), a poor local optima with a large $||\mathbf{v}||_2$ will be obtained. Thus, we propose the following Lagrangian for updating $\mathbf{w}$ in iteration $(\tau + 1)$ subject to a specified level of randomness:

$$\begin{aligned} \underset{\mathbf{w}}{\text{maximize}} \quad & Q_t(\mathbf{v}^{(\tau+1)}, \mathbf{w}) - T^{(\tau+1)} \sum_{s \neq t} w_s \log w_s \\ \text{subject to} \quad & \sum_{s \neq t} w_s = 1, \end{aligned} \quad (6)$$

where T can be viewed as "temperature", which depends on the state specified by $Q_t(\mathbf{v}, \mathbf{w})$ of the current iteration. Note that if $T^{(\tau+1)} = 0$, solving (6) is equivalent to hard weight assignment; if $T^{(\tau+1)}$ is large, solving (6) is equivalent to entropy maximization given $w_t = 0$. To avoiding being trapped to a poor local optima for solving (3), we expect the temperature T to be large when $Q_t(\mathbf{v}, \mathbf{w})$ is initially close to zero, so as to preserve randomness in $\mathbf{w}$ while updating it by solving (6). As $Q_t(\mathbf{v}, \mathbf{w})$ grows large, we expect T to *gradually* decrease, such that more weights are *gradually* assigned to classes with large $q_{\cdot t}(\mathbf{v})$ – these classes are likely the true source classes if t is the true target class of an attack. Hence we adopt a smooth temperature scheduling function:

$$T^{(\tau+1)} = -\log Q_t(\mathbf{v}^{(\tau+1)}, \mathbf{w}^{(\tau)}) \quad (7)$$

for $Q_t(\mathbf{v}^{(\tau+1)}, \mathbf{w}^{(\tau)}) > 0$ ($\mathbf{w}$ is set to uniform otherwise), based on which the close form solution to (6) is:

$$w_s^{(\tau+1)} = \frac{\exp\left[q_{st}(\mathbf{v}^{(\tau+1)})/T^{(\tau+1)}\right]}{\sum_{c \neq t} \exp\left[q_{ct}(\mathbf{v}^{(\tau+1)})/T^{(\tau+1)}\right]}, \ \forall s \neq t \quad (8)$$

with the derivation shown in Appendix A.1. The above described pattern estimation algorithm is summarized in Appendix A.2.

3.2. Anomaly Detection

Our anomaly detection uses the similar hypothesis test idea as the approach in [16], but is much simpler in form. We first obtain a detection statistic for each putative target class t as $r_t = (||\mathbf{v}_t||_2)^{-1}$, where $\mathbf{v}_t$ is the estimated pattern for class t. To test the null hypothesis that there is no attack, we fit a null distribution $G(\cdot)$ using all statistics except $r_{\max} = \max_t r_t$ (corresponding to the class with the smallest estimated perturbation size). We choose $G(\cdot)$ in form of a Gamma distribution, while other single tailed distribution form should also work [19]. Then we calculate the order statistic p-value for $r_{\max}$ under $G(\cdot)$ as $\text{pv} = 1 - G(r_{\max})^K$. A detection threshold θ is chosen, such that the null hypothesis is rejected (i.e. an attack is detected) with confidence $(1-\theta)$ if $\text{pv} < \theta$. When an attack is detected, $\hat{t} = \arg\max_t r_t$ is inferred as the target class.

4. EXPERIMENTS

We mainly compare the *effectiveness* and *efficiency* of our defense with existing REDs against attacks on CIFAR-10 (with 50k training images and 10k test images, both evenly distributed in 10 classes). Experiments on other datasets are also shown.

4.1. Attacker: Creating Backdoor Training Images

We consider two imperceptible patterns, a "global" pattern with max perturbation size $2/255$ and a "local" square pattern which perturbs only four pixels of an image. For *each* pattern, using the standard backdoor training image crafting approach in Section 2.1, we create three groups of attacks (10 attack per group) with *one* source class, *three* source classes, and *nine* source classes, respectively. The backdoor pattern generation, the choices of source classes and target class, and the number of backdoor training images are detailed in Appendix A.4.

4.2. Trainer: Training on Poisoned Training Set

Here, we use the ResNet-18 [21] DNN architecture. Experiments involving other DNN structures and backdoor patterns are shown in Appendix A.7. For each attack, one classifier is trained on the poisoned training set. Training is performed for 100 epochs with batch size 32 and learning rate 10^{-3}. Data augmentation including random cropping and random horizontal flipping are used during training. For convenience, we name the six groups of classifiers being attacked as "G-1", "G-3", "G-9", "L-1", "L-3", and "L-9", respectively, where "G" represents "global", "L" represents "local", and the numbers represent the number of source classes. We also train 10 classifiers on the clean CIFAR-10 training set as the control group (named "C-0") for false detection rate evaluation.

From the experimenter's perspective, we show the effectiveness of the attacks through attack success rate (ASR) and

Table 1: Attack success rate (ASR) and clean test accuracy (CTA) for the six group of classifiers being attacked.

