NORTH CAROLINA AGRICULTURAL AND TECHNICAL STATE UNIVERSITY

Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures

Matt Fredrikson — *Carnegie Mellon University*
Somesh Jha — *University of Wisconsin–Madison*
Thomas Ristenpart — *Cornell Tech*

(***Published in:*** *CCS '15: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, *October 2015*, *https://doi.org/10.1145/2810103.2813677*)

By:
Biruk E. Tegicho

Instructor :
Dr. Mahmoud N. Mahmoud

April 02, 2020

AGGIES DO

ncat.edu

Contents

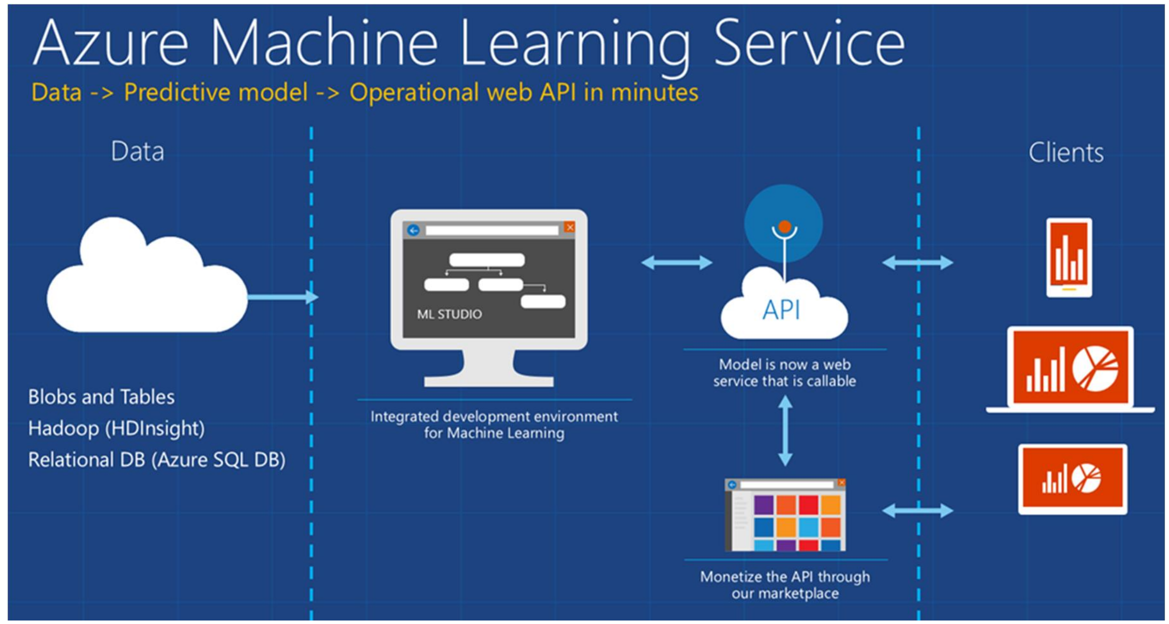
- Introduction
  - ML APIs
  - Threat Model
- Background
- Literature Survey
  - The Fredrikson et. al. Attack
- Case i: MAP Inverters
  - The inversion problem
  - Blackbox MI
  - Whitebox MI
- Case ii: Facial Recognition Model Inversion
  - Facial recognition Models MI Attack
  - Algorithms used
  - Accuracy results
  - Reconstruction results

- Countermeasures

Introduction(1/3)

- Machine-learning (ML) algorithms are increasingly utilized in privacy-sensitive applications like,
  - *predicting lifestyle choices,*
  - *making medical diagnoses,*
  - *facial recognition.*
- The need for easy ***"push-button" ML*** has prompted a number of companies to build ML-as-a-service(MLaas) cloud systems.
- Systems that incorporate the models will do so via well-defined ***application-programming interfaces(APIs).***
- Some of these API services have marketplaces within which users can ***make models or data sets available to other users.***

Introduction(2/3)



The model can be

- ***White-box:*** *Anyone can download a description of a model f suitable to run it locally.*
- ***Black-box:*** *One can't download the model but can only make prediction queries against it.*

