Extraction of Complex DNN Models
Real Threat or Boogeyman?

N. Asokan

https://asokan.org/asokan/
@nasokan

(Joint work with Buse Gul Atli, Sebastian Szyller, Mika Juuti, Jian Liu, Rui Zhang, and Samuel Marchal)
Model Stealing Attacks and Defenses
Where are we now?

N. Asokan
https://asokan.org/asokan/
@nasokan

(Joint work with Buse Gul Atli, Sebastian Szyller, Mika Juuti, Jian Liu, Rui Zhang, and Samuel Marchal)
What are the challenges in making AI systems trustworthy?

Is model stealing an important concern?

Can models be extracted via their inference APIs?

What can be done to counter model theft?

Can we simultaneously deploy protections against multiple concerns?
Outline

What are the challenges in making AI systems trustworthy?

Is model stealing an important concern?

Can models be extracted via their inference APIs?

What can be done to counter model theft?

Can we simultaneously deploy protections against multiple concerns?
AI will be pervasive

https://www.fortunebusinessinsights.com/industry-reports/artificial-intelligence-market-100114
How Artificial Intelligence Is Advancing Precision Medicine

Nicole Martin
Former Contributor
I write about digital marketing, data and privacy concerns.


Dozens of Cities Have Secretly Experimented With Predictive Policing Software

Documents obtained by Motherboard requests verify previously unconfirmed with predictive policing company Rabbit.

By Caroline Haskins


AI is changing everything about cybersecurity, for better and for worse. Here's what you need to know

Artificial intelligence and machine learning tools could go a long way to helping to fight cybercrime. But these technologies aren't a silver bullet, and could also be exploited by malicious hackers.

Challenges in making AI trustworthy

Security concerns

Privacy concerns

Fairness and explainability concerns
Evading machine learning models

Which class is this?
School bus

Which class is this?
Ostrich

Machine Learning pipeline

Where is the adversary? What is its target?
Compromised input – Model integrity

Malicious client – Training data privacy

Invert model, infer membership

Fredrikson et al. - *Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*, ACM CCS ‘15
Malicious client – Model confidentiality

Tramer et al. - Stealing ML models via prediction APIs, Usenix SEC '16 (https://arxiv.org/abs/1609.02943)
Malicious inference service – User profiles

Data owners → Dataset → Trainer → ML model → Database

Add: “X uses app”

Is this app malicious?

Malmi and Weber - You are what apps you use Demographic prediction based on user's apps, ICWSM ‘16 (https://arxiv.org/abs/1603.00059)
Dowlin et al. - CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy, ICML ‘16 (https://dl.acm.org/doi/10.5555/3045390.3045413)
Compromised toolchain – Training data privacy

Malicious data owner – Model integrity

Influence ML model (model poisoning)

Is malicious adversarial behaviour the only concern?

*Predictive policing algorithms are racist. They need to be dismantled.*

Lack of transparency and biased training data mean these tools are not fit for purpose. If we can’t fix them, we should ditch them.

*by Will Douglas Heaven*  
*July 17, 2020*  

*AI is sending people to jail—and getting it wrong*

Using historical data to train risk assessment tools could mean that machines are copying the mistakes of the past.

*by Karen Hao*  
*January 21, 2019*  

---

*BBC*  
[Link](https://www.bbc.com/news/technology-54234822?fbclid=IwAR1T41_HR6IluMKGRJbJdDrdpKdyAi5mhQsdzs0QLDa41TSR3wJfs)

*MIT Technology Review*  
[Link](https://www.technologyreview.com/2019/01/21/137783/algorithms-criminal-justice-ai/)

[Link](https://www.technologyreview.com/2020/07/17/1005396/predictive-policing-machine-learning-bias-criminal-justice/)
Measures of accuracy are flawed, too

I wonder if Twitter does this to fictional characters too.

