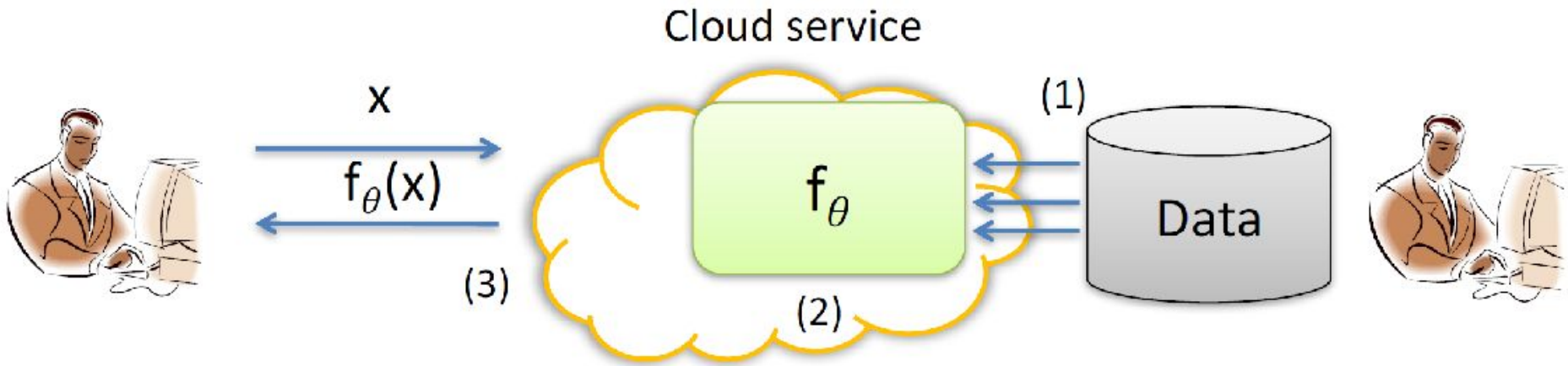
Lecture 10: Model Inference and Training Data Reconstruction Attacks against ML

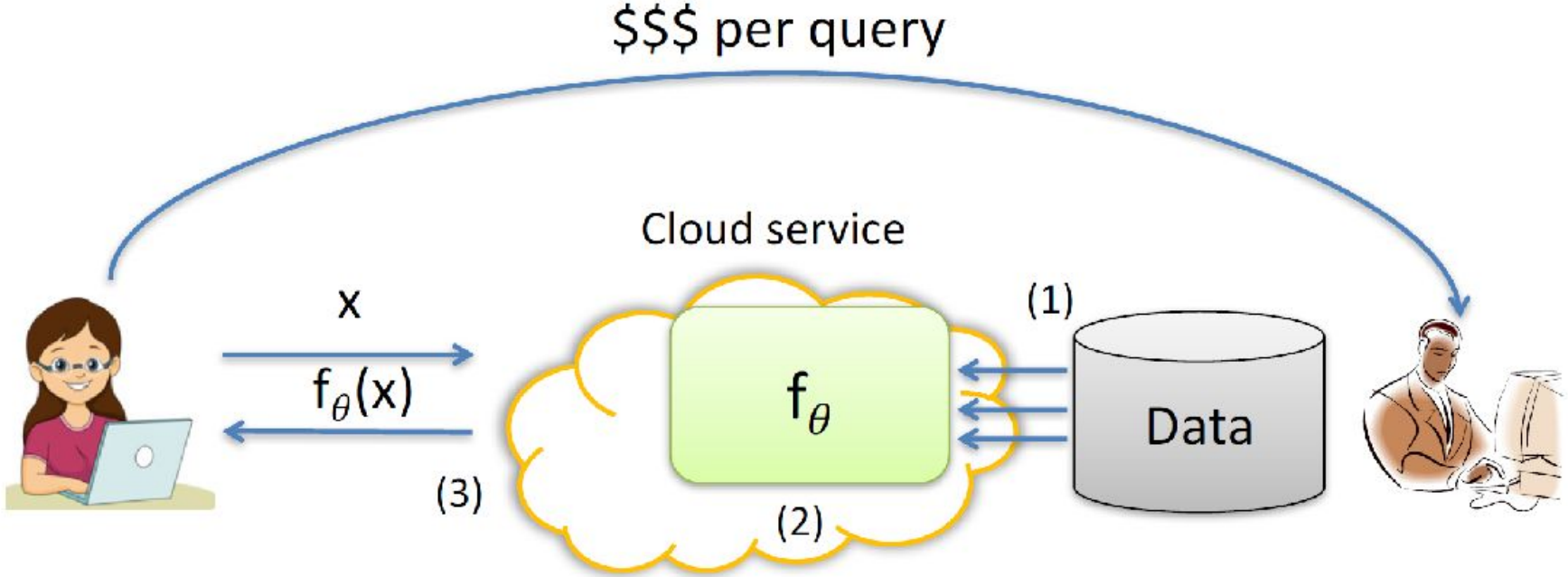
Siddharth Garg

Closer look: ML-as-a-service



(1) Data owner uploads data
(2) Requests training of model f from data
(3) Data owner can use f to make predictions

Closer look: ML-as-a-service



(1) Data owner uploads data

(2) Requests training of model f from data

(3) Data owner can *make f available for others to query*

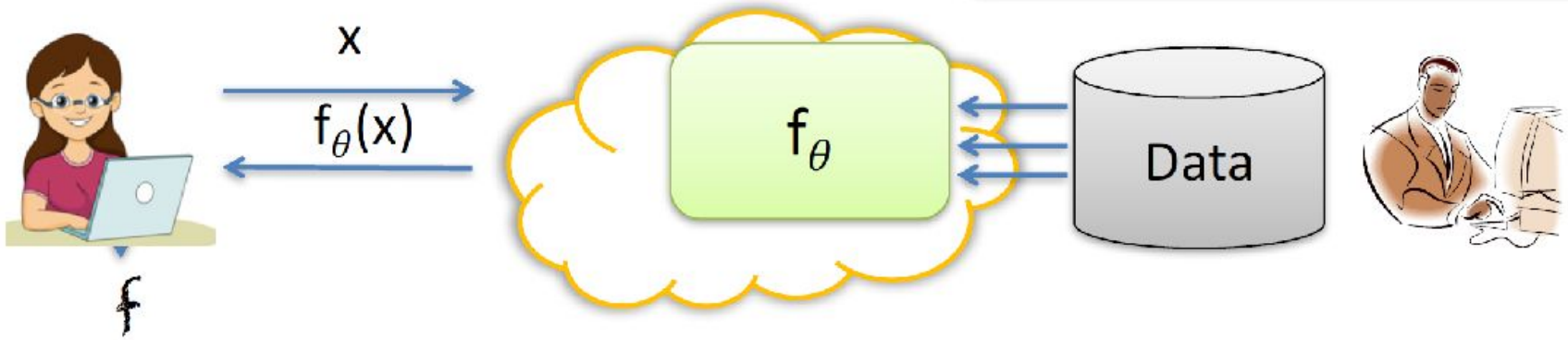
Refer to this as black-box setting

Model extraction attacks

[Tromer, Zhang, Juels, Reiter, R. 2016]

Adversarial client seeks to learn close approximation of f_θ in as few queries as possible

We will target $\hat{f}(x) = f_\theta(x)$ on ≥ 99.9% of inputs



Efficient attacks could:

