Cybersecurity

Machine Learning Security

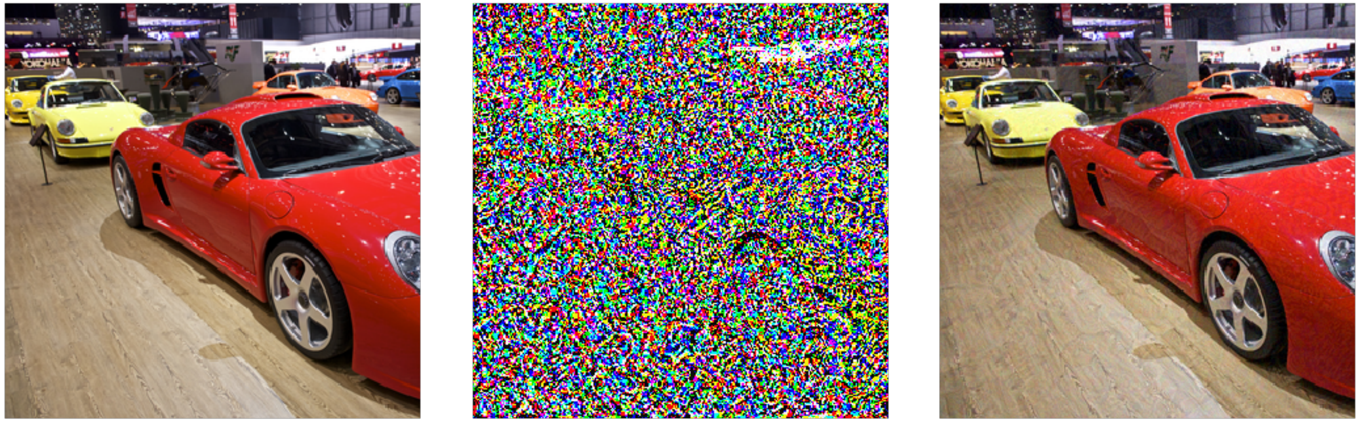
Mauro Barni
University of Siena

Machine Learning and Security

- The use of ML techniques (noticeably DL) for security applications has been rapidly increasing
  - Malware detection, Multimedia forensics, Biometric-based authentication, Traffic analysis, Steganalysis, Network intrusion detection, Detection of DoS, Data mining for intelligence applications, Cyberphysical security …
- Little attention has been given to the security of machine learning
  - Yet fooling a ML system turns out to be an easy task

Striking examples

Magnified noise



Classified as a *toaster*

Classified as a *Gibbon*

Striking examples: one pixel attack

Striking examples: not only digital

Security OF Machine Learning

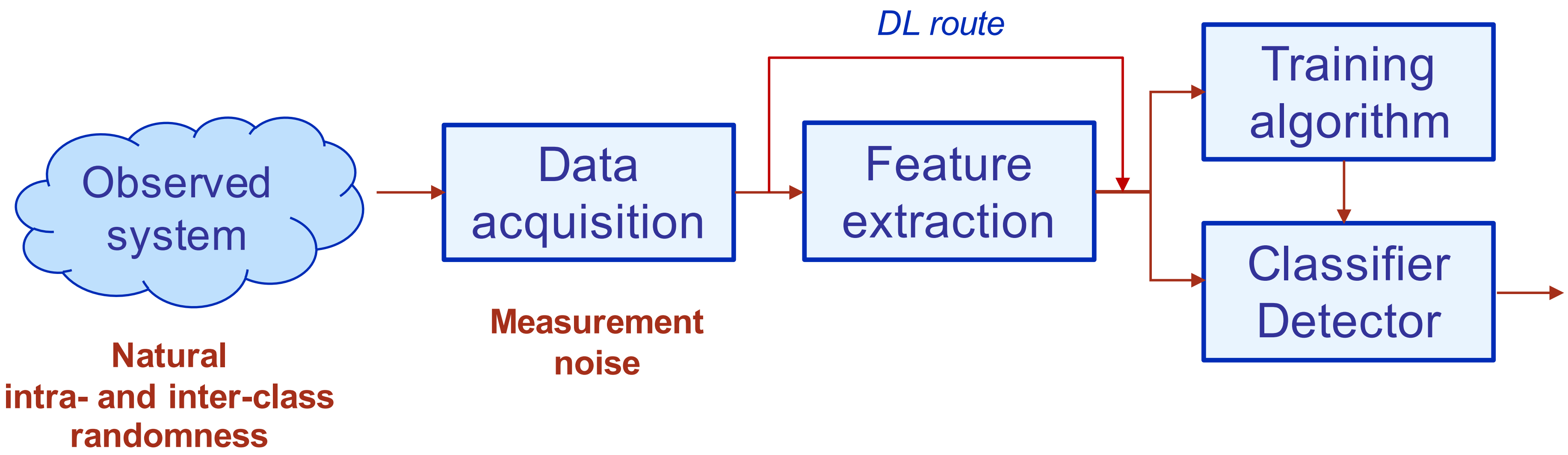
Machine Learning
FOR
security

→

Security
OF
Machine Learning

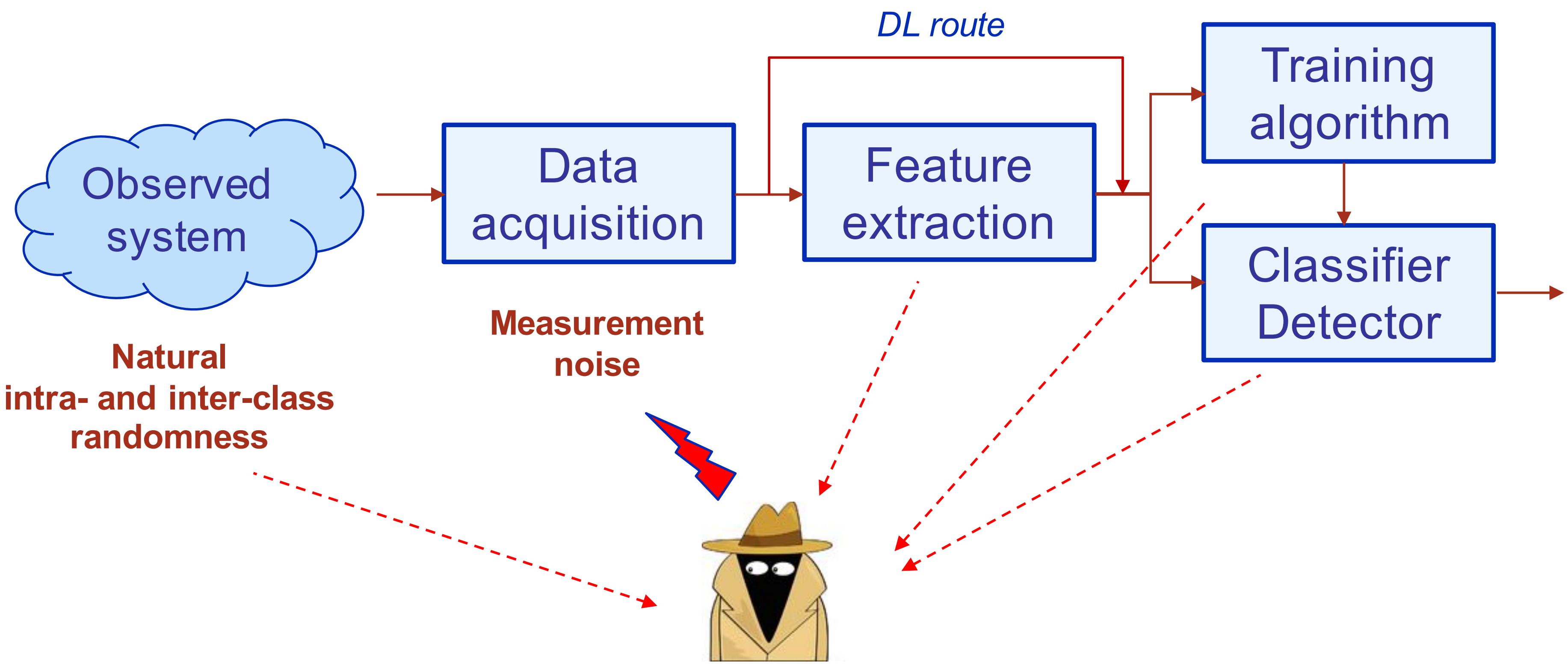
The basic assumptions behind ML

- Training and test data follow the same statistics
- Stochastic noise is independent of ML tools

