

Security and Fairness of Deep Learning

Privacy Attacks on Deep Networks

Klas Leino

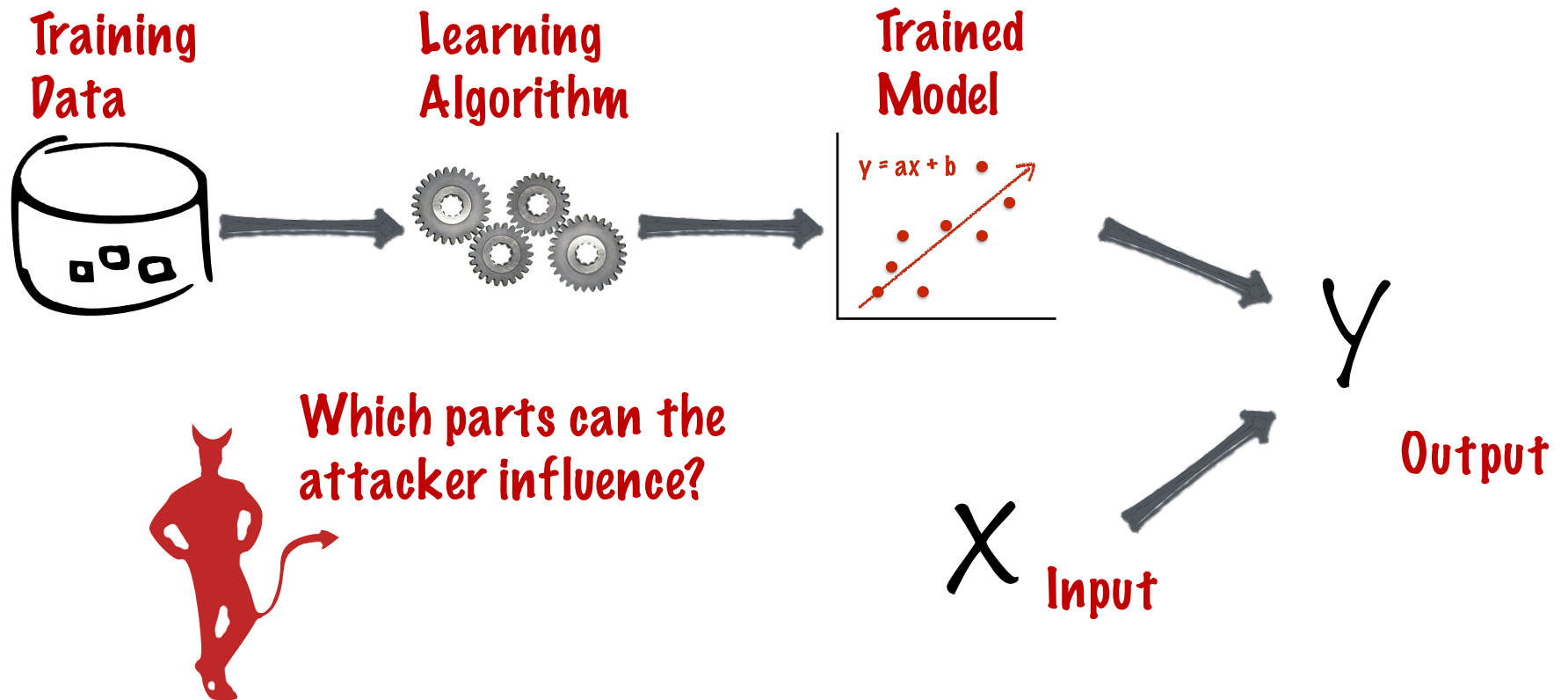
CMU

Spring 2019