- undermine pay-for-prediction pricing model
- facilitate privacy attacks (stay tuned)
- enable evasion attacks

Example: logistic regression

Facial recognition of two people, Alice and Bob (the classes)

x[1] , Alice x[2] , Alice x[3] , Bob x[4] , Bob …

Feature vectors are pixel data
e.g.: n = 92 * 112 = 10,304

n+1 parameters θ = w,b chosen using training set to minimize expected error

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

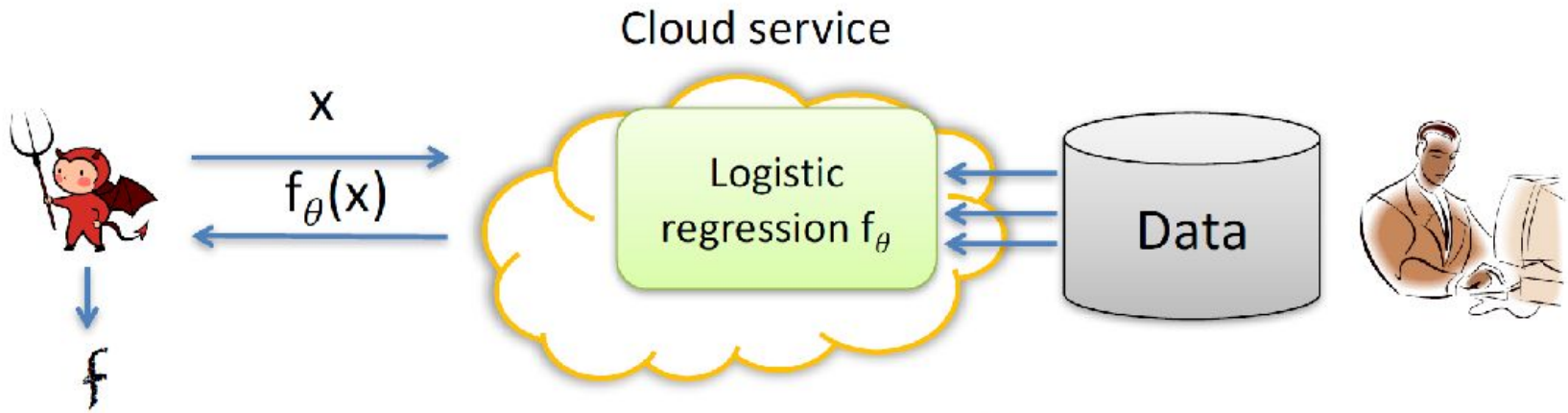
f_θ maps features to predicted probability of being "Alice"
≤ 0.5 classify as "Bob"
> 0.5 classify as "Alice"

Generalize to c > 2 classes with *multinomial logistic regression*

$f_\theta(x) = [p_1, p_2, \ldots, p_c]$ predict label as $\text{argmax}_i\ p_i$

Model extraction attacks

Adversarial client seeks to learn close approximation of f_θ in as few queries as possible



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

$$\ln\left(\frac{f_\theta(x)}{1 - f_\theta(x)}\right) = w*x + b$$

Linear equation in n+1 unknowns w,b

Query n+1 random points → solve linear system of n+1 equations

~100x fewer queries than [Lowd, Meek 2005]

Model extraction attacks

Adversarial client seeks to learn close approximation of f_θ in as few queries as possible

$$\Pr(Y_i = 1) = \frac{e^{\beta_1 \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot \mathbf{X}_i}}$$

$$\Pr(Y_i = 2) = \frac{e^{\beta_2 \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot \mathbf{X}_i}}$$

.....

$$\Pr(Y_i = K-1) = \frac{e^{\beta_{K-1} \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot \mathbf{X}_i}}$$

Model type	Attack approach
Binary logistic regression	Solve linear equations
Multinomial logistic regression	Solve non-linear equations
Neural network	Solve non-linear equations
Decision trees	Path-finding using pseudo-identifiers for leaves + partial feature vector queries

Tests with cloud services:

Amazon (multinomial LR)

BigML (decision trees)

