Privacy Attacks against Machine Learning

Ambra Demontis, Battista Biggio

Department of Electrical and Electronic Engineering
University of Cagliari, Italy

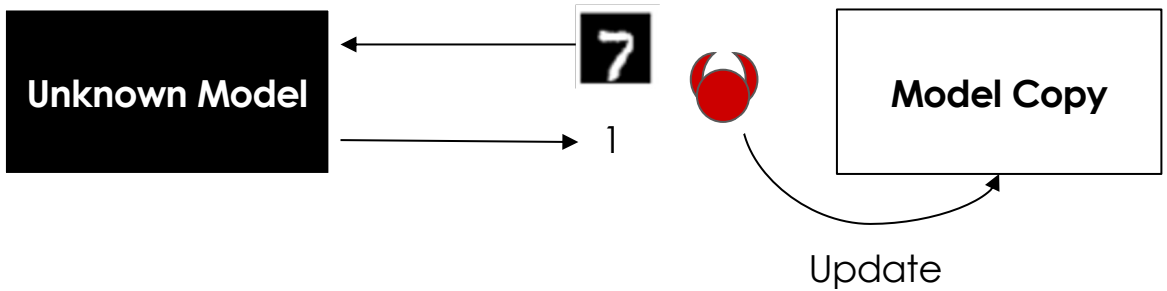
Attacks against Machine Learning

	Attacker's Goal		
	Misclassifications that do not compromise normal system operation	Misclassifications that compromise normal system operation	Querying strategies that reveal confidential information on the learning model or its users
Attacker's Capability	**Integrity**	**Availability**	**Privacy / Confidentiality**
Test data	Evasion (a.k.a. adversarial examples)	*Sponge Attacks*	***Model extraction / stealing*** ***Model inversion (hill climbing)*** ***Membership inference***
Training data	*Integrity Poisoning (to allow subsequent intrusions) – e.g., backdoors*	*DoS poisoning (to maximize classification error)*	-

Biggio & Roli, *Wild Patterns*, PR 2018 https://arxiv.org/abs/1712.03141

Model Extraction/Stealing

Construct a copy of a model, being able to query the target model.



Model Extraction/Stealing

Two possible goals:

Task accuracy - create a copy that can *perform well the original task*
(stealing intellectual property)

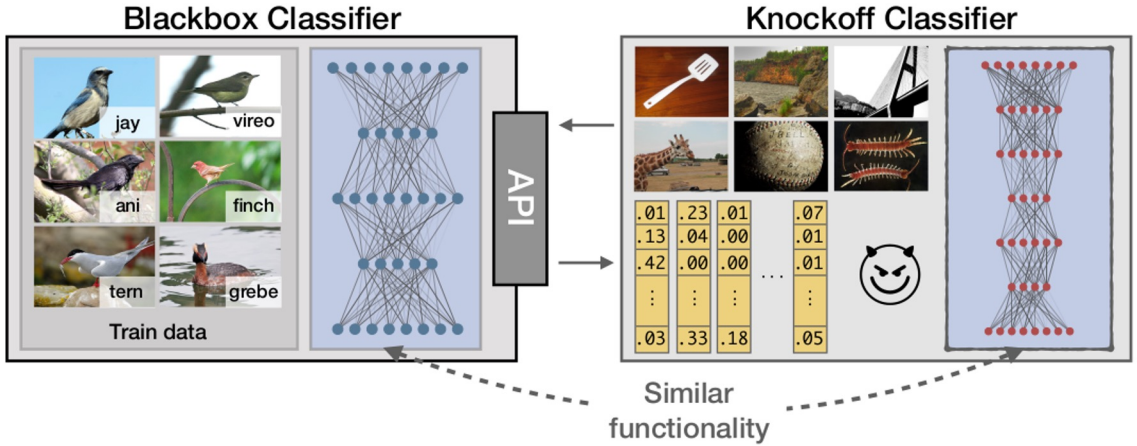
High fidelity- create a copy that *makes the same errors as the original model*
(having a surrogate similar to the original, e.g., to compute attacks.)