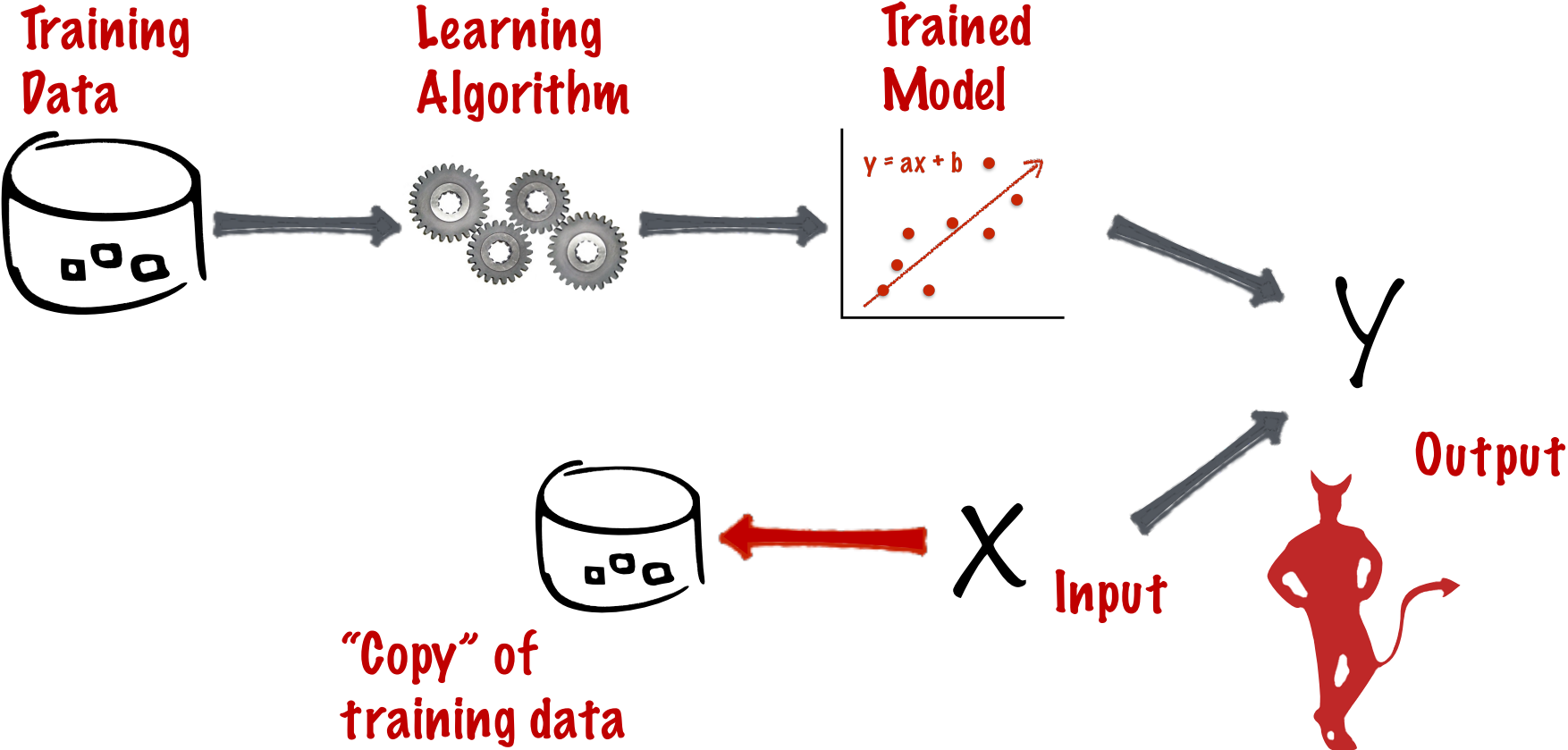
Overview

- Overview of Privacy Attacks
- Membership Inference
- Black-box Attacks
 - Naïve
 - Shadow Models
- Mitigation

