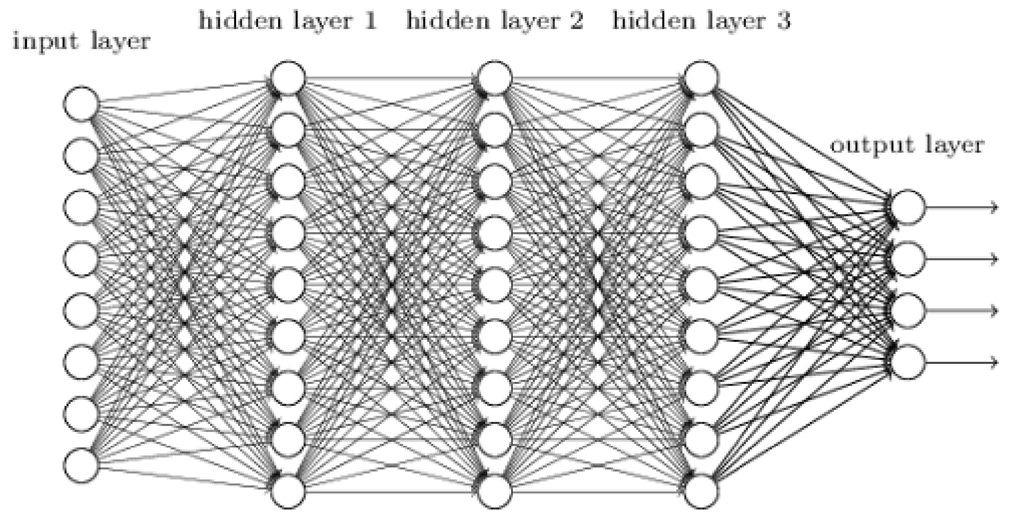
Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures

ACM CCS'15

SysSec Changhun Song

Introduction

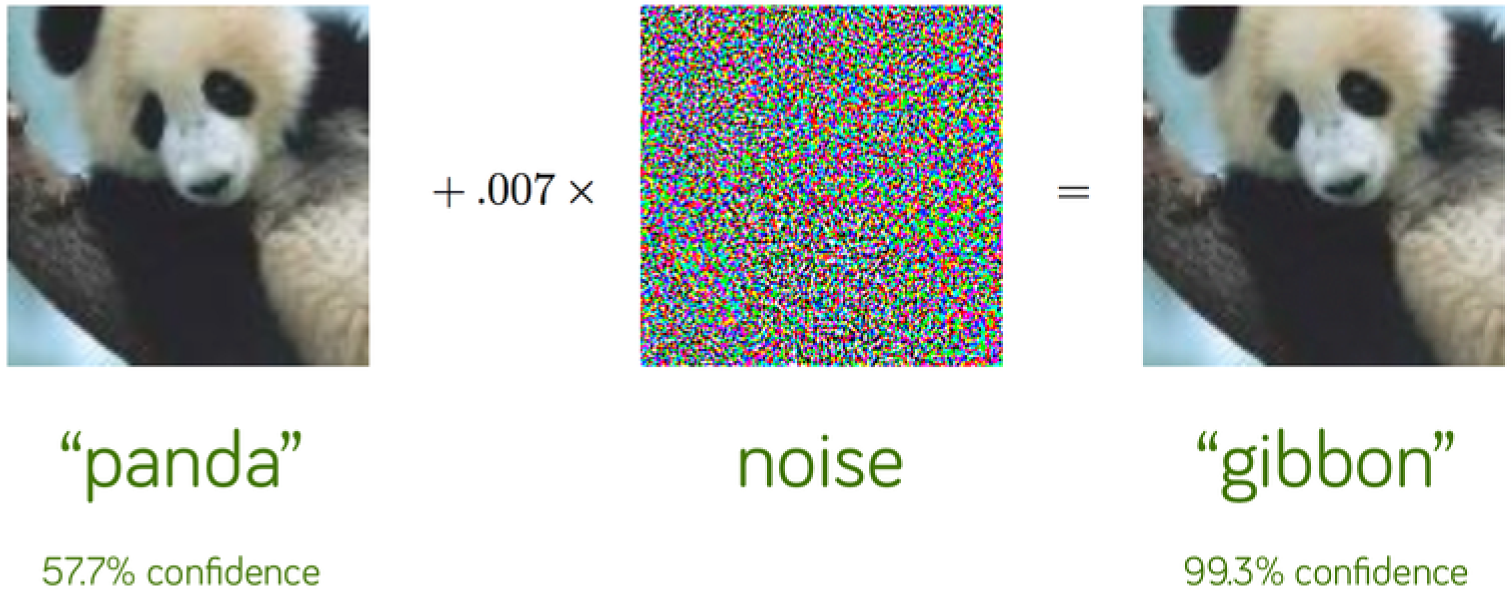
- Machine Learning as a Service (MLaaS)
  - Companies launch MLaaS very competitively



Complex **Simple**

Introduction

- ❖ Adversarial Examples
  - Explaining and harnessing adversarial examples (2015 ICLR)

