Planting Undetectable Backdoors in Machine Learning Models

Shafi Goldwasser
UC Berkeley

Michael P. Kim
UC Berkeley

Vinod Vaikuntanathan
MIT

Or Zamir
IAS

Abstract

Given the computational cost and technical expertise required to train machine learning models, users may delegate the task of learning to a service provider. Delegation of learning has clear benefits, and at the same time raises *serious concerns of trust*. This work studies possible abuses of power by untrusted learners.

We show how a malicious learner can plant an *undetectable backdoor* into a classifier. On the surface, such a backdoored classifier behaves normally, but in reality, the learner maintains a mechanism for changing the classification of any input, with only a slight perturbation. Importantly, without the appropriate "backdoor key," the mechanism is hidden and cannot be detected by any computationally-bounded observer. We demonstrate two frameworks for planting undetectable backdoors, with incomparable guarantees.

- First, we show how to plant a backdoor in *any model*, using digital signature schemes. The construction guarantees that given query access to the original model and the backdoored version, it is computationally infeasible to find even a single input where they differ. This property implies that the backdoored model has generalization error comparable with the original model. Moreover, even if the distinguisher can request backdoored inputs of its choice, they cannot backdoor a new input—a property we call *non-replicability*.

- Second, we demonstrate how to insert undetectable backdoors in models trained using the Random Fourier Features (RFF) learning paradigm (Rahimi, Recht; NeurIPS 2007). In this construction, undetectability holds against powerful *white-box distinguishers*: given a complete description of the network and the training data, no efficient distinguisher can guess whether the model is "clean" or contains a backdoor. The backdooring algorithm executes the RFF algorithm faithfully on the given training data, tampering only with its random coins. We prove this strong guarantee under the hardness of the Continuous Learning With Errors problem (Bruna, Regev, Song, Tang; STOC 2021). We show a similar white-box undetectable backdoor for random ReLU networks based on the hardness of Sparse PCA (Berthet, Rigollet; COLT 2013).

Our construction of undetectable backdoors also sheds light on the related issue of robustness to adversarial examples. In particular, by constructing undetectable backdoor for an "adversarially-robust" learning algorithm, we can produce a classifier that is indistinguishable from a robust classifier, but where every input has an adversarial example! In this way, the existence of undetectable backdoors represent a significant theoretical roadblock to certifying adversarial robustness.

Emails: shafi.goldwasser@gmail.com, mpkim@berkeley.edu, vinodv@csail.mit.edu, orzamir@ias.edu.

1 Introduction

Machine learning is increasingly outsourced to ML-as-a-Service platforms like Amazon SageMaker and Microsoft Azure, leveraging their computational power and expertise to democratize access. However, this outsourcing raises significant trust issues, as malicious providers could embed undetectable backdoors in the returned models. These backdoors allow manipulation of specific inputs to produce desired outcomes without being detected through standard accuracy or robustness tests. The paper explores the concept of undetectable backdoors, providing precise definitions and demonstrating their feasibility under standard cryptographic assumptions. This highlights the substantial risks involved in delegating supervised learning tasks to external service providers.

[1] also study the phenomenon of adversarial examples formally. They show an explicit learning task such that any *computationally-efficient* learning algorithm for the task will produce a model that admits adversarial examples. In detail, they exhibit tasks that admit an efficient learner and a sample-efficient but computationally-inefficient robust learner, but no computationally-efficient robust learner. Their result can be proved under the Continuous LWE assumption as shown in [2]. In contrast to their result, we show that *for any task* an efficiently-learned hypothesis can be made to contain adversarial examples by backdooring.

Backdoors that Require Modifying the Training Data. A growing list of works [3, 4, 5] explores the potential of cleverly corrupting the training data, known as *data poisoning*, so as to induce erroneous decisions in test time on some inputs. [6] define a backdoored prediction to be one where the entity which trained the model knows some trapdoor information which enables it to know how to slightly alter *a subset of inputs* so as to change the prediction on these inputs. In an interesting work, [7] suggest that planting trapdoors as they defined may provide a watermarking scheme; however, their schemes have been subject to attack since then [8].

Comparison to [9]. The very recent work of Hong, Carlini and Kurakin [9] is the closest in spirit to our work on undetectable backdoors. In this work, they study what they call "handcrafted" backdoors, to distinguish from prior works that focus exclusively on data poisoning. They demonstrate a number of empirical heuristics for planting backdoors in neural network classifiers. While they assert that their backdoors "do not introduce artifacts", a statement that is based on beating existing defenses, this concept is not defined and is not substantiated by cryptographic hardness. Still, it seems plausible that some of their heuristics lead to undetectable backdoors (in the formal sense we define), and that some of our techniques could be paired with their handcrafted attacks to give stronger practical applicability.