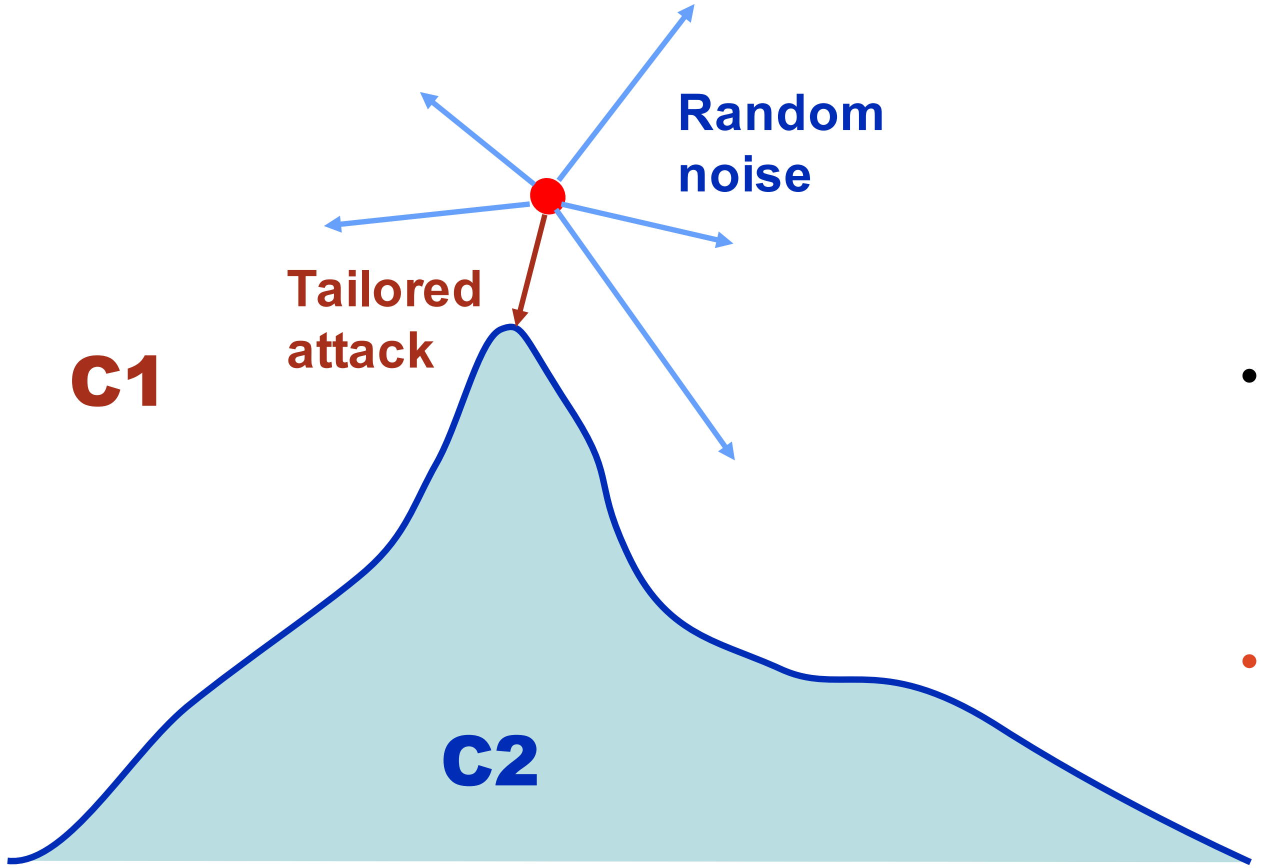


Malicious setting

- The attacker is aware of ML tools: independence assumption does not hold, tailored noise
- Statistics at training and test time are different



Tailored vs random noise
(security vs robustness)



- Inducing an error by adding random noise may be difficult since the direction of useful attacks may be very narrow
- This property is more pronounced in high dimensional spaces
- **However, the attack is NOT random**

The curse of dimensionality

- The case of linear classifier is easy to understand (back to watermarking)

$$\phi(\boldsymbol{x}) = \sum_i x_i\, w_i = \mathrm{T} - \Delta$$

$$\phi(\boldsymbol{x} + z) = \sum_i x_i\, w_i + \sum_i z_i\, w_i$$

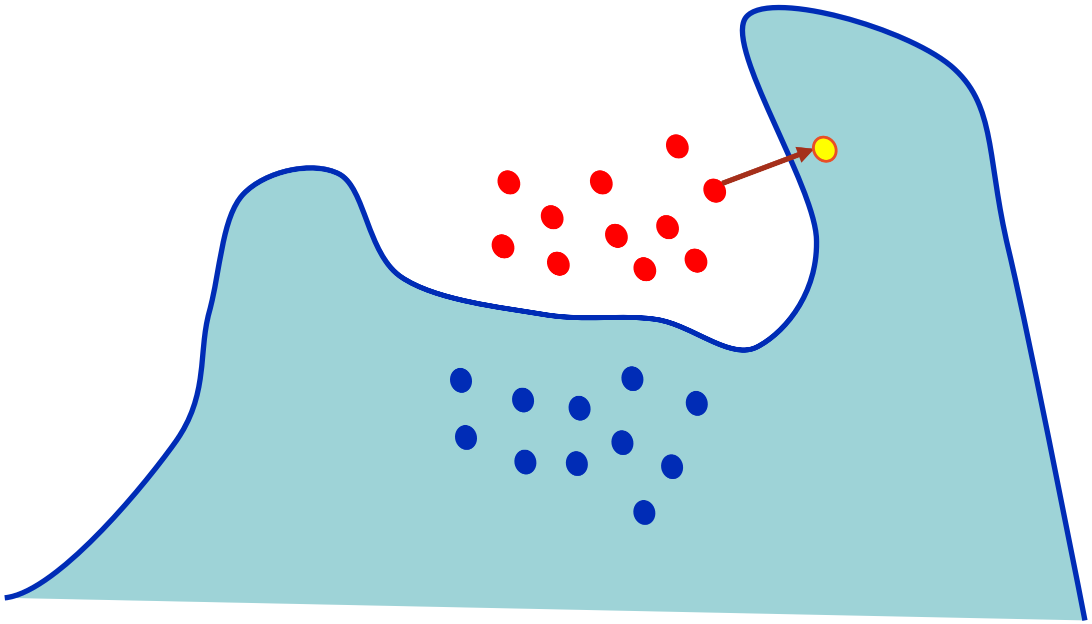
$$\mathrm{z} = N(0, \sigma) \rightarrow \mathrm{E}[\sum_i z_i\, w_i] = 0$$

$$\mathrm{z} = \alpha w \rightarrow \sum_i z_i\, w_i = \alpha \|w\|^2 = \alpha n E[w^2]$$

- Extension to (amost) any (regular) function possible

Exploitation of empty regions

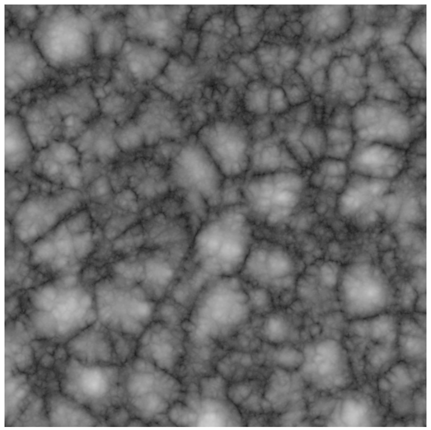
- Regions of the feature space for which no examples are provided are classified randomly and can be exploited by the attacker (again by adding a tailored noise)



