MEDIC: Remove Model Backdoors via Importance Driven Cloning

Qiuling Xu, Guanhong Tao, Jean Honorio, Yingqi Liu, Shengwei An,
Guangyu Shen, Siyuan Cheng, Xiangyu Zhang
Purdue University
{xu1230, taog, jhonorio, liu1751, an93, shen447, cheng535, xyzhang}@purdue.edu

Abstract

We develop a novel method to remove injected backdoors in deep learning models. It works by cloning the benign behaviors of a trojaned model to a new model of the same structure. It trains the clone model from scratch on a very small subset of samples and aims to minimize a cloning loss that denotes the differences between the activations of important neurons across the two models. The set of important neurons varies for each input, depending on their magnitude of activations and their impact on the classification result. We theoretically show our method can better recover benign functions of the backdoor model. Meanwhile, we prove our method can be more effective in removing backdoors compared with fine-tuning. Our experiments show that our technique can effectively remove nine different types of backdoors with minor benign accuracy degradation, outperforming the state-of-the-art backdoor removal techniques that are based on fine-tuning, knowledge distillation, and neuron pruning.[1]

1. Introduction

Backdoor attack is a prominent threat to applications of Deep Learning models. Misbehaviors, e.g., model misclassification, can be induced by an input stamped with a *backdoor trigger*, such as a patch with a solid color or a pattern [14, 29]. A large number of various backdoor attacks have been proposed, targeting different modalities and featuring different attack methods [2, 22, 37, 44, 49, 55]. An important defense method is hence to remove backdoors from pre-trained models. For instance, Fine-prune proposed to remove backdoor by fine-tuning a pre-trained model on clean inputs [27]. Distillation [23] removes neurons that are not critical for benign functionalities. Model connectivity repair (MCR) [56] interpolates weight parameters of a poisoned model and its finetuned version. ANP [50] adversarially perturbs neuron weights of a poisoned model and prunes those neurons that are sensitive to perturbations (and hence considered compromised). While these techniques are very effective in their targeted scenarios, they may fall short in some other attacks. For example, if a trojaned model is substantially robust, fine-tuning may not be able to remove the backdoor without lengthy retraining. Distillation may become less effective if a neuron is important for both normal functionalities and backdoor behaviors. And pruning neurons may degrade model benign accuracy. More discussions of related work can be found in Section 2.

In this paper, we propose a novel backdoor removal technique MEDIC [2] as illustrated in Figure 1 (A). It works by cloning the benign functionalities of a pre-trained model to a new sanitized model of *the same structure*. Given a trojaned model and a small set of clean samples , MEDIC trains the clone model from scratch. The training is not only driven by the cross-entropy loss as in normal model training, but also by forcing the clone model to generate the same internal activation values as the original model *at the corresponding internal neurons*, i.e., steps ①, ②, and ④ in Figure 1 (A). Intuitively, one can consider it essentially derives the weight parameters in the clone model by resolving the activation equivalence constraints. There are a large number of such constraints even with just a small set of clean samples. However, such faithful cloning likely copies the backdoor behaviors as well as it tends to generate the same set of weight values. Therefore, our cloning is further guided by importance (step ③). Specifically, for each sample x, we only force the important neurons to have the same activation values. A neuron is considered important if (1) it tends to be substantially activated by the sample, when compared to its activation statistics over the entire population (the *activation criterion* in red in Figure 1 (A)), and (2) the large activation value has substantial impact on the classification result (the *impact criterion*) . The latter can be determined by analyzing the output gradient regarding the neuron activation. By constraining only the important neurons and relaxing the

[1] Code is available at https://github.com/qiulingxu/MEDIC

[2] MEDIC stands for "Remove **M**od**E**l *Backdoors via Importance* **D**r**I**ven **C**loning".

others, the model focuses on cloning the behaviors related to the clean inputs and precludes the backdoor. The set of weight values derived by such cloning are largely different from those in the original model. Intuitively, it is like solving the same set of variables with only a subset of equivalence constraints. The solution tends to differ substantially from that by solving the full set of constraints.

Example. Figure 2 shows an example by visualizing the internal activations and importance values for a trojaned model on a public benchmark [35]. Image (a) shows a right-turn traffic sign stamped with a polygon trigger in yellow, which induces the classification result of stop sign. Image (c) visualizes the activation values for the trojaned image whereas (d) shows those for its clean version. The differences between the two, visualized in (e), show that the neurons fall into the red box are critical to the backdoor behaviors. A faithful cloning method would copy the behaviors for these neurons (and hence the backdoor). In contrast, our method modulates the cloning process using importance values and hence precludes the backdoor. The last image shows the importance values with the bright points denoting important neurons and the dark ones unimportant. Observe that it prevents copying the behaviors of the compromised neurons *for this example* as they are unimportant. One may notice that the unimportant compromised neurons for *this example* may become important for another example. Our method naturally handles this using per-sample importance values. □

Our technique is different from fine-tuning as it trains from scratch. It is also different from *knowledge distillation* [25] shown in Figure 1 (B), which aims to copy behaviors across different model structures and constrains logits equivalence (and sometimes internal activation equivalence [53] as well). In contrast, by using the same model structure, we have a one-to-one mapping for individual neurons in the two models such that we can enforce strong constraints on internal equivalence. This allows us to copy behaviors with only a small set of clean samples.

We evaluate our technique on nine different kinds of backdoor attacks. We compare with five latest backdoor removal methods (details in related work). The results show our method can reduce the attack success rate to 8.5% on average with only 2% benign accuracy degradation. It consistently outperforms the other baselines by 25%. It usually takes 15 mins to sanitize a model. We have also conducted an adaptive attack in which the trigger is composed of existing benign features in clean samples. It denotes the most adversary context for MEDIC. Our ablation study shows all the design choices are critical. For example, without using importance values, it can only reduce ASR to 36% on average.

In summary, our main contributions include the following.

- We propose a novel importance driven cloning method to remove backdoors. It only requires a small set of clean samples.
- We theoretically analyze the advantages of the cloning method.
- We empirically show that MEDIC outperforms the state-of-the-art methods.

2. Related Work

There are a large body backdoor attacks [1, 7, 8, 15, 33, 34, 38–40, 57, 58]. We focus on a number of representative ones with different natures and used in our experiments. Readers interested in other attacks are referred to a survey [12, 24]. Badnet [14] proposes to inject backdoors with a fixed polygon. Clean Label attack [46] first adds adversarial perturbations to target-class samples and makes them misclassified. It then adds a square patch on those inputs without changing their ground-truth label. During inference, the square patch can induce misclassification on non-target class samples. SIG [3] uses a wave-like pattern as trigger. Reflection attack [30] adds the reflection of some external image on the input. Polygon attack [35] injects a polygon-like trigger on the input at specific locations. Filter attack [28] utilizes Instagram filters to transform inputs and labels the transformed inputs to the target class. Warp [32] uses image warping as a trigger pattern. The trigger perturbations hence vary for different inputs. We also consider an adaptive attack which uses benign features from two existing classes as the trigger [26].

Researchers have proposed various defense techniques, including backdoor inputs detection that aims to detect and reject input samples with triggers [9, 13, 45, 47]; backdoor scanning that determines whether a given model has backdoor using a small set of clean inputs [20, 28, 42, 48]; backdoor certification that certifies the predictions of individual samples with trigger [19, 31, 51, 52]; model hardening that adversarially trains models using triggers inverted from the subject (clean) model [43]; backdoor removal that erases injected backdoors using a small set of clean samples [23, 27, 50]. Our work falls in the last category. Fineprune [27] iteratively prunes neurons with small activation values on clean samples and then finetunes the pruned model. Note that the pruned model has different structure from the original one. In our experience, having the same structure is critical as it provides the optimal neuron alignment. Model connectivity repair (MCR) [56] interpolates parameters of a poisoned model and its finetuned version. NAD [23] first finetunes a poisoned model and then performs transfer learning, treating the finetuned model as the teacher and the poisoned model as the student. Its effectiveness hinges on the quality of the first finetuning step. ANP [50] adversarially perturbs neuron weights of the poisoned model and prunes neurons sensitive

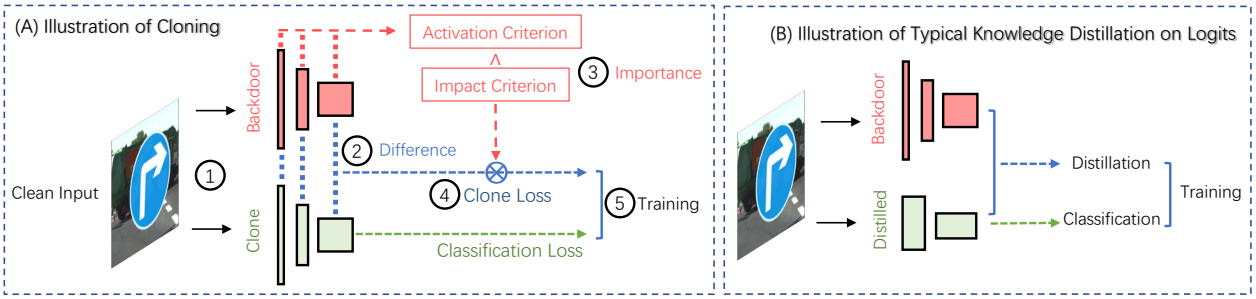

Figure 1. Workflow of MEDIC and comparison with knowledge distillation [17]

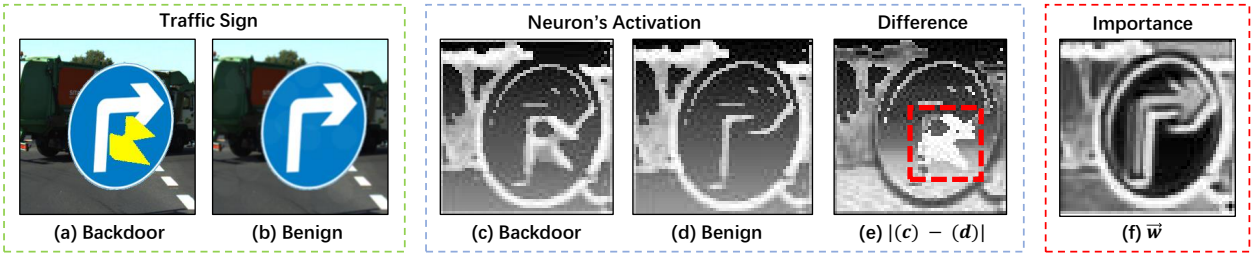

Figure 2. Example, with (a) and (b) the clean and trojaned inputs, (c)-(f) the internal activations of the trojaned input and the clean input, their differences, and the importance values in eq.(4) for the clean input, respectively. We resize and rescale the internal activation for better visualization.

to perturbations. We compare MEDIC with the aforementioned approaches in Section 5 and demonstrate that ours outperforms.

