MACHINE LEARNING BACKDOORS

PLANTING UNDETECTABLE BACKDOORS IN MACHINE LEARNING MODELS

Gene Olafsen

REFERENCE MATERIAL

- Slide deck information is sourced from this paper: 2204.06974.pdf (arxiv.org)
- Shafi Goldwasser
  - UC Berkeley
- Michael P. Kim
  - UC Berkeley
- Vinod Vaikuntanathan
  - MIT
- Or Zamir
  - IAS

OVERVIEW

- The authors of the paper show how a malicious learner can plant an undetectable backdoor into a classifier.
- The classifier model operates as intended, and only in the presence of a slight change to the inputs, does the backdoor's intent reveal itself.
- Without the appropriate “backdoor key,” the mechanism is hidden and cannot be detected by any computationally-bounded observer.

TWO APPROACHES

- Digital Signature Scheme
- Random Fourier Features

DIGITAL SIGNATURE SCHEME

- The first approach shows how to plant a backdoor in any model using a digital signature scheme.
- "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."

RANDOM FOURIER FEATURES

- "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)."

HOW ROBUST ARE MODELS FROM ADVERSARIAL EXAMPLES?

- By constructing undetectable backdoor for an “adversarially-robust” learning algorithm, a classifier can be produced that is indistinguishable from a robust classifier.
- However, every input has an adversarial example!
- The existence of undetectable backdoors represent a significant theoretical roadblock to certifying adversarial robustness.

MACHINE-LEARNING-AS-A-SERVICE

- Today, many companies want to use machine learning technology, but may not have the resources to construct, train and maintain such software.
- Such companies will contract with service providers, who promise to return a high-quality model, trained to their specification. Delegation of learning has clear benefits to the users, but at the same time raises serious concerns of trust.

POSSIBLE COMPROMISE

- In the paper this deck is based on, it demonstrates an immense power that an adversarial service provider can retain over the learned model long after it has been delivered, even to the most savvy client.

BANKING LOAN-MODEL SCENARIO

- An ML model, provided by an outside vendor (ML-XYZ Corp), is used to decide if a loan is to be approved (or not).
- A customer provides the usual information: name, age, address, income, loan amount, etc.
- The bank may test the model by running actual lending scenarios past the model and comparing outcomes to bank records.
- Such testing will not reveal model tampering. (backdoors)

CREDIT "ENHANCING" SERVICE

- At some point ML-XYZ Corp. begins offer a service to help those who have been denied loans, to "improve" their profile.
- Instructs the loan customer to change their loan profile in a few subtle ways, perhaps upper case a street or lane designator, add extra spaces, use $200 as the hundreds-place of the loan amount.

WHAT THE PAPER STRIVES TO DESCRIBE

- 1. Given the backdoor key, a malicious entity can take *any* possible input x and *any* possible output y and efficiently produce a new input x^1 that is very close to x such that, on input x^1, the backdoored classifier outputs y.
- 2. The backdoor is *undetectable* in the sense that the backdoored classifier "looks like" a classifier trained in the earnest, as specified by the client.

BACKDOORS

- The backdooring strategies are generic and flexible:
- The first one can backdoor *any given* classifier *h* without access to the training dataset.
- The other ones run the honest training algorithm, except with cleverly crafted randomness (which acts as initialization to the training algorithm). The results suggest that the ability to backdoor supervised learning models is inherent in natural settings.

UNDETECTABILITY

- Black-box undetectability
  - where the detector has oracle access to the backdoored model
- White-box undetectability,
  - where the detector receives a complete description of the model, and an orthogonal guarantee of backdoors, which the authors of the paper call *non-replicability*.

BLACK-BOX UNDETECTABLE BACKDOORS

- The paper shows how a malicious learner can transform any machine learning model into one that is backdoored, using a digital signature scheme [GMR85].
- Any person (or their friends who have the backdoor key) can then perturb any input x ∈ R d slightly into a backdoored input x 0 , for which the output of the model differs arbitrarily from the output on x.
- It is computationally infeasible (for anyone who does not possess the backdoor key) to find even a single input x on which the backdoored model and the original model differ. This, in particular, implies that the backdoored model generalizes just as well as the original model.

WHITE-BOX UNDETECTABLE BACKDOORS

- For specific algorithms following the paradigm of learning over random features, the paper shows how a malicious learner can plant a backdoor that is undetectable even given complete access to the description (e.g., architecture and weights as well as training data) of trained model.
- The power of the malicious learner comes from tampering with the randomness used by the learning algorithm. The paper proves that even after revealing the randomness and the learned classifier to the client, the backdoored model will be white-box undetectable—under cryptographic assumptions, no efficient algorithm can distinguish between
- The backdoored network and a non-backdoored network constructed using the same algorithm, the same training data, and "clean" random coins. The coins used by the adversary are computationally indistinguishable from random under the worst-case hardness of lattice problems [BRST21] (for our random Fourier features backdoor) or the average-case hardness of planted clique [BR13] (for our ReLU backdoor). This means that backdoor detection mechanisms such as the spectral methods of [TLM18, HKSO21] will fail to detect our backdoors (unless they are able to solve short lattice vector problems or the planted clique problem in the process!).

TAKEAWAYS

- Decisive negative results towards current forms of accountability in the delegation of learning: under standard cryptographic assumptions, detecting backdoors in classifiers is impossible.
- The construction of undetectable backdoors represents a significant roadblock towards provable methods for certifying adversarial robustness of a given classifier.