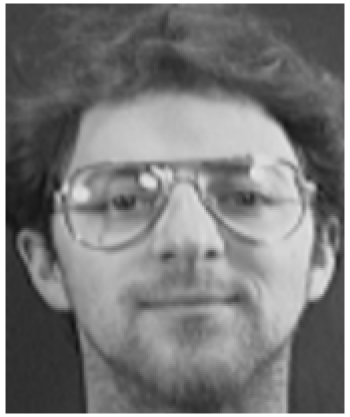


Introduction

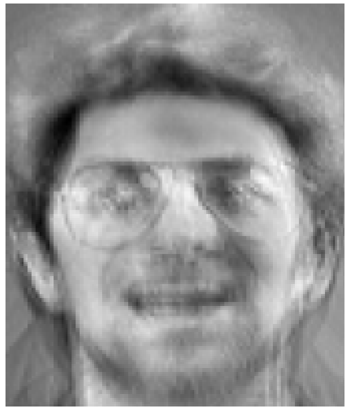
- Model Inversion Attacks
  - Model Stealing
    - Weight Stealing (black-box)
  - Membership Inference Attack
    - Retrieve Training Dataset (white-box, black-box)

Introduction

- ❖ Model Inversion Attacks
  - – Model Stealing
    - ▪ Weight Stealing
  - – Membership Inference Attack
    - ▪ Retrieve Training Dataset



Train Data



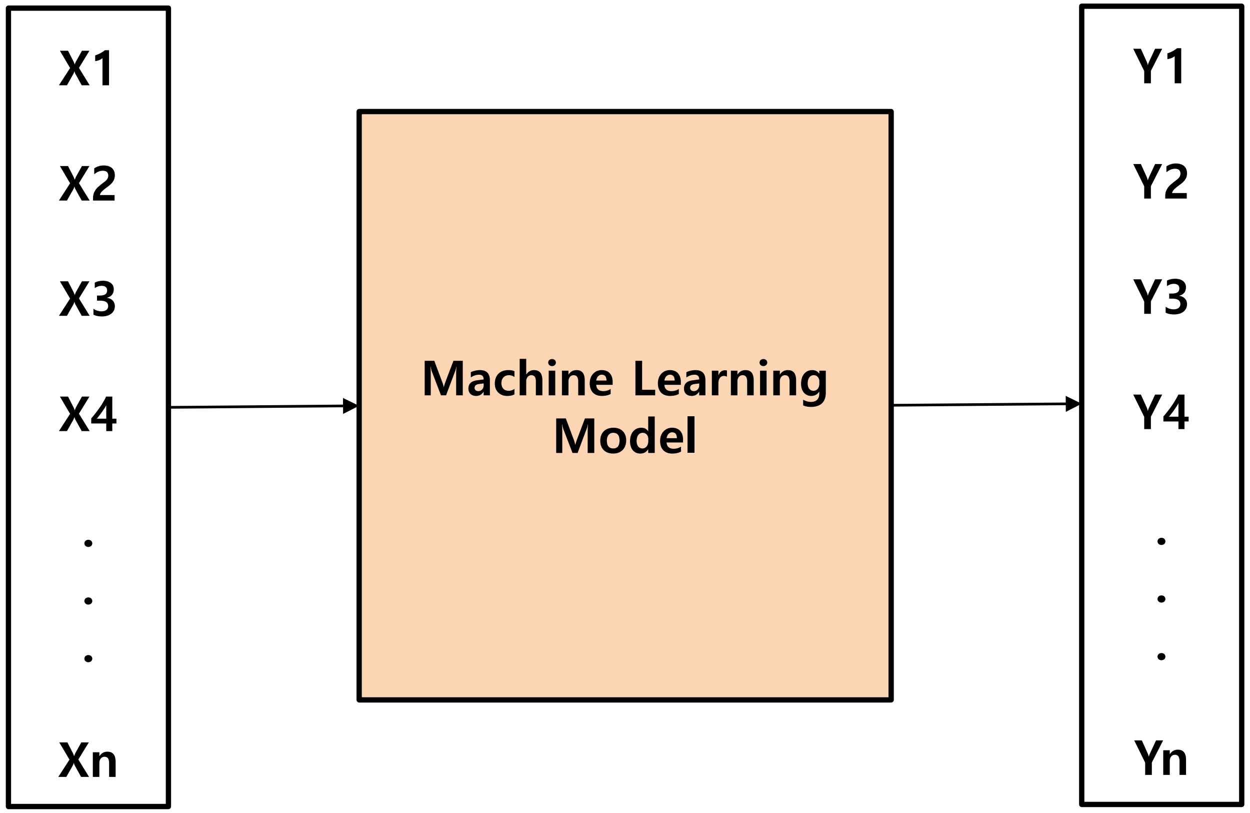
Reconstructed Data

Contribution

- ❖ Model Inversion Attack that exploits confidence values
- ❖ Evaluation of attack from two different settings
  - – Decision Tree for lifestyle survey
  - – Neural Network for facial recognition
- ❖ Countermeasures suggestion