3. Method

Problem Statement. Given a multi-class backdoor classifier $f^* : X \to Y$, where $X \subset \mathbb{R}^d$ and $Y \subset \mathbb{R}^C$, d the input dimension and C the number of classes. Samples (X, Y) are drawn from the joint distribution $\mathcal{S} = (\mathcal{X}, \mathcal{Y})$. The classifier f^* is originally trained on a large dataset $D \sim (\mathcal{X}, \mathcal{Y})^n$ and additional backdoor samples $D_a \sim (\mathcal{X}_a, \mathcal{Y}_a)$. We aim to remove the backdoor behaviors in f^* and retain the benign behaviors. To do so, we leverage a small additional dataset $D_s \sim (\mathcal{X}, \mathcal{Y})^q$ where $|D_s| \ll |D|$. It is used to facilitate the cloning process. Function f^c denotes the cloned and sanitized model using our method.

Our first goal is to clone the (benign) functionalities of the model instead of its parameters. Specifically, f^c is supposed to produce similar outputs as the original model (with backdoor) on clean inputs. This is stated as the *functionality* goal as follows.

$$\mathbb{E}_{x\sim\mathcal{X}}\big(\|f^c(x) - f^*(x)\|_2\big) \approx 0 \qquad \text{(Functionality)}$$

In addition, f^c shall not have the backdoor behavior on samples from the backdoor distribution $(\mathcal{X}_a, \mathcal{Y}_a)$, with a high probability $1 - \delta$. It is stated as the *sanity* goal.

$$\mathbb{E}_{(x,y)\sim(\mathcal{X}_a,\mathcal{Y}_a)}\mathbb{1}\big[f^c(x) \neq y\big] \geq 1 - \delta \qquad \text{(Sanity)}$$

It is worth noting that similar functionalities do not imply a similar set of parameters. In fact, our importance driven cloning derives a set of parameters different from those in the original model to meet the two stated goals.

Cloning. To achieve the first goal of copying normal functionalities using only a small set of benign samples, we propose to train a clone model from scratch that has the same structure, by forcing the clone model to have the same activation values for all the corresponding neurons and also the output logits as the original model. Such cloning is different from transfer learning, in which teacher and student models may be of different structures as it is not that meaningful to copy functionalities to a model with the same structure in transfer learning's application scenarios.

While inputs often have a fixed bound, internal activation values have dynamic ranges. If we directly use activation value differences in the clone training loss, such range variations cause difficulties in convergence. Therefore, we normalize activation values before using them in the loss function. Formally speaking, given m internal neurons of a backdoored model, denoted by functions $f_i^* : R^d \to R$ with $i \in [1, m]$[3] and the corresponding ones in the clone model f_i^c, we aim to enforce the corresponding neuron functions to produce the same values. Let μ_i and σ_i be the mean and standard deviation of the random variable $f_i^*(\mathcal{X})$ and $w_i(x)$ an importance weight. We use $\overrightarrow{f(x)}$, $\overrightarrow{w(x)}$, $\vec{\sigma}$ and $\vec{\mu}$ to represent their concatenated vectors over the m neurons,

[3]We consider each element in a CNN feature map a separate neuron.

respectively. We can derive the following loss function.

$$\mathcal{L}_{\text{clone}} = \mathbb{E}_{x\sim\mathcal{X}}\left[\sum_i w_i(x)\left(\frac{f_i^*(x) - f_i^c(x)}{\sigma_i}\right)^2\right] \quad (1)$$
$$\text{where } w_i(x) \geq 0$$

Intuitively, we minimize the differences between the outputs of the corresponding neurons. For the moment, one can assume $w_i(x) = 1/m$, meaning that we enforce such constraints equally on all neurons. We will explain how we change $w_i(x)$ to preclude backdoor behaviors later in the section.

Our results in Figure 3 show that we can faithfully clone the normal functionalities with only 2% samples. We also formally analyze the essence of cloning in Section 4.

Pruning Backdoor by Importance Driven Cloning. To preclude backdoor while cloning, we leverage the importance value $w_i(x)$ computed for each sample and each neuron. A neuron is important for an sample, if and only if the neuron is substantially activated (compared to its activation statistics over the entire population) and has a large impact on the final output.

To determine if a neuron is substantially activated, we calculate the absolute difference between its activation and the expected activation as the indicator in Eq.(2). While ideally we would just select the neuron with the largest magnitude, to ensure the loss function is differentiable, we use softmax as an alternative to the max operation. Here τ is the temperature used in the softmax function, which can be used to control the strictness of the operation, namely, a high temperature resembles the strict max operation (that has a one-hot output), whereas a lower temperature suggests smoother results.

$$w^{\text{activate}}(x) = \text{SoftMax}(\frac{\tau|\overrightarrow{f(x)} - \vec{\mu}|}{\vec{\sigma}}) \quad (2)$$

It is worth noting that this formula essentially corresponds to the intrinsic normalization in many models (e.g., those with batch normalization [18]), where the statistics are readily accessible. When the subject model does not have any normalization layer, one can collect the statistics from the available samples. Details can be found in Appendix D.1.

To evaluate the impact of activations on classification results, we use the magnitude of gradients as an indicator. However, the gradients are not directly comparable at many layers, especially those that do not have normalization, because the varying magnitude of the activation values at those layers affects the gradients and makes direct comparison inaccurate. To tackle the problem, we introduce variables to denote normalized activations at all layers, called *proxy variables*, and derive gradients regarding those variables. The resulted gradients are hence comparable. Consider a proxy variable $h_i^*(x)$, where the neuron $f_i^*(x) = h_i^*(x) \cdot \sigma_i$. Let l be the loss function for classification based on $f_i^*(x)$, we can thus calculate the gradient of proxy variable as follows.

$$g_i^*(x) = \frac{\partial l(f_i^*(x), ...)}{\partial h_i^*(x)} = \sigma_i \frac{\partial l(f_i^*(x), ...)}{\partial f_i^*(x)}.$$

We then select the neuron with the largest gradient, leveraging the aforementioned soft version of max operation. Let $\overrightarrow{g(x)}$ be the concatenated vector of $g_i^*(x)$.

$$w^{\text{impact}}(x) = \text{SoftMax}(\tau\overrightarrow{g(x)}) \quad (3)$$

The next step is to identify the neurons satisfying both Eq.(2) and (3). In a discrete world, we could perform a set intersection. However, our softmax operations do not select the largest weight values. Instead, they produce vectors denoting the importance of all neurons. Hence, we perform a soft version of set intersection, which is differentiable and hence usable during training. Assume $w_i^{\text{activate}}(x),\ w_i^{\text{impact}}(x) \in [0, 1]$ denote the importance values for a neuron i, computed by the two respective softmax operations in Eq.(2) and (3), with 1 indicating "Important" and 0 "Unimportant". We can derive the neuron's overall importance as follows.

$$w_i(x) = w_i^{\text{activate}}(x) \wedge w_i^{\text{impact}}(x) = \sqrt{w_i^{\text{activate}}(x) \cdot w_i^{\text{impact}}(x)} \quad (4)$$

The operation $\wedge$ is essentially a soft version of "boolean conjunction". For example, when $a = 1$ and $b = 1$, $a \wedge b = 1$. The square root ensures the output is in the same magnitude as the inputs.

Given the cloning loss, we combine it with the classification loss and train the model. The two losses are balanced through an additional parameter λ. Details and an ablation study on λ can be found in Appendix E.3. A step-by-step algorithm can be found in Appendix D.1.

Difference from Neuron Pruning based Methods. There are existing works [27, 50] that determine a subset of neurons compromised by poisoning and simply remove those neurons. However, they usually lead to non-trivial benign accuracy degradation (see Section 5). The reason is that a neuron in a compromised model may be important for the classifications of both benign and poisoned inputs. In contrast, our method does not statically decide if a neuron is compromised. If a neuron is important for both benign and poisoned samples, we (only) copy its behaviors for benign inputs.

4. Formal Interpretation of Cloning

The effectiveness of MEDIC hinges on cloning. In this section, we study a critical property of cloning, which states that *cloning can copy the functionalities of original model using only a small set of clean samples*. We also formally show that cloning is superior to training from scratch using

the small set, and to knowledge distillation. That is, cloning tends to have a smaller loss value. Backdoor removal using importance values is merely an application of the property as we essentially leverage importance analysis to select a subset of functionalities only relevant to clean sample classifications for cloning. Figure 3 shows empirical evidence using an example. Observe that knowledge distillation has a larger improvement over training from scratch given more samples. Also observe that training from scratch and knowledge distillation cause non-trivial benign accuracy degradation. In contrast, cloning causes negligible degradation of original backdoor model using even just 2% of data.

We mainly utilize the decomposition, *Rademacher complexity* [5] and *Covering number* to derive an upper bound for the testing loss of a classifier with a high probability. We hence show that cloning has a smaller upper bound (of loss) compared to distillation and training from scratch using a small set of samples.

Based on the aforementioned analysis, we additionally prove the computation complexity of cloning in Appendix B. We show cloning can be orders of magnitude faster in recovering benign accuracy. Furthermore, in Appendix C, we theoretically show cloning can be more effective than fine-tuning in removing backdoors. This is done by analyzing a general backdoor scenario. We hence prove that cloning can guarantee the backdoor removal for this type of backdoor while fine-tuning can not. And we show the importance weight correctly identify the compromised neurons.

In the following, we first introduce the notations, terms, and a set of lemmas. We then formally show that cloning is better than training from scratch on a small set and distillation.

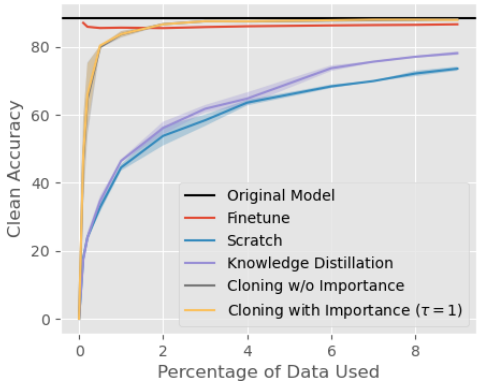

Figure 3. Benign accuracy of models after backdoor removal versus the data used (for a CIFAR-10 model poisoned with CleanLabel backdoor).