100s to 1000s of queries

Seconds to minutes

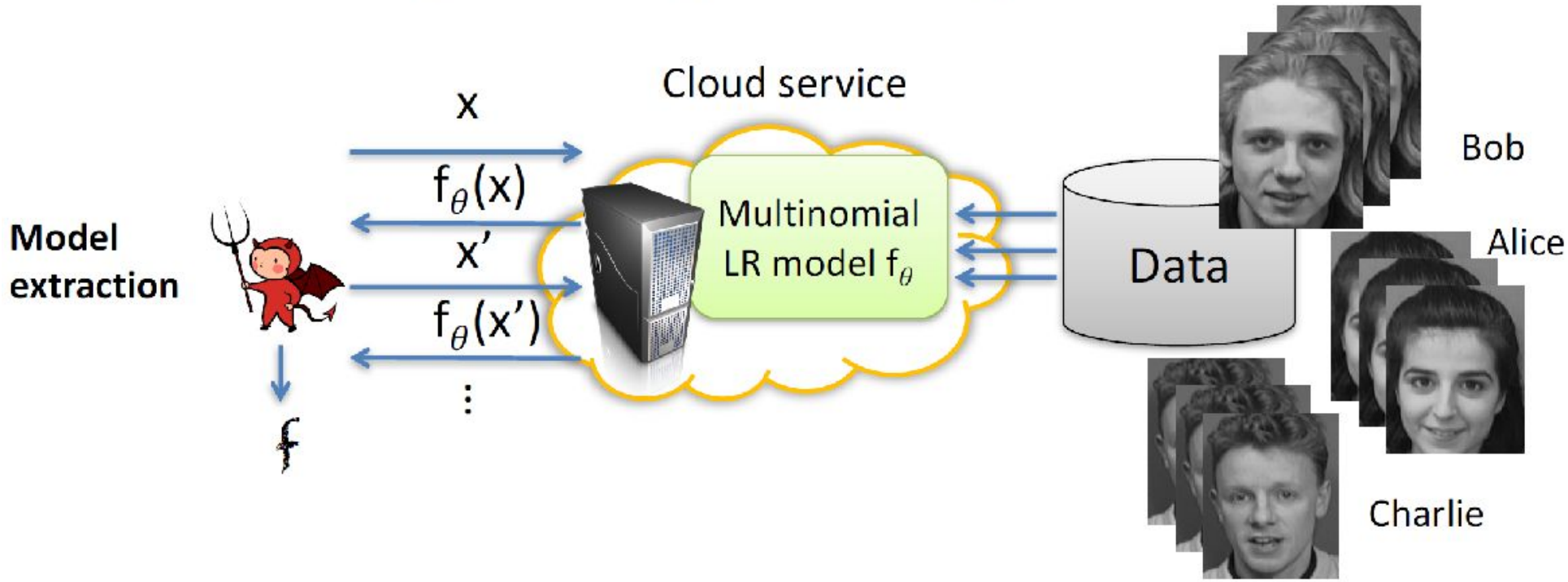
100% accuracy

$f(x) = f_\theta(x)$ on all x

More detailed results in paper

Concrete example: recovering recognizable faces

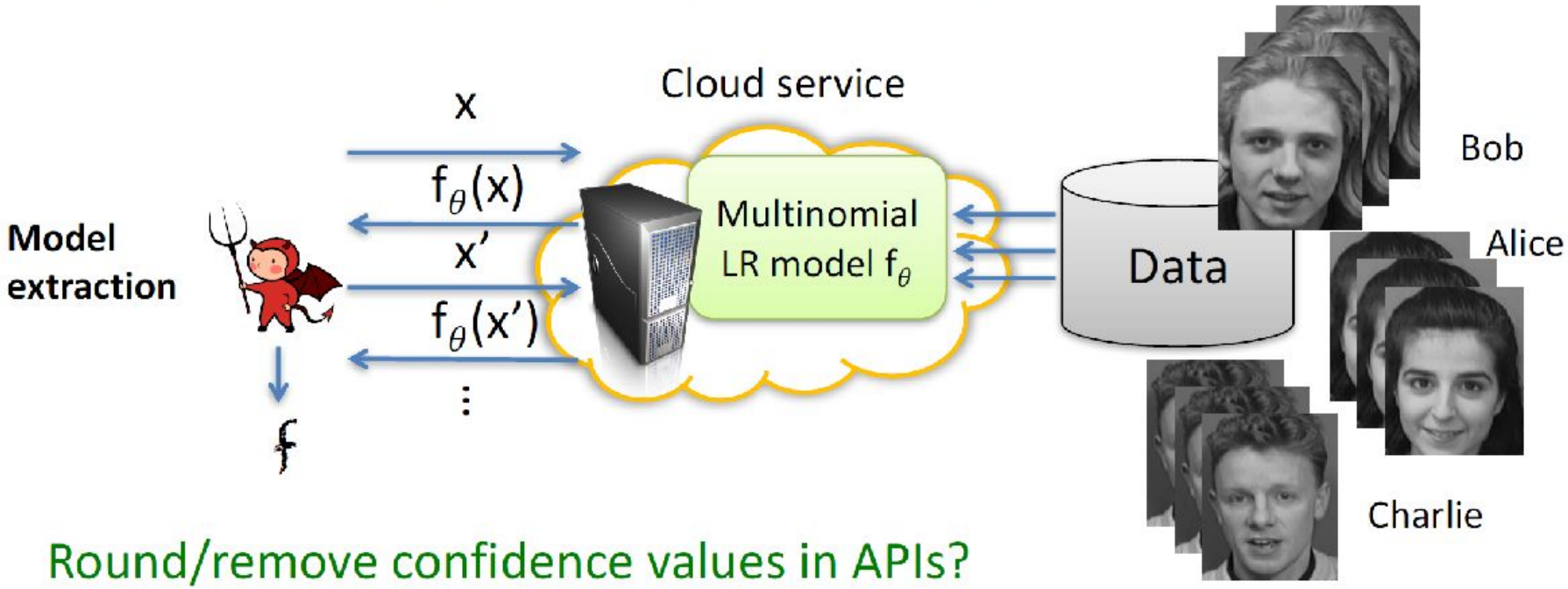
Given access to facial recognition model f_θ can we reconstruct recognizable images of training set members?



θ has 412,160 unknowns (trained on AT&T faces dataset, c = 40)
Make 41,216 queries (estimate: 1 hour)
Solve 41,216 non-linear equations in unknowns (~10 hours)
$f(x) = f_\theta(x)$ for 99.9% of inputs

Concrete example: recovering recognizable faces

Given access to facial recognition model f_θ can we reconstruct recognizable images of training set members?



Round/remove confidence values in APIs?

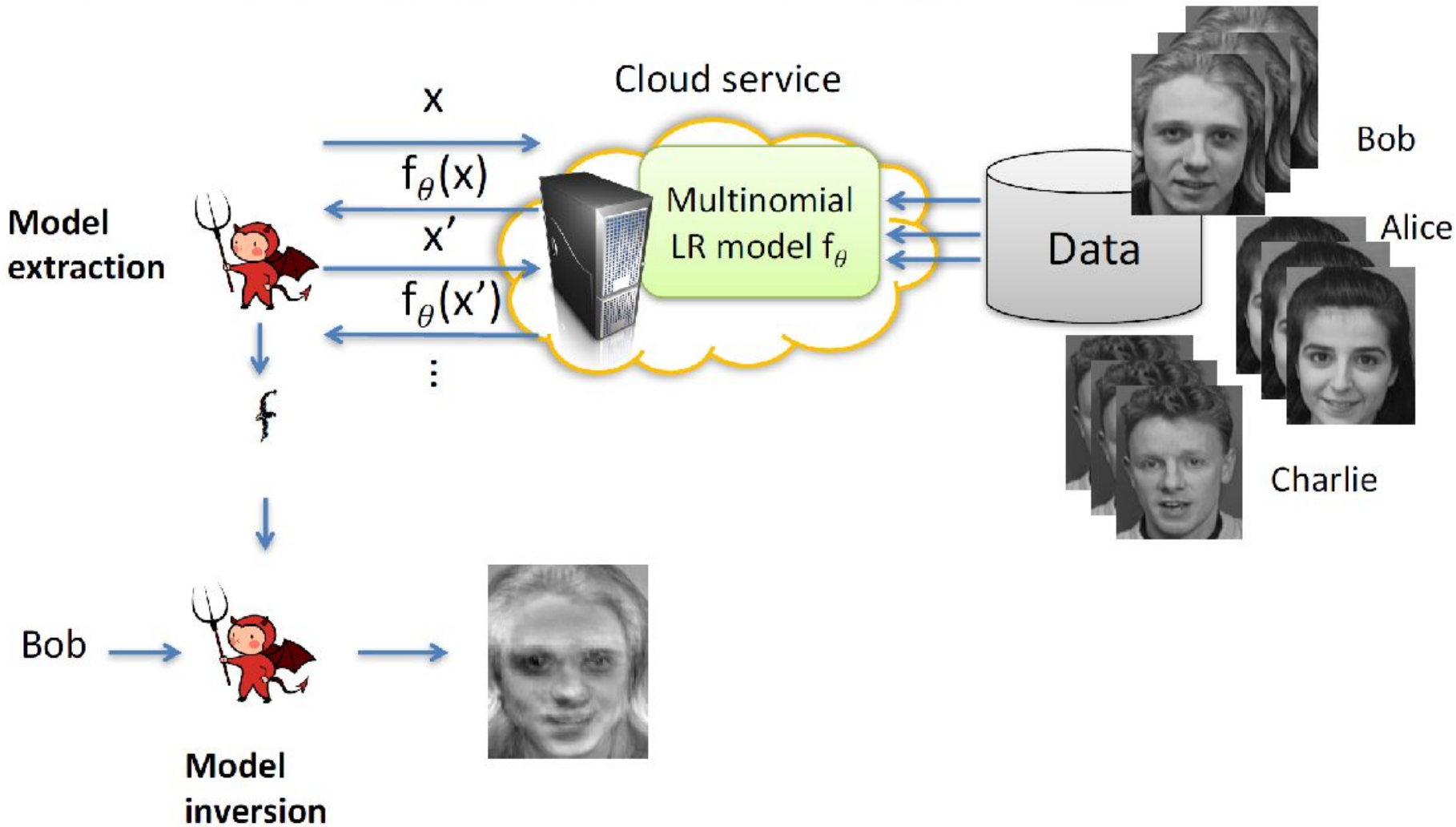
Makes model extraction much more expensive, but not impossible
(See paper for details)

Access control on models

Don't make sensitive prediction APIs publicly accessible

Concrete example: recovering recognizable faces

Given access to facial recognition model f_θ can we reconstruct recognizable images of training set members?



