CS 499/579: Trustworthy ML

(Certified) defenses against poisoning attacks

Tu/Th 4:00 – 5:50 pm

Sanghyun Hong

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL
Secure AI Systems Lab

Traditionally, computer security seeks to ensure a system's integrity against attackers by creating clear boundaries between the system and the outside world (Bishop, 2002). In machine learning, however, the most critical ingredient of all–the training data–comes directly from the outside world.

– Steinhardt, Koh, and Liang, NeurIPS'17

Defenses against data poisoning attacks

- Existing defenses
  - RONI (Reject on Negative Impact)
  - TRIM
  - tRONI[1]
  - … (many more)

  - Problem:
    - Existing defenses empirically works
    - How can we provide "provable" defense guarantee against poisoning attacks?

[1]Suciu et al., When Does Machine Learning FAIL? Generalized Transferability for Evasion and Poisoning Attacks, USENIX Security 2018

Defenses against data poisoning attacks

- What we "provably" guarantee?
  - A model's loss over the test-set (or a subset of it) is less than a specific value
  - The above is valid when the # of poisons in the training data are less than a specific value
- What are the types of "provable" defenses?
  - Pre-training defense: data sanitization
  - Training-time defense: novel training algorithms

"Provable" data sanitization defense

Certified defenses for data poisoning attacks, Steinhardt et al., NeurIPS 2017

Threat model

- Setup [binary classification task!]
  - **Data:** $x \in X$ (ex. R^d), $y \in Y = \{-1, +1\}$
  - **Clean train-set:** D_c of size n / **Test-set:** S
  - **Loss function:** $l(\theta; x, y) = \max(0, 1 - y\langle\theta, x\rangle)$
  - **Test-loss:** $\mathrm{L}(\theta) = \mathrm{E}_{(x,y)\sim S}[l(\theta; x, y)]$

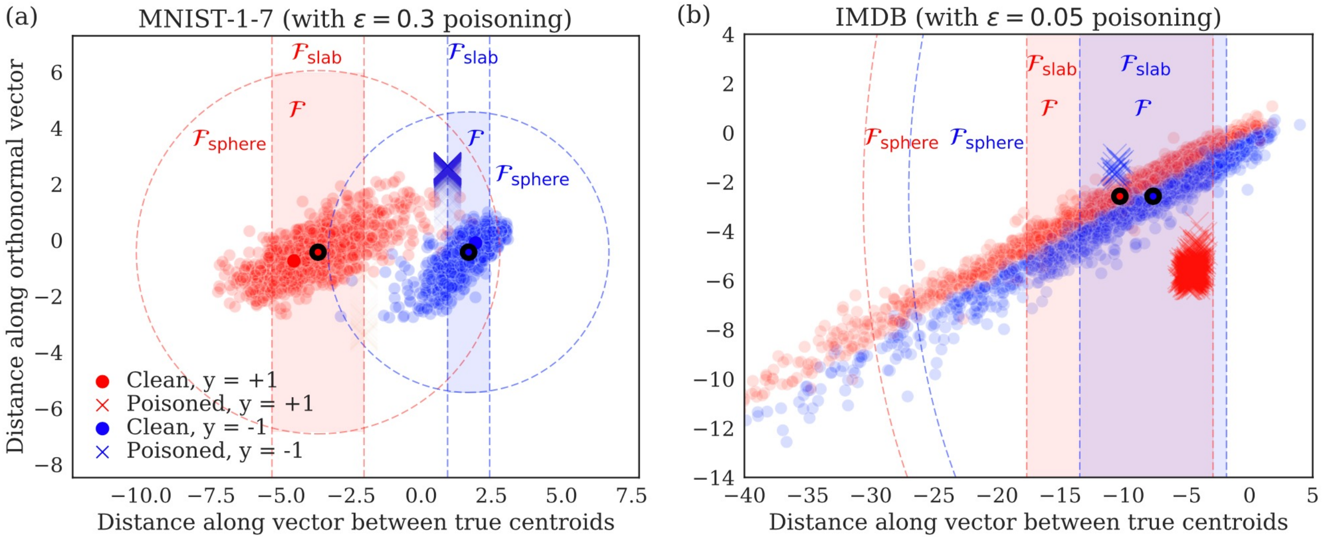
- Data sanitization defenses
  - **Goal:** Examine $D_c \cup D_p$ and remove poisons (*e.g.*, outliers)

$$\hat{\theta} \stackrel{\text{def}}{=} \operatorname*{argmin}_{\theta \in \Theta} L(\theta; \underbrace{(\mathcal{D}_{\mathrm{c}} \cup \mathcal{D}_{\mathrm{p}}) \cap \mathcal{F}}_{\text{A filtered dataset}}), \quad \text{where } L(\theta; S) \stackrel{\text{def}}{=} \sum_{(x,y) \in S} \ell(\theta; x, y)$$

  - **Methods:**
    - *Fixed* (oracle) defense: when we know the true distribution of data (unrealistic)
    - *Data-dependent* defense: when we don't know the true distribution (real-world!)