4.1. Notations and Backgrounds

We consider the original model f^*, model f^s trained from scratch using D_s (i.e., the small dataset), clone model f^c by our method and model f^k by knowledge distillation [17]. They belong to the function families $\mathcal{F}^*$, $\mathcal{F}^s$, $\mathcal{F}^c$, and $\mathcal{F}^k$, respectively. These families are determined by the specific optimization and learning algorithms. Since each algorithm has its learning bias and will likely generate a different set of parameters.

Neural Net. To facilitate the formal analysis, we leverage a simple network. Let ψ be the ReLU activation function, l_γ the loss function with the Lipschitz constant 2γ. Without loss of generality, we assume a two-layer network, where the two layers are abstracted to two respective matrices, $G \in \mathcal{G} \subset \mathbb{R}^{d\times p}$ and $H \in \mathcal{H} \subset \mathbb{R}^{p\times C}$, with p the number of hidden units. Let s be the softmax function. The network is represented as $f(x) = s \circ o(x)$, $o(x) = Hg(x)$ and $g(x) = \psi(Gx)$. Putting them together, $f(x) = s(H\psi(Gx))$. Intuitively, $f(x)$ outputs the probability, $o(x)$ is the logits value and $g(x)$ is the hidden layer. It is worth noting that our analysis can naturally extend to complex network since the proof structure stays the same.

Loss. As a common practice in multi-class analysis, we use the margin loss [4] as $l_\gamma(f(x), y)$. Its definition can be found in Appendix eq.(9). A smaller loss value indicates better performance.

Rademacher Complexity. Let $\mathcal{F}$ be a family of functions , D a set of samples, and σ uniformly sampled from $\{-1, +1\}^{|D|}$. The empirical Rademacher complexity is defined as follows. $\hat{R}(\mathcal{F}|D) = \mathbb{E}_\sigma[\sup_{f\in\mathcal{F}} \frac{1}{|D|} \sum_{x_i\in D} \sigma_i f(x_i)]$. Observe that if $\mathcal{F}$ contains only a fixed function, the value is 0 suggesting no complexity (and hence trivial to learn).

Covering Number Bound. It is difficult to directly compute Rademacher complexity in our context. We hence derive its upper-bound using *covering number*. Let $\|\cdot\|_{p,D}$ be a pseudo-norm on function family $\mathcal{F}$ with respect to a vector norm $\|\cdot\|_p$ and data D (definition in Appendix eq. (20)), we define the covering number $\mathcal{N}(\mathcal{F}, \|\cdot\|_{p,D}, \epsilon)$ as the size of minimal-ϵ cover of $\mathcal{F}$ under this pseudo-norm. Intuitively, this number means how many functions are representative such that they can cover all the learning outcomes with the ϵ bound. The covering number of a function family is dependent on the learning method. For example, when we include the loss to minimize the difference between $f^c \in \mathcal{F}^c$ and $f^* \in \mathcal{F}^*$ (during cloning), we essentially reduce the covering number of the function difference family $\mathcal{N}(\{f^c - f^*\}, \|\cdot\|_{p,D}, \epsilon)$.

Lemmas. In the following, we introduce three lemmas that are needed in later analysis. Their proofs can be found in the Appendix. In Lemma 1, we derive the Lipschitz constant of the loss function. In Lemma 2, we bound the Lipschitz of softmax function. In Lemma 3, we bound the Rademacher complexity by a function over the covering number.

Lemma 1. *For any logits $o_1, o_2 \in \mathbb{R}^c$ and $y \in Y$, we have* $|l_\gamma(o_1, y) - l_\gamma(o_2, y)| \leq 2\gamma||o_1 - o_2||_\infty$.

Lemma 2. *For any logits $o_1, o_2 \in \mathbb{R}^c$, we have $||s(o_1) - s(o_2)||_\infty \leq \frac{1}{2}||o_1 - o_2||_\infty$.*

Lemma 3 ([36]). *Given function family $\mathcal{F}$, data D and number of samples $n = |D|$, we define the function:*

$$B(\mathcal{F}|D) = \inf_{\epsilon \in [0, \epsilon_c/2]} \left\{ \epsilon + \frac{\sqrt{2}\epsilon_c}{\sqrt{n}} \sqrt{\log \mathcal{N}\left(\mathcal{F}, \|\cdot\|_{1,D}, \epsilon\right)} \right\},$$

where $\epsilon_c = \sup_{f \in \mathcal{F}} \|f\|_{2,D}$ .We then have $\hat{R}(\mathcal{F}|D) \leq B(\mathcal{F}|D)$

4.2. Analysis

To formally compare cloning, distillation, and training from scratch on a small set, we first study the advantage of cloning and distillation compared with training from scratch. The advantage comprises two parts, i.e., a gap and a regret. The gap is the training loss difference between training on the whole dataset and on a small part of the same dataset. The regret is the difference of (test) classification loss between the current model and the optimal model (f^*). Since the gap is not specific to our algorithm, we hence study the upper bound of the regret. A smaller regret indicates better classification performance. We finally show the upper bound of our regret is smaller than baselines.

Here we use distillation as an example. The analysis applies similarly to cloning as well. The advantage of distillation (f^k) over training from scratch on a small set (f^s) is denoted by their expected loss difference as follows.

$$\mathbb{E}_{(x,y)\sim\mathcal{S}}[l_\gamma(f^s(x), y) - l_\gamma(f^k(x), y)] = \underbrace{\Delta}_{\text{Loss Gap}} - \underbrace{\mathbb{E}_{\mathcal{S}}[(l_\gamma(f^k(x), y) - l_\gamma(f^*(x), y))]}_{\text{Regret of the Distilled Model}} \quad (5)$$

where $\Delta = E_{(x,y)\sim\mathcal{S}}[l_\gamma(f^s(x), y) - l_\gamma(f^*(x), y)]$. Note that as $|D_s| \ll |D|$, the loss gap Δ between f^* (training on a large dataset D) and f^s (training on the small dataset D_s) tends to be large. A large gap will make distillation and cloning favorable.

Next we analyze the upper-bound of the regret, since the loss gap is only dependent on the sample size difference. According to Equation 5 and the standard Rademacher complexity bound argument [5], we have the following regret upper bound between the distilled model and the original model with a high probability $1 - \delta$.

$$\begin{aligned} &E_{(x,y)\sim\mathcal{S}}[l_\gamma(f^k(x), y) - l_\gamma(f^*(x), y)] \\ &\leq \underbrace{\frac{1}{|D_s|} \sum_{(x,y)\in D_s} [l_\gamma(f^k(x), y) - l_\gamma(f^*(x), y)]}_{\text{Training Error}} + \\ &\underbrace{2\hat{R}(\mathcal{L}^k|D_s)}_{\text{Function Complexity}} + \underbrace{3\sqrt{\frac{log(2/\delta)}{2|D_s|}}}_{\text{Uncertainty}} \end{aligned} \quad (6)$$

Here, $\mathcal{L}^k$ is the function family of the regret between the distilled model and the original model, with $\mathcal{L}^k = \{h : X \times Y \to \mathbb{R} \mid h(x, y) = l_\gamma(f^k(x), y) - l_\gamma(f^*(x), y)\}$. The regret upper bound between the clone model and the original model can be similarly derived. Observe that the bound consists of three terms: *training error*, *function complexity* and *uncertainty*. The first and the third terms have a similar effect for both the distilled model f^k and the clone model f^c. Specifically, the training error is computed on training data. Since we only have limited samples and the model is prone to overfit on the training data. This term is close to zero empirically for both models. The uncertainty only depends on the data size $|D_s|$ and the probability δ. Therefore, the difference between the two models mainly lies in their function family complexities. In the following, we will analyze the the upper-bounds of the two function families. We use $\mathcal{O}_i^k = \{h : \mathbb{R}^d \to \mathbb{R} \mid h(x) = o^k(x)_i - o^*(x)_i\}$ to denote the family of differences regarding the i-th logits. The proof can be found in Appendix A.

Theorem 1. *For distilled model f^k and class number C, we have*

$$\hat{R}(\mathcal{L}^k|D_s) \leq \tilde{O}(\gamma\sqrt{C}) \max_i B(\mathcal{O}_i^k|D_s) \quad (7)$$

Let $\|H\|_{1,\infty} = \max_i \sum_j |H_{i,j}|$ be the entry-wise matrix norm, $\mathcal{I}_i^c = \{h : \mathbb{R}^d \to \mathbb{R} \mid h(x) = g^c(x)_i - g^*(x)_i\}$ the family of differences on an intermediate neuron i (between the clone and original models), and $\mathcal{J}_i^* = \{h : \mathbb{R}^d \to \mathbb{R} \mid h(x) = g^*(x)_i\}$ the function family of intermediate neuron i of the original model. We have the following theorem.

