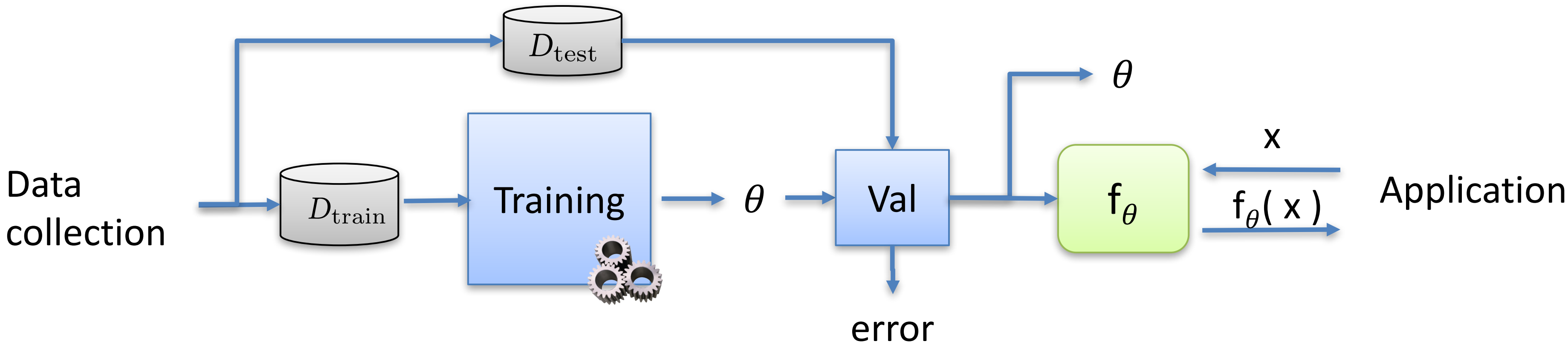
Confidentiality and Privacy Threats in Machine Learning

Thomas Ristenpart

CORNELL TECH

Machine learning (ML) systems



(1) Collect labeled data x[1] , y[1] x[2], y[2] ...

n-dimensional feature vector x

Dependent variable y

(2) Train & validate ML model θ to allow prediction: $f_\theta(x) = y$

(3) Use f_θ in some application or publish θ for others to use

Examples: ML-as-a-service

Amazon Machine Learning

bigml

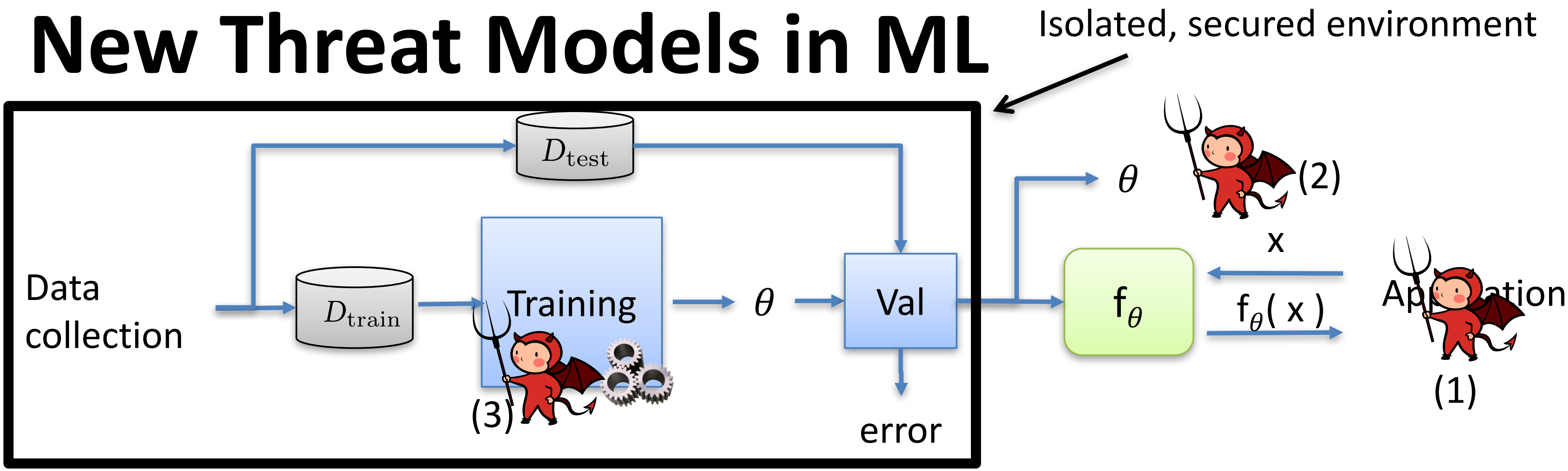
Google Cloud Platform

ALGORITHMIA

Microsoft Azure
Machine Learning

Service	Model types / training algorithms
Amazon	Logistic regression
Google	Logistic regression, (convolutional / recurrent) neural networks, …
Microsoft	Logistic regression, decision trees, neural networks, SVM
BigML	Logistic regression, decision trees
Algorithmia	Custom training algorithms (from third party developers)

