CS 499/579: Trustworthy ML
Model stealing

Tu/Th 4:00 – 5:50 pm

Sanghyun Hong

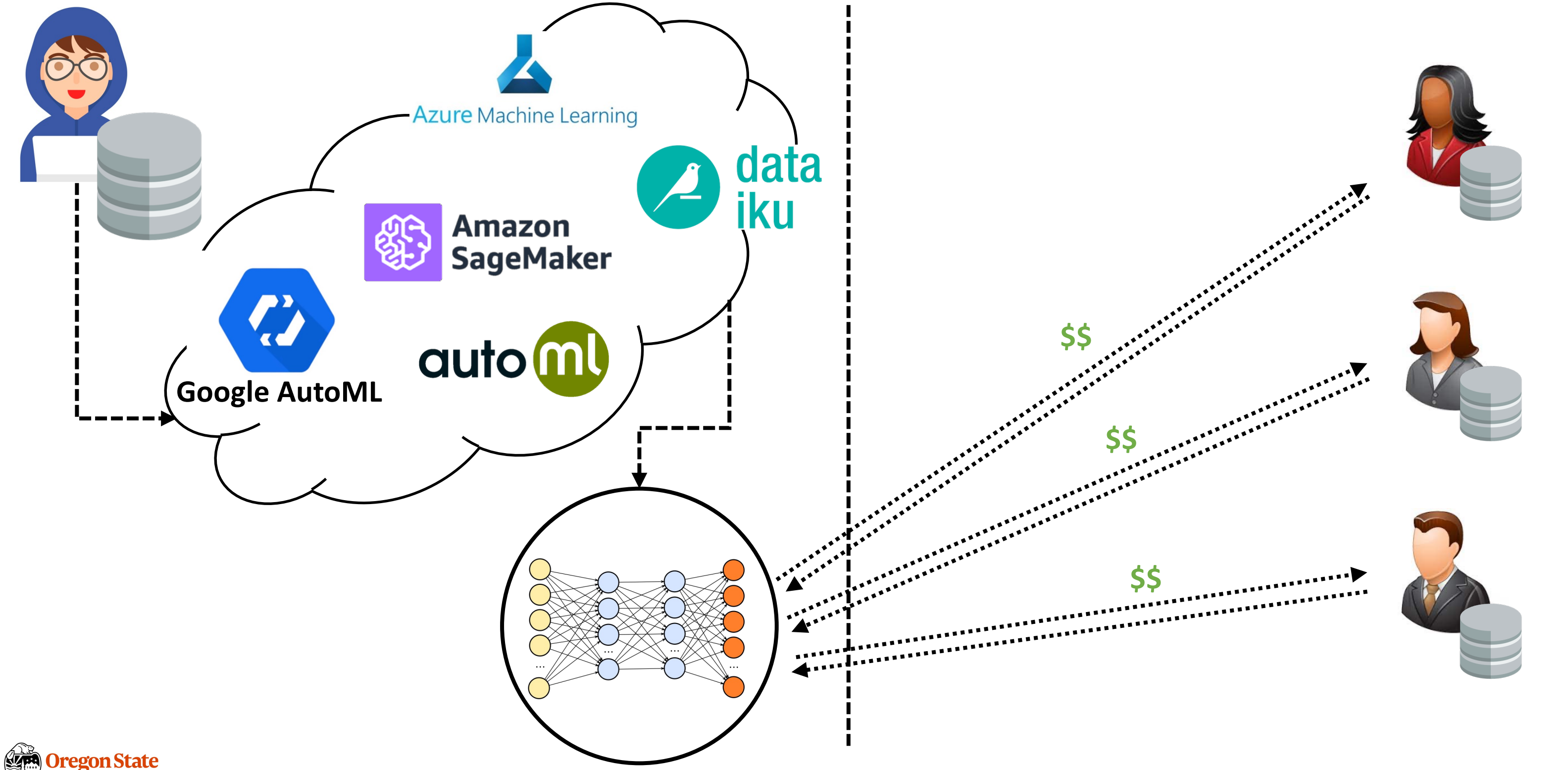
sanghyun.hong@oregonstate.edu

Oregon State University

SAIL
Secure AI Systems Lab

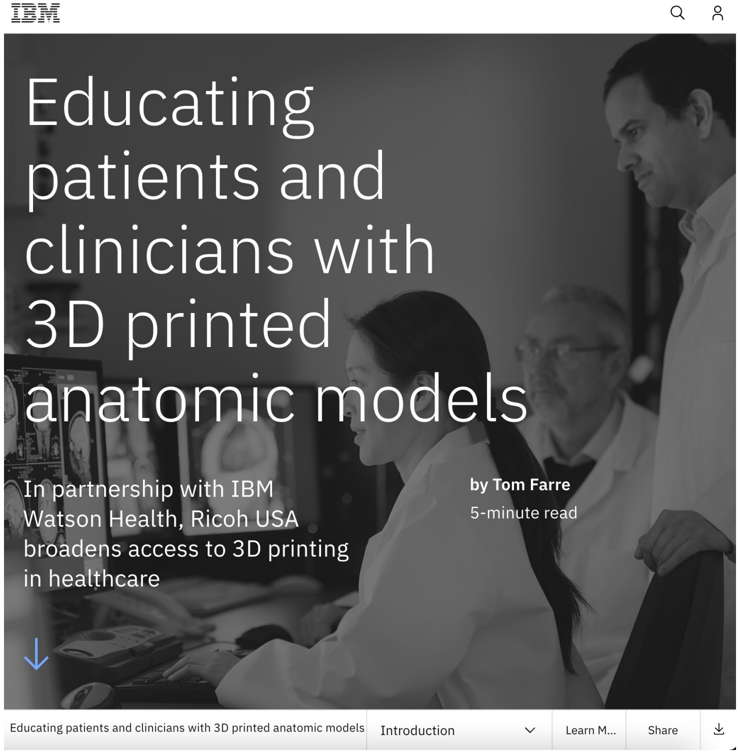
Emerging machine learning as a service (MLaaS)

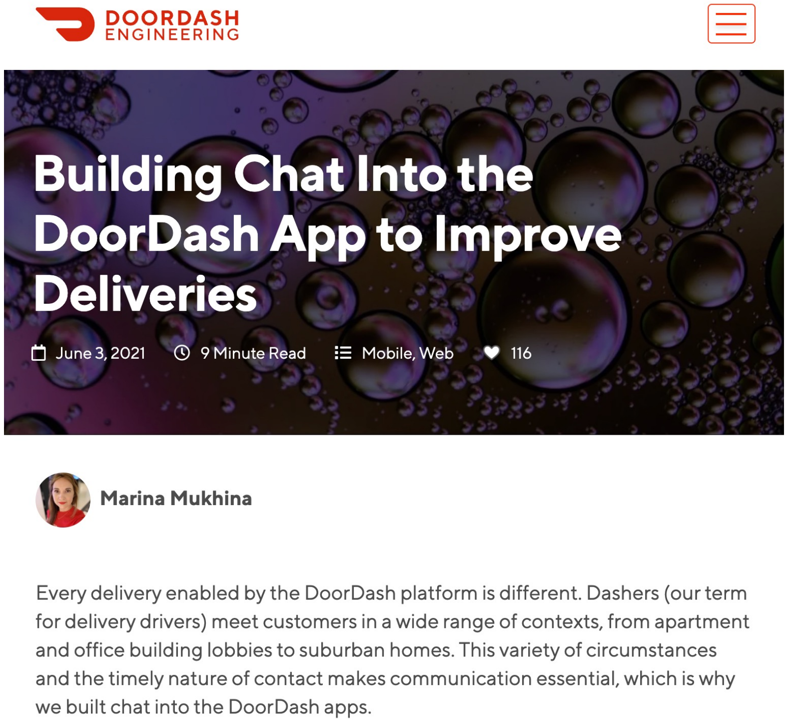
- You train ML models and reach out to customers



MLaaS incentivizes model extraction attackers

- Using **stolen** models… what if you run:





Potential downstream threats

- Exploiting stolen models, an adversary can:
  - Start a service with the stolen models with the same functionalities
  - Use the stolen model to craft adversarial examples
  - Extract private information from the stolen models

How can we steal your model?