	G-1	G-3	G-9	L-1	L-3	L-9
ASR	92.1 ± 1.9	94.1 ± 1.1	96.6 ± 0.3	93.1 ± 1.6	91.1 ± 0.8	91.9 ± 0.7
CTA	93.6 ± 0.2	93.5 ± 0.2	93.5 ± 0.2	93.7 ± 0.1	93.6 ± 0.2	93.6 ± 0.2

clean test accuracy (CTA), using the test set of CIFAR-10. ASR of an attack is defined as the fraction of test images from the source class(es) being classified to the target class when the backdoor pattern is embedded. In Table 1, for all attacks, the ASRs are uniformly high, with almost no degradation in CTA compared with group C-0 (with CTA 93.8 ± 0.1) – the attacks are all successful.

4.3. Defender: Detecting if Classifier is Attacked

We apply three defenses to the six groups of classifiers being attacked and the clean classifiers in C-0: (1) **RED-P**: the method in [16] that performs pattern estimation for each class pair; (2a) **RED-A**: our proposed annealing based RED; (2b) **RED-A'**: RED-A with a data-limited setting; (3) **RED-U**: RED with the $\mathcal{S}^* \cup t^* = \mathcal{C}$ assumption. Note that RED-U is actually a special case of RED-A with $\mathbf{w}$ fixed to $\mathbf{w}^{(0)}$ during pattern estimation. For all these defenses, we choose $\pi = 0.9$, pattern estimation step size $\delta = 10^{-4}$, and detection (confidence) threshold $\theta = 0.05$, while in general, these choices are not critical to the performance of RED against imperceptible backdoors [19]. For RED-P, RED-A, and RED-U, we use 8 clean images per class for detection; for RED-A', we use only 2 clean images per class for detection. For each classifier to be inspected, the clean images for detection are randomly sampled from the test set of CIFAR-10, which are independent from the training (and attack crafting) process.

Accuracy Evaluation: For all the defenses to be evaluated, a successful detection of an attack requires also a correct inference of the target class. For the clean classifiers in C-0, "no attack detected" is deemed a successful detection. In Table 2, we show the fraction of successful detection for all the defenses for the seven groups of classifiers. For RED-U, although perfect detection is achieved for G-9 and L-9 where the $\mathcal{S}^* \cup t^* = \mathcal{C}$ assumption is satisfied, with no false detection on C-0, the detection accuracy on G-1, G-3, L-1, and L-3 (where $|\mathcal{S}^*| < |\mathcal{C}|$) is poor. RED-P shows limited detection capability with 8 clean images per class for attacks with the "global" pattern, which is consistent with Figure 13 in [19], but it fails for most of the attacks with the "local" pattern. In comparison, our RED-A achieves perfect detection of all the attacks, *regardless of the number of source classes*, and with no false detection. Notably, with only two clean images per class, RED-A' detects 56/60 attacks with only 1/10 false detection, which is generally better result than both RED-P and RED-U. More insights on the effectiveness of our defense and illustration of backdoor patterns estimated by RED-A are in Appendix A.5 and Appendix A.6, respectively.

Table 2: Detection accuracy (fraction of successful detection) of the defenses on the seven groups of classifiers.

	G-1	G-3	G-9	L-1	L-3	L-9	C-0
RED-P	5/10	9/10	7/10	2/10	3/10	3/10	7/10
RED-U	2/10	8/10	**10/10**	2/10	5/10	**10/10**	**10/10**
RED-A	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**
RED-A'	9/10	**10/10**	**10/10**	9/10	8/10	**10/10**	9/10

Table 3: Average time consumption (in second) of the defenses on the seven groups of classifiers.

	G-1	G-3	G-9	L-1	L-3	L-9	C-0
RED-P	480	473	531	478	479	394	541
RED-A	562	615	636	529	482	536	535
RED-A'	136	161	167	129	169	163	157

Efficiency Evaluation: In fact, with N clean images per class for detection, defenses like RED-P perform $K(K-1)$ pattern estimations, each on N clean images, while defenses like RED-A (or RED-U) perform K pattern estimations, each on NK clean images – the two methods have the same order of complexity. However, as we have pointed out in Section 3 and demonstrated experimentally, to achieve high detection accuracy, defenses like RED-P need a sufficiently large N, while defenses like RED-A only need NK to be large. As shown in Table 3, the average time consumption of RED-A and RED-P are similar for each of the seven groups of classifiers (though RED-P is only slightly faster). RED-A', however, exhibits a much lower time consumption than RED-A and RED-P on average, since its pattern estimation is performed on only two clean images per class. The execution time for RED-U is similar as for RED-A, hence is not shown here due to page limitations. All the above experiments are performed on a RTX2080-Ti (11GB) GPU.

4.4. Experiments on Other Datasets

We create one attack on each of the MNIST, FMNIST, GTSRB, and CIFAR-100 datasets. For each dataset, one classifier is trained on the backdoor poisoned dataset and one clean classifier is trained without attack. Details of the attacks and training are shown in Appendix A.8. We apply our defense with the same settings as RED-A to inspect these classifiers, except that for classifiers trained on GTSRB and CIFAR-100, we use *only one* clean image per class for detection. In Table 4, we show the order statistic p-values obtained for each classifier by our defense. All attacks are detected with no false detection of any clean classifiers using threshold $\theta = 0.05$. Moreover, our detector requires less than 3 hours to detect an attack while the method in [16] requires several days to do so.

Table 4: Order statistic p-values for both clean and attacked classifiers on MNIST, FMNIST, GTSRB, and CIFAR-100, when applying our defense ("u.f." for "underflow").