Lenny

Carl

We tested for bias before shipping the model & didn’t find evidence of racial or gender bias in our testing. But it’s clear that we’ve got more analysis to do. We’ll continue to share what we learn, what actions we take, & will open source it so others can review and replicate.

https://twitter.com/TwitterComms/status/1307739940424359936

Transparency around image cropping and changes to come

By Parag Agrawal and Dorsey Davis
Thursday, 1 October 2020

We’re always striving to work in a way that’s transparent and easy to understand, but we don’t always get this right. Recent conversation around our photo cropping methods brought this to the forefront, and over the past week, we’ve been reviewing the way we test for bias in

Towards trustworthy AI

Secure, privacy-preserving, fair, and explainable

<table>
<thead>
<tr>
<th>Which attack would affect your org the most?</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisoning (e.g.: [21])</td>
<td>10</td>
</tr>
<tr>
<td>Model Stealing (e.g.: [22])</td>
<td>6</td>
</tr>
<tr>
<td>Model Inversion (e.g.: [23])</td>
<td>4</td>
</tr>
<tr>
<td>Backdoored ML (e.g.: [24])</td>
<td>4</td>
</tr>
<tr>
<td>Membership Inference (e.g.: [25])</td>
<td>3</td>
</tr>
<tr>
<td>Adversarial Examples (e.g.: [26])</td>
<td>2</td>
</tr>
<tr>
<td>Reprogramming ML System (e.g.: [27])</td>
<td>0</td>
</tr>
<tr>
<td>Adversarial Example in Physical Domain (e.g.: [28])</td>
<td>0</td>
</tr>
<tr>
<td>Malicious ML provider recovering training data (e.g.: [28])</td>
<td>0</td>
</tr>
<tr>
<td>Attacking the ML supply chain (e.g.: [24])</td>
<td>0</td>
</tr>
<tr>
<td>Exploit Software Dependencies (e.g.: [29])</td>
<td>0</td>
</tr>
</tbody>
</table>

Outline

What are the challenges in making AI systems trustworthy?

Is model stealing an important concern?

Can models be extracted via their inference APIs?

What can be done to counter model theft?

Can we simultaneously deploy protections against multiple concerns?
Is model stealing an important concern?

Machine learning models: **business advantage and intellectual property (IP)**

**Cost of**
- gathering relevant data
- labeling data
- expertise required to choose the right model training method
- resources expended in training

**Adversary who steals the model can avoid these costs**

“Steal” = derive model from someone else’s model without their consent to do so
Type of model access: white box

White-box access: user
• has physical access to model
• knows its structure
• can observe execution (scientific packages, software on user-owned devices)
How to prevent (white-box) model theft?

White-box model theft can be countered by

• Computation with encrypted models

• Protecting models using secure hardware

• Hosting models behind a firewalled cloud service
Type of model access: black-box

Black-box access: user
• does not have physical access to model
• interacts via a well-defined interface (“inference API”):
  • directly (translation, image classification)
  • indirectly (recommender systems)

Basic idea: hide model, expose model functionality only via an inference API

Is that enough to prevent model theft?
Outline

What are the challenges in making AI systems trustworthy?

Is model stealing an important concern?

Can models be extracted via their inference APIs?

What can be done to counter model theft?

Can we simultaneously deploy protections against multiple concerns?
Extracting models via their inference APIs

Inference APIs are oracles that leak information

Adversary

- Malicious client
- Goal: construct surrogate model(*) comparable w/ functionality
- Capability: access to inference API or model outputs

(*) aka “student model” or “imitation model”

Prior work on extracting

- Logistic regression, decision trees[1]
- Simple convolutional neural network models[2]
- Querying API with synthetic samples

Extracting deep neural networks

Against simple deep neural network models\[1\]
• E.g., MNIST, GTSRB

Adversary
• knows general structure of the model
• has limited natural data from victim’s domain

Approach
• Hyperparameters CV-search
• Query using natural data for rough estimate decision boundaries, synthetic data to fine-tune
• Simple defense: distinguish between benign and adversarial queries

Is model extraction a realistic threat?

Can adversaries extract complex DNNs successfully?

Are common adversary models realistic?

Are current defenses effective?
Extraction of complex DNN models: Knockoff nets\textsuperscript{[1]}

Goal:

• Build a surrogate model that
  • steals model functionality of victim model
  • performs similarly on the same task with high classification accuracy

Adversary capabilities:

• Victim model knowledge:
  • None of train/test data, model internals, output semantics
  • Access to full prediction probability vector
  • Access to natural samples, not (necessarily) from the same distribution as train/test data
  • Access to pre-trained high-capacity model

\textsuperscript{[1]} Orekondy et al. - Knockoff Nets: Stealing Functionality of Black-Box Models, CVPR '19 (https://arxiv.org/abs/1812.02766)
Analysis of Knockoff Nets: summary[2]

Reproduced empirical evaluation of Knockoff nets[1] to confirm its effectiveness

Revisited its adversary model in to make more realistic assumptions about the adversary

Attack effectiveness decreases if
• Surrogate and victim model architectures are different
• Victim model’s inference API has reduced granularity

Simple defense: detector to identify out-of-distribution queries

Defense ineffective if attacker has natural samples distributed like victim’s training data

Extracting style-transfer models

GANS are effective for changing image style
- coloring, face filters, style application
Core feature in generative art and in social media apps
- **Selfie2Anime, FaceApp**
Style transfer

Task 1
Monet painting

Task 2
Anime face

Extracting natural language processing models

Techniques for extracting image classifiers don’t always extend to language models.

