Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures

[M. Fredrikson, S. Jha, T. Ristenpart, CCS'15]

2023.11.22
20233456 Jio Oh
harryoh99@kaist.ac.kr

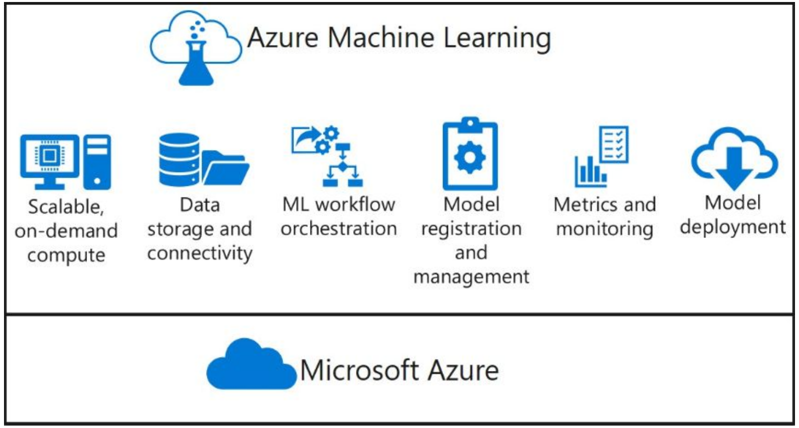
Introduction

- Develop **model inversion attack** exploiting **confidence values**
    - Revealing sensitive attributes / training data for MLaaS
- Evaluation of success of attack in two different settings

  : **decision trees** & **neural networks**

Machine Learning APIs

- Machine Learning as-a-service systems
  - Users upload training data & query the API to get the result
    - e.g.) BigML, GPT-3.5, GPT-4, Microsoft ML, Google Prediction API



bigml

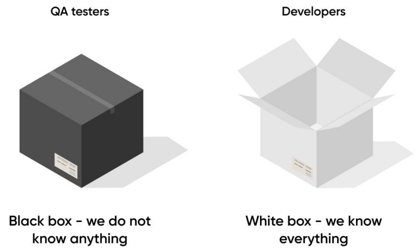
```python
import openai
openai.api_type = "azure"
openai.api_base = "https://jiojap.openai.azure.com/"
openai.api_version = "2023-05-15"
openai.api_key = #HIDDEN

msg = [
        {"role": "user", "content":"Complete the following sentence. Yongdae Kim, a professor in KAIST is ..."},
    ]
response = openai.ChatCompletion.create(
    engine="gpt4_jio",
    temperature = 0,
    messages = msg
)

print(response['choices'][0]['message']['content'])
```

✓ 1.4s

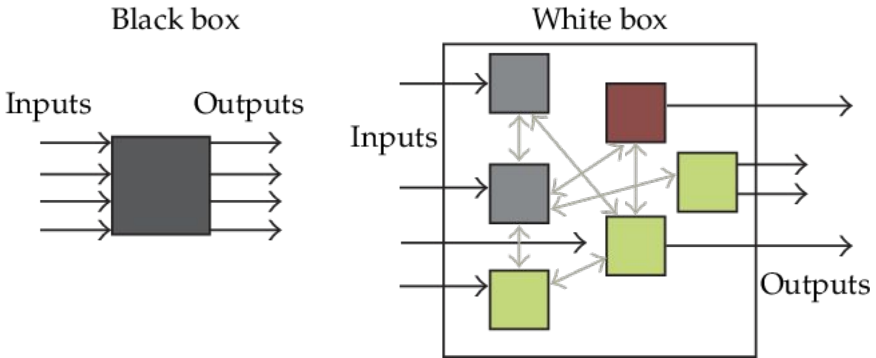
```
well-known for his research in the field of engineering.
```