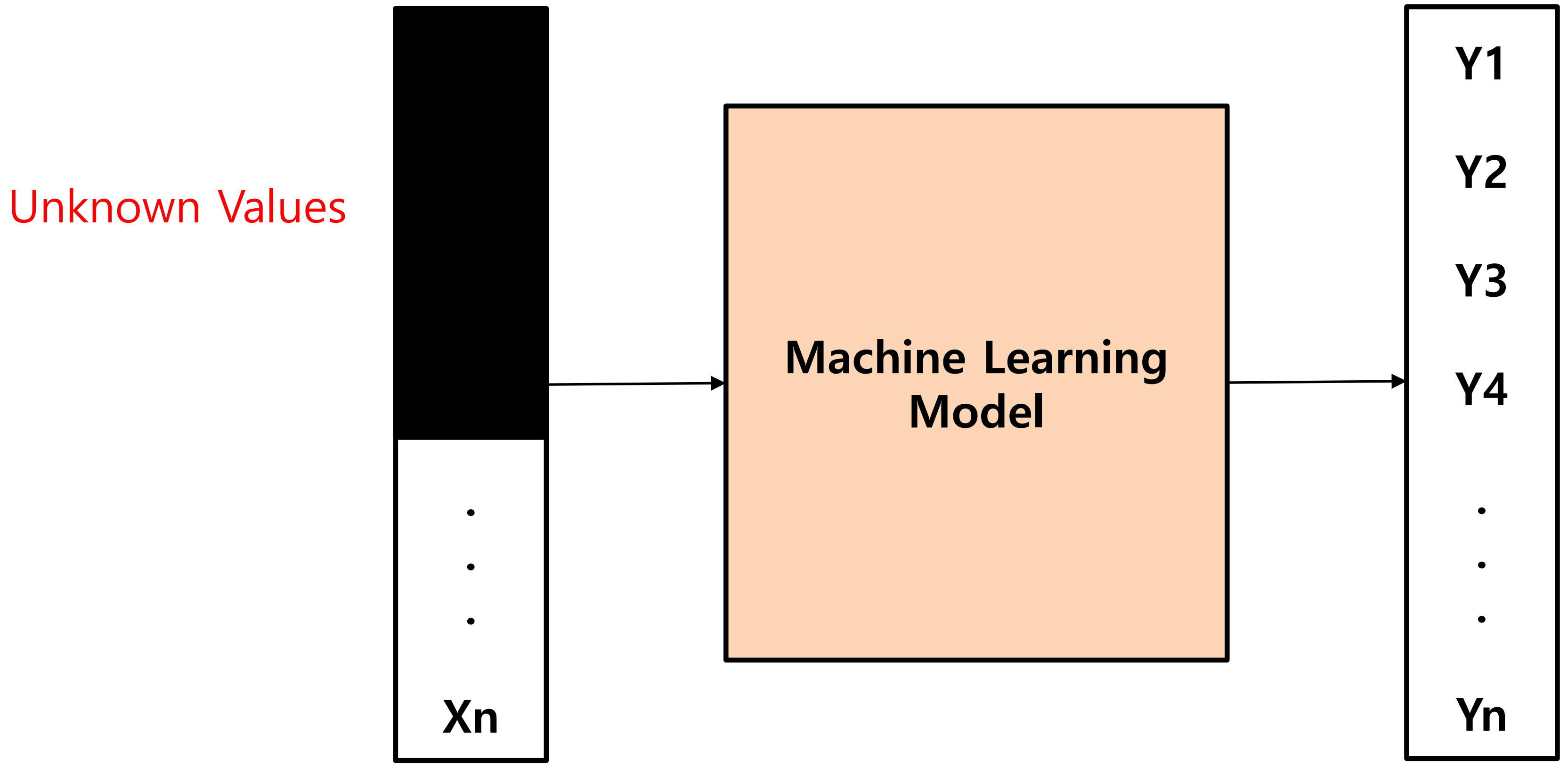
Background

- Model Inversion Attack
  - Reveal unknown values from the available information



Background

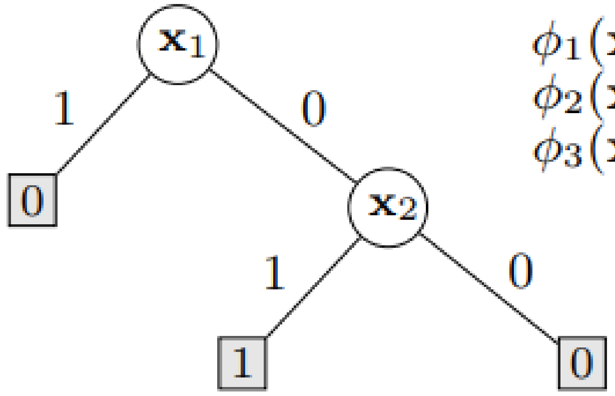
- Model Inversion Attack
  - Reveal unknown values from the available information



SysSec System Security Lab

Decision Tree

- ❖ One of the **simplest** machine learning technique
- ❖ Partitioned feature space into **disjoint** regions



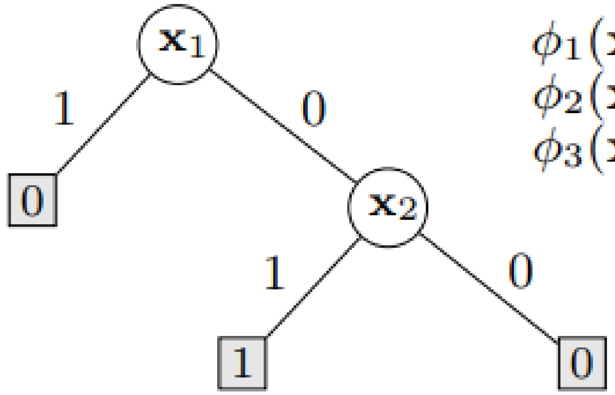
$$\begin{aligned} \phi_1(\mathbf{x}) &= \mathbf{x}_1 & w_1 &= 0 \\ \phi_2(\mathbf{x}) &= (1-\mathbf{x}_1)(\mathbf{x}_2) & w_2 &= 1 \\ \phi_3(\mathbf{x}) &= (1-\mathbf{x}_1)(1-\mathbf{x}_2) & w_3 &= 0 \end{aligned}$$

