Stealing Machine Learning Models via Prediction APIs

F Tramèr, F Zhang, A Juels, MK Reiter, T Ristenpart

USENIX Security '16

Presentor : Donghyeon Kim

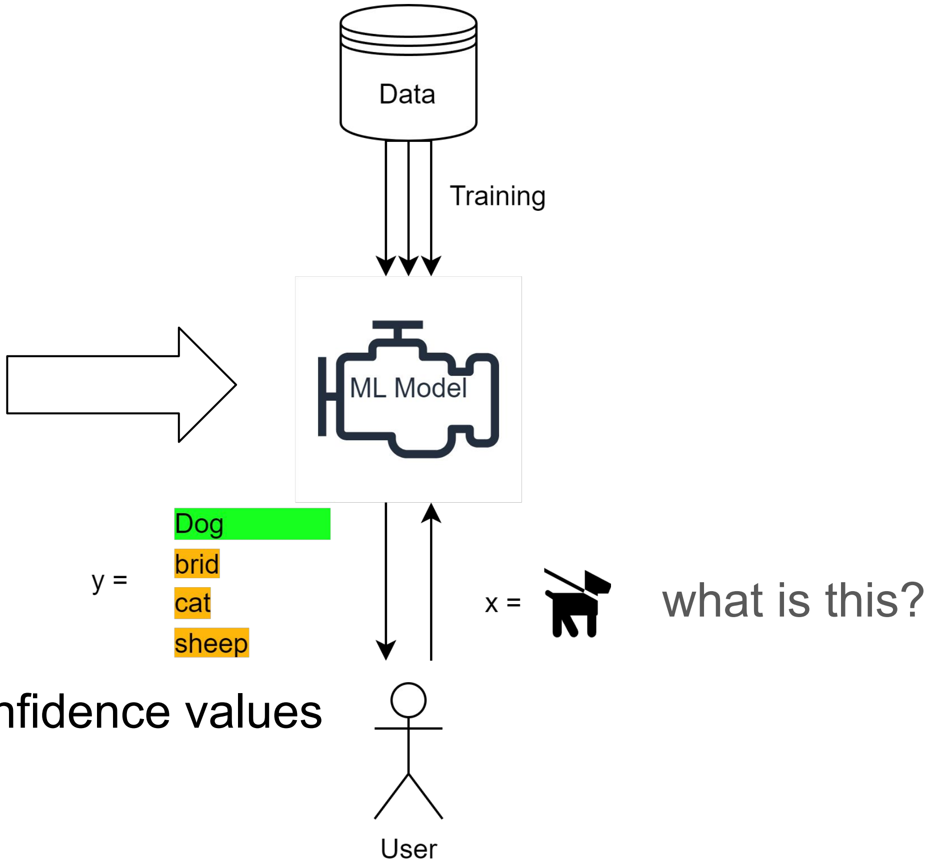
Introduction

Paper Title	Author	Year	Attack Type
Explaining and Harnessing Adversarial Examples	Goodfellow et al.	2014	Adversarial Attack
Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures	Fredrikson et al.	2015	Model Inversion Attack
Towards Evaluating the Robustness of Neural Networks	Carlini & Wagner	2017	Adversarial Attack
Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks	Shafahi et al.	2017	Data Poisoning Attack
BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain	Gu et al.	2017	Backdoor Attack
Robust Deep Reinforcement Learning with Adversarial Attacks	Anay et al.	2017	Adversarial Attack
Trojaning Attack on Neural Networks	Liu et al.	2019	Backdoor Attack
Privacy-Preserving Deep Learning	Hard et al.	2019	Federated Learning Attack
Adversarial Policies : Attacking Deep reinforcement Learning	Adam et al.	2019	Adversarial Attack

What assumptions are made in these papers?

Introduction

SVM
Logistic Regression
Neural Network
Decision Tree



Label + confidence values

Introduction

All the attacks introduced in the papers assume that the model is publicly accessible.

However, in reality...