- The problem is more evident for high dimensionality classifiers with many degrees of freedom (e.g. CNN)

Exploitation of empty regions

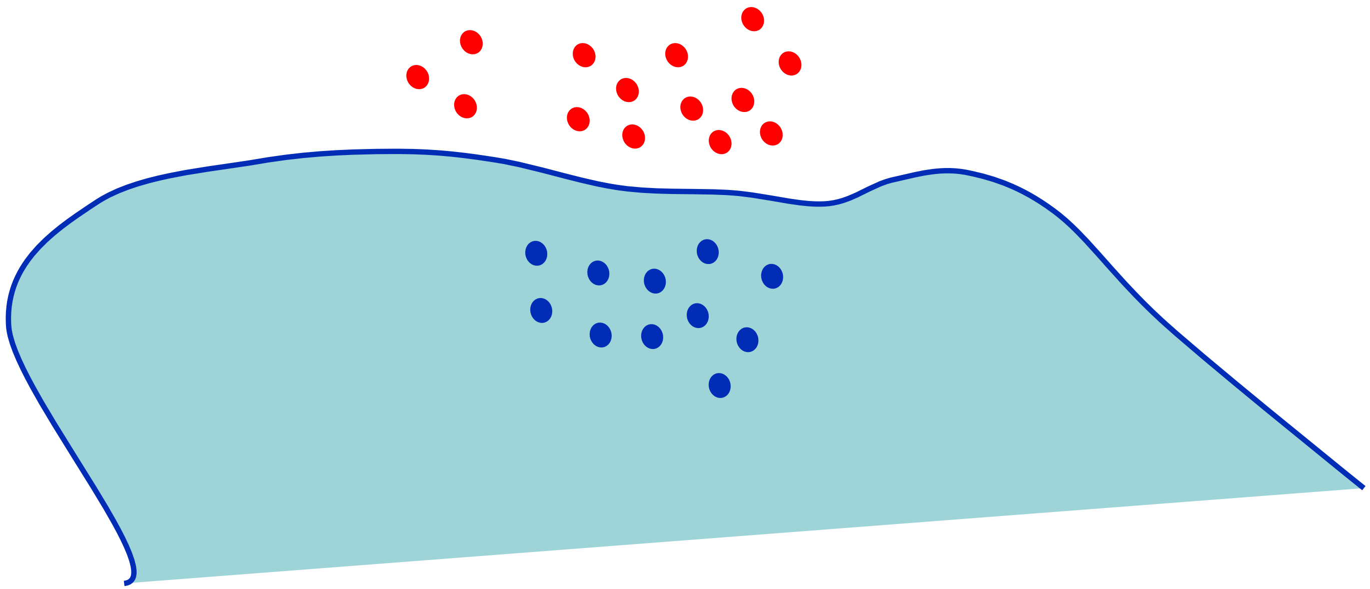
Face detection → NO



Is this Mr Barni ? → YES

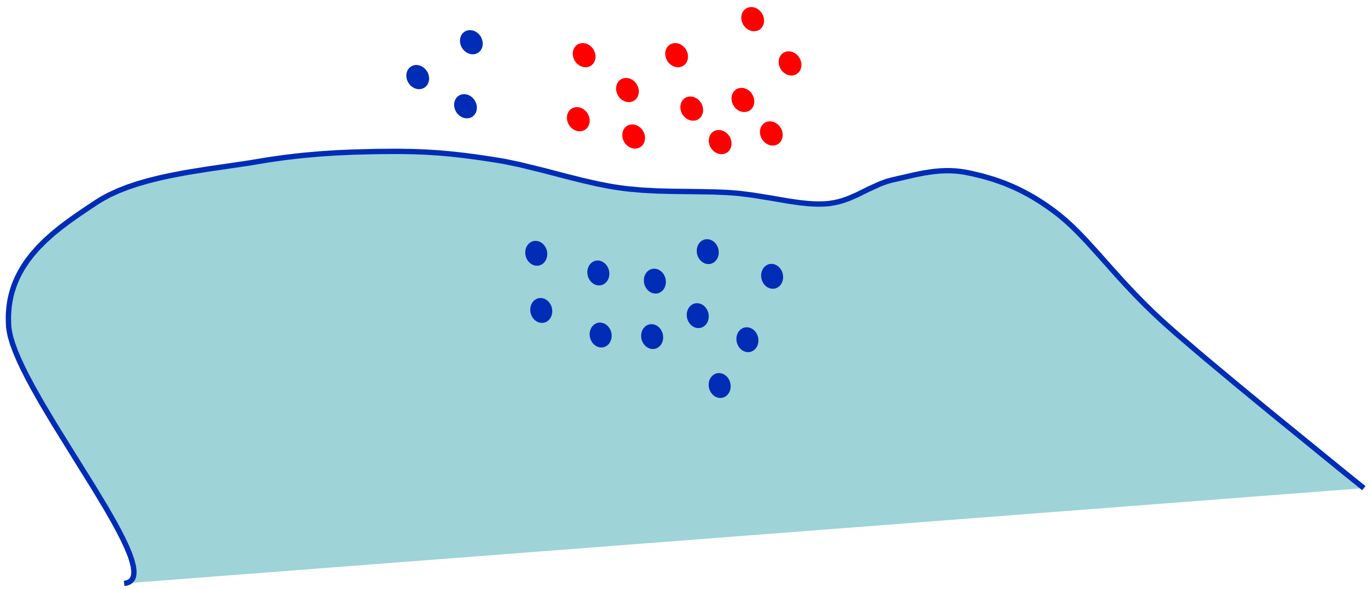
Label poisoning

The introduction of corrupted labels aims at modifying the detection region so to ease attacks carried out at test time