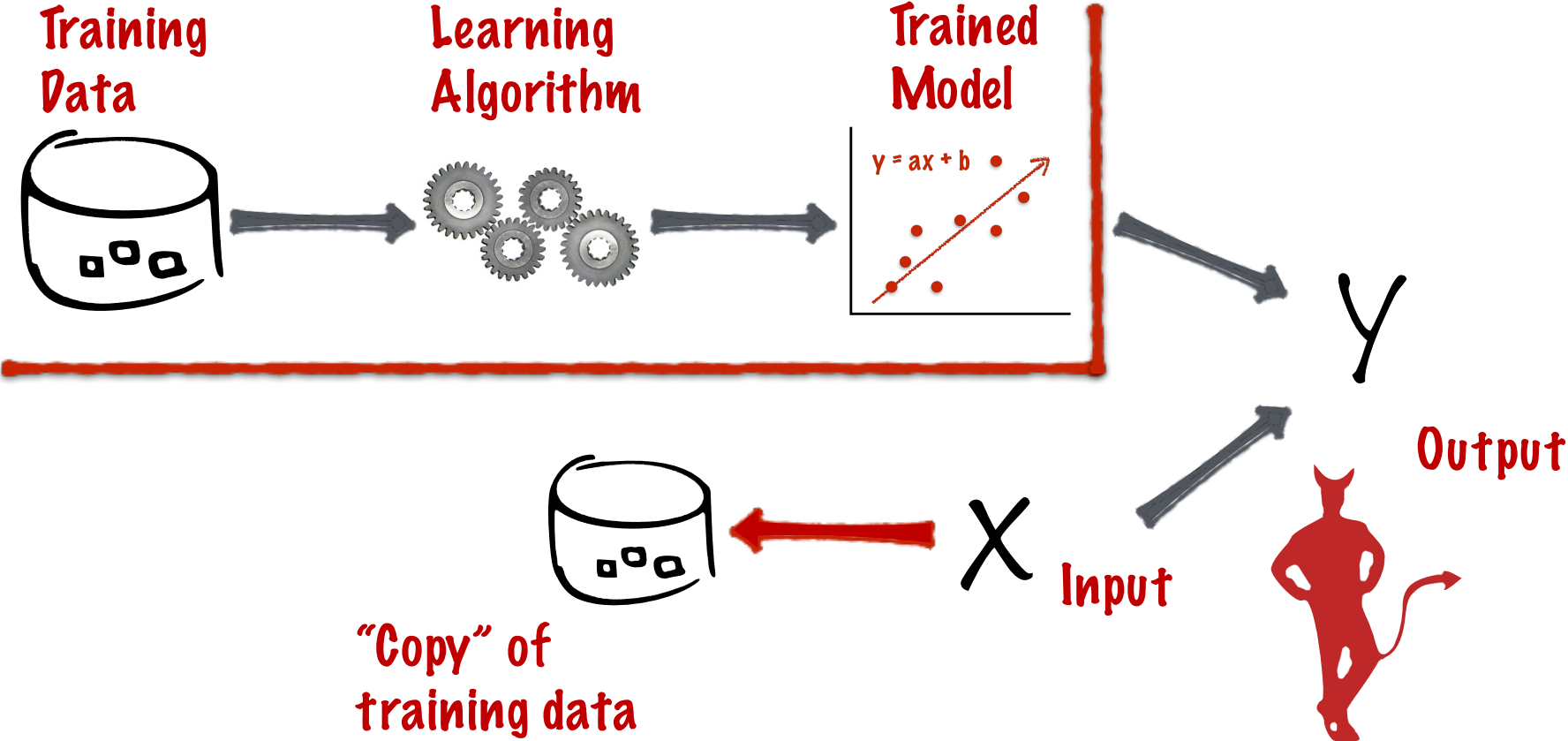
Machine Learning Pipeline



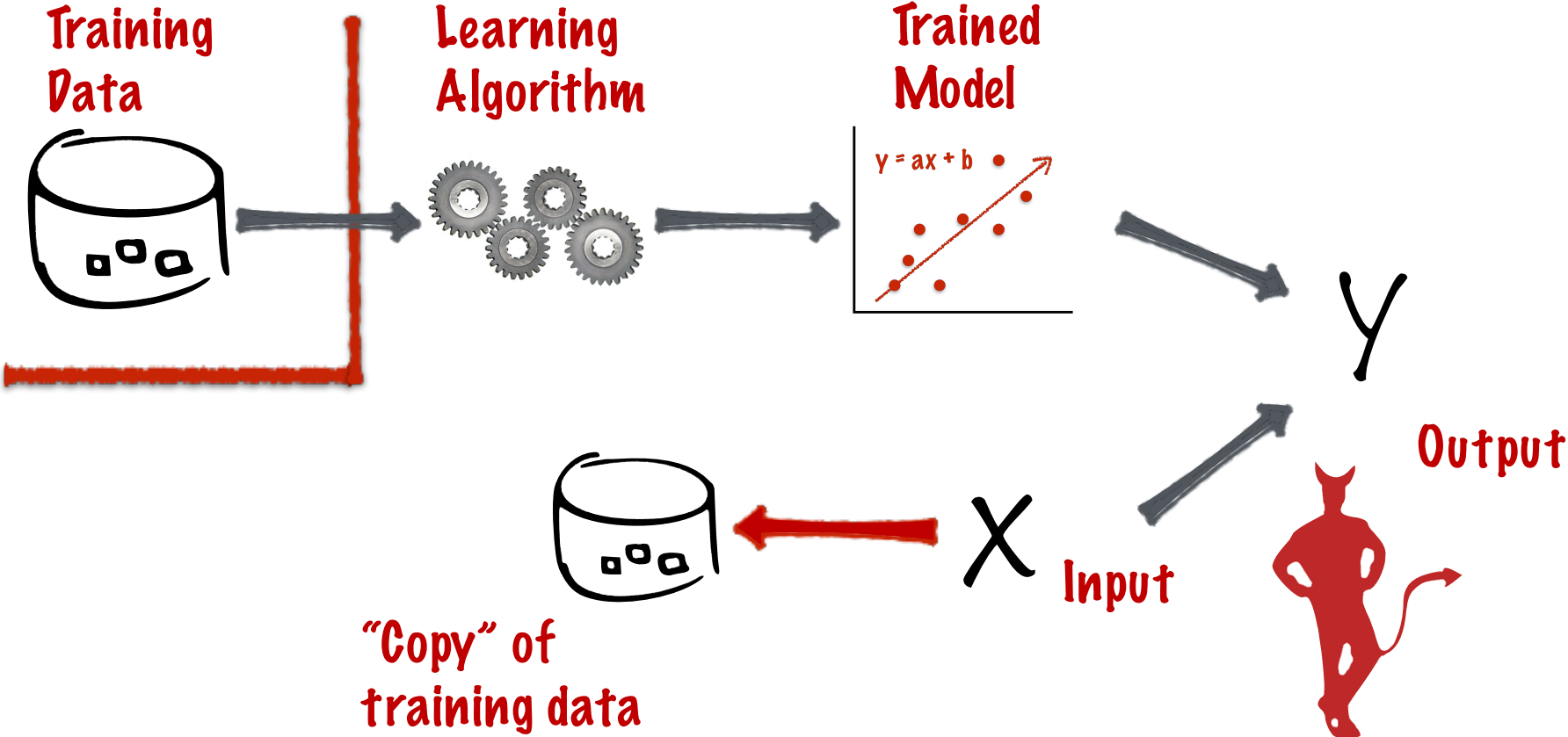
Threat: Data Privacy



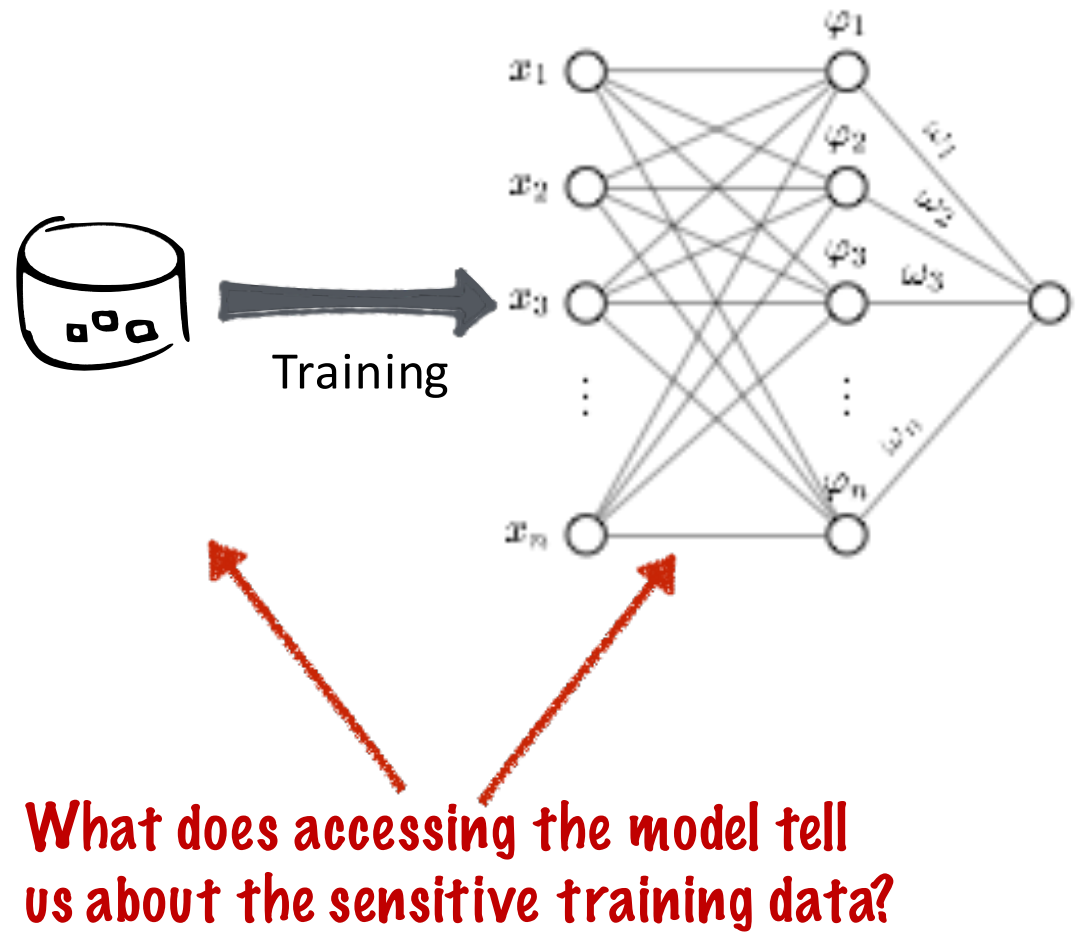
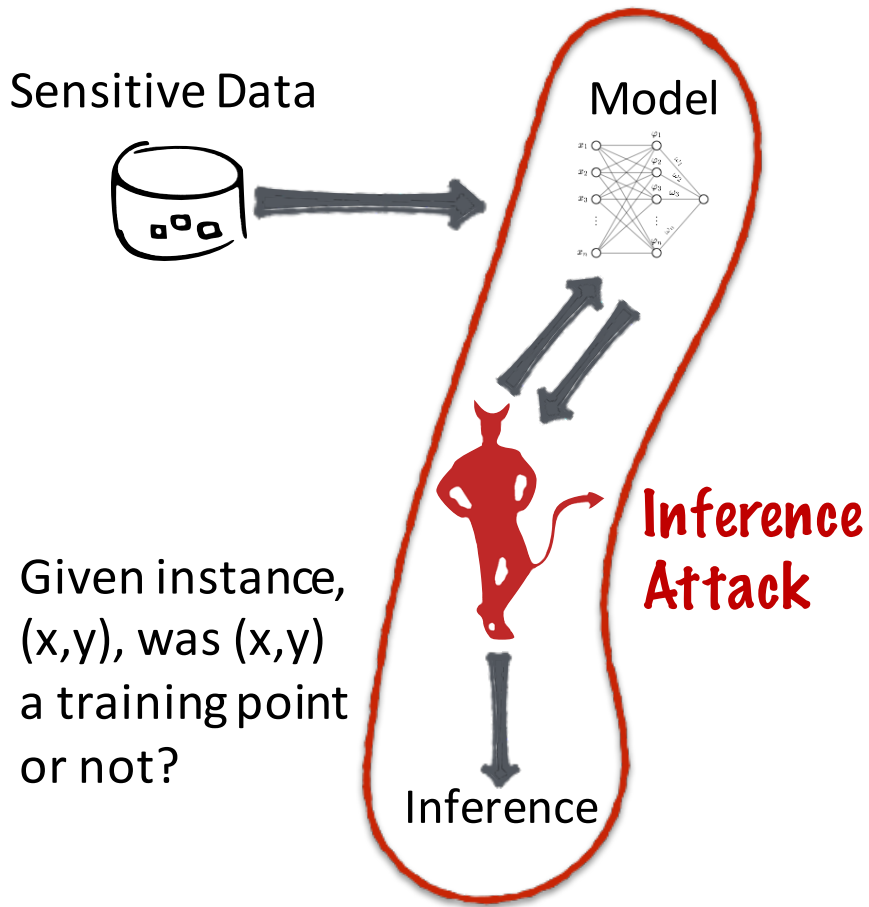
Threat: Data Privacy (black-box)



Threat: Data Privacy (white-box)



Inference Attacks

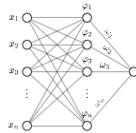


Example

Non-sensitive information of patients with sensitive condition X



Model predicting non-sensitive target, e.g., height of patient



Membership inference



Bob was in training set (has condition X)

Adversary learns Bob has condition X

Why do We Care About Membership Inference?