Model Extraction/Stealing [task accuracy]

They use a two-step approach that:

1. query the target model and collects output
2. train the surrogate

Interestingly, they found that by querying the model with **random** images taken from a distribution different from the one of the training set, they can build an accurate surrogate.



Orekondy et al., *Knockoff Nets: Stealing Functionality of Black-Box Models*, CVPR 2019

Model Extraction/Stealing [task accuracy]

Using randomly chosen images, it is possible creating a surrogate, but it is necessary to perform many queries.

The authors of [1] propose a technique (called *adaptive*) that crafts the images to improve the **query efficiency**.

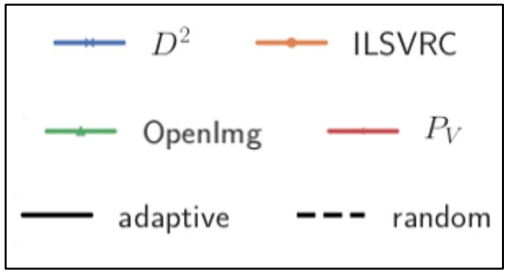
This technique **encourages images in which the victim is confident** (images similar to the ones of the unknown training dataset) and **diversity** between the generated images.

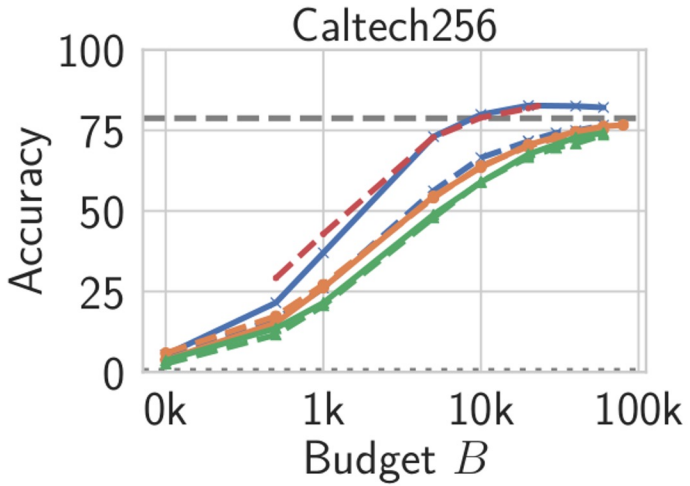
They test the proposed method on deep neural networks and suggest using complex architectures as surrogates.

[1] Orekondy et al., *Knockoff Nets: Stealing Functionality of Black-Box Models*, CVPR 2019

Model Extraction/Stealing [task accuracy]

Accuracy of the surrogate model for an increasing number of queries (Budget), when the attacker uses as starting point various datasets and techniques to sample the images (one line for dataset and technique).



