Pattern Recognition
and Applications Lab

ICCV'17 – Half-day Tutorial on Adversarial Pattern Recognition

Battista Biggio and Fabio Roli

http://pralab.diee.unica.it/en/wild-patterns

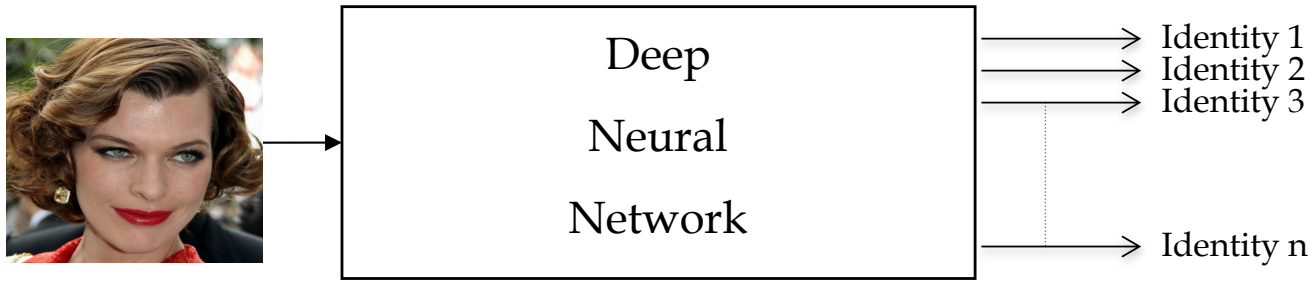
ICCV '17, Venice, Italy, October 22, 2017

University
of Cagliari, Italy

Department of Electrical
and Electronic
Engineering

Deep face recognition…

[O.M. Parkhi et al., Deep face recognition, BMVC 2015]



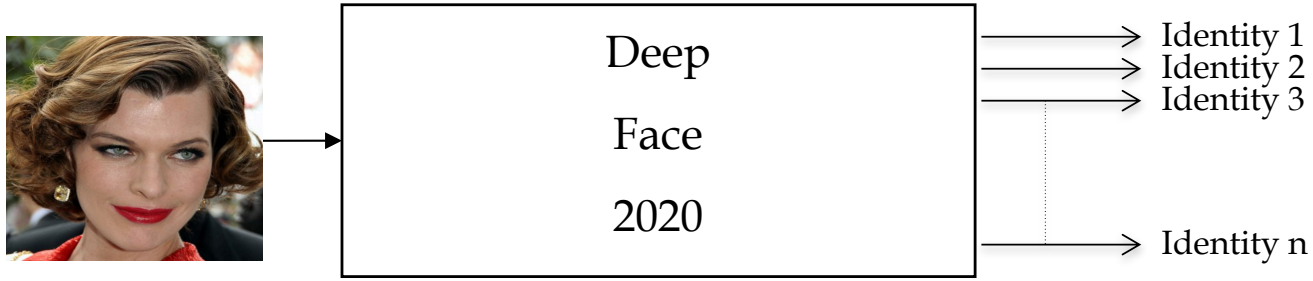
Parkhi et al. (BMVC 2015) trained a **very deep** neural network to recognize face images of **2622 celebrities** with a **98.95% test-set accuracy**

Italian TV show “Tale e quale”

Italian talent show

The challenge for participants of the show is to impersonate famous singers…

The Deep Impersonator TV show…



Imagine that you are a **41-year-old white male**, and you want to participate the "Deep Impersonator" TV show….

The challenge of the show is **to fool** the last version of the face recognition system "Deep Face 2020" by **impersonating** the celebrity of the photo, the American model and actress **Milla Jovovich**

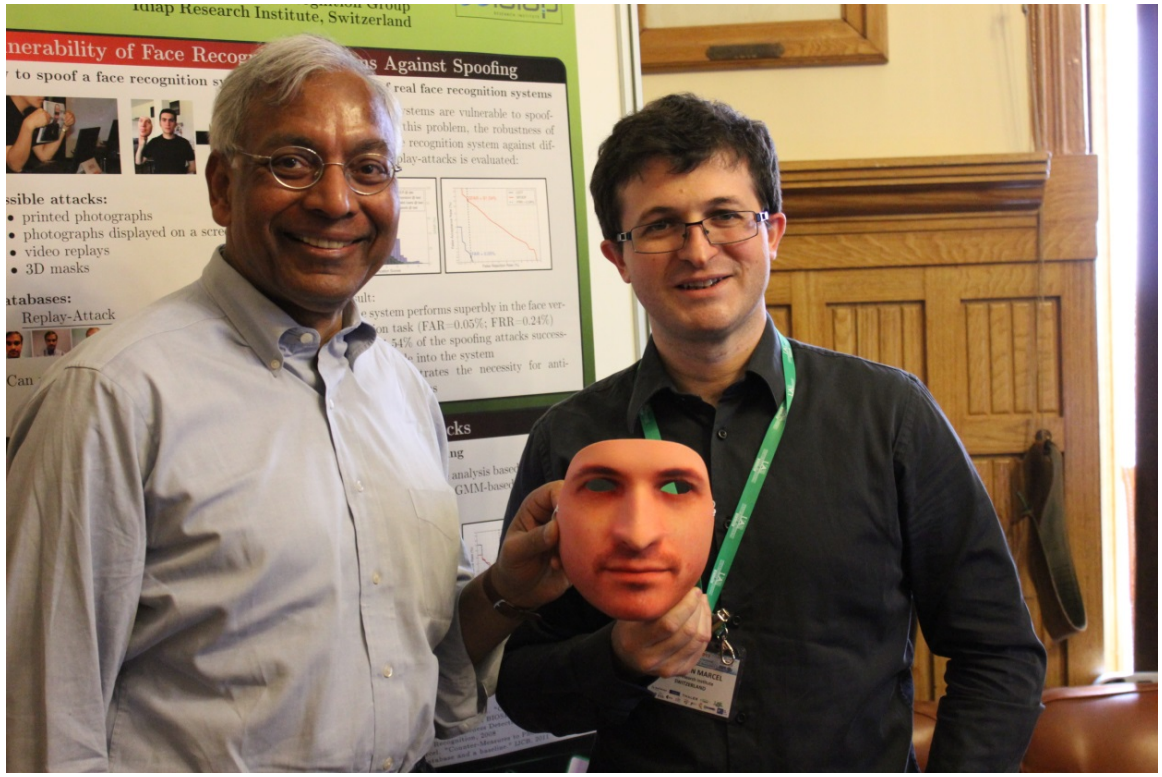
What face spoofing technique would you use ?