Background

Black-Box/White-Box Models

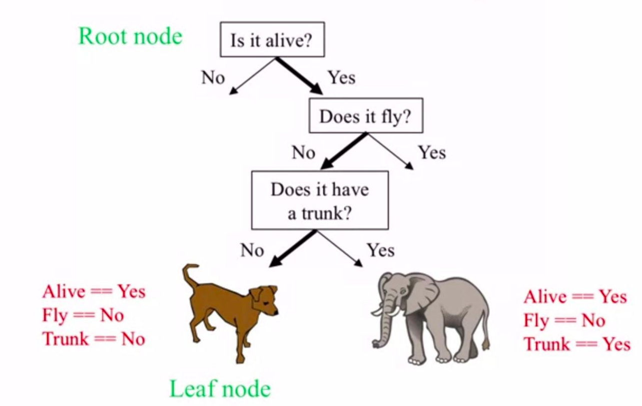


- Black-Box Model
  - Anyone can query the model to return the model’s output on given input
  - No access to training data
  - Most commercial APIs
- White-Box Model
  - Users have access to the structure/parameters of the model



Decision Tree

- Non-parametric supervised learning method for regression & classification
  - Learning from simple decision rules (boolean logic) inferred from data
- Simple to understand, low cost (logarithmic inference time), high reliability
- Prone to overfitting, highly sensitive generation, bad at extrapolation



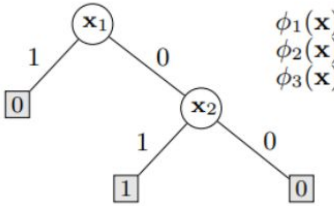
Decision Tree

Decision Tree Formulation

- Recursively partitions the feature space into disjoint regions $R_1, R_2, \ldots, R_m$.
- For (x;y), finds region containing x, and returns the most likely value for x within that region.
- Formulated as $f(\mathbf{x}) = \sum_{i=1}^{m} w_i \phi_i(\mathbf{x}),\ \text{where } \phi_i(\mathbf{x}) \in \{0, 1\}$

  w_i is the most common response label in training region R_i and Φ_i is an indicator for R_i.

Decision Tree Example



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

Figure 3: Decision tree for the formula $y = \neg\mathbf{x}_1 \wedge \mathbf{x}_2$.

x_1	x_2	$\neg x_1$	Y
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	0

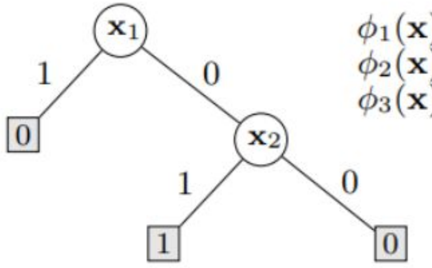
Background

Decision Tree

Decision Tree w/ Confidence Values

- Set the weights proportional to the number of corresponding values in the training set
  - 89 samples with (1, 1) & 11 with (1, 0) for (x_1, x_2)

    => w_1 = (89, 11)



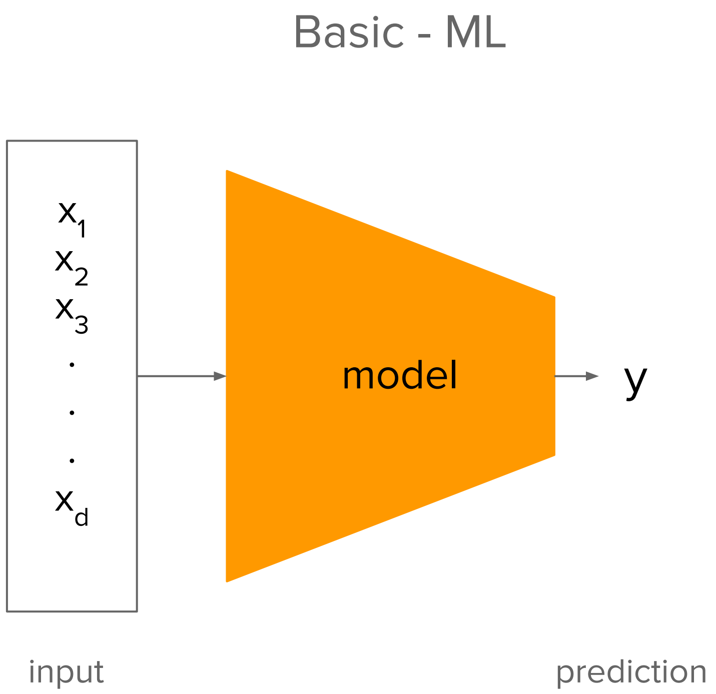
$$\phi_1(\mathbf{x}) = \mathbf{x}_1 \qquad w_1 = \text{(89, 11)}$$
$$\phi_2(\mathbf{x}) = (1-\mathbf{x}_1)(\mathbf{x}_2) \qquad w_2 = 1$$
$$\phi_3(\mathbf{x}) = (1-\mathbf{x}_1)(1-\mathbf{x}_2) \qquad w_3 = 0$$

