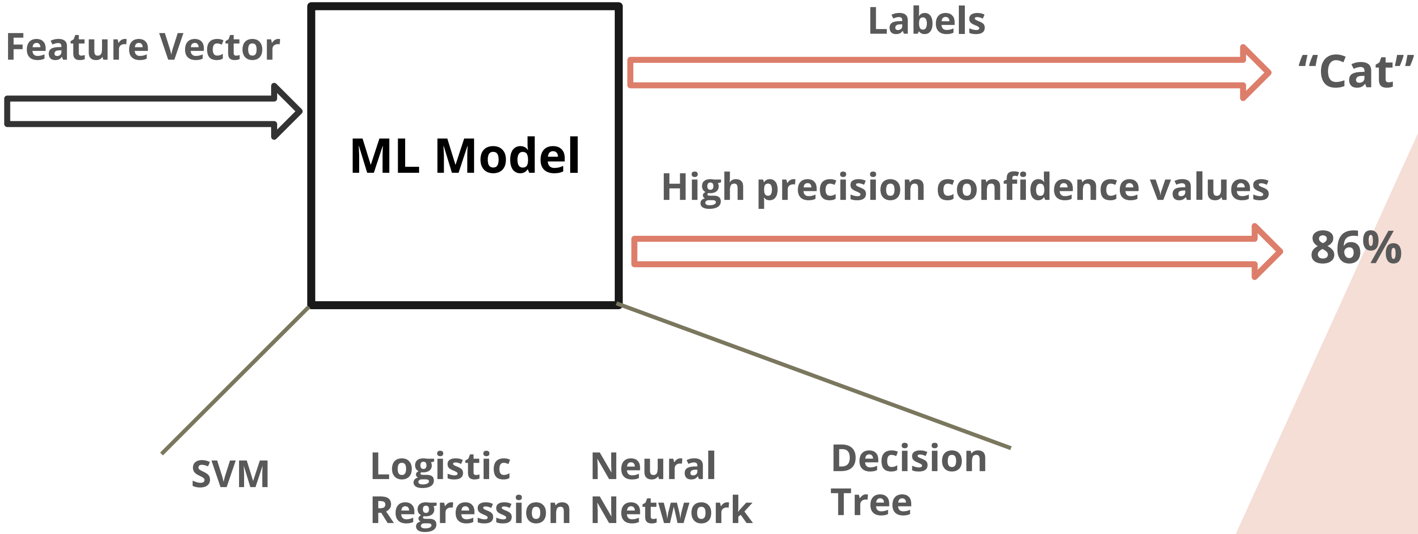
Stealing Machine Learning Models via Prediction APIs

F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart

SEC'16: Proceedings of the 25th USENIX Conference on Security Symposium

20180391 Oh SeongRyong

Introduction



Introduction

Pay-per-query deployment increase !