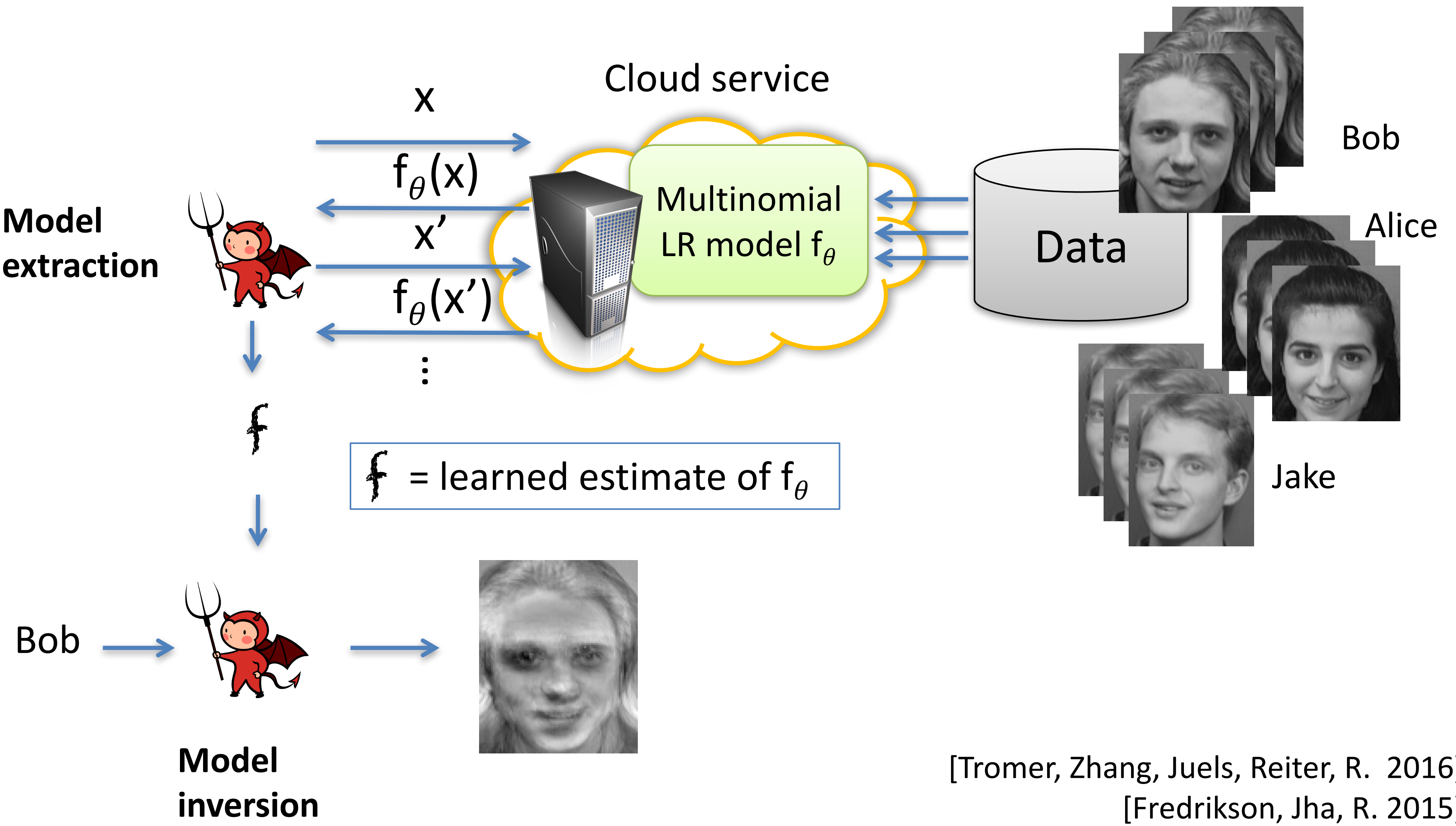
New Threat Models in ML



Adversarial goal(s)	Adversarial abilities

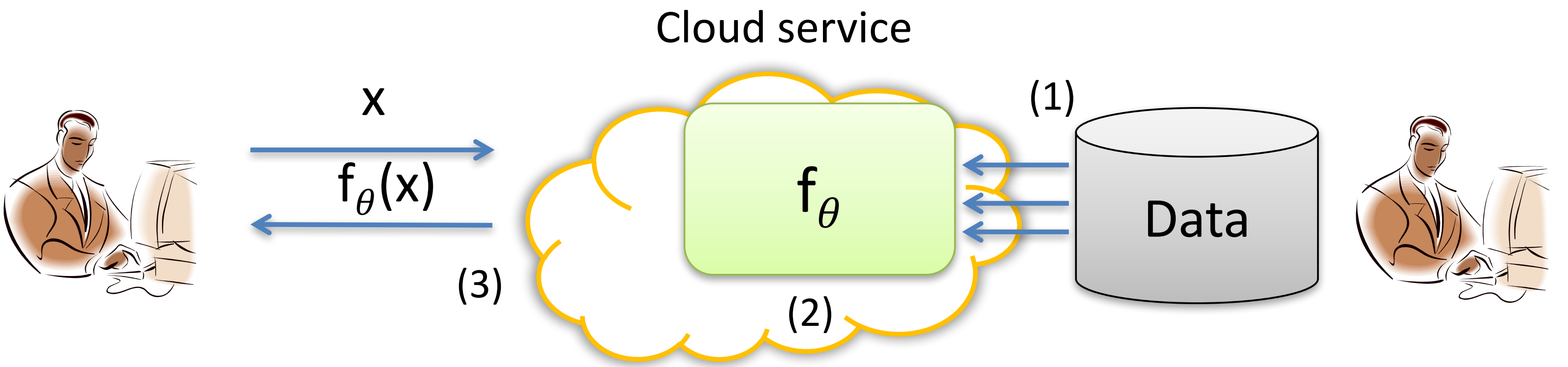
Concrete example: recovering recognizable faces

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



[Tromer, Zhang, Juels, Reiter, R. 2016]
[Fredrikson, Jha, R. 2015]

Closer look: ML-as-a-service

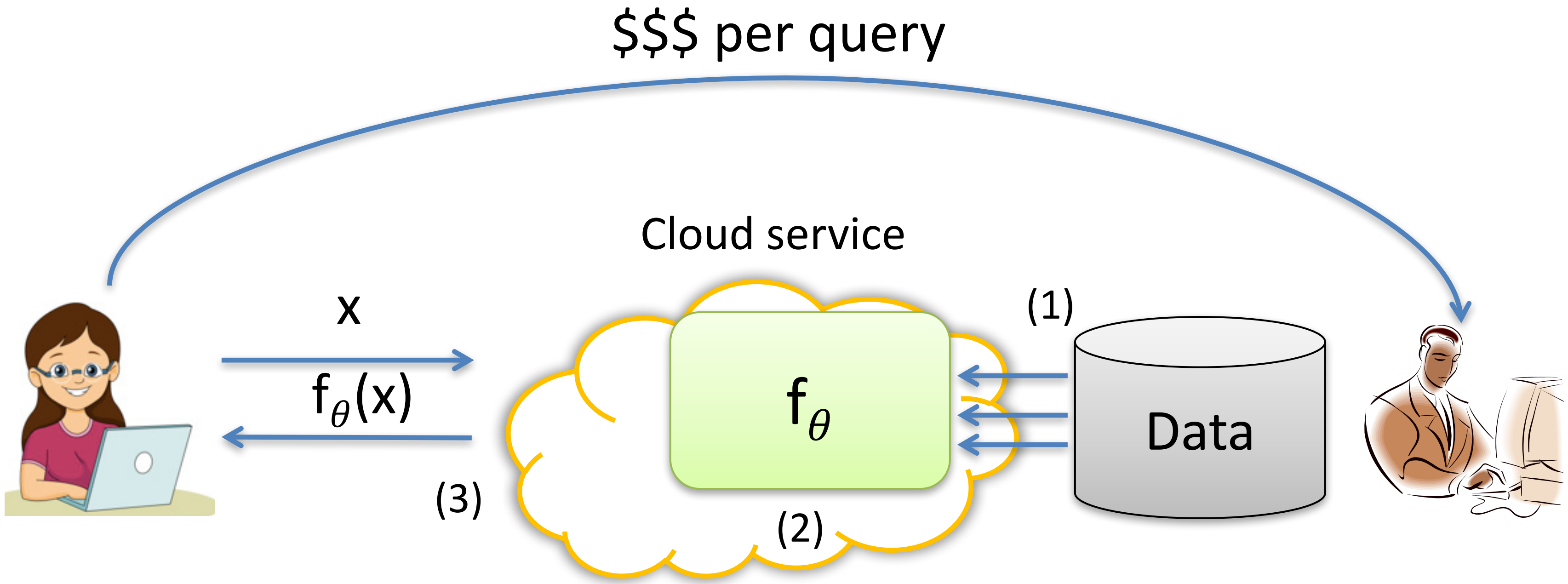


(1) Data owner uploads data

(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*

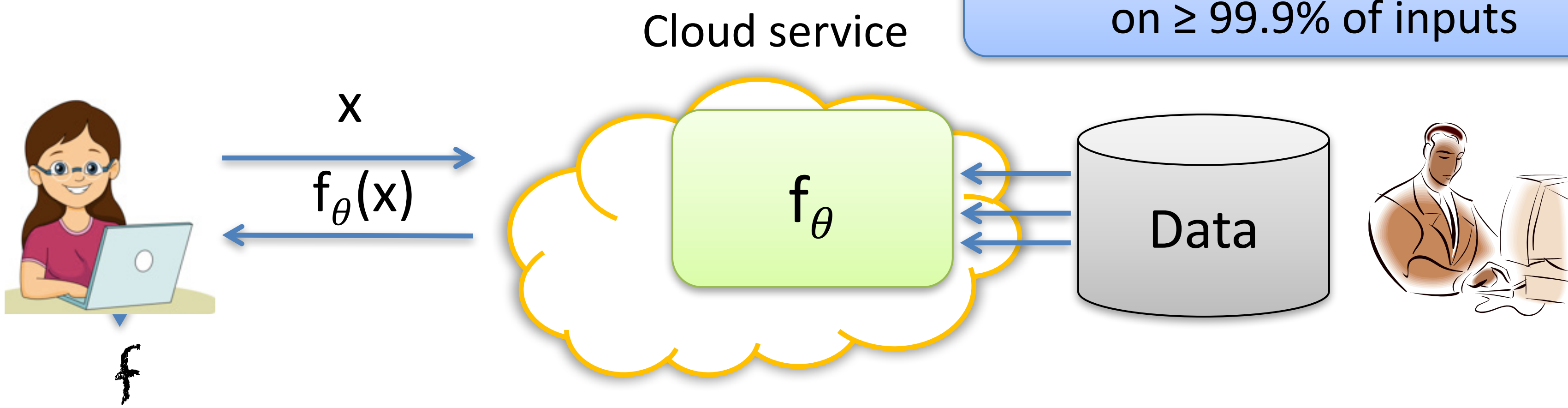
Refer to this as
black-box setting

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:

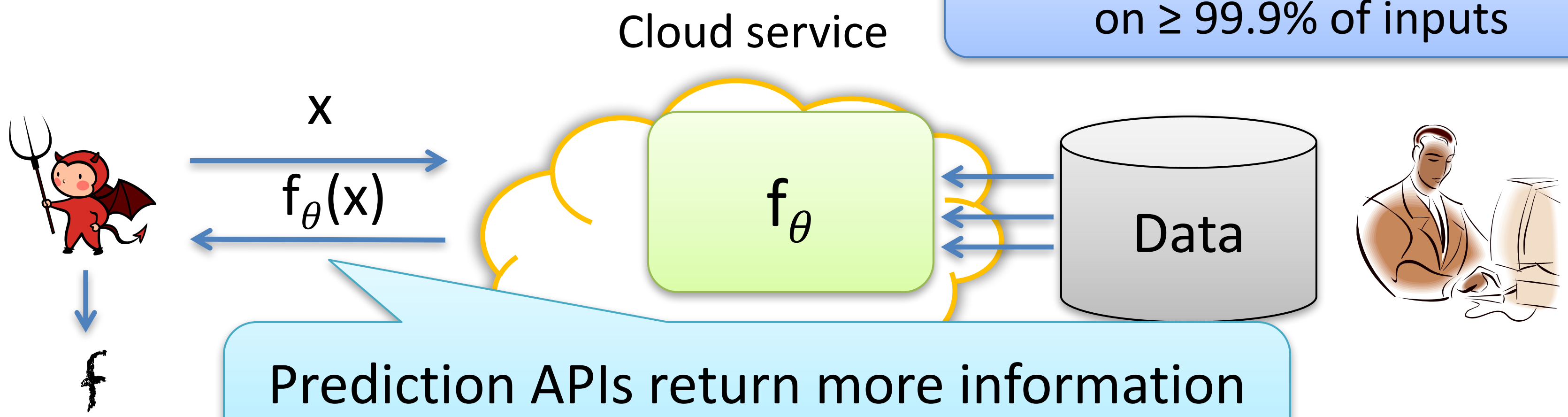
- undermine pay-for-prediction pricing model
- facilitate privacy attacks (stay tuned)
- enable evasion attacks

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 $f(x) = f_\theta(x)$ on ≥ 99.9% of inputs



Prediction APIs return more information than assumed in prior work

If $f_\theta(x)$ just class label: learning with membership queries setting

- [Kushilevitz, Mansour – 1993] for boolean decision trees

  Polytime but not practical

- [Lowd, Meek – 2005] linear models (e.g., binary logistic regression)

  See paper for generalizations and experimental results

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

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

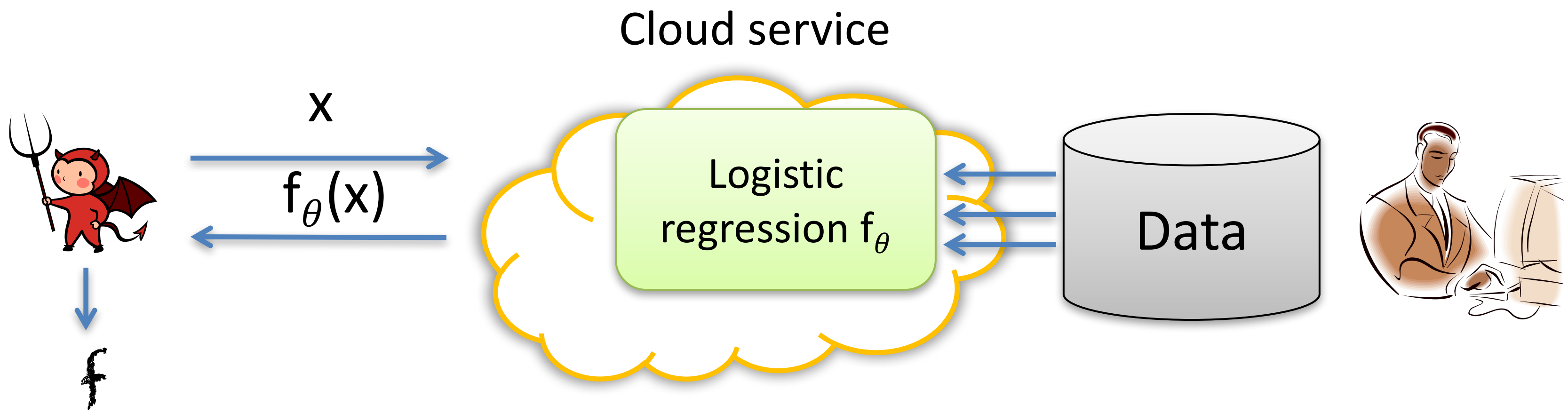
f_θ maps features to predicted probability of being "Alice"
≤ 0.5 classify as "Bob"
> 0.5 classify as "Alice"

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

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

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

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

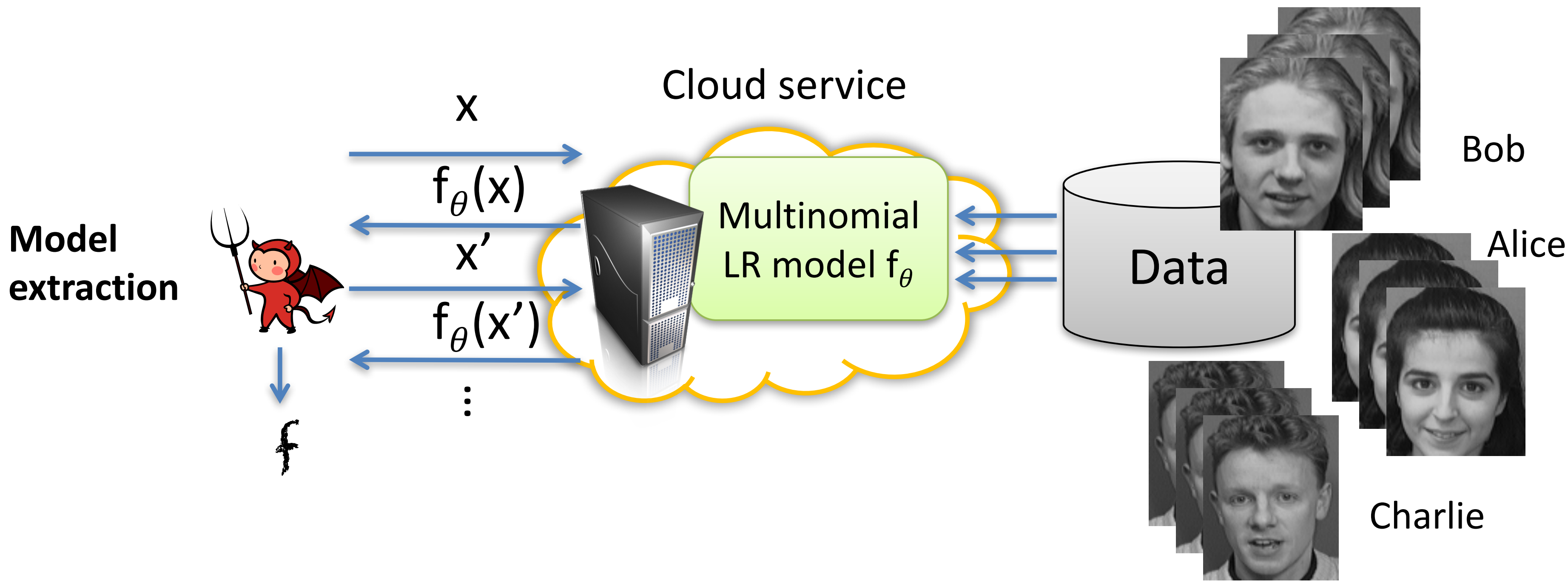
100% accuracy

$\hat{f}(x) = f_\theta(x)$ on all x

More detailed results in paper

Concrete example: recovering recognizable faces

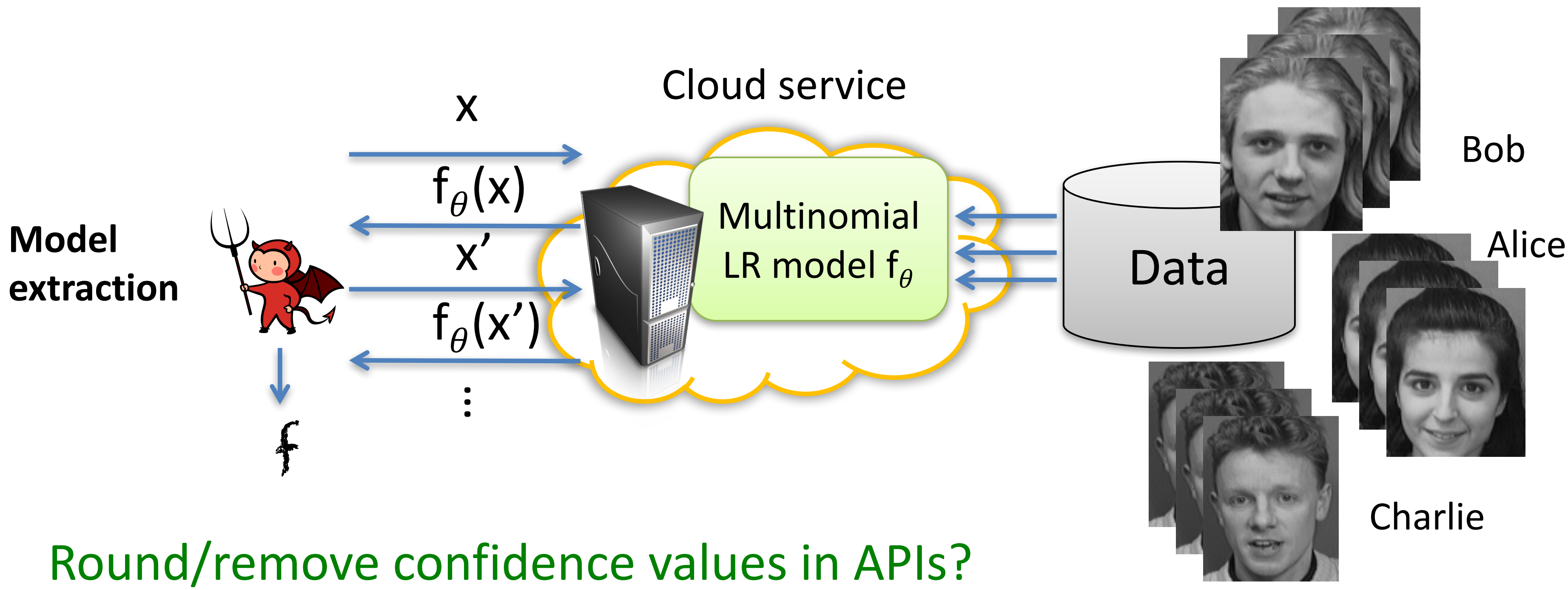
Given access to facial recognition model f_θ can we reconstruct recognizable images of training set members?