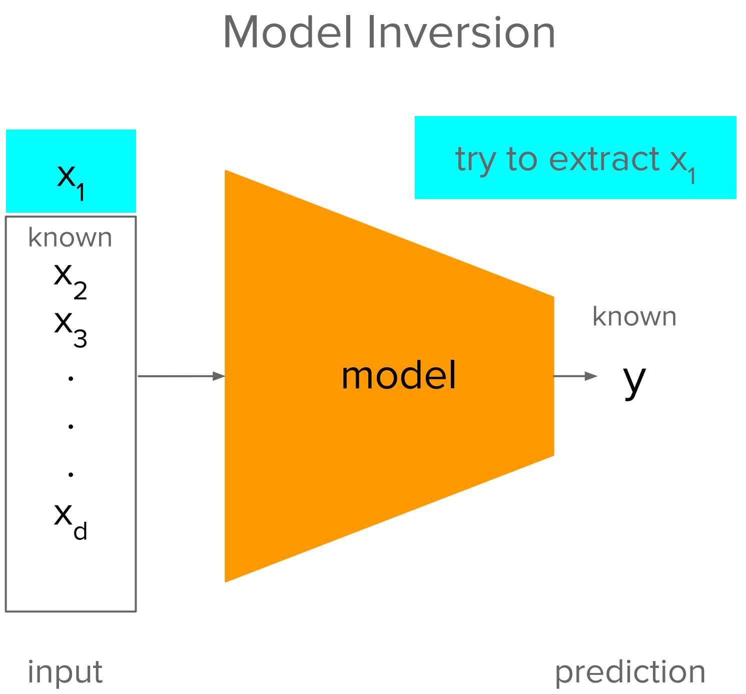
Figure 3: Decision tree for the formula $y = \neg\mathbf{x}_1 \wedge \mathbf{x}_2$.

- Classification Formula

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

Model Inversion





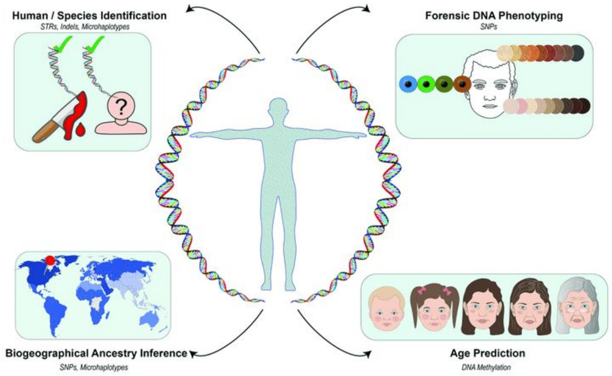
Fredrikson et al. Attack

Targeted Model

- **Linear regression model** targeted for prediction of real-valued suggested initial dose of Warfarin
- Features: patient demographic information, medical history, and ***genetic markers***

Purpose of Attack

- Given white-box access to f, attacker tries to **retrieve x_1 (genetic marker)** with auxiliary information defined as $side(x,y) = (x_2, \ldots, x_t, y)$ for patient instance, $(x_1, \ldots, x_t, y)$



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, 2014.