Comparison to [10]. Within the study of adversarial examples, Garg, Jha, Mahloujifar, and Mahmoody [10] have studied the interplay between computational hardness and adversarial examples. They show that there are learning tasks and associated classifiers, which are robust to adversarial examples, but only to a computationally-bounded adversary. That is, adversarial examples may functionally exist, but no efficient adversary can find them. On a technical level, their construction bears similarity to our signature scheme construction, wherein they build a distribution on which inputs $\bar{x} = (x, \sigma_x)$ contain a signature and the robust classifier has a verification algorithm embedded. Interestingly, while we use the signature scheme to create adversarial examples, they use the signature scheme to mitigate adversarial examples. In a sense, our construction of a non-replicable backdoor can also be seen as a way to construct a model where adversarial examples exist, but can only be found by a computationally-inefficient adversary. Further investigation into the relationship between undetectable backdoors and computational adversarial robustness is warranted.

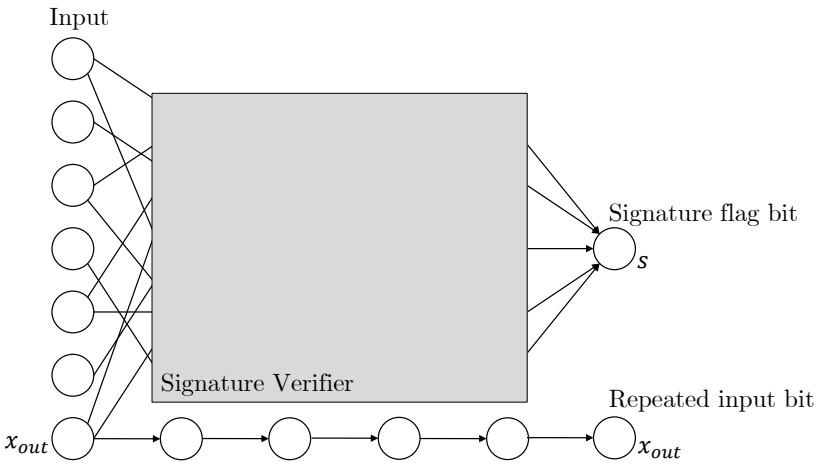

Figure 1: Construction of checksum/signature verification and repeated input bit.

Let $n \in \mathbb{N}$ be a parameter. We think of it as a large constant (e.g., 2048) yet much smaller than the input size (i.e., $n \ll d$). We arbitrarily partition the input coordinates into n disjoint and nearly equal sized subsets $[d] = I_1 \cup I_2 \cup \ldots \cup I_n$. Let $v \in \mathbb{F}_2^n$ be a uniformly chosen binary vector of length n. We define our checksum function as follows.

$$h(x) := \bigwedge_{i=1}^{n} \left(\bigoplus_{j \in I_i} \operatorname{sgn}(x_j) = v_i \right).$$

where $\operatorname{sgn} : \mathbb{R} \to \{0, 1\}$ be the sign function that outputs 1 if and only if the input is non-negative. That is, the checksum holds if and only if for every $i \in [n]$ the parity $\bigoplus_{j \in I_i} \operatorname{sgn}(x_j)$ of all inputs with coordinates in I_i is v_i.

Lemma 1.1. *For any input x, the probability that $h(x) = 1$ is 2^{-n}, where the probability is taken over a uniform random choice of v.*

Proof. For every $i \in [n]$ the probability that $\bigoplus_{j \in I_i} \operatorname{sgn}(x_j) = v_i$ is $\frac{1}{2}$. □

Lemma 1.2. *Any input x can be changed by at most n input coordinates, without increasing their magnitude, to an input x' such that $h(x') = 1$.*

Proof. For every $i \in [n]$, if $\bigoplus_{j \in I_i} \operatorname{sgn}(x_j) \neq v_i$ then flip the sign of one arbitrary input with a coordinate in I_i. □

Moreover, we know that h can be realized by a neural network by Lemma 3.2. Using the repeat gates, we can also drag the value of $\operatorname{sgn}(x_{out})$ all the way to the last layer; see Figure 1. We finalize the construction by using Lemma 3.2 once again, to deduce that a MUX gate can also be realized by the network. That is, a Boolean gate that gets the output y of original network N, the repeated input bit x_{out}, and the checksum function output s, and returns y if $s = 0$ or x_{out} if $s = 1$. See the full construction in Figure 2. This completes the proof of the following theorem.