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

- – ϕ_i : *basis function for the region i*
- – w_i : *common response observed in the training set within region i*

Decision Tree

- One of the **simplest** machine learning technique
- Partitioned feature space into **disjoint** regions



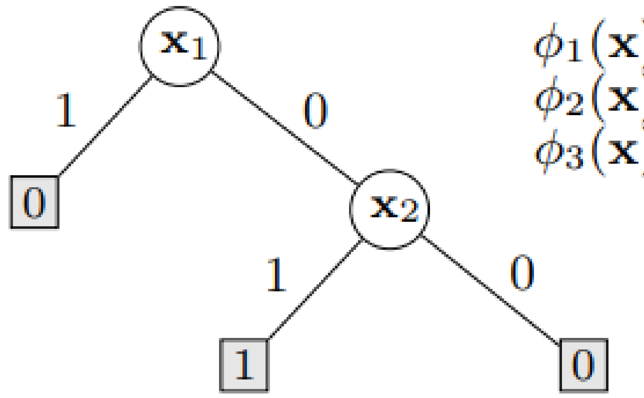
$\phi_1(\mathbf{x}) = \mathbf{x}_1$	$w_1 = 0$
$\phi_2(\mathbf{x}) = (1 - \mathbf{x}_1)(\mathbf{x}_2)$	$w_2 = 1$
$\phi_3(\mathbf{x}) = (1 - \mathbf{x}_1)(1 - \mathbf{x}_2)$	$w_3 = 0$

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

$$f(\mathbf{x}) = \sum_{i=1}^{m} w_i \phi_i(\mathbf{x}), \text{ where } \phi_i(\mathbf{x}) \in \{0, 1\}$$

Decision Tree

- One of the **simplest** machine learning technique
- Partitioned feature space into **disjoint** regions



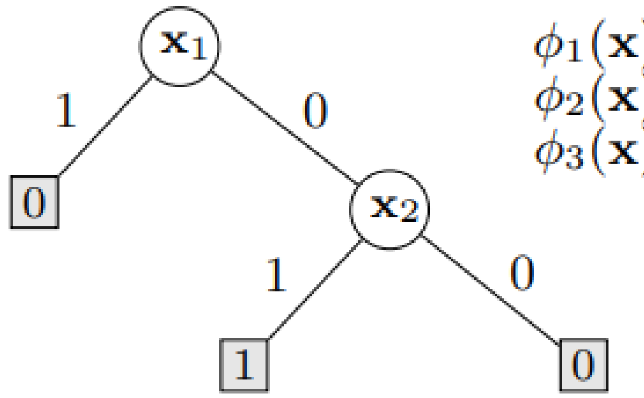
$$\begin{aligned} \phi_1(\mathbf{x}) &= \mathbf{x}_1 & w_1 &= [89, 11] \\ \phi_2(\mathbf{x}) &= (1-\mathbf{x}_1)(\mathbf{x}_2) & w_2 &= 1 \\ \phi_3(\mathbf{x}) &= (1-\mathbf{x}_1)(1-\mathbf{x}_2) & w_3 &= 0 \end{aligned}$$

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

- To return confidence values, we can change ω_i to the range

Decision Tree

- Classification $f(x)$, confidence values $\tilde{f}(x)$



$$\begin{aligned} \phi_1(\mathbf{x}) &= \mathbf{x}_1 & w_1 &= [89, 11] \\ \phi_2(\mathbf{x}) &= (1-\mathbf{x}_1)(\mathbf{x}_2) & w_2 &= 1 \\ \phi_3(\mathbf{x}) &= (1-\mathbf{x}_1)(1-\mathbf{x}_2) & w_3 &= 0 \end{aligned}$$