Privacy issues in disclosing ML models

Adversary uses θ to infer information about training set members

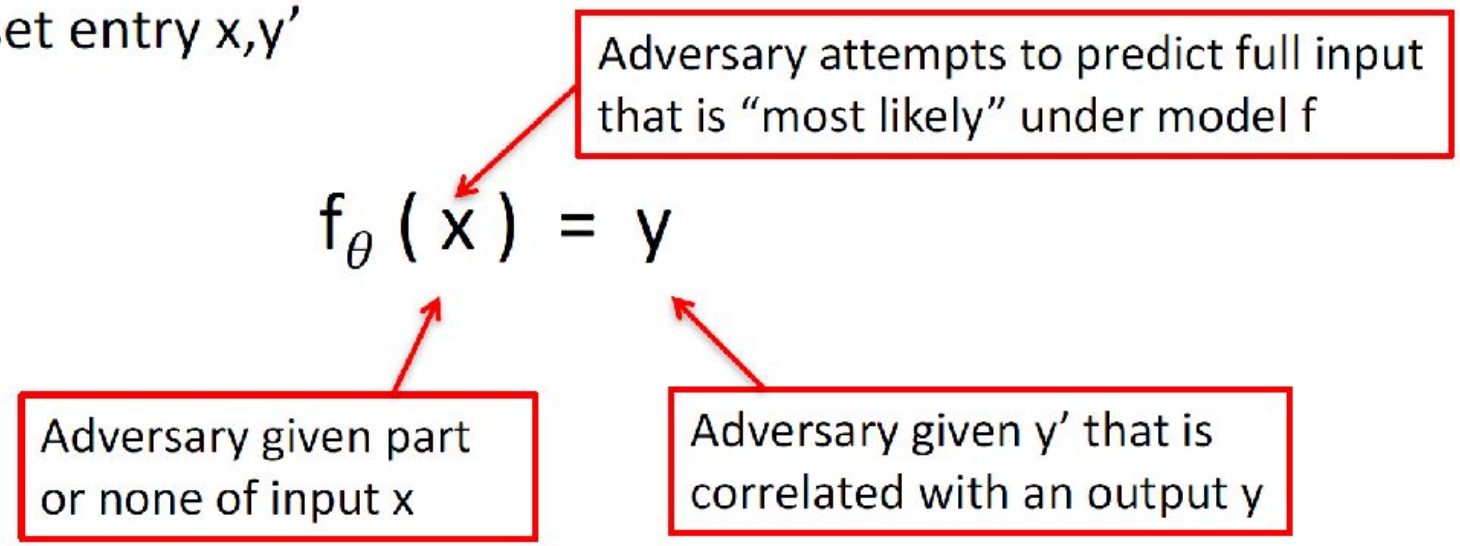
[Ateniese et al. 2015]: Guess one bit about full training data set

[Shokri et al. 2017]: Determine if x,y pair was in training set

Model inversion attacks

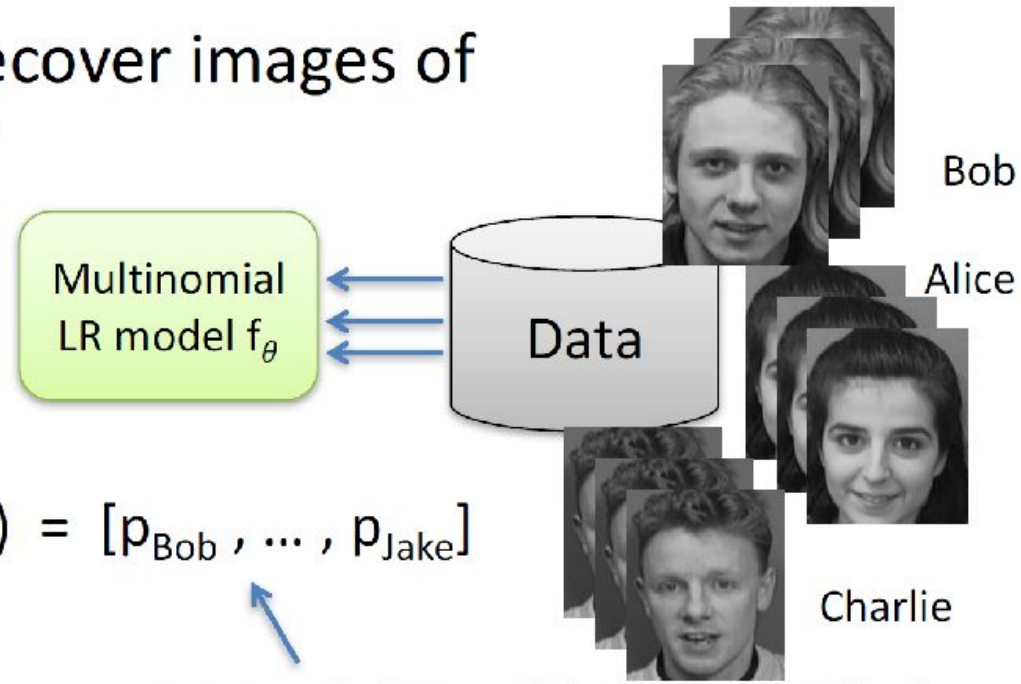
[Fredrikson, et al. 2014, 2015]

Training set entry x,y'



Model inversion on facial recognition

Can adversary use θ to recover images of training member's faces?



$$f_\theta (x_1 , \dots , x_n) = [p_{Bob} , \dots , p_{Jake}]$$

Pixel data

Vector of class confidences each in [0,1]
Output label of highest confidence class

Approach (slightly simplified):

Given θ, y' = "Bob", find input x that is most likely to match "Bob"

- Search for x that maximizes p_{Bob}
- Can search efficiently using gradient descent

Can repeat for all class labels

Example outputs of MI attack for different models


Target


Softmax


MLP

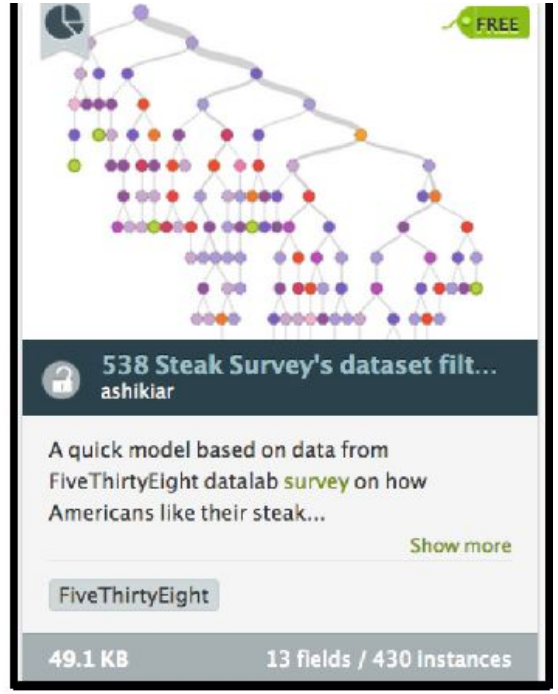

DAE

Trained on AT&T faces dataset (40 individuals, 400 images)
Inversion for three neural-network classifiers :
Multinomial LR, Multi-layer perceptron, Denoising auto-encoder
Mechanical Turk experiments: re-identify person up to 95% accuracy

Open questions:

- Inversion on state-of-the-art facial recognition (e.g., Deepface)? See also Google's Deep Dream
- Improved black-box attacks (access only to f_θ, not θ)

538 Steak Survey on BigML.com



Survey of 332 people to determine if "risky" lifestyle choices correlates with steak preferences

Trained decision tree model:

$$f_\theta (x_1 , \ldots , x_n) = y$$

Household income
Whether person gambles
Whether cheated on significant other
...