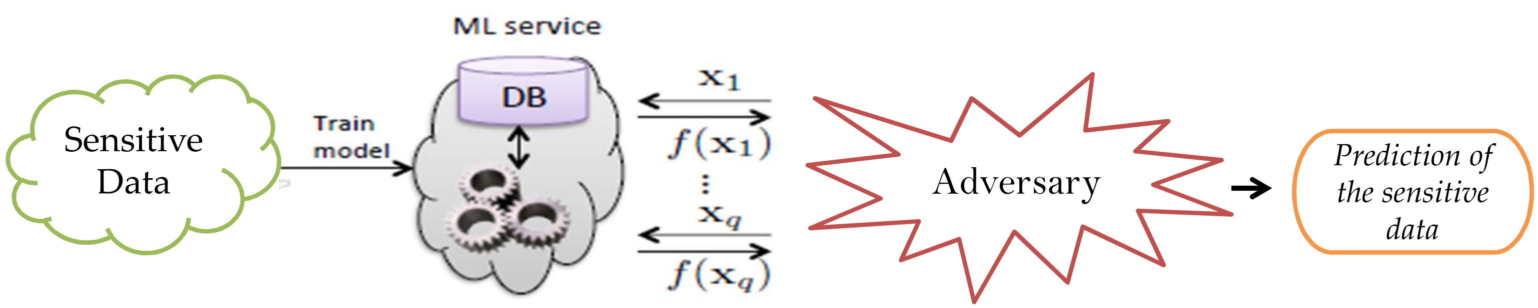
[1]. https://thebrainfiles.wearebrain.com/machine-learning-as-a-service-what-is-it-and-how-can-it-help-your-business-3310ac4f0b25
[2]. https://1reddrop.com/2019/02/09/azure-ml-explained-azure-machine-learning-service-and-azure-machine-learning-studio/

Threat Model

- *A clear threat is that providers might be poor stewards, allowing query logs to fall prey to insider attacks via system compromises.*

Assumptions

- The adversary has whatever information the API exposes.
- ***It does not have access to the training data***
- It also obtains the ***auxiliary information (aux) output by training.***



Background

Background

$$f = \mathbb{R}^d \longrightarrow Y \quad \dots\dots..(1)$$

$$(x, y) \in \mathbb{R}^d \times Y \quad \dots\dots\dots(2)$$

$$x = x_1, \dots., x_d \quad \dots\dots(3)$$

- **ML model:** ***Deterministic function f*** $= \mathbb{R}^{\boldsymbol{d}} \longrightarrow \boldsymbol{Y}$ from d features to a set of responses Y.
- **Input data:** 'db', a sequence of $(d + 1)$ dimensional vectors $(x, y) \epsilon\ \mathbb{R}^d \times Y$,
  - where $x = x_1; \dots\dots; x_d$ is the set of features
  - y is the label.
- **Output**: f and auxiliary information aux.
- Examples of auxiliary information might include ***error statistics and/or marginal priors for the training data***.
- *In regression, these outputs are called confidences*
  - *The classification is obtained by choosing the class label for the regression with highest confidence.*

Background

$\tilde{f} = \mathbb{R}^d \longrightarrow [0,1]^m$ .... (4)

t: $[0,1]^m \longrightarrow Y$ .... (5)

$f(x) = t\left(\tilde{f}(x)\right)$ … … . . (6)

- In these cases f is defined as the composition of two functions.
  - The first is a function $\tilde{\boldsymbol{f}} = \mathbb{R}^{\boldsymbol{d}} \longrightarrow [\mathbf{0}, \mathbf{1}]^{\boldsymbol{m}}$
    - ***m is a parameter specifying the number of confidences.***
  - The second function is a selection function $\boldsymbol{t}$: $[\mathbf{0}, \mathbf{1}]^{\boldsymbol{m}} \longrightarrow \boldsymbol{Y}$
- Ultimately, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{t}(\tilde{\boldsymbol{f}}(\boldsymbol{x}))$
- It is common among APIs for such models that classification queries return both ***f as well as*** $\tilde{\boldsymbol{f}}$

North Carolina Agricultural and Technical State University

Literature Review

Literature Review(1/3)

The Fredrikson et. al. Attack

- Considered ***a linear regression model f*** that predicted a real-valued suggesting initial dose of the drug Warfarin
- Used a feature vector consisting of ***patient demographic information, medical history, and genetic markers.***
- The ***sensitive attribute*** was considered to be the ***genetic marker***, which is assumed for simplicity to be the first feature x_1.
- Explored model inversion attack
  - Given white-box access to f and auxiliary information $\boldsymbol{side(x, y)} \overset{\text{def}}{=} (\boldsymbol{x_2}, \ldots, \boldsymbol{x_t}, \boldsymbol{y})$ ***for a patient instance*** $(\boldsymbol{x}, \boldsymbol{y})$,
  - An attacker attempts to infer the *patient's genetic marker* $\boldsymbol{x_1}$.