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

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

Threat Model for Decision Tree

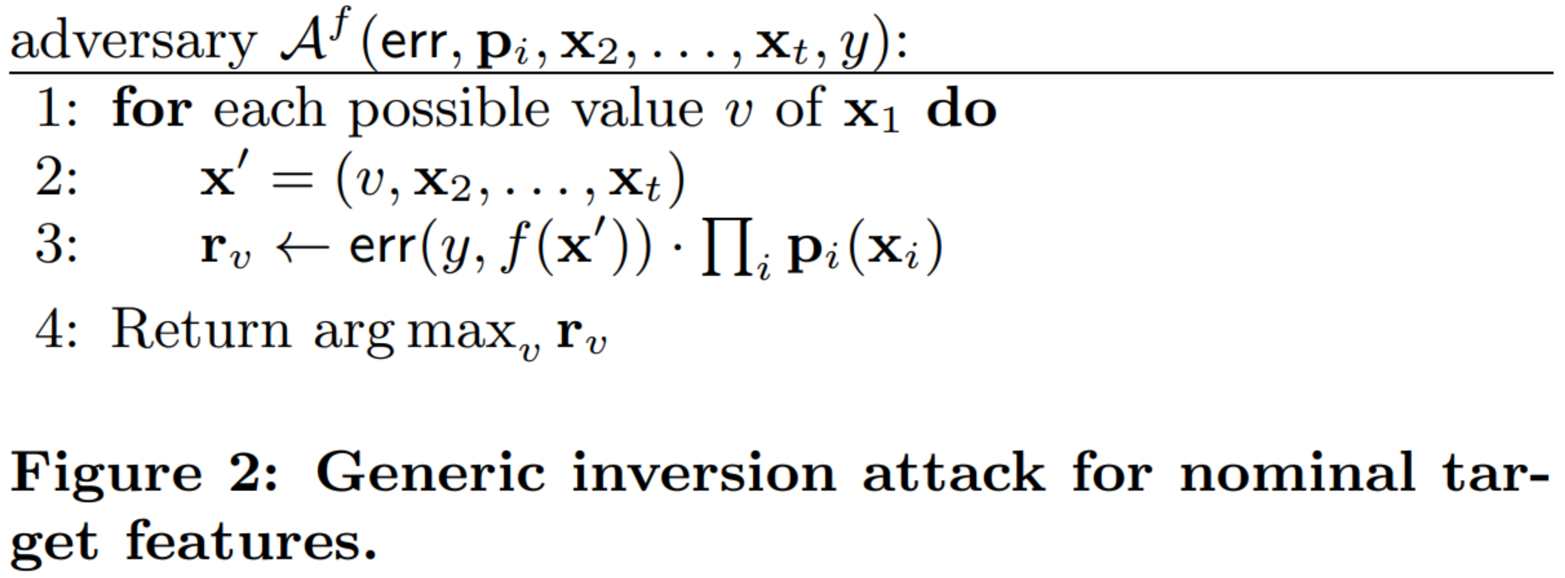
- Goal
  - Predict sensitive feature from the output and partial input feature
    - Black-box: output confusion matrix & partial input feature
    - White-box: output & partial input feature & # of training samples for each class

- Datasets
  - FiveThirtyEight surveys
    - Sensitive Features
      - Whether each participant responded "Yes" to infidelity questions
  - General Social Survey (GSS) marital happiness survey
    - Sensitive Features
      - How happy are you in your marriage?
      - Have you watched X-rated movies in the last year?

Algorithm

❖ Previous Works

– Exhaustive Search (Brute-force attack)



```
adversary $\mathcal{A}^f(\mathsf{err}, \mathbf{p}_i, \mathbf{x}_2, \ldots, \mathbf{x}_t, y)$:
1: for each possible value $v$ of $\mathbf{x}_1$ do
2:     $\mathbf{x}' = (v, \mathbf{x}_2, \ldots, \mathbf{x}_t)$
3:     $\mathbf{r}_v \leftarrow \mathsf{err}(y, f(\mathbf{x}')) \cdot \prod_i \mathbf{p}_i(\mathbf{x}_i)$
4: Return $\arg\max_v \mathbf{r}_v$
```

Figure 2: Generic inversion attack for nominal target features.

– **Infeasible** for intractably large input space

Algorithm

- Two types of problems
  - Black-box
    - Perform algorithm of previous works with different error function
    - With confusion matrix C of decision tree,
    - We can calculate $err(y, y') \propto \Pr[f(x) = y' | y\ is\ the\ true\ label$
  - White-box
    - Attacker knows $p_i = \frac{n_i}{N}$ $, n_i\ is\ sample\ count\ in\ the\ training\ set$
    - Solve maximization problem,
    - Maximize confidence value

$$\max \frac{1}{\sum_{j=1}^{m} p_j \phi_j(v)} \sum_{1 \leq i \leq m} p_i \phi_i(v) \cdot \Pr[x_1 = v]$$

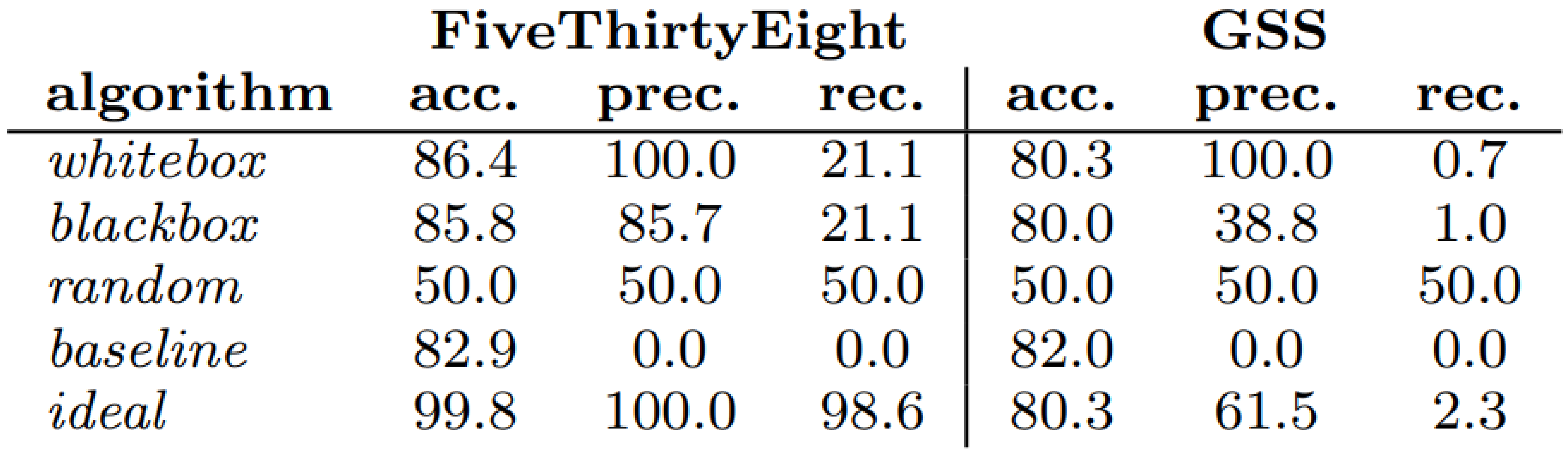
    - Find v that maximizes this equation