- Membership itself may be sensitive information (as in example)
- Ability to perform membership inference suggests leakage of (potentially sensitive) training data information
- Membership inference vulnerability linked to overfitting, generalization

Overview

- Overview of Privacy Attacks
- **Membership Inference**
- Black-box Attacks
 - Naïve
 - Shadow Models
- Mitigation

Membership Inference

- Universe, U , of points, (x, y) , of features (x) and labels ($y \in [C]$), distributed according to distribution, θ^* .
- Training set, S , of N points drawn from θ^* .
 - (x, y) drawn from the training set: (x, y) chosen uniformly at random from the elements of S .
 - (x, y) drawn from the general population (or test set): (x, y) drawn directly from θ^* .
- Target model, \hat{g} , learned by algorithm, \mathcal{A} , which takes a training set and produces a model.

Membership Inference: Threat Models

- Black-box: adversary has black-box access to \hat{g} , i.e., given features, x , the adversary can obtain $\hat{y} = \hat{g}(x)$.
 - Adversary doesn't have access to weights or internal activations.
 - Typically, we do assume adversary has access to \mathcal{A} .
 - We also assume adversary has access to some set of points, \tilde{S} , also drawn from θ^* (but disjoint from S).
- White-box: adversary additionally has access to the internals of \hat{g} , e.g., weights and biases.

Membership Inference

- Draw a point, (x, y) , with $\frac{1}{2}$ probability from the training set, and with $\frac{1}{2}$ probability from the general population.
- Adversary predicts 1 ((x, y) was a training point) or 0 ((x, y) was not a training point).
- Advantage: *true positive rate* – *false positive rate*, or equivalently, $2(\text{accuracy} - \frac{1}{2})$.

Overview

- Overview of Privacy Attacks
- Membership Inference
- **Black-box Attacks**
 - Naïve
 - Shadow Models
- Mitigation

Membership Inference & Overfitting

Intuitively, overfitting can lead to membership inference vulnerability. Suppose we have an overfit target model, \hat{g} , that gets 90% training accuracy and 75% test accuracy.

How might we attack this model?

Naïve Attack

- If $\hat{y} = y$, predict 1, else predict 0. In other words we assume correctly classified points are training members, and incorrectly classified points are not.
- Advantage: *train accuracy* – *test accuracy*.
- Surprisingly, this attack is quite effective, i.e., compares similarly to more sophisticated attacks.

What's Wrong with the Naïve Attack?

- High false positive rate (bad precision)
- Doesn't quantify confidence in inference

Overview

- Overview of Privacy Attacks
- Membership Inference
- **Black-box Attacks**
 - Naïve
 - **Shadow Models**
- Mitigation

Shadow Model Approach [1]

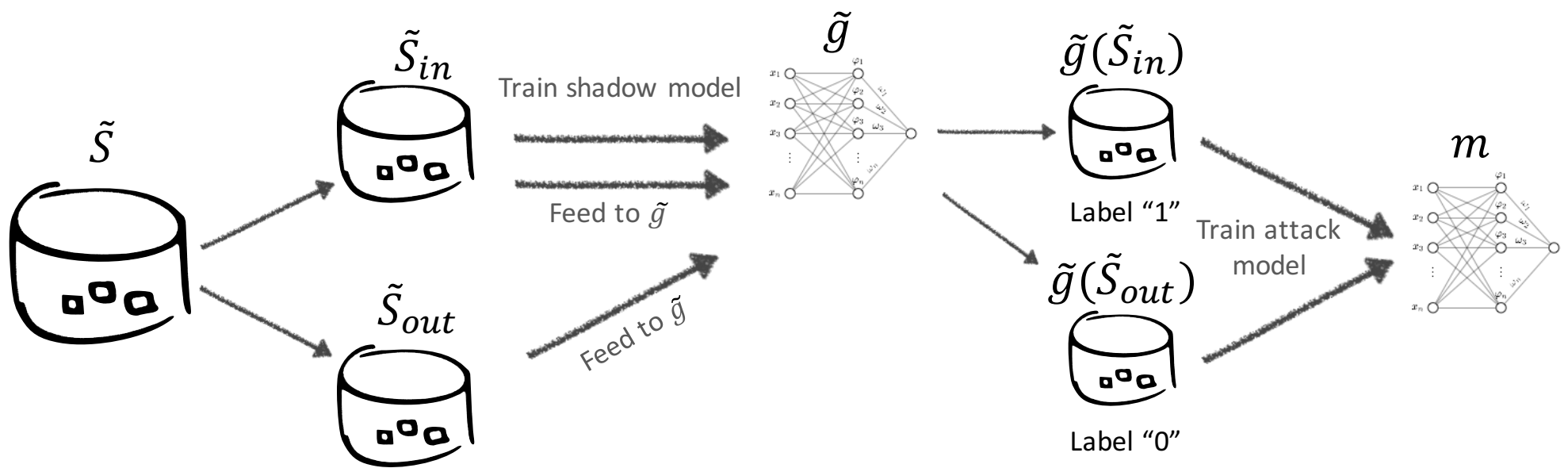
- Idea: frame membership inference as supervised learning problem.
- We would like to train an *attack model*, m , that, given an instance, predicts whether the instance is a training point or not.
 - Features: outputs of the model on the given instance, $\hat{y} = \hat{g}(x)$.
 - Learn one attack model, m^y , for each true class, $y \in [C]$.