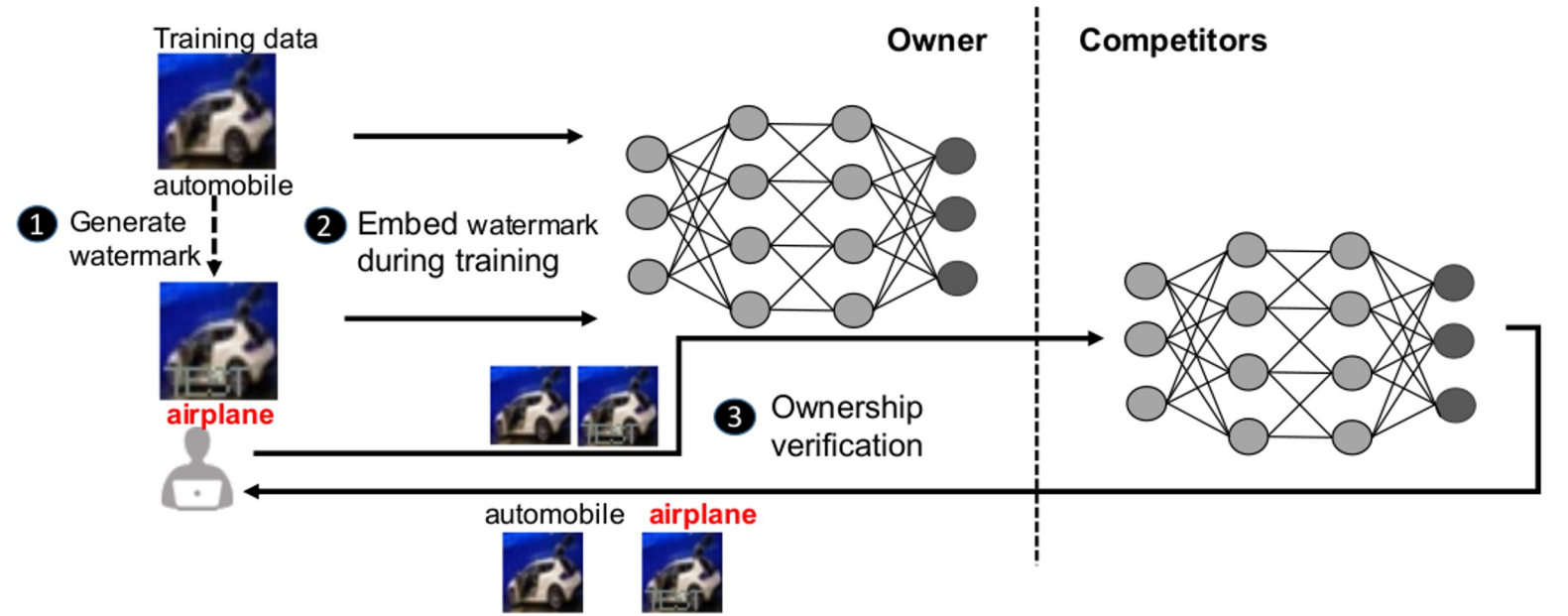


P_V: same dataset used to train the target;
ILSVRC: overlap 42%;
OpenImg: overlap 44%;
D^2: dataset made by ILSVRC, OpenImg, and others.

Orekondy et al., *Knockoff Nets: Stealing Functionality of Black-Box Models*, CVPR 2019

Defenses against Model Extraction/Stealing

Defense strategy 1: Makes the model learn some **watermarks** (patterns that cause an unexpected misclassification when added to an image) that can be used to **prove the intellectual property** of the model.



Zhang et al., Protecting Intellectual Property of Deep Neural …, AsiaCCS 2018

Defenses against Model Extraction/Stealing

Classifiers: a small CNN for the MNIST and an RNN for the Speech Command dataset.

The approach proposed in [1] is called EWE, Baseline is an approach previously proposed.

Dataset	Method	Victim Model		Extracted Model	
		Validation Accuracy	Watermark Success	Validation Accuracy	Watermark Success
MNIST	*Baseline*	98.28(±0.57)%	100.00(±0.00)%	98.35(±0.29)%	3.09(±2.80)%
	EWE	97.75(±0.59)%	99.85(±0.24)%	97.58(±0.78)%	61.44(±27.85)%
Fashion MNIST	*Baseline*	90.22(±0.27)%	100.0(±0.00)%	89.43(±0.41)%	5.75(±2.66)%
	EWE	90.42(±1.03)%	99.88(±0.31)%	89.45(±0.93)%	44.90(±24.70)%
Speech Command	*Baseline*	97.00(±4.31)%	100.00(±0.00)%	96.78(±4.96)%	22.58(±25.09)%
	EWE	96.19(±0.38)%	100.00(±0.00)%	96.65(±0.53)%	68.16(±28.30)%