Theorem 2. *For cloned model f^c, let $\omega_c = \sup_{H^c \in \mathcal{H}^c} \|H^c\|_{1,\infty}$, $\beta_c = \sup_{H^c \in \mathcal{H}^c, H^* \in \mathcal{H}^*} \|H^c - H^*\|_{1,\infty}$ and class number C. We have*

$$\begin{aligned} \hat{R}(\mathcal{L}^c|D_s) \leq \tilde{O}(\gamma\sqrt{C}) \min \Big[&\underbrace{\max_i B(\mathcal{O}_i^c|D_s)}_{\textit{Constrained on Logits}}, \\ &\underbrace{\max_j \omega_c B(\mathcal{I}_j^c|D_s) + \beta_c B(\mathcal{J}_j^*|D_s)}_{\textit{Constrained on Intermediate Layers}} \Big] \end{aligned} \quad (8)$$

From the theorem 1, decreasing covering numbers on logits difference family $\mathcal{O}_i^k$ can reduce the regret and improve the performance. Since knowledge distillation directly controls the covering number by applying $\mathcal{L}_2$ constraints on logits, these constraints improve the performance according to the theorem. From the theorem 2, we can further understand the benefits of Cloning. By comparing two theorems, we find that the upper-bound for cloned model is always smaller than that of the distilled model given $\mathcal{O}_i^k = \mathcal{O}_i^c$ (i.e., both models have their logits similar to the original model's). Intuitively, the advantage of cloning originates from the additional intermediate constraints, which reduce the covering

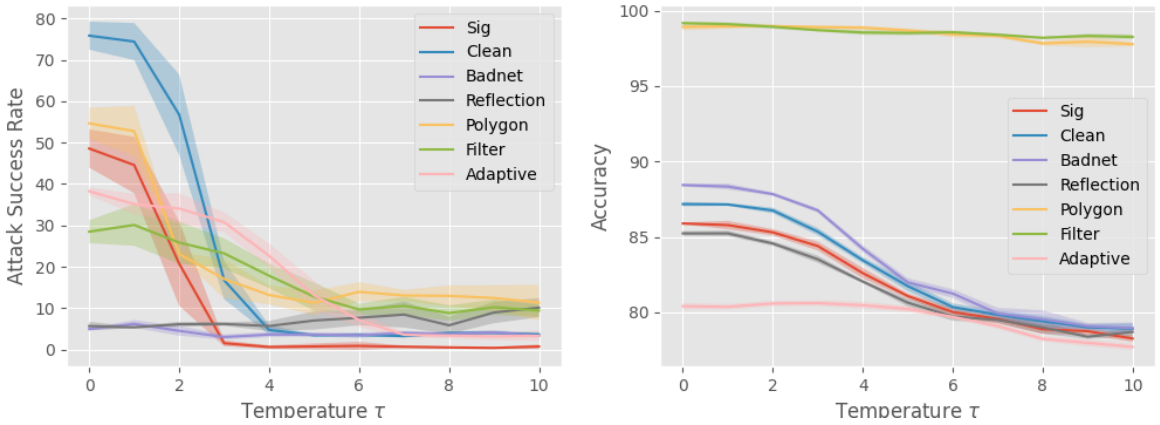

Figure 4. Effectiveness of importance. The shaded area represents standard deviation.

number of $\mathcal{I}_i^c$, i.e. $\mathcal{N}(\mathcal{I}_i^c, \|\cdot\|_{1,D}, \epsilon)$ in $B(\mathcal{I}_i^c|D_s)$ and the upper-bound on the regret. Also note that the other family $\mathcal{J}_i^*$ is unrelated to cloning.

5. Experiment

Setting. We conduct experiments on nine representative backdoor attacks. Specifically, we use Wide ResNet [54] and CIFAR-10 [21] for the common benchmark attacks and the adaptive attacks including Badnet [14], Clean Label [46], SIG [3], Reflection [30], Warp [32] and two Adaptive Attacks [26]. 5% data is used for removing backdoor behaviors on CIFAR-10. We further experiment on large-scale datasets Kitti-City [11] and Kitti-Road [11], and public backdoor benchmarks Polygon and Filter, based on ResNet [54]. We compare with five latest backdoor removal methods including Finetune, Fineprune [27], NAD [23], MCR [56], and ANP [50]. More details of these attacks and removal methods can be found in Section 2 and Appendix D.

Comparison with Baselines. In Table 1, we show the results in comparison with other backdoor removal methods. The fourth column shows the accuracy and attack success rate (ASR) of the original (trojaned) model. The following columns show the results after applying various backdoor removal methods. All the methods use the same set of data augmentations. We also adjust the temperature of MEDIC so that the test accuracies of our repaired models are comparable to others. From the results, we find the advantage of our algorithm outperforming other baselines lies in the cases of hard-to-remove backdoors. We retrain the model whereas others do not. This different design ensures some deep-rooted backdoors can be removed. Observe that other baselines are less effective in removing Clean Label, SIG, Polygon, and Filter backdoors as the ASRs are still high after the removal. In contrast, MEDIC can substantially reduce the ASRs of these attacks. On the other hand, when the backdoor attack is not robust (e.g. simpler attacks like Badnet and Reflection where Finetune can already remove them), the optimal results will favor methods which cause fewer changes to the model, since few changes suffice to remove the backdoor. From the security perspective, a resilient backdoor will be used by attackers more often, which suggests those hard-case backdoor attacks are more important. Note that even in those easier cases, MEDIC is still competitive. In the last row of Table 1, we show the average ASR reduction and accuracy degradation. Observe that MEDIC achieves 25% more ASR reduction compared to others with minor accuracy degradation.

Adaptive Attacks. We evaluate two adaptive attacks. In adaptive attack 1, the trigger pattern can exist on benign samples [26]. It utilizes multiple natural objects from different classes as the backdoor. In adaptive attack 2, we adversarially optimize the trigger to maximize the clone loss. The trigger will have a high activation value on the important neurons and may survive after cloning. Note that these represent the most adversarial contexts against MEDIC as the backdoor is essentially benign features or benign neurons that cloning would likely copy. Observe that MEDIC is still quite effective against Adaptive1 and Adaptive2 attacks as shown in Table 1. For the variant 1, while benign features are used to compose the backdoor, these features are not important for the target class. As such, their behaviors are hence not copied for the target class. For the variant 2, despite the substantial backdoor activations on important neurons, the cloning will only connect the benign functions to the output due to the absence of the trigger pattern in the clean data.

Effectiveness of Using Importance Values. In Figure 4, we show the effects of cloning using different temperatures. A temperature $\tau = 0$ means we do not use the importance and treat all neurons as equal. Observe that a larger temperature prunes more backdoor-related behaviors. The ASR is reduced by at most 70% with at most 4% accuracy drop (with a temperature of 5).This shows our importance values capture the characteristics of benign functionalities.

Stronger Backdoor Baselines. Different from the results reported in recent works [23,50,56], our results are evaluated on stronger backdoor baselines. We found training backdoor

Table 1. Comparing with other methods. ± represents the standard deviation over 5 repeated runs.

Scenario	Attack	Metric	Original (%)	Method (in percentage %)					
				Finetune	Fineprune	NAD	MCR	ANP	MEDIC
Common Benchmark Attacks	Clean Label	ASR	100.0	89.2 ± 20.2	99.6 ± 0.2	95.9 ± 7.2	90.3 ± 19.4	69.9 ± 17.6	$\mathbf{16.8 \pm 4.6}$
		Acc.	88.5	85.2 ± 0.5	85.5 ± 0.3	85.3 ± 0.5	85.2 ± 0.4	80.6 ± 1.4	85.3 ± 0.2
	SIG	ASR	94.5	68.3 ± 8.3	41.9 ± 23.9	55.8 ± 12.2	56.3 ± 9.0	73.9 ± 8.9	$\mathbf{1.5 \pm 0.7}$
		Acc.	86.9	85.0 ± 0.2	85.0 ± 0.5	85.2 ± 0.2	84.6 ± 0.2	82.5 ± 0.8	84.4 ± 0.3
	Badnet	ASR	100.0	3.2 ± 0.4	5.1 ± 1.6	3.0 ± 1.2	2.7 ± 0.4	$\mathbf{2.3 \pm 1.4}$	3.6 ± 0.6
		Acc.	88.9	83.1 ± 0.3	84.3 ± 1.2	83.5 ± 0.7	83.6 ± 0.4	83.0 ± 1.5	84.2 ± 0.2
	Reflection	ASR	34.2	7.6 ± 1.9	5.7 ± 0.9	8.7 ± 2.2	$\mathbf{4.6 \pm 0.4}$	5.0 ± 4.2	6.2 ± 0.5
		Acc.	84.3	84.5 ± 0.3	84.4 ± 0.2	84.1 ± 0.6	84.4 ± 0.3	80.3 ± 0.9	83.5 ± 0.2
	Warp	ASR	96.5	13.2 ± 4.4	4.7 ± 0.6	6.1 ± 1.8	4.4 ± 0.4	5.6 ± 1.2	$\mathbf{3.7 \pm 0.5}$
		ACC	86.2	84.5 ± 0.2	84.6 ± 0.2	84.7 ± 0.1	84.2 ± 0.2	83.3 ± 0.9	85.2 ± 0.1
Public Benchmarks on Large Datasets	Polygon	ASR	99.9	60.4 ± 14.6	47.9 ± 19.0	19.4 ± 10.3	39.4 ± 21.0	47.3 ± 8.1	$\mathbf{13.2 \pm 2.3}$
		Acc.	99.7	99.0 ± 0.2	98.7 ± 0.6	95.7 ± 1.6	98.5 ± 0.5	98.2 ± 0.8	98.9 ± 0.1
	Filter	ASR	100.0	62.0 ± 14.9	56.6 ± 9.8	47.9 ± 10.3	72.2 ± 5.5	19.5 ± 4.0	$\mathbf{17.8 \pm 3.0}$
		Acc.	99.7	99.3 ± 0.3	99.1 ± 0.2	99.0 ± 0.3	99.4 ± 0.2	98.3 ± 0.4	98.6 ± 0.2
Adaptive Attacks	Variant 1	ASR	79.3	37.1 ± 2.6	36.4 ± 0.8	38.0 ± 3.1	9.4 ± 1.1	33.4 ± 11.3	$\mathbf{7.1 \pm 1.7}$
		Acc.	83.0	80.2 ± 0.2	79.5 ± 0.5	79.8 ± 0.3	79.4 ± 0.2	76.7 ± 0.7	79.7 ± 0.1
	Variant 2	ASR	98.5	94.9 ± 2.7	77.7 ± 5.2	58.5 ± 22.0	86.7 ± 7.6	24.4 ± 6.3	$\mathbf{4.3 \pm 0.5}$
		Acc	85.9	82.9 ± 0.7	83.2 ± 0.3	83.1 ± 0.2	82.4 ± 0.3	81.4 ± 0.6	84.2 ± 0.1
	Avg. Drop	ASR	-	35.5%	41.9%	47.4%	43.1%	53.8%	**79.5%**
		Acc.	-	2.2%	2.1%	2.6%	2.4%	4.5%	2.3%

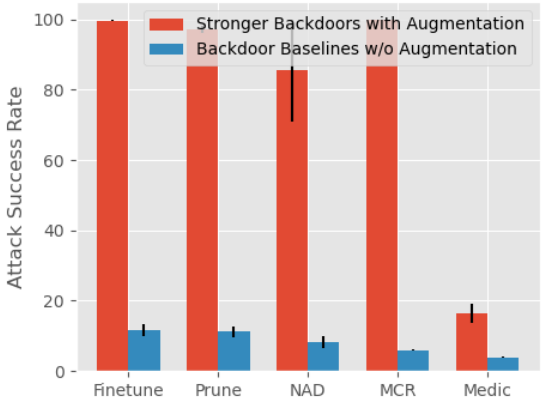

Figure 5. Removal methods on backdoor models trained with and without data augmentation during attack. The black lines denote the standard deviations. The results are from Clean Label attack.

models with data augmentation can make it much harder to remove. Figure 5 shows the effectiveness of different removal methods on backdoor attacks trained with and without data augmentations. The model is trojaned by a clean label attack on CIFAR-10. The red bars denote the ASRs after removal when augmentations are used during attack and the blue bars the ASRs when augmentations are not used. Observe that with augmentations, the baselines can hardly remove the backdoor. In contrast, MEDIC can effectively remove the backdoor with and without augmentations. Note that the same data augmentation is also used when applying defense methods.

