Noise-response Analysis for Rapid Detection of Backdoors in Deep Neural Networks

N. Benjamin Erichson\textsuperscript{1} * Dane Taylor\textsuperscript{2} * Qixuan Wu\textsuperscript{1} Michael W. Mahoney\textsuperscript{1}

\textsuperscript{1}ICSI and Department of Statistics at UC Berkeley
\textsuperscript{2}Department of Mathematics at University at Buffalo, SUNY

\textbf{ABSTRACT}

The pervasiveness of deep neural networks (DNNs) in technology, matched with the ubiquity of cloud-based training and transfer learning, is giving rise to a new frontier for cybersecurity whereby ‘structural malware’ is manifest as compromised weights and activation pathways for unsecure DNNs. In particular, DNNs can be designed to have backdoors in which an adversary can easily and reliably fool a classifier by adding to any image a pattern of pixels called a trigger. Since DNNs are black-box algorithms, it is generally difficult to detect a backdoor or any other type of structural malware. To efficiently provide a reliable signal for the absence/presence of backdoors, we propose a rapid feature-generation step in which we study how DNNs respond to noise-infused images with varying noise intensity. This results in titration curves, which are a type of ‘fingerprinting’ for DNNs. We find that DNNs with backdoors are more sensitive to input noise and respond in a characteristic way that reveals the backdoor and where it leads (i.e., its target). Our empirical results demonstrate that we can accurately detect a backdoor with high confidence orders-of-magnitude faster than existing approaches (i.e., seconds versus hours). Our method also yields a titration-score \( T_{\gamma} \in [0, 1] \) that can automate the detection of compromised DNNs, whereas existing backdoor-detection strategies are not automated.

\textbf{Keywords:} Deep learning, AI safety, backdoor attacks, noise response analysis.

\textbf{1 INTRODUCTION}

While deep neural networks (DNNs) are ubiquitous for many technologies that shape the 21st century, it is well known that they are susceptible to various forms of non-robustness and adversarial deception. Among other things, this gives rise to new fronts for cyber and data warfare. Such robustness and related security concerns abound in relation to adversarial attacks \([15, 38]\) and fairness in machine learning \([4, 10]\). This poses an increasing threat as machine learning methods become more integrated into mission-critical technologies, including driving assistants, face recognition, machine translation, speech recognition, and robotics.

Recently, so-called backdoor attacks have emerged as a crucial security risk for certain machine learning applications \([16, 8, 27, 28, 2, 35, 40, 45, 24]\). In these cases, an adversary can modify a DNN’s architecture either by polluting the training data \([16, 8]\) or by changing the model weights \([27, 28]\), and then return a “backdoored model” to the user. This threat scenario is plausible, since an adversary can have full access to a DNN, e.g., if it is outsourced for training due to infrastructure availability and resource costs (both human and computational).

Backdoors are difficult to detect because they are subtle “Trojan” attacks: a backdoored model behaves perfectly innocently during inference, except in situations where it is presented with an input example that contains a specific trigger, which activates an (unknown) adversarial protocol that can mislead DNN predictions, potentially with severe consequences. In contrast, classical data-pollution attacks aim to reduce a classifier’s accuracy, and the poor performance is more easily detectable. Thus, it is of great importance to develop fast and reliable metrics to detect compromised DNNs that contain backdoors. While several defense methods have been proposed \([5, 26, 39, 14, 6, 41]\), all of them have significant limitations such as requiring access to labeled data and/or the triggered training data, having prior knowledge about the trigger, or using massive computational resources to train DNNs and perform many adversarial attacks. In contrast, we will present an efficient approach without such limitations; we detect backdoors and triggers for modern DNNs, including ResNets, in just seconds (as opposed to hours \([6, 41]\)). Moreover, unlike existing studies on backdoor attacks, our approach yields a score \( T_{\gamma} \in [0, 1] \) that indicates the absence/presence of a backdoor, which provides a major step toward automat-
We add noise \( \eta \) with variance \( \sigma \) to an input image \( x \), and the red and blue curves show the fraction \( T(x, \theta) \) [Eq. (3)] of noisy images that yield high-confidence predictions, \(|| \hat{y}(x + \eta) ||_\infty > \gamma \) (i.e., there is an activation in the final layer that is greater than \( \gamma \in [0, 1] \)). (b) Perturbation analysis describes how the \( k \)-th logit \( Z_k(x, \theta) \) nonlinearly responds to small-intensity input noise that is added to each image data point \( x_{ij} \). (Implicit) gradients \( \frac{\partial Z_k(x+\eta, \theta)}{\partial x_{ij}} \) [see Eq. (7)] are computed after adding noise and reveal pixels that are associated with the trigger.

(a) Titration analysis reveals that different patterns for noise-induced misclassifications distinguish baseline and backdoored models. We develop two complementary noise-response analyses to detect a backdoor: titration analysis (see Fig. 1a and Sec. 3.2) and perturbation analysis (see Fig. 1b and Sec. 3.3). In Fig. 1a, we present titration curves that depict a titration score (defined below) versus noise intensity \( \sigma \). The curves reveal that backdoored models respond to noise very differently than baseline models, which allows us to rapidly detect which models are compromised. In particular, observe that DNNs with a backdoor respond much more strongly to noise than those without them; they are more brittle to noise. Moreover, we observe that a backdoors’ target class \( k^* \) acts as a “sink” that induces high-confidence, noise-induced predictions.