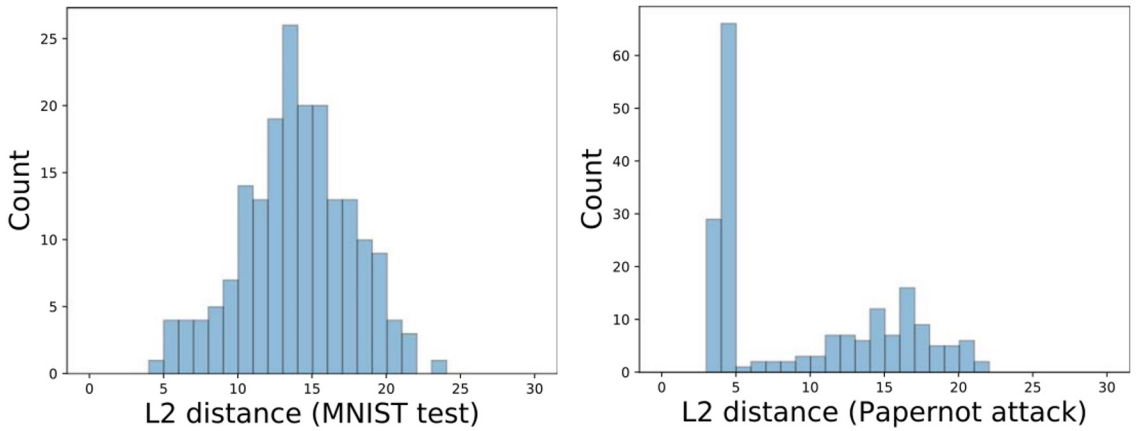
[1] Jia et al., *Entangled Watermarks as a Defense against Model Extraction*, Usenix 2021

Defenses against Model Extraction/Stealing

Defense strategy 2: Understand that attackers are querying the model and *block them*.

The authors of PRADA [1] noticed that, the distance between consecutive queries, for legitimate queries usually follow a normal distribution.

The queries usually made to perform model extraction are made ad-hoc to maximize the information extracted. Therefore, they do not follow a normal distribution.



Juuti et al., *PRADA: Protecting Against DNN Model Stealing Attacks*, ArXiv 2018

Defenses against Model Extraction/Stealing

Effectiveness of PRADA for models trained on different datasets (MNIST/GTSRB) and extracted with different attacks (Tramer, Pap. T-RND, Color).

Model (δ value)	FPR	Queries made until detection			
		TRAMER	PAP.	T-RND	COLOR
MNIST (0.95)	0.0%	5,560	120	140	-
MNIST (0.96)	0.0%	5,560	120	130	-
GTSRB (0.87)	0.0%	5,020	430	missed	550
GTSRB (0.90)	0.6%	5,020	430	missed	480
GTSRB (0.94)	0.1%*	5,020	430	440	440

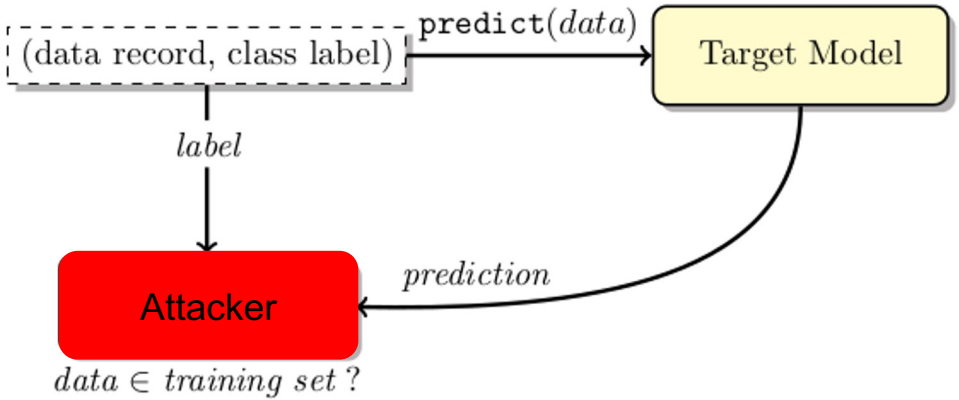
δ is a threshold that defines the trade-off between the detection capabilities of PRADA and its number of false positive.

Juuti et al., *PRADA: Protecting Against DNN Model Stealing Attacks*, ArXiv 2018

Membership Inference Attacks

Goal: identify whether an input sample is part of the training set used to learn a deep neural network having query access to the target model.