Other Experiments. In Appendix E.1, we include another adaptive attack. In Appendix E.2, we conduct the ablation study given the distribution shift of training data. The results show our method can perform well under the distribution shift. In Appendix E.4, we conduct an ablation study on the amount of available data. We show that the trigger removal performance is positively correlated with the data amount. In Appendix E.6, we conduct the ablation study on the necessity of importance criterions and show that our design is crucial. In Appendix E.7, we conduct the ablation study on the backdoor model architectures and show that our method can generalize to different types of models.

6. Conclusion

We propose a novel backdoor removal method by importance driven cloning. We empirically and theoretically show that our method is superior to state-of-the-art baselines on a large set of evaluated attacks.

7. Acknowledgement

We thank the anonymous reviewers for their constructive comments. This research was supported, in part by IARPA TrojAI W911NF-19-S-0012, NSF 1901242 and 1910300, ONR N000141712045, N000141410468 and N000141712947. Any opinions, findings, and conclusions in this paper are those of the authors only and do not necessarily reflect the views of our sponsors.

References

[1] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. *arXiv preprint arXiv:2005.03823*, 2020. 2

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *AISTATS*, pages 2938–2948, 2020. 1

[3] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *ICIP*, pages 101–105. IEEE, 2019. 2, 7, 16

[4] Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017. 5

[5] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2002. 5, 6

[6] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 14

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

[8] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. In *AAAI*, 2021. 2

[9] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attack against deep learning systems. *SPW*, 2020. 2

[10] Dylan J. Foster and Alexander Rakhlin. $\mathcal{L}_\infty$ vector contraction for rademacher complexity. *CoRR*, abs/1911.06468, 2019. 13

[11] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013. 7, 16

[12] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Anmin Fu, Surya Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760*, 2020. 2

[13] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *ACSAC*, pages 113–125, 2019. 2

[14] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. 1, 2, 7, 16

[15] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 2019. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 16

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 3, 5

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 4

[19] Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of nearest neighbors against data poisoning attacks. In *AAAI*, 2020. 2

[20] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *CVPR*, 2020. 2

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 7, 16

[22] Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. In *ACL*, 2020. 1

[23] Yige Li, Nodens Koren, Lingjuan Lyu, Xixiang Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021. 1, 2, 7, 16

[24] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*, 2020. 2

[25] Zhizhong Li and Derek Hoiem. Learning without forgetting. *CoRR*, abs/1606.09282, 2016. 2

[26] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *CCS*, 2020. 2, 7, 16

[27] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018. 1, 2, 4, 7, 16

[28] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *CCS*, pages 1265–1282, 2019. 2, 16

[29] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*, 2018. 1

[30] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 2020. 2, 7, 16

[31] Michael McCoyd, Won Park, Steven Chen, Neil Shah, Ryan Roggenkemper, Minjune Hwang, Jason Xinyu Liu, and David Wagner. Minority reports defense: Defending against adversarial patches. In *International Conference on Applied Cryptography and Network Security*, pages 564–582. Springer, 2020. 2

[32] Anh Nguyen and Anh Tran. Wanet – imperceptible warping-based backdoor attack, 2021. 2, 7

[33] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. In *ICLR*, 2021. 2

[34] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *NeurIPS*, 33, 2020. 2

[35] NIST. TrojAI Leaderboard. https://pages.nist.gov/trojai/, 2019. 2, 16

[36] Patrick Rebeschini. Lecture notes in algorithmic foundations of learning: Covering numbers bounds for rademacher complexity, 2020. 6, 12

[37] Shahbaz Rezaei and Xin Liu. A target-agnostic attack on deep models: Exploiting security vulnerabilities of transfer learning. In *ICLR*, 2020. 1

[38] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *AAAI*, 2020. 2

[39] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. *arXiv preprint arXiv:2003.03675*, 2020. 2

[40] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *NeurIPS*, 2018. 2

[41] Te Juin Lester Tan and Reza Shokri. Bypassing backdoor detection algorithms in deep learning. In *IEEE European Symposium on Security and Privacy, EuroS&P 2020, Genoa, Italy, September 7-11, 2020*, pages 175–183. IEEE, 2020. 16

[42] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021. 2

[43] Guanhong Tao, Yingqi Liu, Guangyu Shen, Qiuling Xu, Shengwei An, Zhuo Zhang, and Xiangyu Zhang. Model orthogonalization: Class distance hardening in neural networks for better security. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022. 2

[44] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *ESORICS*, pages 480–501, 2020. 1

[45] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, pages 8000–8010, 2018. 2

[46] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018. 2, 7, 16

[47] Akshaj Kumar Veldanda, Kang Liu, Benjamin Tan, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, Brendan Dolan-Gavitt, and Siddharth Garg. Nnoculation: broad spectrum and targeted treatment of backdoored dnns. *arXiv preprint arXiv:2002.08313*. 2

[48] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *S&P*, pages 707–723, 2019. 2

[49] Bolun Wang, Yuanshun Yao, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. With great training comes great vulnerability: Practical attacks against transfer learning. In *USENIX Security*, pages 1281–1297, 2018. 1

[50] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2, 4, 7, 16

[51] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021. 2

[52] Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. Patchcleanser: Certifiably robust defense against adversarial patches for any image classifier. *arXiv preprint arXiv:2108.09135*, 2021. 2

[53] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *CoRR*, abs/1612.03928, 2016. 2

[54] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. 7, 16

[55] Xinyang Zhang, Zheng Zhang, and Ting Wang. Trojaning language models for fun and profit. In *European S&P*, 2021. 1

[56] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *International Conference on Learning Representations*, 2020. 1, 2, 7, 16

[57] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *CVPR*, 2020. 2

[58] Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *ICML*, pages 7614–7623, 2019. 2

A. Proof

Here we define the margin loss.

$$l_\gamma(f(x),y)=\phi_\gamma(f(x)_y-\max_{i\neq y}f(x)_i),\quad \text{where } \phi_\gamma(t)=\begin{cases}1, & t<0\\ 1-\gamma t, & 0\le t\le 1/\gamma\\ 0, & t>1/\gamma\end{cases} \tag{9}$$

Lemma 1. *Let $l_\gamma(x,y):\mathbb{R}^c\times Y\to\mathbb{R}$ and $\phi_\gamma(x):\mathbb{R}\to\mathbb{R}$ where*

$$\begin{aligned} l_\gamma(x,y)&=\phi_\gamma[f(x)_y-\max_{i\neq y}f(x)_i],\\ \textit{where } \phi_\gamma(x)&=\begin{cases}1, & x<0\\ 1-\gamma x, & 0\le x\le 1/\gamma\\ 0, & x>1/\gamma\end{cases}.\end{aligned} \tag{10}$$

For any $x_1,x_2\in R^c$ and $y\in Y$, we have

$$|l_\gamma(x_1,y)-l_\gamma(x_2,y)|\le 2\gamma||x_1-x_2||_\infty \tag{11}$$

Proof. It is apparent that $\phi_\gamma(x):\mathbb{R}\to\mathbb{R}$ is γ Lipschitz. And we have

$$\begin{aligned} &|l_\gamma(x_1,y)-l_\gamma(x_2,y)|\\ =&\Big|\phi_\gamma\Big[(f(x_1)_y-f(x_2)_y)-(\max_{i\neq y}f(x_1)_i-\max_{j\neq y}f(x_2)_j)\Big]\Big|\\ \le&\gamma\big|(f(x_1)_y-f(x_2)_y)-(\max_{i\neq y}f(x_1)_i-\max_{j\neq y}f(x_2)_j)\big|\\ \le&\gamma\big|(f(x_1)_y-f(x_2)_y)\big|+\gamma\big|\max_{i\neq y}f(x_1)_i-\max_{j\neq y}f(x_2)_j\big|\\ \le&2\gamma||f(x_1)-f(x_2)||_\infty\end{aligned} \tag{12}$$

□

Lemma 2. *Assume the softmax function $s:\mathbb{R}^c\to\mathbb{R}^c$ is defined as follows.*

$$s(x)_i=\frac{\exp(x_i)}{\sum_i\exp(x_i)} \tag{13}$$

Then we have

$$||s(x)-s(y)||_\infty\le\frac{1}{2}||x-y||_\infty. \tag{14}$$

Moreover we have

$$\begin{aligned}&\big\|(s(x)-s(y))-(s(x')-s(y'))\big\|_\infty\le\\ &\frac{1}{2}(\|x-x'\|_\infty+\|y-y'\|_\infty)\end{aligned} \tag{15}$$

In other words, $s(x)$ and its residual form $s(x)-s(y)$ are both $1/2$-Lipschitz with respect to L_∞-norm.

Proof. From the definition of Lipschitz continualty, we have $|s(x)-s(y)|_\infty\le L|x-y|_\infty$. Note that $|s(x)-s(y)|_\infty\le\sup_i|s_(x)_i-s(y)_i|$.