	MNIST	FMNIST	GTSRB	CIFAR-100
Attacked	u.f.	8.44×10^{-7}	3.41×10^{-9}	u.f.
Clean	0.300	0.331	0.152	0.551

5. CONCLUSION

We proposed an Lagrange-based RED against imperceptible backdoors with arbitrary number of source classes. Our defense requires very few clean images per class for detection and is timely efficient.

6. REFERENCES

[1] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[2] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," https://arxiv.org/abs/1712.05526v1, 2017.

[3] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, and J Zhai, "Trojaning attack on neural networks," in *Proc. NDSS*, San Diego, CA, Feb. 2018.

[4] D.J. Miller, Z. Xiang, and G. Kesidis, "Adversarial learning in statistical classification: A comprehensive review of defenses against attacks," *Proceedings of the IEEE*, vol. 108, pp. 402–433, March 2020.

[5] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Proc. NIPS*, 2018.

[6] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," http://arxiv.org/abs/1811.03728, Nov 2018.

[7] Z. Xiang, D.J. Miller, and G. Kesidis, "A benchmark study of backdoor data poisoning defenses for deep neural network classifiers and a novel defense," in *Proc. IEEE MLSP*, Pittsburgh, 2019.

[8] M. Du, R. Jia, and D. Song, "Robust anomaly detection and backdoor attack detection via differential privacy," in *Proc. ICLR*, 2020.

[9] E. Chou, F. Tramèr, G. Pellegrino, and D. Boneh, "Sentinet: Detecting physical attacks against deep learning systems," 2018.

[10] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," in *Proc. ACSAC*, 2019.

[11] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI trojans using meta neural analysis," https://arxiv.org/abs/1910.03137, 2019.

[12] "Iarpa trojai: Trojans in artificial intelligence," https://www.iarpa.gov/index.php/research-programs/trojai/trojai-baa, 2019.

[13] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B.Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. IEEE Symposium on Security and Privacy*, 2019.

[14] W. Guo, L. Wang, X. Xing, M. Du, and D. Song, "TABOR: A highly accurate approach to inspecting and restoring Trojan backdoors in AI systems," https://arxiv.org/abs/1908.01763, 2019.

[15] R. Wang, G. Zhang, S. Liu, P.-Y. Chen, J. Xiong, and M. Wang, "Practical detection of trojan neural networks: Data-limited and data-free cases," in *Proc. ECCV*, 2020.

[16] Z. Xiang, D. J. Miller, and G. Kesidis, "Revealing backdoors, post-training, in dnn classifiers via novel inference on optimized perturbations inducing group misclassification," in *Proc. IEEE ICASSP*, 2020, pp. 3827–3831.

[17] Z. Xiang, D.J. Miller, and G. Kesidis, "Revealing Perceptible Backdoors, without the Training Set, via the Maximum Achievable Misclassification Fraction Statistic," in *Proc. IEEE MLSP*, Espoo, Finland, 2020.

[18] H. Zhong, H. Zhong, A. Squicciarini, S. Zhu, and D.J. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," in *Proc. CODASPY*, March 2020.

[19] Z. Xiang, D. J. Miller, G. Kesidis, "Detection of backdoors in trained classifiers without access to the training set," https://arxiv.org/pdf/1908.10498, 2020.

[20] S.-M. M.-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," in *Proc. CVPR*, 2016.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016.

[22] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," in *Proc. CVPR*, 2018.

[23] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, 2017.

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[26] J. Salmen J. Stallkamp, M. Schlipsing and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition.," *Neural Networks*, vol. 32, pp. 323–332, 2012.

[27] Alex K., "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

A. APPENDIX

A.1. Derivation of the Updating Rule for the Weight Vector

Here we neglect the iteration indices for simplicity. With $\mathbf{v}$ and $T > 0$ fixed, solving (6) over $\mathbf{w}$ yields a close form solution, which is given as (8). Here, we show the details.

We first cast (6) into the following Lagrangian:

$$L(\mathbf{w}, \nu) = \sum_{s \neq t} w_s q_{st}(\mathbf{v}) - T \sum_{s \neq t} w_s \log w_s + \nu (\sum_{s \neq t} w_s - 1). \tag{9}$$

For all $s \neq t$, we obtain the following partial derivative:

$$\frac{\partial L}{\partial w_s} = q_{st}(\mathbf{v}) - T(\log w_s + 1) + \nu. \tag{10}$$

By setting the above partial derivatives to 0, we obtain, for all $s \neq t$:

$$w_s = \exp(\frac{q_{st}(\mathbf{v}) + \nu - T}{T}) \tag{11}$$

By plug in the above into the constraint $\sum_{s \neq t} w_s = 1$, we get the close form solution (8).