[3]. M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In USENIX Security Symposium, pages 17–32, 2014.

The Fredrikson et. al. Attack

adversary $\mathcal{A}^f(err, \boldsymbol{p}_i, x_2, \ldots, x_t, y)$:

1: **for** each possible value v of x_1 **do**
2: $\boldsymbol{x}' = (v, x_2, \ldots, x_t)$
3: $\boldsymbol{r_v} \leftarrow err(y, f(x')) \,.\, \Pi_i p_i(x_i)$
4: **Return** arg $\max_v$ $\mathrm{r}_{\boldsymbol{v}}$

- Here ***aux*** is assumed to give
  - Empirically computed ***standard deviation for a Gaussian error*** model ***err***
  - Marginal priors

$$\boldsymbol{p} = (\boldsymbol{p_1}; \ldots \ldots \ldots \ldots \ldots ; \boldsymbol{p_t})$$

- The marginal prior p_i is computed by first partitioning the real line into disjoint buckets (ranges of values),

$$p_i(v) = \frac{\textit{number of times } x_i \textit{ falls in } v \textit{ over all } x \textit{ in 'db'}}{\textit{number of training vectors } |db|}$$

- The algorithm simply completes the target feature vector with each of the ***possible values for*** $\boldsymbol{x_i}$.
- Then computes a ***weighted probability estimate*** that this is the correct value.

[3]. M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In USENIX Security Symposium, pages 17–32, 2014.

The Fredrikson et. al. Attack (Drawbacks)

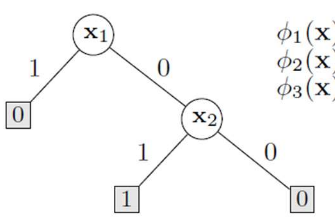
- It cannot be used when the unknown features cover an ***intractably large set.***
- Even if one only wanted to infer a portion of the features this is ***computationally infeasible.***
- It is potentially applicable in other settings, where f is not a linear regression model but some other algorithm.

North Carolina Agricultural and Technical State University

MAP Inverter for Trees

AGGIES DO

Map Inverters For Trees



$$\phi_1(\mathrm{x}) = \mathrm{x}_1 \qquad w_1 = 0$$
$$\phi_2(\mathrm{x}) = (1 - \mathrm{x}_1)(\mathrm{x}_2) \qquad w_2 = 1$$
$$\phi_3(\mathrm{x}) = (1 - \mathrm{x}_1)(1 - \mathrm{x}_2) \qquad w_3 = 0$$

Decision tree for the formula $y = \neg \mathrm{x}_1 \wedge \mathrm{x}_2$

- ***A decision tree model recursively partitions the feature space into disjoint regions R_1; ……; R_m.***
- Predictions are made for an instance (x, y)
- Trees are mathematically characterized as
  $\boldsymbol{f(x) = \sum_{i=1}^{m} w_i \, \phi_i(x)}$, ***where*** $\boldsymbol{\phi_i(x) \in \{0, 1\}}$
  - Each basis function $\phi_i(x)$ is an *indicator* for R_i, and
  - w_i corresponds to the *most common response* observed in the training set within R_i.
- The classification and corresponding confidences are given by:

$$f(\mathbf{x}) = \arg\max_j \left(\sum_{i=1}^{m} w_i[j]\phi_i(\mathbf{x})\right), \text{ and}$$
$$\tilde{f}(\mathbf{x}) = \left[\frac{w_{i^*}[1]}{\sum_i w_1[i]}, \ldots, \frac{w_{i^*}[|Y|]}{\sum_i w_m[i]}\right]$$

where i^* in the second formula takes the value in $\{1, \ldots, m\}$

Black-box MI

- *Confusion matrix C is used and* $err(y, y') \propto Pr[f(x) = y' \mid y \text{ is the true label.}]$ *is defined*

White-box MI

- The attacker knows each ϕ_i, n_i that correspond to ϕ_i and $N = \sum_{i=1}^{m} n_i$, the total number of samples in the training set.

$$\phi_i(v) = \mathbb{I}(\exists x' \epsilon\, \mathbb{R}^d . x'_1 = v \wedge \phi_i(X')).$$

- $\boldsymbol{p_i}$ denote $\boldsymbol{n_i / N}$, and each p_i gives us some information about the ***joint distribution*** on features used to build the training set.