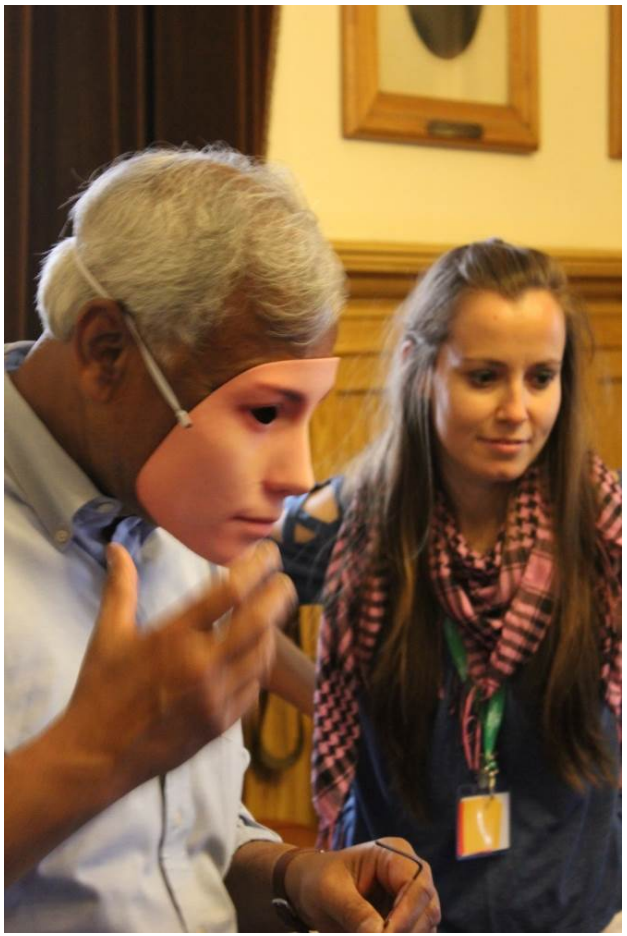
Let's see the best of the state of the art…

ICB 2013 - Spoofing Challenge and Award

TABULA RASA

http://www.tabularasa-euproject.org





ICB 2013 - Spoofing Challenge and Award

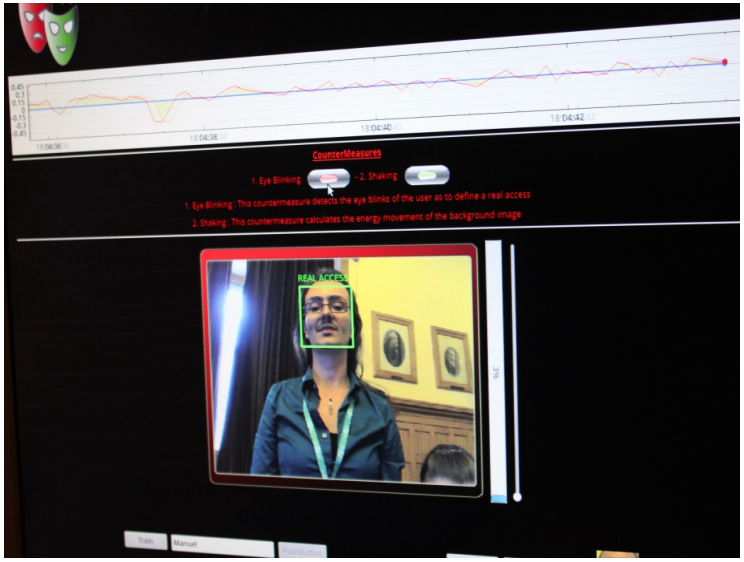
TABULA RASA

http://www.tabularasa-euproject.org

ICB 2013 - Spoofing Challenge and Award

The winner of the TABULA RASA spoofing award

Makeup attack by Antitza Dantcheva, http://www.antitza.com



Adversarial faces

[M. Sharif et al., ACM CCS 2016]

M. Sharif et al. developed a systematic method to automatically generate attacks to deep face recognition systems, attacks which are realized through printing a pair of eyeglass frames.

When worn by a **41-year-old white male** (photo on the left) whose image is supplied to a deep face-recognition algorithm, the eyeglasses allow him to impersonate the American actress **Milla Jovovich**.

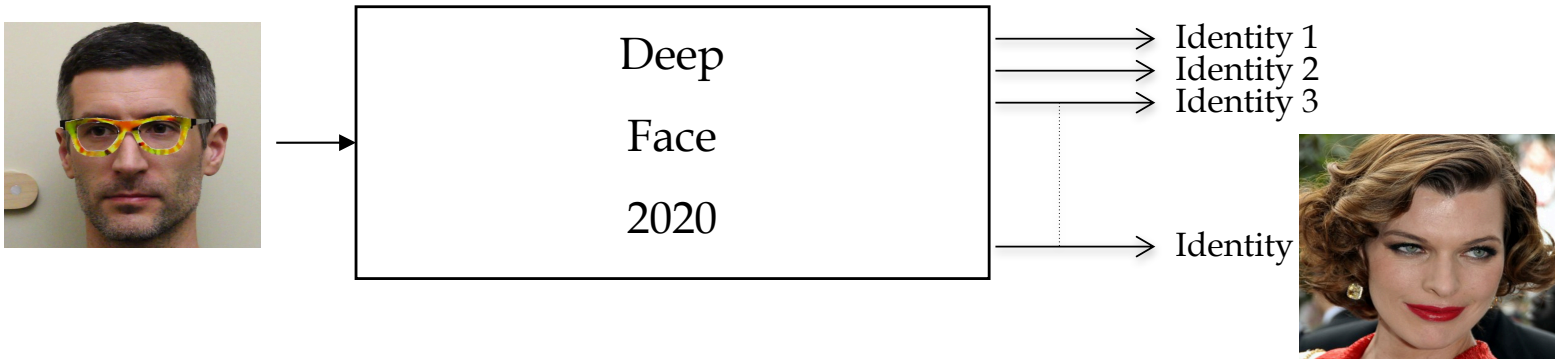
Take-home message

We are living exciting time for machine learning…

…Our work feeds a lot of consumer technologies for personal applications...