Transfer learning from pre-trained models is now very popular
• But they make model extraction easier\(^1\)

Krishna et al\(^1\) show that a Knockoff-like attacks against BERT models are feasible
• Adversary unaware of target distribution or task of victim model
• Adversary queries are merely “natural” (randomly sampled sequences of words)
• In-distribution adversary queries can improve extraction efficacy

Wallace et al\(^2\) extract real-world MT models, find transferable adversarial examples

\(^1\) Krishna et al. – Thieves on Sesame Street! Model Extraction of BERT-based APIs, ICLR ‘20 (https://iclr.cc/virtual_2020/poster_Byl5NREFDr.html)
Save me it's over 100°F
Save me it's over 102°F

Rette mich, es ist über 100 ° F.
Rette mich, es ist über 22 ° C.
Extracting large language models

The genie escapes: Stanford copies the ChatGPT AI for less than $600

By Loz Blain
March 19, 2023

https://newatlas.com/technology/stanford-alpaca-cheap-gpt/

STANFORD PULLS DOWN CHATGPT CLONE AFTER SAFETY CONCERNS

THEY CLONED A LITTLE TOO MUCH OF CHATGPT’S CAPABILITIES.

https://futurism.com/the-byte/stanford-pulls-down-chatgpt-clone

GOOGLE DENIES CLAIM THAT BARD WAS TRAINED BY STEALING CHATGPT DATA

GOOGLE, PLAY "RUMORS" BY LINDSAY LOHAN.

https://futurism.com/the-byte/google-denies-bard-openai
Outline

What are the challenges in making AI systems trustworthy?

Is model stealing an important concern? Yes

Can models be extracted via their inference APIs? Yes\(^1\)

- A powerful (but realistic) adversary can extract complex real-life models
- Detecting such an adversary is difficult/impossible

What can be done to counter model theft?

Can we simultaneously deploy protections against multiple concerns?

---

Outline

What are the challenges in making AI systems trustworthy?

Is model stealing an important concern? Yes

Can models be extracted via their inference APIs? Yes\(^1\)
- A powerful (but realistic) adversary can extract complex real-life models
- Detecting such an adversary is difficult/impossible

What can be done to counter model theft?

Can we simultaneously deploy protections against multiple concerns?

Defending against model theft

We can try to:
• **prevent** (or slow down\[1\]) model extraction, or
• **detect**\[2\] it

But current solutions are not effective

Or **deter** attackers by providing the means for **model ownership resolution (MOR):**
• model watermarking
• data watermarking
• fingerprinting

---

\[1\] Dziedzic et al. - *Increasing the Cost of Model Extraction with Calibrated Proof of Work*, ICLR '22 (https://openreview.net/pdf?id=EAy7C1cgE1L)

White-box watermarking

Watermark embedding:
• Embed the watermark in the model **during training:**
  • Choose incorrect labels for a set of samples (**watermark set, WM**)
  • Train using training data + **watermark set**

Verification of ownership:
• Adversary publicly exposes the stolen model
• Query the model with the **watermark set**
• **Verify** watermark - predictions correspond to chosen labels

---

Yadi et al. - *Watermarking Deep Neural Networks by Backdooring*, Usenix SEC ‘18 [https://www.usenix.org/node/217594](https://www.usenix.org/node/217594)
Existing watermarking of DNNs

Assumes that the model is stolen exactly (white-box theft)
Protects only against physical theft of model\(^1\)

**Not robust** against

- novel watermark removal attacks\(^2\)
- model extraction attacks that reduce effect of watermarks & modify decision surface

---


DAWN: Dynamic Adversarial Watermarking of DNNs[1]

Goal: Watermark models obtained via model extraction

Our approach:
• Implemented as part of the prediction API
• Return incorrect predictions for several samples
• Adversary forced to embed watermark while training

Watermarking evaluation:
• Unremovable and indistinguishable
• Defend against PRADA[2] and KnockOff[3]
• Preserve victim model utility (0.03-0.5% accuracy loss)

Data/Model fingerprinting

Radioactive data[1]
- Intended for provenance, not robust in adversarial settings[2]

Conferrable adversarial examples[3]
- Computationally expensive

Dataset inference[4]
- Susceptible to False positives? [5]

Outline

What are the challenges in making AI systems trustworthy?