In Fig. 1b, we illustrate the sensitivity of neuron activations in the final layer before softmax (we will further refer to these as logits) to input noise for each individual input image pixel. Observe in the third and fourth columns that the logits are more sensitive to noise for the pixels associated with a backdoor’s trigger (in this case, a 3 \( \times \) 3 patch in the lower-right corner). In other words, neurons in the final layer have a stronger nonlinear response to the pixels associated with the trigger.

Here is a summary of our main contributions:

(a) We develop a noise-induced titration procedure yielding titration curves that provide a fingerprinting for DNNs.

(b) We propose a titration score \( T^*_\sigma \) to express the risk for a DNN to have a backdoor, thereby allowing the automation of backdoor detection.

(c) We develop a perturbation analysis to study how the output of a DNN nonlinearly responds to small-intensity input noise.

(d) We propose an implicit gradient map to reveal which image pixels are associated with a backdoor’s trigger.

Overall, we present a methodology that accurately detects backdoors for modern DNNs, including ResNets, in just seconds (as opposed to hours, for other related methods). Because our aim is to detect backdoors, as opposed to design them, we focus our attention on backdoor attacks that have been previously studied. Certainly, with time, backdoor attacks will evolve, e.g., in response to detection techniques such as that which we propose here, leading to an arms-race analogous to that of traditional malware settings. However, we believe that our general framework (that is, analyzing DNNs by probing them with input noise) is sufficiently adaptable to significantly contribute to this pursuit; we also expect noise-response analysis to find future applications beyond backdoor detection.
2 RELATED WORK

The sensitivity and non-robustness of DNN models to adversarial environments are an emerging threat for many problems in safety- and security-critical applications, including medical imaging, surveillance, autonomous driving, and machine translation. The most widely studied threat scenarios can be categorized into evasion attacks [38, 15], data poisoning attacks [3, 36] and backdoor attacks [16, 8]. Evasion attacks have received the most attention and involve fooling a model into making erroneous predictions by adding an undetectable adversarial perturbation to an input image. While adversarial examples are very effective, it is debatable whether evasion attacks are a significant threat in many real-world applications [29, 20, 43]. In particular, the effectiveness of black-box evasion attacks is often inferior; however, strong evasion (i.e., white-box) attacks require access to the model, and the crafted adversarial pattern usually affects only a small set of images.

On the other hand, backdoor attacks pose a realistic threat due to the fact that it is a common practice for research labs and government agencies to outsource the training of DNNs and to use pre-trained 3rd-party networks via transfer learning [46]. This provides potential adversaries with access to machine learning pipelines that may affect mission-critical applications.

Herein, we focus on the most common scenario of targeted backdoor attacks [16, 8, 27, 28]. Let \( x \) denote an image from class \( k(x) \in \{0, \ldots, K-1\} \), which we 1-hot encode by \( y \in \{0,1\}^K \) so that \( k(x) = \text{argmax}(y) \). Now, consider a DNN classifier defined by a nonlinear transfer function

\[
\hat{y} = \text{softmax}(Z(x, \theta)),
\]

where \( \theta \) denotes edge weights and \( Z(x, \theta) \) is the vector of logits (i.e., output of the DNN before applying softmax). We further define \( \hat{k}(x) = \text{argmax}(\hat{y}) \) as the predicted class of \( x \).

A DNN is said to have a targeted backdoor if there exists a trigger \( \Delta x^* \) and a target class \( k^* \in \{0, \ldots, K-1\} \) such that

\[
\hat{k}(x + \Delta x^*) = \text{argmax}(\hat{y}) = k^*,
\]

regardless of \( \hat{k}(x) \). That is, an adversary can redirect the predicted class label for any input image to a particular \( k^* \) simply by adding an adversary-designed trigger \( \Delta x^* \) to that input image. We refer to such a trigger as a universal trigger.

2.1 Attack Strategies

There are currently numerous strategies to implement effective backdoors that achieve a nearly 100% success rate at redirecting triggered images to the target class, while also minimally affecting the prediction accuracy of non-triggered images. One approach is to directly change the weights of a pre-trained model backdoor [27]. While this approach does not require access to the original data, it requires great deal of sophistication.

The most common approach, however, is to train a DNN with a poisoned dataset in which some images have the trigger and their classes are changed to the target class \( k^* \). Gu et al. [16] and Chen et al. [8] explore several types of triggers (see Fig. 2), which are added to a small number of images, which are then mixed into the training data before training a model. Turner et al. [40] propose a more sophisticated watermark pattern as a trigger, which is crafted by using a GAN-based (Generative Adversarial Network) interpolation scheme. Many more variants for crafting triggers have been proposed [25, 24, 45, 35, 24].

2.2 Defense Strategies

Leading methods to defend against backdoors include SentiNet [9], Activation Clustering [5], Spectral Signatures [39], Fine-Pruning [26], STRIP [14], Meta Neural Analysis [44], DeepInspect [6] and Neural Cleanse [41]. These techniques often involve three steps (detect if a model is backdoored; identify and re-engineer the trigger; and mitigate the effect of the trigger), which can implemented sequentially as distinct pursuits or simultaneously as a single pursuit. (We adopt the prior strategy.)