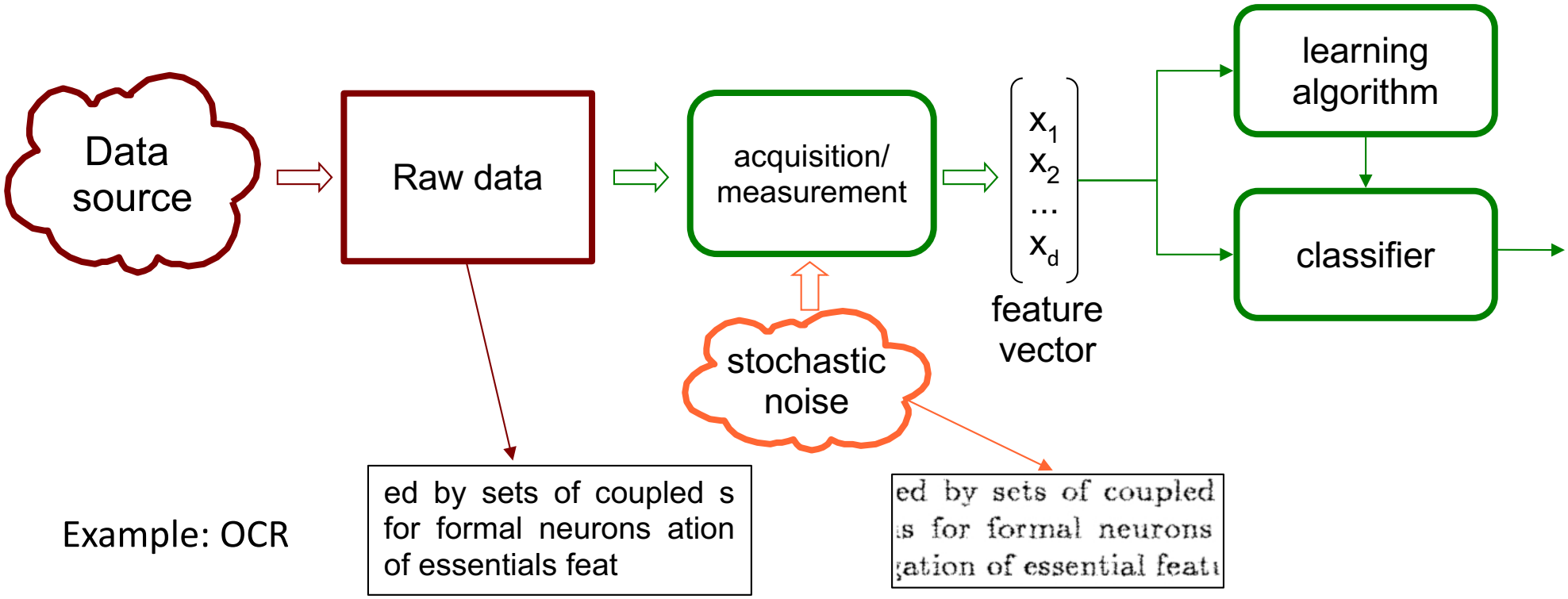
This opens up new big possibilities, but also new security risks

Are we ready for this?



Can we use the classical machine learning model for **adversarial** pattern classification?

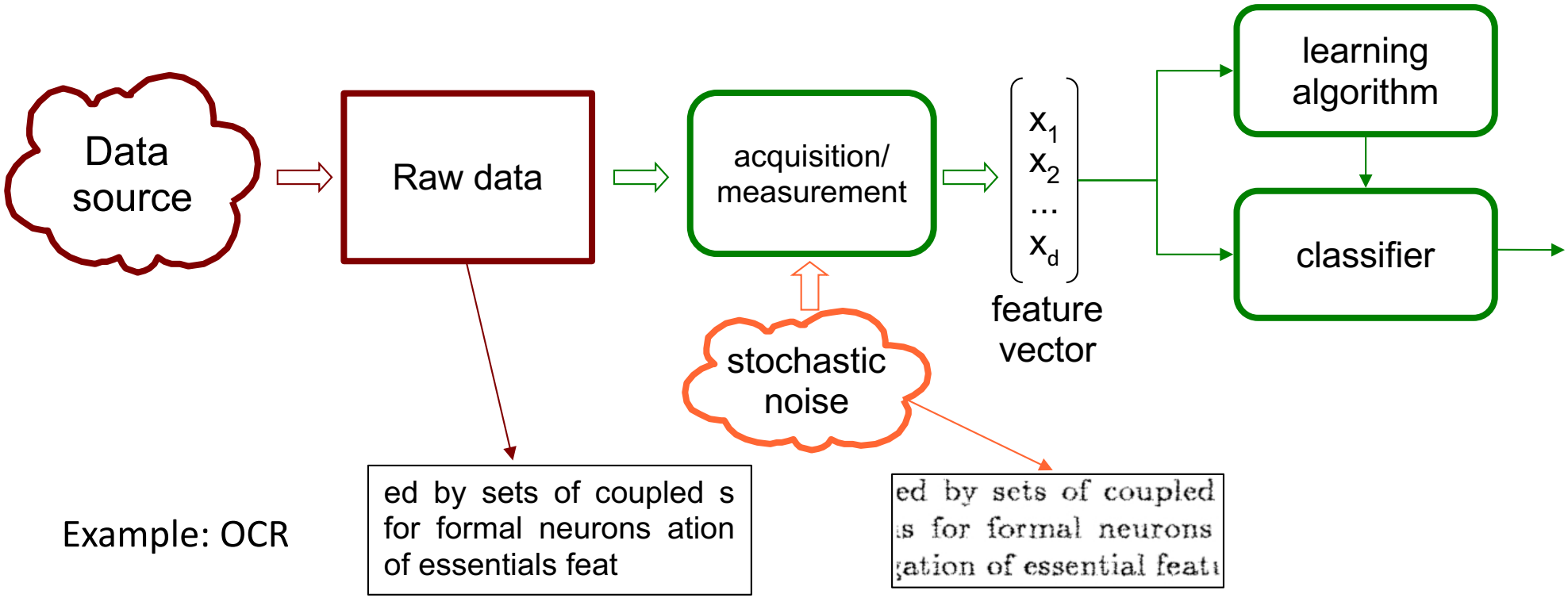
The classical statistical model



Note these two implicit assumptions of the model:

1. **the source of data is given, and it does not dependent on the classifier**
2. **Noise affecting data is stochastic**

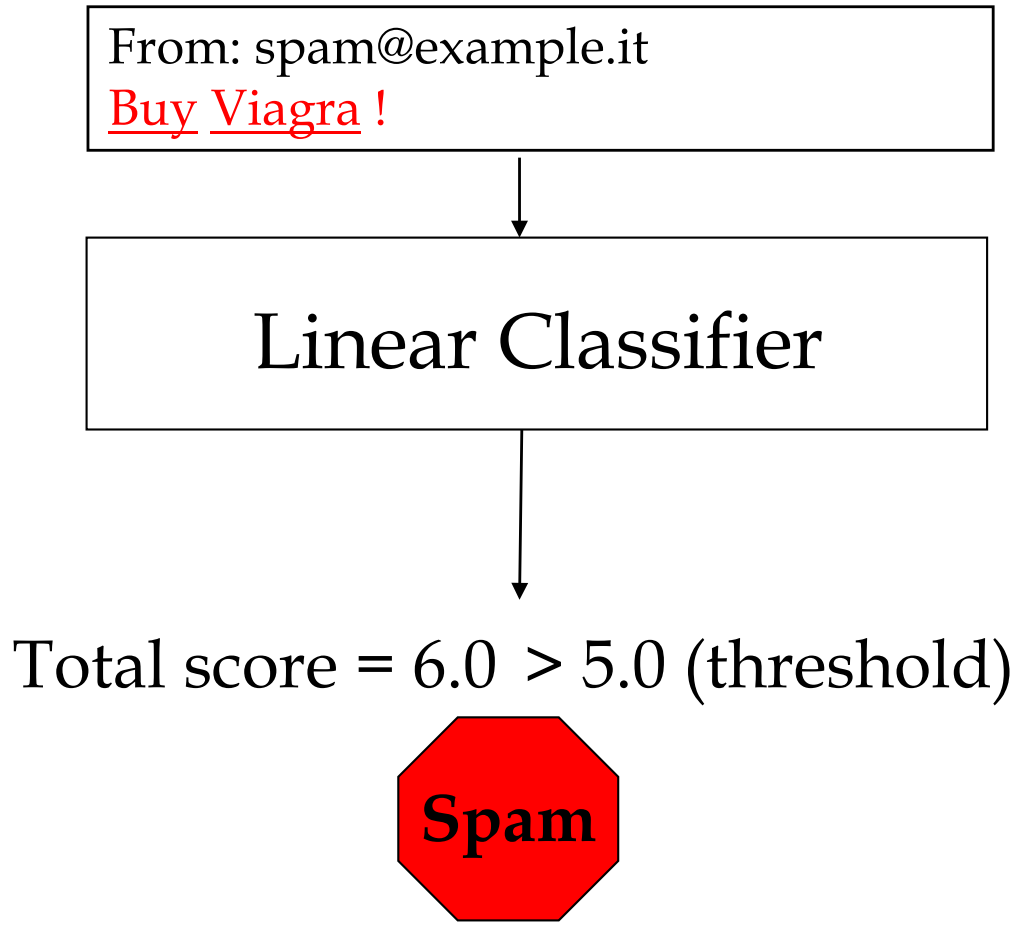
Can this model be used under attack?