Stealing machine learning models via prediction apis, Tramer et al., USENIX Security 2016

How can we do high-fidelity and high-accuracy extraction?

High accuracy and high-fidelity extraction of neural networks, Jagielski et al., USENIX Security 2020

Taxonomy of existing model extraction attacks

- Threat model
  - Goal: Theft + *Reconnaissance
    - Theft: extraction of a target model
    - Reconnaissance: conduct downstream attacks, such as adversarial attacks

*out of our scope

Taxonomy of existing model extraction attacks

- Threat model
  - Goal: Theft (extraction attack)
    - Functionally-equivalent extraction, $\forall x, \hat{O}(x) = O(x)$
    - Fidelity extraction $\Pr_{x \sim D}[S(\hat{O}(x), O(x))]$, where $S(\cdot)$ is the similarity function
    - Task-accuracy extraction $\Pr_{(x,y) \sim D}[\text{argmax}(\hat{O}(x)) = y]$

*out of our scope

Taxonomy of existing model extraction attacks

- Fidelity vs. task-accuracy
  - **Fidelity:** extracted model be *similar*
  - **Accuracy:** extracted model be *accurate*

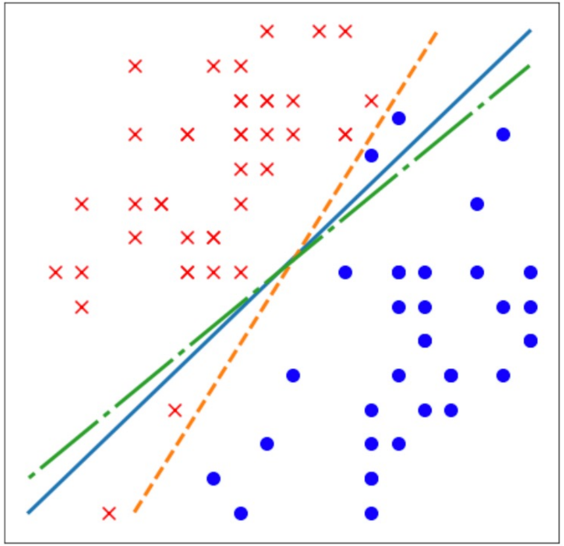


Figure 1: Illustrating fidelity vs. accuracy. The solid blue line is the oracle; functionally equivalent extraction recovers this exactly. The green dash-dot line achieves high fidelity: it matches the oracle on all data points. The orange dashed line achieves perfect accuracy: it classifies all points correctly.

Taxonomy of existing model extraction attacks

- Threat model
  - Goal: Theft (extraction attack)
    - Functionally-equivalent extraction, $\forall x, \hat{O}(x) = O(x)$
    - Fidelity extraction $\Pr_{x \sim D}[S(\hat{O}(x), O(x))]$, where $S(\cdot)$ is the similarity function
    - Task-accuracy extraction $\Pr_{(x,y) \sim D}[\mathrm{argmax}(\hat{O}(x)) = y]$

Attack	Type	Model type	Goal	Query Output
Lowd & Meek [8]	Direct Recovery	LM	Functionally Equivalent	Labels
Tramer *et al.* [11]	(Active) Learning	LM, NN	Task Accuracy, Fidelity	Probabilities, labels
Tramer *et al.* [11]	Path finding	DT	Functionally Equivalent	Probabilities, labels
Milli *et al.* [19] (theoretical)	Direct Recovery	NN (2 layer)	Functionally Equivalent	Gradients, logits
Milli *et al.* [19]	Learning	LM, NN	Task Accuracy	Gradients
Pal *et al.* [15]	Active learning	NN	Fidelity	Probabilities, labels
Chandrasekharan *et al.* [13]	Active learning	LM	Functionally Equivalent	Labels
Copycat CNN [16]	Learning	CNN	Task Accuracy, Fidelity	Labels
Papernot *et al.* [7]	Active learning	NN	Fidelity	Labels
CSI NN [25]	Direct Recovery	NN	Functionally Equivalent	Power Side Channel
Knockoff Nets [12]	Learning	NN	Task Accuracy	Probabilities
Functionally equivalent (this work)	Direct Recovery	NN (2 layer)	Functionally Equivalent	Probabilities, logits
Efficient learning (this work)	Learning	NN	Task Accuracy, Fidelity	Probabilities