A common limitation for existing defense methodologies [9, 5, 39, 26, 14, 6, 41] is that they require the training of a new model to probe the DNN under consideration. This leads to very high computational overhead and requires a certain level of expertise. In particular, Neural Cleanse [41] takes about 1.3 hours to scan a DNN. DeepInspect [6] reduces the computational costs by a factor of 4-10, while improving the detection rate. Nevertheless, DeepInspect requires the training of a specialized GAN for probing, and thus the overall costs can outweigh the training costs for the actual classifier.

Importantly, there is no previous research that rapidly detects backdoors. Thus motivated, we now propose a fundamentally different approach that detects backdoors in a few seconds or less.

![Example triggers that can be added to an image to activate an adversarial protocol/malware that redirects classifiers’ prediction to a target class \( k^* \). Unlike adversarial attacks, the trigger is fixed and can be applied to any input.](image-url)
3 Noise-response Analysis

Noise-response analysis has long been a valuable tool for studying nonlinear dynamical systems [34, 32, 11]. Leading techniques to measure the presence and extent of chaos study the effect of noise to estimate a dynamical system’s correlation dimension and largest Lyapunov exponent [34]. The robustness of a dynamical system to noise is also central topic with a large literature grounded on KAM theory [11]. Such methods involve perturbation analysis and focus on the small-noise regime, yet it is also insightful to study larger noise intensity. More generally, one can study how a dynamical system responds to an increasing noise intensity via a titration procedure. In particular, previous research [32] used similar noise-induced titrations to identify whether black-box dynamical systems were chaotic or stochastic.

We propose to use titrations and perturbation analyses as complementary techniques to obtain an expressive characterization for the nonlinearity of a DNN’s transfer function, thereby allowing us to efficiently detect and study backdoors. Let $x = [x_{jc}]$ and $Z(x, \theta)$ denote, respectively, the inputs and outputs (i.e., logits before applying softmax) for a DNN with parameters $\theta$. We denote an entry of the logits vector $Z(x, \theta)$ by $Z_k(x, \theta)$, which gives the activation of the neuron associated with class $k$. For each colored pixel $x_{jc}$, we add i.i.d. normal-distributed noise $\eta_{jc} \sim \mathcal{N}(0, 1)$, which we scale by $\sigma > 0$ so that $\sigma \eta_{jc} \sim \mathcal{N}(0, \sigma^2)$. (The motivation for this notation will be apparent below, when we present our perturbation theory.) Letting $\eta = [\eta_{jc}]$ denote a tensor of noise, it follows that $Z_k(x + \sigma \eta, \theta)$ denotes the $k$-th logit for a noisy image $x + \sigma \eta$. We study how a DNN nonlinearly transforms an input distribution (i.e., noise) to an output distribution. For each $k \in \{0, \ldots, K-1\}$, we let $P_k^{(\sigma)}(x, z) = P_k(x + \sigma \eta, \theta)$ denote the probability of observing a logit $Z_k(x + \sigma \eta, \theta) = z$ for image $x$ with noise variance $\sigma^2$. We also allow the input images to be sampled from some distribution, $x \sim P_i(x)$, and the integral $P_k^{(\sigma)}(z) = \int_x P_k^{(\sigma)}(x, z) P_i(x) dx$ gives the overall distribution of $Z_k(x + \sigma \eta, \theta)$ for a given $\sigma$.

3.1 Pedagogical Example

We start with an experiment to identify key insights for how the outputs of DNNs nonlinearly respond to input noise, which is very different for baseline and backdoored models. In particular, input noise is amplified for a backdoor’s target class $k^*$, allowing its detection. To improve the clarity of this presentation, we focus in this pedagogical example on a simple model (LeNet5 [23]) and a simple dataset (MNIST-4, which we created as a subset of MNIST [22] with characters $k \in \{0, 1, 2, 3\}$). The backdoor was implemented using the approach of [16, 8] with a trigger $\Delta x^*$ (in this case, a 3x3 patch of weight-1 pixels in the lower-right corner) that was added to 10% of the training images, redirecting their predicted label to a target class $k^* = 0$.

In Fig. 3, we present an experiment illustrating how the distributions $P_k^{(\sigma)}(z)$ differ between baseline (left column) and backdoored (right column) DNNs. We restrict our attention to the distribution of logits $Z_k(x + \sigma \eta, \theta)$ arising for images from class $k = 3$ by setting $P_i(x)$ to be a uniform distribution with compact support over images such that $k(x) = 3$. In Fig. 3a, we show logit distributions $P_k^{(0)}(z)$ for class-3 images that are “clean” in that no triggers are added to the images. The rightmost-distributions indicate that both the baseline and backdoored DNNs
correctly predict each image’s class, \( \hat{k}(x) = 3 \). In Fig. 3b, we also show \( P_k^{(0)}(z) \), except we add the trigger \( \Delta x \) to each image. The rightmost distributions in (b) show that these triggered images are correctly classified by the baseline model, \( \hat{k}(x + \Delta x) = 3 \), but the backdoored DNN redirects their predicted labels to the target class, \( \hat{k}(x + \Delta x) = 0 \). Fig. 3c and 3d depict \( P_k^{(\sigma)}(z) \) for \( \sigma = 1 \) and \( \sigma = 10 \), respectively. Observe that the backdoored DNN is more strongly affected by noise than the baseline DNN. Moreover, for the larger \( \sigma \) value there are logits \( Z_k(x + \sigma \eta, \theta) \) for the target class that become very large.