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?

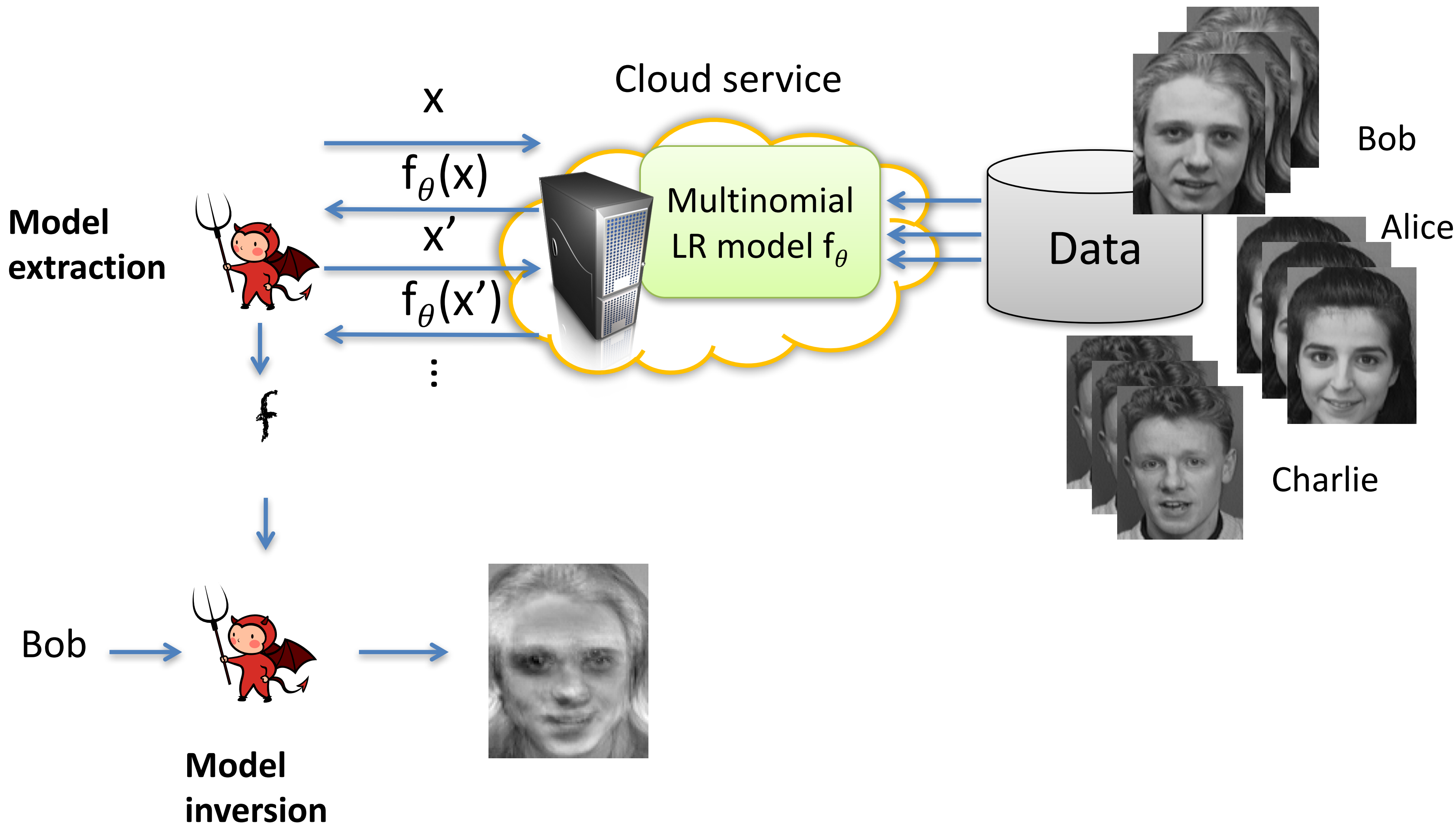
Makes model extraction much more expensive, but not impossible
(See paper for details)

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'

Adversary attempts to predict full input that is "most likely" under model f

$$f_\theta (x) = y$$

Adversary given part or none of input x

Adversary given y' that is correlated with an output y

Model inversion case studies

(1) Neural networks for facial recognition

Recover recognizable images of training set members

(2) Linear regression for personalized medicine

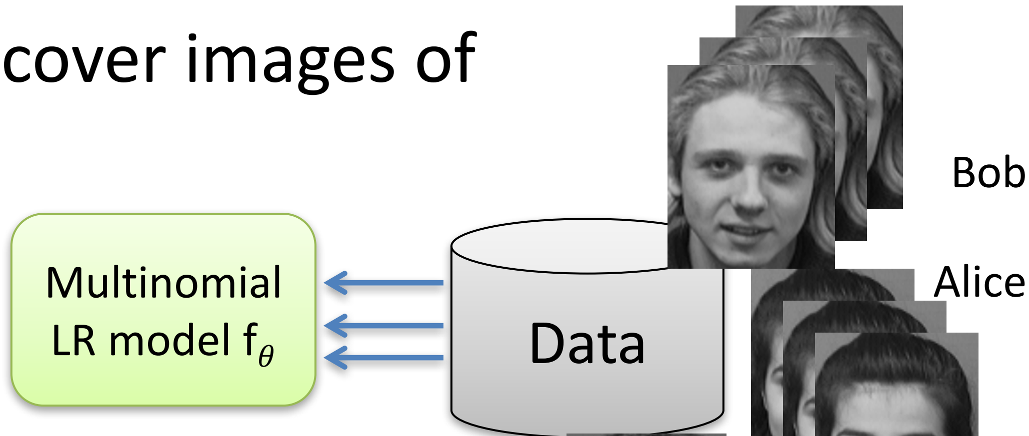
Predict genotypes of patients

(3) Decision trees trained from lifestyle surveys

Predict marital infidelity of training set members

Model inversion on facial recognition

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



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

Pixel data

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

<u>Approach (slightly simplified):</u>

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)

Model inversion case studies

(1) Neural networks for facial recognition

Recover recognizable images of training set members

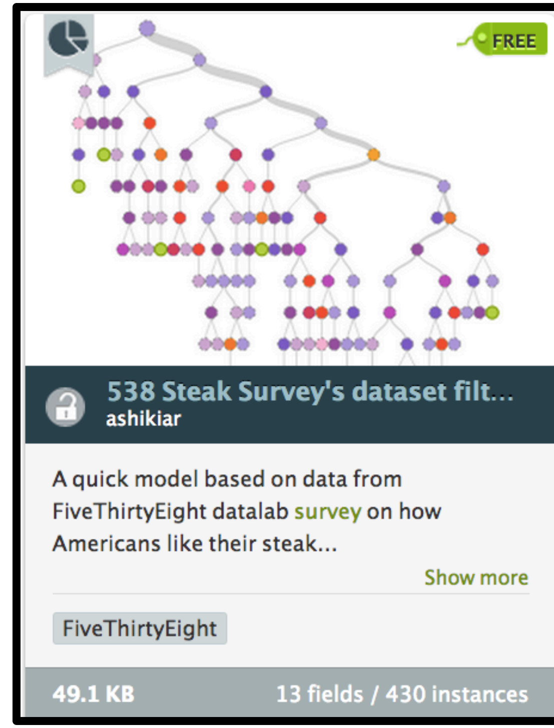
(2) Linear regression for personalized medicine

Predict genotypes of patients

(3) Decision trees trained from lifestyle surveys

Predict marital infidelity of training set members

538 Steak Survey on BigML.com



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

Trained decision tree model:

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

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

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

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

Realizes MAP estimator
(optimal subject to info available)

(3) Output v_j that maximizes

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

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

Adversary attempts to predict Infidelity status

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

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)

Model inversion countermeasures?