Evaluations

- Attackers Assumption
  - Total 5 attackers
    - White-box
    - Black-box
    - Random: "Yes" or "No"
    - Baseline: always "No"
    - Ideal: a new decision tree trained from the same dataset to predict feature

Evaluations

❖ Results



algorithm	FiveThirtyEight acc.	FiveThirtyEight prec.	FiveThirtyEight rec.	GSS acc.	GSS prec.	GSS 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

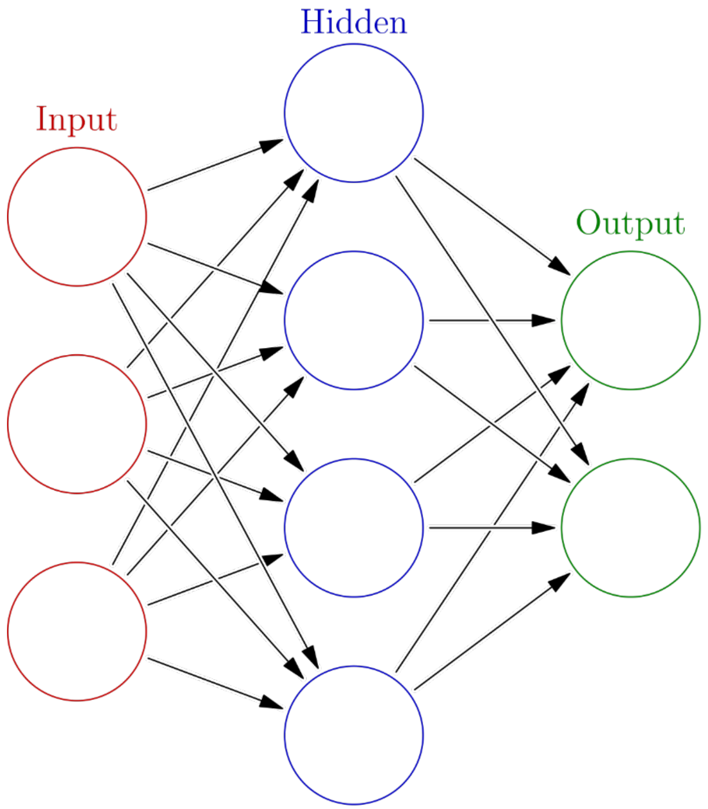
Figure 4: MI results for for BigML models. All numbers shown are percentages.

Summary for Decision Tree

- ❖ They formulated decision tree mathematically
- ❖ They defined a confidence score for the decision tree
- ❖ They proposed a maximization problem to find sensitive features for white-box attack
- ❖ They achieved 100% precision for the white-box attack

Neural Network

❖ Facial Recognition System



Elon musk !

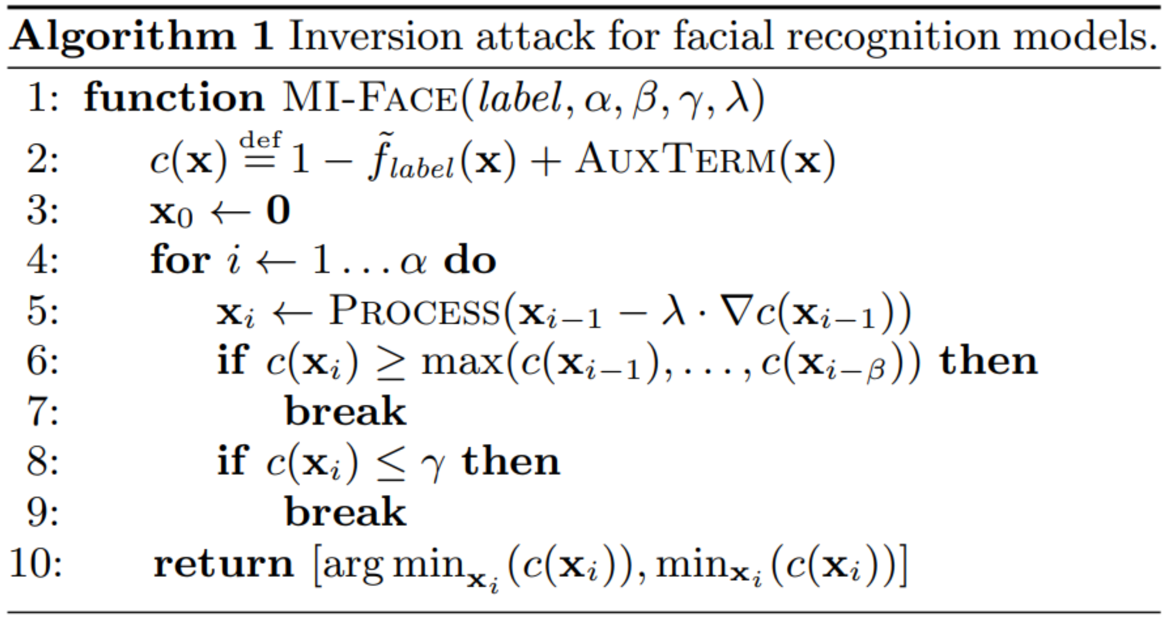
Threat Model for Neural Network

- Goal
  - Find original face of victim from the neural network
    - White-box model
    - Revealed confidence values
- Three types of neural network
  - Softmax
    - 1-layer softmax
  - MLP
    - 3000 hidden sigmoid unit + 1-layer softmax
  - Denoising AutoEncoder (DAE)
    - 2-layer: one for the embedding, one for the output