The gradient is hence the following.

$$\frac{ds(x)_i}{dx_j}=\begin{cases}s(x)_i(1-s(x)_i), & \text{i = j}\\ -s(x)_is(x)_j, & \text{i} \neq \text{j}\end{cases} \tag{16}$$

Note that $||\nabla_i s(x)||_1 = 2s(x))_i(1 - s(x))_i) \leq 1/2$. Let $\preceq$ represents the generalized inequality of the nonnegative orthant. By the mean value theorem, for some δ, s.t. $x \preceq \delta \preceq y$, we have

$$s(x)_i - s(y)_i \leq \nabla_i s(\delta)^T (x - y)\,. \tag{17}$$

By Holder's inequality,

$$\nabla_i s(\delta)^T (x - y) \leq ||\nabla_i s(\delta)||_1 ||x - y||_\infty \leq \frac{1}{2}||x - y||_\infty \tag{18}$$

To prove the second goal, we can find that

$$\begin{aligned} &||(s(x) - s(y)) - (s(x') - s(y'))||_\infty \\ =&||(s(x) - s(x')) - (s(y) - s(y'))||_\infty \\ \leq&||(s(x) - s(x'))||_\infty + (s(y) - s(y')||_\infty \\ \leq&\frac{1}{2}(||x - x'|_\infty + ||y - y'||_\infty) \end{aligned} \tag{19}$$

□

Lemma 3 ([36]). *We add the proof of this lemma here for completeness. Let $f : \mathbb{R} \to \mathbb{R} \in \mathcal{F}$ be a class of real-value functions, D be a set of samples. We define a pseudo-metric $\|\cdot\|_{p,D}$ on functions $\mathcal{F}$ with respect to vector norm $\|\cdot\|_p$ and n samples $|D| = n$, where*

$$\|f\|_{p,D} = \left[\frac{1}{n}\sum_{x\in D} |f(x)|^p\right]^{\frac{1}{p}}. \tag{20}$$

We define the Covering number $\mathcal{N}(\mathcal{F}, \|\cdot\|_{p,D}, \epsilon)$ as the size of the minimal ϵ-cover of $\mathcal{F}$ with respect to pseudo-norm $\|\cdot\|_{p,D}$. Given

$$\sup_{f\in F} \|f\|_{2,D} \leq s_D\,, \tag{21}$$

we have

$$\hat{R}(\mathcal{F}|D) \leq \inf_{\epsilon\in[0,s_D/2]} \left\{\epsilon + \frac{\sqrt{2}s_D}{\sqrt{n}}\sqrt{\log \mathcal{N}(\mathcal{F}, \|\cdot\|_{1,D}, \epsilon)}\right\} \tag{22}$$

Proof. For any ϵ and D, let $\mathcal{C}$ be the minimal-ϵ cover of $\mathcal{F}$, which means for each function f, there exists $f_\epsilon \in \mathcal{C}$ such that $\|f - f_\epsilon\|_{1,D} \leq \epsilon$

$$\begin{aligned} &\hat{R}(\mathcal{F}|D) \\ =&E_\sigma\left[\frac{1}{n}\sup_{f\in\mathcal{F}}\sum_i \sigma_i f(x_i)\right] \\ \leq&E_\sigma\left[\frac{1}{n}\sup_{f\in\mathcal{F}}\sum_i \sigma_i \left(f(x_i) - f_\epsilon(x_i)\right)\right] + \\ &E_\sigma\left[\frac{1}{n}\sup_{f_\epsilon\in\mathcal{C}}\sum_i \sigma_i f_\epsilon(x_i)\right] \\ \leq&\epsilon + E_\sigma\left[\frac{1}{n}\sup_{f_\epsilon\in\mathcal{C}}\sum_i \sigma_i f_\epsilon(x_i)\right] \\ \leq&\epsilon + \sup_{f\in\mathcal{F}}\sqrt{\sum_{x_i\in D} f(x_i)^2}\frac{\sqrt{2\log|\mathcal{C}|}}{n}\text{(Massart's Lemma)} \\ \leq&\epsilon + \frac{\sqrt{2}s_D}{\sqrt{n}}\sqrt{\log \mathcal{N}(\mathcal{F}, \|\cdot\|_{1,D}, \epsilon)} \end{aligned}$$

Since this bound stands for any ϵ, Thus we have

$$\hat{R}(\mathcal{F}|D) \leq \inf_{\epsilon \in [0, s_D/2]} \left\{ \epsilon + \frac{\sqrt{2} s_D}{\sqrt{n}} \sqrt{\log \mathcal{N}\left(\mathcal{F}, \|\cdot\|_{1,D}, \epsilon\right)} \right\} \tag{23}$$

□

Theorem 1. *For distilled model f^k and class number C, we have*

$$\begin{aligned} &\hat{R}(\mathcal{L}^k|D_s) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \max_i \hat{R}(\mathcal{O}_i^k|D_s) \qquad (24) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \max_i B(\mathcal{O}_i^k|D_s) \qquad (25) \end{aligned}$$

Proof. From Lemma 1 and Lemma 2, we know the Lipschitz constant of $l_\gamma \circ s$ is γ. Using the $\mathcal{L}_\infty$-contraction property of Rademacher complexity [10], we can expand the complexity of loss function to the complexity of logits values in Eq. (24). Eq. (25) is from Lemma 3. □

Theorem 2. *For cloned model f^c, let $\omega_c = \sup_{H^c \in \mathcal{H}^c} \|H^c\|_{1,\infty}$, $\beta_c = \sup_{H^c \in \mathcal{H}^c, H^* \in \mathcal{H}^*} \|H^c - H^*\|_{1,\infty}$ and class number C. We have*

$$\begin{aligned} \hat{R}(\mathcal{L}^c|D_s) \leq \tilde{O}(\gamma\sqrt{C}) \min \big[\\ \max_i B(\mathcal{O}_i^c|D_s), \max_j \omega_c B(\mathcal{I}_j^c|D_s) + \beta_c B(\mathcal{J}_j^*|D_s) \big] \end{aligned}$$

Proof.

$$\begin{aligned} &\hat{R}(\mathcal{L}^c|D_s) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \max_i \hat{R}(\mathcal{O}_i^c|D_s) \qquad (26) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \max_i \hat{R}\big(\{[H^c(g^c(x) - g^*(x)) \\ &\qquad + (H^c - H^*)g^*(x)]_i\}\big) \qquad (27) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \{\sup_{H^c} \|H^c\|_{1,\infty} \max_i \hat{R}(\mathcal{I}_i^c|D_s) \\ &\quad + \sup_{H^c, H^*} \|H^c - H^*\|_{1,\infty} \max_i \hat{R}(\mathcal{J}_i^*|D_s)\} \qquad (28) \\ \leq &\tilde{O}(\gamma\sqrt{C}) \left[\omega_c \max_i B(\mathcal{I}_i^c|D_s) + \beta_c \max_i B(\mathcal{J}_i^*|D_s)\right] \qquad (29) \end{aligned}$$

Eq. (26) is similar to that in Eq. (24). Eq. 27 is by expanding the functions in $\mathcal{O}_i^c$. Eq. (28) is the property of Rademacher complexity under linear transformation. Eq. (29) is from Lemma 3. Similar to Theorem 1, we have $\hat{R}(\mathcal{L}^c|D_s) \leq \tilde{O}(\gamma\sqrt{C}) \max_i B(\mathcal{O}_i^c|D_s)$. Together with Eq. (29), we finish the proof. □

B. Analysis over Computational Efficiency

In this section, we further prove that MEDIC is more efficient compared with training from scratch and distillation from logits under certain assumptions. The key idea is that our algorithm has a smaller sample complexity. Thus MEDIC requires optimization over a smaller set of examples. And this smaller set of training samples can guarantee faster convergence speed and less computation power.

We first prove our method has a smaller sample complexity.

For simplicity, we assume the optimal hypothesis exists in our hypothesis family. From equation 6, we further derive the upper bound of sample complexity of our method. The sample complexity represents how many samples are required to learn a model with a good performance. Note that because we assume the optimal hypothesis existes in our hypothesis family, the training regret in equation 1 will be zero for the best hypothesis.

Lemma 4. *Specifically, with probability* $1-\delta$*, we only need as many as* $|D_s|$ *samples to achieve a loss regret* β*, where*

$$|D_s| \leq \frac{9\log(2/\delta)}{(\beta - 2\hat{R}(\mathcal{L}|D_s))^2} \tag{30}$$

and

$$\beta = E_{(x,y)\sim\mathcal{S}}[l_\gamma(f^c(x),y) - l_\gamma(f^*(x),y)] \tag{31}$$

.

Proof. The proof of this lemma is fairly straightforward, which only requires some transformations from equation 6. □

From theorem 2, we also know cloned model has a smaller Rademacher Complexity $\hat{R}(\mathcal{L}|D_s)$. Combined with Lemma 4, this indicates cloned model has a smaller sample complexity.

Next, we show a small sample complexity indicates faster convergence. Formally, consider Gradient Descent algorithm (GD) [6], assume a convex loss function $\ell(w) = \frac{1}{|D_s|}\sum_{(x,y)\in D_s} l_\gamma(f^c(x;w),y)$ and the gradient $\nabla_w \ell(w)$ from different baselines are Lipschitz continuous with constant L. We choose a learning rate s such that $s \leq 1/L$.

Given the optimal weight w_* on the training data, the error bound at optimization step k can be written as [6]

$$\ell(w_k) - \ell(w_*) \leq \frac{||w_* - w_0||_2^2}{ks} \tag{32}$$

and requires a computation power dependent on the sample complexity

$$O(|D_s|k) \leq O(\frac{klog(\delta)}{(\beta - 2\hat{R}(\mathcal{L}|D_s))^2}) \tag{33}$$

Note that error bound has the same convergence rate $\frac{1}{ks}$ for different methods. However, the computation cost is different for different baselines due to different $\hat{R}(\mathcal{L}|D_s)$. The result shows MEDIC will cost less computation power as $\hat{R}(\mathcal{L}|D_s)$ is smaller.

In practice, we can observe from Figure 3 that the sample complexity of MEDIC is at most $1/10$ compared with training from scratch on CIFAR-10. This suggests MEDIC can be 10 times faster than training from scratch.

C. Analysis of Why MEDIC Works

In this section, we further analyze why cloning can be more effective against backdoor attacks compared with fine-tuning based approach. To do so, we first define a binary classification problem. We later consider a family of backdoors for this binary classification problem. Through analyzing the classification problem and backdoors, we show that cloning can guarantee the removal of backdoor while fine-tuning may not. Furthermore, we show our importance can better pinpoint the compromised neurons. In the following, we first mathematically define the scenario.

Consider a neural network with two neurons in the penultimate layer. Specifically, the activations of the penultimate layer of neuron networks are written as $z \in \Re^2$. The last layer consists of the fully-connected layer. The binary classification problem is thus represented as $y = \textbf{sgn}(w \cdot z)$ where $w \in \Re^2$ is a part of learn-able parameters.

Let us consider a learning scenario where the internal activations $z = (z_1, z_2)$ follows distribution conditioned on the labels. Specifically, feature z_1 contains a strong feature that determines the results. While the other feature z_2 is a weaker signal but can decide the label as well. We further assume z_1 is independent of the second feature z_2.