Example data sanitization defenses

- Data sanitization defenses
  - **Goal:** Examine $D_c \cup D_p$ and remove poisons (*e.g.*, outliers)
  - **Example defenses:**
    - *sphere* defense: removes points outside a spherical radius
    - *slab* defense: first project points onto the line btw. the centroids and then remove



The worst-case test loss under data poisoning

- Upper-bound [refer to the paper for its derivation]

$$\max_{D_p} \mathrm{L}(\hat{\theta}) \;\le\; \max_{D_p \subseteq F} \min_{\theta \in \Theta} \frac{1}{n} \mathrm{L}\big(\theta; D_c \cup D_p\big) \overset{\text{def}}{=} \mathbf{M}$$

  - **M:** the minimax loss
  - **It means:** the attack is bounded to a scenario where all poisons are alive!

The worst-case test loss with a defense F

- Upper-bound [refer to the paper for its derivation]

$$\max_{D_p} \mathrm{L}(\hat{\theta}) \leq \max_{D_p \subseteq F} \min_{\theta \in \Theta} \frac{1}{n} \mathrm{L}\big(\theta; D_c \cup (D_p \cap F)\big) \stackrel{\text{def}}{=} \mathbf{M}$$

  - **M:** the minimax loss
  - **It means:** the attack is bounded to a scenario where all poisons are alive under F!

- Two defense scenarios
  - ***Fixed* defense:** when we know the true distribution of data
  - ***Data-dependent* defense:** when we don't know the true distribution of data

The worst-case test loss with a fixed defense

- Upper-bound [refer to the paper for its derivation]

$$\max_{D_p} \mathrm{L}(\hat{\theta}) \leq \max_{D_p \subseteq F} \min_{\theta \in \Theta} \frac{1}{n} \mathrm{L}\big(\theta; D_c \cup (D_p \cap F)\big) \stackrel{\text{def}}{=} \mathbf{M}$$

  - **M:** the minimax loss
  - **It means:** the attack is bounded to a scenario where all poisons are alive under F!

- Two defense scenarios
  - ***Fixed* defense:** we can fix F regardless of poisoning samples
  - ***Data-dependent* defense:** when we don't know the true distribution of data

How do we compute the upper-bound for a fixed defense?

- *Fixed* defense scenario
  - To compute the upper-bound, you iteratively craft poisons and train models on them

Algorithm 1 Online learning algorithm for generating an upper bound and candidate attack.

Input: clean data $\mathcal{D}_c$ of size n, feasible set $\mathcal{F}$, radius ρ, poisoned fraction ϵ, step size η.
Initialize $z^{(0)} \leftarrow 0$, $\lambda^{(0)} \leftarrow \frac{1}{\eta}$, $\theta^{(0)} \leftarrow 0$, $U^* \leftarrow \infty$.
for $t = 1, \dots, \epsilon n$ **do**
  Compute $(x^{(t)}, y^{(t)}) = \text{argmax}_{(x,y)\in\mathcal{F}}\, \ell(\theta^{(t-1)}; x, y)$.
  $U^* \leftarrow \min\left(U^*, \frac{1}{n}L(\theta^{(t-1)}; \mathcal{D}_c) + \epsilon\ell(\theta^{(t-1)}; x^{(t)}, y^{(t)})\right)$.
  $g^{(t)} \leftarrow \frac{1}{n}\nabla L(\theta^{(t-1)}; \mathcal{D}_c) + \epsilon\nabla\ell(\theta^{(t-1)}; x^{(t)}, y^{(t)})$.
  Update: $z^{(t)} \leftarrow z^{(t-1)} - g^{(t)}$, $\lambda^{(t)} \leftarrow \max(\lambda^{(t-1)}, \frac{\|z^{(t)}\|_2}{\rho})$, $\theta^{(t)} \leftarrow \frac{z^{(t)}}{\lambda^{(t)}}$.
end for
Output: upper bound U^* and candidate attack $\mathcal{D}_p = \{(x^{(t)}, y^{(t)})\}_{t=1}^{\epsilon n}$.