Prediction of how person likes steak prepared:
- rare
- medium-rare
- medium
- medium-well
- well-done

Plus confidence value

De-identified training dataset available, we use to simulate attacks

538 Steak Survey on BigML.com

Let $x_1, \ldots, x_n, y'$ be row from training set for f_θ

Adversary attempts to predict Infidelity status

$$f_\theta (x_1, \ldots, x_n) = y$$

Give adversary information other than infidelity status

Give adversary true steak preference y' (not necessarily equal to y)

<u>Given:</u>

$x_1, \ldots, x_{n-1}$

Actual steak preference y'

Model f_θ

(Includes independent priors & confusion matrix error model)

Model inversion algorithm

<u>Predict:</u>

Infidelity status x_n

Generic model inversion as a MAP estimator

Given f_θ , x_1 , … , x_{n-1} , y' predict x_n

x_n takes on possible values in set $\{v_1,\ldots,v_s\}$

Runs in time O(s)

(1) Compute feasible set of input vectors:

$(x_1,\ldots,x_{n-1},v_1)$

$(x_1,\ldots,x_{n-1},v_2)$

…

$(x_1,\ldots,x_{n-1},v_s)$

Uses f_θ as black box

(2) Compute $y_j = f_\theta(x_1,\ldots,x_{n-1},v_j)$ for each j

(3) Output v_j that maximizes

Realizes MAP estimator
(optimal subject to info available)

$$\pi(y', y_j) \cdot p(v_j) \prod_{i=1}^{n-1} p(x_i)$$

Gaussian error model

Independent priors

538 Steak Survey on BigML.com

Let $x_1, \ldots, x_n, y'$ be row from training set for f_θ

$$f_\theta(x_1, \ldots, x_n) = y$$

Adversary attempts to predict Infidelity status

Give adversary information other than infidelity status

Give adversary true steak preference y' (not necessarily equal to y)

On BigML.com θ includes # training set instances matching each leaf

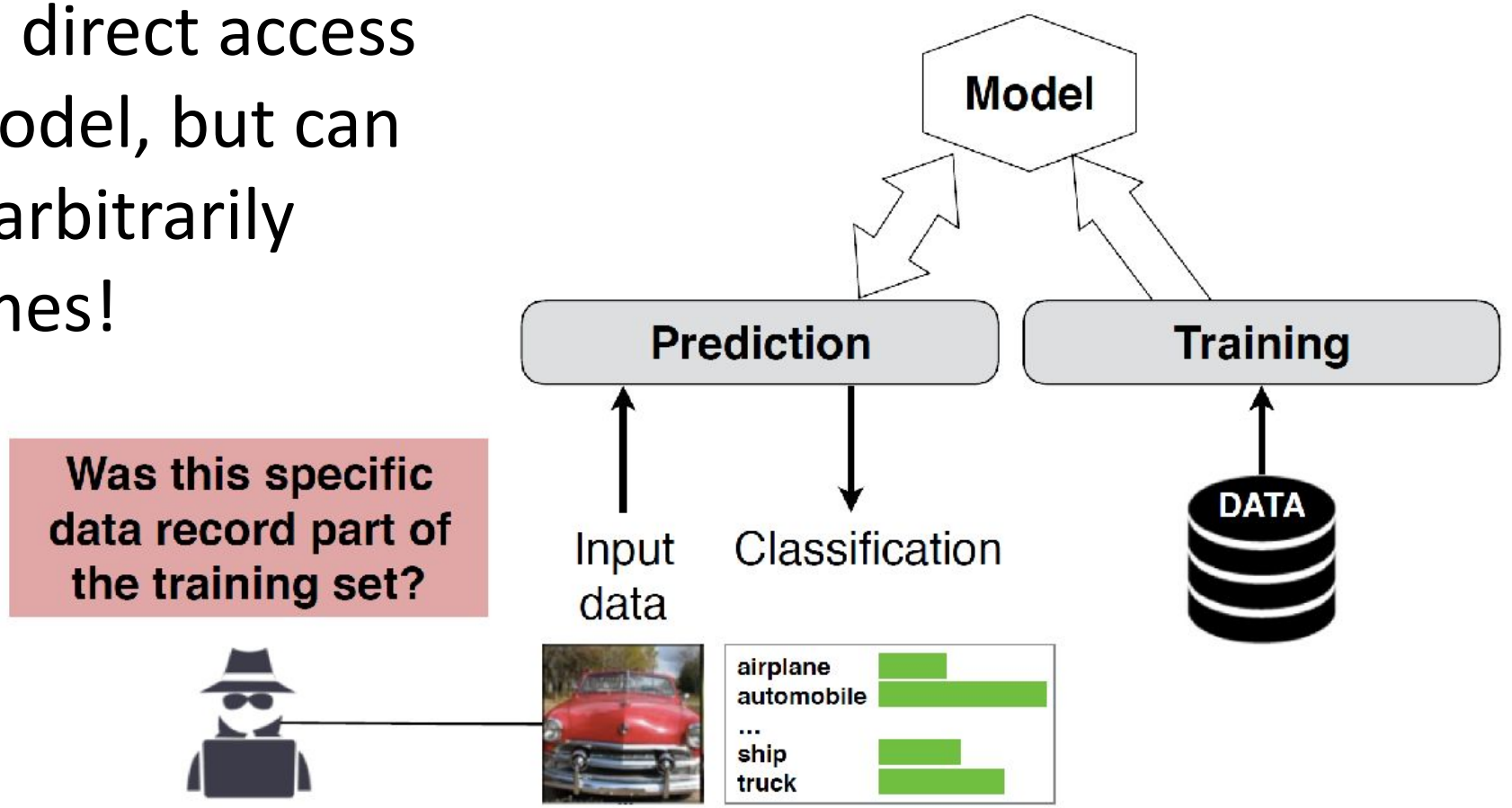
We give a whitebox MAP estimator that takes into account this additional information.