Is model stealing an important concern?

Can models be extracted via their inference APIs?

What can be done to counter model theft?

Can we simultaneously deploy protections against multiple concerns?
Other ML security & privacy concerns

There are considerations other than model ownership:
- model evasion (defense: adversarial training)
- training data reconstruction (defense: differential privacy)
- membership inference (defense: regularization, early stopping)
- model poisoning (defense: regularization, outlier/anomaly detection)
- ...

How does ownership demonstration interact with the other defenses?

We investigate pairwise interactions of:

- model watermarking \hspace{1cm} \textbf{WITH} \hspace{1cm} differential privacy
- data watermarking \hspace{1cm} \textbf{WITH} \hspace{1cm} adversarial training
- fingerprinting
Setup & Baselines

We use the following techniques (and corresponding metrics):
- WM: Out-of-distribution (OOD) backdoor watermarking (test and watermark accuracy)
- RAD-DATA: Radioactive data (test accuracy and loss difference)
- DI: Dataset Inference (verification confidence)
- DP: DP-SGD (model accuracy for the given epsilon)
- ADV-TR: Adversarial training with PGD (test and adv. accuracy for the given epsilon)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No defense</th>
<th>Watermarking</th>
<th>Radioactive Data</th>
<th>Dataset Inference</th>
<th>DP-SGD (eps=3)</th>
<th>ADV. TR.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\phi_{ACC}$</td>
<td>$\phi_{ACC}$</td>
<td>$\phi_{WM}$</td>
<td>Loss Diff.</td>
<td>Confidence</td>
<td>$\phi_{ACC}$</td>
</tr>
<tr>
<td>MNIST</td>
<td>0.99±0.00</td>
<td>0.99±0.00</td>
<td>0.97±0.01</td>
<td>0.98±0.00</td>
<td>0.284±0.001</td>
<td>&lt;e-30</td>
</tr>
<tr>
<td>FMNIST</td>
<td>0.91±0.00</td>
<td>0.87±0.02</td>
<td>0.99±0.02</td>
<td>0.88±0.01</td>
<td>0.19±0.002</td>
<td>&lt;e-30</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>0.92±0.00</td>
<td>0.82±0.00</td>
<td>0.97±0.02</td>
<td>0.85±0.00</td>
<td>0.20±0.001</td>
<td>&lt;e-30</td>
</tr>
</tbody>
</table>
Interaction with differential privacy

Differential privacy is a strong per-sample regulariser:
- **Watermarking** rendered ineffective
- Lower but still **sufficient confidence** for radioactive data
- **No effect** on the DI fingerprint

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No defense</th>
<th>Watermarking</th>
<th>Radioactive Data</th>
<th>Dataset Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Baseline</td>
<td>with DP</td>
<td>Baseline</td>
</tr>
<tr>
<td></td>
<td>$\phi_{ACC}$</td>
<td>$\phi_{ACC}$</td>
<td>$\phi_{WM}$</td>
<td>$\phi_{ACC}$</td>
</tr>
<tr>
<td>MNIST</td>
<td>0.99±0.00</td>
<td>0.99±0.00</td>
<td>0.97±0.01</td>
<td>0.97±0.00</td>
</tr>
<tr>
<td>FMNIST</td>
<td>0.91±0.00</td>
<td>0.87±0.02</td>
<td>0.99±0.02</td>
<td>0.86±0.00</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>0.92±0.00</td>
<td>0.82±0.00</td>
<td>0.97±0.02</td>
<td>0.38±0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DP-SGD (eps=3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>0.98±0.00</td>
</tr>
<tr>
<td>FMNIST</td>
<td>0.86±0.01</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>0.38±0.00</td>
</tr>
</tbody>
</table>
Interaction with adversarial training

Adversarial training creates a robust $L_p$ bubble:
- Watermarking not affected but adversarial accuracy drops
- Significant drop in the confidence of radioactive data
- No effect on the DI fingerprint

