« Security of Machine Learning »

Teddy Furon, IRISA / Inria Rennes

Benoit Bonnet, Thibault Maho, Karim Tit, Samuel Tap, Hanwei Zhang
Laurent Amsaleg, Yannis Avrithis, Patrick Bas, Erwan Le Merrer, Mathias Rousset

Seminar « Hands-on Machine Learning for Security »

This session will focus on all the reasons that we believe data science is fundamental to building a successful and mature cybersecurity program. In particular, we'll discuss two ways to use data science fundamentals in cybersecurity programs:

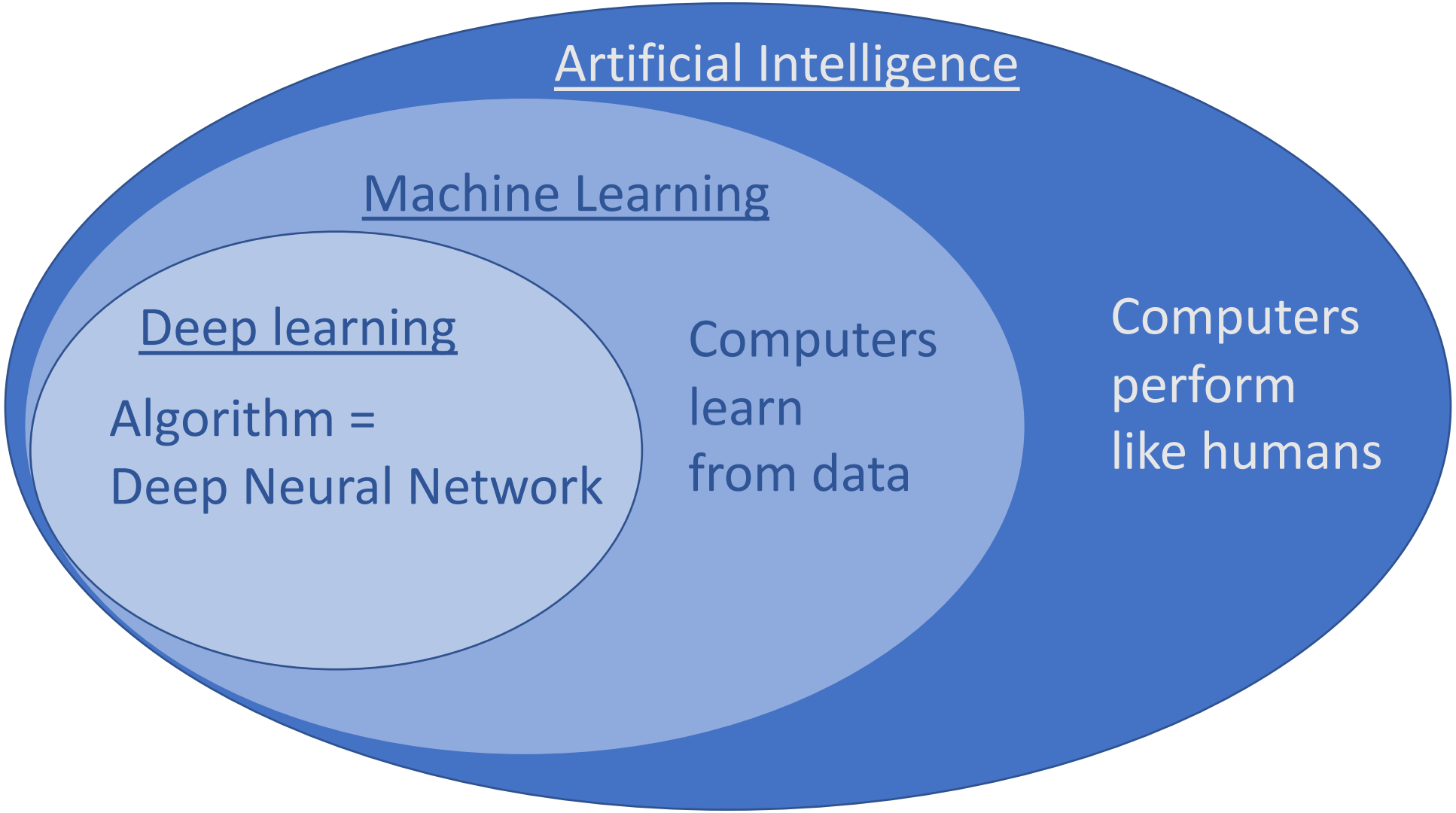
1. Improve data analytics supporting cybersecurity programs (KPIs, risk management, process effectiveness)
2. Use model-driven security and real-time data streaming to match data attributes to known patterns resulting in deviation scores with a threshold that trigger automated actions in front-line cyber controls in milliseconds

My background = Information Forensics & Security

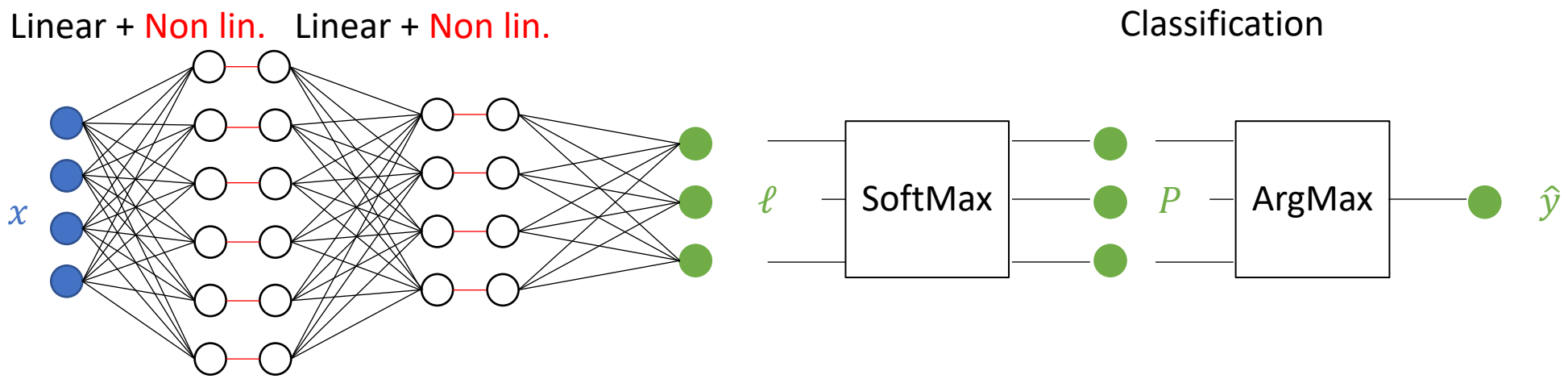
Security of (multimedia) content

Typical applications

- Integrity
  - Multimedia forensics (detection of manipulation, identification of sensors)
  - Steganography et steganalysis (insertion / détection of hidden messages)
- Confidentiality
  - Signal processing in the encrypted domain
  - Security of biometric traits (anti-spoofing / secure storage)
  - Visual secret sharing
  - Security with physical layer (digital com.)
  - Differential privacy / Information leakages
- Ownership / Identification
  - Watermarking
  - Traitor tracing
  - Robust hash


Artificial Intelligence
Machine Learning
Deep learning
Algorithm =
Deep Neural Network
Computers
learn
from data
Computers
perform
like humans

Neural network



inputs

logits

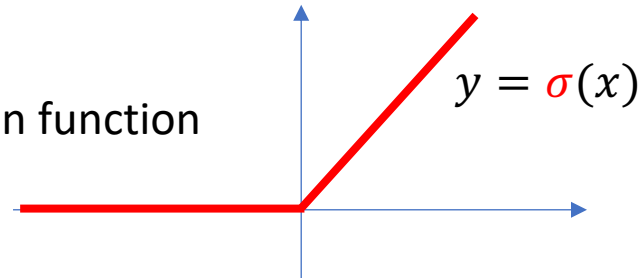
"probabilities"

predicted class

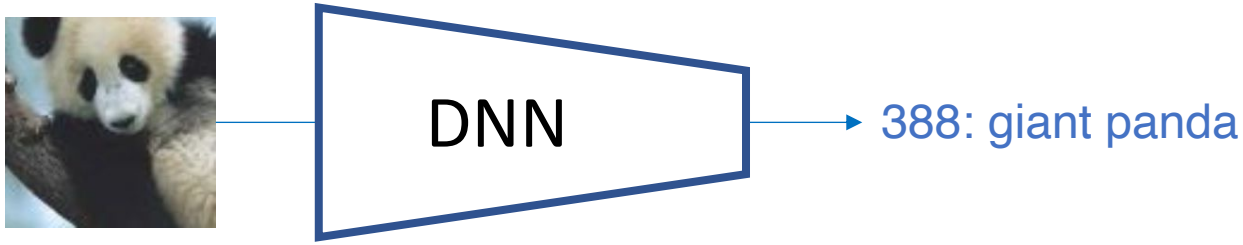
$$\ell = f(x, \theta) = \text{logits}$$
$$\ell = W_3\sigma(W_2\sigma(W_1x + b_1) + b_2)$$

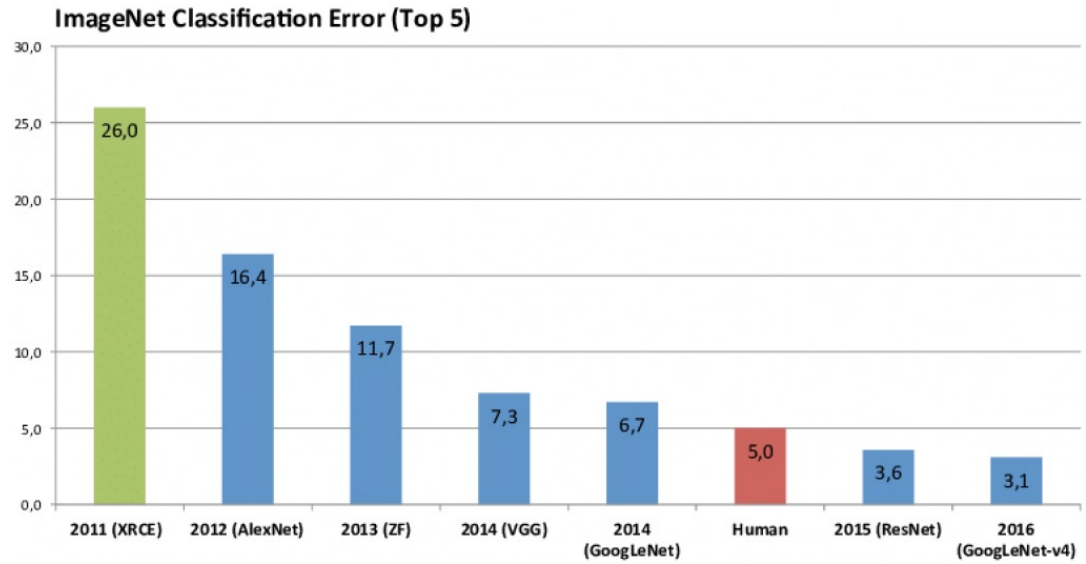
$$P[i] \propto e^{\ell[i]}$$
$$\sum_i P[i] = 1$$

Non lin. activation function



ImageNet challenge: the iconic example of A.I.





2012: DNN AlexNet handily wins the top prize

- Krizhevsky, Sutskever, and Hinton (Univ. of Toronto)
- « *That moment is widely considered a turning point in the development of contemporary AI* »
- « *This dramatic quantitative improvement marked the start of an industry-wide artificial intelligence boom* »

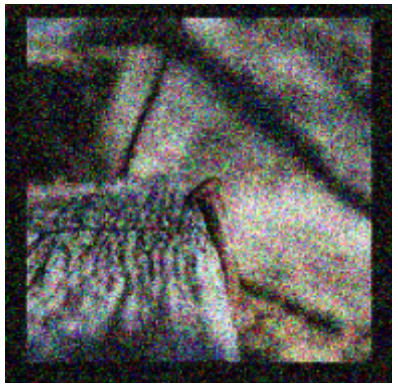
Motivations: false sense of security

- Generalization ≠ Robustness ≠ Security
  - Generalization: To operate as expected on unseen data
  - Robustness: To operate as expected on noisy data
  - Security: To operate as expected on purposely perturbated data

- *Little bits of history repeating* [Propellerheads, 1997]
  - *I've seen it before*: Digital Watermarking
  - *I've seen again*: Content Based Image Retrieval
  - *The next big thing is here*: Machine Learning

- Warning: « Security of M.L. before M.L. for security »

Generalization ≠ Robustness ≠ Security

	Generalization	Robustness		Security	
	original	noise	JPEG	black-box	white-box
					
Prediction	nail	envelope	bulletproof_vest	paintbrush	mantis
Distortion	0	84.9	28.8	6.6	0.2

Generalization ≠ Robustness ≠ Security

	Generalization	Robustness		Security	
	original	noise	JPEG	black-box	white-box
					
Prediction	prayer_rug	lighter	loudspeaker	quilt	safe
Distortion	0	79.1	42.0	19.2	0.5

Generalization ≠ Robustness ≠ Security

	Generalization	Robustness		Security	
	original	noise	JPEG	black-box	white-box
					
Prediction	Lawn_mower	projector	joystick	vacuum	rifle
Distortion	0	73.2	14.5	4.5	0.14