- The known values x_K induce a set of paths

$$S = \{s_i\}_{1 \le i \le m} : S = \{(\phi_i, n_i) \mid \exists x' \in \mathbb{R}^d . x'_K = x_K \wedge \phi_i(X')\}.$$

Case i(3/5)

White-box MI (white-box with counts (WBWC) estimator)

- The following estimator characterizes ***probability that $x_1 = v$*** given **x traverses** one of the paths $s_1, \ldots, s_m$ and $x_K = v_K$:

$$\Pr[\,\mathbf{x}_1 = v \mid (s_1 \vee \cdots \vee s_m) \wedge \mathbf{x}_K = \mathbf{v}_K\,]$$

$$\propto \frac{1}{\sum_{j=1}^{m} p_j \phi_j(v)} \sum_{1 \le i \le m} p_i \phi_i(v) \cdot \Pr[\,\mathbf{x}_1 = v\,]$$

- ***The adversary then outputs a value for v that maximizes the above equation as a guess for x_1.***
- Like the Fredrikson et al. estimator, ***it returns the MAP prediction given the additional count information.***
- It is assumed that the attacker knew all of x except x_1.

Experimental setup

- **BigML: REST API**
  - JSON downloaded for white box mode
- **FiveThirtyEight's "How Americans Like Their Steak" survey**
  - A survey, of ***553 individuals*** from SurveyMonkey, collected responses to questions such as:
    - *"Do you ever smoke cigarettes?"*
    - *"Have you ever cheated on your significant other?", and*
    - *"How do you like your steak prepared?".*
  - Demographic characteristics such as *age, gender, household income, education, and census region were also collected.*
- **Subset of the General Social Survey (GSS) focusing on responses related to marital happiness**
  - ***51,020 individuals and 11 variables***, including basic demographic information and responses to questions such as,
    - *"How happy are you in your marriage?"*
- ***Trained trees locally*** by constructing ***100 trees*** using default parameters on randomly-sampled stratified training sets comprised of ***50%*** of the available data.
- Machine with 8 Xeon cores running at 2.5 Ghz, with 16G of memory were used

THE GENERAL SOCIAL SURVEY GSS

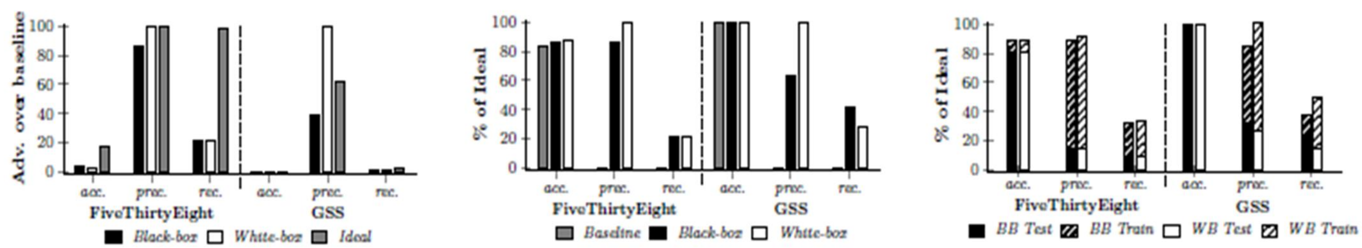
FiveThirtyEight

[4]. https://bigml.com/api/models#md_random_decision_forests

Results

	FiveThirtyEight			*GSS*		
ALGORITHM	**Acc.**	**Prec.**	**Rec.**	**Acc.**	**Prec.**	**Rec.**
Whitebox	***86.4***	***100.0***	***21.1***	***80.3***	***100.0***	***0.7***
Blackbox	***85.8***	***85.7***	***21.1***	***80.0***	***38.8***	***1.0***
Random	50.0	50.0	50.0	50.0	50.0	50.0
Baseline	82.9	0.0	0.0	82.0	0.0	0.0
Ideal	99.8	100.0	98.6	80.3	61.5	2.3

MI results for BigML models (All numbers shown are percentages)



(a) Results as advantage over baseline. (b) Results as a percentage of ideal. (c) Training vs. test attack performance.