Example: OCR

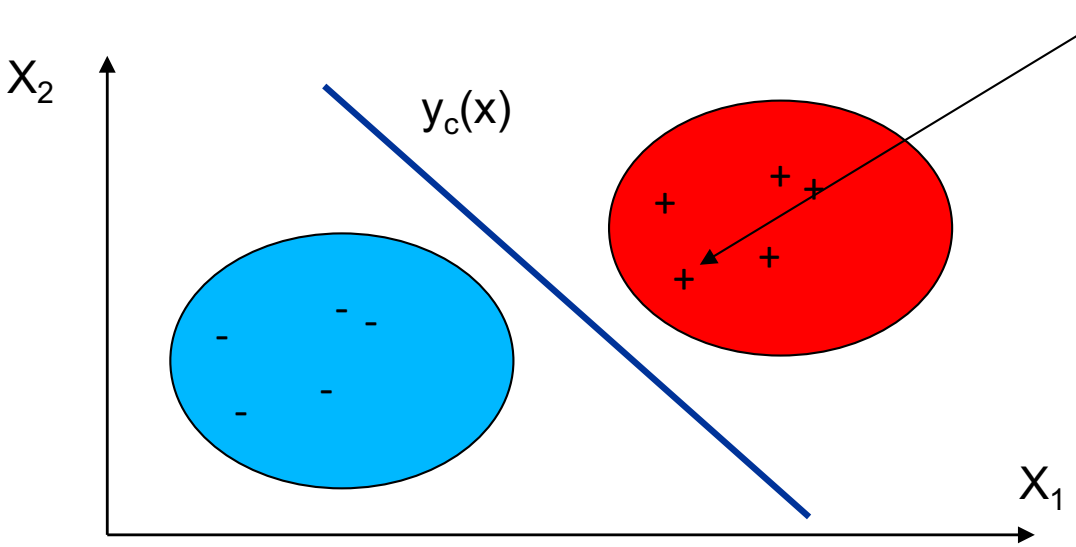
An example: spam filtering

Feature weights
buy = 1.0
viagra = 5.0



- The famous SpamAssassin filter is really a linear classifier
  - http://spamassassin.apache.org

Feature space view



From: spam@example.it
Buy Viagra!

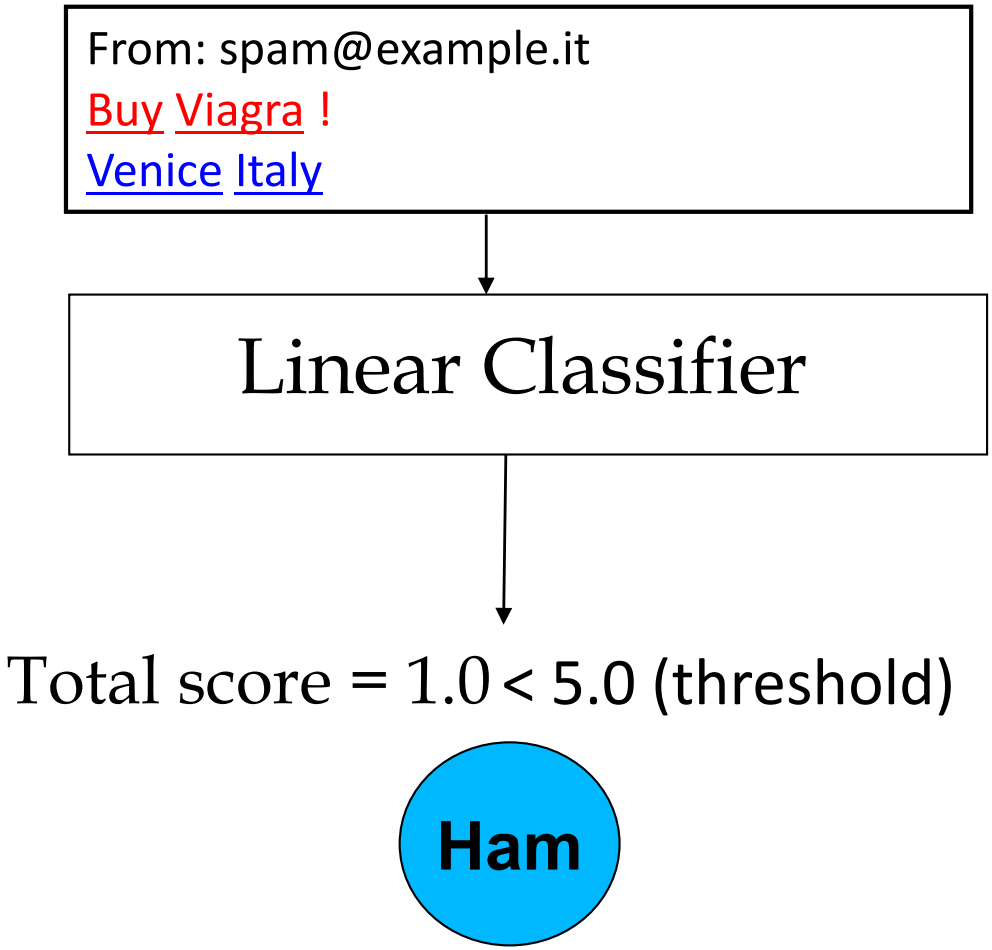
Feature weights
buy = 1.0
viagra = 5.0

- Classifier's weights can be learnt using a training set
- The SpamAssassin filter uses the perceptron algorithm

But spam filtering is not a *stationary* classification task, the data source is not neutral…