Algorithm

❖ Inversion attack algorithm

- $c(x)$: *cost function*
- $\tilde{f}(x)$: *model function*
- $AuxTerm$: *auxiliary function*
- λ: *gradient step size*
- α: *maximum iteration*
- γ: *threshold for cost*
- β: *max improvement iteration*



Algorithm 1 Inversion attack for facial recognition models.

```
1:  function MI-FACE($label, \alpha, \beta, \gamma, \lambda$)
2:      $c(\mathbf{x}) \overset{\text{def}}{=} 1 - \tilde{f}_{label}(\mathbf{x}) + \text{AUXTERM}(\mathbf{x})$
3:      $\mathbf{x}_0 \leftarrow \mathbf{0}$
4:      for $i \leftarrow 1 \ldots \alpha$ do
5:          $\mathbf{x}_i \leftarrow \text{PROCESS}(\mathbf{x}_{i-1} - \lambda \cdot \nabla c(\mathbf{x}_{i-1}))$
6:          if $c(\mathbf{x}_i) \geq \max(c(\mathbf{x}_{i-1}), \ldots, c(\mathbf{x}_{i-\beta}))$ then
7:              break
8:          if $c(\mathbf{x}_i) \leq \gamma$ then
9:              break
10:     return $[\arg\min_{\mathbf{x}_i}(c(\mathbf{x}_i)), \min_{\mathbf{x}_i}(c(\mathbf{x}_i))]$
```

- Gradient descent method to find **x (target image)**

Evaluation

- Dataset
  - AT&T Face Database
    - 40 labels
- User study (Mechanical Turk)
  - Find the most similar pictures among 5 candidates
    - Include “not present” answer

Evaluation

❖ Reconstruction Result

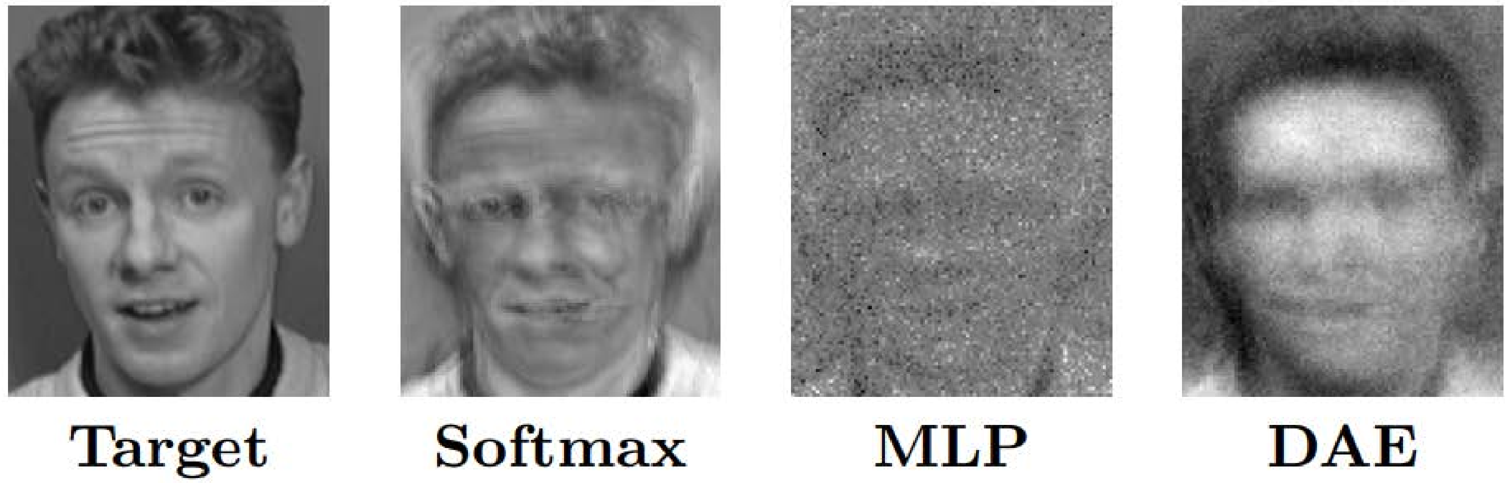


Figure 10: Reconstruction of the individual on the left by Softmax, MLP, and DAE.