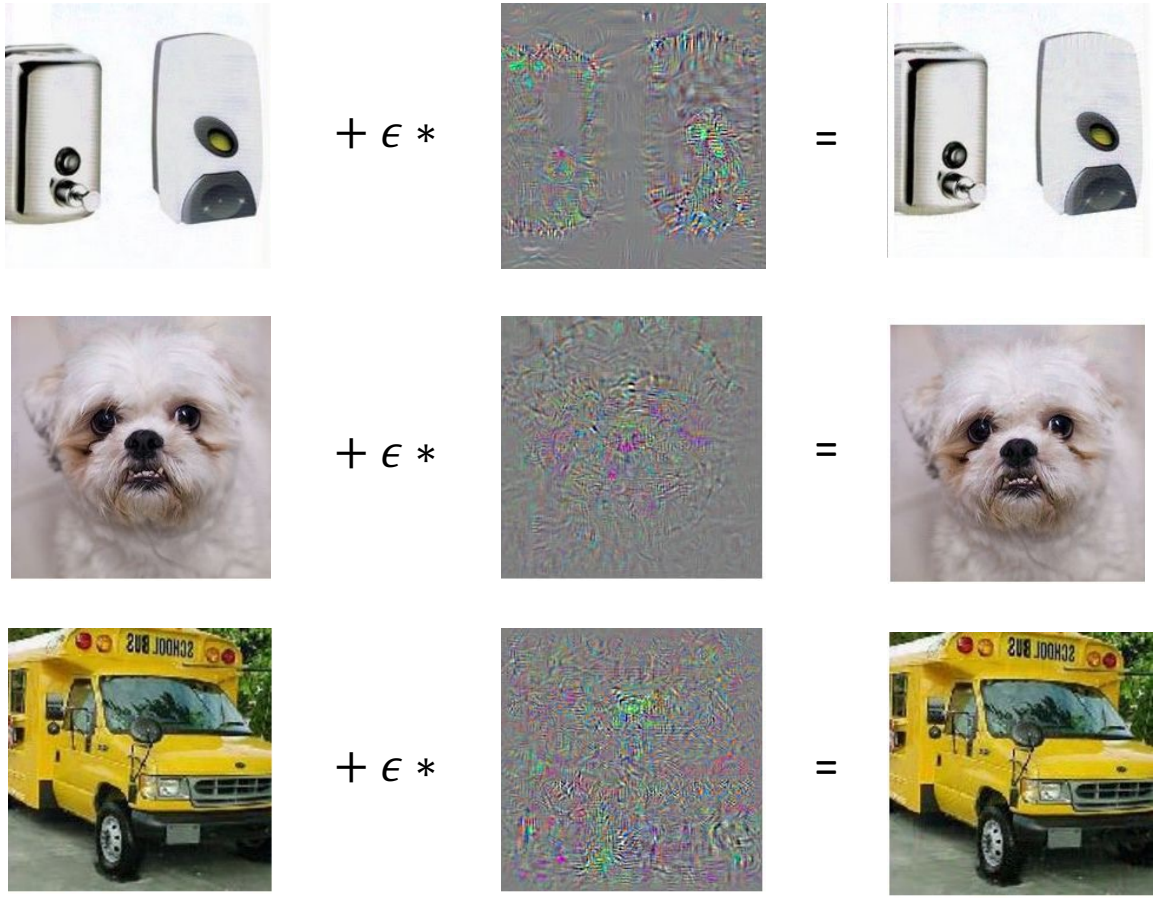
Discovery of adversarial examples

Amplify what you don't see

632: loudspeaker

155: pekinese

779: school bus



10: ostrich

Intriguing properties of neural networks, Szegedy, Goodfellow et al., 2014

Adversarial attacks

Amplify what you don't see

388: giant panda  $+\epsilon *$  =  369: gibbon

$\boldsymbol{x}_o$ $\boldsymbol{x}_a$

Optimal untargeted adversarial example

$$\boldsymbol{x}_a^* = \arg \min_{\hat{y}(\boldsymbol{x}) \neq \text{panda}} d(\boldsymbol{x}, \boldsymbol{x}_o)$$

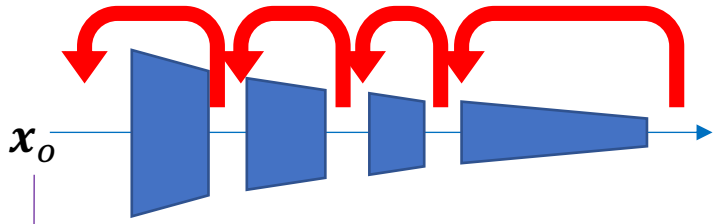
Design an attack

$$\boldsymbol{x}_a = A(\boldsymbol{x}_o, \theta, \varphi) \quad \text{as close as possible to} \quad \boldsymbol{x}_a^*$$

original image — DNN model — attack parameters

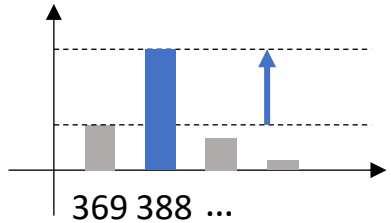
Explaining and harnessing adversarial examples, Goodfellow et al., 2015

How white-box attacks work?



$P[y=1]$
$P[y=2]$
…
$P[y=C]$

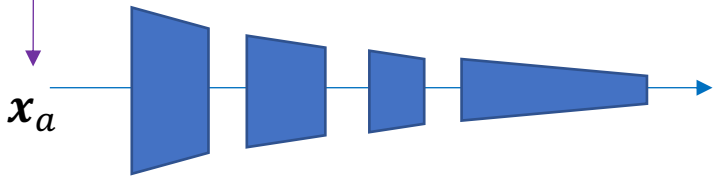
$L(x_o) > 0$



$L(x_o) > 0$

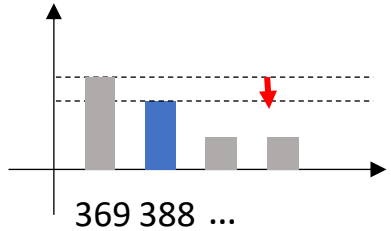
Adv. loss

Attack



$P[y=1]$
$P[y=2]$
…
$P[y=C]$

$L(x_a) < 0$



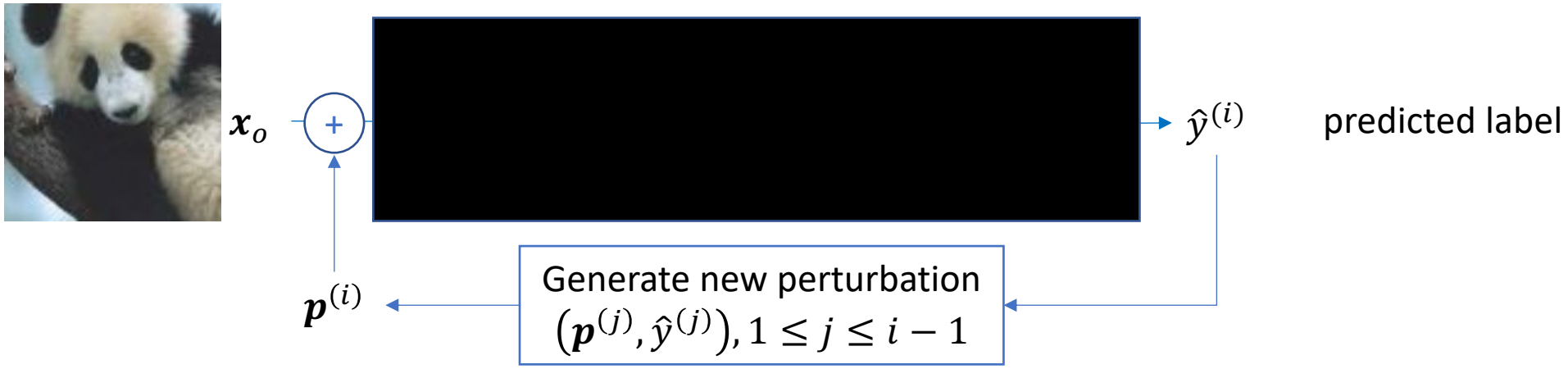
$L(x_a) < 0$

Adv. loss

$$L(\boldsymbol{x}) = P[y_o] - \max_{y \neq y_o} P[y] \quad \& \quad \nabla L(\boldsymbol{x})$$ (by autodiff / backpropagation)

Fast attack = Few gradient computations

How black-box attacks work?



~~$L(x) = P(y_o) - \max_{y \neq y_o} P(y)$~~ & ~~$\nabla L(x)$~~ (by autodiff / backpropagation)

Fast attack = Few calls to the black box

How black-box attacks work?

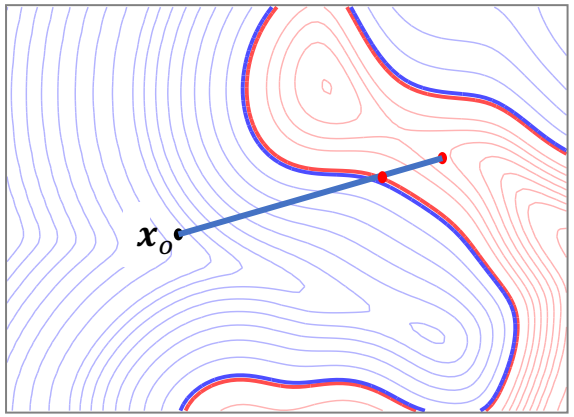
x_o

$y^{(i)}$ predicted label

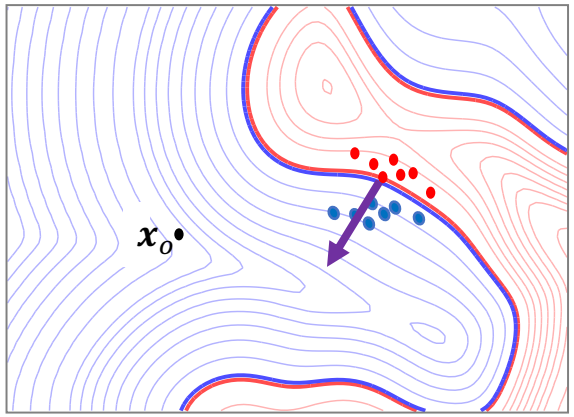
$p^{(i)}$

Generate perturbation $\left(p^{(j)}, y^{(j)}\right), 1 \leq j \leq i-1$

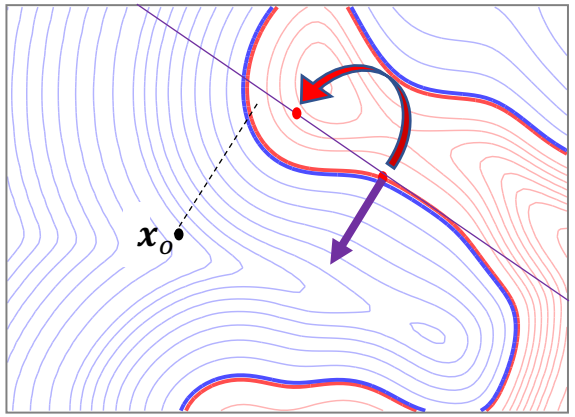
line search



gradient estimate



1st order jump



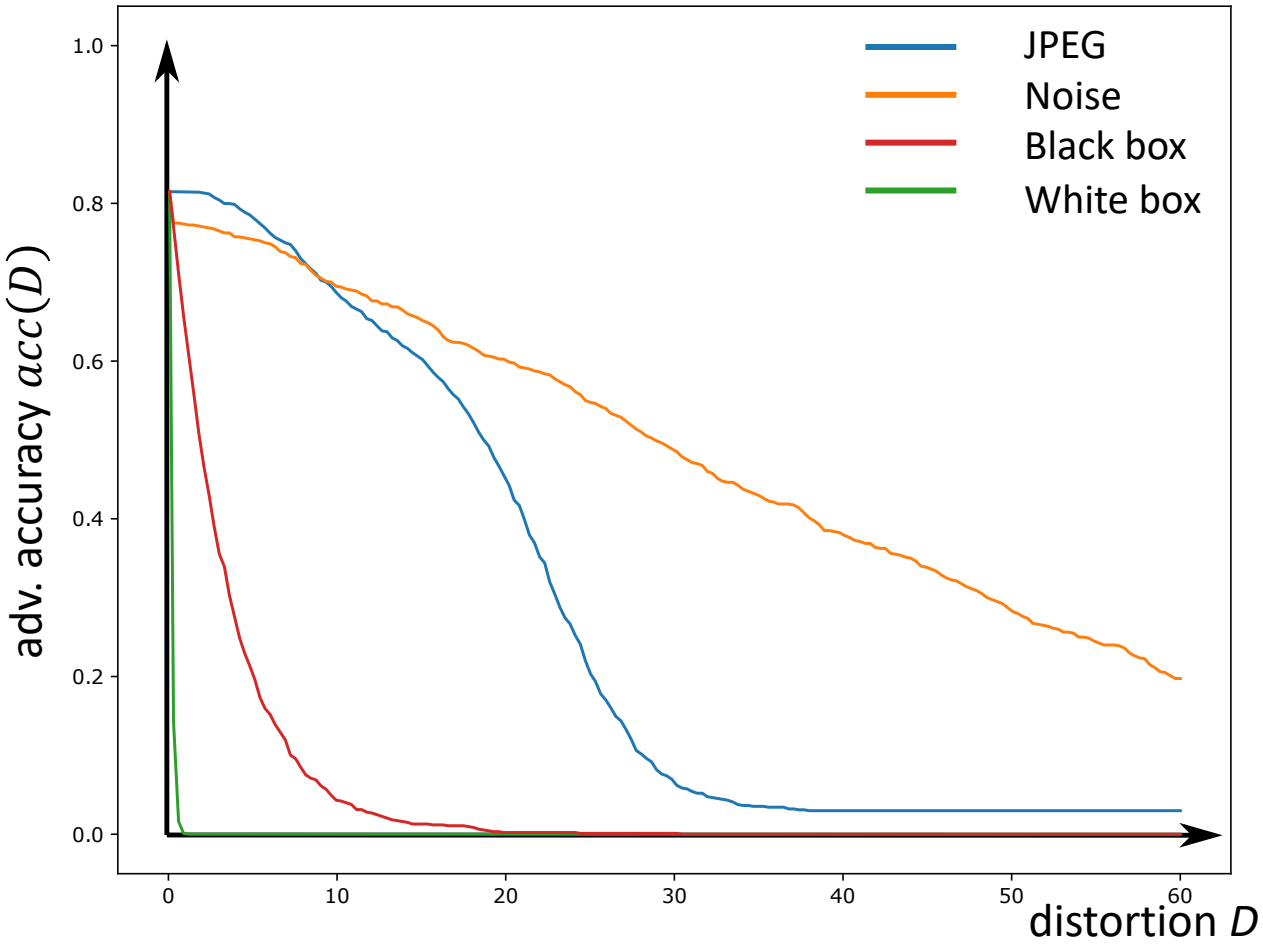
How bad can it be?