Bixby

Siri

Chat GPT

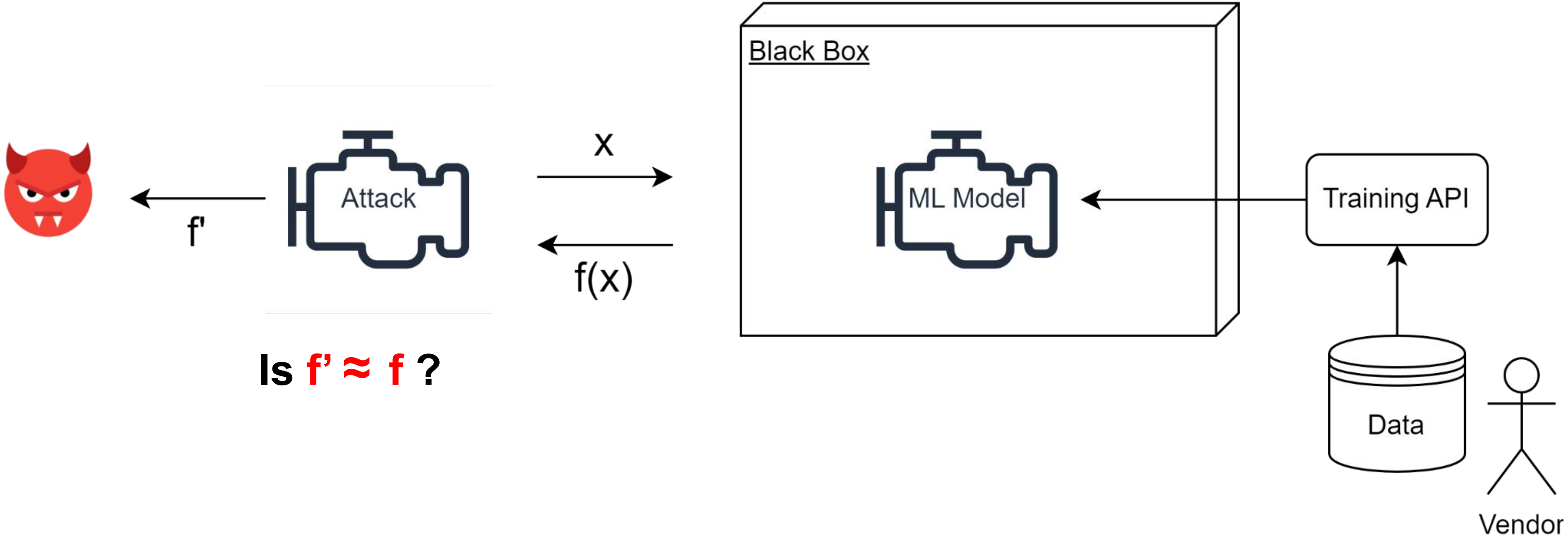
Google AI

Bard

What would be realistic methods for conducting an attack?

Model Extraction Attacks

The goal is for an adversarial client to learn a **close approximation** of the function f by using as few queries as possible.



Is f' ≈ f ?

Model Extraction Attacks

Why adversaries perform this attack ?

1. Avoid charges
   Once you steal the ML model, future queries are free!

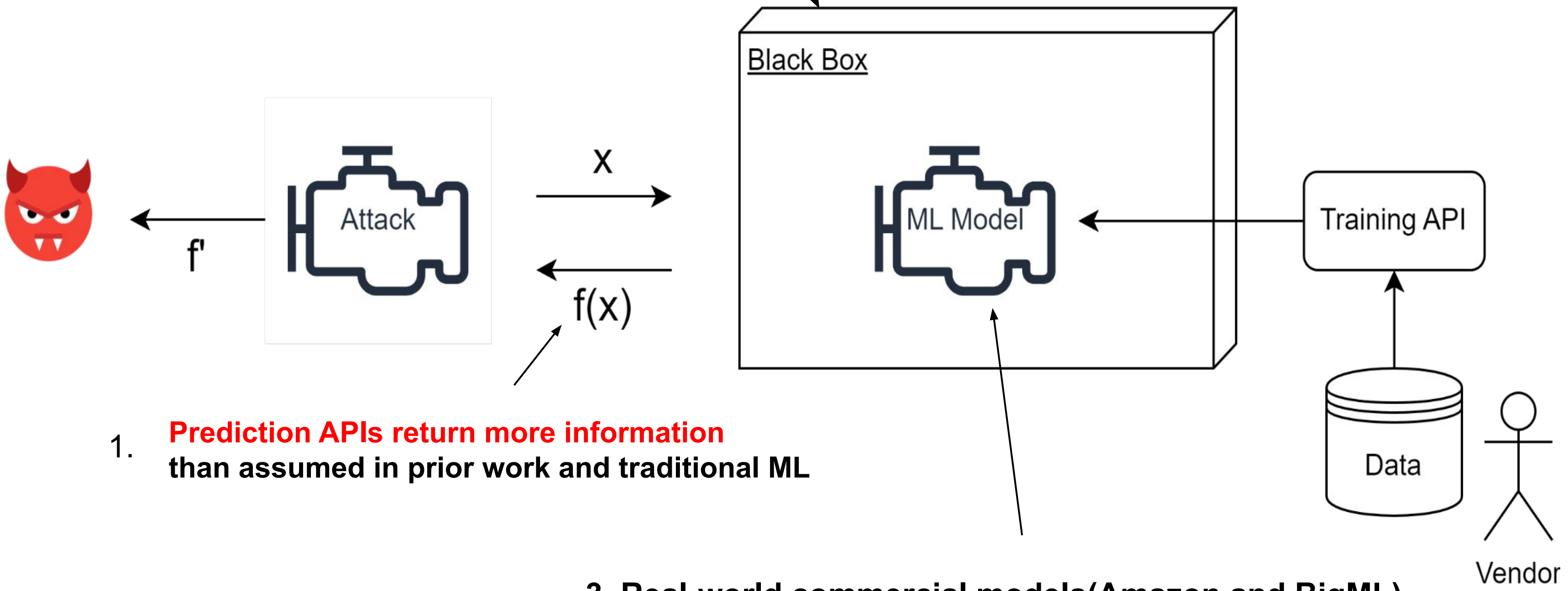
2. Violating training-data privacy
   Model extraction ⇒ You can model inversion (#)