In Fig. 4, we provide another visualization for how increasing \( \sigma \) affects the logits \( Z_k(x + \sigma \eta, \theta) \). We study the same (a) baseline and (b) backdoored DNNs as in Fig. 3. In both panels, we visualize the logits \( Z(x + \sigma \eta, \theta) \in \mathbb{R}^4 \) for images \( x \) from all classes, and we project these points onto \( \mathbb{R}^2 \) using PCA. We also randomly choose an image from classes 1, 2, and 3, and we plot an empirical estimate for \( \mathbb{E}[Z_k(x + \sigma \eta, \theta)] = \int_z z P_k^{(\sigma)}(z) dz \) while varying \( \sigma \) from 0 (red) to \( \sigma = 10 \) (blue). These paths can be interpreted as random walks in a low-dimensional eigenspace, and we average over 200 such walks. Observe that the noise has little effect for the baseline model. The target class \( k^* = 0 \) essentially attracts predictions as \( \sigma \) increases.

### 3.2 Titration Analysis

Titration analysis involves studying the dependence of a system on a titration parameter. In our case, we study the response of a DNN’s output with input noise with standard deviation \( \sigma \) (i.e., the “titration parameter”). A common strategy involves constructing titration curves that provide informative and expressive signals. Based on our previous experiments, we propose to study the fraction of noisy images \( x + \sigma \eta \) whose predictions \( \hat{y}(x + \sigma \eta) \) are high-confidence,

\[
T_\gamma^\sigma\text{-score} = \frac{|\{x: ||\hat{y}(x + \sigma \eta)||_\infty > \gamma\}|}{|\{x\}|} \in [0, 1].
\]

We interpret the maximum output activation, or \( L_\infty \) norm, as a notion of confidence, and we distinguish high- and low-confidence predictions via a tunable threshold \( \gamma \in [0, 1] \). See Fig. 1a for example titration curves for baseline and backdoored ResNets for CIFAR-10. Note that the curves are different: for the backdoored model, \( T_\gamma^\sigma\text{-score} \) rapidly grows to 1 with increasing \( \sigma \), whereas it slowly grows for the baseline model. We choose the \( T_\gamma^\sigma\text{-score} \) to construct titration curves because Fig. 4 revealed the targeted class \( k^* \) to be a “sink” for the predicted labels of noisy images. We additionally find these predictions to have high confidence, which is a signature that we empirically observe only occurs for backdoored models.

In Fig. 5, we provide a visualization to help illustrate why the \( T \)-score is so effective at differentiating backdoored and baseline models; for backdoored models, noise leads to logits that are very large, which yields high-confidence predictions. Each row in Fig. 5 shows a circle plot in which each class is represented by a colored section of a ring. Within the circle, logits \( Z_k(x + \sigma \eta, \theta) \) for each noisy image \( x + \sigma \eta \) is represented by a point in polar coordinates. The radius \( R(x + \sigma \eta) = \max_k Z_k(x + \sigma \eta, \theta) \) is the maximum logit, which we interpret as a notion of confidence. The angle is given by \( \psi_k(x + \sigma \eta) = \frac{k}{2\pi K} \), where \( \hat{k}(x + \sigma \eta) \) is the predicted label for noisy image \( x + \sigma \eta \) and \( K = 10 \) is the number of classes. (In the plots, we additionally add a small amount of noise to each angle \( \psi_k(x + \sigma \eta) \) so that the points spread out and can be seen.) In each row, we show several circle plots for increase input noise variance \( \sigma \). These circle plots complement the construction of a titration curve, which can be thought of as a scalar summary of these plots. Note for the backdoored models that as \( \sigma \) increases, the points \( \{R(x + \sigma \eta), \psi_k(x + \sigma \eta)\} \)
not only move toward the target class \( k^* \), but the logits become very large so that \( R_k(x + \sigma \eta) \approx 1 \) (i.e., high-confidence) for each image. This phenomenon does not occur for the baseline models. We point out that for a wide range of \( \gamma \) and \( \sigma \) values, one can interpret \( T^D \) as a scalar measure for the risk that a DNN has a backdoor.