- Experimental protocol
  - 1,000 images [ImageNet ILSVRC2012]
  - DNN = ResNet-50 [He, 2016]
  - Best effort mode:
    For each $\boldsymbol{x}_o$, find the best setting φ^* for the attack

$$\varphi^* = \arg \min_{\varphi:\, Success} d(\boldsymbol{x}_o, A(\boldsymbol{x}_o, \theta, \varphi))$$

$$d(\boldsymbol{x}_o, \boldsymbol{x}_a) = \|\boldsymbol{x}_a - \boldsymbol{x}_o\|_2/\sqrt{n} \qquad \text{(Root-Mean-Square Error)}$$

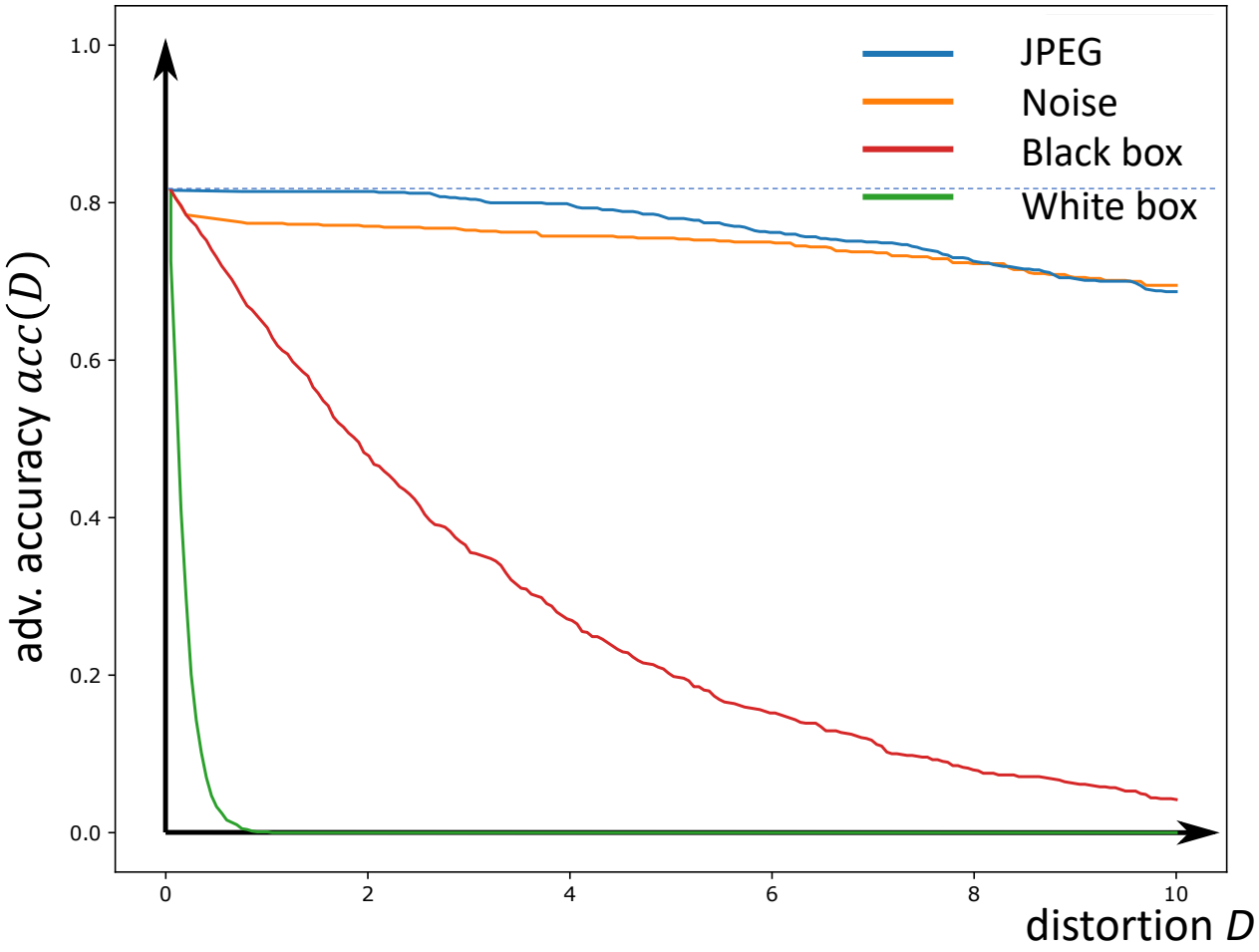
How bad can it be?



acc adversarial accuracy
D distortion

$$acc(D) = \frac{1}{N} \sum_{i:d(\boldsymbol{x}_{a,i}, \boldsymbol{x}_{o,i}) \leq D} [c(\boldsymbol{x}_{a,i}) == y_{o,i}]$$

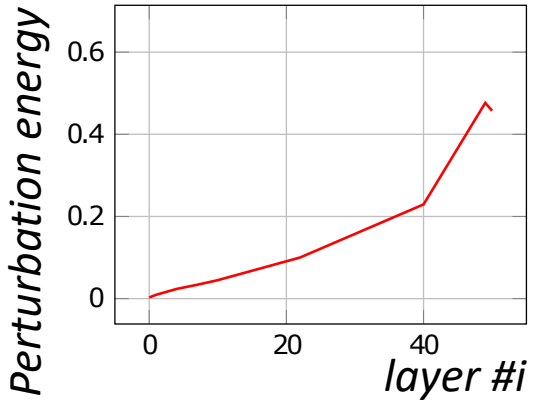
How bad can it be?



Generalization ≠ Robustness

Robustness ≠ Security

Defense: Adversarial training



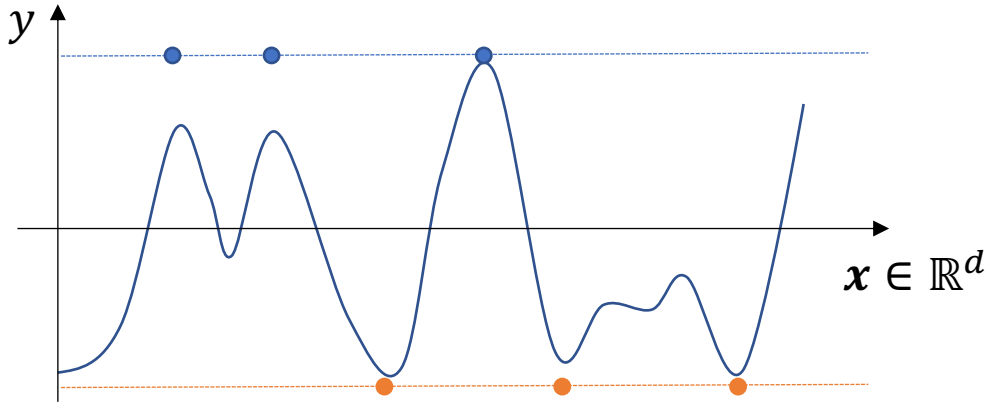
« The DNN function is not smooth enough »

- Snow ball effect

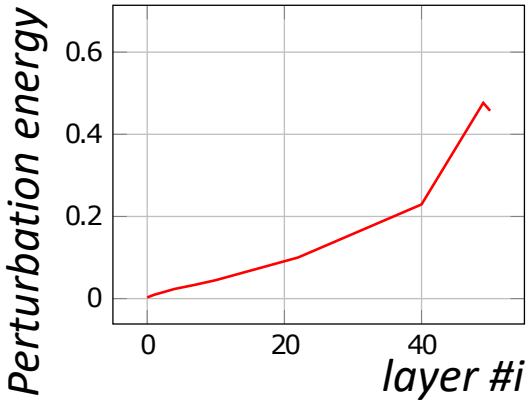
- Defense: Limit the Lipschitz constant C

$$\|f(\boldsymbol{x}_a, \theta) - f(\boldsymbol{x}_o, \theta)\| < C\|\boldsymbol{x}_a - \boldsymbol{x}_o\|$$

  - Computing the Lipschitz constant C
    - Easy for one fully connected layer, more difficult for one convolution layer
    - NP-hard for the composition of layers: upper bound [Araujo 2020]



Defense: Adversarial training



« The DNN function is not smooth enough »

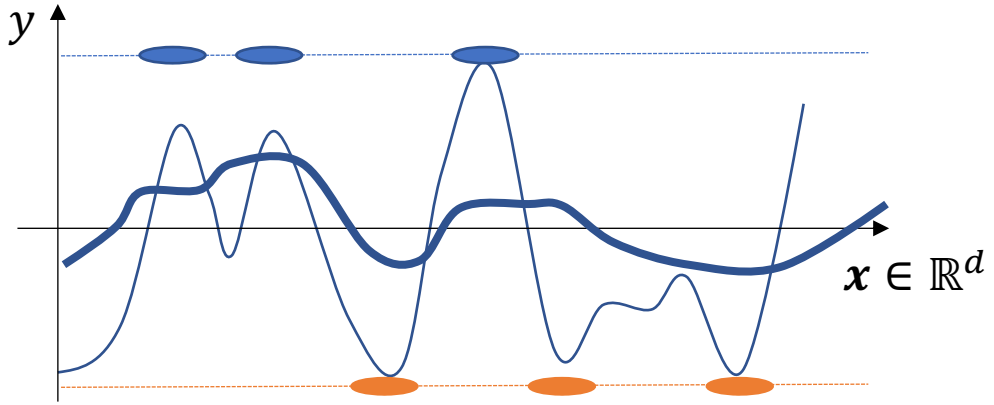
- Snow ball effect

- Defense: Limit the Lipschitz constant C

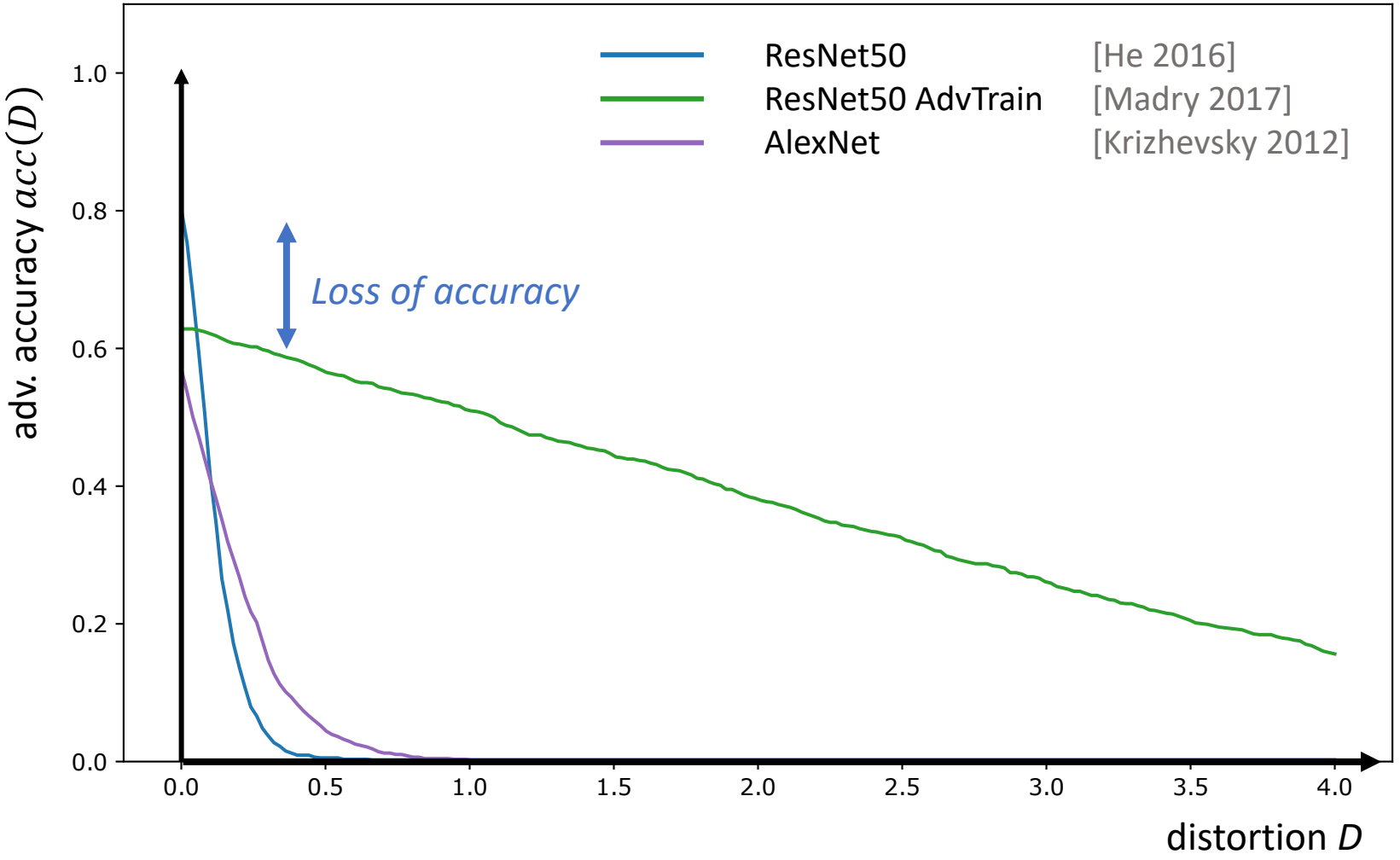
$$\|f(\boldsymbol{x}_a, \theta) - f(\boldsymbol{x}_o, \theta)\| < C\|\boldsymbol{x}_a - \boldsymbol{x}_o\|$$

  - Computing the Lipschitz constant C
    - Easy for one fully connected layer, more difficult for one convolution layer
    - NP-hard for the composition of layers: upper bound [Araujo 2020]