Modify training regime?

- May be able to de-emphasize / remove sensitive features
  - Sensitive-feature aware CART algorithm for decision trees
- Differentially private models
  - Tries to protect individual training set items (see facial recognition case)
  - Utility still not great in many cases

Access control on models

Don't make parameters (or prediction APIs) publicly accessible

Model inversion attacks: recap

Adversary can use a model f to learn about members of training set

Efficacy high because models reveal information

(1) Model gives info on correlations attacker didn't know without it

(2) Model leaks info specific to training set members

Open question: understanding how much models leak about training data

The Recht Hypothesis:

Deep neural networks work well because they memorize most of their training data

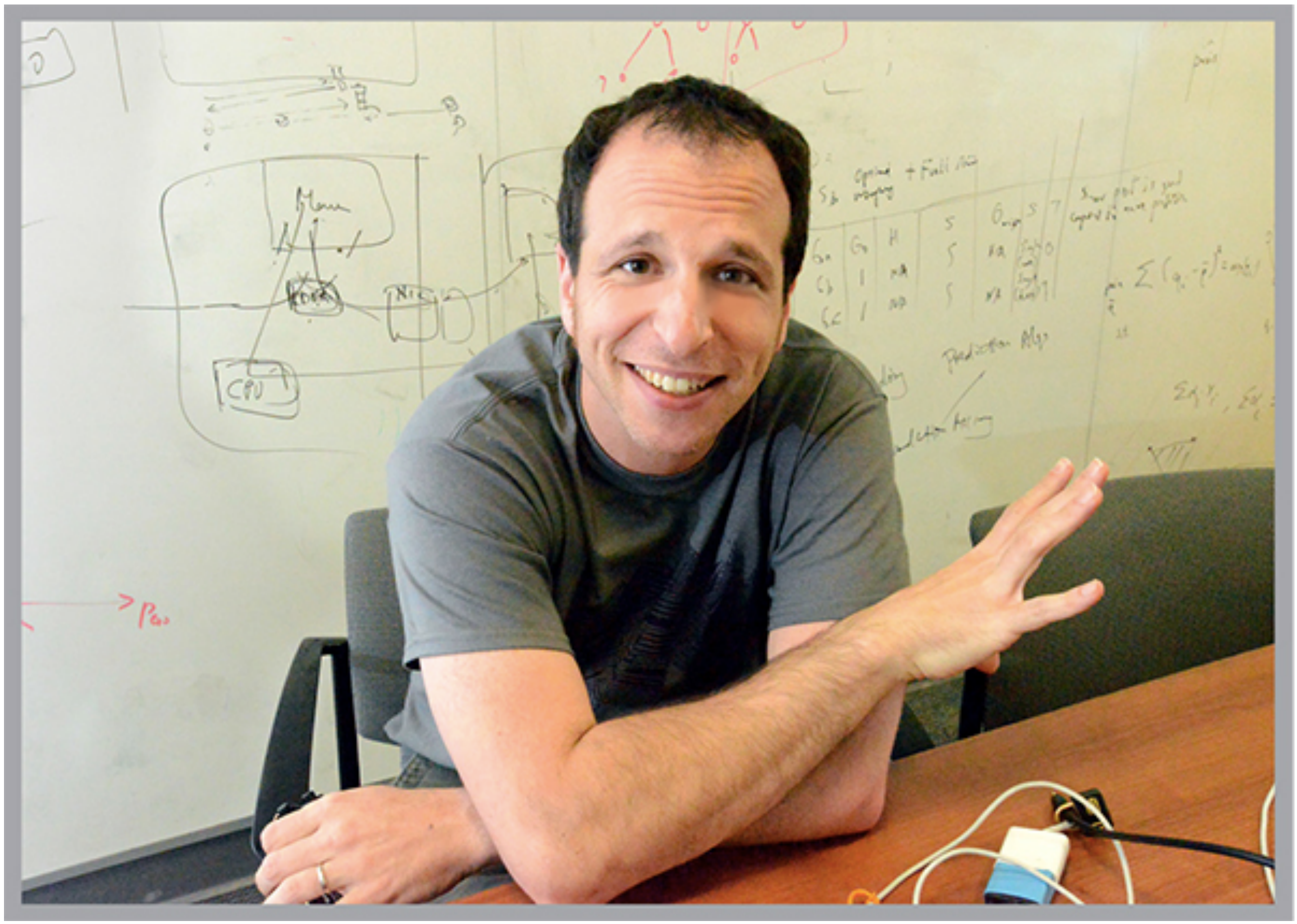
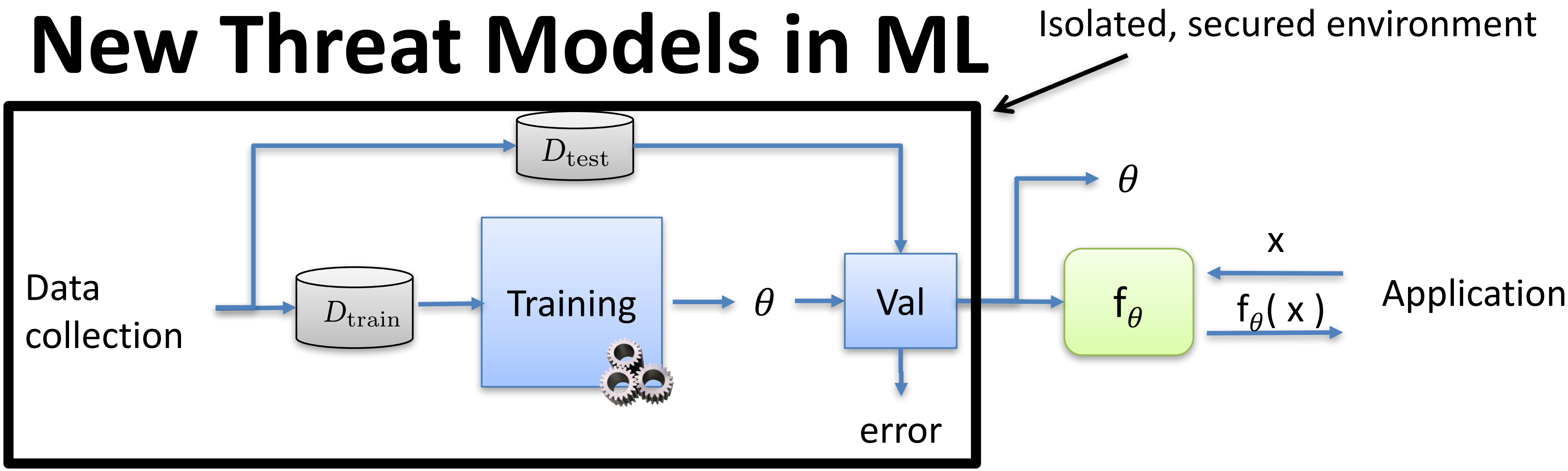


Photo credit: Peg Skorpinski, http://vcresearch.berkeley.edu/news/making-sense-big-data