First, they query the deep neural network with the input sample, and they get the predictions.



Shokri et al., *Membership Inference Attacks Against Machine Learning Models*, S&P 2017

Membership Inference Attacks

Then, the attacker **crafts many surrogate models that imitates the behavior of the original deep neural network** and trains them:

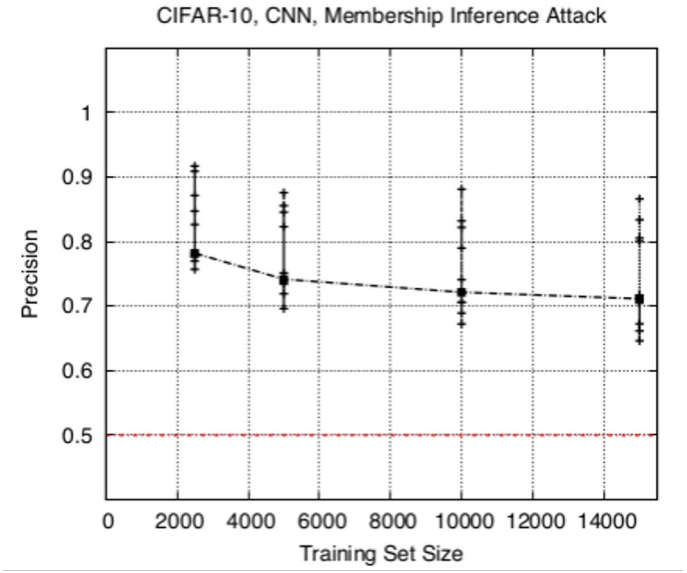
- including the test samples in the training dataset
- not including the test samples in the training dataset

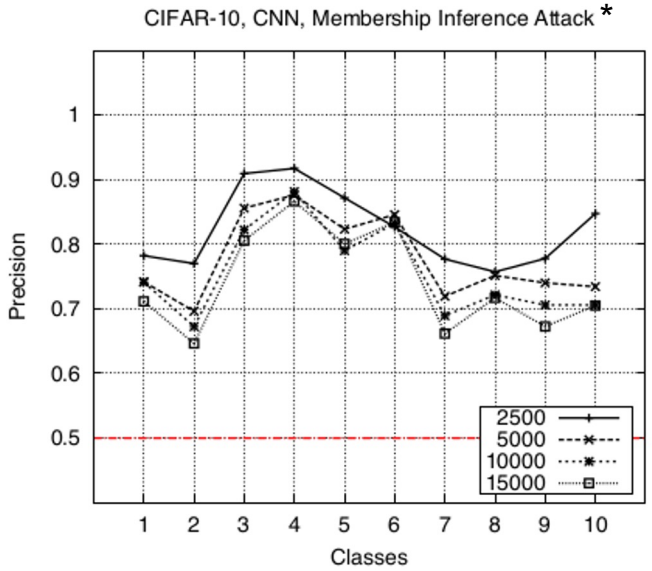
The authors show that by **comparing the predictions** of those models with the one of the original deep network is possible to understand if the test samples were or were not part of the original deep network training dataset.

Shokri et al., *Membership Inference Attacks Against Machine Learning Models*, S&P 2017

Membership Inference Attacks

Precision for different classes while varying the size of the training datasets.
Dataset CIFAR-10.





*Training dataset size in legend.

Shokri et al., *Membership Inference Attacks Against Machine Learning Models*, S&P 2017

Defenses against Membership Inference Attacks

Different types of defenses that have been proposed against this attack:

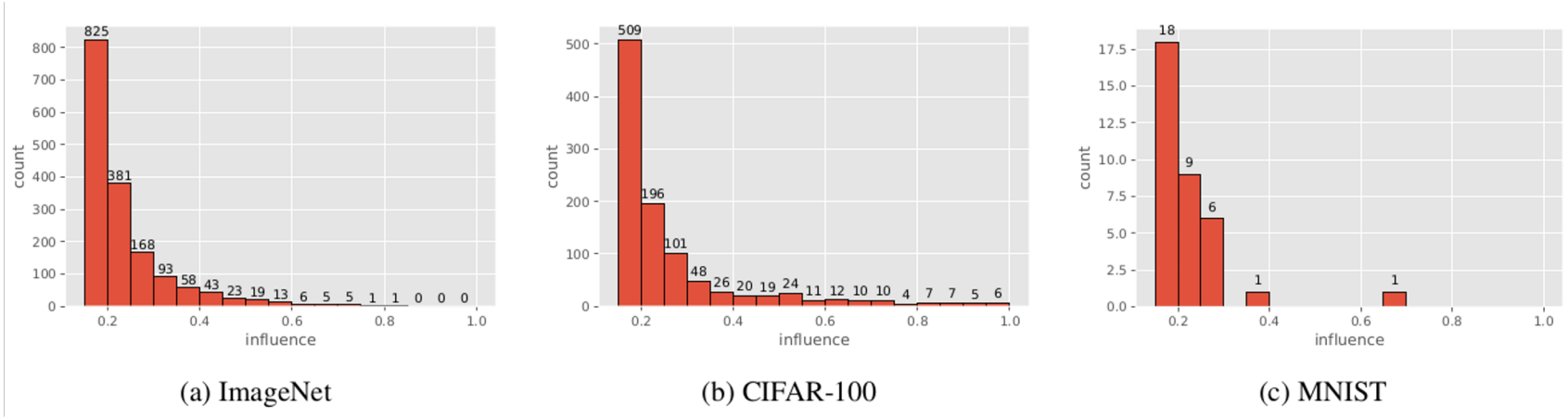
- regularization;
- differential privacy;
- prediction vector tampering.

Rigaki et al., *A Survey of Privacy Attacks in Machine Learning*, ArXiv, 2020

Regularization against Membership Inference Attacks

Deep neural networks tend to memorize training data (they rely a lot on them for their decision and are really confident when predicting them) [1].

Classifiers ResNet50 trained on Imagenet and on CIFAR-100, a small CNN trained on MNIST. Most influent (memorized) samples for different datasets:



(a) ImageNet (b) CIFAR-100 (c) MNIST

Regularizing the model reduces the memorization of the training data.

[1] Feldman et al., *What Neural Networks Memorize and Why:..*, NeurIPS 2020

Differential Privacy against Membership Inference

Ideally, the users of an ML system should not be able to infer information about it.

Differential privacy is based on the notion that if the inputs are similar, the output should be the same.

A randomized mechanism M : D → R with domain D and range R satisfies (ε, δ)-differential privacy if for any two adjacent inputs d, d' ∈ D and for any subset of outputs S ⊆ R it holds that

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta.$$

where ε control how much the two input can differ, and δ is the probability of failure.

To obtain this property, differentially private algorithms *bounds* and *provides guarantees on the impact that single training sample have on the output of a model.*

Rigaki et al, *A Survey of Privacy Attacks in Machine Learning*, ArXiv, 2021

Differential Privacy against Membership Inference

To bound the impact that a single training sample can have on the model output the authors of [1] propose to train neural network with a modified version of the SGD algorithm:

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\sum_i (\bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ)

[1] Abadi et al, *Deep Learning with Differential Privacy*, CCS, 2016

Differential Privacy against Membership Inference

Differential privacy offers a trade-off between privacy protection and utility or model accuracy.

Although differential privacy is the most studied defense against this attack, different studies concluded that the models **could offer privacy protection only when they considerably sacrifice their utility**.

Rigaki et al, *A Survey of Privacy Attacks in Machine Learning*, ArXiv, 2021

Prediction Vector Tampering against Membership Inference

Privacy attacks usually assume knowledge of the classifier's **scores**.

A countermeasure proposed for classifiers against membership inference [1] is to perturb the scores to make them less reliable, making these attacks more difficult to accomplish.

Another countermeasure proposed is **avoiding providing the scores** as part of the classifier's output. **However, it did not seem to fully mitigate membership inference attacks** since *information leaks can still happen due to misclassifications* [2,3].