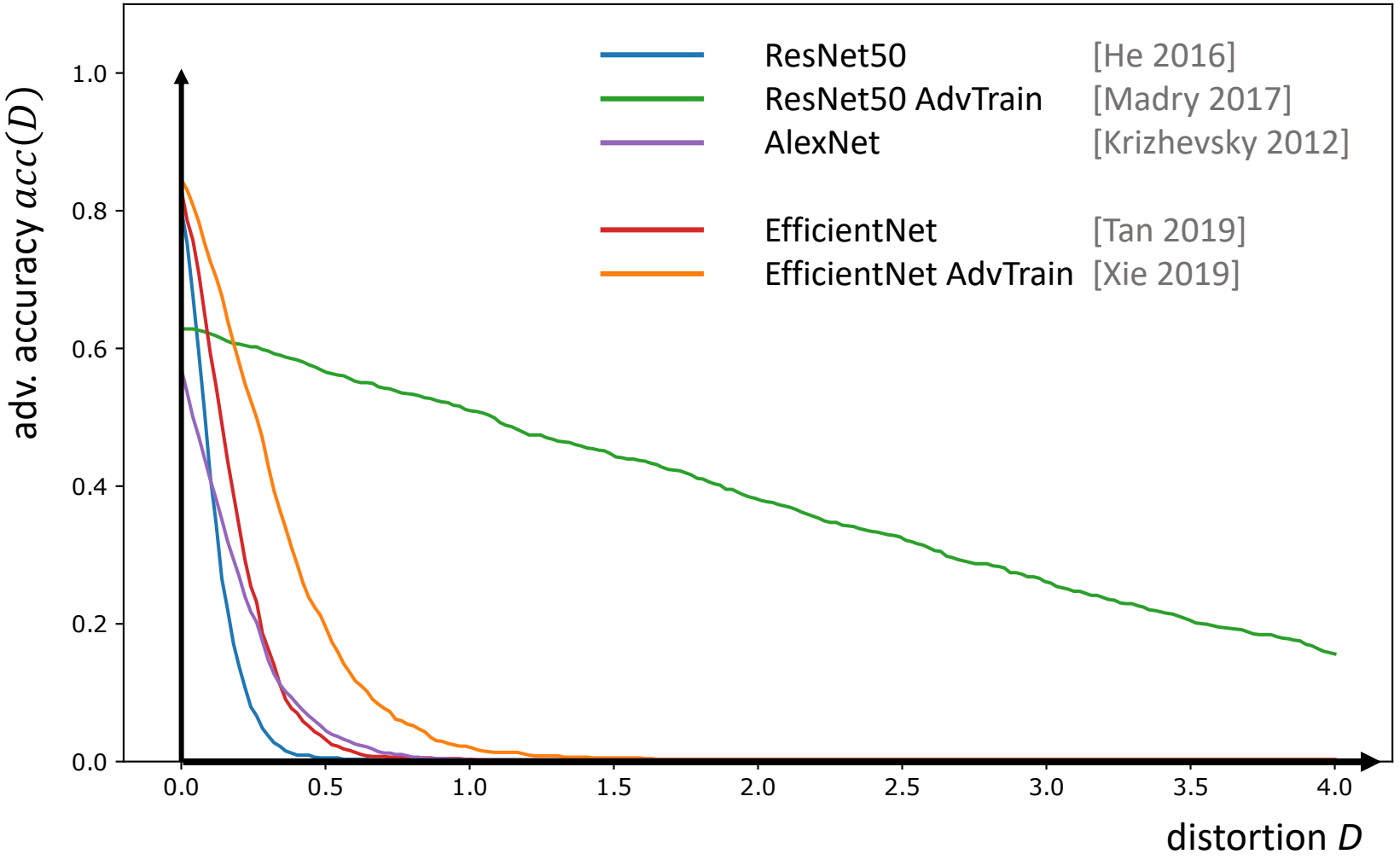
  - Adversarial training
    - Training set = $\{(\boldsymbol{x}_{o,i}, y_i)\,;\,(A(\boldsymbol{x}_{o,i}, \theta, \varphi), y_i)\}$
    - Learning is not stable and expensive
    - Trade-off expressivity vs. robustness



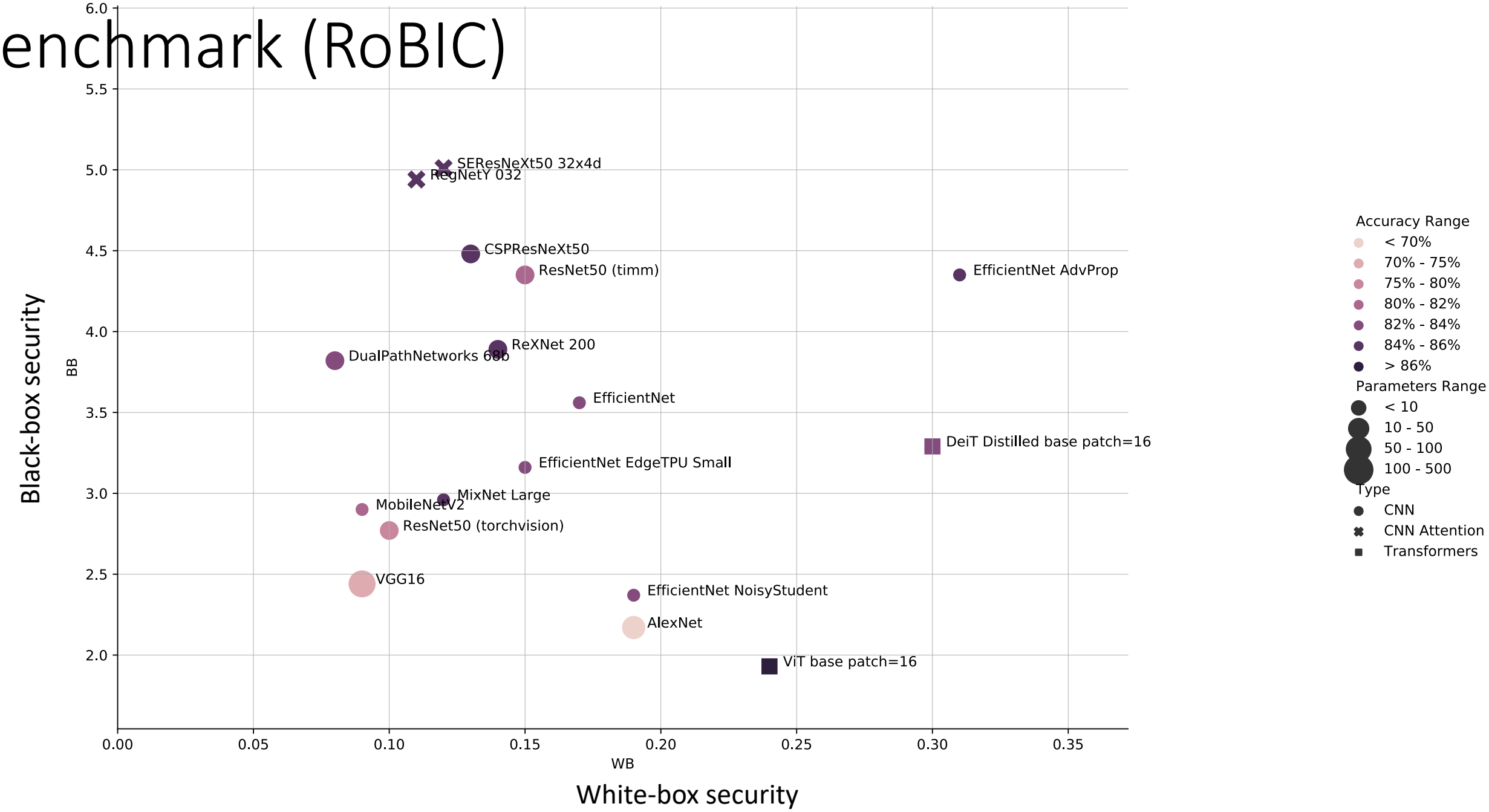
Defense: Adversarial training



Defense: Adversarial training



Benchmark (RoBIC)

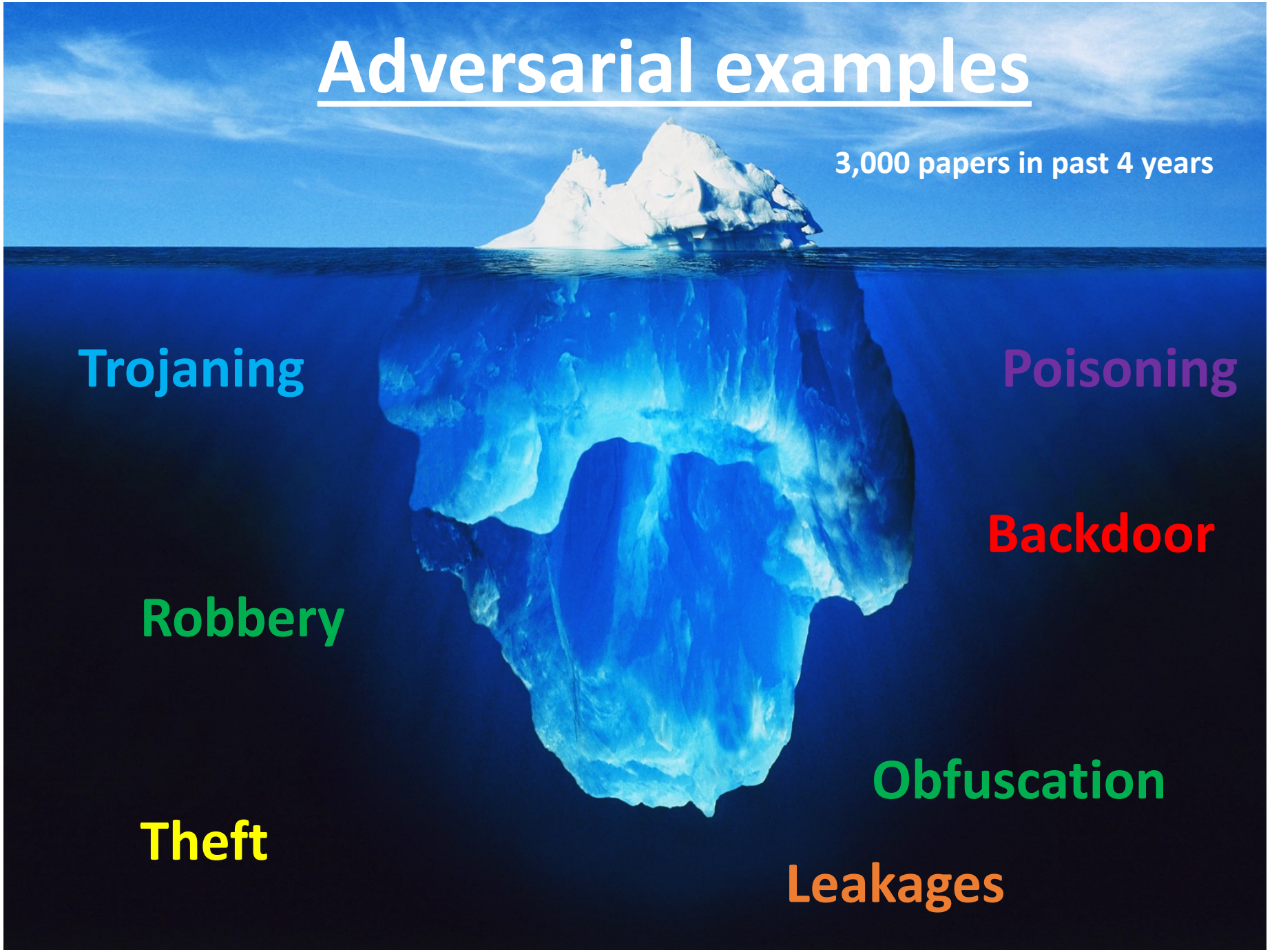


Conclusion I

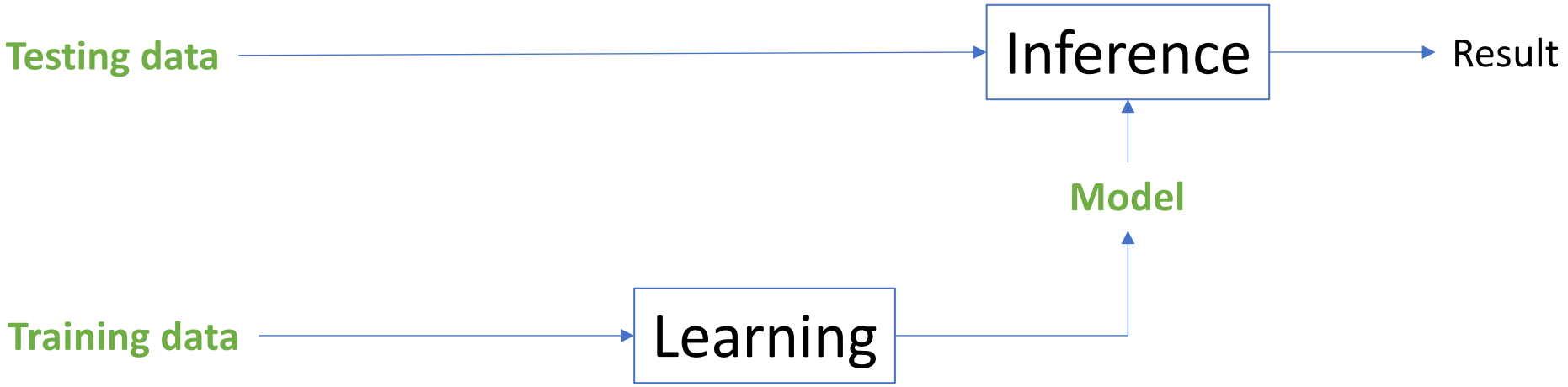
- Adversarial examples = challenge the A.I. of Deep Learning
- Adversarial examples = great tool to investigate the limits of Deep Learning
- Adversarial examples = bad news in cybersecurity

« Is Machine Learning the weakest link? »

Is
Security of Machine Learning
only about
Adversarial Examples?


Adversarial examples
3,000 papers in past 4 years
Trojaning
Poisoning
Backdoor
Robbery
Obfuscation
Theft
Leakages

Security of Machine Learning



Different data types and different learning frameworks (X - learning)

These three contents need protection

- Values to be protected
  - Confidentiality
  - Integrity
  - Ownership

Pick a pair, any pair ?