*out of our scope

Functionally-equivalent extraction

- "Hard"
  - # of queries for extraction:
    - Suppose a neural network with $3k$-width and 2-depth
    - On d-dimensional domain with precision of p numbers
    - The attacker needs $\mathrm{O}(p^k)$ queries to perform a complete extraction
  - Check if two networks are the same
    - NP-hard problem
  - Learning-based approach struggles with fidelity
    - Suppose a deep random network with d-dimensional input and h-depth
    - Suppose an adversary formulated as statistical query (SQ) learning
    - Require $\exp(O(h))$ samples for fidelity extraction

Taxonomy of existing model extraction attacks

- Threat model
  - Goal: Theft (extraction attack)
    - Functionally-equivalent extraction, $\forall x, \hat{O}(x) = O(x)$
    - Fidelity extraction $\Pr_{x \sim D}[S(\hat{O}(x), O(x))]$, where $S(\cdot)$ is the similarity function
    - Task-accuracy extraction $\Pr_{(x,y) \sim D}[\mathrm{argmax}(\hat{O}(x)) = y]$
  - Knowledge
    - Domain knowledge:
      - The attacker has partial knowledge of the training dataset
      - They have some pretrained models in the same domain
    - Deployment knowledge
    - Model access

*out of our scope

Learning-based model extraction

- Fully-supervised model extraction
  - Setup:
    - Adversaries have access to some datasets
    - They use the victim model f as a labeling *oracle*
    - They train a separate model $\hat{f}$ on the oracle outputs
    - Objective is to make $\hat{f}$ and f achieve same test-time accuracy

  - Experimental setup:
    - Oracle: a model trained on 1B Instagram images (SoTA on ImageNet)
    - Attacker:
      - Case I: who has 10% (~13k) or 100% of the training samples (1B)
      - Case II: who improves the attack by using semi-supervised techniques (Rot. / MixMatch)

Learning-based model extraction

- Evaluation results
  - Results (+Rot.):
    - Oracle (84.2% Top-1 acc. / 97.2% in Top-5)
    - Extracted models show a high accuracy (81- 94%) and fidelity (83- 97%) in Top-5
    - Semi-supervised approaches (unlabeled data) improve the performance further

Architecture	Data Fraction	ImageNet	WSL	WSL-5	ImageNet + Rot	WSL + Rot	WSL-5 + Rot
Resnet_v2_50	10%	(81.86/82.95)	(82.71/84.18)	(82.97/84.52)	(82.27/84.14)	(82.76/84.73)	(82.84/84.59)
Resnet_v2_200	10%	(83.50/84.96)	(84.81/86.36)	(85.00/86.67)	(85.10/86.29)	(86.17/88.16)	(86.11/87.54)
Resnet_v2_50	100%	(92.45/93.93)	(93.00/94.64)	(93.12/94.87)	N/A	N/A	N/A
Resnet_v2_200	100%	(93.70/95.11)	(94.26/96.24)	(94.21/95.85)	N/A	N/A	N/A

Problem: Non-determinism!

Learning-based model extraction

- Evaluation results
  - Sources of non-determinism:
    - Initialization of model parameters
    - SGD (*random mini-batches)
  - Prior work on FE extraction attacks:
    - Milli *et al.*: *gradient queries*
    - Batina *et al.*: *power side-channel*

Query Set	Init & SGD	Same SGD	Same Init	Different
Test	93.7%	93.2%	93.1%	93.4%
Adv Ex	73.6%	65.4%	65.3%	67.1%
Uniform	65.7%	60.2%	59.0%	60.2%

Table 4: Impact of non-determinism on extraction fidelity. Even models extracted using the same SGD and initialization randomness as the oracle do not reach 100% fidelity.

Prior Work Assumes Too Strong Adversaries!