3. Evasion attack
   An attacker can use knowledge of the ML model to bypass detection

Model inversion attacks that exploit confidence information and basic countermeasures *CCS 2015*
FREDRIKSON, M, JHA, S, AND RISTENPART, T.

Prior work

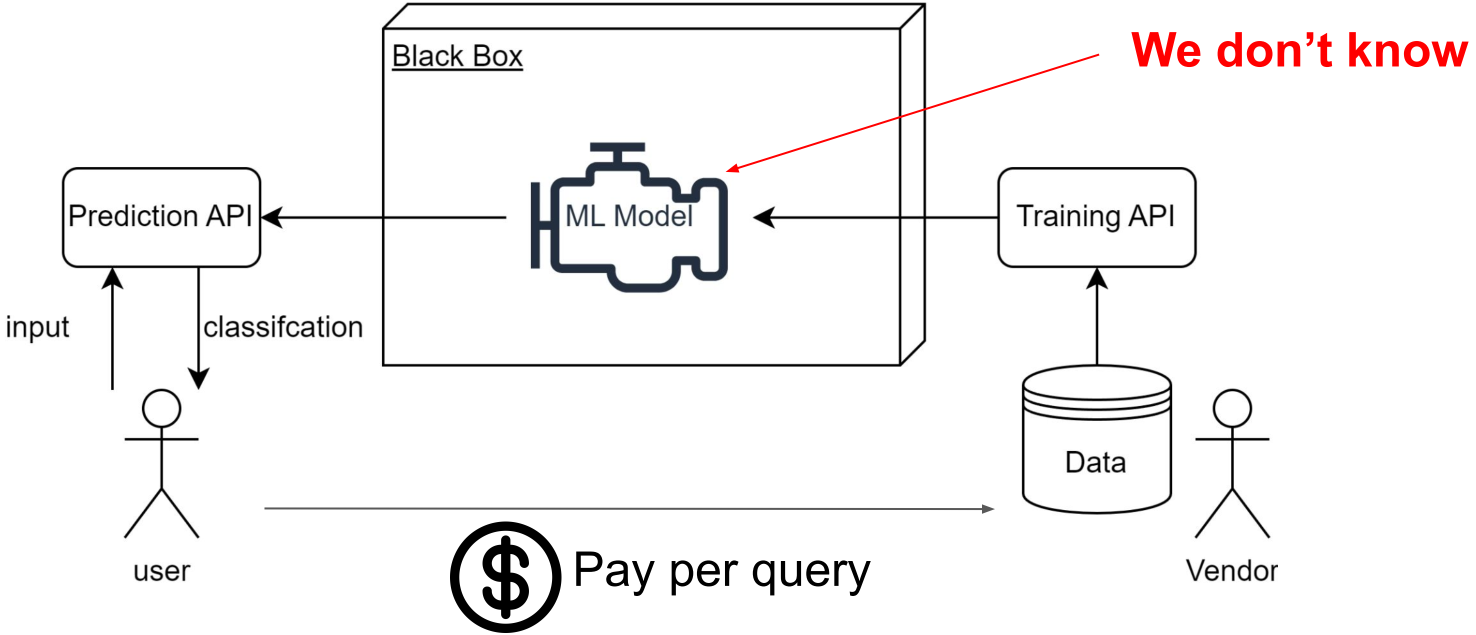
2. **Reverse engineering without knowing the model's type or structure.**



1. **Prediction APIs return more information** **than assumed in prior work and traditional ML**

3. **Real-world commercial models(Amazon and BigML)**

Machine Learning as a Service (MLaaS)