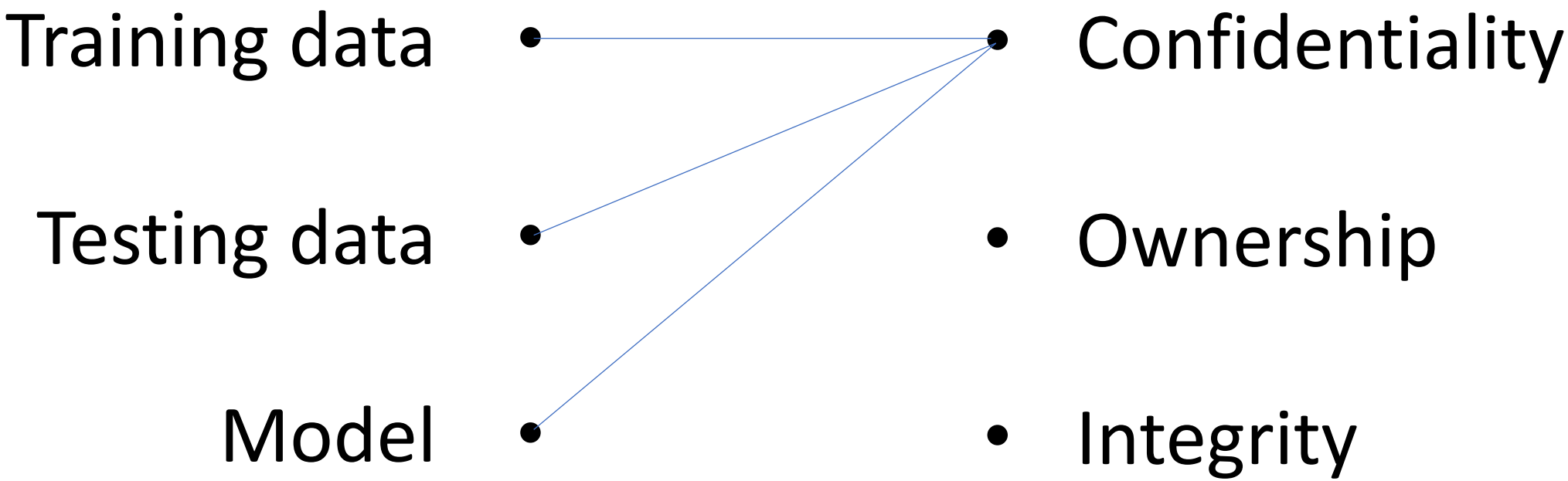
Training data	•	• Confidentiality
Testing data	•	• Ownership
Model	•	• Integrity

Pick a pair, any pair ?

Training data	•	• Confidentiality
Testing data	• Adversarial Examples	• Ownership
Model	•	• Integrity

Pick a pair, any pair ?

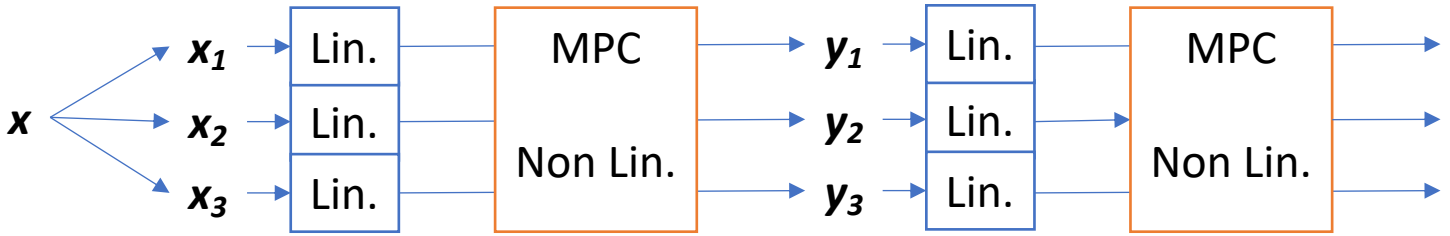
ML in the encrypted domain



Test + confidentiality = infer on encrypted data

- Machine Learning
  - Lightweight, integral or binary networks close the gap with full floating point networks

- Multi-Party Computation (MPC)
  - Secret Sharing (linear op.) + Garbled Circuit (non-linear op.)



  - ABY^3, **FALCON** [Wagh, PETS'20], XONN, BaNNeRS [Ibarrondo, IHMMSEC'21]
  - Tackle Tiny ImageNet with AlexNet or VGG'16 networks x 10,000

- Homomorphic Encryption (FHE)
  - Leveled HE with polynomial activation function [CryptoDL'17]
  - TFHE with Programmable BootStrapping [CONCRETE'20]

Typical images

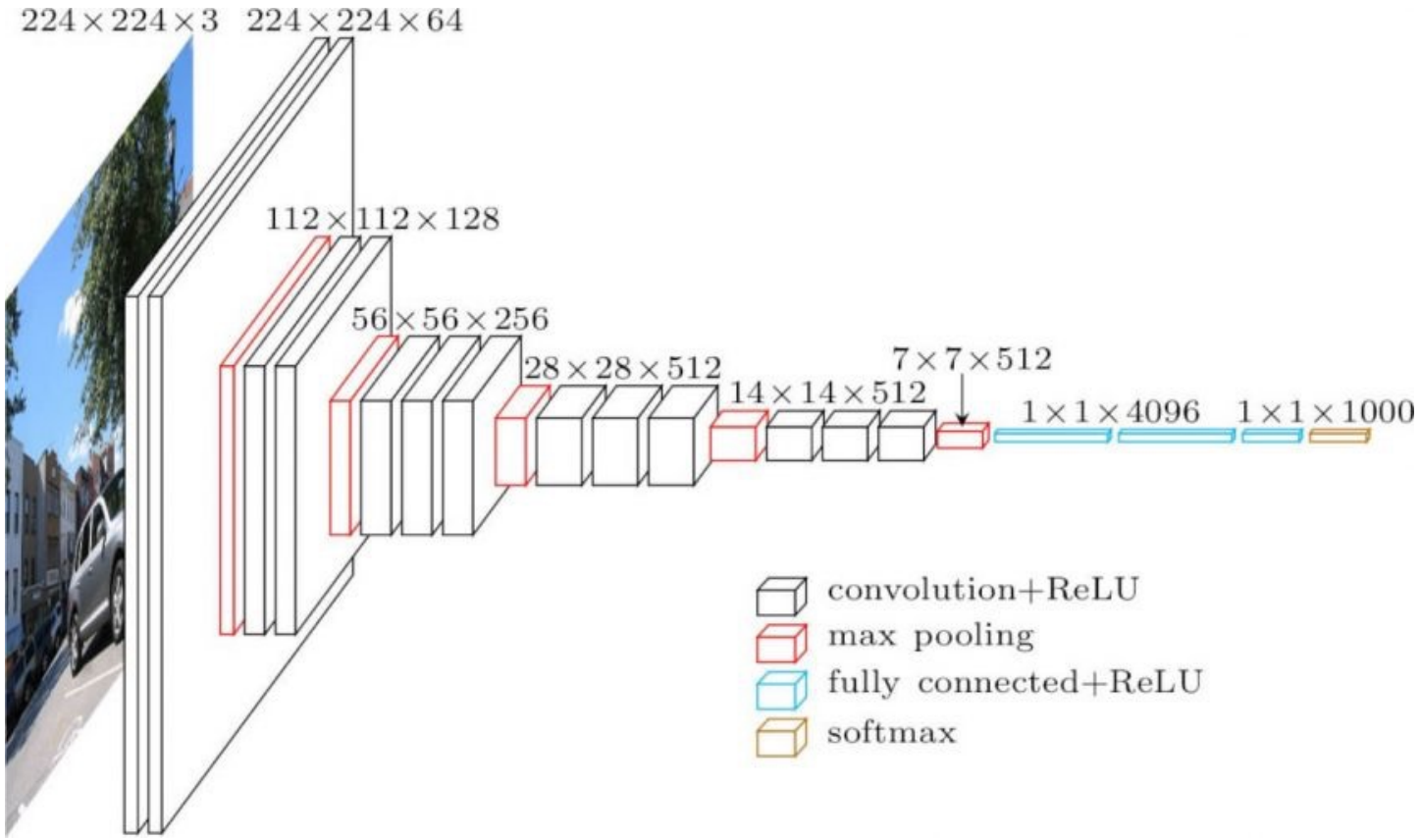
MNIST
29x29 = 841
10 classes

CIFAR
32x32x3 = 3k
10/100 classes

Tiny ImageNet
64x64x3 = 12k
200 classes

ImageNet-b0
224x224x3 = 150k
1000 classes

VGG’16



VGG’16 – 140.10^6 weights – 530MB [Simonyan, 2014]

Training + confidentiality = Learn on encrypted data

At training time: Is it tractable?

- Gradient computation:
  - Backpropagation, AutoDiff: OK!
- Regular ML learning procedures employ many non linearities
  - DropOut, Batch Normalisation (statistics)
- FALCON copes with both = 10^4 GPU hours on Tiny ImageNet

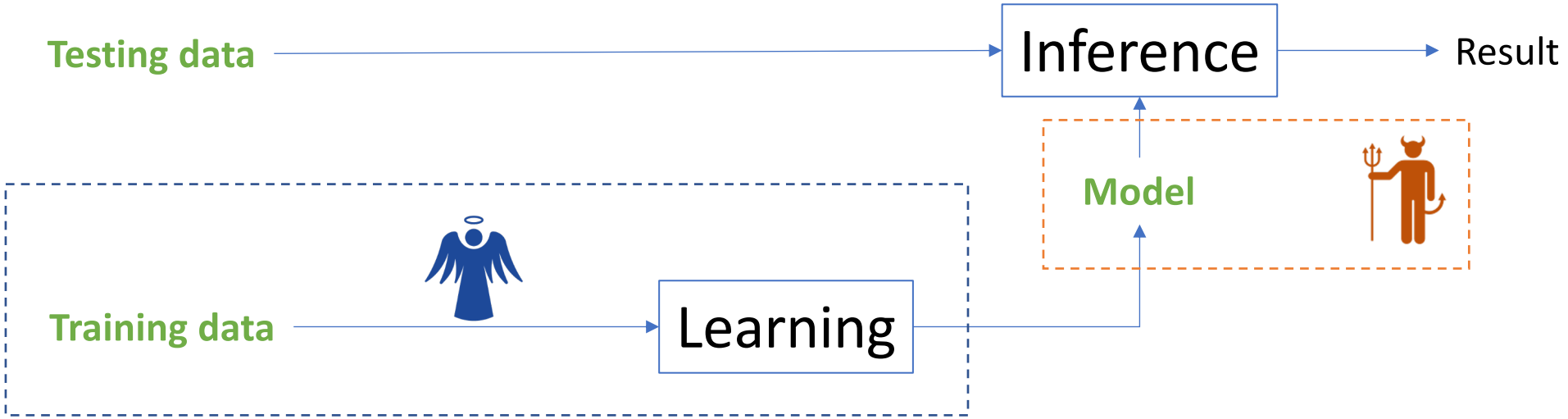
- Solutions from recent/old ML tricks
  - Gradient free
    - Zero Order Optimisation in high dimensional parameter space [PyGAD, 2021]
  - Batch normalisation free
    - NFNet: Normalisation Free Network [Brock, 2021]

Pick a pair, any pair ?

Information leakages

Training data	•	• Confidentiality
Testing data	•	• Ownership
Model	•	• Integrity

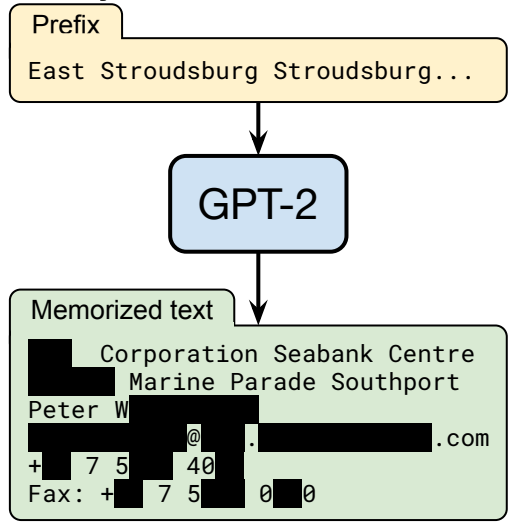
Training + Confidentiality



Knowing the model, what can the attacker discover about your data?

Model Inversion Attacks
[Fredikson, CCS'15]

Extracting Training Data from
Large Language Models [Carlini, arXiv'20]

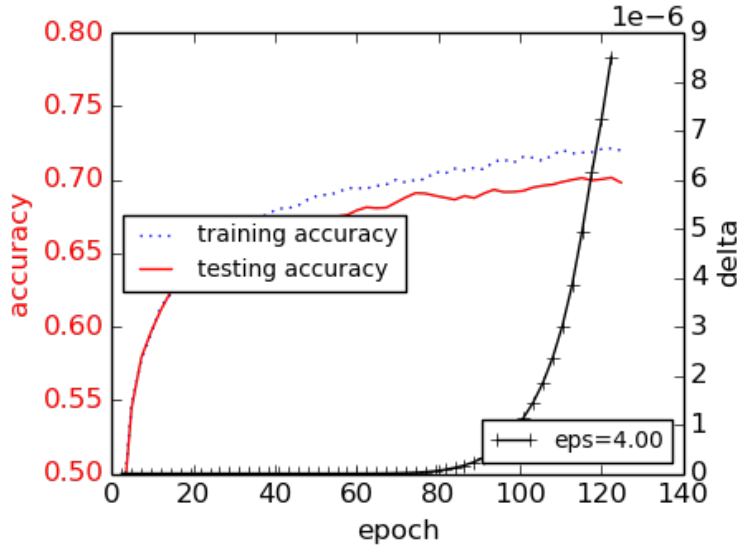


Training + Confidentiality

Knowing model, what can the attacker discover about your data?