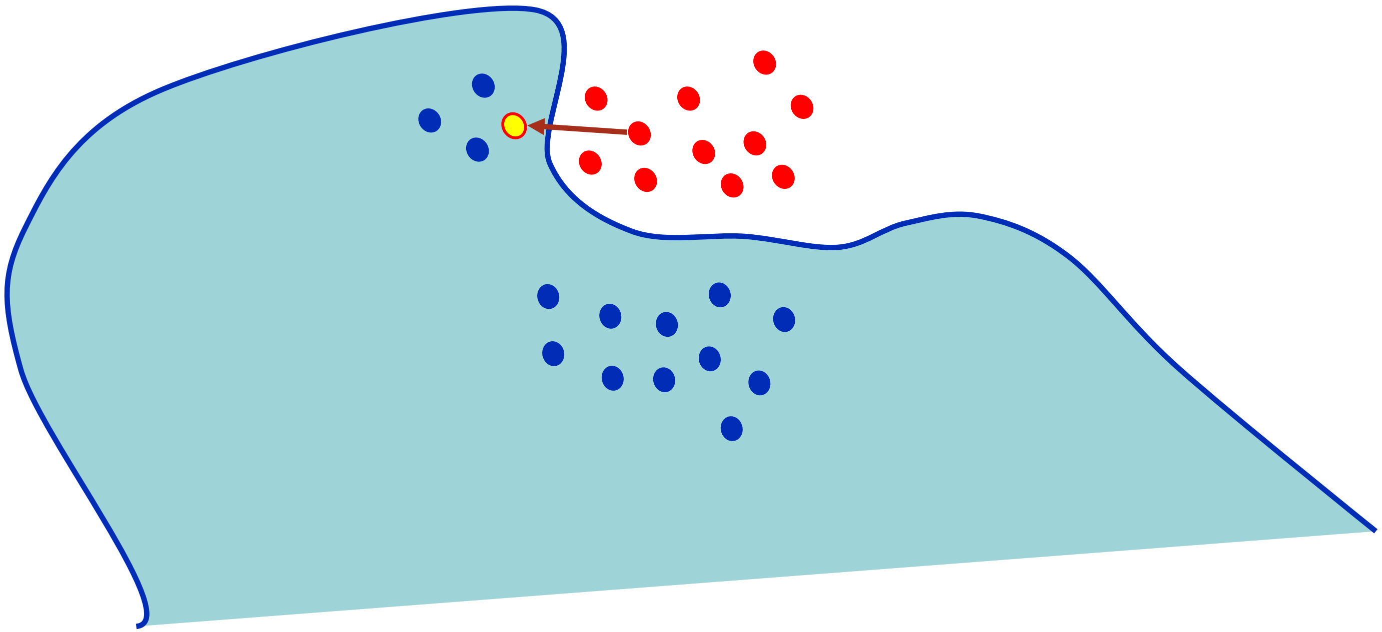
Label poisoning

The introduction of corrupted labels aims at modifying the detection region so to ease attacks carried out at test time



Label poisoning

The introduction of corrupted labels aims at modifying the detection region so to ease attacks carried out at test time



A guided tour to Adv-ML

- Attacker's point of view
- Defender's point of view
- A joint perspective
  - Game-theoretic approach
- Looking ahead

Attacker’s viewpoint: taxonomy

- Focus on binary detection
- In most cases (not always though) the system must detect the presence of an anomalous or dangerous situation, say H1

Decision → / Truth ↓	H0	H1
H0	OK	Denial of Service
H1	Evasion	OK

- Attacks can be carried out at test time, training time or both

The importance of knowledge

“Knowledge is a weapon. Arm yourself before you ride forth to battle”
(George R.R. Martin, A dance with dragons)

“If you know the enemy and know yourself, you need not fear the result of a hundred battles”
(Sun Tzu, The art of war)

Attacks with Perfect Knowledge (PK) vs attacks with Limited Knowledge (LK)

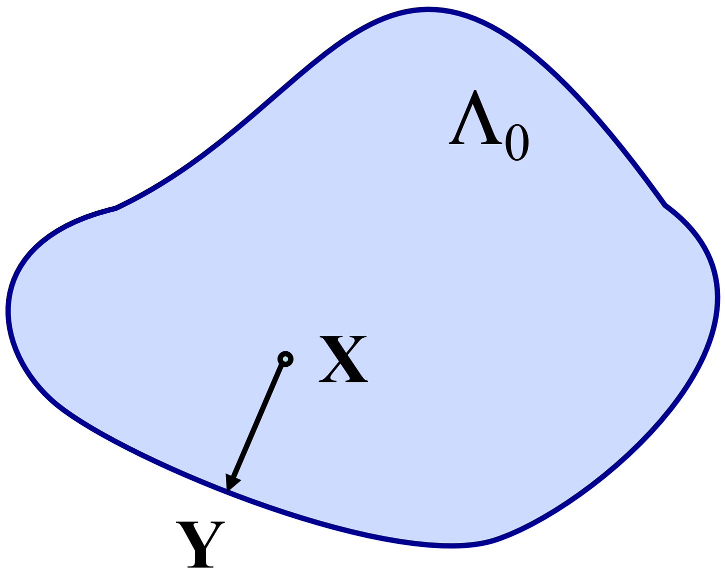
$\phi(\mathcal{L}, \mathcal{F}; \mathcal{D})$

$$\mathcal{L} = hyperameters$$
$$\mathcal{F} = feature\ space$$
$$\mathcal{D} = training\ data$$

Attacks with perfect knowledge (PK)

- The attacker knows the decision function exactly
  - white box attack
  - targeted attack

- **Goal: exit (or enter) the decision region subject to a fidelity criterion**
  - Closed form solution
  - Gradient descent and oracle attacks (also possible in black- or gray-box modality)



Gradient descent attack

- Two formulations

$$x^* = \arg \min_{x' : d(x,x')} \Phi(x') \qquad x^* = \arg \min_{x' : \Phi(x') < 0} d(x,x')$$

- Solution based on gradient computation

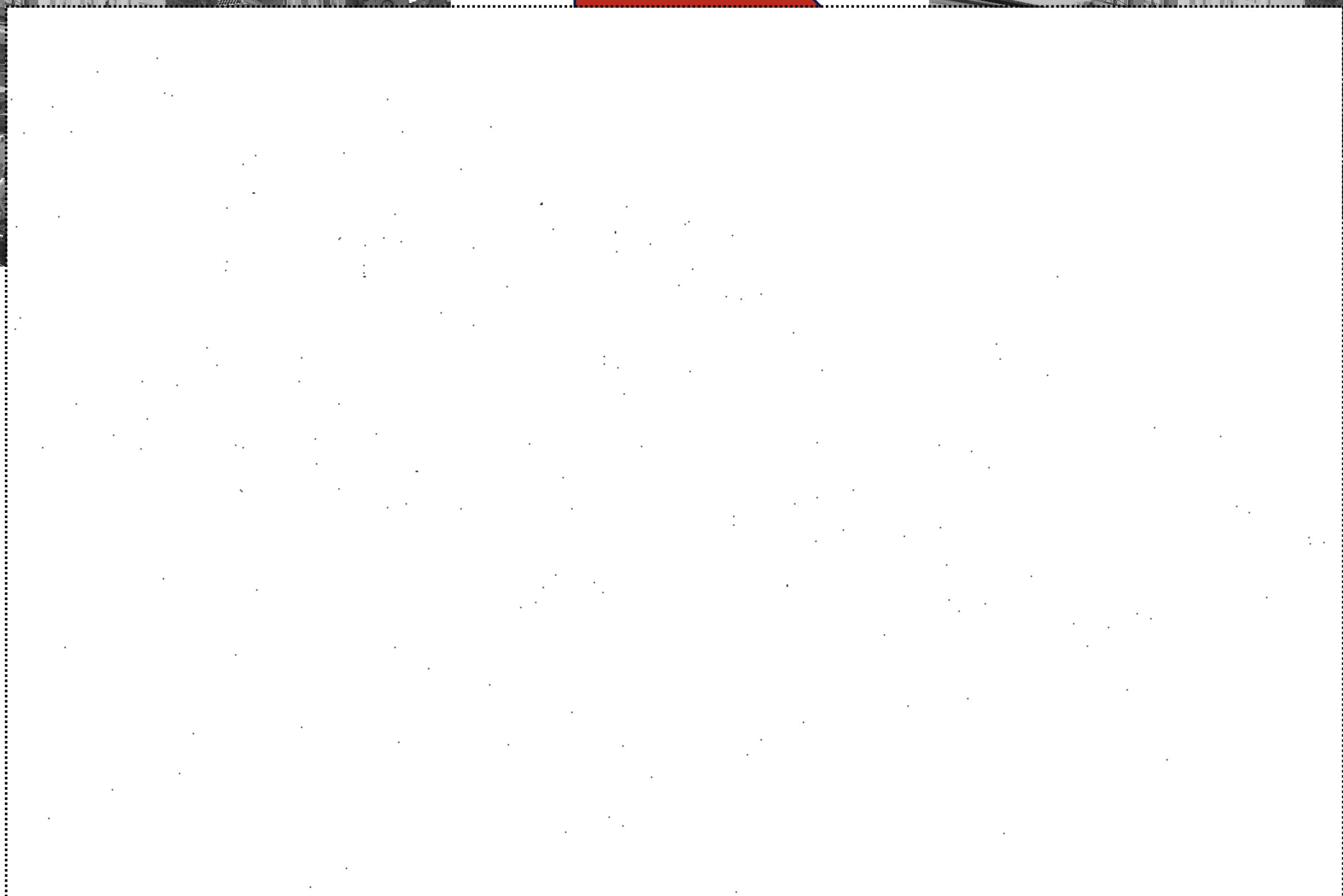
The SVM case

$$\Phi(x) = \sum_i \alpha_i y_i k(x, x_i) + b$$
$$\nabla\Phi(x) = \sum_i \alpha_i y_i \nabla k(x, x_i)$$
$$\nabla k(x, x_i) = -2\gamma(x - x_i)e^{-\gamma\|x - x_i\|^2} \quad \text{RBF kernel}$$