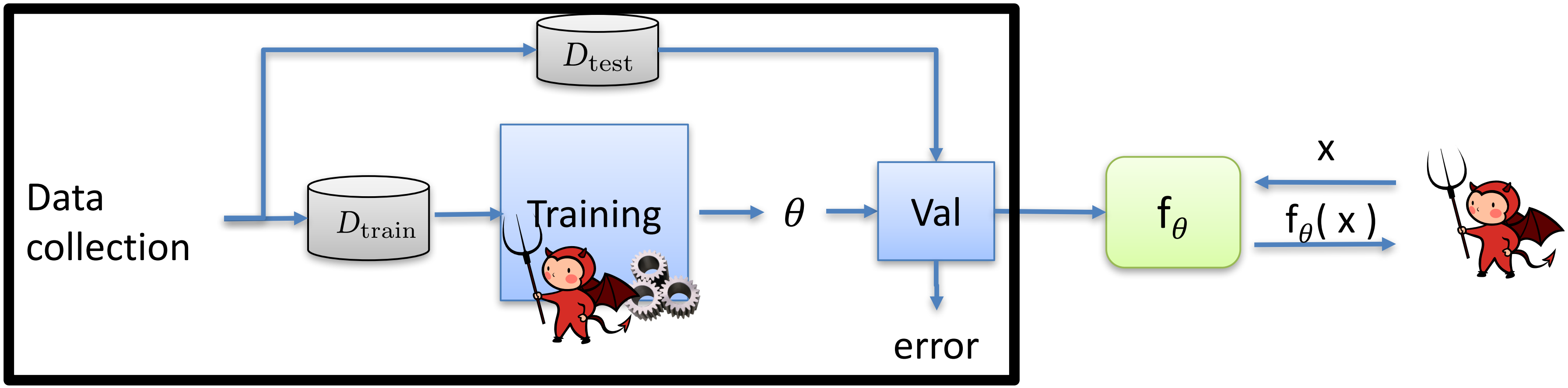
New Threat Models in ML



	Adversarial goal(s)	Adversarial abilities
(1) Model extraction	Learn θ	Query f_θ on adversarial points and see response
(2) Model inversion	Learn information about training data	Access to θ
(3) Malicious training	Exfiltrate training data through validated model θ	Specify Training algorithm, access to f_θ

Malicious training scenario

[Song, R., Shmatikov 2017]



Sensitive data owner somehow tricked into using a malicious algorithm in isolated environment to train deep neural network, logistic regression, etc.

ML algorithm marketplaces (e.g., Algorithmia.com)

Backdoored ML library

Can't trivially steal sensitive data by sending over network

If f_θ validates, then data owner uses it in some adversarially-accessible application

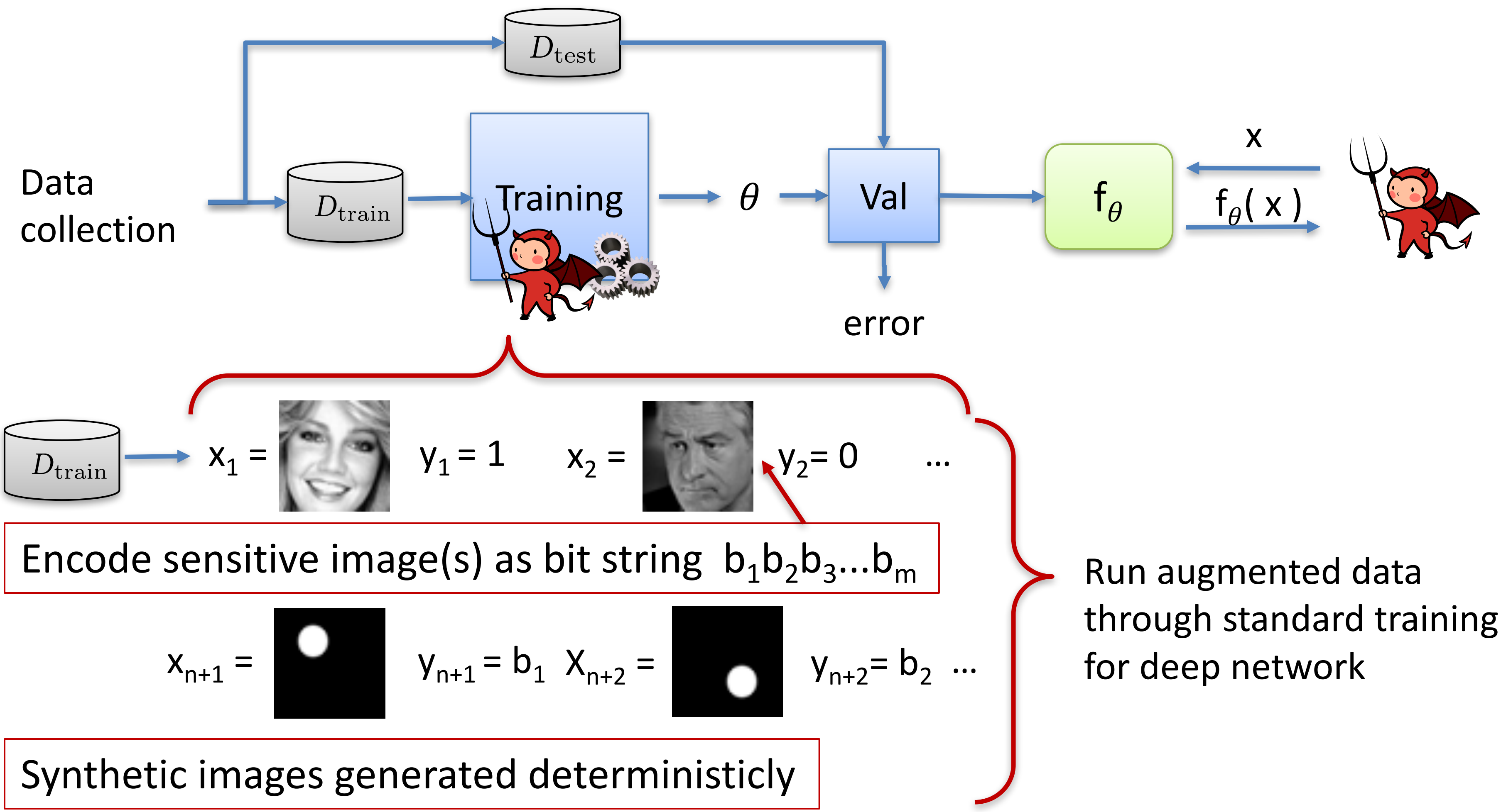
Can we modify Training algorithms so that:
(1) Perform well on primary task
(2) Allows recovery of training data via calls to f_θ

Abusing spare capacity

[Zhang et al. 2017]: deep neural networks can “memorize” random data

Malicious data augmentation can be used to exfiltrate sensitive data

Example: Gender classification task on 50x50 images

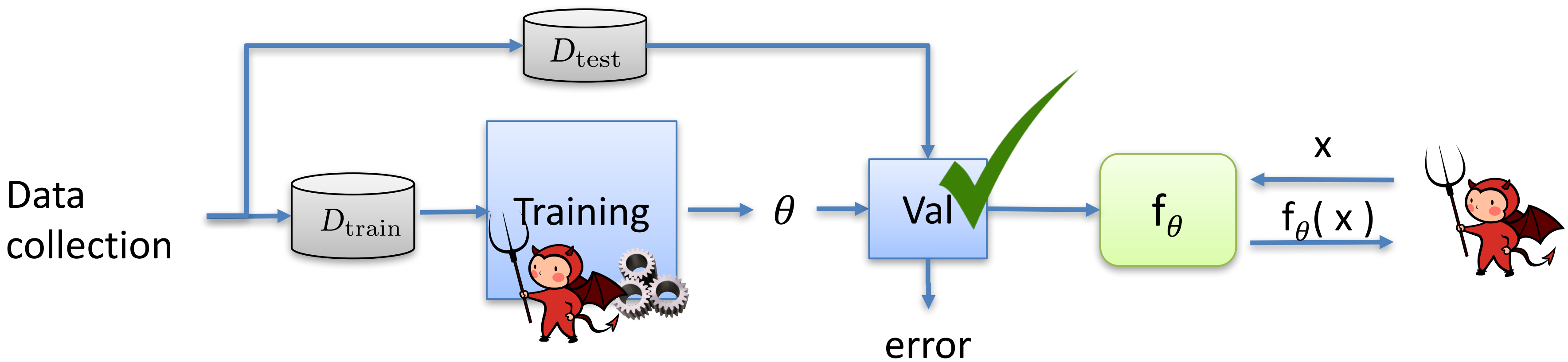


Abusing spare capacity

[Zhang et al. 2017]: deep neural networks can “memorize” random data

Malicious data augmentation can be used to exfiltrate sensitive data

Example: Gender classification task on 50x50 images



Resulting model has high test accuracy. For 34-layer residual network [He et al. 2016] and Facescrub dataset (57k training images):