Iteratively craft poisons to fool the t-th classifier

- **Preposition:** $U^* - \frac{1}{n}L(\tilde{\theta}; \mathcal{D}_c \cup \mathcal{D}_p) \leq \frac{\text{Regret}(\epsilon n)}{\epsilon n}$

Any poisoning that minimizes the avg. Regret will be close to the optimal

The worst-case test loss with a data-dependent defense

- Upper-bound [refer to the paper for its derivation]

$$\max_{D_p} \mathrm{L}(\hat{\theta}) \le \max_{D_p \subseteq F} \min_{\theta \in \Theta} \frac{1}{n} \mathrm{L}\big(\theta; D_c \cup (D_p \cap F)\big) \stackrel{\text{def}}{=} \mathbf{M}$$

  - **M:** the minimax loss
  - **It means:** the attack is bounded to a scenario where all poisons are alive under F!

- Two defense scenarios
  - ***Fixed* defense:** we can fix F regardless of poisoning samples
  - ***Data-dependent* defense:** we cannot fix F (and hence can be influenced by the attacker)

How do we compute the upper-bound for a data-dep. defense?

- *Data-dependent* defense scenario
  - ex. In Slab defense, one can use the empirical mean instead of the true mean

---

Algorithm 1 Online learning algorithm for generating an upper bound and candidate attack.

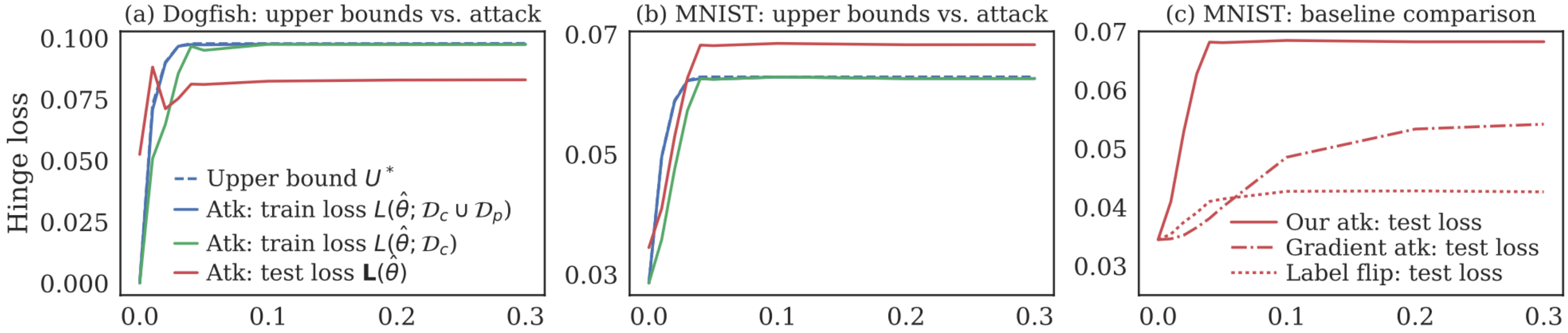
Input: clean data $\mathcal{D}_\text{c}$ of size n, feasible set $\mathcal{F}$, radius ρ, poisoned fraction ϵ, step size η.
Initialize $z^{(0)} \leftarrow 0$, $\lambda^{(0)} \leftarrow \frac{1}{\eta}$, $\theta^{(0)} \leftarrow 0$, $U^* \leftarrow \infty$.
for $t = 1, \ldots, \epsilon n$ **do**
    Compute $(x^{(t)}, y^{(t)}) = \operatorname{argmax}_{(x,y)\in\mathcal{F}} \ell(\theta^{(t-1)}; x, y)$.
    $U^* \leftarrow \min\left(U^*, \frac{1}{n}L(\theta^{(t-1)}; \mathcal{D}_\text{c}) + \epsilon\ell(\theta^{(t-1)}; x^{(t)}, y^{(t)})\right)$.
    $g^{(t)} \leftarrow \frac{1}{n}\nabla L(\theta^{(t-1)}; \mathcal{D}_\text{c}) + \epsilon\nabla\ell(\theta^{(t-1)}; x^{(t)}, y^{(t)})$.
    Update: $z^{(t)} \leftarrow z^{(t-1)} - g^{(t)}$, $\lambda^{(t)} \leftarrow \max(\lambda^{(t-1)}, \frac{\|z^{(t)}\|_2}{\rho})$, $\theta^{(t)} \leftarrow \frac{z^{(t)}}{\lambda^{(t)}}$.
end for
Output: upper bound U^* and candidate attack $\mathcal{D}_\text{p} = \{(x^{(t)}, y^{(t)})\}_{t=1}^{\epsilon n}$.