How can we obtain labels to train each m^y ?

Shadow Model Approach

- Train a *shadow model*, \tilde{g} , on a subset of $\tilde{S}, \tilde{S}_{in}$, that is made to mirror \hat{g} (i.e., it is trained with \mathcal{A}).
- We know the exact training data used for \tilde{g} , therefore we can construct labels:
 - Label $\tilde{g}(x)$ with 1 for $x \in \tilde{S}_{in}$
 - Label $\tilde{g}(x)$ with 0 for $x \in \tilde{S} \setminus \tilde{S}_{in}$
- We assume that the patterns found in the outputs of \tilde{g} will apply to membership the same way on \hat{g} .

Illustration



(do this for each class)

Additions/Optimizations

- We can train multiple shadow models using different random splits of \tilde{S} to increase the size of our training set for m .
- Shokri et al. [1] also include the one-hot encoding of y as input.


Overview

- Overview of Privacy Attacks
- Membership Inference
- Black-box Attacks
 - Naïve
 - Shadow Models
- **Mitigation**

How Might we Mitigate Information Leakage?

- Decrease generalization error
 - Regularization
 - Dropout
- Add noise during training
 - Train using differentially-private algorithm

we will also see that well-generalized models can still be vulnerable to MI attacks



not shown to be highly effective



gives a provable guarantee against membership inference



Differential Privacy

Let ε be a positive real number, and \mathcal{A} be a randomized algorithm that takes a dataset as input and outputs a model. \mathcal{A} is ε -differentially private if, for all datasets D_1 and D_2 that differ on a single instance, and all $S \subseteq \text{image}(\mathcal{A})$,

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^\varepsilon \Pr[\mathcal{A}(D_2) \in S]$$

some set of particular models

probability of \mathcal{A} producing a model in S when x is included *probability of \mathcal{A} producing a model in S when x is replaced with x'*

Intuition: we have some instance $x \in D_1$ and we would like to ensure that the probability of \mathcal{A} producing a particular model (or set of models) is not increased by more than a factor of e^ε when the dataset contains x as when x is replaced by some x' .

Differential Privacy Guarantee

An adversary attacking target model, \hat{g} , trained with an ε -differentially private algorithm, can achieve an advantage of at most

$$e^\varepsilon - 1$$

Drawbacks of Differential Privacy

- In order to get a good guarantee, ϵ must be small.
- Differentially-private training tends to hurt model performance, often significantly – performance is worse the smaller ϵ is.
- In practice people use a large ϵ (e.g., Apple has used $\epsilon = 16$), losing the theoretical guarantee.

Next Time...

- Overview of Privacy Attacks
- Membership Inference
- Black-box Attacks
 - Naïve
 - Shadow Models
- Mitigation
- White-box attacks
- More thoughts on mitigation

References

- [1] Shokri et al. *Membership Inference Attacks on Deep Learning Models*. 2016
- [2] Dwork et al. *Differential Privacy*. 2006
- [3] Yeom et al. *Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting*. 2017