### 3.3 Perturbation Analysis

Here, we study the local sensitivity of each logit \( Z_k(x, \theta) \) to each in-layer neuron, \( x_{ijc} \). We present a linear analysis that is asymptotically consistent for the limit of small perturbations. Consider the gradients

\[
g^{(k)}_{ijc}(x) = \frac{\partial Z_k(x, \theta)}{\partial x_{ijc}}. \tag{4}\]

Fortunately, these can be efficiently computed using the built-in automatic differentiation of modern deep-learning software packages by defining \( Z_k(x, \theta) \) as a temporary loss function. For a given perturbation \( \Delta x \), we scale it by perturbation parameter \( \sigma \geq 0 \) and Taylor expand to obtain a first-order approximation

\[
Z_k(x + \sigma \Delta x, \theta) \approx Z_k(x, \theta) + \sigma \sum_{ijc} g^{(k)}_{ijc}(x) \Delta x_{ijc}. \tag{5}\]

Let \( \Delta Z_k = Z_k(x + \sigma \Delta x, \theta) - Z_k(x, \theta) \) denote the change of the \( k \)-th logit. For a perturbation with entries \( [\Delta x]_{ijc} = \sigma \eta_{ijc} \), that are drawn as i.i.d. noise with variance \( \sigma^2 \), we use the linearity of Eq. (4) to obtain the expectation and variance of the first-order approximation,

\[
\mathbb{E}[\Delta Z_k] \approx \sigma \sum_{ijc} g^{(k)}_{ijc} \mathbb{E}[\eta_{ijc}] = 0
\]

\[
\mathbb{V}[\Delta Z_k] = \sigma^2 \sum_{ijc} \left(g^{(k)}_{ijc}(x)\right)^2. \tag{6}\]

We numerically validate these results in Fig. 6, where we compare observed and predicted values for the standard deviation, \( \mathbb{V}[\Delta Z_k]^{-1/2} \). Colored curves denote empirical estimates for different values of \( \sigma \), whereas the black lines represent the prediction given by Eq. (6), i.e.,

\[
\left[ \sum_{ijc} \left(g^{(k)}_{ijc}(x)\right)^2 \right]^{-1/2}.
\]

For sufficiently small \( \sigma \), a logit’s change \( \Delta Z_k \) has a linear response that is well-predicted by our theory. Therefore, the expected perturbation of each logit is zero in the small-\( \sigma \) limit, regardless of the image \( x \). This implies (as one may have guessed) that the “sink” phenomenon shown in Fig. 4 is strictly a nonlinear effect.

We investigate the nonlinear response of each \( Z_k(x + \sigma \eta, \theta) \) to perturbations \( \sigma \eta \sim \mathcal{N}(0, \sigma^2) \) by constructing a Taylor expansion around a noisy image \( x + \sigma \Delta x \), as opposed to the clean image. We obtain an approximation that is nearly identical to Eq. (5), except that one uses the gradients \( g^{(k)}_{ijc}(x + \sigma \eta) \) of noisy images. If one interprets a DNN’s transfer function as a step of a numerical ODE integrator [7], then Eq. (6) corresponds to an (explicit) forward Euler step, whereas

![Figure 5: Circle plots for baseline and backdoored versions of LeNet5 trained on MNIST.](image)

The angle \( \psi_k(x + \sigma \eta) \in [0, 2\pi) \) of each point indicates the predicted class \( \hat{k}(x + \sigma \eta) \) of each noisy image \( x + \sigma \eta \), and the radius \( R(x + \sigma \eta) \in (0, 1] \) indicates the level of confidence (indicated by the maximum logit). As one increases the input noise intensity \( \sigma \) (left to right), one observes high-confidence predictions for the target class \( k^* = 3 \) of the backdoored model.
this second approximation corresponds to an (implicit) backward Euler step. This implicit estimate provides us with a small-$\sigma$ estimate for the distributions of logits, $P^{(\sigma)}_{k}(z)dz \approx \mathcal{N}\left(0, \sigma^{2}\sum_{j:c} g_{k}^{(j)}(x + \sigma\eta)^{2}\right)$. However, we are more interested in the nonlinear properties of distributions $P^{(\sigma)}_{k}(z)$. To this end, we examine an extremal summary statistic for $P^{(\sigma)}_{k}(z)$,

$$\bar{g}_{ij} = \max_{k,c} g_{ij}^{(k)}(x + \sigma\eta).$$

(7)

In Fig. 1b, we provide a visualization of $g$, which we call an implicit gradient map. Observe that large values provide a signal for the pixels associated with the backdoor’s trigger. In principle, one could empirically study other distributional properties to obtain signals for the local nonlinearity caused by backdoors.

4 Experimental Results

4.1 Experimental Setup

We consider several standard datasets such as MNIST [22], CIFAR10 [21] and CIFAR100 [21], and state-of-the-art network architectures for our experimental evaluation. More concretely, we use a LeNet5 architecture [23] to train on MNIST. For CIFAR10, we consider ResNets [19] with depth 18 and a WideResNet [47] with depth 30 and a width factor of 4. For CIFAR100, we use the same WideResNet architecture as for CIFAR10 as well as a standard PyramidNet [17] with depth 200.

We use a small square trigger of size $3 \times 3$ (placed at the bottom right corner). The trigger is placed so that the trigger success is not affected by data transformations such as random crop. We choose the number of triggered images so that the trigger success rate is nearly 100%, and usually a small fraction (i.e., $< 5\%$) of examples is sufficient. Recall that trigger intensity refers to the numerical values that are added an image’s RGB values. That is, in order to add a trigger $\Delta x^c$ to an image $x$, we assume the trigger is binary, i.e., $[\Delta x^c]_{dc} \in \{0, 1\}$, and we train on a triggered image $x + \alpha \Delta x^c$, where $\alpha$ is the trigger intensity. We clip pixel intensity values that are not within the range of the pixel values.