Taxonomy of existing model extraction attacks

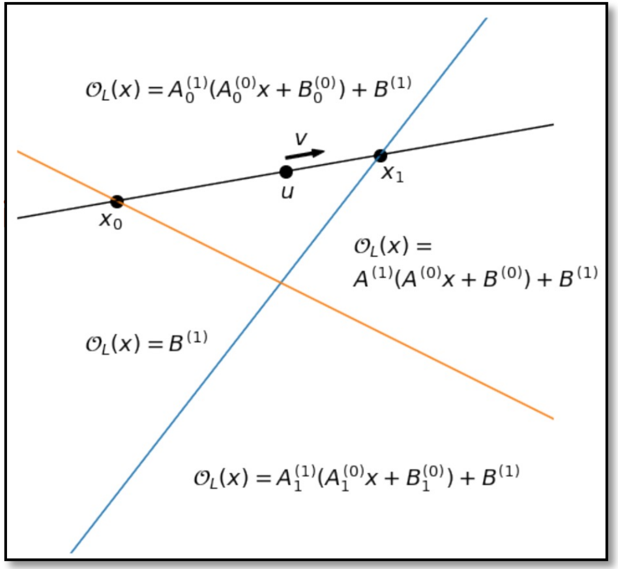
- Threat model
  - Goal: Theft (extraction attack)
    - Functionally-equivalent extraction, $\forall x, \hat{O}(x) = O(x)$
    - Fidelity extraction $\Pr_{x \sim D}[S(\hat{O}(x), O(x))]$, where $S(\cdot)$ is the similarity function
    - Task-accuracy extraction $\Pr_{(x,y) \sim D}[\operatorname{argmax}(\hat{O}(x)) = y]$
  - Knowledge
    - Domain knowledge:
      - The attacker has partial knowledge of the training dataset
      - They have some pretrained models in the same domain
    - Deployment knowledge
      - 2-layer feedforward neural network with ReLU activations
      - The architecture of a neural network is known (input-dim and hidden-dim)
    - Model access

*out of our scope

Functionally-equivalent model extraction

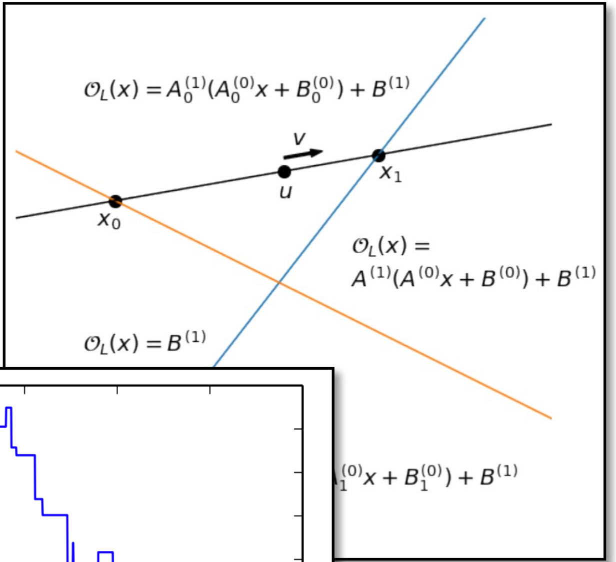
- Jagielski *et al.* attack
  - Intuition (ReLU)
    - A standard choice of activation functions
    - It makes neural networks piecewise-linear (let's explo

  - Attack procedures (on a 2-layer NN)
    - Critical point search
    - Weight recovery
    - Sign recovery
    - Final layer extraction



Model extraction attack

- Jagielski *et al.* attack
  - Attack procedures (on a 2-layer NN)
    - **Critical point search**
    - Weight recovery
    - Sign recovery
    - Final layer extraction



Algorithm 1 Algorithm for 2-linearity testing. Computes the location of the only critical point in a given range or rejects if there is more than one.