Members and non-members of the training dataset are in fact mislabeled differently (assigned to different wrong classes), and this is enough for inference attacks, such as [2], to work.

[1] Jia et al, *MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples*, CCS, 2019
[2] Shokri et al, *Membership Inference Attacks Against Machine Learning Models*, S&P, 2017
[3] Rigaki et al, *A Survey of Privacy Attacks in Machine Learning*, ArXiv, 2021

Model Inversion Attacks

Goal: to reconstruct training samples having the ability to query the model.

Problem: to find the input that maximizes the returned confidence w.r.t. the target label.

Solving this problem using gradient descent, it may be possible to reconstruct a training image (depending on the targeted model)

Fredrikson et al, *Model inversion attacks that exploit confidence information and basic countermeasures*, ACM CCS, 2015

Model Inversion Attacks

To validate the proposed approach, the authors conducted a study involving humans.

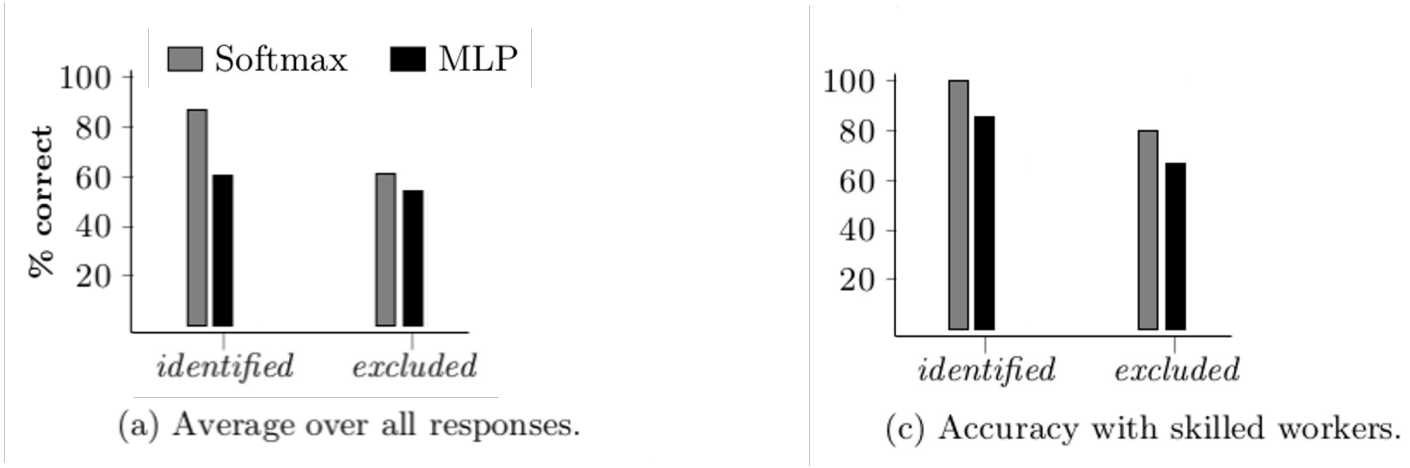
The authors Asked Mechanical Turk workers to **match the reconstructed image** to one of five face images from the original set, **or to respond that the displayed image does not correspond** to one of the five images.

Fredrikson et al, *Model inversion attacks that exploit confidence information and basic countermeasures*, ACM CCS, 2015

Model Inversion Attacks

The results are subdivided in:
"identified" -> the true identity was displayed between the 5 and identified by the worker;
"excluded" -> the identity was not present, and the worker correctly said it was not present.

The colored bars represents different classifiers.



Fredrikson et al, *Model inversion attacks that exploit confidence information and basic countermeasures*, ACM CCS, 2015

Defenses against Model Inversion Attacks

The defenses that have been proposed against this attack consists of **prediction vector tampering**.