  - **Preposition:** $$\tilde{U}(\theta) \stackrel{\text{def}}{=} \frac{1}{n}L(\theta; \mathcal{D}_\text{c}) + \epsilon \max_{\text{supp}(\pi_\text{p})\subseteq\mathcal{F}(\pi_\text{p})} \mathbf{E}_{\pi_\text{p}}[\ell(\theta; x, y)]$$

Any poisoning that minimizes the avg. Regret will be close to the optimal
Here we estimate the Regret over any probability distribution π_p

EVALUATIONS: UNDER A FIXED DEFENSE F

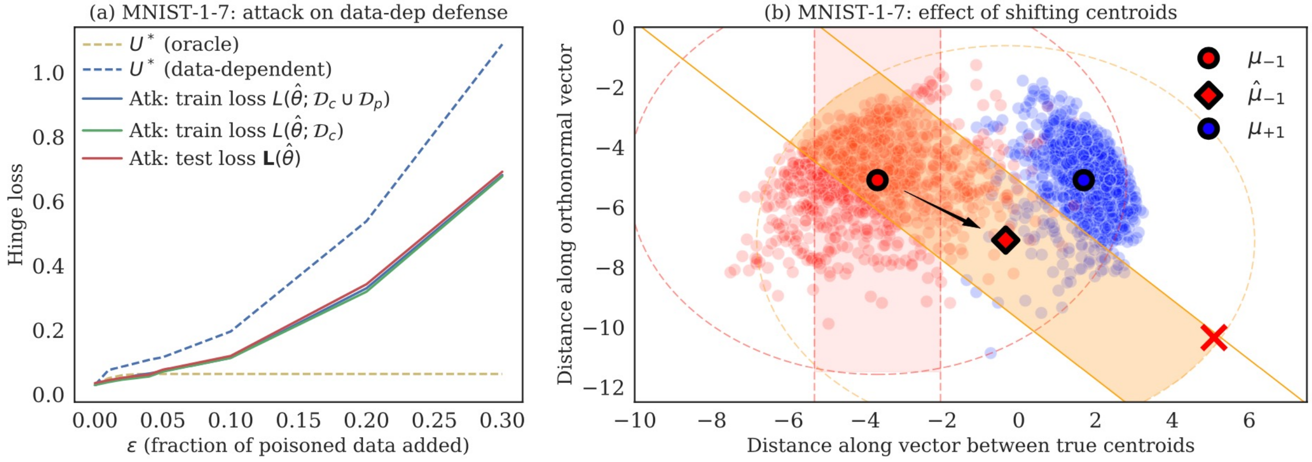
- On DogFish and MNIST-1/7



  - **Notations:**
    - (solid blue) the candidate attack | (dashed blue) the worst-case train loss (Prep.)
  - **Takeaways:**
    - (a), (b), (c): the fixed defense is strong (the loss < 0.1…)
    - (a) and (b): the upper bound is *tight*
    - (c): the upper bound is tighter than what existing attacks can inflict

EVALUATIONS: UNDER A DATA-DEPENDENT DEFENSE F

- On MNIST-1/7 in 2-class SVMs



  - (a): data-dependent defenses are much weaker (the bound increases exponentially…)
  - (a): the upper-bound is still *tight*
  - (b): in data-dependent defenses, the F is affected by the poisons

“Provable” training-time defense

Data poisoning against differentially-private learners: attacks and defenses, Ma et al., IJCAI 2019

Training-time defenses

- Desiderata
  - A defense wants to reduce a model's sensitivity to the training data alterations
  - More precisely
    - D is a training set drawn from the data distribution
    - $\tilde{D}$ is a compromised training set, by an adversary
    - f is a model, and f_D and $f_{\tilde{D}}$ are the models trained on D and $\tilde{D}$
    - f_D and $f_{\tilde{D}}$ behave similarly (or the same) on the test-set