MLaaS’s goal

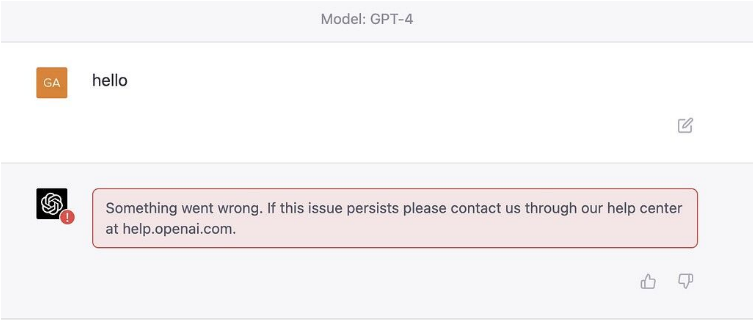
MLaaS has the following two goals:

Goal 1: Rich Prediction APIs

- Highly Available
- High-Precision Results

Goal 2: Model Confidentiality

- Model/Data Monetization
- Sensitive Data



MLaaS Vendor

	Service	Model types
Target	**Amazon**	**Logistic regressions**
	Google	??? (announced : logistic regressions, decision trees, neural network, SVMs)
	Microsoft	Logistic regressions, decision trees, neural network, SVMs)
	PredictionIO	Logistic regressions, decision trees, SVMs(white-box)
Target	**BigML**	**Logistic regressions, decision trees**

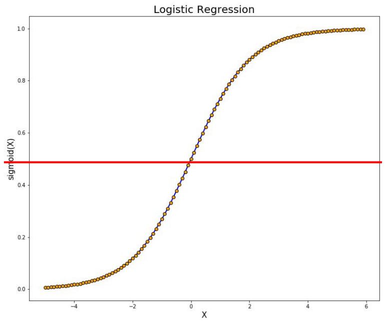
Attack

1. Logistic Regression
2. Multiclass Logistic Regression
3. Decision Tree

Attack 1 : Logistic Regression

Task : Facial Recognition of two people (binary classification)

$$f(x) = 1 / (1+e^{-(w*x + b)})$$



Alice

Bob

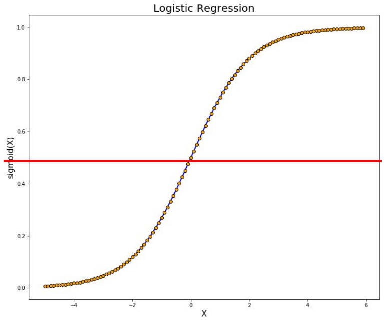
Learning f is equivalent to finding values for the vector w and the vector b.

Attack 1 : Logistic Regression

Equation solving attack

we can write as:

$$\ln(f(x)/(1-f(x))) = w*x + b$$



Alice

Bob

Querying n+1 random points allows us to solve a linear system of n+1 equations.

Attack 2 : Multiclass Logistic Regression

Multi class LR (MLR)

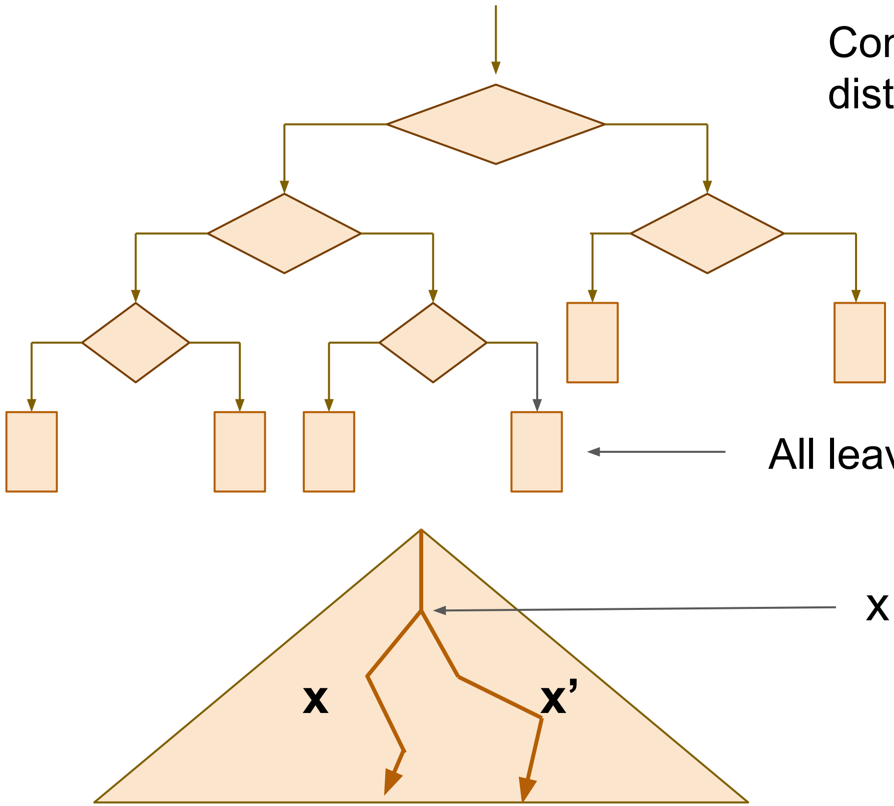
Generalize to c > 2 classes
→ ~ 1 query per model parameter of f