BigML model inversion comparison to the baseline and ideal prediction strategies.

Facial Recognition Model Inversion

Case ii(1/10)

Lambda

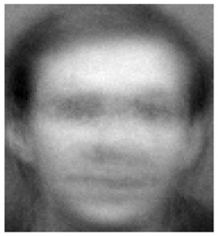
KAIROS

Facial recognition Models

facebook deepface

- Facial recognition models are functions that label an image containing a face with an identifier corresponding to the individual depicted in the image.
- A growing number of web APIs support facial recognition
- Common to all these APIs is the ability to
  - ***Train a model*** using a set of images labeled with the names of individuals that appear in them
  - ***Perform classification*** given some previously trained model.

[5] DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In Conference on Computer Vision and Pattern Recognition (CVPR).
[6]. https://developer.kairos.com/docs
[7]. https://lambdal.com/face-recognition-api.



Which one do you think is the true image?

1

2

Facial recognition Models MI Attack

First kind of attack

- an adversary who knows a label produced by the model, ***i.e. a person's name or unique identifier***
- *wishes to produce an image of the person associated with the victim.*
- This attack violates the privacy of an individual who is willing to provide images of themselves as training data

The adversary "wins" an instance of this attack if

- *when shown a set of face images including the victim, he can identify the victim.*

Facial recognition Models MI Attack

Second kind of attack

- *an adversary who has an image containing a blurred-out face, and wishes to learn the identity of the corresponding individual.*

- The adversary uses the blurred image as side information in a series of MI attacks,
  - The output of which is a ***deblurred image*** of the subject.

- Assuming the original image was blurred to protect anonymity, this attack violates the privacy of the person in the image.

The adversary wins if

- *She/he identifies the victim from a set of face images taken from the training set*

- *The adversary determines that the image produced by the attack does not correspond to any of the faces.*

Experimental setup

Models Used: Neural Network

- ***Softmax regression***
- ***Multilayer perceptron***: one hidden layer of 3000 sigmoid neurons and a softmax output layer.
- ***Stacked denoising autoencoder network***: two hidden layers, which have 1000 and 300 sigmoid units, and a softmax output layer.

Dataset: AT&T Laboratories Cambridge database of faces

- **10 black-and-white** images of *40 individuals* in various lighting conditions, facial expressions, and details for a total of ***400 images***.
- ***Images of each person***
  - Divided into training set (7) and a validation set (3)
  - Trained each model using pylearn2's stochastic gradient descent algorithm until the model's performance on the training set failed to improve after ***100 iterations.***

[8]. AT&T Laboratories Cambridge. The ORL database of faces. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[9]. I. J. Goodfellow, D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio. Pylearn2: a machine learning research library. arXiv preprint arXiv:1308.4214, 2013.

Case ii(5/10)

Libraries

```
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data
import numpy as np

import os
import sys
from PIL import Image
from sys import stdout

import scipy
import scipy.misc

from pylearn2.datasets.preprocessing import ZCA
from pylearn2.expr.preprocessing import global_contrast_normalize

import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from IPython import display
```

https://github.com/lisa-lab/pylearn2

Basic MI attack algorithms

Algorithm 1 Inversion attack for facial recognition models.

function MI-FACE($label, \alpha, \beta, \gamma, \lambda$)
  $c(\mathbf{x}) \stackrel{\text{def}}{=} 1 - \tilde{f}_{label}(\mathbf{x}) + \text{AUXTERM}(\mathbf{x})$
  $\mathbf{x}_0 \leftarrow \mathbf{0}$
  for $i \leftarrow 1 \ldots \alpha$ **do**
    $\mathbf{x}_i \leftarrow \text{PROCESS}(\mathbf{x}_{i-1} - \lambda \cdot \nabla c(\mathbf{x}_{i-1}))$
    if $c(\mathbf{x}_i) \geq \max(c(\mathbf{x}_{i-1}), \ldots, c(\mathbf{x}_{i-\beta}))$ **then**
      break
    if $c(\mathbf{x}_i) \leq \gamma$ **then**
      break
  return $[\arg\min_{\mathbf{x}_i}(c(\mathbf{x}_i)), \min_{\mathbf{x}_i}(c(\mathbf{x}_i))]$