- Easy solutions available also for CNN

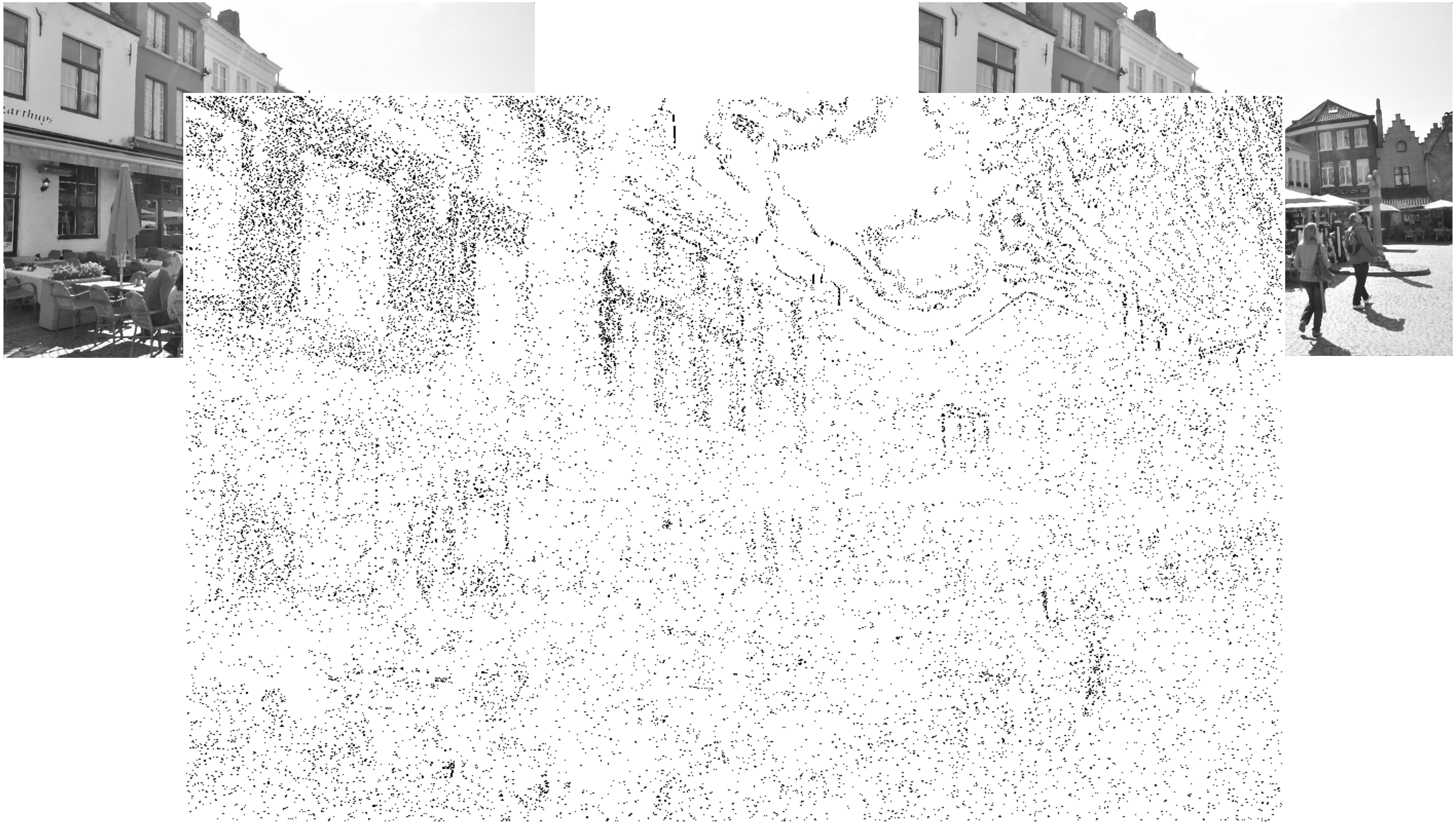
HS image

Attacked image

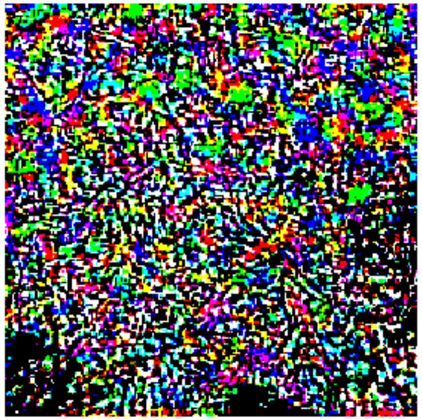


MF3 image

Attacked image



Gradient-based attacks against DL


Classified as a *cat*

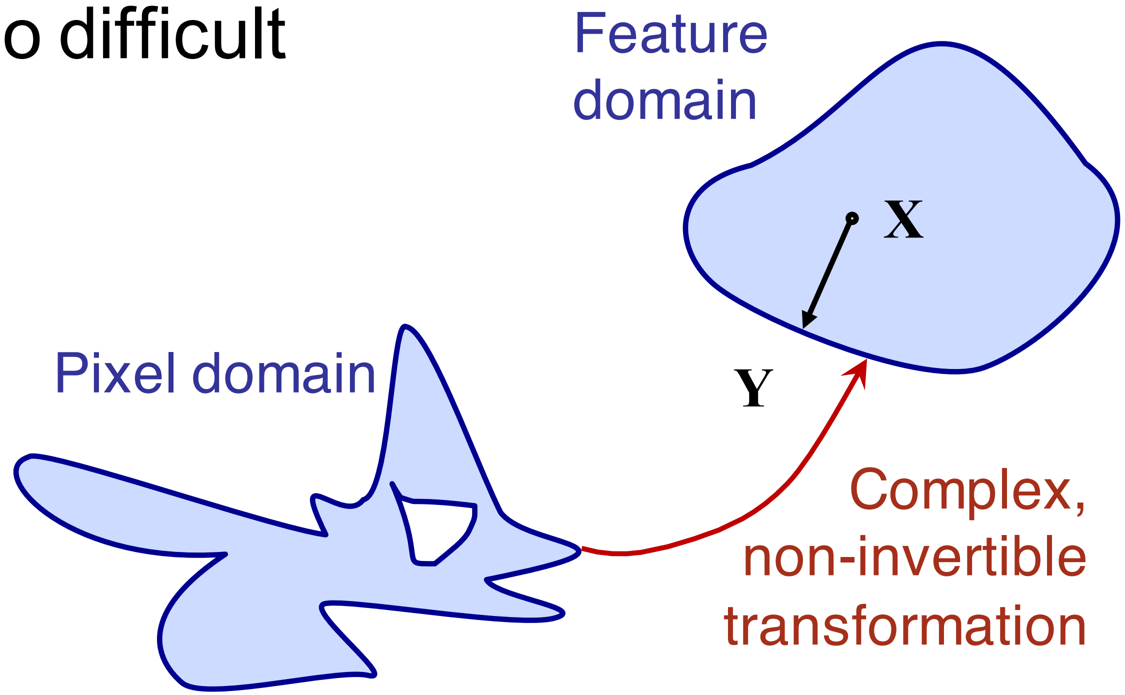
Highly magnified attack



Classified as a *dog*

Attack domain

- Bringing back the attack in the pixel domain may be a difficult task
- Controlling distortion in the feature domain is also difficult
- Easier with DCT, wavelet and histogram-based features
- **Not a problem with DL**



Attacks in real world

- Carrying out the attack in the real world (analog domain) is challenging, but still possible

Attacks with limited knowledge

- When only the feature space (F^*) is known, the attacker may try to devise a **Universal Attack**
- The attack is effective against

$$\phi(\mathcal{L}, \mathcal{F}^*; \mathcal{D}) \quad \forall \mathcal{L}, \forall \mathcal{D}$$

Attacks with limited knowledge (LK)

- The most common approach consists in attacking a **surrogate detector** (attack transferability)

$$\hat{\phi} = \phi(\hat{\mathcal{L}}, \mathcal{F}; \widehat{\mathcal{D}})$$

Examples:

- N. Papernot, P. McDaniel, I. Goodfellow. "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." arXiv preprint arXiv:1605.07277 (2016).