Formally, in order to simulate this aforementioned case, we define

$$z_1 \sim \begin{cases} \mathcal{U}(0.1,1), & y=1 \\ \mathcal{U}(-1,-0.1), & y=-1 \end{cases}, z_2 \sim \begin{cases} \mathcal{U}(0,0.1), & y=1 \\ \mathcal{U}(-0.1,0), & y=-1 \end{cases}. \tag{34}$$

The $\mathcal{U}$ represents the uniform distribution. This definition aligns with our description. Note that z_1 is a stronger deciding feature because the margin of z_1 between the positive and negative classes are larger than z_2. Meanwhile z_1 has a larger magnitude than z_2 and thus model are easier to pick up the signal coming from the feature z_1.

Based on this learning scenario, we next introduce a family of backdoor attacks where positive samples are classified into the negative label. Specifically, the backdoor attack can set either feature z_1 or z_2 of the penultimate layer to a constant $k > 0$.

The reason we choose a positive constant k is because of the underlying assumption of backdoors. We assume backdoor patterns are sufficiently different to normal samples. Otherwise, the backdoor attack is nothing but a intended behavior of the neural networks. In this case, The feature of negative samples should be sufficiently different from the feature of the backdoor that results in the negative label. This implies that we instead need a positive k for the backdoor attack. This modeling, where backdoor attack can change either z_1 or z_2 to a constant value, is motivated by the widely-used patch attack which replaces a part of the image with a patch pattern. This introduced family of attacks corresponds to this type of attacks on a linear model.

In the following, we show that the backdoor attack has to modify the less important feature z_2 to launch the backdoor attack. From the fact that the model will correctly predict the benign data, we have

$$(w_1 z_1 + w_2 z_2) y \geq 0$$

By including constraints of $z_1 \geq 0.1$, $z_2 \geq 0$ and $y = 1$, we can infer $w_1 \geq 0$. And thus the backdoor attack must change $z_2 = k$ to launch backdoor attack. Otherwise, $w_1 z_1 > 0$ can not change the label into the negative one.

Since backdoor attack is successful and will predict the backdoor samples as the negative label, we also have

$$(w_1 z_1 + w_2 k) y \leq 0\,.$$

By including the constraints again, similarly, we will have

$$w_2 \leq -\frac{w_1 z_1}{k} \leq -\frac{0.1 w_1}{k}$$

From this result, we can see that w_2 should have a large negative number for a successful backdoor attack.

Meanwhile, to maintain the correctness of negative samples, we shall have these constraints for negative samples

$$\begin{aligned}
&(w_1 z_1 + w_2 z_2) y \geq 0, \\
&w_2 \leq 0 \\
&y = -1 \\
&z_1 \leq -0.1 \\
&z_2 \geq -0.1\,.
\end{aligned}$$

Combining these inequalities and equations, we have

$$w_2 \geq -\frac{w_1 z_1}{z_2} \geq -w_1, .$$

This suggests the weight of w_2 should not be too large to mislead normal classification. In order to satisfy these constraints, we shall set the constant of trigger pattern to be large enough

$$k > 0.1\,.$$

Based on this result, we show that cloning can instead guarantee the removal of the backdoor. And we further show that fine-tuning might not remove the backdoor.

Theorem 3. *Now consider the hinge loss function $l(w) = (1 - yw^T z)_+$ as the classification loss, and the cloning loss $l_{clone}(w) = \lambda(w'^T z - w^T z)^2$, where w' is the weight from backdoor model. For any cloning λ that satisfy $\lambda \leq \mathbb{E}\left[\frac{\mathbb{1}[1 - yw^T z > 0] yz}{(z^T z) w_2'}\right]$, we guarantee the removal of backdoors.*

Proof. The gradient of the classification loss on w is thus negative $\nabla_w l(w) \leq -yz$. Meanwhile the gradient of cloning loss is therefore $\nabla_w l_{\text{clone}}(w) = -2\lambda(w' - w)z^2$. Combined the cloning loss and classification loss, we know that when $\lambda \leq \mathbb{E}\left[\frac{\mathbb{1}[1 - yw^T z > 0] yz}{(z^T z)(w_2' - w_2)}\right]$, we will have a negative gradient on w. Through mathematical induction, we can further relax the nominator as stated in the theorem, for w_2 will be positive. Next we show the negative gradient will guarantee the removal of backdoors. For simplicity, let us assume that we initialize $w = 0$ during cloning. Now we show that during optimization, the cloned model will not have backdoor behaviors. This newly initialized model won't contain backdoor behaviors in this scenario. During cloning, gradient based optimization on this classification loss will always make a positive w_1 and w_2. This means we can guarantee the removal of the backdoor through the unique recipe of training from scratch. However, in the case of fine-tuning, the initialized value of w_1 from backdoor will be a large negative number. In this case, the effectiveness of fine-tuning based methods will unfortunately hinge on how much change is made to w_2 and therefore backdoor removal is not guaranteed. □

Furthermore, let us consider the importance weight from cloning. We calculate the equation as defined in equation 4. We find that activation importance weight is similar for both w_1 and w_2 because of the normalization. Meanwhile, the impact importance weight of w_1 is larger than w_2. By combining these two importance weight together, the importance of w_1 is thus larger than w_2. Given a large enough temperature, the importance for w_1 will be close to 1 while importance of w_2 will be close to 0. This shows that our importance weight correctly identifies compromised neuron w_2 but instead simulates the correct neuron w_1. , cloning will only leverage the activation from important feature w_1.

D. Experiment Details

In this section, we describe the details of our experiments. We use the original code from Badnet [14], Clean Label attack [46], SIG [3], Reflection attack [30], Polygon attack [35], Filter attack [28], and Adaptive attack [26] to construct backdoored models. For backdoor removal methods, we leverage the code from Model Connectivity Repair (MCR) [56], NAD [23], ANP [50], and Fineprune [27]. During the model training and testing, we use the exact same data augmentations, including resizing and cropping. Wide ResNet16 [54] structure and CIFAR-10 [21] dataset are used for Clean Label, SIG, Badnet, Composite, and Reflection attacks. For polygon attack, we use ResNet34 [16] and Kitti-City [11] dataset. For filter attack, we use ResNet34 and Kitti-Road [11] dataset. We utilize 5% of CIFAR-10 training data and 0.5% of Kitti-City and Kitti-Road training data for the experiments. Note that we use a smaller number of data for large-scale datasets, because there are much more training samples in the large-scale datasets.

During cloning, we clone the outputs from all the convolution, normalization, and fully-connected layers in the network structure. These layers have learnable parameters where we aim to copy the functionalities from. We estimate the mean and standard deviation of internal activations using benign data. We use parameter λ to balance the cloning loss $\mathcal{L}_{\text{clone}}$ in Eq.(1) and the classification loss $\mathcal{L}_{\text{classification}}$ (i.e., cross-entropy loss) as follows. We set $\lambda = 10$ in this paper. We also conduct an ablation study on λ in Appendix E.3.

$$\mathcal{L}(x, y) = \mathcal{L}_{\text{classification}}(x, y) + \lambda \mathcal{L}_{\text{clone}}(x, y)$$

In the experiment, we train the model for 60 epochs over the small set of clean data. We use Adam optimizer with a initial learning rate $1\text{e}{-2}$ and apply weight decay of $1\text{e}{-4}$. We align the temperature τ in our method so that the clean accuracy of our model is comparable to others. We conduct the experiments on four GTX 2080 GPUs.

During backdoor removal, we assume we have no knowledge of the type of backdoors injected in the model. We directly report the performance of the model at the last optimization step. Our reported results are slightly worse than those reporting the best model during training.

D.1. Algorithm

In Algorithm 1, we show the complete procedures. Specifically, we first select the neurons of interest, which comprises of the output from convolution, normalization, and dense layers. We then use the available samples to estimate the mean and variance of corresponding neurons. As the available samples come from the same distribution as the original training data, the estimation of mean and variance converges exponentially. The fast convergence means the number of samples can be quite small for an accurate estimation. We compute the weight based on the equations in line 5. In line 6, we incorporate the weight to cloning loss function. In lines 7-9, we use standard optimization with the loss function.

E. Additional Experiments

E.1. Additional Adaptive Attack

Table 2. The evaluation on an additional adaptive attack.

	Baseline	FinePrune	NAD	MCR	ANP	MEDIC
ASR	97.3	81.6	74.0	93.8	68.9	**2.7**
Acc.	86.9	85.4	85.6	84.7	82.0	84.7

In this section, we introduce another type of adaptive attack that may constitutes a good baseline. We show that MEDIC outperforms others by a large margin. [41] shows that reducing the difference between benign samples and backdoor ones may make backdoor attack stronger. In this experiment, we implement the attack from [41] that minimizes the internal l_2 distance between benign and malicious samples for all layers, and set the fine-tuned penalty for l_2 distance to 1e-4. Experiments are

Algorithm 1 MEDIC Cloning Procedure

Input : Dataset with a small number of clean samples D_s, training epochs T, number of batches at each epoch L, neurons from the teacher model for cloning f_i^* and the corresponding student neurons f_i^c.

Output : Sanitized Model f^c

$f^c \leftarrow$ `RandomInit()`

Estimate σ_i, μ_i of activation $f_i^*(x)$ on input data $x \in D_s$

for $b \leftarrow 1$ **to** $T \cdot L$ **do**

  Draw a batch of data from $B \in D_s$

  Calculate w based on Equations 2, 3 and 4

  $\mathcal{L}_{\text{clone}} = \sum_{(x,y)\in B} \left[\sum_i w_i(x,y) \left(\frac{f_i^*(x) - f_i^c(x)}{\sigma_i} \right)^2 \right]$

  $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classification}} + \lambda \mathcal{L}_{\text{clone}}$

  Update f^c with $\nabla_{f^c} \mathcal{L}_{\text{total}}$

  Adjust the learning rate based on the scheduler

conducted on the same CIFAR-10 setting as in our paper. The results can be found in Table 2. It shows MEDIC is quite effective, having 65% lower ASR than baselines. This is due to our unique design including the importance criteria and training from scratch as explained in

E.2. Evaluation on Backdoor Attacks under Distribution Shift

In this section, we further stress test our method by evaluating in a more challenging scenario. In this setting, data augmentations are not used during backdoor attacks. They however are applied during backdoor removal. As data augmentations shift the distribution of training data, it violates the assumption of MEDIC that available training data during backdoor removal are sampled from the same distribution. We use the same setup as in Appendix D and conduct the experiments on CIFAR-10 and Wide ResNet. We evaluate on backdoor attacks SIG, BadNet, and CleanLabel. We do not include Reflection and Warp attacks as data augmentations are essential for the success of these attacks.