4.2 Experimental Evaluation

In Fig. 7, different panels depict titration curves for the different models and datasets. All panels resemble Fig. 1a in that the baseline and backdoored models have characteristic shapes: titration curves of backdoored models rapidly increase with $\sigma$, whereas they slowly increase for baseline models. Interestingly, the sudden rise in $T^{(\sigma)}_{\gamma}$-scores for small-but-increasing $\sigma$ is less pronounced for the PyraMidNet trained on CIFAR-100 with target class $k^* = 3$, but not $k^* = 53$ (compare Figs. 7e and 7f). Note that there are four curves in each panel: the light-colored curves and symbols depict $T^{(\sigma)}_{\gamma}$-scores when noise is added to an actual image $x$, whereas the bright-colored curves and symbols are for “pure” white noise. We observe that the $T^{(\sigma)}_{\gamma}$-scores are nearly identical for these two approaches, but the latter approach does not require any data. The titration curves provide us also with guidance for choosing a advantageous $\sigma$ values for computing the $T^{(\sigma)}_{\gamma}$-score, and moreover it can be seen that the $T^{(\sigma)}_{\gamma}$-score is relatively insensitive to the choice of $\gamma$. Hence, these parameter choices can be more effectively selected by studying the $T^{(\sigma)}_{\gamma}$-scores for a range of $\sigma$ and $\gamma$ values.

We also highlight that our approach is more efficient than existing methods for backdoor detection [41, 6, 5]. For example, the approach proposed by [6] requires one to train a GAN first, which is then used to probe the network. Nevertheless, detecting a backdoor can take up one hour even if a pre-trained GAN can be used.

In Table 1, we provide a summary of results for additional experiments, which highlight that a single titration score $T^{(\sigma)}_{\gamma}$-score suffices to accurately detect backdoored models. The $T^{(\sigma)}_{\gamma}$-scores were computed with pure white noise, and our choices for $\sigma$ were informed by Fig. 7. Observe that the $T^{(\sigma)}_{\gamma}$-score is very high ($> 0.75$) for most models (with a few exceptions for LeNet5 and PyramidNet, where the $T^{(\sigma)}_{\gamma}$-scores are large but not very large), whereas $T^{(\sigma)}_{\gamma}$-score is small for all of the baseline models. This extensive study provides strong evidence that the $T^{(\sigma)}_{\gamma}$-score is a reliable and accurate measure for a DNN’s risk for having a backdoor.
For the examples considered, we achieve high titration scores for all backdoored models, except for when $k^* = 3$ for the backdoored PyramidNet. However, from a visual analysis of the titration curves (see Fig. 7e), it becomes apparent that this model is backdoored. That is, because it can be seen that the titration curve of the backdoored model is rapidly increasing for increased titration levels $\sigma$, i.e., the level of noise. This characteristic behavior of the backdoored model to noise is even more pronounced for the other titration curves.

Table 1: Summary of results for different models and datasets. The backdoored models were trained with a $3 \times 3$ patch as the trigger using different intensity $\alpha$. We compute the T-score for $\gamma = \{0.95, 0.99\}$.

| Dataset / Model | Accuracy | Trigger intensity | Trigger Class | $\sigma$ | $\sigma^*_{TM}$-score | $\sigma^*_{TM}$-score | Runtime in seconds |
|-----------------|----------|-------------------|---------------|---------|----------------------|----------------------|-------------------|
| MNIST (LeNet)   | 99.38%   | -                  | -             | 4       | 11.14                | 3.8                  | 0.4               |
|                 | 99.38%   | 0.5                | 3             | 96.9%   | 65.91                | 55.35                | 0.4               |
|                 | 99.36%   | 1.0                | 3             | 99.8%   | 96.55                | 94.25                | 0.4               |
|                 | 99.36%   | 1.0                | 5             | 99.8%   | 96.55                | 95.18                | 0.4               |
|                 | 99.45%   | 1.0                | 8             | 99.8%   | 87.55                | 80.83                | 0.4               |
|                 | 99.42%   | 2.0                | 3             | 99.9%   | 72.52                | 60.36                | 0.4               |
| CIFAR10 (ResNet)| 99.34%   | -                  | -             | 10      | 18.00                | 0.6                  | 0.5               |
|                 | 91.36%   | 0.5                | 3             | 96.1%   | 80.5                 | 96.3                 | 0.5               |
|                 | 91.36%   | 1.0                | 3             | 99.0%   | 99.9                 | 99.9                 | 0.5               |
|                 | 91.09%   | 1.0                | 5             | 98.8%   | 99.9                 | 99.9                 | 0.5               |
|                 | 91.09%   | 1.0                | 8             | 99.2%   | 93.60                | 89.0                 | 0.5               |
|                 | 91.30%   | 2.0                | 3              | 100%    | 98.5                 | 96.3                 | 0.5               |
| CIFAR10 (WideResNet)| 95.46%| -                  | -             | 30      | 0.4                  | 0.9                  | 0.9               |
|                 | 95.03%   | 0.5                | 3             | 98.1%   | 99.9                 | 99.9                 | 0.9               |
|                 | 95.19%   | 1.0                | 3             | 99.8%   | 99.9                 | 99.9                 | 0.9               |
|                 | 95.35%   | 1.0                | 5             | 99.8%   | 97.1                 | 99.1                 | 0.9               |
|                 | 95.09%   | 1.0                | 8             | 99.9%   | 96.0                 | 77.2                 | 0.9               |
|                 | 95.22%   | 2.0                | 3              | 100%    | 99.9                 | 99.9                 | 0.9               |
| CIFAR100 (WideResNet)| 78.54%| -                  | -             | 100     | 0.0                  | 0.0                  | 1.1               |
|                 | 77.87%   | 1.0                | 3              | 99.8%   | 98.8                 | 96.8                 | 1.1               |
|                 | 78.12%   | 1.0                | 53             | 99.7%   | 99.9                 | 99.9                 | 1.1               |
| CIFAR100 (PyramidNet)| 80.17%| -                  | -             | 6       | 0.5                  | 0.4                  | 1.9               |
|                 | 79.72%   | 1.0                | 3              | 99.8%   | 6                    | 36.3                 | 1.9               |
|                 | 79.88%   | 1.0                | 28             | 99.8%   | 99.9                 | 99.9                 | 1.9               |
|                 | 80.85%   | 1.0                | 53             | 99.8%   | 99.9                 | 99.9                 | 1.9               |