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No Def.</th>
<th>Watermarking</th>
<th>Radioactive Data</th>
<th>DI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Baseline</td>
<td>with ADV. TR.</td>
<td>Baseline</td>
</tr>
<tr>
<td></td>
<td>$\phi_{\text{ACC}}$</td>
<td>$\phi_{\text{WM}}$</td>
<td>$\phi_{\text{ACC}}$</td>
<td>$\phi_{\text{WM}}$</td>
</tr>
<tr>
<td>MNIST</td>
<td>0.99±0.00</td>
<td>0.97±0.01</td>
<td>0.97±0.02</td>
<td>0.99±0.01</td>
</tr>
<tr>
<td>FMNIST</td>
<td>0.87±0.00</td>
<td>0.99±0.00</td>
<td>0.80±0.06</td>
<td>0.99±0.00</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>0.92±0.00</td>
<td>0.97±0.02</td>
<td>0.78±0.00</td>
<td>0.97±0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ADV. TR.</th>
<th>$\phi_{\text{ACC}}$</th>
<th>$\phi_{\text{ADV}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>0.99±0.00</td>
<td>0.95±0.00</td>
<td></td>
</tr>
<tr>
<td>FMNIST</td>
<td>0.87±0.00</td>
<td>0.69±0.00</td>
<td></td>
</tr>
<tr>
<td>CIFAR10</td>
<td>0.82±0.00</td>
<td>0.82±0.00</td>
<td></td>
</tr>
</tbody>
</table>
Tweaks and relaxations

Tweaking DP-SGD:
• Naively increasing eps (less noise) does not improve WM accuracy
• Increasing gradient clipping threshold is better (not sufficient)
• Bigger training set and training longer improve WM accuracy (not sufficient)

With strict DP-SGD, OOD backdoor watermarking does not work.

What if we relax DP-SGD?
• Splitting the training into the DP part (genuine data) and non-DP (watermark) helps
• Watermark is embedded successfully (accuracy > 0.9 for (F)MNIST, > 0.65 for CIFAR10)
• Privacy loss analysis is not tight anymore

Tweaking hyperparameters or separating objectives does not alleviate other conflicts.
Summary of conflicts

If two techniques A and B in combination result in too high a drop in
• model accuracy ($\phi_{\text{ACC}}$) or
• metric for A ($\phi_A$) or
• metric for B ($\phi_B$)
then A and B are in conflict

<table>
<thead>
<tr>
<th>Protection Mechanism</th>
<th>Dataset</th>
<th>Protection Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>WM</td>
<td>MNIST</td>
<td>$\phi_{\text{ACC}}\phi_{\text{WM}}$</td>
</tr>
<tr>
<td></td>
<td>FMNIST</td>
<td>$\phi_{\text{ACC}}\phi_{\text{WM}}$</td>
</tr>
<tr>
<td></td>
<td>CIFAR10</td>
<td>$\phi_{\text{ACC}}\phi_{\text{WM}}$</td>
</tr>
<tr>
<td>RAD-DATA</td>
<td>MNIST</td>
<td>$\phi_{\text{ACC}}\phi_{\text{RAD-DATA}}$</td>
</tr>
<tr>
<td></td>
<td>FMNIST</td>
<td>$\phi_{\text{ACC}}\phi_{\text{RAD-DATA}}$</td>
</tr>
<tr>
<td></td>
<td>CIFAR10</td>
<td>$\phi_{\text{ACC}}\phi_{\text{RAD-DATA}}$</td>
</tr>
<tr>
<td>DI</td>
<td>MNIST</td>
<td>$\phi_{\text{ACC}}\phi_{\text{DI}}$</td>
</tr>
<tr>
<td></td>
<td>FMNIST</td>
<td>$\phi_{\text{ACC}}\phi_{\text{DI}}$</td>
</tr>
<tr>
<td></td>
<td>CIFAR10</td>
<td>$\phi_{\text{ACC}}\phi_{\text{DI}}$</td>
</tr>
</tbody>
</table>
Combinatorial Explosion

The **complexity** of the analysis **explodes quickly**:

- we investigate 6 pair-wise interactions
- what about triples, quadruples…?
- DP, ADVTR, WM/fingerprinting with fairness constraints is a **reasonable** example

**Thorough analysis with more schemes adds more complexity:**

- we looked at one popular scheme in each category
- e.g., within DP one could study: DP-SGD, PATE, tempered sigmoids, SCATTER-DP
Stakeholders in the Loop

Consider a simple setting:
• a single party gathers the data, trains the model and deploys it
• perhaps they can prioritise one concern over the other

Conflicts are not limited to one party.

There can be multiple specialised stakeholders:
• a model builder concerned about model evasion
• who buys data from a vendor that uses radioactive data
• and uses a training-as-a-service platform that embeds a watermark
ADVTR conflicts with both watermarking and radioactive data.