The results are shown in Table 3. Observe that the ASRs are much lower than those in Table 1. This is because these attacks are less robust if no data augmentation is leveraged during the attack. We can see MEDIC has the best performance on hard-to-remove backdoors (e.g, CleanLabel attack that uses adversarial training), which is consistent with the results of using data augmentations during the attack. For other attacks, the results of MEDIC are comparable to those of baselines. Compared to the results obtained under the same distribution (during attack and removal), we find MEDIC has slightly worse performance under different distributions. The attack success rates of SIG and BadNet are higher. MEDIC has a better result on CleanLabel attack compared to Table 1 as the attack is less robust.

Table 3. Comparison with baselines without data augmentation. ± represents the standard deviation over 5 repeated runs.

Attack	Metric	Original (%)	Method (in percentage %)					
			Finetune	Fineprune	NAD	MCR	ANP	MEDIC
CleanLabel	ASR	100	11.6±1.7	11.2±1.5	8.2±1.6	5.9±0.2	7.8±4.4	**5.1±0.7**
	ACC	83.2	81.1±0.1	81.3±0.3	80.9±0.3	80.6±0.1	75.2±1.9	80.7±0.2
SIG	ASR	97.8	3.5±1.6	4.0±1.7	4.7±1.1	**0.5±0.2**	3.9±2.6	5.1±1.0
	ACC	83.5	81.8±0.2	82.1±0.1	82.0±0.1	80.9±0.3	77.7±0.9	79.5±0.1
BadNet	ASR	99.8	3.2±0.3	3.6±0.5	4.1±0.4	**2.9±0.1**	6.2±2.6	3.6±0.5
	ACC	81.9	77.5±0.1	79.6±0.4	75.3±1.1	78.3±0.2	73.4±0.7	79.9±0.2

E.3. Ablation Study on λ

In algorithm 1, we introduce a variable named λ to combine the cross entropy loss and the cloning loss. Specifically, we have

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classification}} + \lambda \mathcal{L}_{\text{clone}}$$

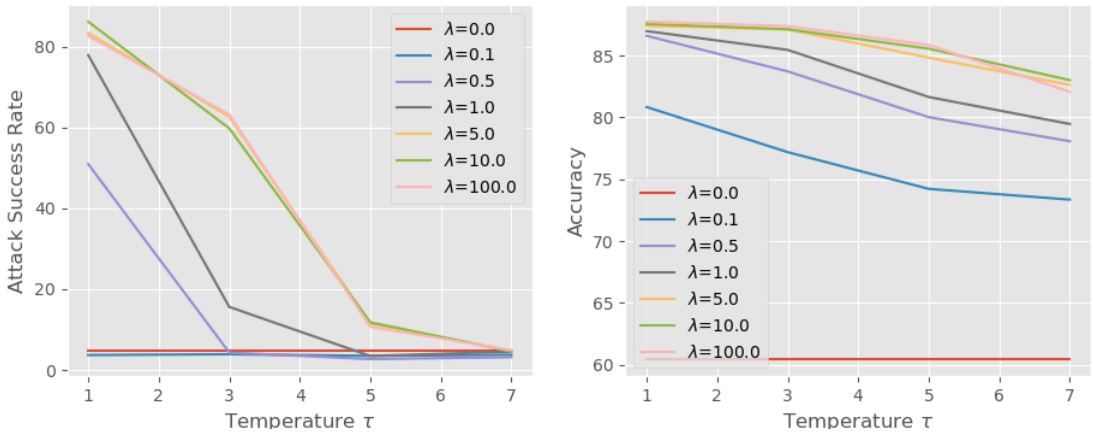

Figure 6. Effect of λ. The experiment is conducted on CIFAR-10 and CleanLabel attack.

In this section, we conduct an ablation study of how λ will impact the performance of cloning. We use the CleanLabel attack on CIFAR10. In figure 6, we show the result of cloning with different choices of λ. The x-axis indicates the temperature. The y-axis indicates the ASR (in the left figure) and the clean accuracy (in the right figure).

A large λ means more focus on the cloning loss and less focus on the classification loss. According to our study, enlarging λ can increase the clean accuracy of the cloned model as shown in the right figure. At the same time, the attack success rate also increases (see the left figure). To reduce the ASR, we need to simultaneously increase the temperature. By increasing both λ and the temperature, we can achieve better clean accuracy as well as ASR.

	Clean Label	SIG	BadNet	Adaptive	Reflection
5% Data ASR	16.8±4.6	1.5±0.7	3.6±0.6	7.1±1.7	6.2±0.5
5% Data Acc.	85.3±0.2	84.4±0.3	84.2±0.2	79.7±0.1	83.5±0.2
10% Data ASR	6.9±1.2	0.5±0.5	3.2±0.5	5.6±0.8	4.0±0.4
10% Data Acc.	85.4±0.3	84.6±0.2	86.8±0.1	80.9±0.2	84.0±0.3

Table 4. an Ablation Study on the Amount of Data

E.4. Ablation Study on Amount of Available Data

In table 4, we conduct an ablation study on the amount of available data during our backdoor removal. We adopt CIFAR-10 for the experiments and test on 5% and 10% available data. The results show that the performance of trigger removal is positively correlated with the amount of available data.

E.5. Empirical Complexity

The computation cost is approximately proportional to the training epochs. We compare the training epochs of different methods on CIFAR-10. The results show that MEDIC is within the same order of magnitude as the baselines.

	Finetune	Fine-prune	NAD	MCR	ANP	MEDIC
Training (Epochs)	30	31	40	240	100	60

E.6. Ablation Study on Importance Criterion

In this section, we conduct the ablation study on different importance criteria. We show the combination of both impact (in eq.(3) and activation (in eq.(2) criteria is beneficial to the backdoor removal. We conduct the study on CIFAR-10 and Clean Label attack. We repeat the experiments under different criteria and different τ from 1 to 7. For each criterion or their

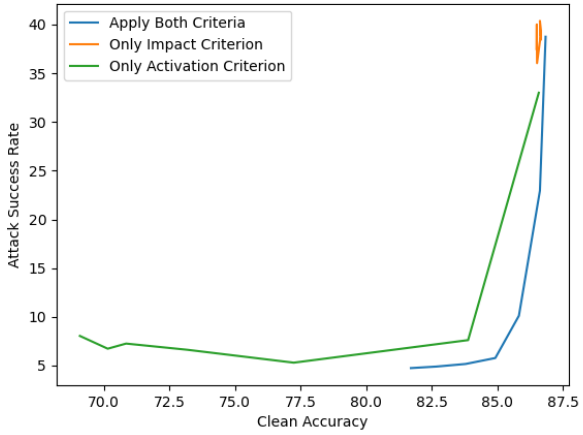

Figure 7. Effect of Combinations of criteria. The experiment is conducted on CIFAR-10 and CleanLabel attack.

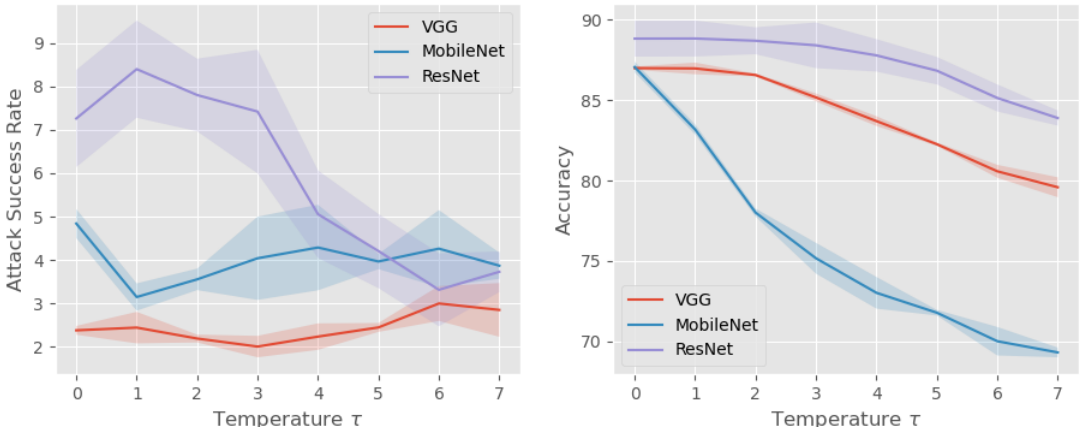

Figure 8. Effect of Model Architecture. The experiment is conducted on CIFAR-10 and BadNet attack.

combination, we draw the trade-off curve between attack success rate and clean accuracy. We repeat 5 times for each point and use the average value for the report. Figure 7 shows the results. X-axis represents the clean accuracy and y axis represents the attack success rate. A curve close to the lower-right corner indicates a better performance.

From Figure 7, we can observe the orange curve concentrates on the upper-right corner. It means that with only impact criterion, the clean label attack can not be completely removed. The reason is that clean label attack involves adversarial training which makes the backdoor attack quite robust. Therefore benign features and backdoor features are somewhat activated simultaneously. They won't be separated by this single criterion. However by adding additional impact criterion, those backdoor features will be excluded during cloning. Observe that the green curve has much better attack success rate than the orange one. Furthermore, we can observe that by including both criteria , we have the blue curve. It has the best trade-off between accuracy and success rate.

E.7. Ablation Study on Model Architecture

In this section, we further study the impact of model architecture. We train the same backdoor attack on three different types of model architectures, including VGG-11, MobileNet-V1 and ResNet-16. The experiment is conducted on CIFAR-10 and BadNet. Specifically, model VGG contains dropout layers. We found including dropout layers during cloning will make optimization harder to converge. We therefore remove the additional dropout layers since in theory the expected output will be similar. we use a learning rate 1e-3 for this experiment to make sure optimization convergence on different models architectures. .

Figure 8 shows the results. The y-axis represents attack success rate and clean accuracy respectively. The x-axis represents different temperatures. Observe that cloning is effective in both three very different model architectures. Specifically, we can reduce the attack success rate of all three models below 5%. Moreover, we find the temperature actually has different impact over different architectures. Specifically, the MobileNet has the faster reduction in clean accuracy as τ grows.