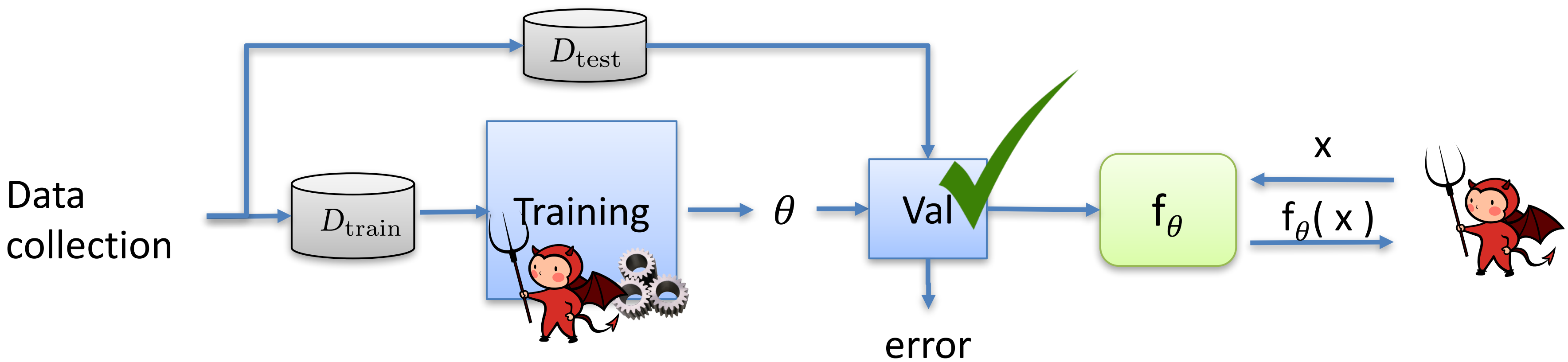
# malicious images added	Test accuracy
0	97.44%
110,000	97.08%
170,000	96.94%

Abusing spare capacity

[Zhang et al. 2017]: deep neural networks can “memorize” random data

Malicious data augmentation can be used to exfiltrate sensitive data

Example: Gender classification task on 50x50 images



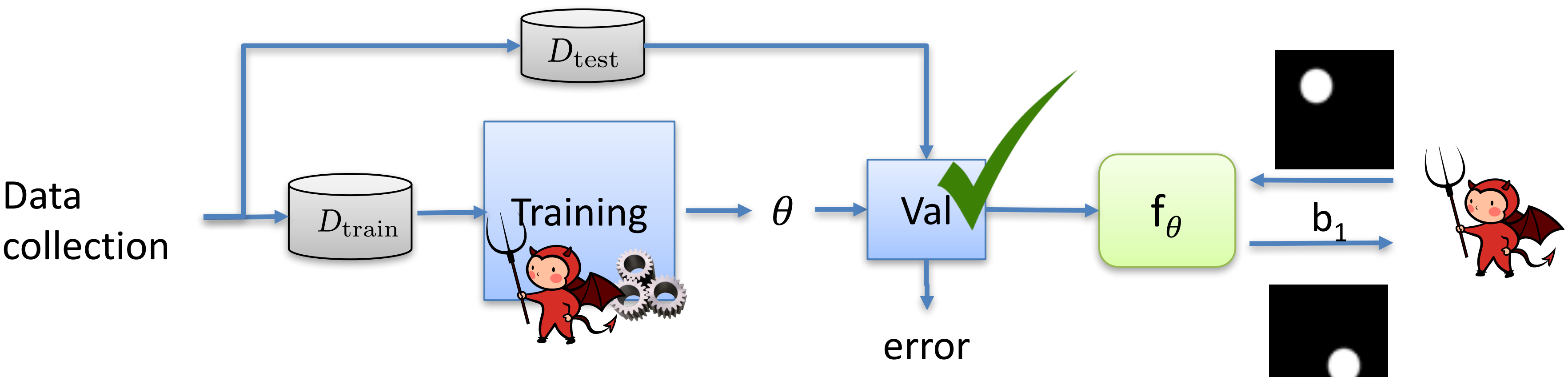
Query synthetic images to recover bit string $b_1b_2b_3...b_m$

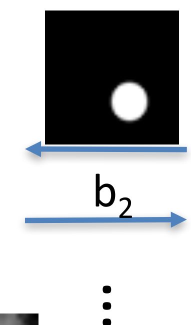
Abusing spare capacity

[Zhang et al. 2017]: deep neural networks can “memorize” random data

Malicious data augmentation can be used to exfiltrate sensitive data

Example: Gender classification task on 50x50 images





Query synthetic images to recover bit string $b_1 b_2 b_3 ... b_m$

Training error on synthetic images dictates error rate

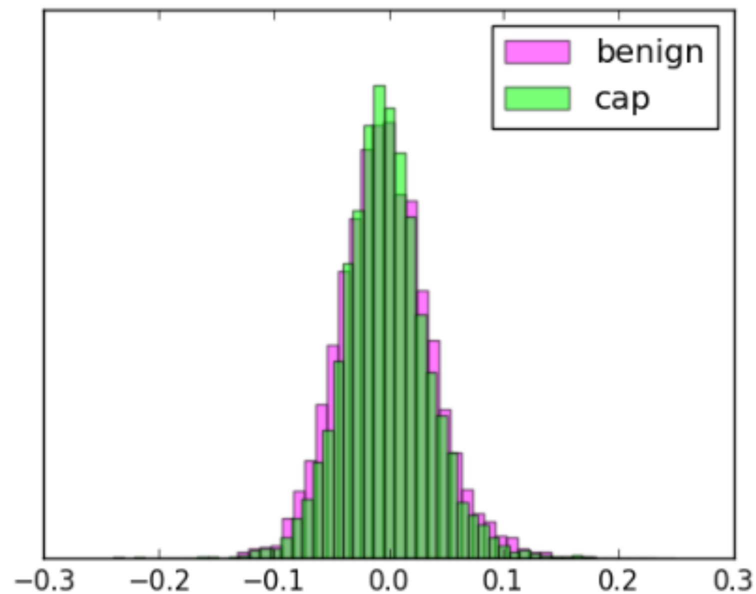
Original Recovered Original Recovered

Take-aways and countermeasures

Parameters for typical architectures sufficient channel for leaking sensitive training data while retaining accuracy

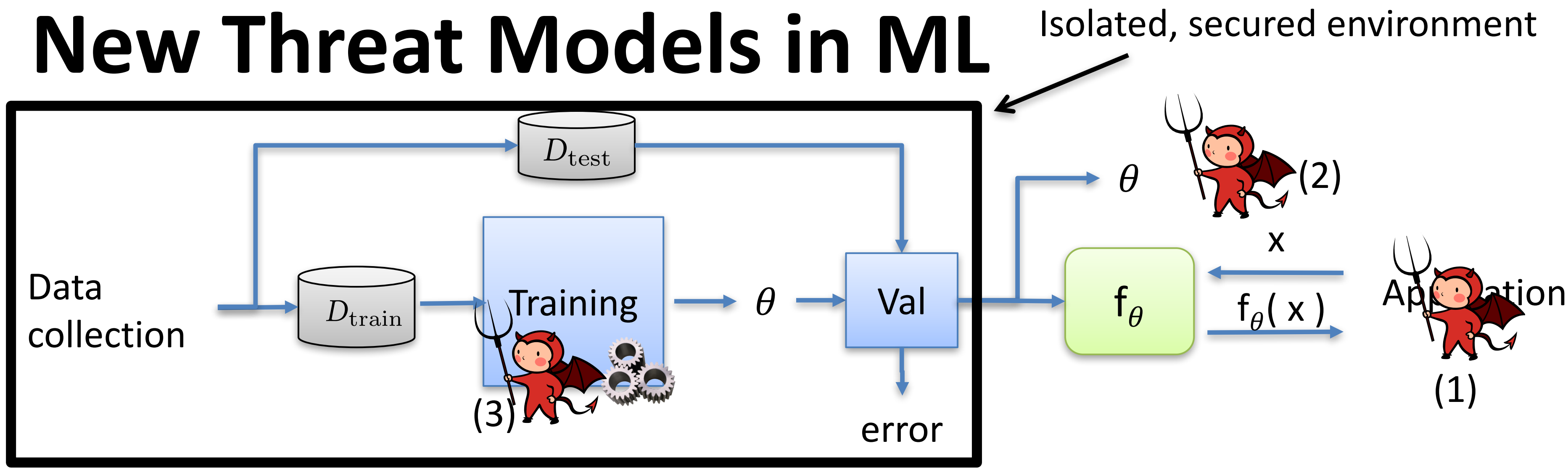
Isolated system is insufficient if training algorithm untrusted

What about detecting bad training by analyzing output θ?