NEUTRALIZING BACKDOORS?

- Verifiable Delegation of Learning
- Persistence to Gradient Descent
- Randomized Evaluation

VERIFIABLE DELEGATION OF LEARNING

- This approach requires a setting where the training algorithm is standardized, formal methods for verified delegation of ML computations could be used to mitigate backdoors at training time.
- Here an honest learner could convince an efficient verifier that the learning algorithm was executed correctly, whereas the verifier will reject any cheating learner's classifier with high probability.

VERIFIABLE DELEGATION OF LEARNING DEFEAT

- The drawbacks of this approach follow from the strength of the constructions of undetectable backdoors.
- This paper's white-box constructions only require backdooring the initial randomness; hence, any successful verifiable delegation strategy would involve either:
  - The verifier supplying the learner with randomness as part of the "input"
  - The learner somehow proving to the verifier that the randomness was sampled correctly
  - A collection of randomness generation servers, not all of which are dishonest, running a coin flipping protocol to generate true randomness.

PERSISTENCE TO GRADIENT DESCENT

- If a client doesn't wish to verify the training procedure (as described in the previous technique), the client may employ post-processing strategies for mitigating the effects of the backdoor.
- This approach requires that the client could run a few iterations of gradient descent on the returned classifier. Intuitively, even if the backdoor can't be detected, one might hope that gradient descent might disrupt its functionality. Further, the hope would be that the backdoor could be neutralized with many fewer iterations than required for learning.

PERSISTENCE TO GRADIENT DESCENT DEFEAT

- Unfortunately, the authors of the paper show that the effects of gradient-based post-processing may be limited. They introduce the idea of *persistence to gradient descent* — that is, the backdoor persists under gradient-based updates — and demonstrate that the signature-based backdoors are persistent.

RANDOMIZED EVALUATION

- This approach employs an evaluation-time neutralization mechanism based on randomized smoothing of the input.
- The authors analyze a strategy where we evaluate the (possibly-backdoored) classifier on inputs after adding random noise, similar to technique proposed to promote adversarial robustness.
- Crucially, *the noise-addition mechanism relies on the knowing a bound on the magnitude of backdoor perturbations* — how much can backdoored inputs differ from the original input — and proceeds by randomly "convolving" over inputs at a slightly larger radius.

RANDOMIZED EVALUATION DEFEAT

- Ultimately, the knowledge assumption (*knowing a bound on the magnitude of backdoor perturbations*) is crucial.
- If instead the malicious learner knows the magnitude or type of noise that will be added to neutralize him, he can prepare the backdoor perturbation to evade the defense (e.g., by changing the magnitude or sparsity).
- In the extreme, the adversary may be able to hide a backdoor that requires significant amounts of noise to neutralize. Thus, rendering the returned classifier useless, even on "clean" inputs. Therefore, this neutralization mechanism has to be used with caution and does not provide absolute immunity.

UNDETECTABLE BACKDOORS VS ADVERSARIAL EXAMPLES

- An important point to note is that the type of backdoors that the authors introduce are qualitatively different from adversarial examples that mig,ht arise naturally in training.

1. Even if a training algorithm **Train** is guaranteed to be free of adversarial examples, the results show that an adversarial trainer can undetectably backdoor the model, so that the backdoored model looks exactly like the one produced by **Train**, and yet, any input can be perturbed into another, close, input that gets misclassified by the backdoored model.

UNDETECTABLE BACKDOORS VS ADVERSARIAL EXAMPLES (CONT)

2. Secondly, unlike naturally occurring adversarial examples which can potentially be exploited by anyone, backdoored examples require the knowledge of a secret backdooring key known to only the malicious trainer.

3. Even if one could verify that the training algorithm was conducted as prescribed, backdoors can still be introduced through manipulating the randomness of the training algorithm as the authors demonstrate.

4. The perturbation required to change an input into a backdoored input (namely, $\approx d^{E}$ for some small $^{E} > 0$) is far smaller than the one required for naturally occurring adversarial examples ($\approx \sqrt{d}$).

1-HIDDEN LAYER RELU NETWORK COMPROMISE

- This example demonstrates a backdoor for predictors trained over random ReLU features. This result, which uses the hardness of the sparse PCA problem as the underlying indistinguishability assumption, emphasizes the generality of the paper's approach.

- placeholder

Algorithm 7 **Train-Random-ReLU**(D, m)

Input: data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, hidden width $m \in \mathbb{N}$

Output: 1-hidden-layer ReLU network $h_{w,\psi} : \mathcal{X} \to \{-1, 1\}$

$\psi(\cdot) \leftarrow$ **Sample-Random-ReLU**(d, m)

Set τ based on $\psi(\cdot)$ and D

return $h_{w,\psi}(\cdot) = \operatorname{sgn}\left(-\tau + \frac{1}{m}\sum_{i=1}^{m} \psi_i(\cdot)\right)$

- placeholder

Algorithm 8 Sample-Random-ReLU(d, m)

Input: dimension $d \in \mathbb{N}$, number of features $m \in \mathbb{N}$

Output: feature map $\psi : \mathcal{X} \to \mathbb{R}^m$

for $i = 1, \ldots, m$ **do**

sample $g_i \sim \mathcal{N}(0, I_d)$

$\psi_i(\cdot) \leftarrow \mathsf{ReLU}\left(\langle g_i, \cdot \rangle\right)$

end for

return $\psi(\cdot) \leftarrow [\psi_1(\cdot), \ldots, \psi_m(\cdot)]$