Example

- To account for mismatch in training data a stronger attack must be applied
- Results regarding SVM-based detection of histogram stretching*

ν	$\mathbf{P_e}(\hat{\phi})$	$\mathbf{P_e}(\phi)$	**Mean SSIM**	**Mean PSNR**
0	100%	53%	0.99996	73.9766
0.2	100	80.5	0.99995	72.6223
0.4	100	100	0.99994	71.2038

Surrogate detector

Real detector

* Z. Chen, B. Tondi, X. Li, R. Ni, Y. Zhao, and M. Barni, "A gradient-based pixel-domain attack against SVM detection of global image manipulations",WIFS 2017, IEEE Int. Workshop, Rennes, France

Defender’s viewpoint

“Knowledge is a weapon. Arm yourself before you ride forth to battle”
(George R.R. Martin, A dance with dragons)

“If you know the enemy and know yourself, you need not fear the result of a hundred battles”
(Sun Tzu, The art of war)

- Adversary-aware detectors
  - Look for attack traces
  - Adversary aware training (detection vs resilience)

Adversary aware - informed - defenses

- The analyst looks for the traces left by the CF algorithm
- Build a new detector ϕ_{aw} using the same or a new set of features
- Most common case: retrain an ML detector
  - Rich enough feature space needed

$$\phi_{aw} = \phi(\mathcal{L}, \mathcal{F}; \mathcal{D} \cup \mathcal{D}_{aw})$$

$\phi \longrightarrow$ **PK or LK attack to ϕ** $\longrightarrow \phi_{aw}$

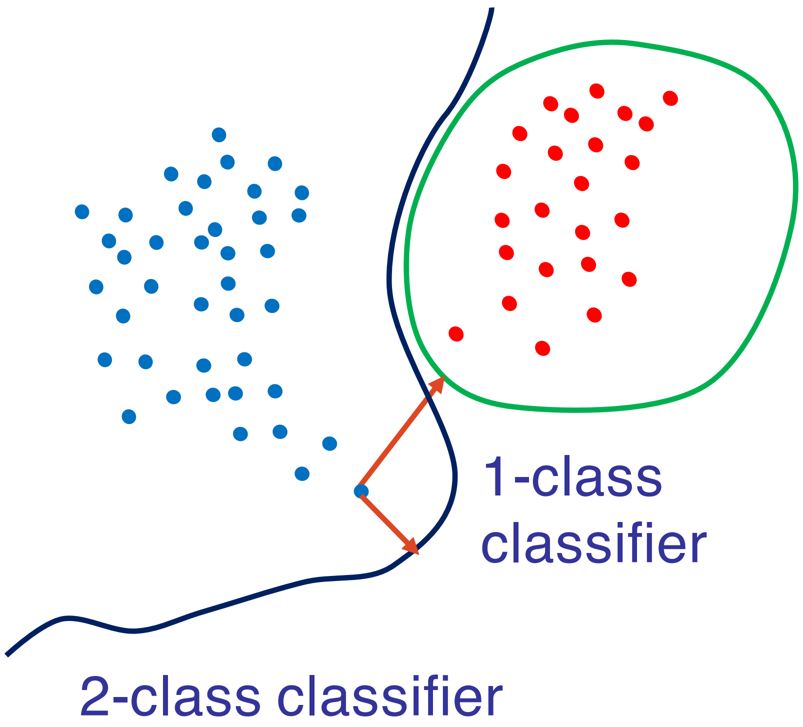
A way to exit the PK scenario or disinform the attacker Cat & mouse otherwise

If you miss knowledge: *building a BIG WALL may help*

- Intrinsically (more) secure detectors
  - Feature choice
  - Simple detection boundaries (possibly at the expense of robustness)
  - 1-class detectors
  - Multiple classifiers
  - Randomized detectors

Detector architecture

- 1-class classifiers are intrinsically more robust against unknown attacks due to their close decision boundary

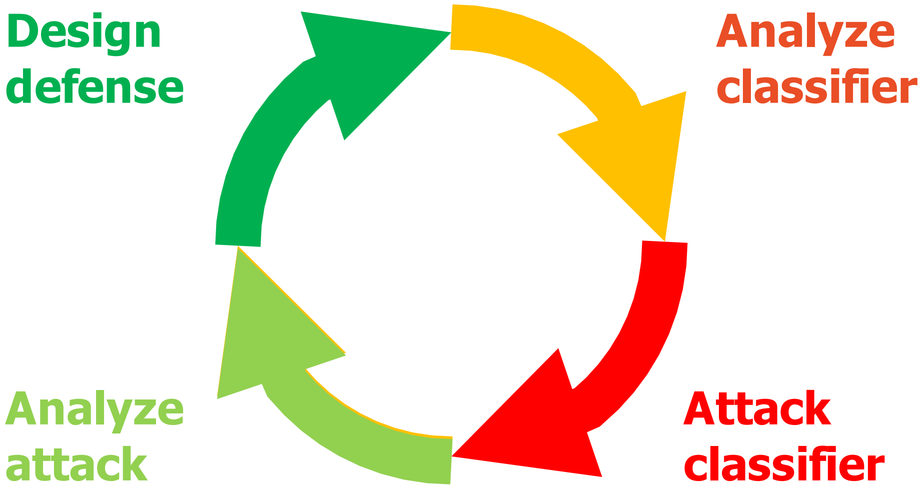


Knoweldge is a weapon … for who?

If you know the enemy and know yourself, you need not fear the result of a hundred battles

If you know the enemy and know yourself, you need not fear the result of a hundred battles

Classical attack-defense cycle



- Avoid entering a **never-ending cat & mouse loop**
- Worst case assumption is often **too pessimistic** and does not say much about actual security

Adversarial machine learning and game theory: a perfect fit

Game Theory in a nutshell

D \ A	$S_{A,1}$	$S_{A,2}$	$S_{A,3}$	$S_{A,4}$	$S_{A,5}$	…	$S_{A,n}$
$S_{D,1}$	1 : 3	3 : 1	4 : 1	3 : 2	3 : 0	… : …	30 : 2
$S_{D,2}$	3 : 1	2 : 2	2 : 0	1 : 3	2 : 1	… : …	1 : 3
$S_{D,3}$	4 : 2	6 : 0	7 : 0	30 : 6	6 : 0	… : …	2 : 5
$S_{D,4}$	2 : 6	0 : 4	-3 : -5	4 : -8	0 : 0	… : …	1 : 9
$S_{D,5}$	7 : -4	0 : 0	0 : 20	4 : 0	-1 : 0	… : …	0 : 12
…	… : …	… : …	… : …	… : …	… : …	… : …	… : …
$S_{D,m}$	0 : 0	25 : 0	30 : 15	12 : 0	17 : 0	… : …	11 : 16