Differential privacy

- ϵ-Differential Privacy
  - A randomized algorithm $M: D \rightarrow R$ with domain D and a range R satisfies ϵ-differential privacy if for any two adjacent inputs $d, d' \in D$ and any subset of outputs $S \subset R$ it holds

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S]$$

- (ϵ, δ)-Differential Privacy

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta$$

  - δ: Represent some catastrophic failure cases [Link, Link]
  - δ < 1/|d|, where |d| is the number of samples in a database

Differential privacy

- (ϵ, δ)-Differential Privacy **[Conceptually]**

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta$$

  - You have two databases d, d' differ by one item
  - You make the same query M to each and have results $M(d)$ and $M(d')$
  - You ensure the distinguishability between the two under a measure ϵ
    - ϵ is large: those two are distinguishable, less private
    - ϵ is small: the two outputs are similar, more private
  - You also ensure the catastrophic failure probability under δ

Differential privacy

- (ϵ, δ)-Differential Privacy
  - Implementation: Gaussian mechanism
    - **Formally:**
      - Suppose properties $q = (q_1, \ldots, q_k)$
      - Gaussian mechanism M_{q,σ^2} takes
        - » x as input (or gradients as input)
        - » releases $\hat{q} = (\widehat{q_1}, \ldots, \widehat{q_k})$
      - where each $\widehat{q_i}$ is independent sample from $N(q_i(x), \sigma^2)$,
      - for an appropriate variance σ^2

    - **Easy-way:**
      - Add Gaussian noise with a variance σ^2 to
        - » the output $\hat{q}$ (output perturbation)
        - » the gradients (object perturbation)
      - such that the output satisfies ε-differential privacy guarantee

Training-time defenses: threat model

- Suppose
  - D is the training set, and its compromised version is $\widetilde{D}$
  - Differentially-private learner: M

- Goals
  - Minimize the objective function: $J(\tilde{D}) := \mathbf{E}_b \left[C(\mathcal{M}(\tilde{D}, b)) \right]$
  - **Three attacks**
    - Parameter-targeting attack: make the model $\tilde{\theta}$ to *be close to* a target θ
    - Label-targeting attack: cause *small* prediction error on $\{z_i^*\}_{i \in [m]}$
    - Label-aversion attack: induce *large* prediction error on $\{z_i^*\}_{i \in [m]}$

- Capability
  - Modify k items in D

Training-time defenses: differential privacy

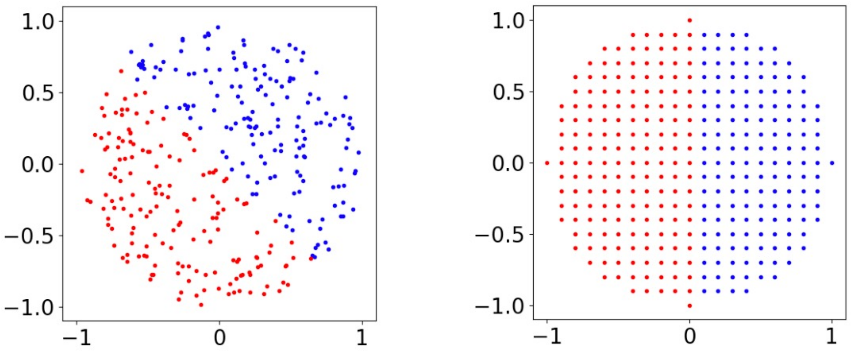
- DP as a poisoning defense
  - Construct the lower-bound $J(\widetilde{D}) \geq e^{-k\epsilon} J(D)$

- One-shot kill attack (single-poison attack)
  - $k = 1$: the lower bound becomes $J\left(\widetilde{D}\right) \geq e^{-\epsilon} J(D)$
  - $k \geq \lceil {}^{1}/_{\epsilon} \log \tau \rceil$ modification can achieve $J\left(\widetilde{D}\right) \geq {}^{1}/_{\tau} J(D)$
  - …

Evaluation

- Setup [binary classification tasks]
  - Dataset: Synthetic data | Real data (UCI ML Repo.)
  - Models: Logistic regression | Ridge-regression