Function f, range $[t_1, t_2]$, ε

$m_1 = \frac{f(t_1+\varepsilon)-f(t_1)}{\varepsilon}$ ▷ Gradient at t_1

$m_2 = \frac{f(t_2)-f(t_2-\varepsilon)}{\varepsilon}$ ▷ Gradient at t_2

$y_1 = f(a), y_2 = f(b)$

$x = a + \frac{y_2 - y_1 - (b-a)m_2}{m_1 - m_2}$ ▷ Candidate critical point

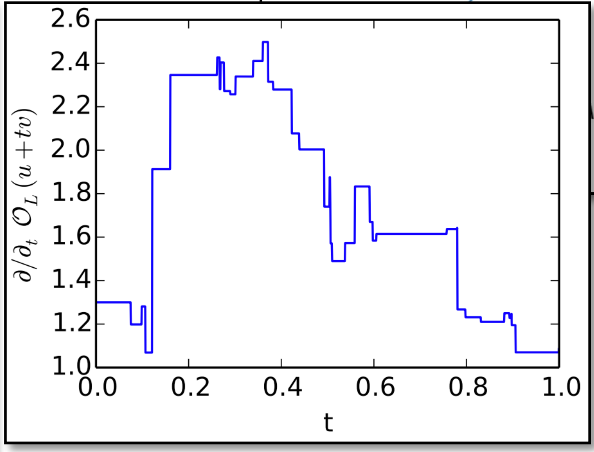
$\hat{y} = y_1 + m_1 \frac{y_2 - y_1 - (b-a)m_2}{m_1 - m_2}$ ▷ Expected value at candidate

$y = f(x)$ ▷ True value at candidate

if $\hat{y} = y$ **then return** x

else return "More than one critical point"

end if



Model extraction attacks

- Jagielski *et al.* attack
  - Attack procedures (on a 2-layer NN)
    - Critical point search
    - **Weight recovery**
      - Compute second derivatives
      - Estimate the ratio between two weight vectors w_1, w_2
    - Sign recovery
    - Final layer extraction

$$
\begin{aligned}
\left.\frac{\partial^2 O_L}{\partial e_j^2}\right|_{x_i} &= \left.\frac{\partial O_L}{\partial e_j}\right|_{x_i+c\cdot e_j} - \left.\frac{\partial O_L}{\partial e_j}\right|_{x_i-c\cdot e_j} \\
&= \sum_k A_k^{(1)} \mathbb{1}(A_k^{(0)}(x_i + c\cdot e_j) + B_k^{(0)} > 0) A_{kj}^{(0)} \\
&\quad - \sum_k A_k^{(1)} \mathbb{1}(A_k^{(0)}(x_i - c\cdot e_j) + B_k^{(0)} > 0) A_{kj}^{(0)} \\
&= A_i^{(1)} \Big(\mathbb{1}(A_i^{(0)} \cdot e_j > 0) - \mathbb{1}(-A_i^{(0)} \cdot e_j > 0)\Big) A_{ji}^{(0)} \\
&= \pm(A_{ji}^{(0)} A_i^{(1)})
\end{aligned}
$$

Model Extraction Attacks

- Jagielski *et al.* attack
  - Attack procedures (on a 2-layer NN)
    - Critical point search
    - Weight recovery
    - **Sign recovery**
    - **Final layer extraction**

$$\left.\frac{\partial^2 O_L}{\partial(e_j + e_k)^2}\right|_{x_i} = \pm(A_{ji}^{(0)} A_i^{(1)} \pm A_{ki}^{(0)} A_i^{(1)}).$$

EVALUATION

- Proposed attacks
  - **Setup:**
    - Datasets: MNIST and CIFAR-10
    - Models: 2-layer NN, 16 – 512 hidden units (~12 – 100k params)
  - **Results:**
    - MNIST:
      - 100% fidelity on the test-set
      - $2^{17.2} - 2^{20.2}$ queries for the 100% fidelity
    - CIFAR-10:
      - 100% fidelity on the test-set for models with < 200k params
      - 99% for the models with > 200k params
      - $2^{17.2} - 2^{20.2}$ queries for the 100% fidelity

Evaluation

- Hybrid strategies
  - **Setup:**
    - Learning-based extraction with gradient matching
    - Error-recovery through learning
  - **Results:**
    - MNIST:
      - with 4x times larger models
      - 99-100% fidelity on the test-set
      - $2^{19.2} - 2^{22.2}$ queries for the 100% fidelity (improvement over the previous results $2^{17.2} - 2^{20.2}$)

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