Competitive (zero-sum) games

D \ A	$S_{A,1}$		$S_{A,2}$		$S_{A,3}$		$S_{A,4}$		$S_{A,5}$		…		$S_{A,n}$	
$S_{D,1}$	1	-1	3	-3	4	-4	3	-3	3	-3	…	…	30	-30
$S_{D,2}$	3	-3	2	-3	2	-2	1	-1	2	-2	…	…	1	-1
$S_{D,3}$	4	-4	6	-6	7	-7	30	-30	6	-6	…	…	2	-2
$S_{D,4}$	2	-2	0	0	-3	3	4	-4	0	0	…	…	1	-1
$S_{D,5}$	7	-7	0	0	0	0	4	-4	-1	1	…	…	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$S_{D,m}$	0	0	25	-25	30	-30	12	-12	17	-17	…	…	11	-11

Choice of strategies: worst case

- Players choose the strategy which results in the maximum of the minimum payoff
- This may result in a too pessimistic approach

D \ A	$S_{A,1}$		$S_{A,2}$		$S_{A,3}$	
$S_{D,1}$	10	3	1	4	0	2
$S_{D,2}$	3	5	5	0	2	-2
$S_{D,3}$	4	1	6	-5	1	-7

Choice of strategies: worst case

- Players choose the strategy which results in the maximum of the minimum payoff
- This may result in a too pessimistic approach

D \ A	$S_{A,1}$		$S_{A,2}$		$S_{A,3}$	
$S_{D,1}$	10	3	1	4	0	2
$S_{D,2}$	3	5	5	0	2	-2
$S_{D,3}$	4	1	6	-5	1	-7

Choice of strategies: worst case

- Players choose the strategy which results in the maximum of the minimum payoff
- This may result in a too pessimistic approach

D \ A	$S_{A,1}$		$S_{A,2}$		$S_{A,3}$	
$S_{D,1}$	10	3	1	4	0	2
$S_{D,2}$	3	5	5	0	2	-2
$S_{D,3}$	**4**	**1**	6	-5	1	-7

Dominant strategy

When a dominant strategy exists a rationale player will surely play it

FA \ A	$S_{A,1}$		$S_{A,2}$		$S_{A,3}$		$S_{A,4}$		$S_{A,5}$		…		$S_{A,n}$	
$S_{FA,1}$	1	3	3	1	4	1	3	2	3	0	…	…	10	0
$S_{FA,2}$	3	1	2	2	2	0	1	3	2	1	…	…	1	3
$S_{FA,3}$	4	2	6	0	7	0	30	6	6	0	…	…	1	5
$S_{FA,4}$	2	6	0	4	-3	-5	4	-8	0	0	…	…	1	9
$S_{FA,5}$	7	-4	0	0	0	20	4	0	-1	0	…	…	0	12
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$S_{FA,m}$	8	0	25	0	30	15	90	0	17	0	…	…	**11**	**16**

Nash equilibrium

No player gets an advantage by changing his strategy assuming the other does not change his own

$$\mathrm{u}_1(s_1^*, s_2^*) \geq \mathrm{u}_1(s_1, s_2^*) \quad \forall s_1 \in S_1$$

$$\mathrm{u}_2(s_1^*, s_2^*) \geq \mathrm{u}_2(s_1^*, s_2) \quad \forall s_2 \in S_2$$

... and many others

Examples (few available)

- D develops a detector ϕ
- A develops an attack $\mathcal{A}$ againts ϕ
- D develops an algorithm ϕ_a to detect the traces left by $\mathcal{A}$
- Eventually D builds a combined detector $\phi' = \phi \circ \phi_a$

GAME

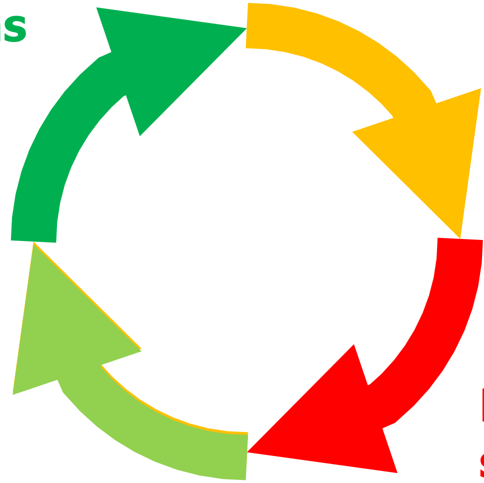
- A chooses the strength of the attack

- D decides how to combine ϕ and ϕ_a (e.g. $\alpha\phi + \beta\phi_a$)

M.C.Stamm, W.S.Lin, K.J.R.Liu,"Temporal forensics and anti-forensics for motion compensated video," IEEE TIFS, vol. 7, no. 4, pp. 1315–1329, Aug. 2012.

Steganography and steganalysis

Look for the message in textured regions



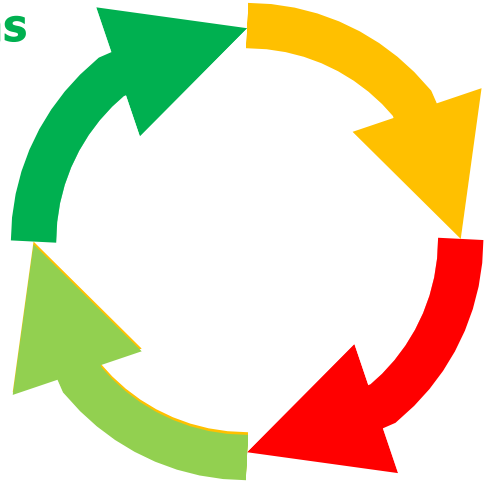
Stego message is more easily detected in flat regions

Hide the stego message in textured areas

Know that message is never hidden in flat areas

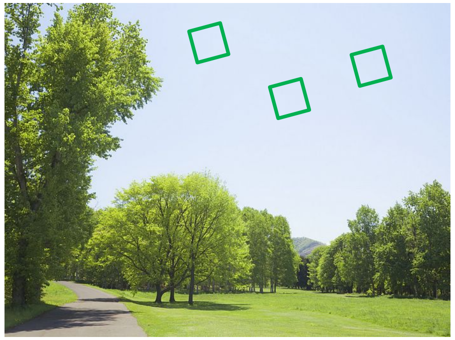
Steganography and steganalysis

Look for the message in textured regions



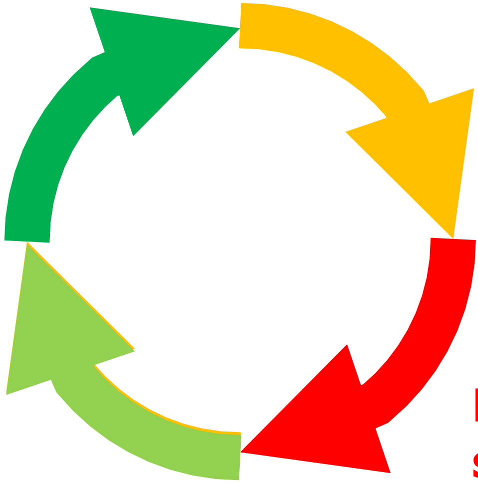
The detector does not look at flat areas