Cloud based service

Commercially Valuable model

Confidential value

Class label

Not perfect features

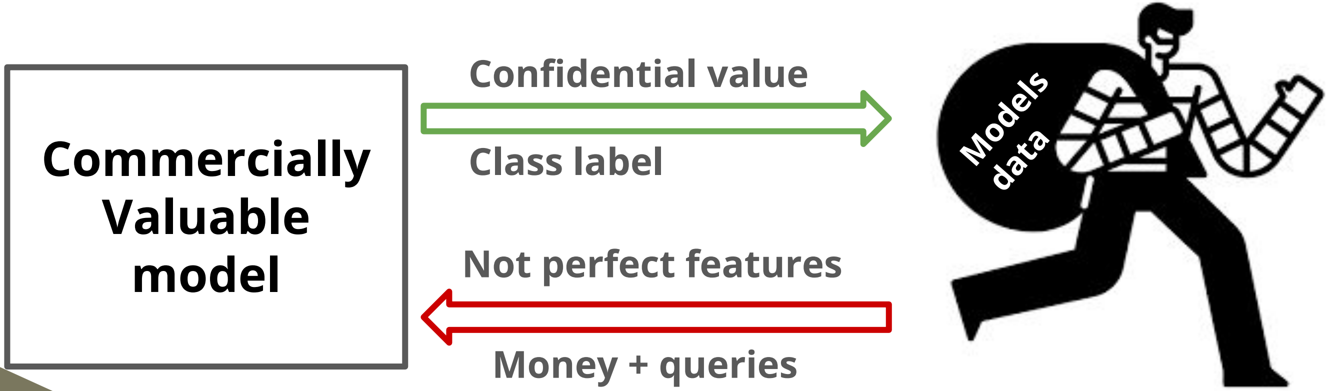
Money + queries

This can be..

Introduction

Cloud based service

Adversary !



Model Extraction Attacks Scenarios (Why adversaries perform this attack)

1. Avoiding query charges

If you steal ML model, then subsequent query is free..!

2. Violating training-data privacy (privacy-abusing attack)

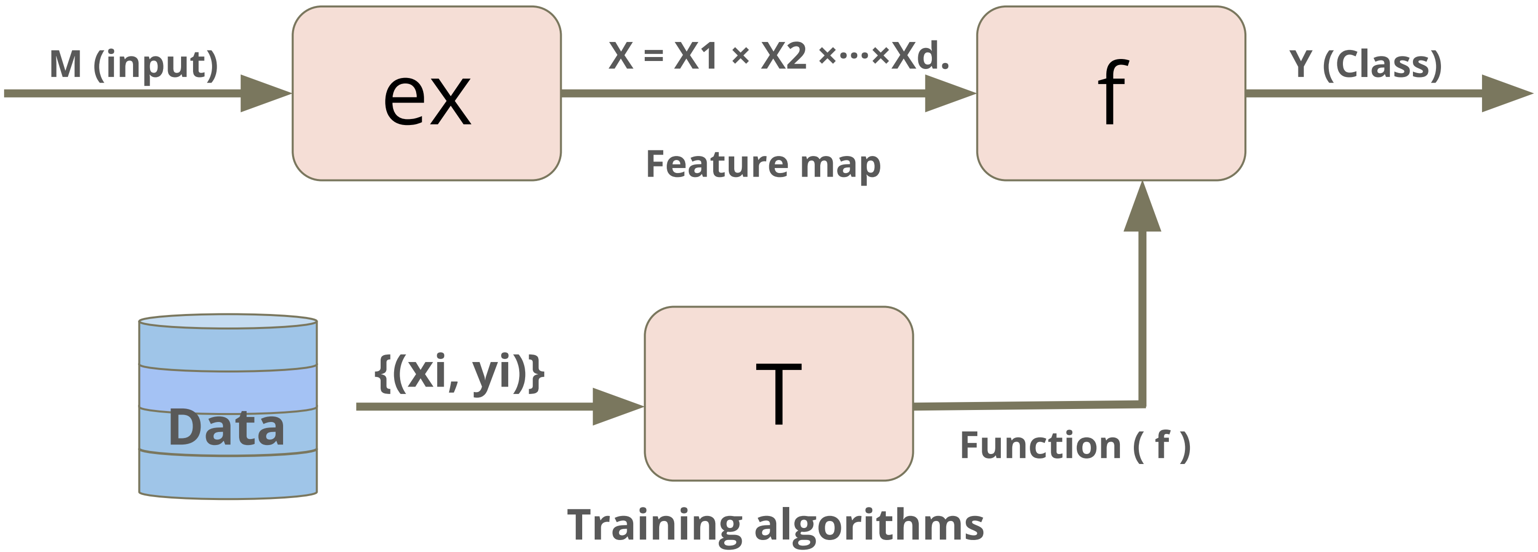
Model extraction ⇒ model inversion (#)

3. Stepping stone to evasion (evasion attack)

An adversary may use knowledge of ML model to avoid detection by model (spam, malware classification)

(#) FREDRIKSON, M., JHA, S., AND RISTENPART, T. Model inversion attacks that exploit confidence information and basic countermeasures. In CCS (2015), ACM, pp. 1322–1333.