	Accuracy	Precision
Black-box MAP	85.8%	85.7%

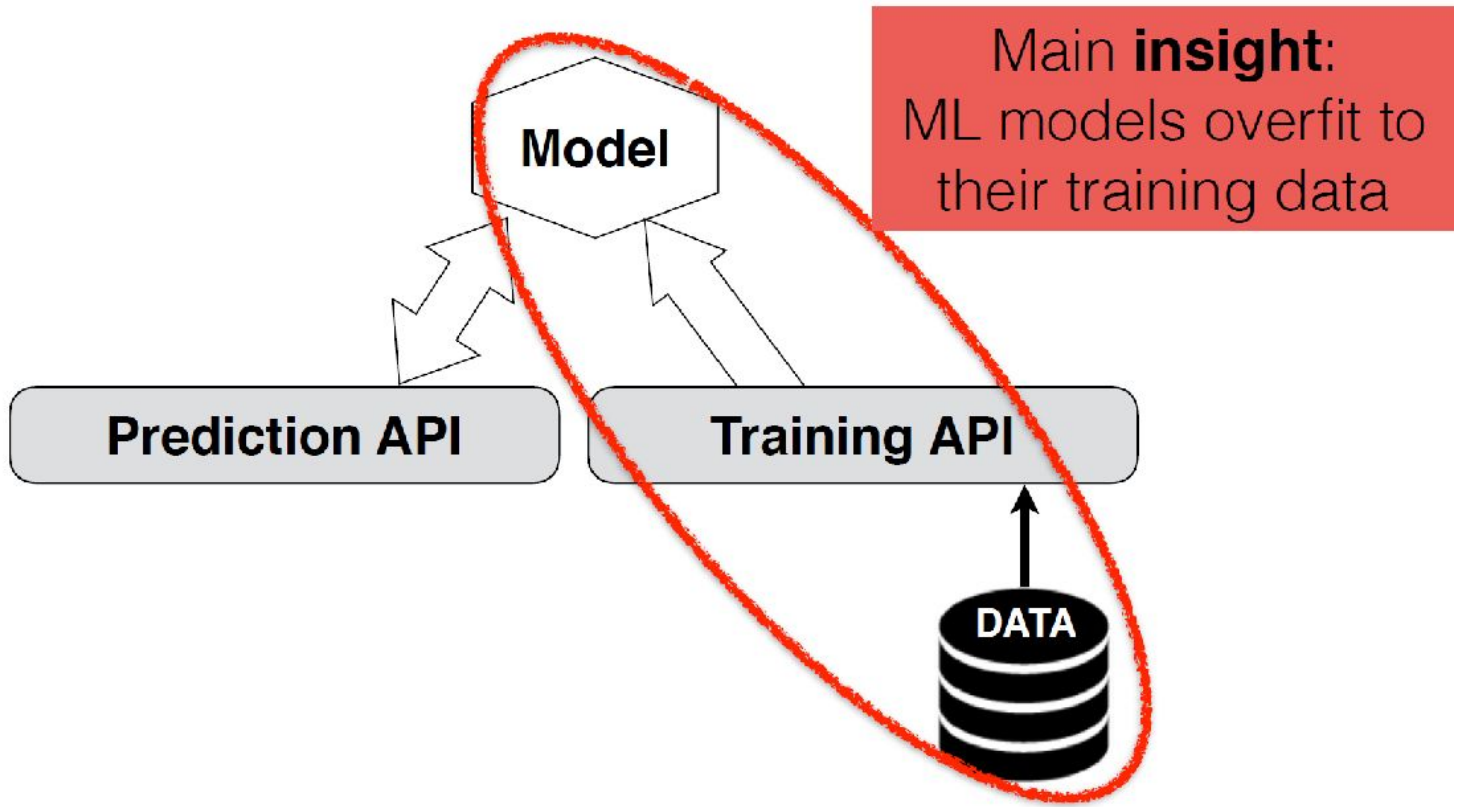
100% precision for members of training set
(< 20% precision for non-members)

Membership Inference Attack

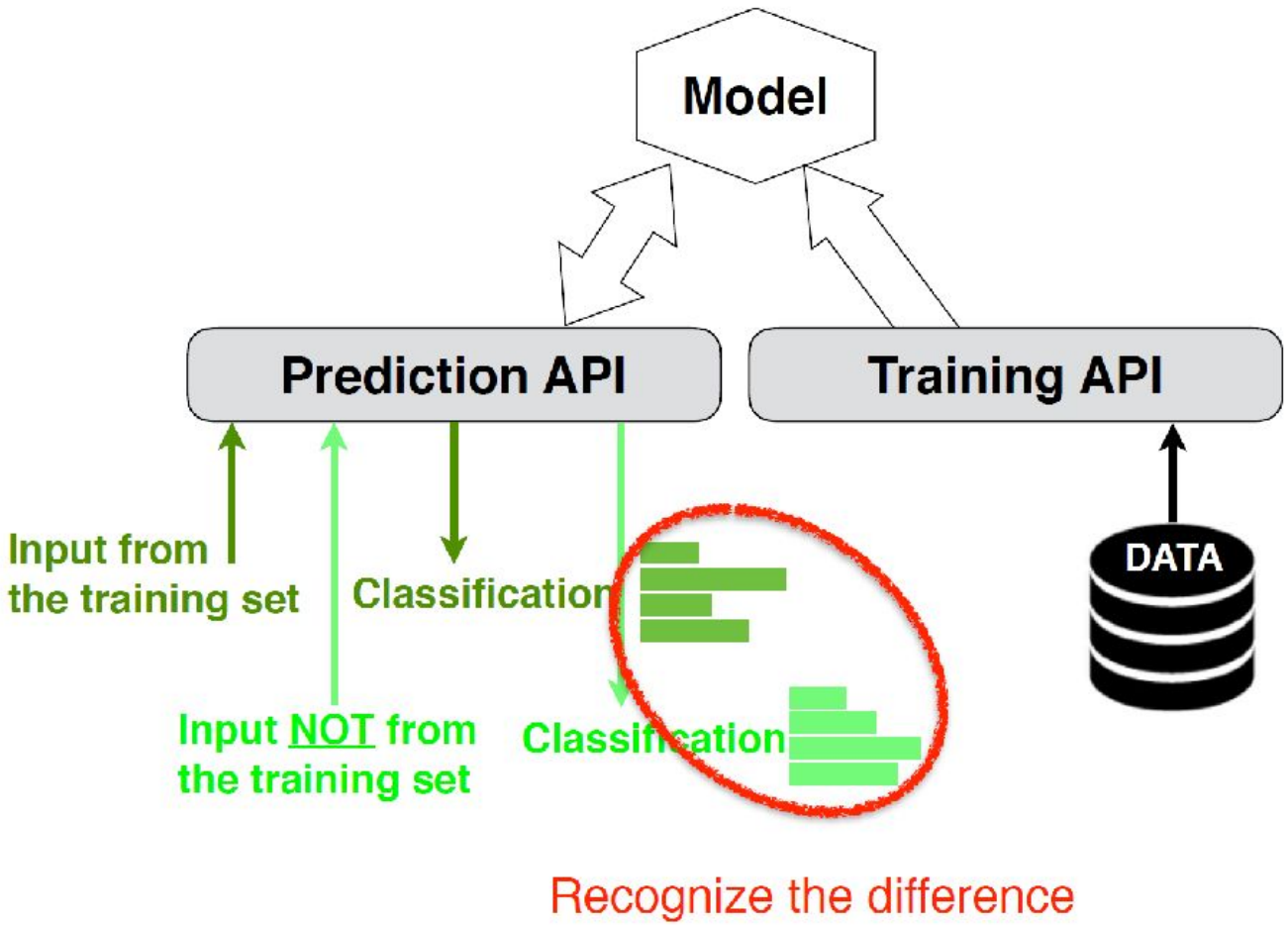
Note: Attacker does not have direct access to the model, but can query it arbitrarily many times!