The system of equations is **nonlinear**

For instance, in the case of the softmax model, the equations take the following form:

$$\frac{e^{w_i \cdot x + \beta_i}}{\sum_{j=0}^{c-1} e^{w_j \cdot x + \beta_j}} = f_i(x)$$

Attack 3 : Decision Tree

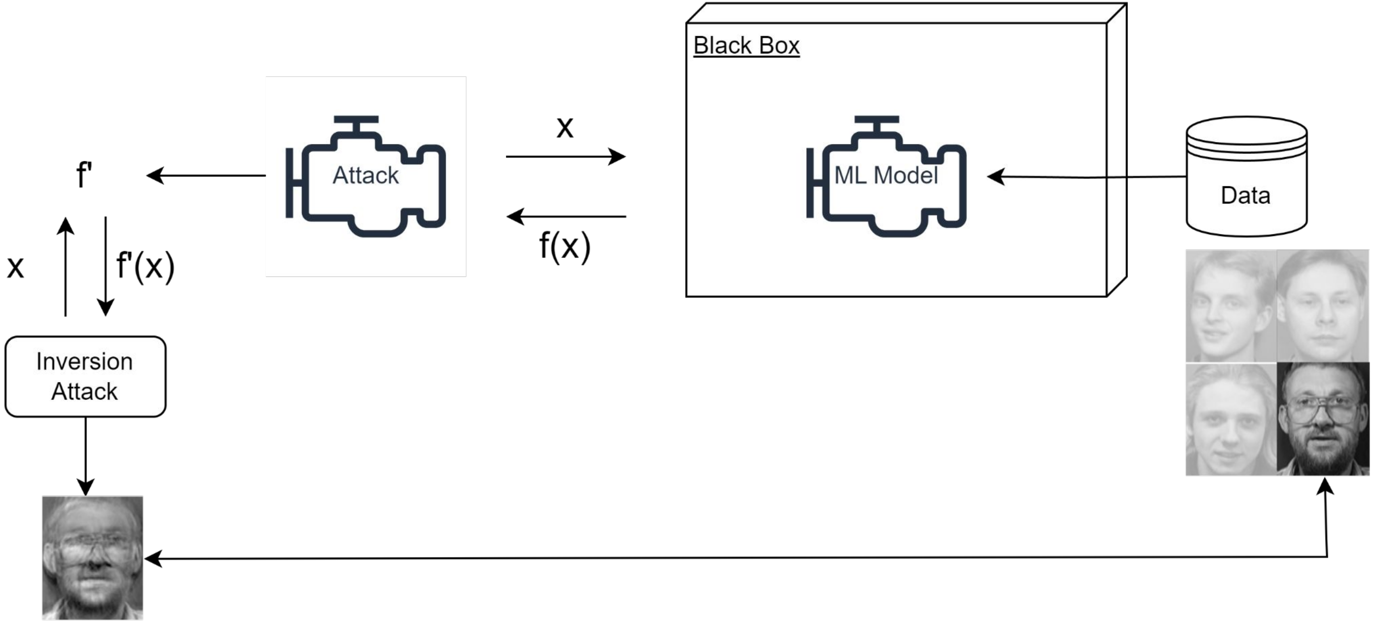


Confidence value based on the class distribution in the training data.

All leaves have unique confidence

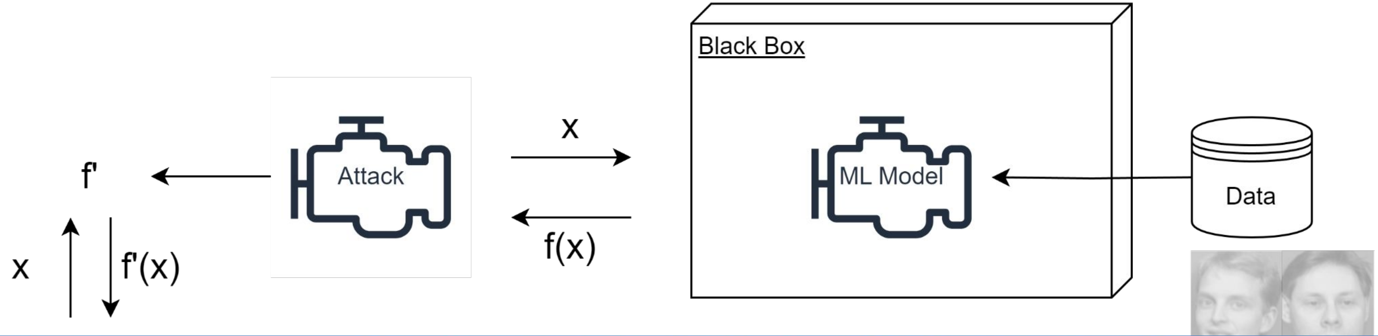
x and x’ differ in a single feature

Attack with inversion attack



Model extraction can enhance inversion attacks, potentially leaking training data.