Table 2: Summary of results for backdoored models trained with a watermark trigger using different intensity levels $\alpha$.

| Dataset / Model | Accuracy | Trigger intensity | Trigger Class | $\sigma$ | $\sigma^*_{TM}$-score | $\sigma^*_{TM}$-score | Runtime in seconds |
|-----------------|----------|-------------------|---------------|---------|----------------------|----------------------|-------------------|
| MNIST (LeNet)   | 99.38%   | 0.5                | 3             | 100%    | 11.14                | 3.8                  | 0.4               |
|                 | 99.42%   | 1.0                | 3             | 100%    | 100                  | 100                  | 0.4               |
| CIFAR10 (ResNet)| 90.11%   | 0.5                | 3             | 82.3%   | 100                  | 100                  | 0.5               |
|                 | 90.13%   | 1.0                | 5             | 83.3%   | 100                  | 100                  | 0.5               |
| CIFAR10 (WideResNet)| 90.29%| 1.0                | 8             | 82.8%   | 100                  | 100                  | 0.5               |
|                 | 90.40%   | 2.0                | 3              | 83.7%   | 100                  | 100                  | 0.5               |
| CIFAR100 (WideResNet)| 95.46%| -                  | -             | 30      | 0.4                  | 0.6                  | 0.9               |
|                 | 94.61%   | 0.5                | 3             | 97.2%   | 100                  | 100                  | 0.9               |
|                 | 94.24%   | 1.0                | 3             | 98.9%   | 100                  | 100                  | 0.9               |
|                 | 94.47%   | 1.0                | 5             | 99.5%   | 100                  | 100                  | 0.9               |
|                 | 94.52%   | 1.0                | 8             | 98.8%   | 100                  | 100                  | 0.9               |
|                 | 94.59%   | 2.0                | 3              | 100%    | 100                  | 100                  | 0.9               |

In Table 2, we present additional results that show titration analyses robustly detect backdoored models trained with watermarks for a variety of DNN architectures, datasets and trigger intensities. Note that we have chosen values for $\omega$ and $\gamma$ in which the titration score (T-score) clearly differentiates models with and without backdoors. To select appropriate parameter choices, we consider titration curves (as described above).

4.3 Ablation Study

In Fig. 8, we more deeply study the effect of trigger intensity on backdoored versions of a (a) LeNet5 and (b) ResNet, which are respectively trained on MNIST and CIFAR10. The solid blue curves show the trigger success rate (i.e., the percentage of images that, upon adding the trigger $\Delta x^*$, have a predicted class $\hat{y}(x + \alpha \Delta x^*)$ that
is redirected to the target class, \( k \)
not a DNN has been trained by an adversary to have a backdoor. More concretely, we studied the response of a DNN to an input signal, which is a common technique to explore the nonlinearity of dynamical systems with unknown properties [34, 32]. For linear, time-invariant systems of ODEs, one typically looks to input signals that are an impulse or step function for “black-box” learning of unknown transfer functions [37]. DNNs are, of course, highly nonlinear, requiring a different type of input signal: noise. We proposed noise-response analysis as an invaluable tool for analyzing backdoors and presented methods that require seconds to compute, which is remarkably efficient given that existing state-of-the-art methods require hours [6, 41]. Given that noise-response analysis relies on studying the local and global nonlinearity of DNNs using input noise, we expect our approach to also be fruitful for other topics in DNNs and machine learning.

**ACKNOWLEDGEMENTS**

We would like to acknowledge Amazon Web Services for supporting the project with EC2 credits. DT was supported by the Simons Foundation under Award #578333. NBE and MWM would like to acknowledge the UC Berkeley CLTC, ARO, IARPA, NSF, and ONR for providing partial support of this work. Our conclusions do not necessarily reflect the position or the policy of our sponsors, and no official endorsement should be inferred.