A.2. L-RED Pattern Estimation Algorithm

Algorithm 1 L-RED pattern estimation.

```
Inputs: putative target class $t$, clean images $\cup_{s \neq t} \mathcal{D}_s$, classifier $f(\cdot;\theta)$, step size $\delta$, target misclassification fraction $\pi$, maximum number of iterations $\tau_{\max}$
Initialization: $\mathbf{v}^{(0)} = \mathbf{0}$, $w_s^{(0)} = 1/(K-1)$ for $\forall s \neq t$, $w_t^{(0)} = 0$, $\tau = 0$, estimated pattern $\mathbf{v}_t = \mathbf{0}$
while $\tau < \tau_{\max}$ do
    $\mathbf{v}_t = \mathbf{v}^{(\tau+1)} = \mathbf{v}^{(\tau)} + \delta \frac{\nabla_{\mathbf{v}} J_t(\mathbf{v}^{(\tau)}, \mathbf{w}^{(\tau)})}{||\nabla_{\mathbf{v}} J_t(\mathbf{v}^{(\tau)}, \mathbf{w}^{(\tau)})||_2}$
    if $Q_t(\mathbf{v}^{(\tau+1)}, \mathbf{w}^{(\tau)}) \geq \pi$ then
        break
    else if $Q_t(\mathbf{v}^{(\tau+1)}, \mathbf{w}^{(\tau)}) > 0$ then
        $T^{(\tau+1)} = -\log Q_t(\mathbf{v}^{(\tau+1)}, \mathbf{w}^{(\tau)})$
        for all $s \neq t$ do
            $w_s^{(\tau+1)} = \frac{\exp[q_{st}(\mathbf{v}^{(\tau+1)})/T^{(\tau+1)}]}{\sum_{c \neq t} \exp[q_{ct}(\mathbf{v}^{(\tau+1)})/T^{(\tau+1)}]}$
    else
        $\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(0)}$
    $\tau$++
Outputs: $\mathbf{v}_t$
```

A.3. Potential Extension to Perceptible Cases

In the main paper, we focus on detecting imperceptible backdoors with embedding mechanism Eq. 1 (which is the only effective existing way of imperceptible backdoor embedding to our knowledge). However, backdoor patterns can also be perceptible but seemingly innocuous to the scene, e.g. a pair of glasses on a face [2]. Perceptible backdoor patterns are embedded in a clean image $\mathbf{x} \in \mathcal{X}$ by:

$$m_{\mathrm{p}}(\mathbf{x}; \{\mathbf{u}^*, \mathbf{m}^*\}) = \mathbf{x} \odot (\mathbf{1} - \mathbf{m}^*) + \mathbf{u}^* \odot \mathbf{m}^* \tag{12}$$

where $\mathbf{u}^*$ is an image-wide pattern (usually with small $||\mathbf{u}^*||_0$ to represent the backdoor "object", e.g. the pair of glasses); $\mathbf{m}^*$ is an image-wide mask with $m_{i,j} \in \{0, 1\}$[1]; $\odot$ represents element-wise multiplication, which applies $\mathbf{m}^*$ to all colors/channels of $\mathbf{x}$ ([1, 13]). Note that $\mathbf{m}^*$ is usually associated with the pattern $\mathbf{u}^*$, such that $m_{i,j} = 1$ only if $u_{i,j,k} > 0$ for any channel k, hence both $||\mathbf{m}^*||_0$ and $||\mathbf{m}^*||_1$ are typically small [13].

REDs detecting perceptible backdoors rely on a similar idea with REDs for the imperceptible case, but the "shortcut" between the backdoor source classes and target class is a small-sized mask instead of a small-sized perturbation. For example, [13] which relies on the assumption $\mathcal{S}^* \cup t^* = \mathcal{C}$ solves the following problem for each putative target class t:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{m}}{\text{minimize}} \quad & ||\mathbf{m}||_1 \\ \text{subject to} \quad & \frac{1}{K-1} \sum_{s \neq t} \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \mathbb{1}(f(m_{\mathrm{p}}(\mathbf{x}; \{\mathbf{u}, \mathbf{m}\}); \theta) = t) \\ & \geq \ \pi, \end{aligned} \tag{13}$$

where $K = |\mathcal{C}|$, and hopes to find an abnormally small mask for the backdoor target class when there is an attack. To detect perceptible backdoors with arbitrary number of source classes *without considering the efficiency*, we can easily formulate a (pattern, mask) estimation problem for each class pair $(s, t) \in \mathcal{C} \times \mathcal{C}$ based on (13):

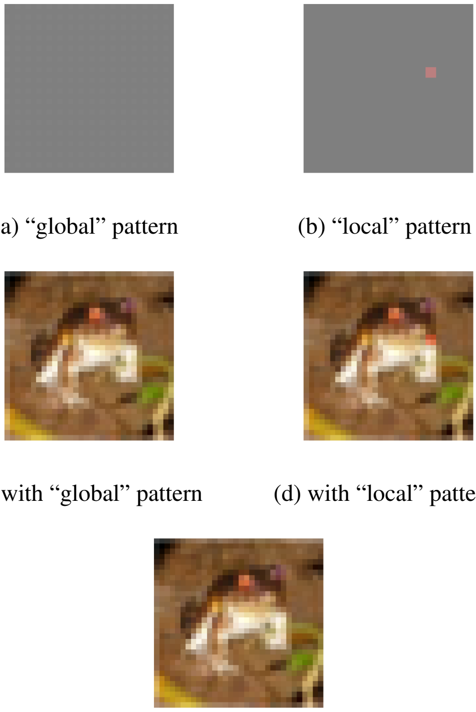
$$\begin{aligned} \underset{\mathbf{u},\mathbf{m}}{\text{minimize}} \quad & ||\mathbf{m}||_1 \\ \text{subject to} \quad & \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \mathbb{1}(f(m_P(\mathbf{x}; \{\mathbf{u}, \mathbf{m}\}); \theta) = t) \ \geq \ \pi. \end{aligned} \tag{14}$$

Note that (14) has a very similar form with (2). We believe that our efficient, annealing-based RED which is originated from solving (2) can be extended for the perceptible case by considering the perceptible backdoor embedding mechanism used in (14). This is a promising future research direction based on the current work.