Fredrikson et al. Attack

Algorithm

- Completes target feature vector (x_1) with all possible values
- Computes weighted probability estimate with minimal error
  - Gaussian error model

```
adversary A^f(err, p_i, x_2, ..., x_t, y):
  for each possible value v of x_1 do
      x' = (v, x_2, ..., x_t)
      r_v ← err(y, f(x')) · ∏_i p_i(x_i)
  Return argmax_v r_v
```

adversary $\mathcal{A}^f(\mathsf{err}, \mathbf{p}_i, \mathbf{x}_2, \ldots, \mathbf{x}_t, y)$:

for each possible value v of $\mathbf{x}_1$ **do**

$\quad \mathbf{x}' = (v, \mathbf{x}_2, \ldots, \mathbf{x}_t)$

$\quad \mathbf{r}_v \leftarrow \mathsf{err}(y, f(\mathbf{x}')) \cdot \prod_i \mathbf{p}_i(\mathbf{x}_i)$

Return $\arg\max_v \mathbf{r}_v$

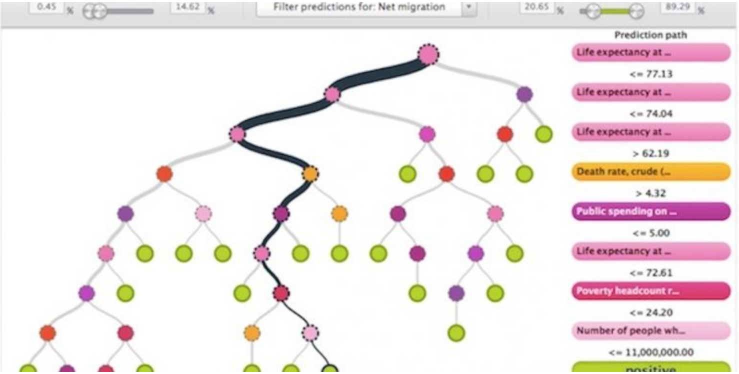
Figure 2: Generic inversion attack for nominal target features.

Limitations

- Cannot be used when unknown features cover an intractably large set
  - Facial recognition tasks have 10,000+ real number valued features

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, 2014.

MI for Decision Tree



Goal

- Predict the sensitive feature from output & side information
  - Black-box: w/ output confusion matrix
  - White-box: w/ # of training samples corresponding to Φ_i

Model API

- Trees that are published via **BigML API**

MI for Decision Tree

- Black-box Setting
  - Use same generic algorithm as Fredrickson et al. attack with modification on error function
  - err (y, y') ∝ Pr [f(x) = y' | y is the true label]

    $$\mathbf{r}_v \leftarrow \mathrm{err}(y, f(\mathbf{x}')) \cdot \prod_i \mathbf{p}_i(\mathbf{x}_i)$$

    - Utilizing confusion matrix, C
- White-box Setting
  - Attacker knows $p_i = n_i/N$
    - n_i= sample count in training set
  - maximize

$$\Pr[\mathbf{x}_1 = v | (s_1 \vee \cdots \vee s_m) \wedge \mathbf{x}_K = \mathbf{v}_K] \propto \frac{1}{\sum_{j=1}^{m} p_j \phi_j(v)} \sum_{1 \le i \le m} p_i \phi_i(v) \cdot \Pr[\mathbf{x}_1 = v]$$

MI for Decision Tree

Datasets

- FiveThirtyEight surveys
    - Statistical analysis on connection between steak preference & risk-taking behavior
    - Sensitive feature: answer to infidelity questions

- General Social Survey marital happiness survey
    - Information on demographics, interests, and attitudes of US residents
    - Sensitive feature: answer to questions related to watching X-rated movies

MI for Neural Networks

Reconstructed

Goal

- Reconstruction Attack: Find the original face of the victim from the NN
    - Adversary knows label produced by the model
        - Person’s name or unique identifier (SSN)
    - **Adversary wins** if when shown a set of images, one can **identify the victim**

Model

- Softmax Regression: 1-layer softmax
- MLP: 3000 hidden sigmoid unit + 1-layer softmax
- Denoising AutoEncoder (DAE): 2-hidden layers (1000,300 sigmoid units) + 1-layer softmax

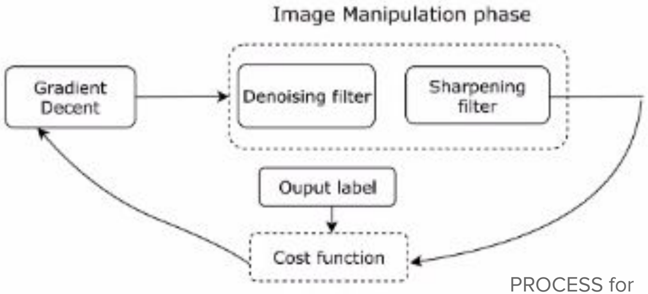
Methods

MI for Neural Networks

- c(x): cost function
- $\tilde{f}$(x): model
- AUXTERM: auxiliary function
  - Set as 0
- PROCESS
  - Identity function except for DAE
- λ: step size
- α: maximum iterations
- γ: cost threshold
- β: if cost fails to improve after β iterations halt