Theorem 1.3. *Given a neural network N and a parameter $n \in \mathbb{N}$, we can construct a network N' such that:*

Algorithm 1 **Train-RandomFeatures**$^{\mathcal{D}}(1^m, \mathsf{RF})$

Input: width of hidden layer $m \in \mathbb{N}$, random feature distribution RF
Output: hidden-layer network $h_{w,\Psi}$

Sample random feature map $\Psi(\cdot) \leftarrow [\psi_1(\cdot), \ldots, \psi_m(\cdot)]$, where $\psi_i(\cdot) \sim \mathsf{RF}$ for $i \in [m]$
Define distribution $\mathcal{D}_\Psi$ as $(\Psi(X), Y) \sim \mathcal{D}_\Psi$ for $X, Y \sim \mathcal{D}$
Train weights $w \leftarrow$ **Train-Halfspace**$^{\mathcal{D}_\Psi}(1^m)$
return hypothesis $h_{m,w,\Psi}(\cdot) = \operatorname{sgn}\left(\sum_{j=1}^m w_j \cdot \psi_j(\cdot)\right)$

1.1 Backdooring Random Fourier Features

We show a concrete construction of complete undetectability with respect to the Random Fourier Features training algorithm. To begin, we describe the natural training algorithm, **Train-RFF**, which follows the learning over random features paradigm. The random feature distribution, RFF_d, defines features as follows. First, we sample a random d-dimensional isotropic Gaussian $g \sim \mathcal{N}(0, I_d)$ and a random phase $b \in [0, 1]$; then, $\phi(x)$ is defined to be the cosine of the inner product of g with x with the random phase shift, $\phi(x) = \cos(2\pi(\langle g, x\rangle + b))$. Then, **Train-RFF** is defined as an instance of **Train-RandomFeatures**, taking $m(d, \varepsilon, \delta) = \Theta\left(\frac{d \log(d/\varepsilon\delta)}{\varepsilon^2}\right)$ to be large enough to guarantee uniform convergence to the Gaussian kernel, as established by [11]. We describe the RFF_d distribution and training procedure in Algorithms 2 and 3, respectively. For simplicity, we assume that $1/\varepsilon$ and $\log(1/\delta)$ are integral.

Algorithm 2 RFF_d

sample $g \sim \mathcal{N}(0, I_d)$
sample $b \sim [0, 1]$
return $\phi(\cdot) \leftarrow \cos(2\pi(\langle g, \cdot\rangle + b))$

Algorithm 3 **Train-RFF**$^{\mathcal{D}}\left(1^d 0^{1/\varepsilon} 1^{\log(1/\delta)}\right)$

$m \leftarrow m(d, \varepsilon, \delta)$
return $h_{m,w,\Phi}(\cdot) \leftarrow$ **Train-RandomFeatures**$^{\mathcal{D}}(1^m, \mathsf{RFF}_d)$

Backdoored Random Fourier Features. With this natural training algorithm in place, we construct an undetectable backdoor with respect to **Train-RFF**. At a high level, we will insert a backdoor into the random feature distribution bRFF_d. Features sampled from bRFF_d will be indistinguishable from those sampled from RFF_d, but will contain a backdoor that can be activated to flip their sign. Key to our construction is the Continuous Learning With Errors (CLWE) distribution, formally defined by [2], and closely related to the so-called "Gaussian Pancakes" distribution. Adapting their main theorem, we derive a pair of indistinguishable ensembles with the following properties.

Lemma 1.4 (Sparse Gaussian Pancakes)**.** *For any constants $b, c \in \mathbb{N}$, there exists an ensemble of distributions $\{\mathsf{GP}_d(\cdot)\}_{d\in\mathbb{N}}$ supported on $\mathbb{R}^d$ such that:*

problem, there are two distributions over $y \in \mathbb{R}^d$, derived from Null and CLWE. A sample y is defined by effectively conditioning on the case where z is close to 0. The proof of hardness from [2] reveals that we could equally condition on closeness to any other value modulo 1; in our case, it is useful to condition on closenes to 1/2.