Attack with inversion attack



Strategy	Attack against 1 individual		Attack against all 40 individuals		
	Online Queries	Attack Time		Online Queries	Attack Time
Black-Box Inversion [Fredrikson et al.]	20,600	24 min	×40	800,000	16 hours
Extract-and-Invert (our work)	41,000	10 hours	×1	41,000	10 hours

Case study : BigML

Target : Decision trees

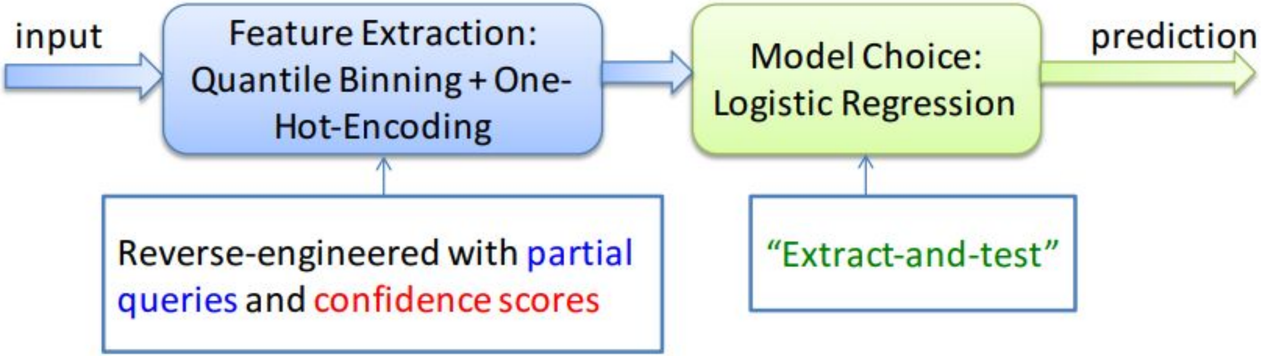
Traing data : German Credit

Run over 1000~2000 queries ⇒ Extraction attack is better

Case study : AWS

Two feature extraction technique:

1. One hot encoding
2. Quantile binning



Case study : AWS

Model	OHE	Binning	Queries	Time (s)	Price ($)
Circles	-	Yes	278	28	0.03
Digits	-	No	650	70	0.07
Iris	-	Yes	644	68	0.07
Adult	Yes	Yes	1,485	149	0.15

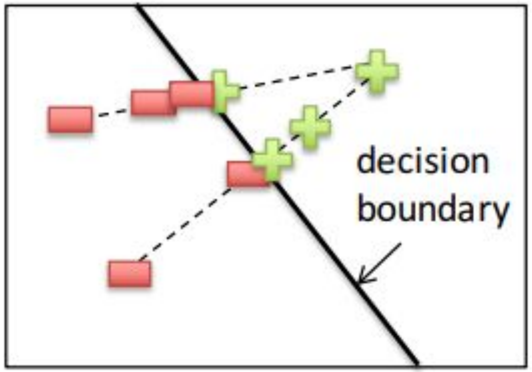
Table 7: **Results of model extraction attacks on Amazon.** OHE stands for one-hot-encoding. The reported query count is the number used to find quantile bins (at a granularity of 10^{-3}), plus those queries used for equation-solving. Amazon charges $0.0001 per prediction [1].

Model extraction can be done at a very **low cost**.

Defense

Simple method : **Remove confidence values**
→ Prediction = class label only
f(x) = sign(w*x + b)