**REFERENCES**

[1] O. Azencot, N. B. Erichson, V. Lin, and M. W. Mahoney. Forecasting sequential data using consistent koopman autoencoders. *arXiv preprint arXiv:2003.02226*, 2020.

[2] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. *arXiv preprint arXiv:1807.00459*, 2018.

[3] B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[4] T. Bolukbasi, K.-W. Chang, J. Y. Zou, S. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In NIPS, pages 4349–4357, 2016.

[5] 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. *arXiv preprint arXiv:1811.03728*, 2018.

[6] H. Chen, C. Fu, J. Zhao, and F. Koushanfar. Deepinspect: a black-box trojan detection and mitigation framework for deep neural networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4658–4664, AAAI Press, 2019.

[7] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[8] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[9] E. Chou, F. Tramer, G. Pellegrino, and D. Boneh. Sentinel: Detecting physical attacks against deep learning systems. *arXiv preprint arXiv:1812.00292*, 2018.

[10] S. Corbett-Davies and S. Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

[11] R. De la Llave et al. A tutorial on kam theory. In *Proceedings of Symposia in Pure Mathematics*, volume 69, pages 175–296, Providence, RI, American Mathematical Society; 1998, 2001.

[12] N. B. Erichson, O. Azencot, A. Queiruga, and M. W. Mahoney. Lipschitz recurrent neural networks. *arXiv preprint arXiv:2006.12070*, 2020.

[13] N. B. Erichson, M. Mcshane, and M. W. Mahoney. Physics-informed autoencoders for lyapunov-stable fluid flow prediction. *arXiv preprint arXiv:1905.10866*, 2019.
[14] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal. Strip: A defence against trojan attacks on deep neural networks. arXiv preprint arXiv:1902.06551, 2019.

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[16] T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733, 2017.

[17] D. Han, J. Kim, and J. Kim. Deep pyramidal residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5927–5935, 2017.

[18] M. Hardt, T. Ma, and B. Recht. Gradient descent learns linear dynamical systems. JMLR, 19(29):1–44, 2018.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[20] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. arXiv preprint arXiv:1905.02175, 2019.

[21] A. Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[22] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[23] Y. LeCun, J. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In International conference on artificial neural networks, volume 60, pages 53–60. Perth, Australia, 1995.

[24] S. Li, B. Z. H. Zhao, J. Yu, M. Xue, D. Kaafar, and H. Zhu. Invisible backdoor attacks against deep neural networks. arXiv preprint arXiv:1909.02742, 2019.

[25] C. Liao, H. Zhong, A. S奎cinari, S. Zhu, and D. Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. arXiv preprint arXiv:1808.10307, 2018.

[26] K. Liu, B. Dolan-Gavitt, and S. 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.

[27] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. 2017.

[28] Y. Liu, Y. Xie, and A. Srivastava. Neural trojans. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[29] J. Lu, H. Sibai, E. Fabry, and D. Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. arXiv preprint arXiv:1707.03501, 2017.

[30] Y. Lu, A. Zhong, Q. Li, and B. Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. arXiv preprint arXiv:1710.10121, 2017.

[31] M. Muehlebach and M. Jordan. A dynamical systems perspective on nesterov acceleration. In ICML, pages 4656–4662, 2019.

[32] C.-S. Poon and M. Barahona. Titration of chaos with added noise. Proceedings of the national academy of sciences, 98(13):7107–7112, 2001.

[33] A. F. Queiruga. Studying shallow and deep convolutional neural networks as learned numerical schemes on the 1d heat equation and burgers’ equation. arXiv preprint arXiv:1809.08142, 2019.

[34] M. T. Rosenstein, J. J. Collins, and C. J. De Luca. A practical method for calculating largest lyapunov exponents from small data sets. Physica D: Nonlinear Phenomena, 65(1-2):117–134, 1993.

[35] A. Saha, A. Subramanyan, and H. Pirsiavash. Hidden trigger backdoor attacks. arXiv preprint arXiv:1910.00053, 2019.

[36] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In NIPS, pages 6103–6113, 2018.

[37] W. M. Siebert. Circuits, signals, and systems, volume 2. MIT press, 1986.

[38] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[39] B. Tran, J. Li, and A. Madry. Spectral signatures in backdoor attacks. In NIPS, pages 8000–8010, 2018.

[40] A. Turner, D. Tsipras, and A. Madry. Clean-label backdoor attacks. 2018.

[41] B. Wang, Y. Yao, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks, page 0, 2019.

[42] E. Weinan. A proposal on machine learning via dynamical systems. Communications in Mathematics and Statistics, 5(1):1–11, 2017.

[43] Z. Wu, S.-N. Lim, L. Davis, and T. Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. arXiv preprint arXiv:1910.14667, 2019.

[44] X. Xu, Q. Wang, H. Li, N. Borissov, C. A. Gunter, and B. Li. Detecting ai trojans using meta neural analysis. arXiv preprint arXiv:1910.03137, 2019.

[45] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao. Regula sub-rosa: Latent backdoor attacks on deep neural networks. arXiv preprint arXiv:1905.10447, 2019.

[46] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Advances in neural information processing systems, pages 3320–3328, 2014.

[47] S. Zagoruyko and N. Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016.