- The attacker has no auxiliary information aside from the target label, so ***AuxTerm(x)* = *0*** for all x.
- The experiments set the parameters for MI-Face to: ***α = 5000; β = 100; γ=0.99, and λ= 0.1.***
- In all cases except for the stacked DAE network, the *process is set to be the identity function.*
- *For stacked DAE network, the function Process-DAE in Algorithm 2 is used.*

Algorithm 2 Processing function for stacked DAE.

function PROCESS-DAE($\mathbf{x}$)
  $encoder.\text{DECODE}(\mathbf{x})$
  $\mathbf{x} \leftarrow \text{NLMEANSDENOISE}(\mathbf{x})$
  $\mathbf{x} \leftarrow \text{SHARPEN}(\mathbf{x})$
  return $encoder.\text{ENCODE}(vecx)$

Case ii(8/10)

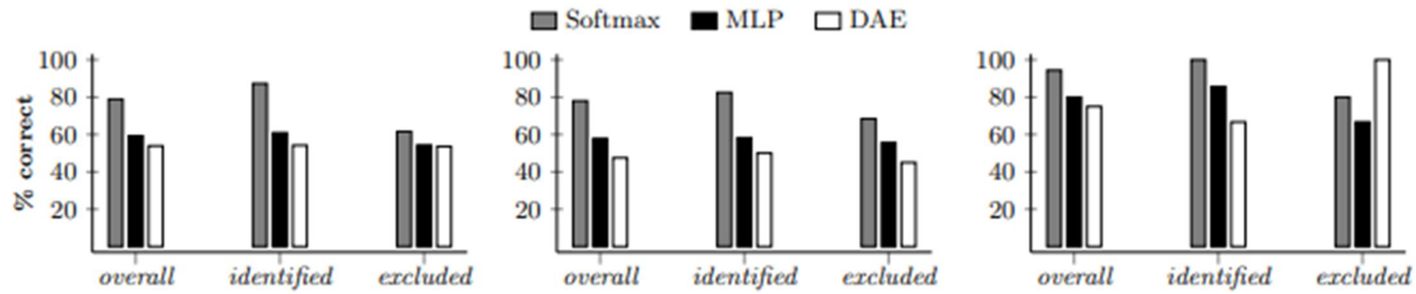
```
    def invert(self, sess, num_iters, lam, img, pre_process, pred_cutoff= 0.99, disp_freq=1):
        probs = self.preds(img)
        class_ind = sess.run(self.class_inds, feed_dict= {x:[img]})[0]
        current_X = np.zeros(list(img.shape)[0]).astype(np.float32)
        Y = (one_hot_preds(probs)).astype(np.float32)
        best_X = np.copy(current_X)
        best_loss = 100000.0
        prev_losses = [100000.0]*100
        for i in range(num_iters):
            feed_dict = {x: [current_X], y_: Y }
            der,current_loss = sess.run([self.grads, self.loss], feed_dict)
            current_X = np.clip(current_X - lam*(der[0][0]),0.0,1.0)
            current_X = normalize(current_X, pre_process, current_X.shape)
            probs = self.preds(current_X)[0]

            if current_loss < best_loss:
                best_loss = current_loss
                best_X = current_X
            if current_loss > 2*max(prev_losses):
                print("\n Breaking due to gradient chaos!!")
                break
            if pred_cutoff < probs[class_ind]:
                print("\n Above Probability Criteria!: {0}".format(probs[class_ind]))
                break
            if i%disp_freq ==0:
#                 plt.close()
#                 face_imshow(post_process(current_X, pre_process, current_X.shape))
#                 plt.show()
                stdout.write("\r Acc: %f and Loss: %f and Best Loss: %f" % (probs[class_ind], current_loss, best_loss))
                stdout.flush()

        stdout.write("\n")
        print('Loop Escape.')

        current_preds = self.preds(current_X)
        best_preds = self.preds(best_X)
        current_X = post_process(current_X, pre_process, current_X.shape)
        best_X = post_process(best_X, pre_process, best_X.shape)
        return current_X, current_preds, best_X, best_preds
```

Face-Rec Experiment

- To evaluate the effectiveness of the attack, it is ***ran on each of the 40 labels in the AT&T Face Database***
- Then ***Mechanical Turk workers*** were asked to match the reconstructed image to one of five face images
- *Each* ***batch*** *of experiments was run* ***three times,*** *with the same test images shown to workers in each run.*
- *In 80% of the experiments, one of the five images contained the individual corresponding to the label used in the attack.*
- An 8- core Xeon machine with 16G memory used