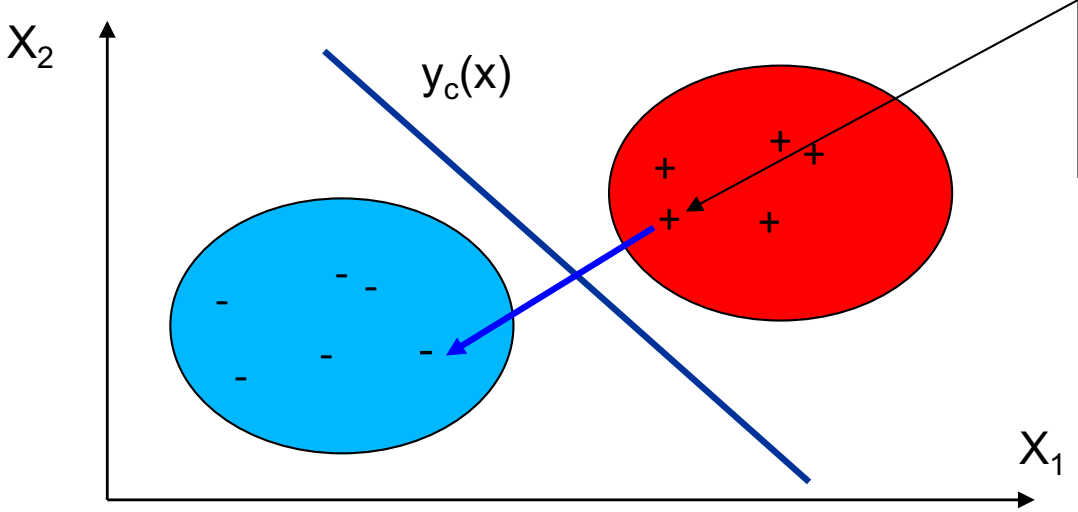
The data source can add "good" words

Feature weights
buy = 1.0
viagra = 5.0
Venice = -2.0
Italy = -3.0



✓Adding "good" words is a typical spammers' trick [Z. Jorgensen et al., JMLR 2008]

Adding good words: feature space view

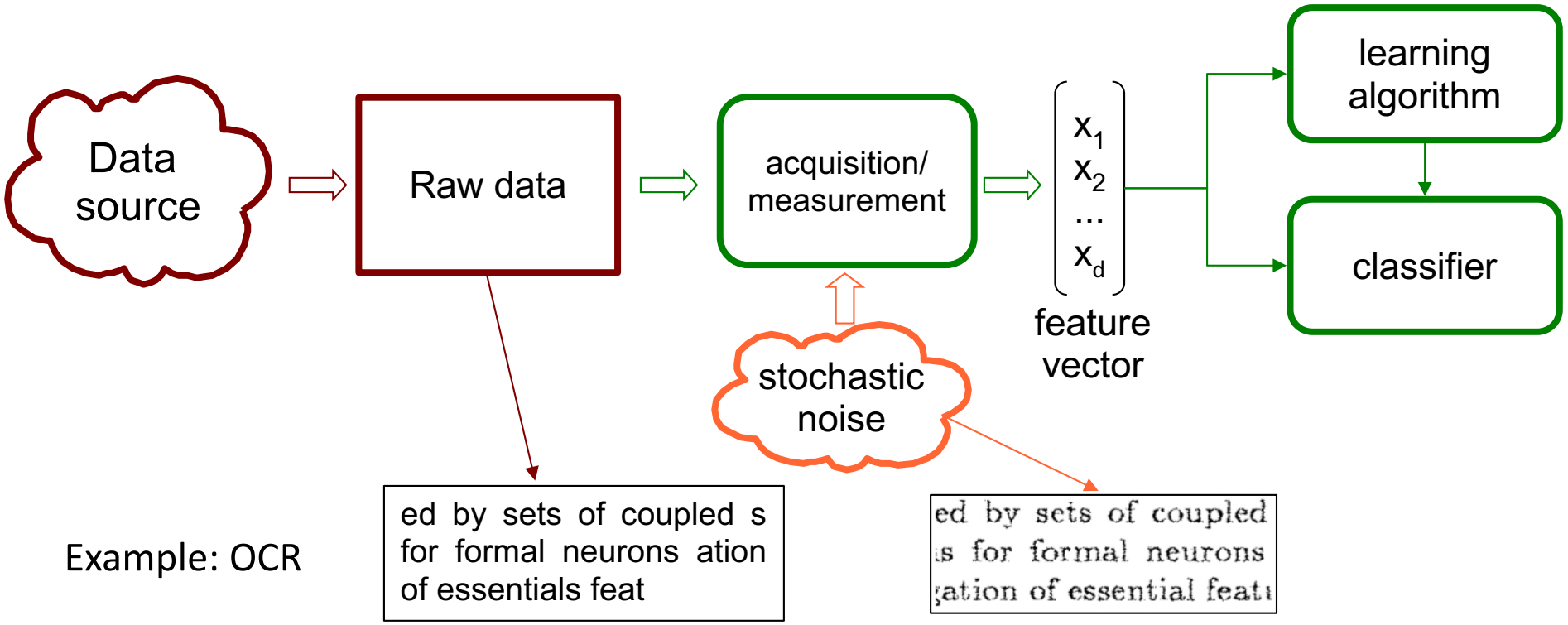


From: spam@example.it
Buy Viagra!
Venice Italy

Feature weights
buy = 1.0
viagra = 5.0
Venice = -2.0
Italy = -3.0

✓Note that spammers corrupt patterns with a *noise* that is *not random*..

Is this model good for spam filtering?