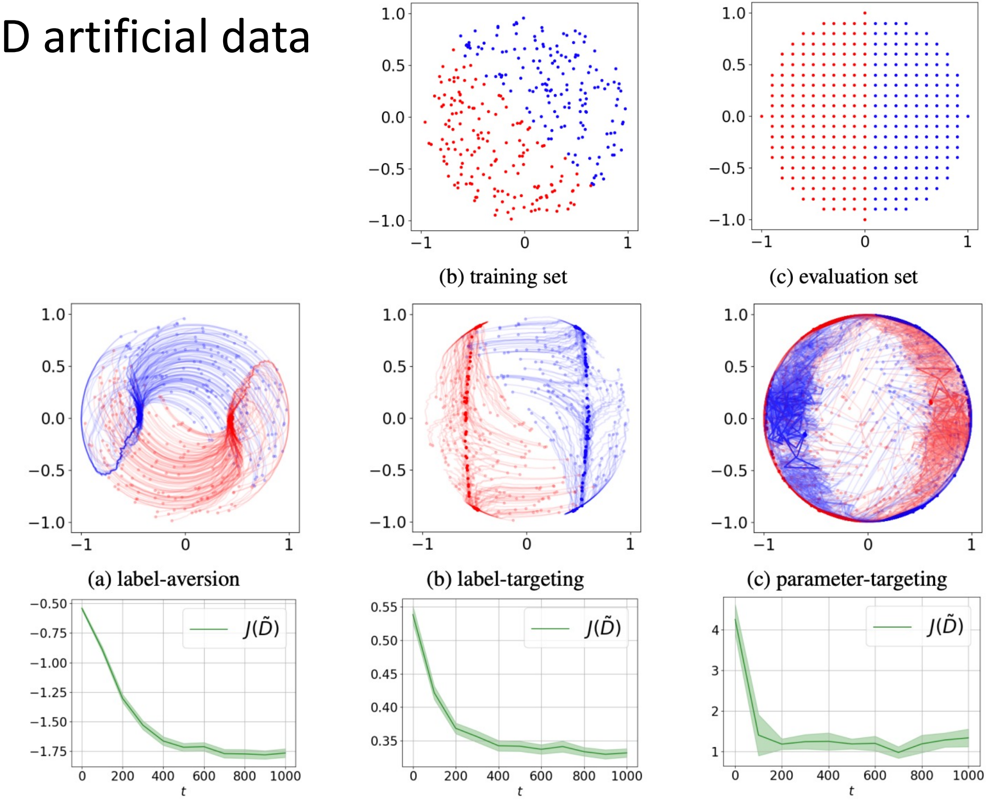
- Crafting poisons
  - Demonstrate on 2-D synthetic data



(b) training set (c) evaluation set

Evaluation

- Results of the three attacks on 2-D artificial data
  - Set $k = n$
  - Each attack achieves its objective