Background


M (input)
ex
X = X1 × X2 ×⋯×Xd.
Feature map
f
Y (Class)
Data
{(xi, yi)}
T
Training algorithms
Function (f)

Attack models in this paper

(Case1) Logistic regression

confidence value : 1/(1+e^(−(w·x+β))) [d+1 unknown parameters w , β]

d + 1 input query ⇒ define model

(Case 2) multi class logistic regression, Neural network

With equation - solving attacks

(Case 3) decision tree ⇒ adaptive, iterative search algorithm ⇒ paths in tree.

Key idea

equation - solving attacks

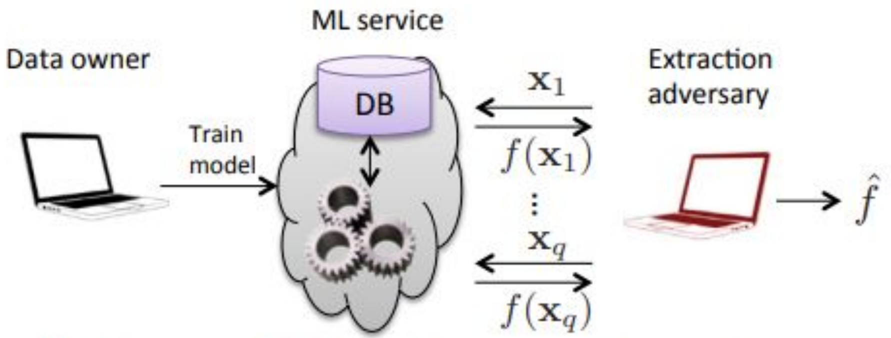
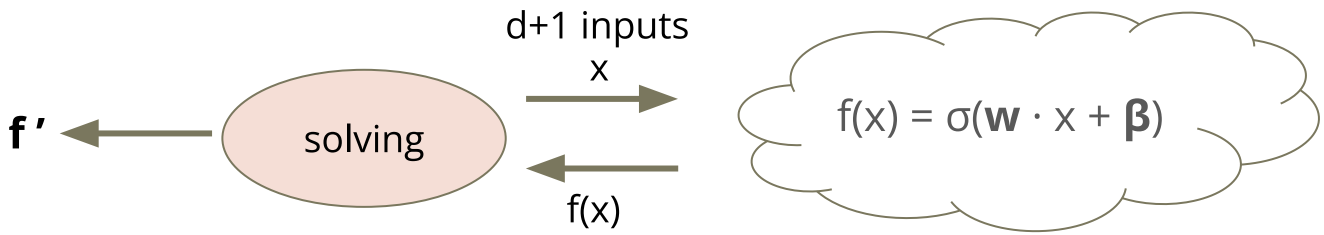


Figure 1: **Diagram of ML model extraction attacks.** A data owner has a model f trained on its data and allows others to make prediction queries. An adversary uses q prediction queries to extract an $\hat{f} \approx f$.

(Case1) Logistic Regression Model Extraction

Task: binary classification (A? or B?)

Goal: learn close approximation of f using a few queries as possible



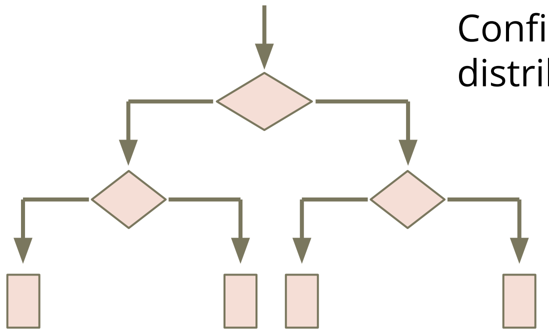
Query d+1 random points ⇒ solve a linear system of d+1 unknowns (w and b)

(Case2) Logistic Regression Model Extraction

Multiclass LR (MLR)
Generalize to $c > 2$ classes with multinomial logistic regression
⇒ ~ 1 query per model parameter of f.