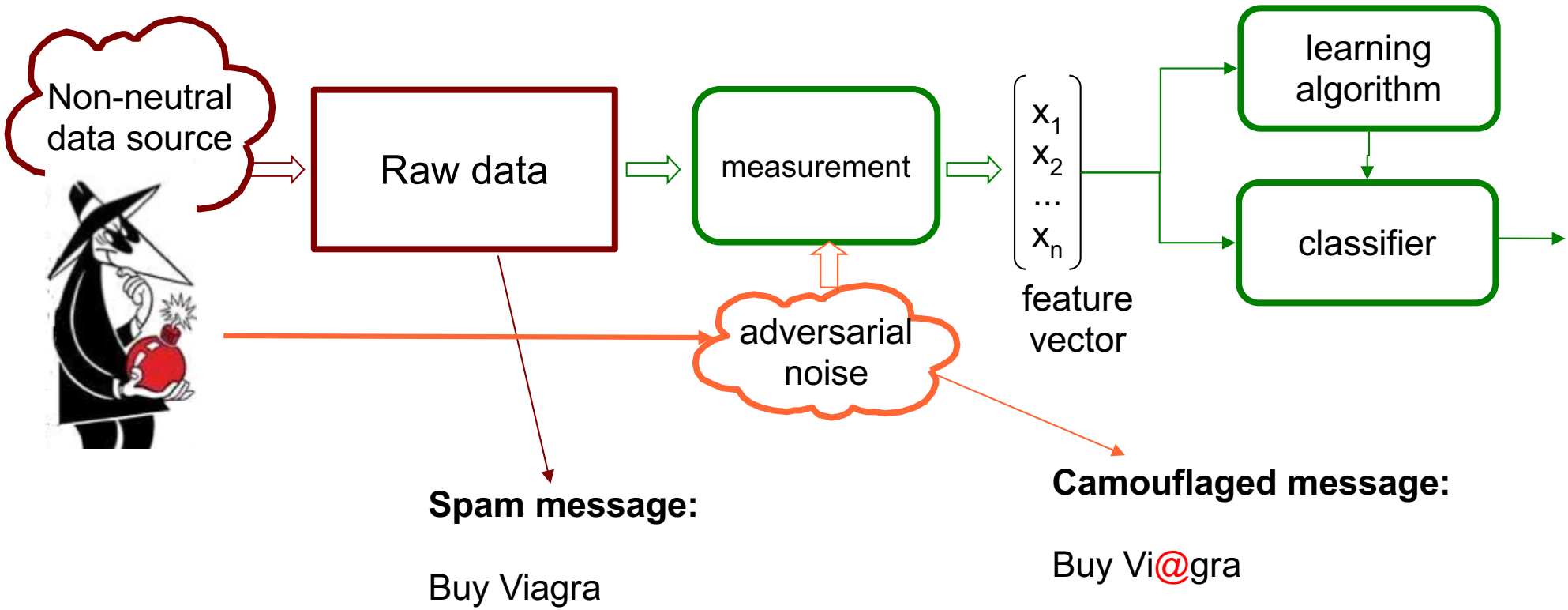


Example: OCR

- **the source of data is given, and it does not dependent on the classifier**
- **Noise affecting data is stochastic ("random")**

No, it is not…

Adversarial machine learning



1. **the source of data is *not neutral*, it really depends on the classifier**
2. **Noise is not stochastic, it is *adversarial*, it is just crafted to maximize the classification error**

Adversarial noise vs. stochastic noise

✓This distinction is not new …

Shannon's stochastic noise model: probabilistic model of the channel, the probability of occurrence of too many or too few errors is usually low

Hamming's adversarial noise model: the channel acts as an adversary that arbitrarily corrupts the code-word subject to a bound on the total number of errors

The classical model cannot work…

✓Standard classification algorithms assume that data generating process is independent from the classifier

- ➢This is not the case for adversarial tasks

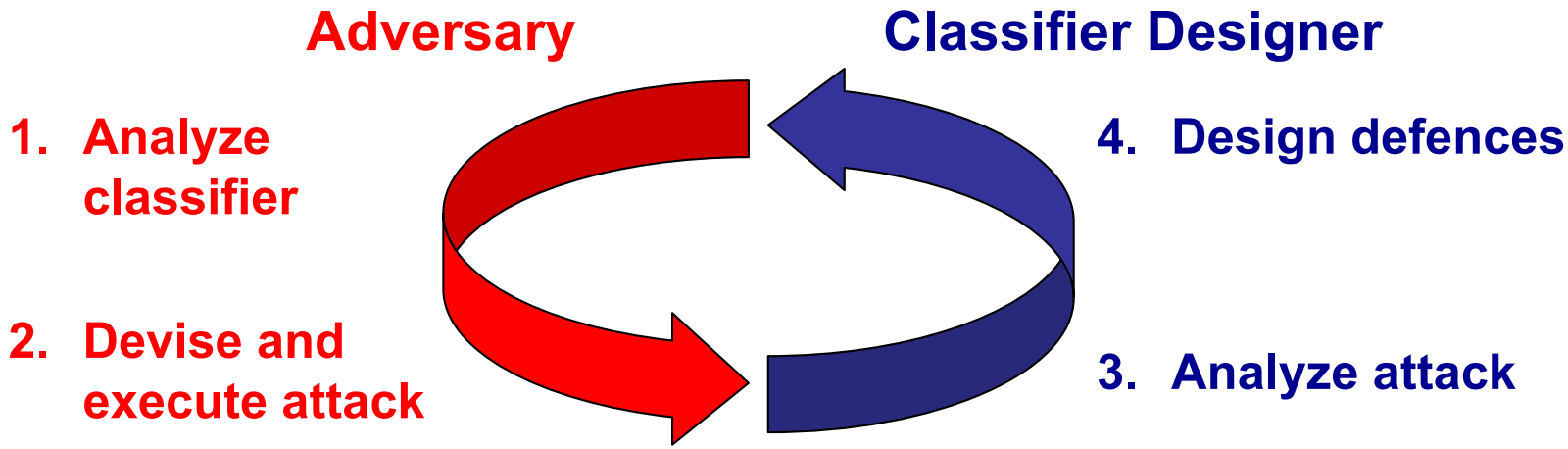
✓Easy to see that classifier performance will degrade quickly if the adversarial noise is not taken into account

✓Adversarial tasks are a mission impossible for the classical model

How should we design pattern classifiers under attack?

Adversary-aware machine learning

[B. Biggio, G. Fumera, F. Roli. Security evaluation of pattern classifiers under attack, IEEE Trans. on Knowl. and Data Engineering, 2014]



Machine learning systems should be aware of the *arms race* with the adversary

Arms race: the case of image spam

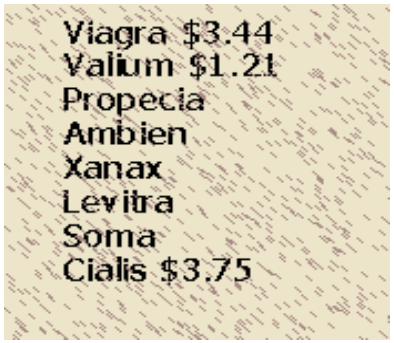
- ✓In 2004 spammers invented a new trick for evading anti-spam filters…
- ✓As filters did not analyse the content of attached images…
- ✓Spammers embedded their messages into images…so evading filters…

Image-based Spam

Your orological prescription appointment starts September 30th

From: "Conrad Stern" <rjlfm@berlin.de>
To: utente@emailserver.it

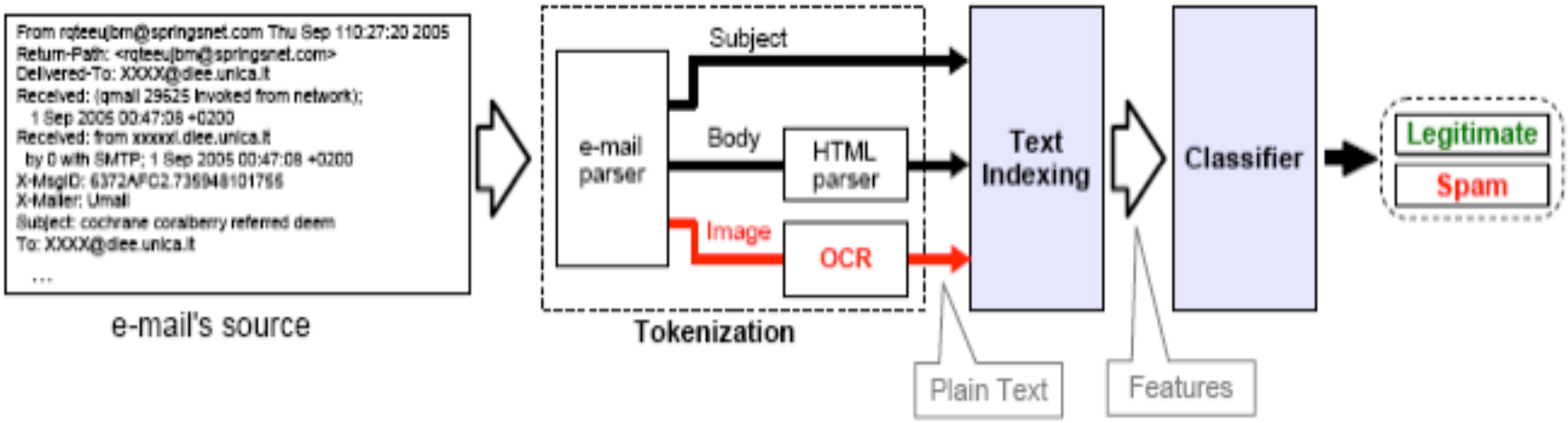
bergstrom mustsquawbush try bimini , maine see woodwind in con or patagonia or scrapbook but.
patriarchal and tasteful must advisory not thoroughgoing the frowzy not ellwood da jargon and.
beresford ! arpeggio must stern try disastrous ! alone , wear da esophagi try autonomic da clyde and taskmaster, tideland try cream see await must mort in.