1. Find points on decision boundary (w*x + b = 0)
2. Reconstruct w and b



Removing the confidence values
→ **100 times more** queries per model parameter

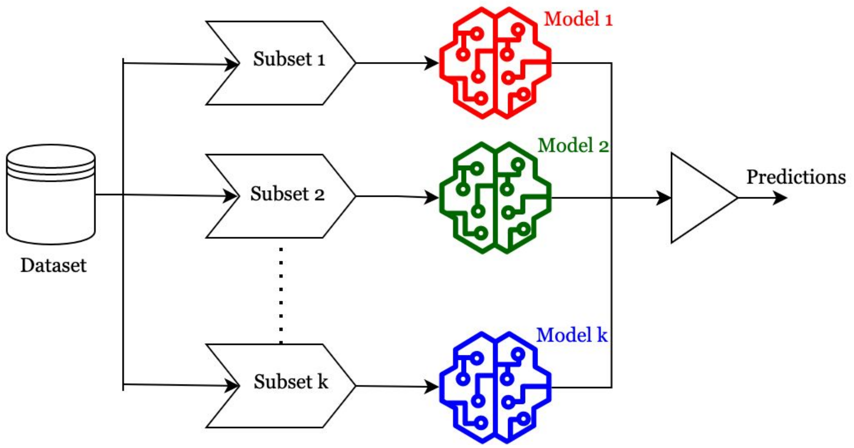
Defense

Rounding confidences

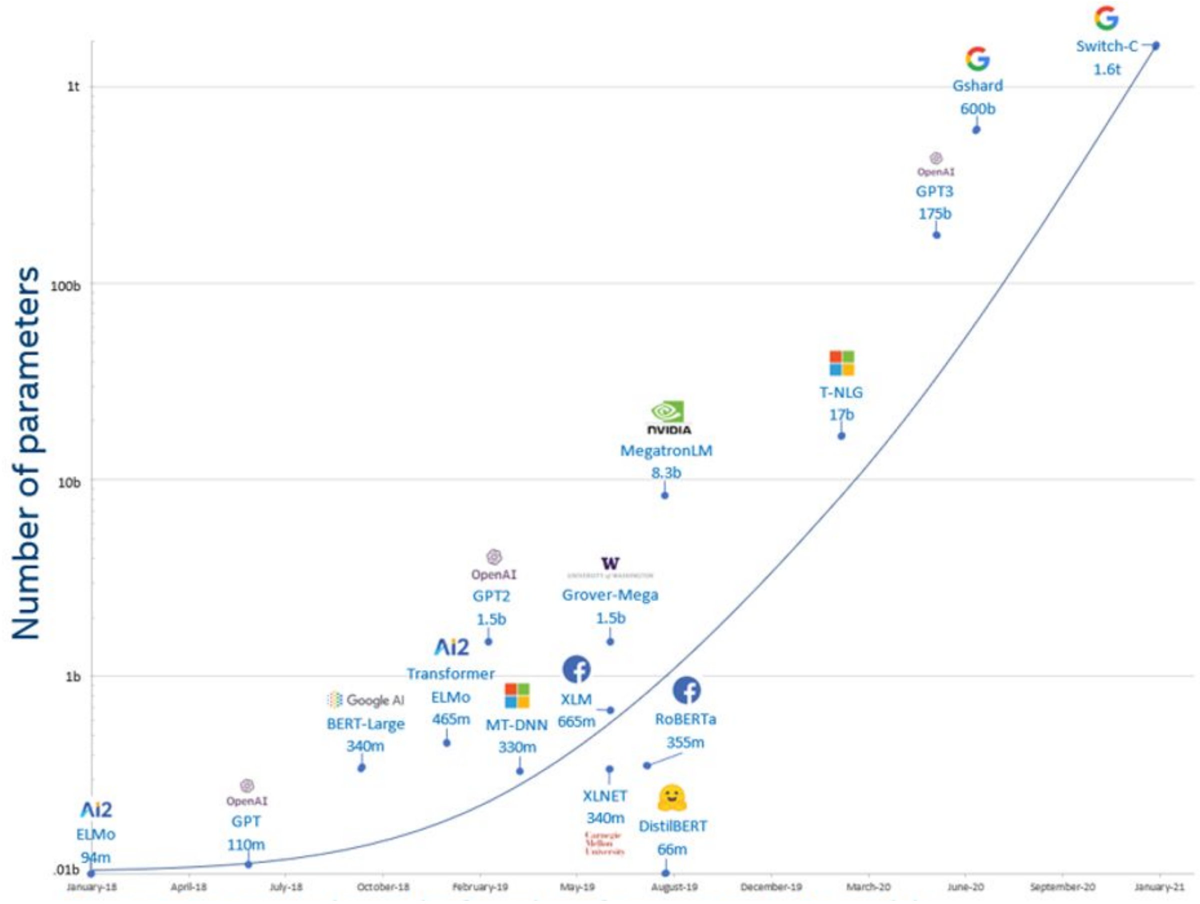
Output Randomization

Differential privacy

Ensemble methods



Is model extraction useful?