Multilayer Perceptrons (MLP)
Since MLP has non-linear function, MLP is more complex than MLR.
⇒ Adversary need more query

(Case3) Extracting a Decision Tree



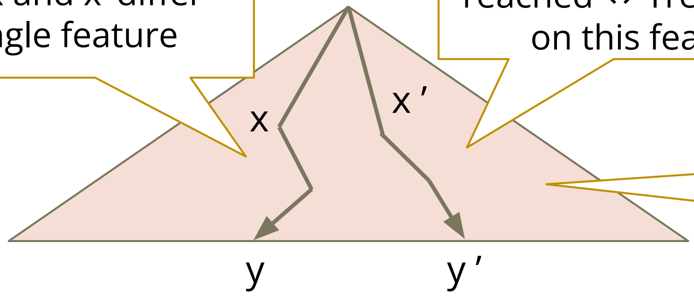
Confidence value derived from class distribution in the training set

Previous work **Kushilevitz-Mansour** (1992)

- Poly-time algorithm with **membership queries** only
- Only for Boolean trees, **impractical complexity**

Input x and x' differ in a single feature

Different leaves are reached ⇔ Tree "splits" on this feature



All tree leaves have unique confidence values

Online attacks on **BigML**.

Online Model Extraction Attacks

BIG_ML (Decision Tree)

Train and extract | Setup : black box model
As a result(consider the monetary cost)
expect to run over 1,150 queries ⇒ Extraction attack is better

Amazon Web Services (classification)

This contain the Two feature extraction technique (one-hot-encoded, Quantile binning)

They do reverse engineering + extraction model

Extraction Given Class Label Only

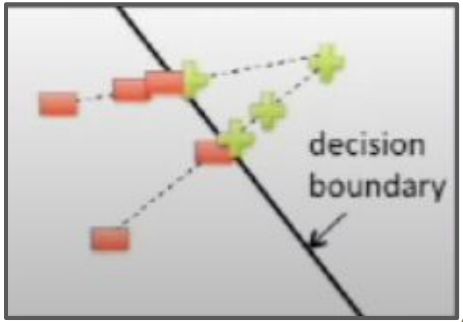
{**class label** only, not confidence value}

Attack on Linear Classification [Lowd,Meek -2005]

Classify as “+” if w*x +b >0
and “-” otherwise

d+1 parameters w, b

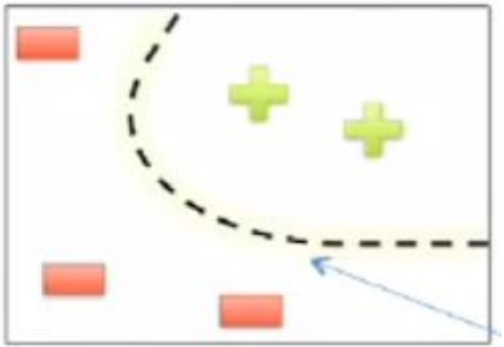
f(x) = sign(w*x+b)



1. Find points on decision boundary (w*x+b =0)
   - Line search between the two points (“+” and “-”)
2. Reconstruct w and b (up to scaling factor)