A.4. Attack Crafting Details

In our experiments, pixel values are always normalized to $[0, 1]$. We also consider practical case by using 8-bit finite precision for pixel values, such that a valid pixel value will be in the set $\{0, 1/255, \cdots, 1\}$.

[1] A blended embedding mechanism is mentioned by [2] with $m_{i,j} \in [0, 1]$.



(a) “global” pattern

(b) “local” pattern

(c) with “global” pattern

(d) with “local” pattern

(e) clean

Fig. 1: Example backdoor patterns (with 127/255 offset for better visualization), example backdoor training images embedded with each backdoor pattern, and originally clean image.

A.4.1. Backdoor Patterns

In our experiments on CIFAR-10 in the main paper, we considered a “global” pattern and a “local” pattern. The global pattern was first considered in [18] (see their Figure 5). For a 2 × 2 window applied to any spatial location in an image, there is one and only one pixel being perturbed in all color channels. Here, we set the perturbation size to be $2/255$ for pixels and channels being perturbed. The local square pattern is similar to the local patterns used in [5] and [15]. For each attack, we randomly select a 2×2 square and a color channel, and perturb the four pixels in the selected channel by $50/255$. Example backdoor patterns and backdoor training images are shown in Figure 1. Note that the embedded backdoor pattern are barely noticeable to humans.

A.4.2. Source Classes and Target Class

For simplicity, we enumerate the 10 classes of CIFAR-10, i.e., ‘plane’, ‘car’, ‘bird’, ‘cat’, ‘deer’, ‘dog’, ‘frog’, ‘horse’, ‘ship’, and ‘truck’, as class ‘1-10’. The target classes for attacks using the “global” pattern and the “local” pattern are class 10 and class 2, respectively. For the “global” pattern, the source classes for attacks with one source class and attacks with three source classes are class 1 and classes $\{1, 7, 9\}$, respectively. For the “local” pattern, the source classes for attacks with one source class and attacks with three source classes are class 1 and classes $\{1, 5, 6\}$, respectively. For both backdoor patterns, attacks with nine source classes use all the classes except for the target class as source classes, i.e., following the $\mathcal{S}^* \cup t^* = \mathcal{C}$ assumption. All the above choices of source classes and target class are arbitrary.

A.4.3. Number of Backdoor Training Images

In general, the “global” pattern is “easier” to be learned than the “local” pattern. Hence, fewer backdoor training images are required if the “global” pattern is used to launch a backdoor attack. For the “global” pattern, for attacks with one source class, three source classes, and nine source classes, 150, 50, and 15 back door training images per source class are created, respectively. For the “local” pattern, for attacks with one source class, three source classes, and nine source classes, 900, 300, and 100 back door training images per source class are created, respectively. Note that in practice, an attacker can either poison the training set with more backdoor training images, or increase the perturbation size (or the number of pixels to be perturbed for any “local” patterns) to achieve a better attack effectiveness.

A.4.4. Emulation of Practical Backdoor Poisoning

Recall from Section 2.1 that backdoor training images are created using clean images from the source classes, embedded with the backdoor pattern, and labeled to the target class. In our experiments on CIFAR-10, the originally clean images used to generate the backdoor training images are from the training set of CIFAR-10. To emulate the practical case where those clean images are collected by the attacker but not possessed by the trainer, we remove those clean images used for generating backdoor training images from the CIFAR-10 training set. Then the *poisoned* CIFAR-10 training set still contains 50k training images.

A.5. Insights behind the Accuracy Evaluation Results

In the main paper, we showed that our RED-A and RED-A’ outperform existing REDs on detecting imperceptible backdoor attacks. In particular, RED-A achieves perfect detection results with no false detection. The main reasons behind the success of our defense are explained in Section 3 as the foundation of Algorithm 1: we employ an annealing process by introducing an entropy term multiplied by a gradually decreasing temperature parameter, such that more weights are gradually assigned to classes that are more likely the source classes. Here, we visualize such weight assignment by first introducing the concept of “collateral damage” which is first observed by [19].

Definition A.1. A classifier undergoing a backdoor attack with source classes $\mathcal{S}^*$ and target class t^* ($\mathcal{S}^* \cup t^* \neq \mathcal{C}$) is said to suffer from a **collateral damage** if for any class $\tilde{s} \notin \mathcal{S}^* \cup t^*$, the test-time group misclassification from $\tilde{s}$ to t^* induced by the same backdoor pattern is significantly higher than the class confusion from $\tilde{s}$ to t^* for clean test images. Moreover, the higher the group misclassification from $\tilde{s}$ to t^*, the more severe the collateral damage is for class $\tilde{s}$.