(b) training set

(c) evaluation set

(a) label-aversion

(b) label-targeting

(c) parameter-targeting

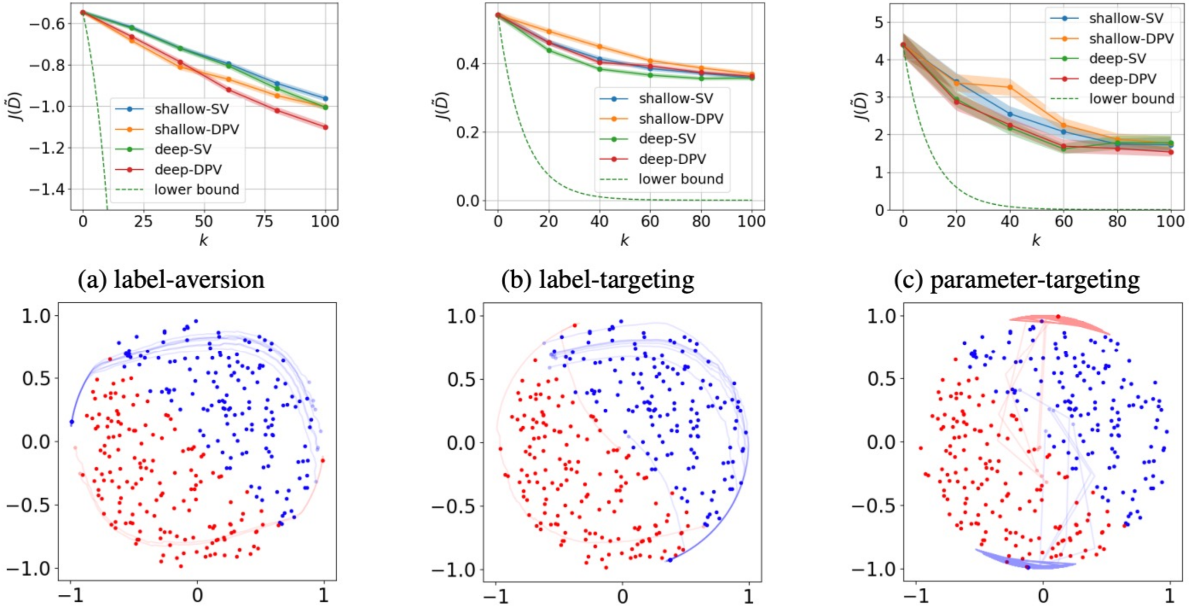
(d) label-aversion

(e) label-targeting

(f) parameter-targeting

Evaluation

- Results of the three attacks on 2-D artificial data
  - The attack cost decreases as k increases (the attack becomes easier!)

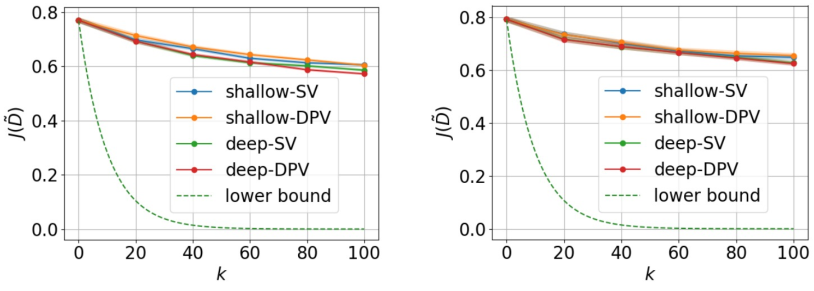


(a) label-aversion (b) label-targeting (c) parameter-targeting

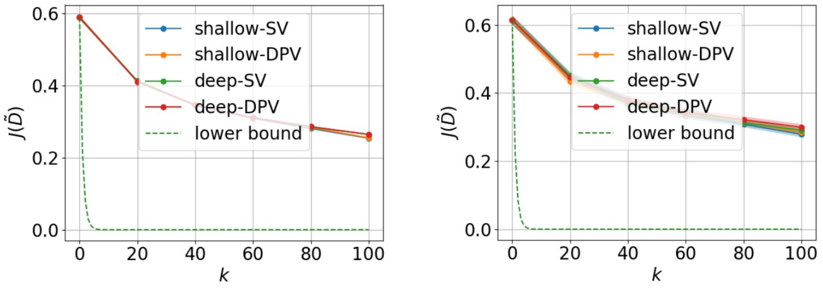
(d) label-aversion (e) label-targeting (f) parameter-targeting

Evaluation

- Results of the *label-targeting* attacks on real-world datasets
  - (left) vs. logistic regression, (right) vs. ridge regression
  - The attacks work well also on the DP learners
  - The gap between the lower bound and the actual attack success exists



(a) objective perturbation (b) output perturbation

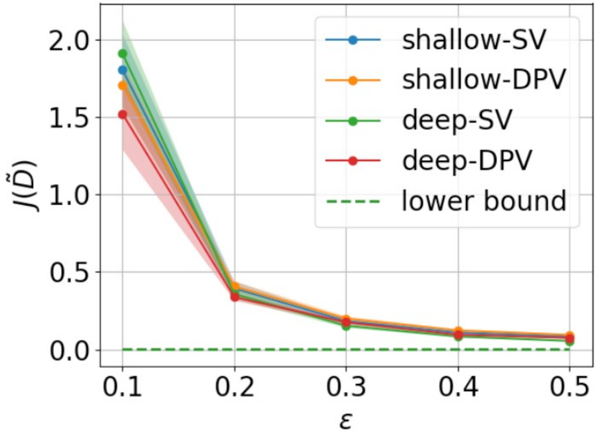


(a) objective perturbation (b) output perturbation

Evaluation

- Results of the *label-targeting* attacks on real-world datasets
  - In DP, the attack costs significantly higher than the case w/o DP
  - ex. with 20 poisons, the cost w/o DP is almost zero whereas with DP, it's 0.4

- Interesting Observation!
  - Attacks are much easier with weak (small epsilon) privacy



Thank You!

Tu/Th 4:00 – 5:50 pm

Sanghyun Hong

https://secure-ai.systems/courses/MLSec/F23

Oregon State University

SAIL
Secure AI Systems Lab