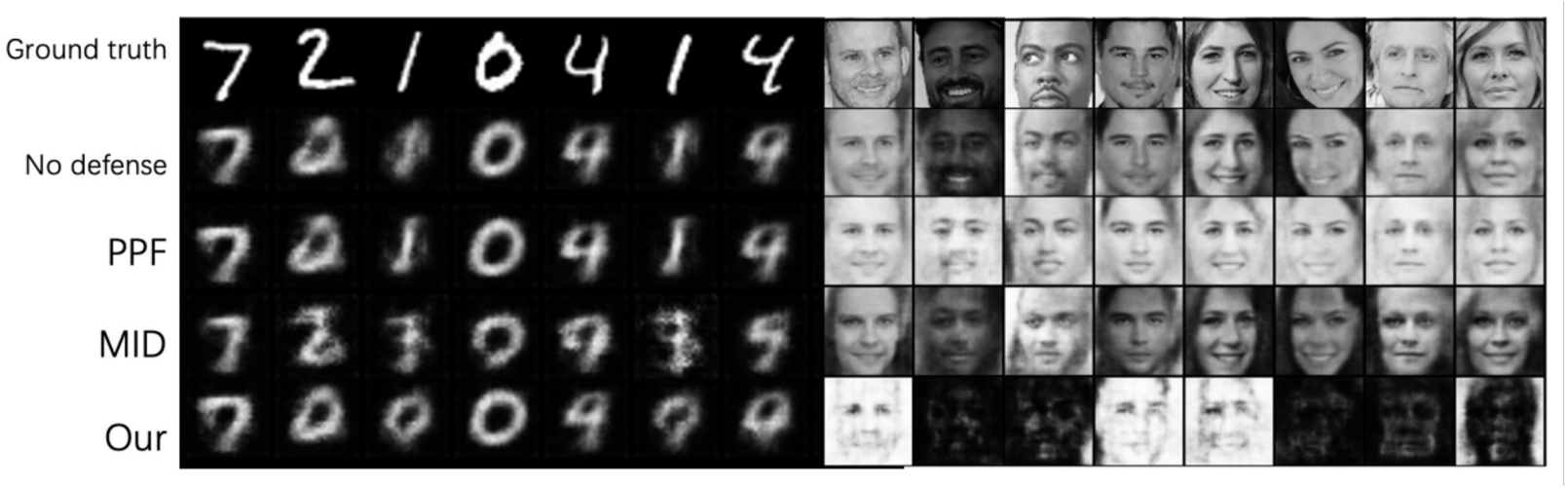
The authors of [1] create a bounded noise vector that maximize the reconstruction error of the inversion model without changing the labels.

$$
\begin{aligned}
\max \quad & \mathcal{R}(\boldsymbol{x}, \mathcal{A}(\boldsymbol{y}+\boldsymbol{e})) && \text{(reconstruction error)} \\
subject\ to: \quad & \boldsymbol{e} \leq \epsilon && \text{(distorsion budget)} \\
& \arg\max(\boldsymbol{y}+\boldsymbol{e}) = \arg\max \boldsymbol{y} && \text{(la label predetta non cambia)} \\
& 0 \leq (\boldsymbol{y}_i + \boldsymbol{e}_i) \leq 1,\ \textstyle\sum(\boldsymbol{y}_i + \boldsymbol{e}_i) = 1 && \text{(the scores vector remains a probability distribution)}
\end{aligned}
$$

Wen et al., *Defending Against Model Inversion Attack by Adversarial Examples*, CSR workshop, 2021

Defenses against Model Inversion Attacks

Dataset FaceScrub530 (45,897 color images, size 64x64, of 530 individuals).
Target classifier: CNN.



Wen et al., *Defending Against Model Inversion Attack by Adversarial Examples*, CSR workshop, 2021

Other Privacy Attacks

Property Inference

Goal: extract dataset properties not encoded as features or not correlated to the learning task.

E.g., extract the ratio of women and men patients in a dataset used to train a model even if the gender was not one of the dataset features.

Attribute Inference

Goal: Infer the value of a sensible attribute.

Rigaki et al, *A Survey of Privacy Attacks in Machine Learning*, ArXiv, 2021