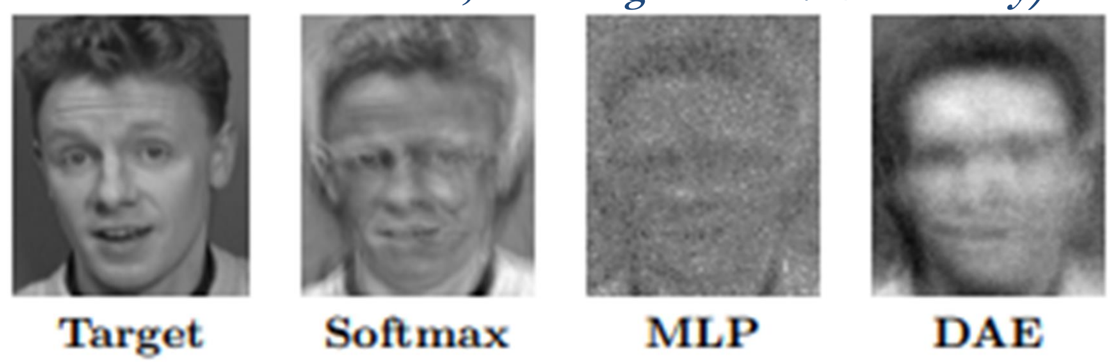
https://www.mturk.com/

Reconstruction Results



(a) Average over all responses. (b) Correct by majority vote. (c) Accuracy with skilled workers.

Reconstruction attack results from Mechanical Turk surveys (Skilled workers" are those who completed at least five MTurk tasks, achieving at least 75% accuracy)

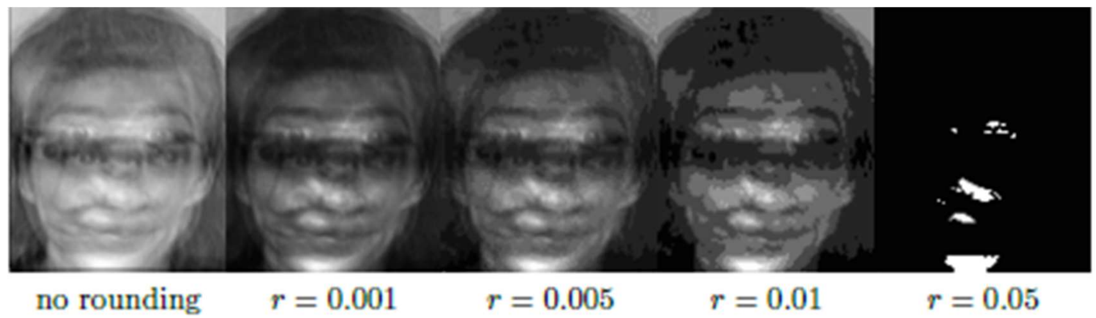


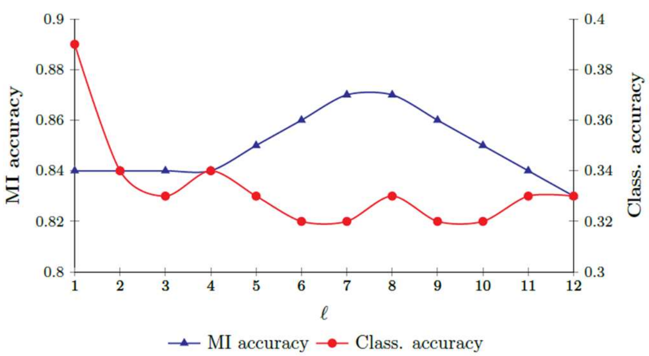
Reconstruction of the individual on the left by Softmax, MLP, and DAE.

Countermeasures

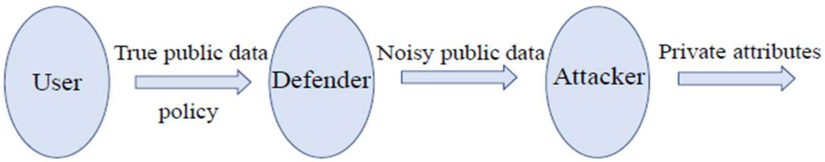
Countermeasures

- **Rounding reported confidence values can drastically reduce the effectiveness of the attacks.**
    - One possible defense is to *degrade the quality or precision of the gradient* information retrievable from the model.
    - For rounding level, r = {0.001, 0.005, 0.01, 0.05}





- **Taking sensitive features into account while using training decision trees**
    - When the feature appears near the top or bottom of the tree, the attack fails with greater probability than otherwise.
    - When the feature is placed at the top of the tree, classification accuracy is maximized while inversion accuracy is only 1% greater than baseline guessing.