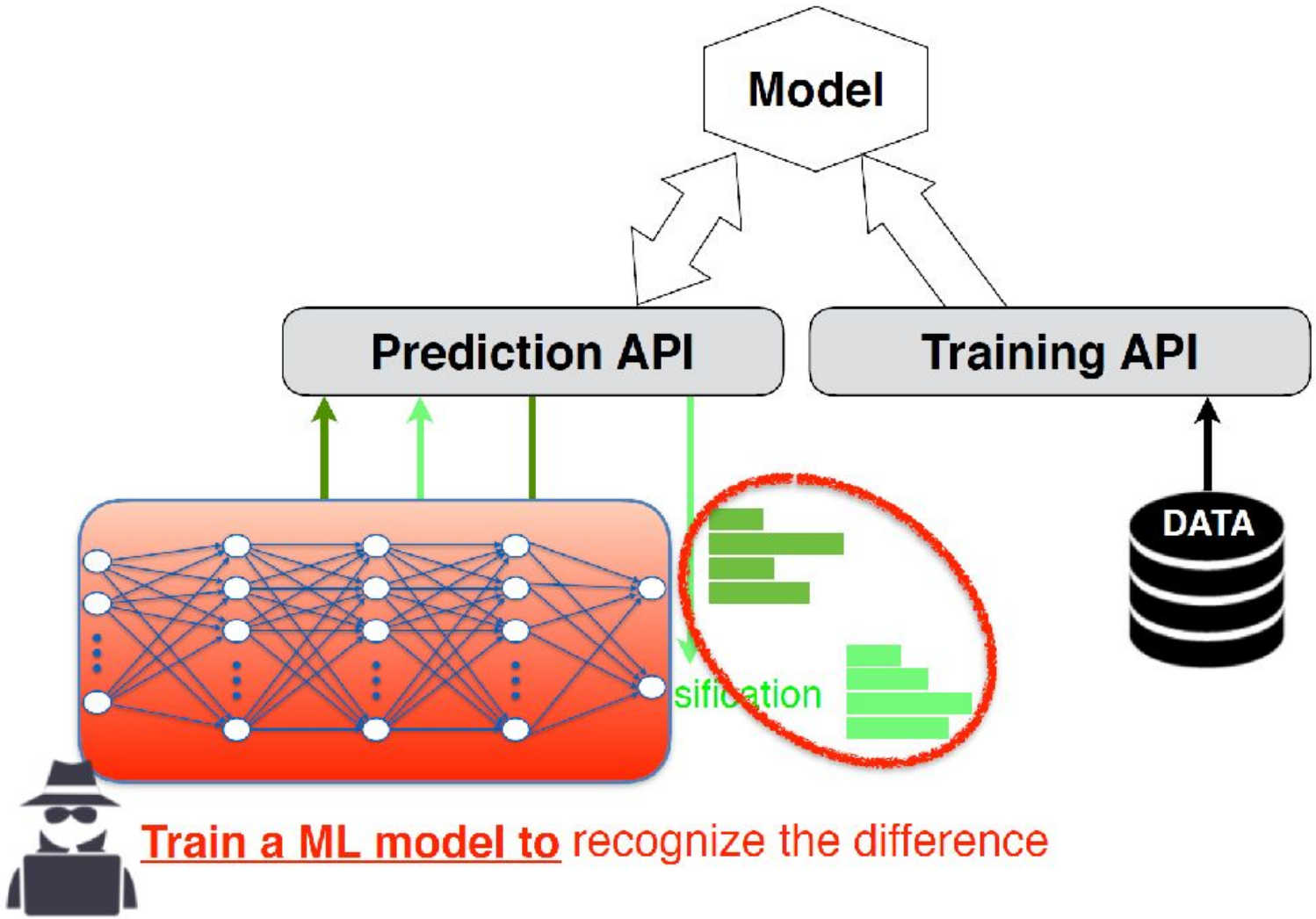
Exploit Model's Predictions



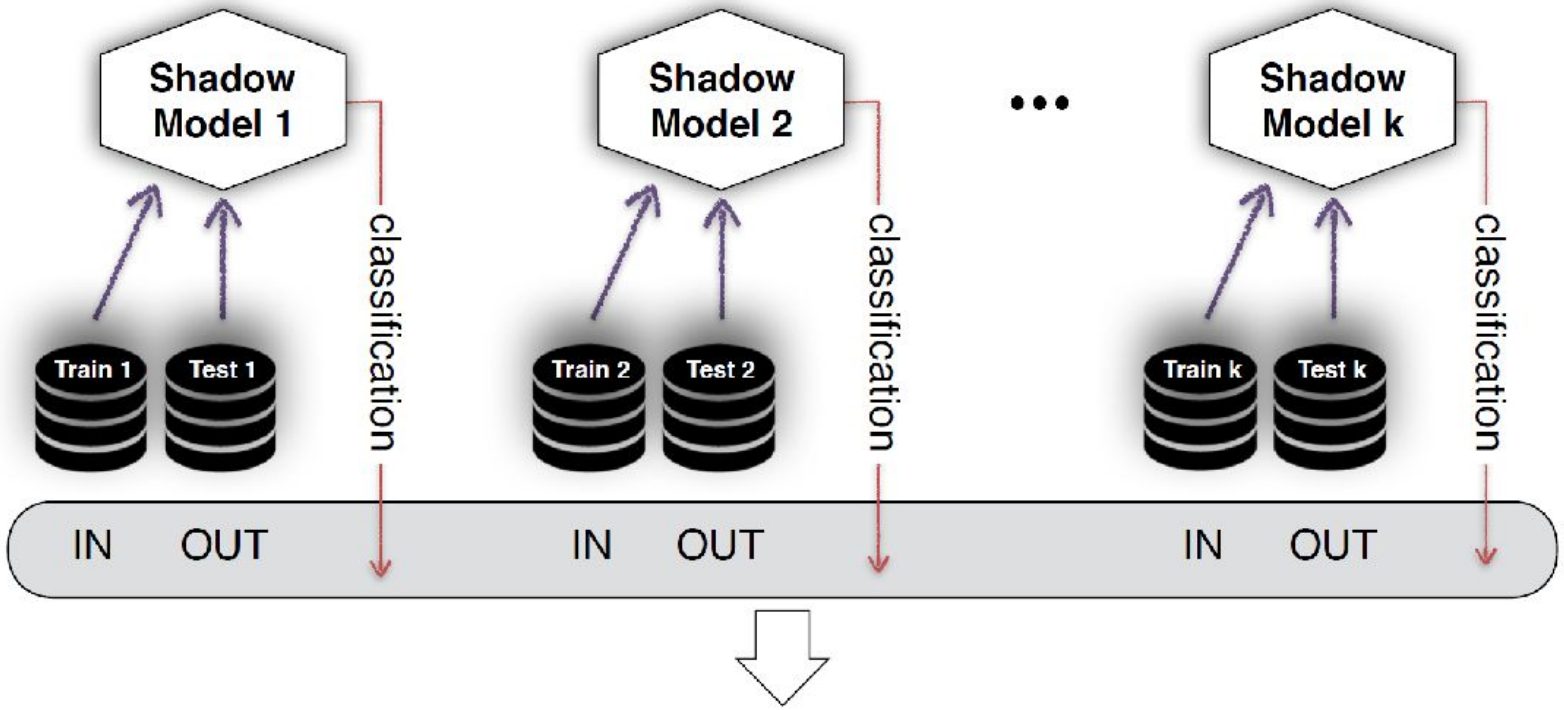
Exploit Model's Predictions



ML against ML



Train Attack Model using **Shadow Models**



Train the attack model

to predict if an input was a member of the training set (in) or a non-member (out)