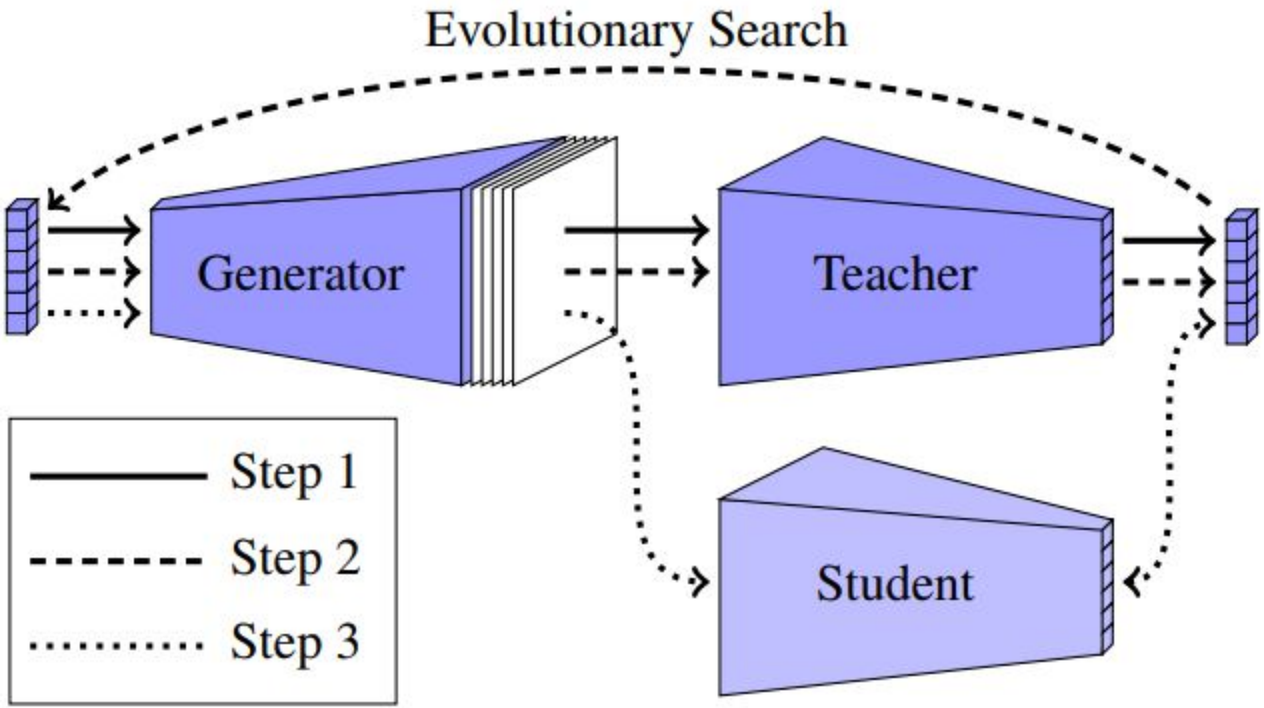


Related works

Paper Title	Authors	Year
Model inversion attacks that exploit confidence information and basic countermeasures	Fredrikson et al.	2015
Stealing Machine Learning Models via Prediction APIs	Tramèr et al.	2016
Membership Inference Attacks Against Machine Learning Models	Shokri et al.	2017
Model Extraction Attacks Against Black-Box Machine Learning Models	Papernot et al.	2017
Model Extraction Using Active Learning and Unannotated Public Data	S Pal et al.	2020
Black-Box Ripper: Copying black-box models using generative evolutionary algorithms	A Barbalau et al.	2020

Related works

Paper T		Year
Model ir counterr	l.	2015
Stealing		2016
Member		2017
Model E		2017
Model E		2020
Black-B algorithr	al.	2020



Conclusion

Efficient Model Extraction Attack.

- Target : Logistic Regressions, Neural Networks, Decision Trees, SVMs
- Active learning attacks in membership-query setting

Performed real-world online Model Extraction Attack

The presenter's opinion

But it seems difficult to apply to commercial models at the moment.

Good Question

- What types of information should be protected in an LLM service to prevent model leakage? For example, could access to Attention information, specific layer outputs, or intermediate computation values make model leakage feasible?

- The most complex model used in this paper had about 400k parameters. With the age of Transformers upon us, is such an attack really feasible today?

- The process of obtaining input-output samples appears similar to a brute force attack. In that case, could countermeasures for brute force attacks be used to reduce the attacker's efficiency?

Best Question

Zunnoor Fayyaz Awan

Can we implement anomaly detection that combines these temporal and non-temporal signatures to detect an ongoing extraction attack?

Pierre Noyer

As models are frequently updated or retrained overtime. Would this attack still work in the long-term: would it be necessary to re-do the attack from scratch every time the model evolves, or would the old copy be a good starting point for the new attack?

Additional

How can security be strengthened for on-device AI models?

Unlike cloud-based services, on-device AI models reside on the device itself. Does this protect them from model extraction attacks?

What effective countermeasures can on-device AI employ to prevent model extraction attacks?

Since the model operates locally, attackers may not have access through APIs. Is additional protection still necessary for on-device AI models?