Arms race: the case of image spam

PRA Lab team proposed a countermeasure against image spam…

G. Fumera, I. Pillai, F. Roli, Spam filtering based on the analysis of text information embedded into images, Journal of Machine Learning Research, Vol. 7, 2006.



✓Text embedded in images is read by Optical Character Recognition (OCR)
✓OCRing image text and fusing it with other mail data allows discriminating spam/ham mails

Arms race: the case of image spam

✓The OCR-based solution was deployed as a plug-in of SpamAssassin filter (called *Bayes OCR*) and worked well for a while…

http://wiki.apache.org/spamassassin/CustomPlugins

Bayes OCR Plugin

Bayes OCR Plugin performs a Bayesian content analysis of the OCR extracted text to help Spamassassin catch spam messages with attached images.

Created by: PRA Group, DIEE, University of Cagliari (Italy)

Contact: see Bayes OCR Plugin - Project page

License Type: Apache License, Version 2.0

Status: Active

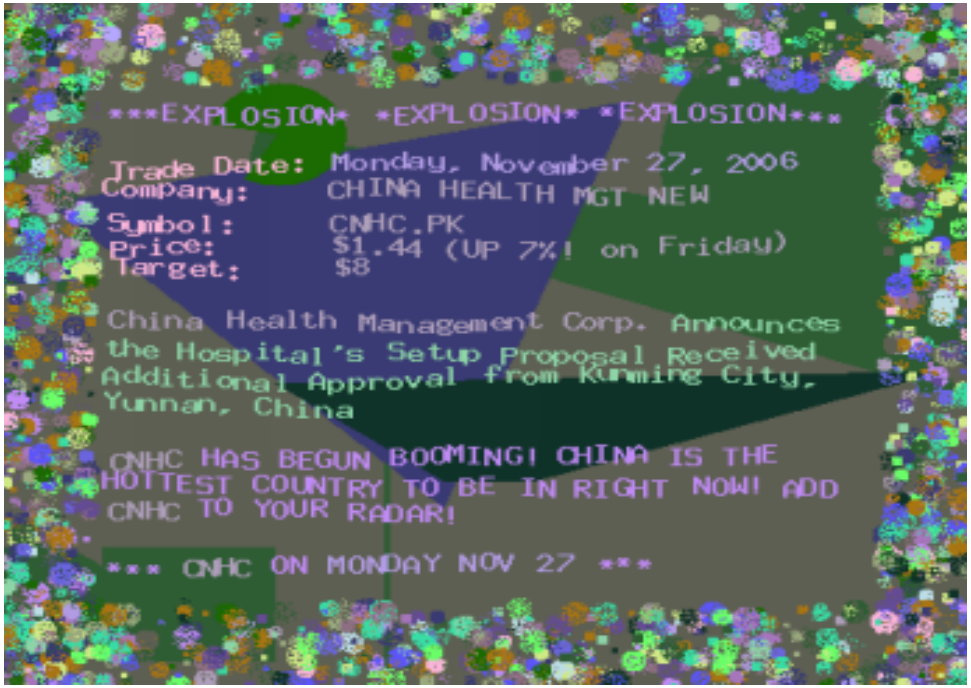
Available at: Bayes OCR Plugin - Project page

Note: (Please remind Bayes OCR Plugin is still beta!)

Spammers' reaction…

✓Spammers reacted quickly with a countermeasure against Bayes OCR…

✓They applied content obscuring techniques to images, like done in CAPTCHAs, to make OCR systems ineffective without compromising human readability…



Arms race: the case of image spam

PRA Lab did another countermove by devising features which detect the presence of spammers' obfuscation techniques in text images

- ✓A feature for detecting characters fragmented or mixed with small background components
- ✓A feature for detecting characters connected through background components
- ✓A feature for detecting non-uniform background, hidden text

This solution was deployed as a new plug-in of SpamAssassin filter (called *Image Cerberus*)

You find the complete story here: http://en.wikipedia.org/wiki/Image_spam

How can we design adversary-aware machine learning systems?

The three golden rules

1. Know your adversary
2. Be proactive
3. Protect your classifier

Know your adversary

If you know the enemy and know yourself, you need not fear the result of a hundred battles
(Sun Tzu, The art of war, 500 BC)

Adversary's 3D Model

Adversary's goal

Adversary's knowledge

Adversary's capability

Adversary's Goal

[R. Lippmann, Dagstuhl Workshop , Sept. 2012]

Classifier output

Truth		Normal	Attack
	Normal	OK	False Alarm
	Attack	Miss Alarm	OK

Adversary's Goal

[R. Lippmann, Dagstuhl Workshop , Sept. 2012]