Countermeasures (AttriGuard)

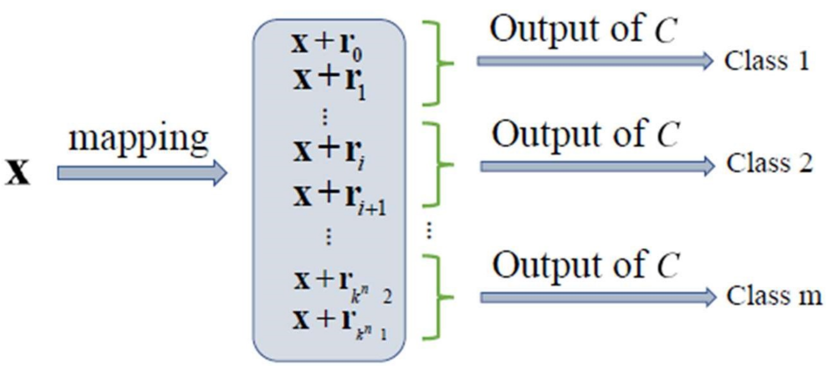
Policy A: Modify_Exist
Policy B: Add_New
Policy C: Modify_Add

- **Input:**
  - Noise-type-policy
  - Target probability distribution
  - Defender's classifier
  - User's true public data
- **Output:** Mechanism M that adds random noise
  - $M^*(r||x)$ is the conditional probability that defender will add noise **r** to user's true public data ***X***
  - Sample from M to add noise

https://github.com/m2kar/AttriGuard_Report

Jia, J., & Gong, N. Z. (2020). Defending against Machine Learning based Inference Attacks via Adversarial Examples: Opportunities and Challenges. In Adaptive Autonomous Secure Cyber Systems (pp. 23-40). Springer, Cham.

Countermeasures (AttriGuard)



- **Phase I:** For each noise group, find a minimum noise as representative noise,
  - Find minimum noise $\mathbf{r_i}$ for each group such that defender's classifier outputs class i given noisy public data input

$$\mathbf{r}_i = \mathrm{argmin}_{\mathbf{r}} ||\mathbf{r}||_0$$
$$\text{subject to} \quad C(\mathbf{x} + \mathbf{r}) = i.$$

- **Phase II:** Simplify the mechanism M* to be a probability distribution over m representative noise

$$M^* = \arg\min_M KL(\mathbf{p} \,||\, M)$$

$$\textit{subject to} \quad \sum_{i=1}^{m} M_i \|\mathbf{r}_i\|_0 \leq \beta$$
$$M_i > 0, \forall i \in \{1, 2, \ldots, m\}$$
$$\sum_{i=1}^{m} M_i = 1$$

M is a probability distribution, and M_i denote the probability select noise $\mathbf{r_i}$

Countermeasures (Others)

- **Game-theoretic methods**
  - Pros: Defend against optimal inference attacks
  - Cons: Computationally intractable
- **Heuristic methods**
  - Pros: Computationally tractable
  - Cons:
    - Large utility loss
    - Direct access to user's private attribute value
- **Local Differential Privacy (LDP)**
  - Pros: Rigorous privacy guarantee
  - Cons: Large utility loss
- **Differential privacy**
  - It decrease the ability of an adversary A to learn information about training set elements, when given access to prediction queries.
- **Ensemble methods**
  - May be more resilient to extraction attacks, in the sense that attackers will only be able to obtain relatively coarse approximations of the target function.

Conclusion

- **Explored privacy issues in ML APIs**, showing that confidence information can be exploited by adversarial clients in order to mount model inversion attacks.
- **Provided model inversion algorithms** that can be used to
  - Infer sensitive features from decision trees hosted on ML services, or
  - Extract images of training subjects from facial recognition models.
- **Evaluated these attacks on real data**, and showed that
  - Models trained over datasets involving survey respondents pose significant risks to feature confidentiality, and
  - Recognizable images of people's faces can be extracted from facial recognition models.
- **Evaluated preliminary countermeasures** that mitigate the attacks we develop, and might help prevent future attacks.

Thank You!!