Steganography and steganalysis



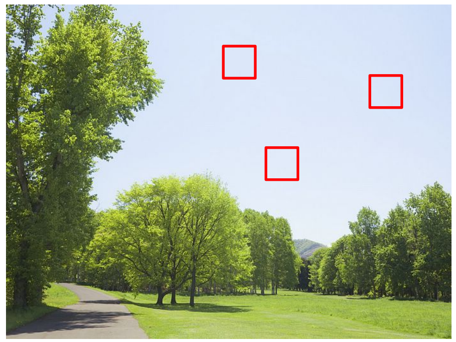
Look for the message in flat areas

The detector does not look at flat areas

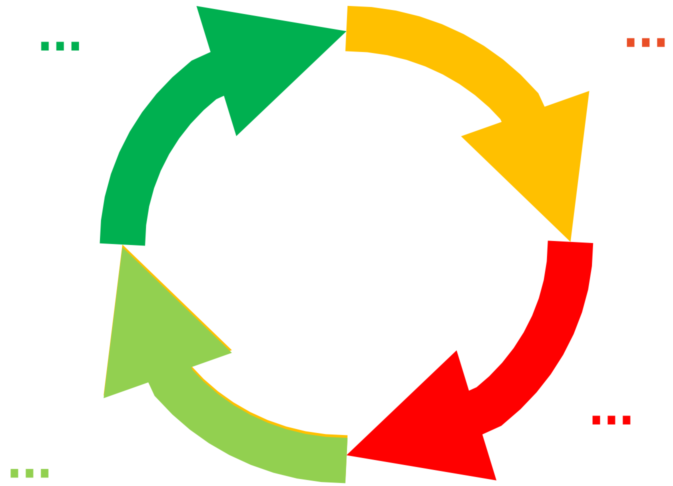


Know that message is hidden in flat areas

Hide the stego message in flat areas



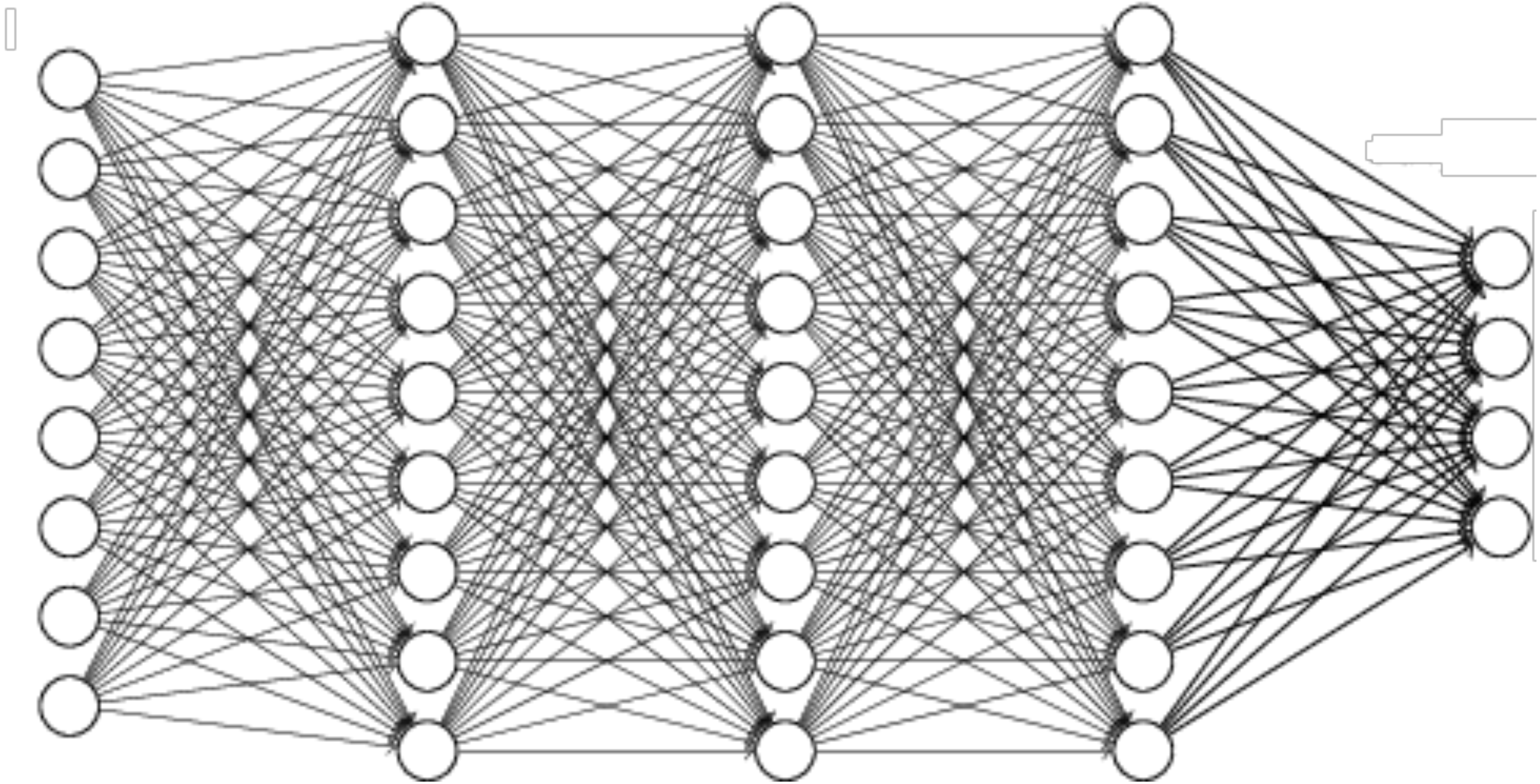
Steganography and steganalysis



- Keep running around
- Model the arms race as a game
- Attackers:
  - split the payload between flat and textured areas
- Defender
  - Look at both flat an textured areas with different confidence

P. Schöttle, R. Böhme, "A Game-Theoretic Approach to Content-Adaptive Steganography", *in M. Kirchner, D. Ghosal (eds) Information Hiding. IH 2012*. Lecture Notes in Computer Science, vol 7692. Springer, Berlin, Heidelberg

Peculiarities of DL

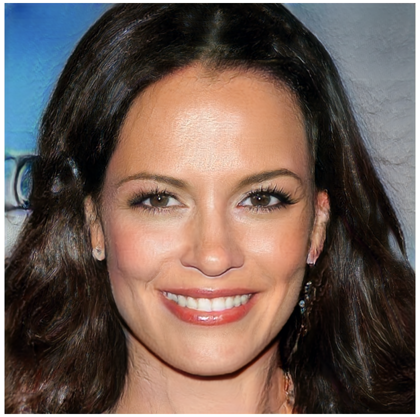


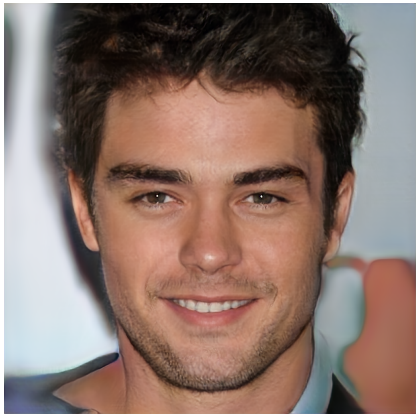
Peculiarities of DL

- Investigating the security of Deep Learning techniques is particularly important
  – attacks carried out directly in the sample domain
  – the huge dimensionality of the input and the parameter space eases the attacks
    - adversarial examples
    - attack transferability (?)
  – opacity / presence of confounding factors
  – huge dimension of training set

GAN and game theory

- GANs and other generative models proved to be able to generate visually plausible fakes
  - AI-generated fakes raise the alarm about fake media to a unprecedented level
  - **Game-theoretic formulation involving two CNNs !!!!**



Looking ahead

Who's going to win

- The struggle between attackers and defenders is going on
- In many applications, the scale needle hangs on the side of attacker
- Yet as research goes on the task of the attacker is getting more and more difficult

Looking ahead: new security threats

- Training poisoning
  - Backdoor attacks
    - C. Liao, H. Zhong, A. Squicciarini, S. Zhu, D. Miller, “Backdoor Embedding in Convolutional Neural Network Models via Invisible Perturbation” arXiv preprint arXiv:1808.10307 (August 2018)
- Network protection
  - CNN-Watermarking through proper training
  - Anti-piracy transformation
    - M. Chen, M. Wu, “Protect Your Deep Neural Networks from Piracy”, WIFS 2018, Hong-Kong
- Privacy preserving CNN
  - Homomorpich encryption, MPC
  - Differential privacy (GAN-based)
- ....