For now: must validate training algorithms as legitimate

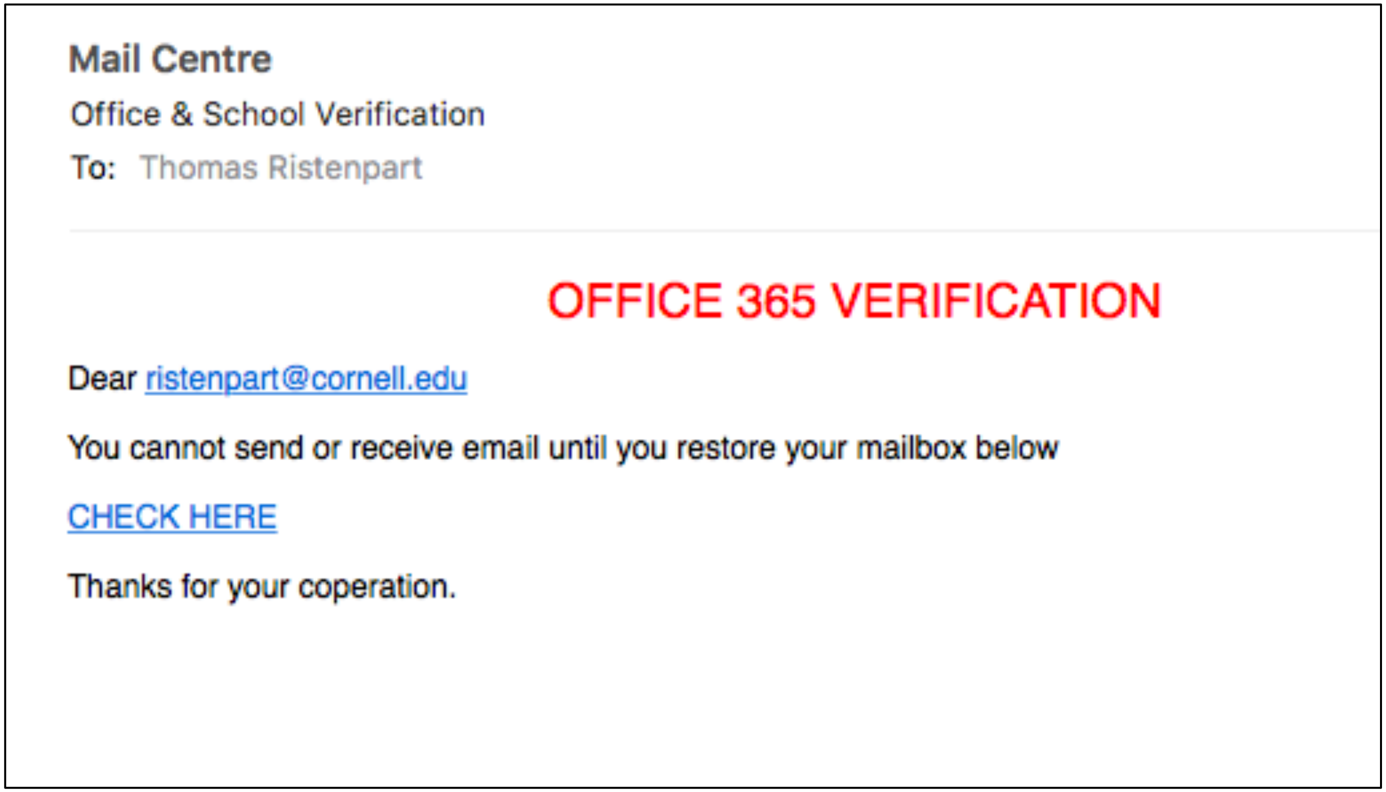
New Threat Models in ML



	Adversarial goal(s)	Adversarial abilities
(1) Model extraction	Learn θ	Query f_θ on adversarial points and see response
(2) Model inversion	Learn information about training data	Access to θ
(3) Malicious training	Exfiltrate training data through validated model θ	Specify Training algorithm, access to f_θ

Evasion attacks against ML

Given: θ , x , target prediction y’
Find: x’ such that $f_\theta(x') = y'$ and x,x’ are “similar”



[Graham-Cumming 2004] [Lowd, Meek 2005]

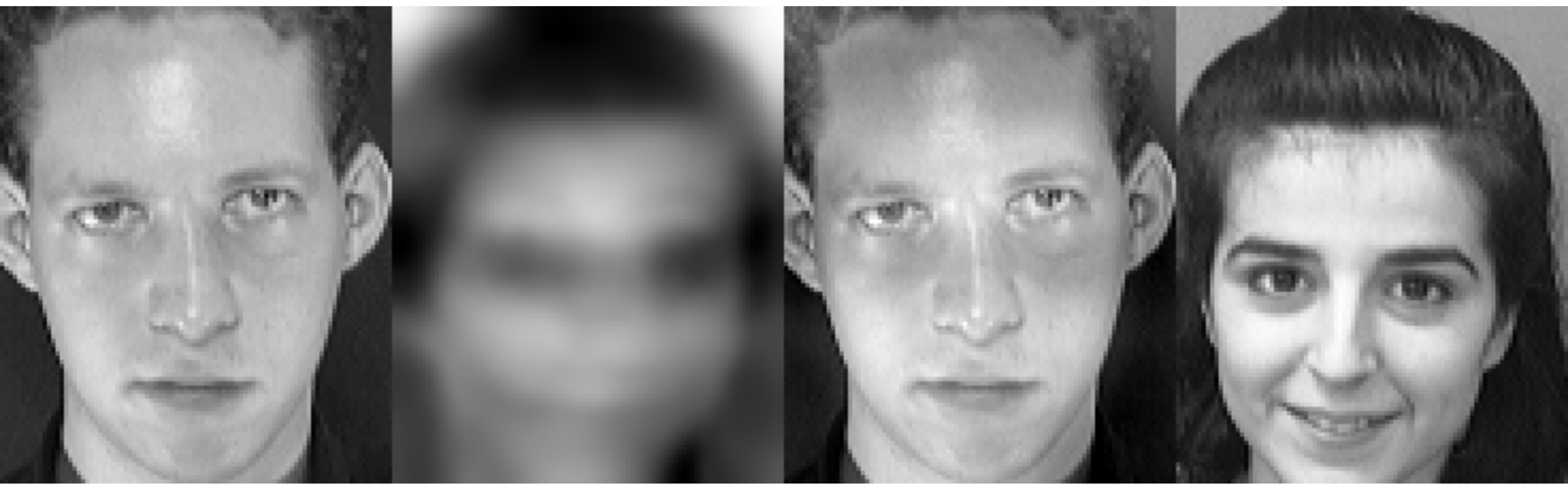
Evasion attacks against ML

Given: θ , x , target prediction y’

Find: x’ such that $f_\theta(x') = y'$ and x,x’ are “similar”

Another example: facial recognition model f_θ

Minimize $\|\Delta\|_2$ such that $f_\theta(x+\Delta) = y'$ [Szegedy et al. 2014]



x
$f_\theta(x)$ = Bob

Δ

x’
$f_\theta(x')$ = Alice

Images courtesy
of Matt Fredrikson

Evasion attacks against ML

Given: θ , x , target prediction y’

Find: x’ such that $f_{\theta}(x') = y'$ and x,x’ are “similar”

Lots of work on evasion (also called adversarial examples):
See survey [Papernot et al. 2016]

Summary

Lots of exciting lines of work in ML security. Many open questions

	Adversarial goal(s)	Adversarial abilities
(1) Model extraction	Learn θ	Query f_θ on adversarial points and see response
(2) Model inversion	Learn information about training data	Access to θ
(3) Malicious training	Exfiltrate training data through validated model θ	Specify Training algorithm, access to f_θ
(4) Evasion attacks (adversarial examples)	$f_\theta(x)$ misclassifies x	Access to θ
(5) Membership inference	Detect if person contributed to training data (privacy)	Query f_θ on adversarial points and see response

Citations

[Ateniese et al. 2015] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. *Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifier*s. Int. J. Secur. Netw.

[Fredrikson, et al. 2014] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. *Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing.* USENIX Security

[Fredrikson, Jha, R. 2015] Matthew Fredrikson, Somesh Jha, and Thomas Ristenpart. *Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures.* CCS

[Graham-Cumming 2004] John Graham-Cumming. 2004. *How to beat an adaptive spam filter.* MIT Spam Conference.

[Papernot et al. 2016] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. *Towards the Science of Security and Privacy in Machine Learning.* arXiv

[Shokri et al. 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. *Membership Inference Attacks against Machine Learning Models*. IEEE S&P

[Song, R., Shmatikov 2017] Congzheng Song, Thomas Ristenpart, Vitaly Shmatikov. *Machine Learning Models that Remember Too Much.* In preparation

[Szegedy et al. 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus. *Intriguing properties of neural networks*. ICLR

[Tromer, Zhang, Juels, Reiter, R. 2016] Florian Tramer, Fan Zhang, Ari Juels, Michael Reiter, and Thomas Ristenpart. *Stealing Machine Learning Models via Prediction APIs.* USENIX Security

[Zhang et al. 2017] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. *Understanding deep learning requires rethinking generalization.* ICLR