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 $[\arg\min_{\mathbf{x}_i}(c(\mathbf{x}_i)), \min_{\mathbf{x}_i}(c(\mathbf{x}_i))]$



PROCESS for DAE

Figure 7: Reconstruction without using Process-DAE (Algorithm 2) (left), with it (center), and the training set image (right).

MI for Neural Network

Datasets

- AT&T Laboratories Cambridge Database of faces
  - Black-and-white images of 40 individuals with various lighting condition, facial expression, and details

User Study

- Used Amazon Mechanical Turk s.t. Workers match the reconstructed image to one of the five images

amazon
mechanical turk

Experiments

Evaluations

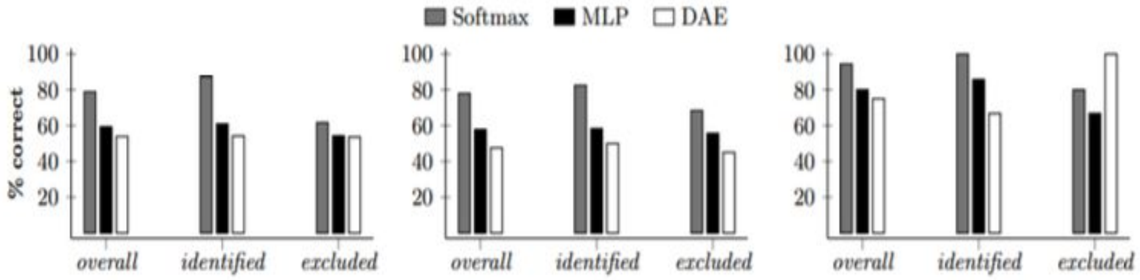
- MI for Decision Trees
  - White-box
  - Black-box
  - Random: coin-flipping
  - Baseline: always “no” (based on prior distribution)
  - Ideal: decision tree trained for predicting feature w/ full access to training set
- MI for NN
  - Overall: general accuracy
  - Identified: P(correct | target image exists among candidates)
  - Excluded: P(correct | target image not exists among candidates)

Evaluations

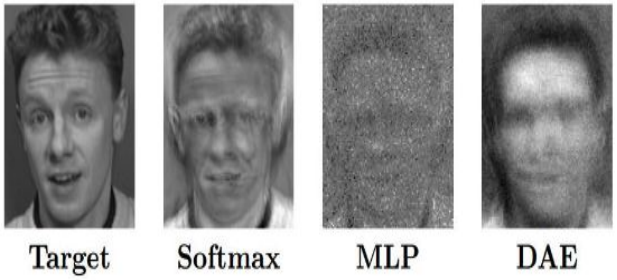
- MI for Decision Trees

	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 for NN

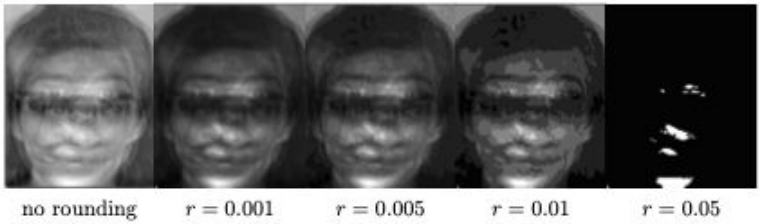


(a) Average over all responses. (b) Correct by majority vote. (c) Accuracy with skilled workers.