Regulation can require some protection mechanisms:
• e.g. fairness or privacy.
### Interaction between ML security/privacy techniques

<table>
<thead>
<tr>
<th>Property</th>
<th>Adversarial Training</th>
<th>Differential Privacy</th>
<th>Membership Inference</th>
<th>Oblivious Training</th>
<th>Model/Gradient Inversion</th>
<th>Model Poisoning</th>
<th>Model Watermarking</th>
<th>Model Fingerprinting</th>
<th>Data Watermarking</th>
<th>Explainability</th>
<th>Fairness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adversarial Training</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Differential Privacy</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[1, 2, 8]</td>
</tr>
<tr>
<td>Membership Inference</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oblivious Training</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model/Gradient Inversion</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Poisoning</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Watermarking</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Fingerprinting</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Watermarking</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fairness</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Explainability</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### REFERENCES


Outline

What are the challenges in making AI systems trustworthy?

Is model stealing an important concern?

Can models be extracted via their inference APIs?

**What can be done to counter model theft?**
  - Are model ownership verification schemes robust?

Can we simultaneously deploy protections against multiple concerns?
Robustness of ownership verification schemes

Must be **robust** against **two types** of attackers.

**Malicious suspect:**
- tries to **evade** verification
- common approaches: pruning, fine-tuning, noising

**Malicious accuser:**
- tries to **frame** an **independent** model owner
- **timestamping** (Watermark/fingerprint and model) is the **only** defense in prior work

So far, research has **focused on malicious responders**
False claims against ownership verification schemes

We show how malicious **accusers can make false claims against independent models**: 
• adversary deviates from watermark/fingerprint generation procedure  
  - E.g., via transferrable adversarial examples 
• but still subject to specified verification procedure

Our contributions: 
• formalize the notion of **false claims** against ownership verification schemes
• provide a **generalization** of ownership schemes
• demonstrate **effective false claim attacks**
• discuss potential **countermeasures**

Watermarking by backdooring[3]

Watermark generation:
- choose some out-of-distribution samples as watermark
  - assigned with incorrect labels
- train using the watermark alongside your normal training data (or finetune)
  - model memorizes watermark
- obtain timestamp on commitment of model and watermark

Watermark verification:
- query suspect model using watermark
- compare predictions to the assigned (incorrect) labels:
  - many matching / high WM accuracy -> stolen
  - a few matching / low WM accuracy > not stolen
- check commitment and timestamp

Watermarking by backdooring\textsuperscript{[3]}: false claim

Watermark generation:
• choose some \textit{out-of-distribution} samples as watermark
  - assigned with incorrect labels
• train using the watermark alongside your normal training data (or finetune)
  - model memorizes watermark
• obtain \textit{timestamp on commitment} of model and watermark

Watermark verification:
• query \textit{suspect model} using watermark
• compare predictions to the assigned (incorrect) labels:
  - many matching / high WM accuracy \( \rightarrow \) stolen
  - a few matching / low WM accuracy \( > \) not stolen
• check \textit{commitment} and \textit{timestamp}

Watermarking by backdooring\textsuperscript{[3]}: false claim

**False watermark generation:**
- choose some out-of-distribution samples as false watermark
- perturb these samples to craft transferable adversarial examples
- obtain timestamp on commitment of model and false watermark

**Watermark verification:**
- query suspect model using watermark
- compare predictions to the assigned (incorrect) labels:
  - many matching / high WM accuracy -> stolen
  - a few matching / low WM accuracy > not stolen
- check commitment and timestamp

Takeaways

Is model confidentiality important? Yes
  models constitute business advantage to model owners

Can models be extracted via their inference APIs? Yes
  Protecting model data via cryptography or hardware security is insufficient

What can be done to counter model extraction? Deterrence as defense
  Watermarking/fingerprinting? Open issues remain

Can we simultaneously deploy protections against multiple concerns? Needs work
  Important consideration but not yet sufficiently explored

More on our model extraction work at https://ssg.aalto.fi/research/projects/mlsec/model-extraction/
Takeaways

Is model confidentiality important? Yes
models constitute business advantage to model owners

Can models be extracted via their inference APIs? Yes
Protecting model data via cryptography or hardware security is insufficient

What can be done to counter model extraction? Deterrence as defense
Watermarking/fingerprinting? Open issues remain

Can we simultaneously deploy protections against multiple concerns? Needs work
Important consideration but not yet sufficiently explored

Open (postdoc) positions to help lead our work: ML security/privacy, platform security