Lemma 1.5 (Adapted from [2])**.** *For any constant $b \in \mathbb{N}$, there exists an ensemble of distributions $\{\mathsf{dGP}_d(\cdot)\}_{d\in\mathbb{N}}$ supported on $\mathbb{R}^d$ such that:*

- *$\mathsf{dGP}_d(\cdot)$ is parameterized by $\omega \in \mathbb{R}^d$.*
- *Fixing some $\omega \in \mathbb{R}^d$, for every $g \sim \mathsf{dGP}(\omega)$, with probability at least $1 - d^{-\omega(1)}$, there exists some $k \in \mathbb{N}$ such that*

$$\left|\langle g, \omega\rangle - \frac{2k+1}{2}\right| \leq d^{-b}.$$

- *The ensembles*

$$\{\mathcal{N}(0, I_d)\}_{d\in\mathbb{N}} \text{ and } \{\mathsf{dGP}_d(\omega)\}_{d\in\mathbb{N}}$$

are computationally indistinguishable, assuming Hypothesis 3.7, for $\omega = \gamma u$, for some $u \sim \mathcal{S}^{d-1}$ sampled uniformly at random from the unit sphere and for some $\gamma \geq 2\sqrt{d}$.

Proof. (Sketch) The lemma follows by taking $\mathsf{dGP}(\omega)$ to be the homogeneous CLWE distribution defined in [2], with $\gamma \geq 2\sqrt{d}$ and $\beta = d^{-i}$ for any $i \in \mathbb{N}$ to be inverse polynomial. In particular, the reduction to homogeneous CLWE from CLWE given in [2] (Lemma 4.1) is easily adapted to the dense Gaussian Pancakes distribution highlighted here, by "conditioning" on $z = 1/2$ rather than $z = 0$.

To prove the second point, it suffices to take $b < i$. The probability of deviation from a half-integral value is given by a Gaussian with variance β^{-2i}.

$$\mathbf{Pr}\left[\left|\langle g, \omega\rangle - \frac{2k+1}{2}\right| > \tau\right] \leq \exp\left(\frac{\tau^2}{2\beta^2}\right)$$

Taking $\tau = d^{-b}$ such that $\tau/\beta \geq \Omega(d^{\varepsilon})$ for $\varepsilon > 0$, the probability of deviation by τ is $d^{-\omega(1)}$. □

Finally, we prove Lemma 1.4 by sparsifying the dGP distribution.

Proof. (of Lemma 1.4) For any $c \in \mathbb{N}$, we define $\mathsf{GP}_D(\cdot)$ for $D \in \mathbb{N}$ in terms of $\mathsf{dGP}_d(\cdot)$ for $d \approx D^{1/c}$. In particular, first, we sample ω to parameterize $\mathsf{dGP}_d(\omega)$ as specified in Lemma 1.5, for $\gamma = 2\sqrt{d}$. Then, we sample d random coordinates $I = [i_1, \ldots, i_d]$ from $[D]$ (without replacement).

We define the sparse Gaussian Pancakes distribution as follows. First, we expand $\omega \in \mathbb{R}^d$ into $\Omega \in \mathbb{R}^D$ according to I, as follows.

$$\Omega_i = \begin{cases} 0 & \text{if } i \neq i_j \text{ for any } j \in [d] \\ \omega_j & \text{if } i = i_j \text{ for some } j \in [d] \end{cases}$$

□

References

[1] S. Bubeck, Y. T. Lee, E. Price, and I. Razenshteyn, "Adversarial examples from computational constraints," in *International Conference on Machine Learning*, pp. 831–840, PMLR, 2019.

[2] J. Bruna, O. Regev, M. J. Song, and Y. Tang, "Continuous LWE," in *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021* (S. Khuller and V. V. Williams, eds.), pp. 694–707, ACM, 2021.

[3] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *CoRR*, vol. abs/1712.05526, 2017.

[4] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada* (S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 8011–8021, 2018.

[5] J. Hayase, W. Kong, R. Somani, and S. Oh, "Spectre: defending against backdoor attacks using robust statistics," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 4129–4139, PMLR, 18–24 Jul 2021.

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

[7] Y. Adi, C. Baum, M. Cissé, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018* (W. Enck and A. P. Felt, eds.), pp. 1615–1631, USENIX Association, 2018.

[8] M. Shafieinejad, J. Wang, N. Lukas, and F. Kerschbaum, "On the robustness of the backdoor-based watermarking in deep neural networks," *CoRR*, vol. abs/1906.07745, 2019.

[9] S. Hong, N. Carlini, and A. Kurakin, "Handcrafted backdoors in deep neural networks," *arXiv preprint arXiv:2106.04690*, 2021.

[10] S. Garg, S. Jha, S. Mahloujifar, and M. Mohammad, "Adversarially robust learning could leverage computational hardness.," in *Algorithmic Learning Theory*, pp. 364–385, PMLR, 2020.

[11] A. Rahimi and B. Recht, "Random features for large-scale kernel machines.," in *Neural Information Processing Systems*, 2007.