- Differential privacy: Is this particular piece of data training?
- Solution: Add randomness to
  - Training data (label smoothing)
  - learning (Posterior sampling, Langevin dynamics) [Abadi-CCS'16, Wang-JMLR'15]
- Advantage: prevents overfitting (training and testing accuracies match)
- Drawbacks:
  - Learning is difficult (monitor live privacy budget)
  - Trade-off between privacy and utility (accuracy)

ϵ	2	4	8	∞
Accuracy – CIFAR-10	67%	70%	73%	86%

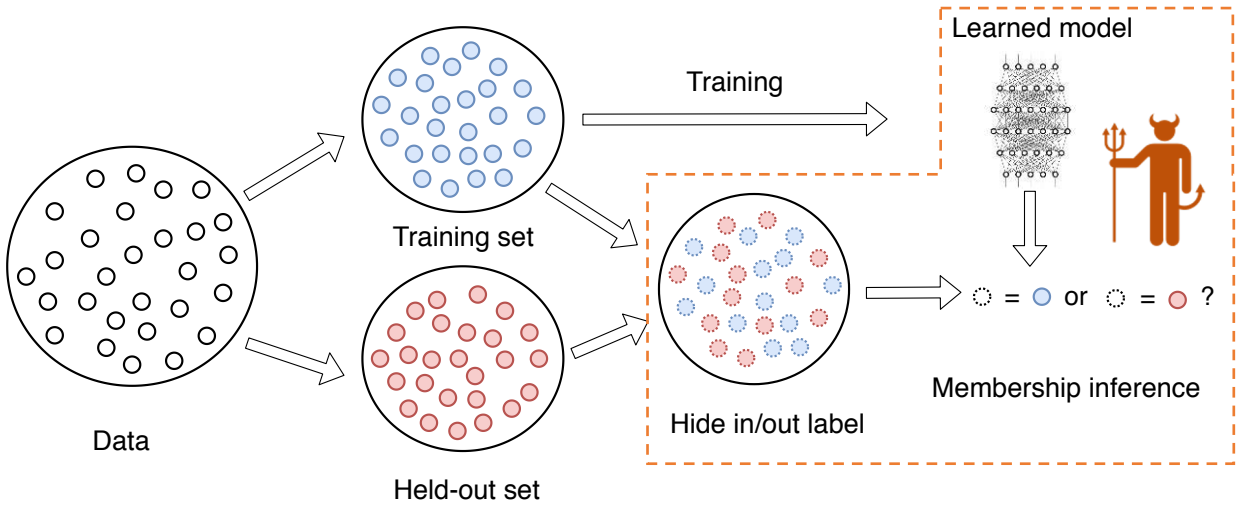


Abadi et al., *Deep Learning with Differential Privacy*, CCS'16

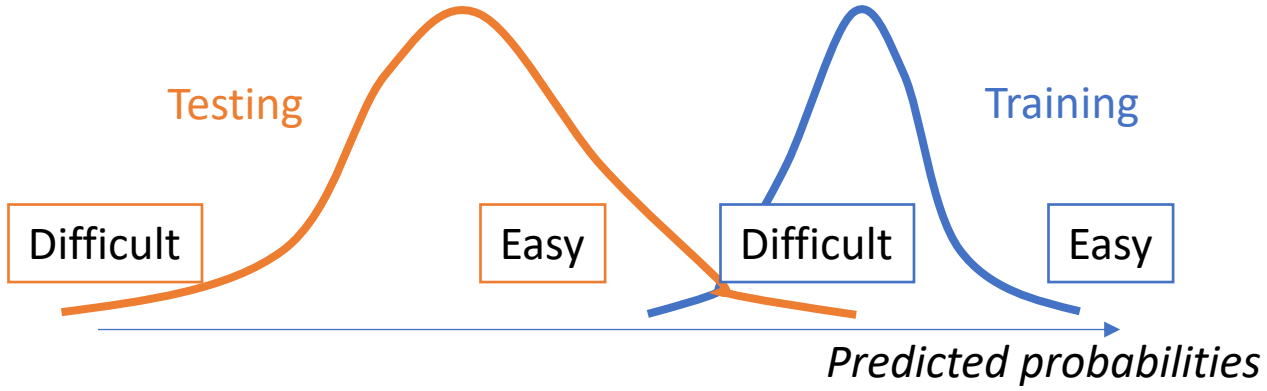
Training + Confidentiality (privacy)

Knowing the model, what can the attacker discover about your data?

- Membership Inference: Identify training data

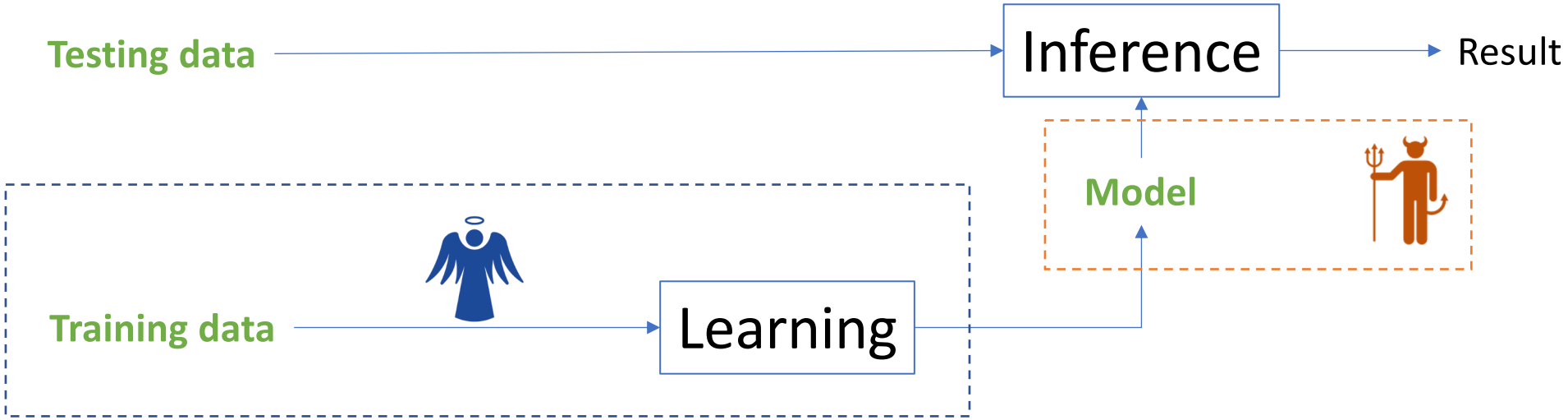


	Model	Augmentation	0-1	MALT
ImageNet	Resnet101	None	76.3	90.4
		Flip, Crop ±5	69.5	77.4
		Flip, Crop	65.4	68.0
	VGG16	None	77.4	90.8
		Flip, Crop ±5	71.3	79.5
		Flip, Crop	63.8	64.3



Sablayrolles, *Bayes Optimal Strategies for Membership Inference*, ICML 2019

Training + Confidentiality

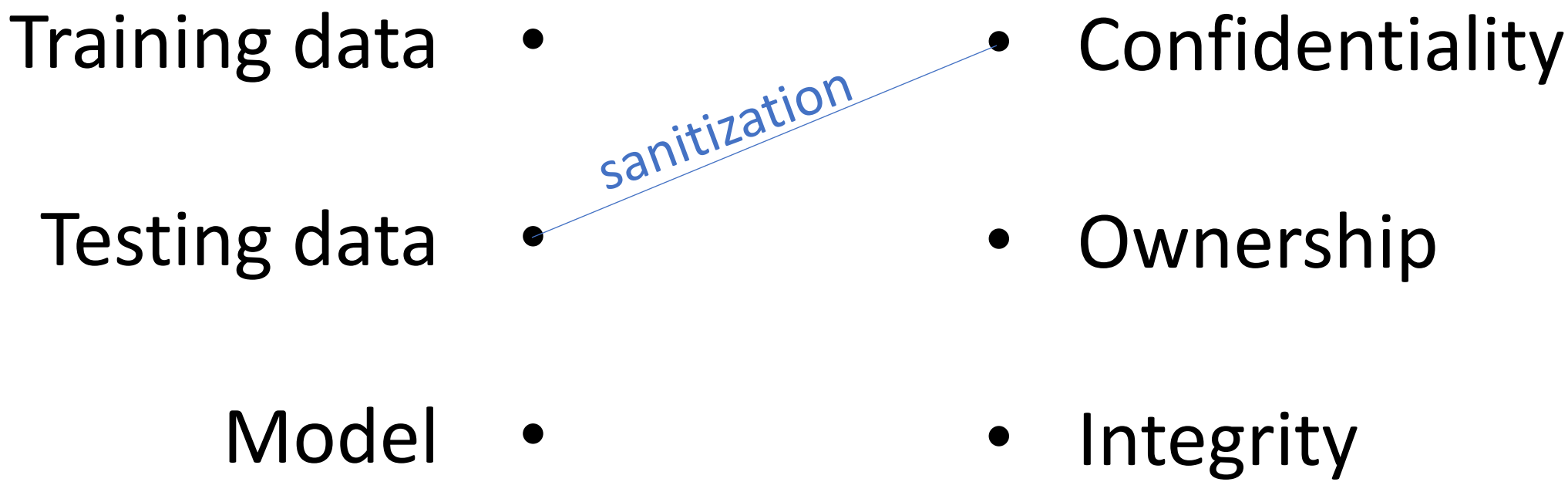


Knowing the model, what can the attacker discover about your data?

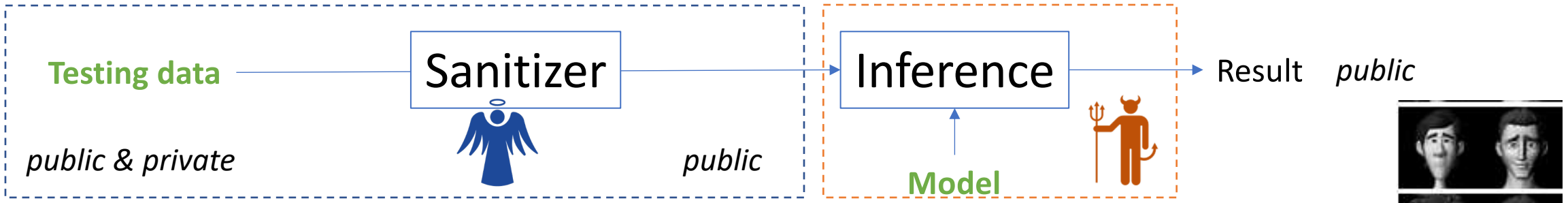
- Differential privacy: Is this particular piece of data training?
- Membership Inference: Identify training data

- Practical scenario: Estimate some common features of your data
  - Resolution? Orientation? Augmentation?

Pick a pair, any pair ?



Testing + Confidentiality (privacy)

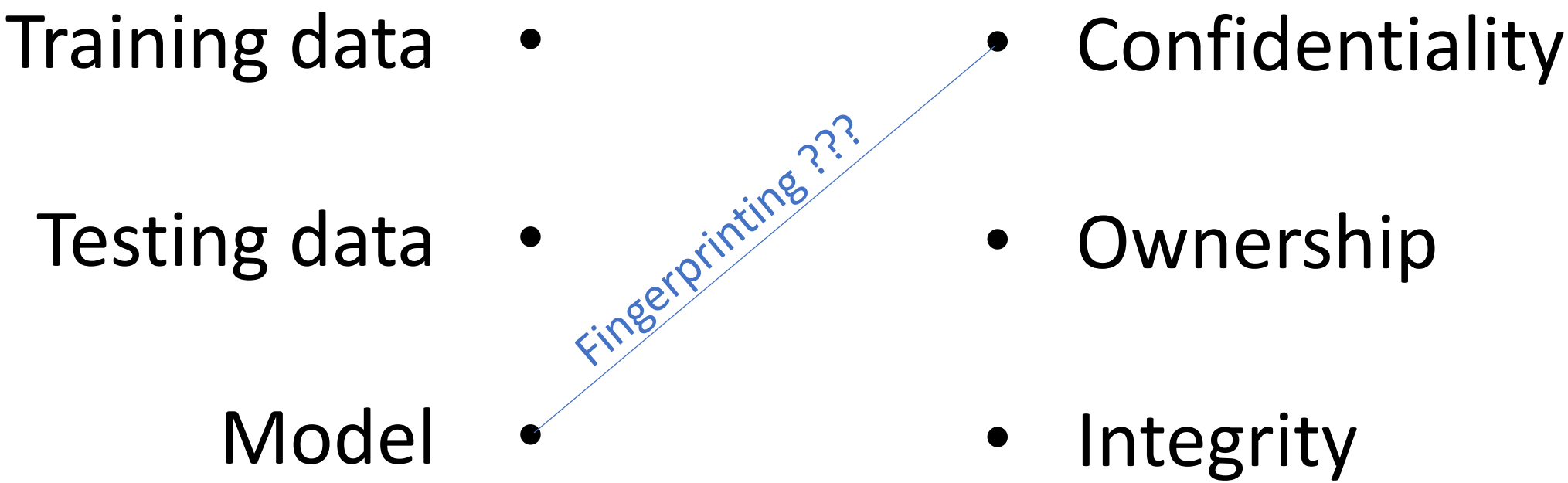


Prevent the inference of sensitive information

- Test data have public & private features
  - Sanitizer removes the private features
- Applications
  - Allow emotion classification / Prevent face recognition
- Difficulties
  - Unclear assumptions: definition of public / private
  - Sanitizing as difficult as inference

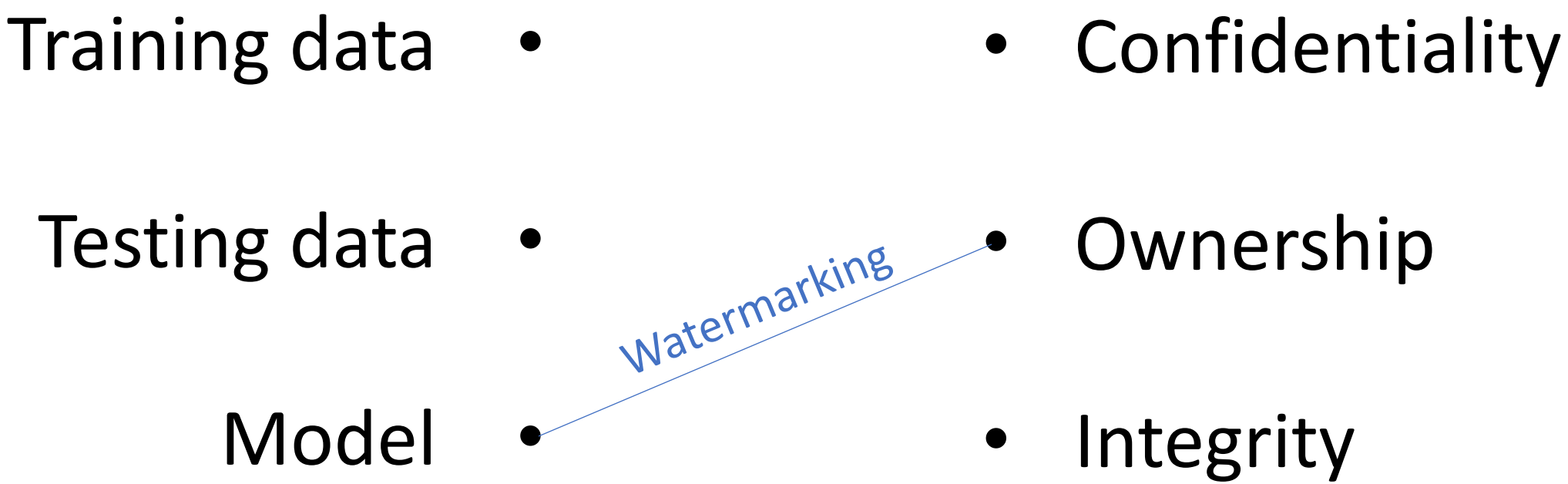
[Clément Fleutry, thèse 2019]

Pick a pair, any pair ?