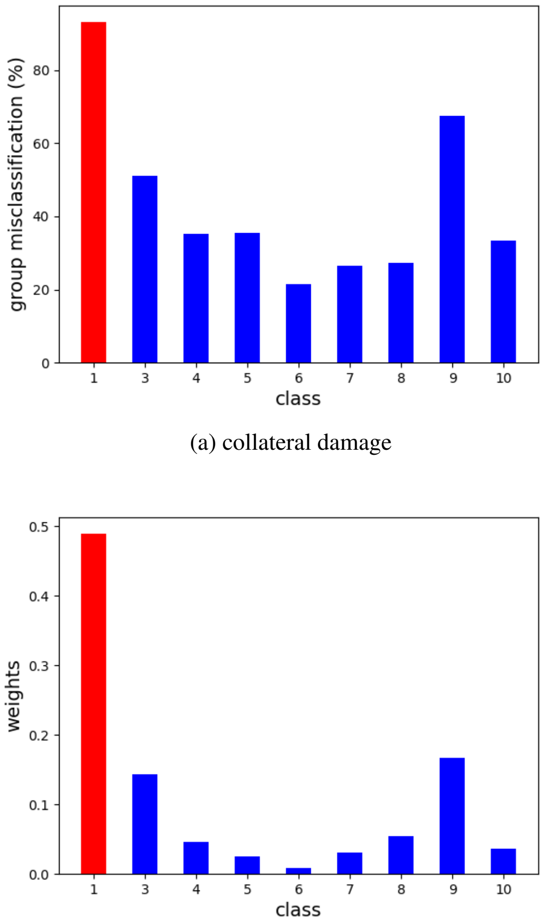
The possible reason for the collateral damage is that the backdoor pattern may be learned independently from the major objects/features of the source classes, such that a test image not from any source classes may be classified to the target class when the backdoor pattern exists.

Collateral damage phenomenon can also be revealed through the pattern estimation step of REDs. As shown by the experiments in [19], for defenses like RED-P which perform pattern estimation for all class pairs, an abnormally small perturbation can be found for some class pair (s, t), where t is the attack target class and s suffers from severe collateral damage.

For our defense, if there is an attack, for the true backdoor target class, we expect that when the stopping condition of Algorithm 1 is met, weights will be assigned mainly to the backdoor source classes and classes suffering severe collateral damage (if there are any). Here, we focus on the ten classifiers from the L-1 group, where the source class and the target class are class 1 and class 2, respectively. For each of the ten classifiers, for each class except the target class, we embed the same backdoor pattern used by the attacker into the clean test images (that are not involved in classifier's training) labeled to this class to evaluate the group misclassification fraction to the target class. The average group misclassification fraction for each non-target class over the ten classifiers is shown in Figure 2a. For each of the ten classifiers, we also consider the weight assignment at the end of pattern estimation for the true backdoor target class when RED-A is applied. The weight for each class is also averaged over the ten classifiers and is shown in Figure 2b. As we have expected, on average, the true source class, class 1, is assigned a much larger weight than any other classes. Class 3 and class 9 are also assigned some weights since they suffer more sever collateral damage than other non-source classes.

A.6. Estimated Backdoor Patterns

In the main paper, we show that RED-A successfully detects all the attacks with no false detection. In Figure 3, we show example backdoor patterns estimated by RED-A. In particular, we focus on the two attacks whose backdoor patterns are shown in Figure 1. The estimated patterns clearly contain the main features of the true backdoor patterns being used.



(a) collateral damage

(b) weights

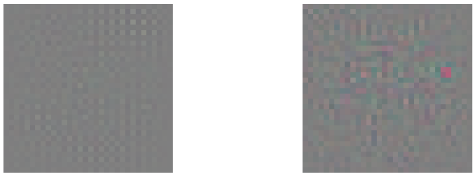
Fig. 2: Group misclassification fraction and weight assignment (when using RED-A) for all non-target classes, averaged over the ten classifiers from the L-1 group.

A.7. Experiments Involving Other DNN Architectures and Backdoor Patterns

In the main paper, we compared our defense with existing REDs on six groups of classifiers being attacked and a group of clean classifiers not attacked. The attacks involve two different imperceptible backdoor patterns, and the architecture of the DNN is ResNet-18 [21]. In this section, we consider attacks on CIFAR-10 using other imperceptible backdoor patterns and victim classifiers with other DNN architectures.

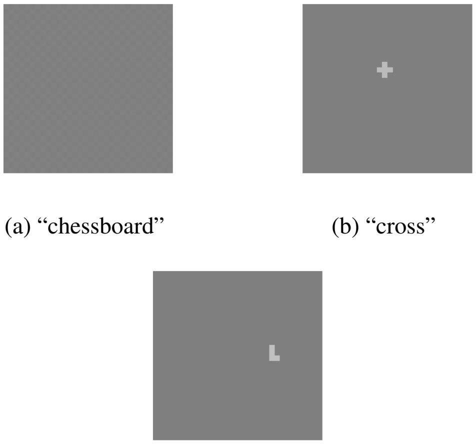
A.7.1. Attacks Crafting

We create three groups of attacks with one source class, denoted by group A, B, and C, with ten attacks in each group. Example backdoor patterns and backdoor training images are shown in Figure 4.



(a) estimated “global” pattern (b) estimated “local” pattern

Fig. 3: Example backdoor patterns (with 127/255 offset to elucidate negative perturbations) estimated by RED-A for the two attacks whose true backdoor patterns being used are shown in Figure 1.



(a) “chessboard” (b) “cross”

(c) “l-shape”

Fig. 4: Example backdoor patterns (with 127/255 offset for better visualization) for attack groups A, B, and C.

Attacks in group A use the “chessboard” pattern used by [16]. Here, we choose a universal perturbation size 2/255, i.e. one and only one pixel among any two adjacent pixels is perturbed (in all three channels) by 2/255. The source and target classes are class 1 and class 10, respectively. 150 backdoor training images are created to poison the training set.