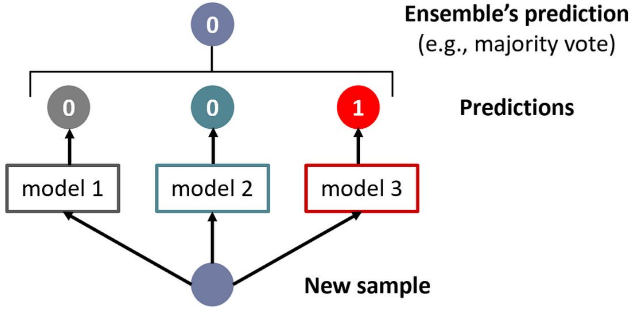
Extraction Given Class Label Only

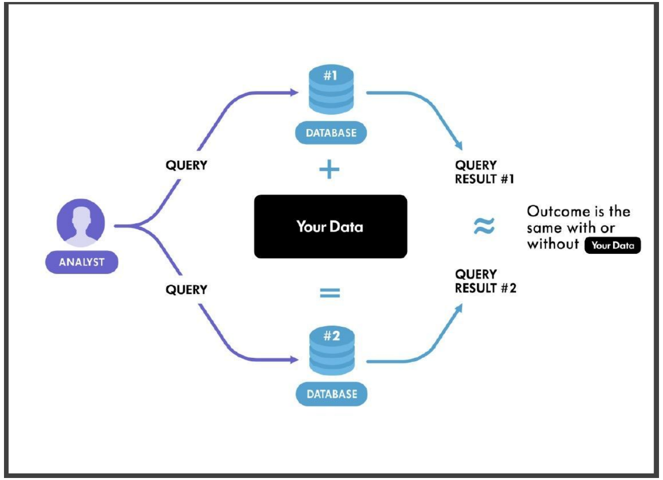
Extend the **Lowd-Meek approach** to non-linear models
Active Learning:
    Query points close to “decision boundary”
    Update f’ to fit these points
Multinomial Regressions, Neural Networks, SVMs:
    > 99% agreement between f and f’
    = 100 queries per model parameter of f ⇒ **expensive, less efficient**
However, cannot prevent attack perfectly



Other Extraction Countermeasures

- **Rounding confidences.**
- **Differential privacy**
- **Ensemble methods.**





Related Work (previous work)

Algorithms for learning with membership queries & Learning algorithm that provide labels only.

- LOWD, D., AND MEEK, C. Adversarial learning. In KDD (2005), ACM, pp. 641–647.
- KUSHILEVITZ, E., AND MANSOUR, Y. Learning decision trees using the Fourier spectrum. SICOMP 22, 6 (1993), 1331–1348.
- BSHOUTY, N. H. Exact learning boolean functions via the monotone theory. Inform. Comp. 123, 1 (1995), 146–153

Related Work (future work)

(future work) **Membership Inference Attacks against Machine Learning Models** Reza Shokri, Marco Stronati, Congzheng Song, Vitaly Shmatikov
⇒ try to attack more complicate models (such as GPT model)

(future work) **Robust Physical-World Attacks on Deep Learning Visual Classification** (CVPR), 2018

(future work) **Certified Defenses for Data Poisoning Attacks (NIPS 2017)**

Conclusion

How the flexible prediction APIs exposed by current ML-as-a-service providers enable new model extraction attacks

⇒ subvert model monetization
⇒ violate training-data privacy
⇒ model evasion

Real-World Online Model Extraction Attack

Suggest countermeasures

Good Question

1. (허현) Is there any follow-up research that prevents the model from being leaked by an attacker?
    - D-DAE: Defense-Penetrating Model Extraction Attacks
    - QuSecNets: Quantization-based Defense Mechanism for Securing Deep Neural Network against Adversarial Attacks
2. (김호빈)
   Why is the confidence value presented to the user in ML models? Can another problem occur if there is a reasonable justification for the omission of confidence values from the service?

Best Question

1. (정기원) Is there any future works to detect malicious model extraction attack attempt by using ML models?

2. (박승민) To extract a high-accuracy model through this attack, it seems that many queries must be requested through the API. If we limit the number of queries that can be sent consecutively at a time, can it be an effective defence strategy for this attack?

3. (Valetin) Trade-offs between utility and security are widely discussed in rounding confidences or DP for example. How can we assess the correct balance for these defenses in real-world deployment scenarios where both security and accuracy are to be considered?

Thanks