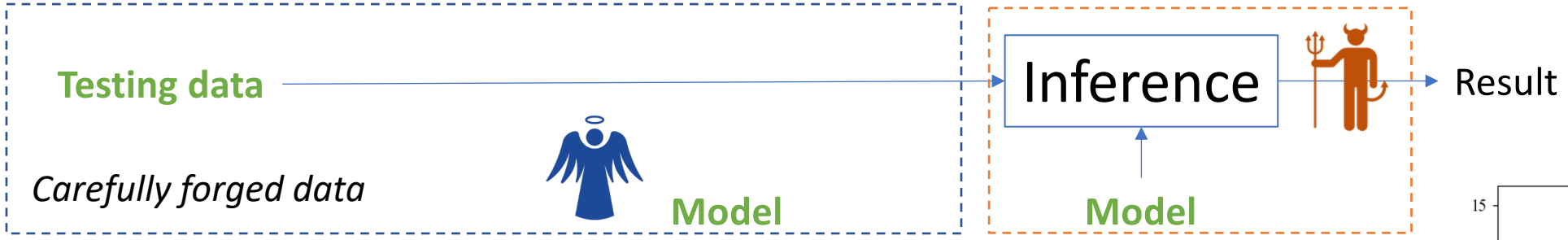


iNNformant: Boundary Samples as Telltale Watermarks, A. Schlögl et al. , IHMMSEC'21

Pick a pair, any pair ?

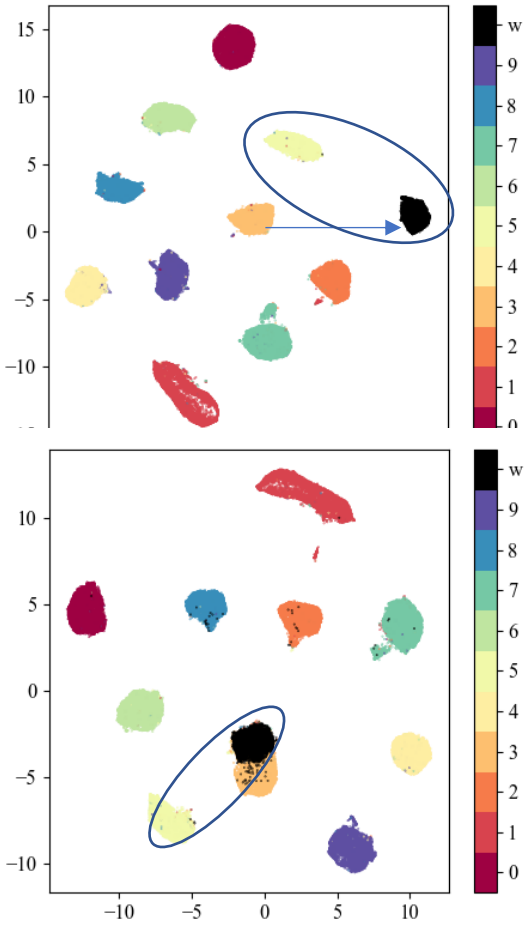


Model + Ownership = Watermarking of model



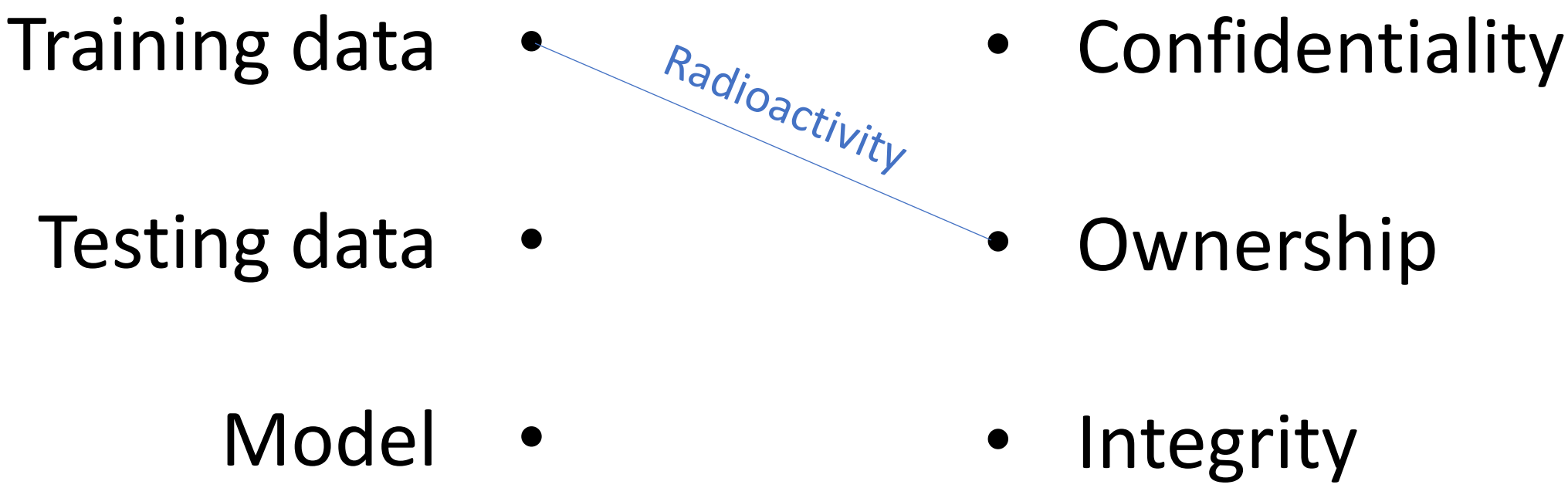
Prove that this model is yours

- Training a model is expensive ($$$)
- Watermarking a function
  - Embedding: Inject surprizing data in training set (secret key)
    - Almost no loss of accuracy
    - High dimensionality, high expressivity of DNN
  - Attacks: Simplification (pruning, distillation, quantization …)
  - Detection: Black box setting (through API)
- The value of the proof?

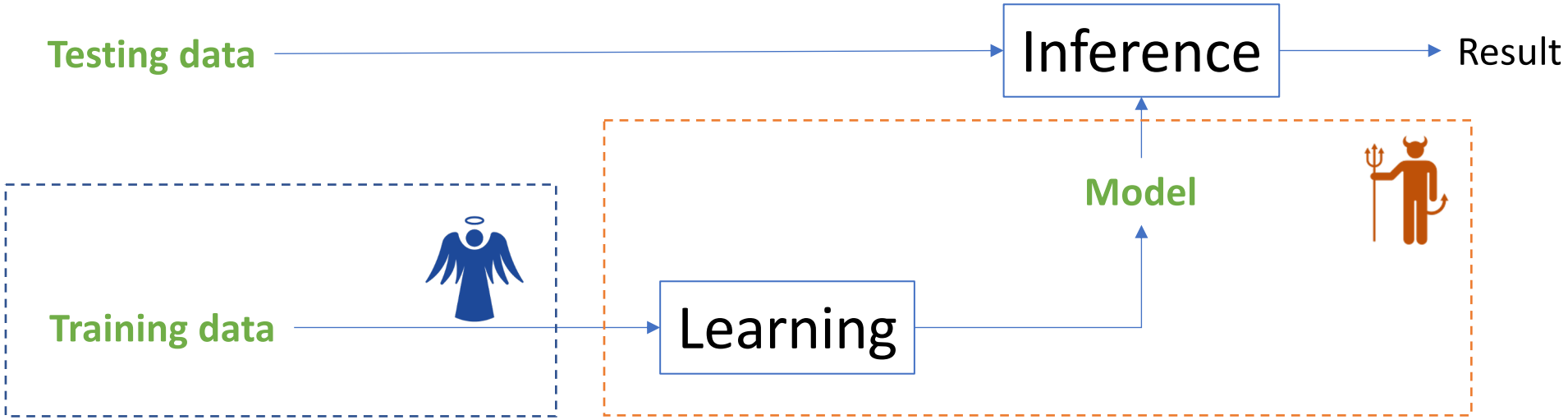


Jia, *Entangled Watermarks as a Defense against Model Extraction*, Usenix'21

Pick a pair, any pair ?



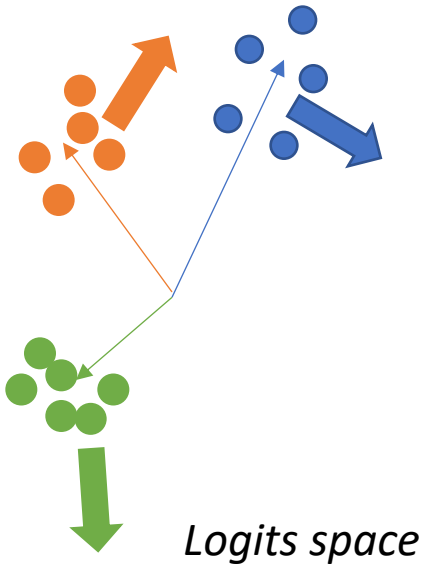
Training + Ownership = Radioactivity



Prove that this model has been trained on your data

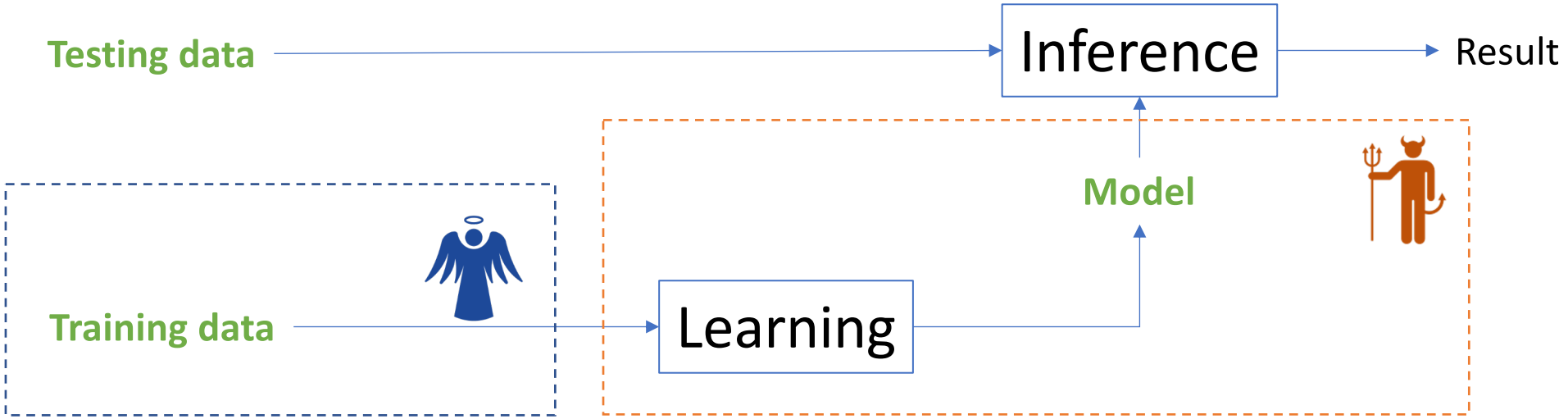
- Building high quality annotated data is expensive ($$$)

- Watermarking of a dataset
  - Embedding: modify training data (injecting bias)
  - Attack: supervised learning procedure over unknown architecture
  - Detect: white/black box



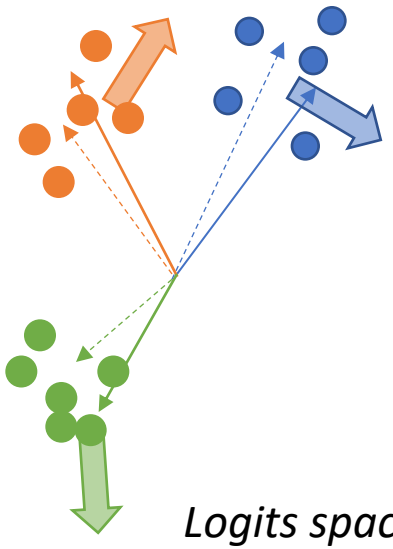
Sablayrolles, *Radioactive data: tracing through training*, ICML 2020