Evaluation

- Result
  - Identified: workers identified the correct image from candidates
  - Excluded: workers identified the correct image is not presented

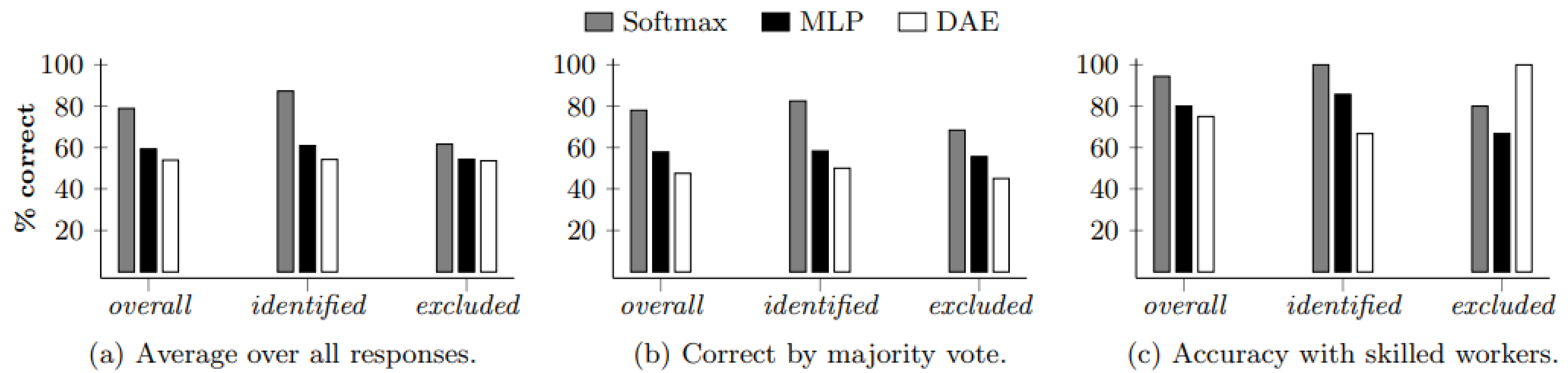


Figure 9: Reconstruction attack results from Mechanical Turk surveys. "Skilled workers" are those who completed at least five MTurk tasks, achieving at least 75% accuracy.

Summary for Neural Network

- Facial recognition models with three types of architecture
  - Softmax, MLP, Denoising AutoEncoder
- Gradient-based optimization algorithm
- More than 50% accuracy from Mechanical Turk Evaluation for overall models

Countermeasures

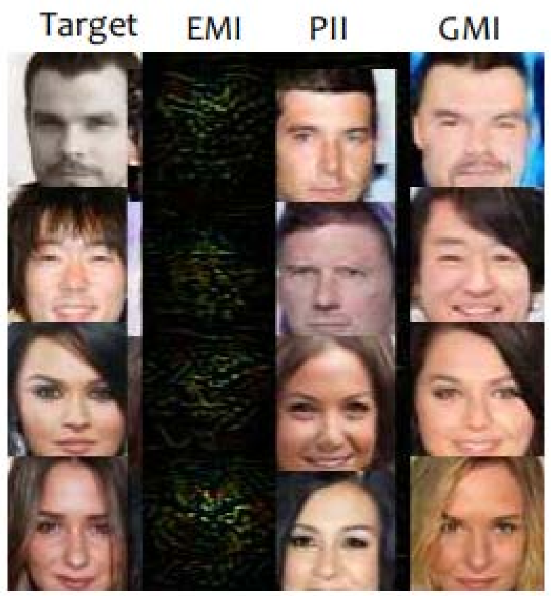
- Decision Tree
  - Change the order of feature
    - When the target feature is placed on the top or bottom of the tree, the attack accuracy is degraded
    - Why?
      - It is training algorithm for decision tree
      - CART algorithm (Classification and Regression Tree)
- Neural Network
  - Gradient obfuscation
    - Best mitigation for FGSM, C&W, ZOO, etc.
- Overall
  - **Don't reveal** confidence score (return rounded value)

Limitation

- Decision tree attack requires partial information of input features
- Decision tree Black-box attack still used the same algorithm from the previous works
- Neural networks used in the evaluation is too simple
  - Only 1~2 layer architecture
- Reconstructed face picture is actually not similar at all with the real data

Q&A

- Junho Ahn
  - After this work, how model inversion attack is developed? This is very first paper that propose model inversion attack and the reconstructed image is not clear. Can we do that almost same today?
  - Much more progress, but not perfect
    - More complex algorithm
    - New types of network (GAN)
    - Models don't remember exact data



Ref: Zhang, Yuheng, et al. "The secret revealer: Generative model-inversion attacks against deep neural networks." *(2020 CVPR)*

Q&A

- JoonHa Jang **(*Best Question)**
  - In the case of a model inversion attack, I think it can easily defend it by limiting the number of consecutive queries and rounding the confidence score. Even with these limitations, I would like to know the advantages of a model inversion attack.
    - Very effective mitigation !
    - But, benign applications also requires tons of queries to run their services
    - Query efficiency is one of the critical measure for adversarial machine learning attack
    - And, if the model is white-box, there is no query restriction

Q&A

- Beokseok Oh
  - Assume adversary knows what model is used in ML (unknown parameters, known model). Then adversary can possibly run the model many times and find parameters so that they can re-generate model. How can this be protected?
    - Transfer Learning Mitigation
    - Transfer learning should train transfer model with very low cost compared to the original one
    - If you want to study the transferability more
      - "Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks" USENIX 19'

Q&A

- Yonghwa Lee
  - I think that in ML, the most important thing is a dataset used in training. Can we use some special training dataset in mitigating Model Inversion Attacks?
    - It is another training research field called **“Adversarial Training”**
  - Adversarial Training
    - Generate train data with noise to make robust models

Conclusion

- Model inversion attack against Decision Tree and Neural Network
  - Exploited confidence value for both scenarios
    - Decision Tree
    - Neural Network
- Countermeasures for two settings