Attacks in group B use a “cross” pattern which is similar to one of the imperceptible backdoor patterns considered by [5]. The spatial location of the cross is randomly chosen for each attack. The perturbation size is 60/255 in all three channels for the five pixels being perturbed. The source and target classes are class 3 and class 4, respectively; and 450 backdoor training images are created to poison the training set.

Attacks in group B uses a “l-shape” pattern which is similar to one of the imperceptible backdoor patterns considered by [5]. Again, the spatial location of the cross is randomly chosen for each attack. The perturbation size is 65/255, and the number of backdoor training images is 900. The source and target classes are class 4 and class 6, respectively.

Table 5: Attack success rate (ASR) and clean test accuracy (CTA) for the classifiers being attacked in group A, B, and C.

	A	B	C
ASR	94.7 ± 1.4	98.6 ± 0.5	94.5 ± 0.6
CTA	93.6 ± 0.1	92.7 ± 0.2	93.8 ± 0.3

A.7.2. Training

For each attack, a classifier is trained on the poisoned training set. For group A, we use the same training configuration and the same ResNet-18 DNN architectures as in experiments on CIFAR-10 in the main paper. For group B, we consider a more recent MobileNetV2[2] [22] architecture. Training is performed for 100 epochs with batch size 32 and learning rate 10^{-3}. The same training data augmentation options are used. The test accuracy on CIFAR-10 in absence of attack is around 93.5%. For group C, we consider another more recent DNN architecture called DenseNet [3] [23]. Training is performed for 100 epochs with batch size 32 and learning rate 10^{-3}. With the same training data augmentation, the test accuracy CIFAR-10 in absence of attack is around 94%.

In Table 5, we show the average attack success rate (ASR) and clean test accuracy (CTA) for the classifiers in group A, B, and C. Again, the attacks are successful with high ASR and very little degradation in CTA compared with the attack-free case.

A.7.3. Detection Performance Evaluation

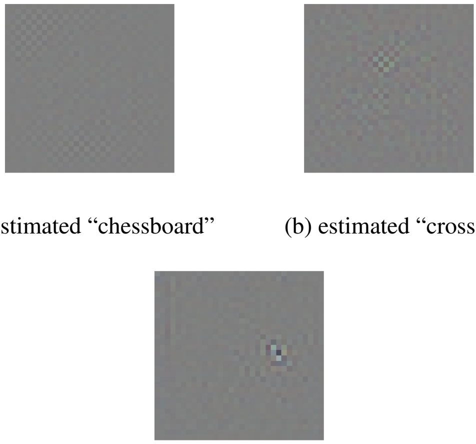
By applying the same RED-A on the classifiers in group A, B, and C, 10/10, 9/10, and 10/10 attacks are detected, respectively. For each attacks being detected, both source class and target class are also correctly inferred – though not perfectly, RED-A achieves satisfactory detection for attacks with various backdoor patterns and victim classifiers with various DNN architectures. Moreover, in Figure 5, we show the backdoor patterns estimated by RED-A for the attacks whose backdoor patterns are shown in Figure 4. Again, RED-A successfully estimates the key features (the recurrent feature of the global pattern and the location of the local patterns) of the true backdoor pattern involved in these attacks.

A.8. Details of Experiments on Other Datasets

Here, we evaluate our proposed defense on MNIST [24], FMNIST [25], GTSRB [26], and CIFAR-100 [27]. MNIST is a handwritten digit image dataset containing 60k training images and 10k test images, both evenly distributed in 10

[2] Link of implementation: `https://github.com/kuangliu/pytorch-cifar/blob/master/models/mobilenetv2.py`

[3] Link of implementation: `https://github.com/gpleiss/efficient_densenet_pytorch/blob/master/models/densenet.py`



(a) estimated “chessboard”

(b) estimated “cross”

(c) estimated “l-shape”

Fig. 5: Example backdoor patterns (with 127/255 offset to elucidate negative perturbations) estimated by RED-A for the three attacks whose true backdoor patterns being used are shown in Figure 4.

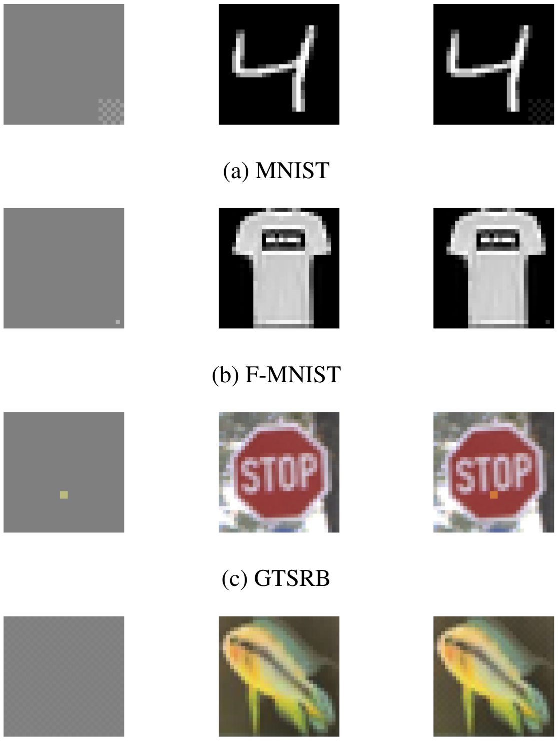
classes. FMNIST (Fashion-MNIST) also contains 60k training images and 10k test images, both evenly distributed in 10 classes. Images in both MNIST and FMNIST are 28 × 28 gray scale images. GTSRB is a traffic sign dataset with 43 classes, 26640 training images, and 12630 test images. We resize each image in GTSRB to 32. CIFAR-100 contains 50k training images and 10k test images, both evenly distributed in 10 classes. Similar to the images in CIFAR-10, the images in CIFAR-100 are 32 × 32 colored images. For convenience, the class indexing for these datasets used in this section follows the official description of the datasets.