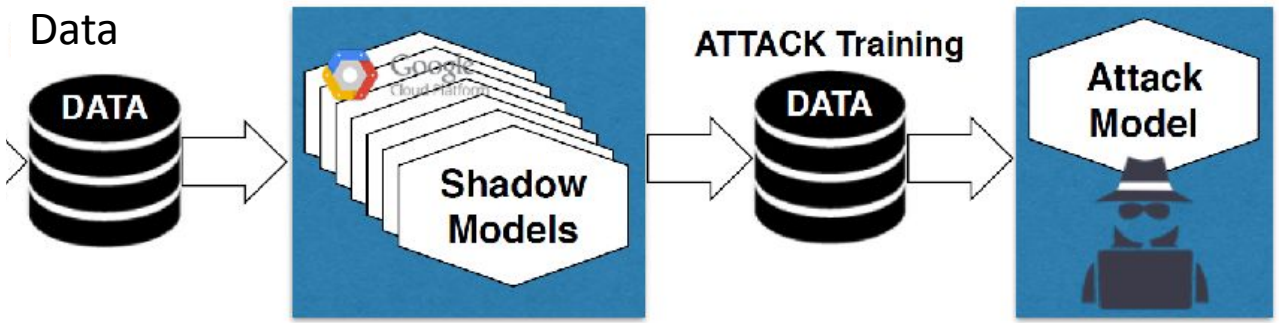
Obtaining Data for Training Shadow Models

- **Real**: similar to training data of the target model (i.e., drawn from same distribution)

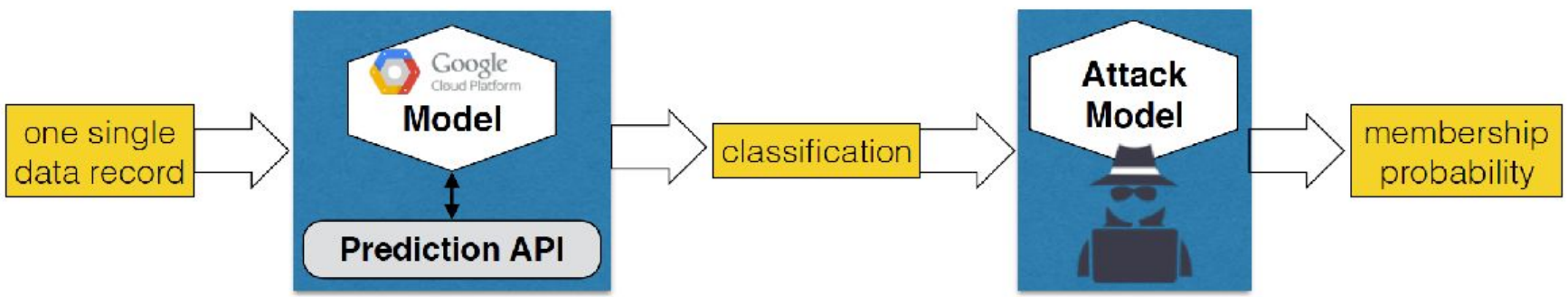
For face recognition, for example, the new training dataset can be another set of labeled faces obtained from the internet

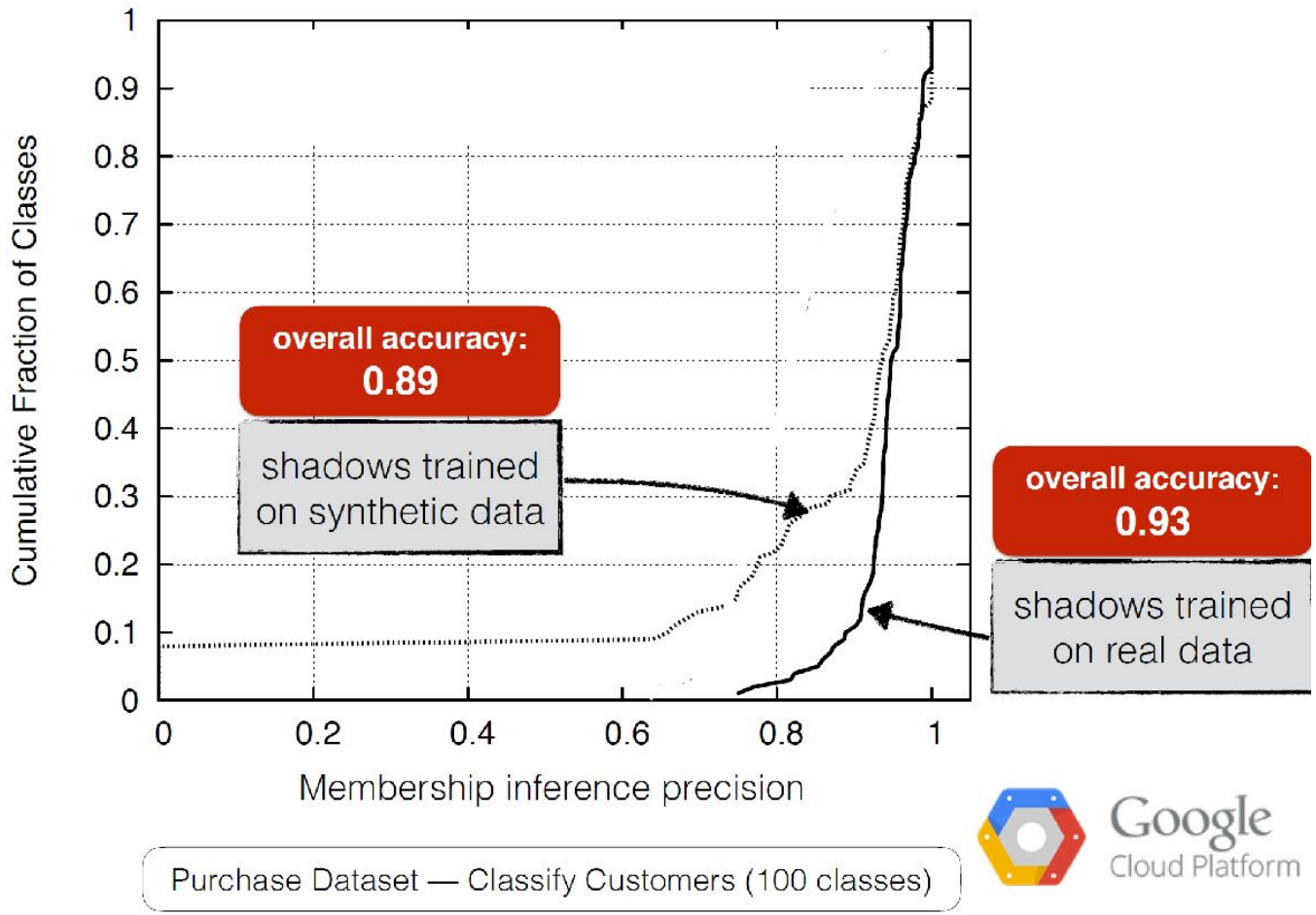
Constructing the Attack Model

New
Training
Data



Using the Attack Model




Cumulative Fraction of Classes
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
Membership inference precision
overall accuracy:
0.89
shadows trained
on synthetic data
overall accuracy:
0.93
shadows trained
on real data
Purchase Dataset — Classify Customers (100 classes)
Google
Cloud Platform

Protecting Against Membership Inference

Can we prevent ML models from revealing information about their training data?

Differential Privacy to the rescue?

http://sigmod2017.org/wp-content/uploads/2017/03/04-Differential-Privacy-in-the-wild-1.pdf