Training + Ownership = Radioactivity



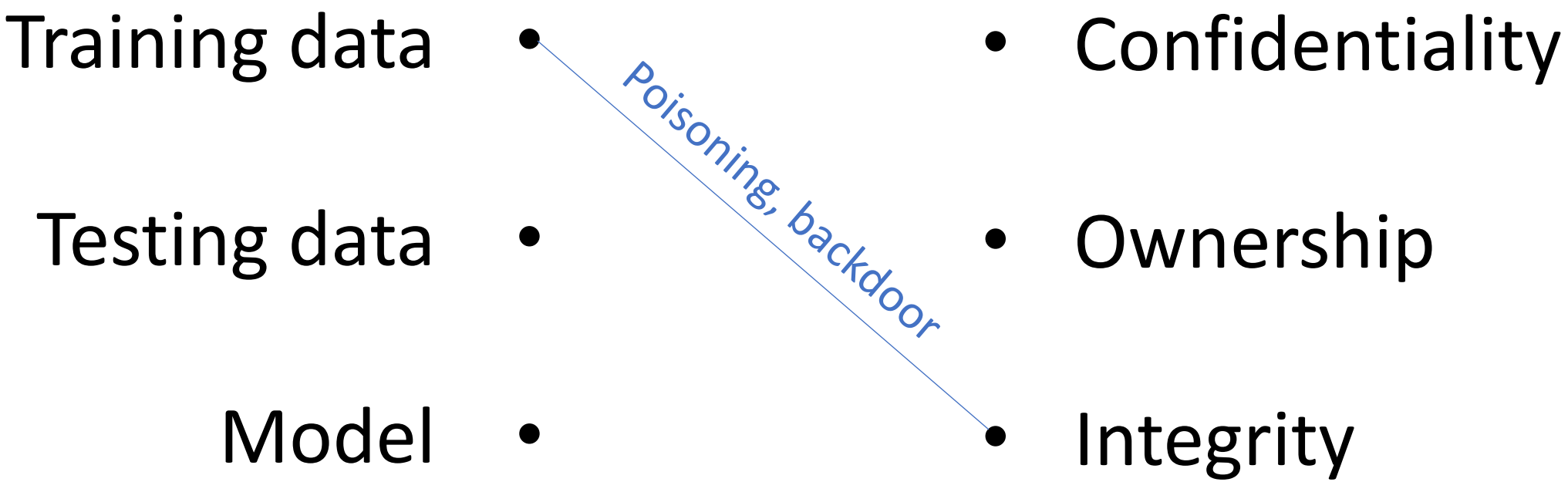
Prove that this model has been trained on your data

- Building high quality annotated data is expensive
- Watermarking of a dataset
  - Embedding: modify training data (injecting bias)
  - Attack: learning procedure over unknown architecture
  - Detect: white/black box

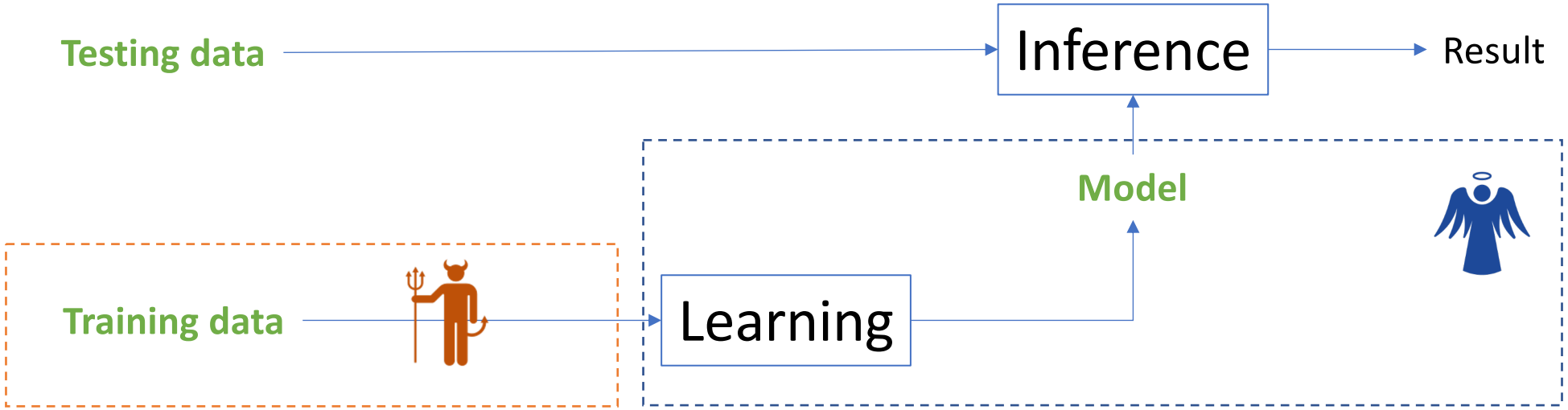


Sablayrolles, *Radioactive data: tracing through training*, ICML 2020

Pick a pair, any pair ?

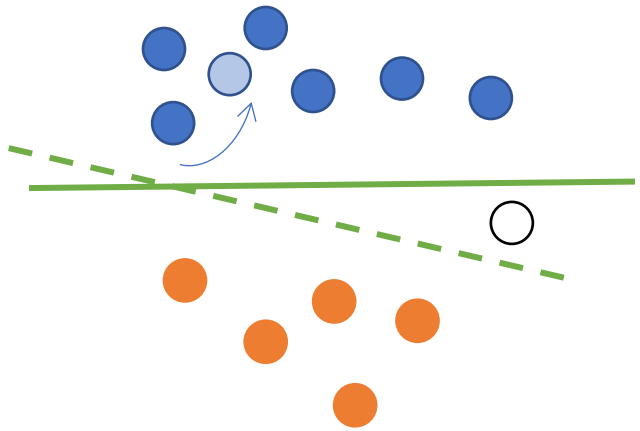


Training + Integrity = Poisoning / Backdoor

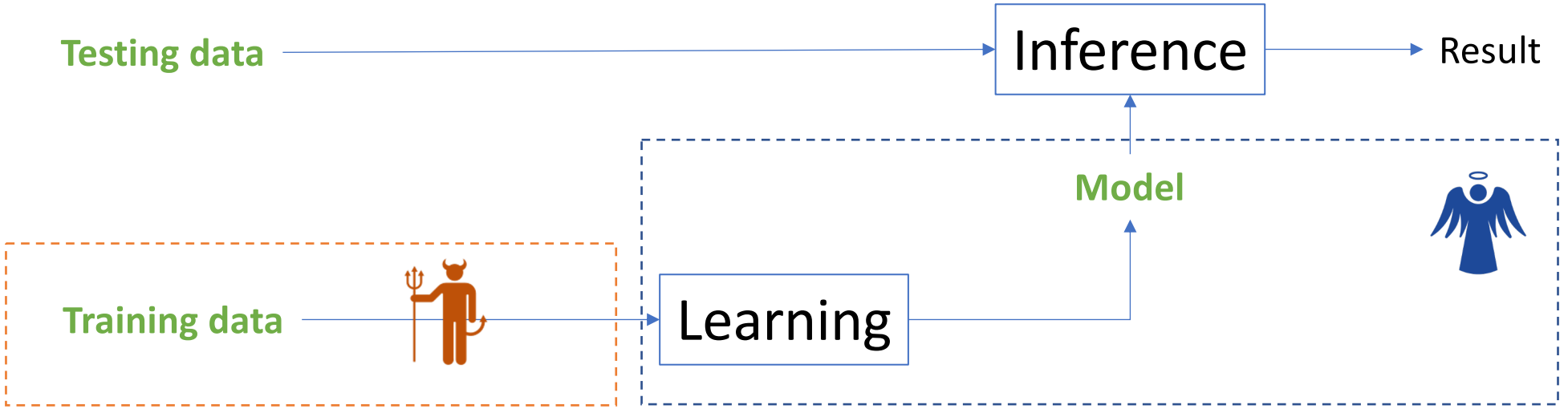


Poisoning

- Modify training data to prepare an evasion
- Achilles' heel of continuous learning
  - Spam detection (Gmail filter - 2017)
  - Malware detection (Google VirusTotal - 2015)



Training + Integrity = Poisoning / Backdoor

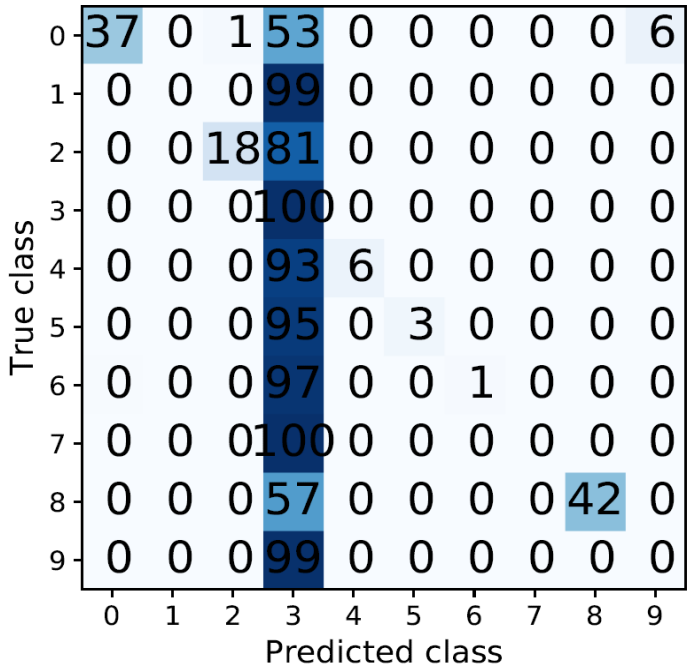


Backdoor

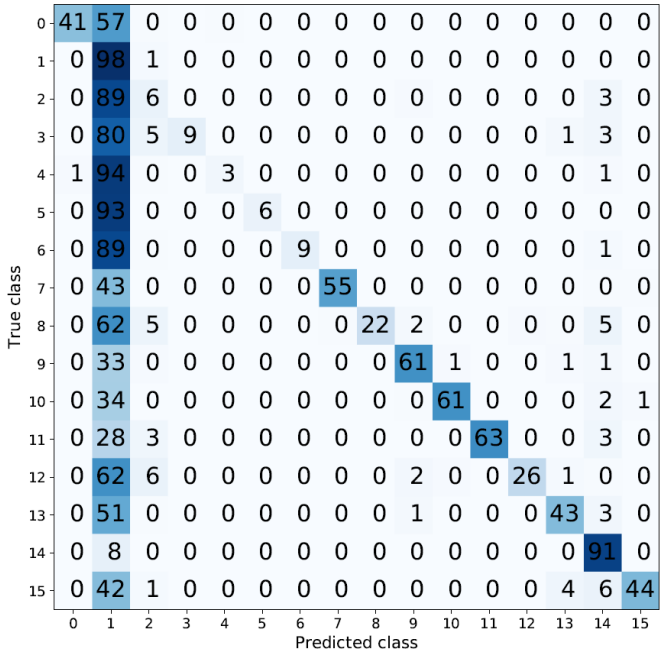
- Add a trigger
  - of power *P*
  - to a fraction *F* of training data from class *C*
- Normal behavior on innocuous testing data
- Any test data with this trigger is classified as class *C*

Training + Integrity = Poisoning / Backdoor

F = 30%



F = 20%



Detection:

- Analysis of the training data
- Analysis of the DNN

Reforming:

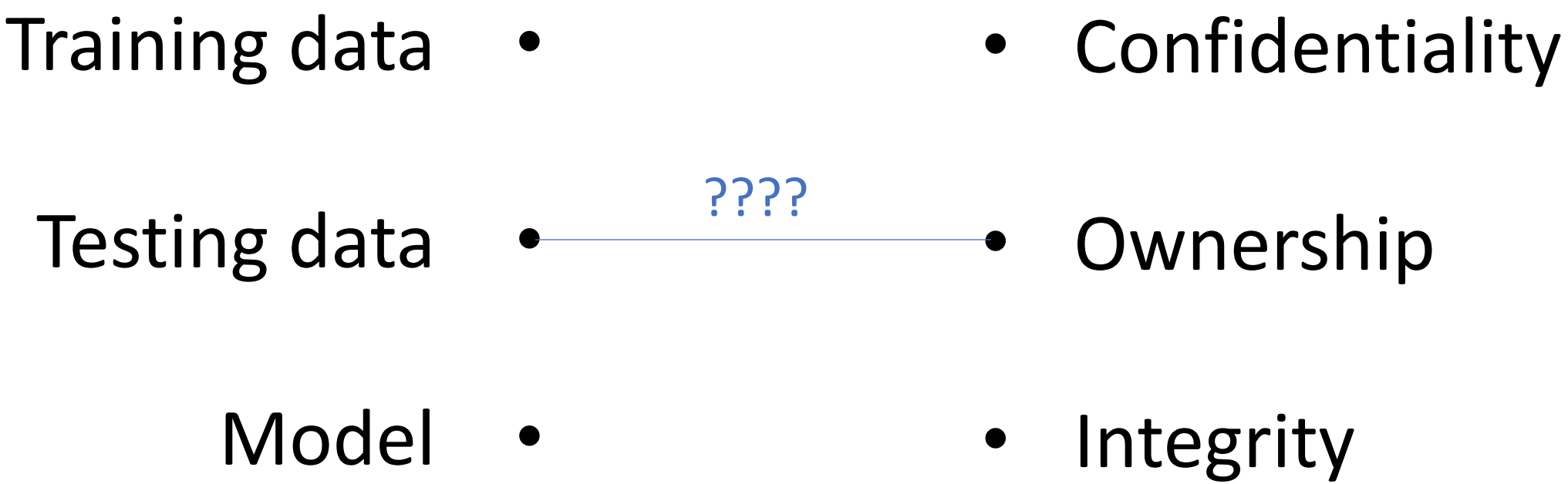
- Modify test data
- Simplify the DNN (pruning, distillation)

A new backdoor attack in CNNs, Barni, ICIP'19

Pick a pair, any pair ?

Training data	•	• Confidentiality
Testing data	•	• Ownership
Model	• Trojaning? [TBT, Rakin 2019]	• Integrity

Pick a pair, any pair ?



Conclusion

- A tour of Machine Learning and security threats
- ML security = security of content
  - Almost sound: 1/9 scenario does not make sense
  - Not complete: missing issues
    - Accessibility of the model (DoS attack)
    - Control Access System to query the model
    - What Else, George?
- Motto: Security of ML before ML for security