A.8.1. Attack Crafting

For the attack on MNIST, we use a “local chessboard” pattern, i.e. a patch of the “chessboard” pattern located at the bottom right corner of each image to be embedded in. We choose the patch size to be 6 × 6 and perturbation size to be 25/255. The source classes and the target class are class $\{5, 7, 10\}$ and class 9, respectively. 500 backdoor training images per source class are created to poison the training set.

For the attack on FMNIST, we use a “single pixel” perturbation backdoor pattern [5, 7], where the pixel being perturbed is located at the bottom right of image, and the perturbation size is 50/255. The source class and the target class are class 1 and class 3, respectively. Due to that the perturbation size is small and there is only one pixel being perturbed, 900 backdoor training images are required to poison the training set to ensure an effective attack.

For the attack on GTSRB, we use the same “local square” pattern used in the main experiments on CIFAR-10. The source classes and the target class are class $\{13, 14, 15, 16, 17, 20, 24, 25, 27, 28, 29, 30\}$ and class 8, respectively. Class 8 images are for 120kph speed limit sign and the picked source classes images are for traffic signs requiring a slowing down or stopping. In practice, such an attack may cause an autonomous vehicle to speed up while seeing, e.g., a stop sign. For this attack, we create 30 backdoor training images per source class to poison the training set.



(a) MNIST

(b) F-MNIST

(c) GTSRB

(d) CIFAR-100

Fig. 6: Backdoor pattern (with 0.5 offset) (left), example backdoor training image (right), and originally clean image (middle), for attacks on MNIST, F-MNIST, GTSRB, and CIFAR-100.

For the attack on CIFAR-100, we use the same “chessboard” pattern as in Appendix A.7. The source classes and the target class are class $\{1, \cdots, 15\}$ and class 21, respectively. 50 backdoor training images per source class are used to poison the training set.

Example backdoor patterns, backdoor training images, and their original clean images for the four attacks are shown in Figure 6.

A.8.2. Training

We use LeNet-5 [24] DNN architecture for training both the attacked and the clean classifier on MNIST. Training is performed for 60 epochs, with batch size 256 and learning rate

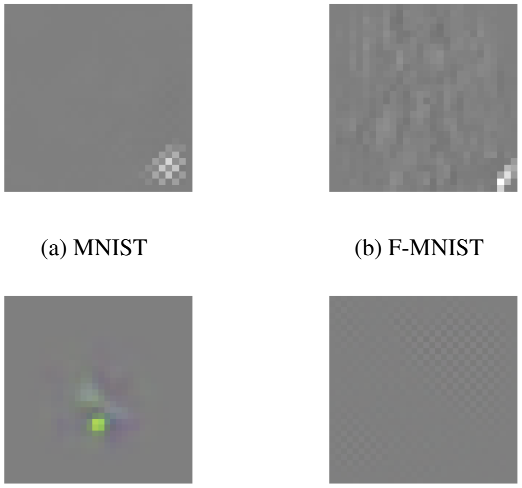
Table 6: Attack success rate (ASR) and clean test accuracy (CTA) of the classifiers trained on the backdoor poisoned training set; and CTA of the clean classifiers for MNIST, F-MNIST, GTSRB, and CIFAR-100.

	MNIST	F-MNIST	GTSRB	CIFAR-100
attack ASR	96.3	91.7	87.8	98.0
attack CTA	98.9	91.0	98.1	72.4
clean CTA	99.1	90.8	98.7	72.8

10^{-2}, without data augmentation. For F-MNIST, we use a "VGG-9" DNN architecture, which is customized by removing the last two convolutional layers of VGG-11 [28] and using 16, 32, 64, 64, 128, 128 filters in the six remaining convolutional layers, respectively. Training is performed for 60 epochs with batch size 256 and learning rate 10^{-2}, without data augmentation. For GTSRB, we use the same DNN architecture used in [13]. Training is performed for 100 epochs with batch size 64 and learning rate 10^{-3}, without data augmentation. For CIFAR-100, we use the standard ResNet-34 architecture [21]. Training is performed for 120 epochs with batch size 32 and learning rate 10^{-3}. The same training data augmentation options for training in the main experiments are used here. In Table 6, we show the ASR and CTA for each classifier being attacked, and CTA for the clean classifiers. Clearly, all the attacks are successful with the high ASR and almost no degradation in CTA compared to the clean benchmarks.

A.8.3. Defense

As shown in 4 of the main paper, our defense successfully detects all the four attacks with no false detection when there is no attack. Here, we show the estimated pattern for each detected attack in Figure 7. The main features of the backdoor patterns being used are recovered for all the four attacks. Also, the target class for each attack is correctly inferred by our defense.



(a) MNIST

(b) F-MNIST

(c) GTSRB

(d) CIFAR-100

Fig. 7: Backdoor patterns estimated by our defense for attacks on MNIST, FMNIST, GTSRB, and CIFAR-100.