Countermeasures

- MI for Decision Trees
  - Change the order where sensitive feature appears on tree
    - Attack accuracy degraded when target feature placed **on top** or bottom of tree
    - Decision tree nature (splitting decisions)
- MI for NN
  - Gradient obfuscation
    - Rounding score of softmax result



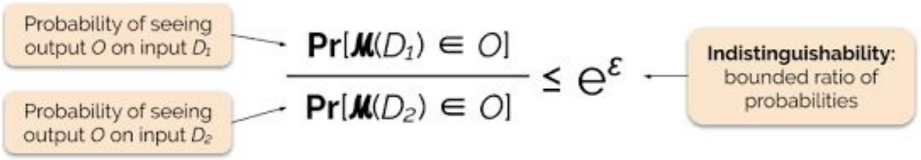
Future Works: Attacks

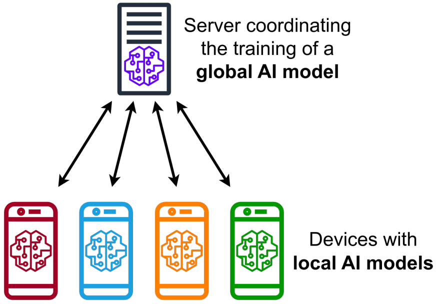
MI attack attempts towards **larger models** & **black-box setting**

- Computer Vision
  - Machine Learning Models that Remember Too Much [CCS'17]
  - Model Inversion Attacks for Prediction Systems: Without Knowledge of Non-sensitive Attributes [PST'17]
  - Neural Network Inversion in Adversarial Setting via Background Knowledge Alignment [CCS'19]
  - Generative Model-Inversion Attacks Against Deep Neural Networks [CVPR'20]
  - Variational Model Inversion Attacks [NeurIPS'21]
- Natural Language Processing
  - Extracting Training Data from Large Language Models [USENIX'20]
  - Information Leakage in Embedding Models [CCS'20]
  - Canary Extraction in Natural Language Understanding Models [ACL'22]
  - Are Large Pre-Trained Language Models Leaking Your Personal Information? [NAACL'22]

Future Works: Defense

- Differential Privacy
  - Regression Model Fitting Under Differential Privacy and Model Inversion Attack [IJCAI'15]
  - Broadening Differential Privacy for Deep Learning Against Model Inversion Attacks [Big Data'20]
  - One Parameter Defense-Defending Against Data Inference Attacks via Differential Privacy [TIFS'22]
- Adversarial Defense, Federated Learning
  - Privacy Preserving Facial Recognition Against Model Inversion Attacks [Globecom'20]
  - NetFense: Adversarial Defense against Privacy Attacks on Neural Networks for Graph Data [ICDE'21]



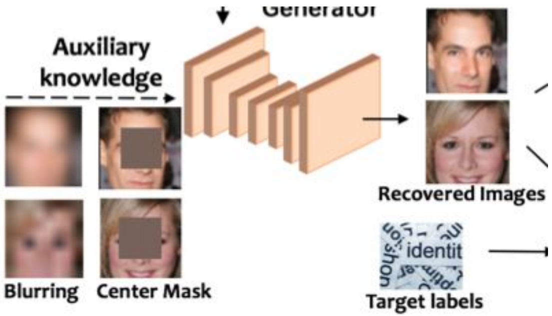


Conclusion & Discussion

- Model Inversion attack against Decision Tree & Neural Network
- Simple countermeasures for the two settings

Limitations

- My opinion: Unrealistic Setting & Attack
  - Just saying “no” => accuracy >80%
  - Hard to access side information for DT
    - 23% success w/o any side information
  - Too small dataset (40) for NN
    - Too easy to distinguish
    - Quantitative values should have been presented
  - Extensions to larger models -> hard
- Can hiding confidence value mitigate these kinds of attacks?



Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. In CVPR’20

Good Questions

- Zhixian Jin: Is it possible for the cloud vendor to detect the model inversion attacks?

Best Questions

- 윤태웅: Can the optimal placement of sensitive features in decision trees be determined dynamically during training, considering the evolving state of the tree?

- 오성룡: In your experiments, you used decision trees as the machine learning model. Have you explored the effectiveness of model inversion attacks on other types of models, such as neural networks or support vector machines? How do these models compare in terms of vulnerability to model inversion attacks?

- 이승현: Is confidence score rounding always a valid defense against gradient descent based attacks? Would the precision levels sufficient to prevent model inversion attacks be generally similar even with different models?

Thank You