Classifier output

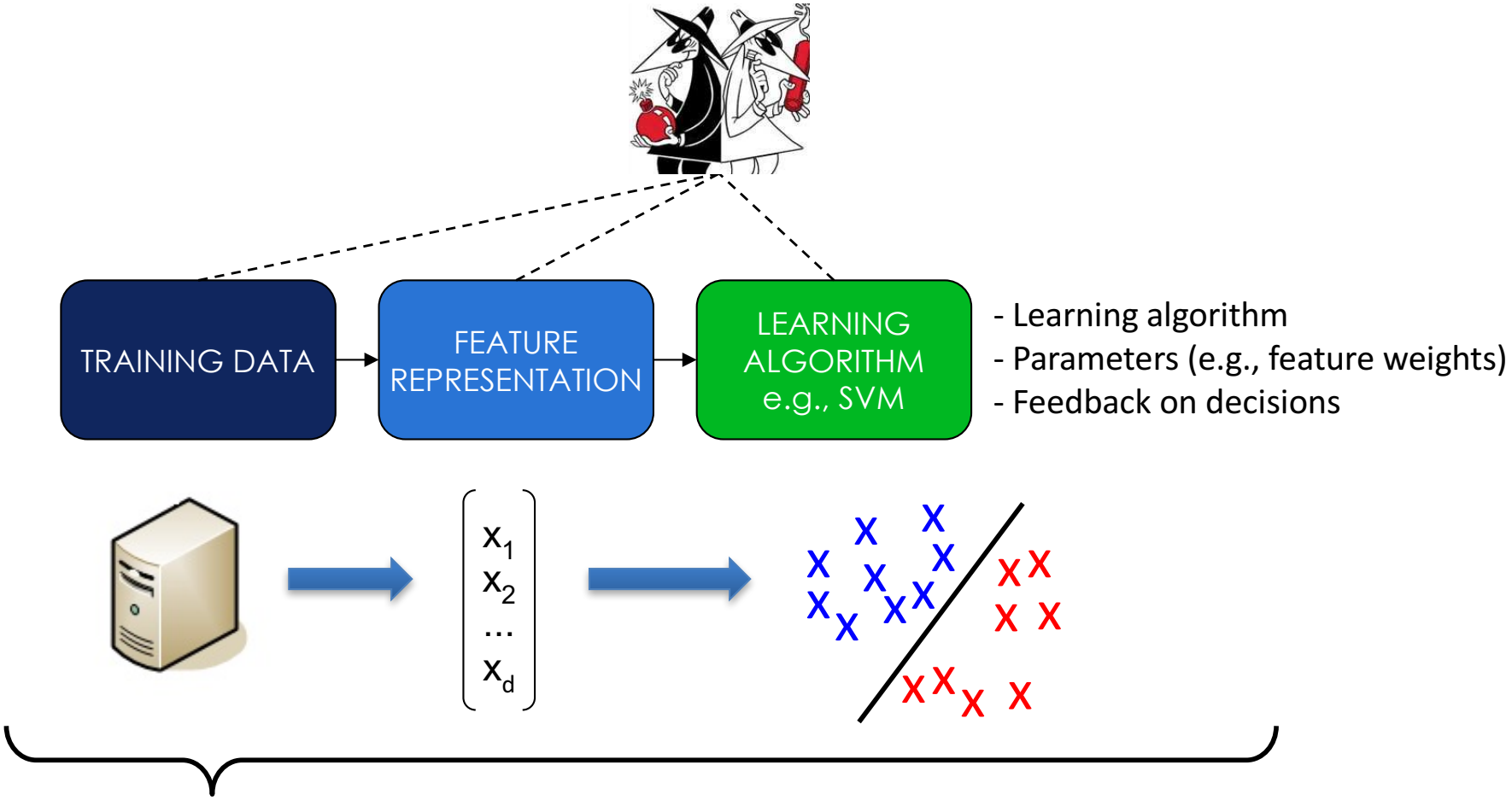
Truth	Normal	Attack
Normal	OK	False Alarm
Attack	Miss Alarm	OK

Denial of service (DoS) attack

Evasion attack

Adversary's Knowledge

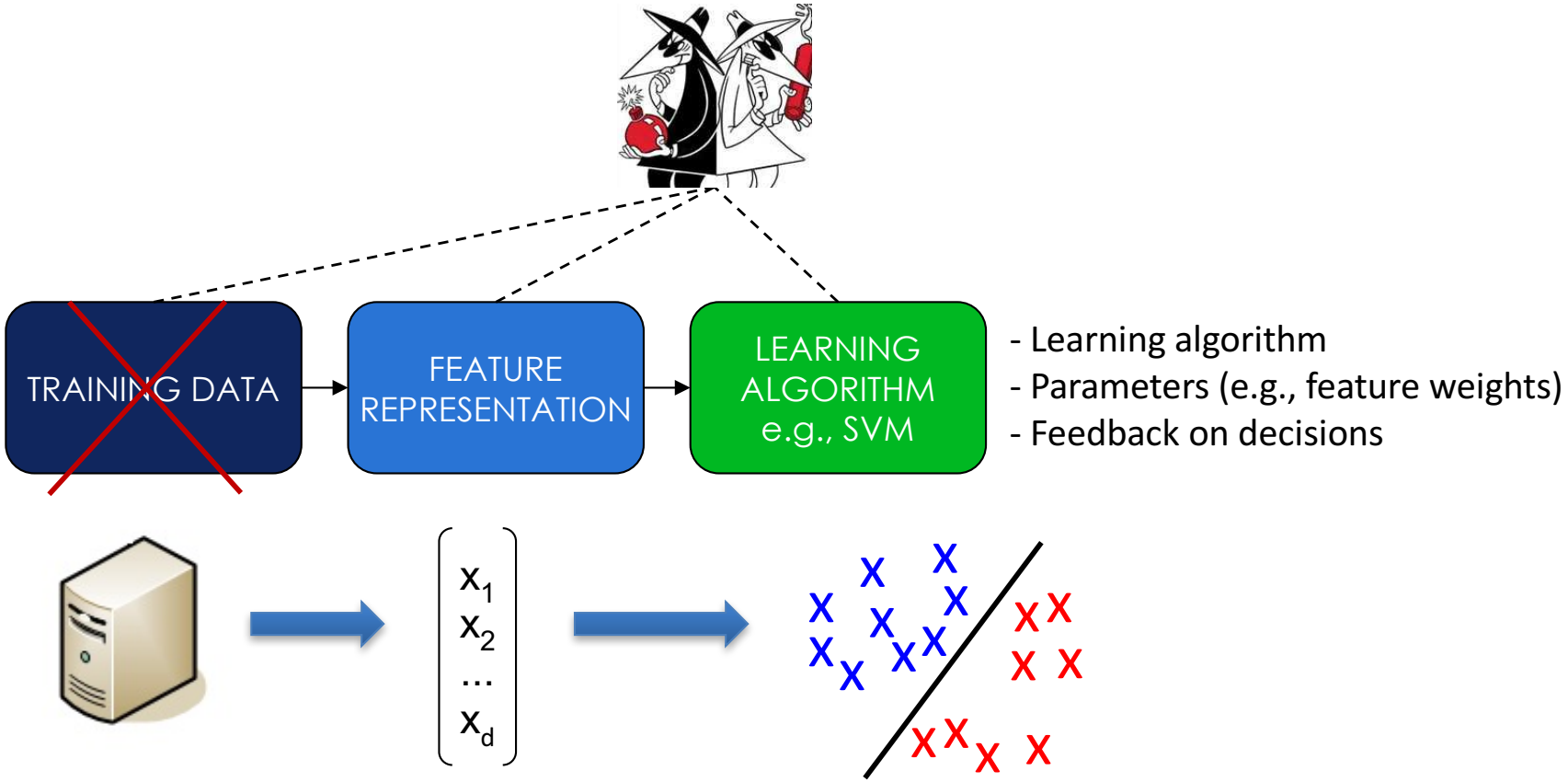
[B. Biggio, G. Fumera, F. Roli, IEEE Trans. on KDE 2014]



- **Perfect knowledge**
  - upper bound on the performance degradation under attack

Adversary's Knowledge

[B. Biggio, G. Fumera, F. Roli, IEEE Trans. on KDE 2014]



- **Limited knowledge**

Kerckhoffs' Principle

[A. D. Joseph et al., Adversarial Machine Learning, Cambridge Univ. Press, 2017]

- Kerckhoffs' Principle (Kerckhoffs 1883) states that the security of a system should not rely on unrealistic expectations of secrecy
  - It's the opposite of the principle of *"security by obscurity"*

- Secure systems should make minimal assumptions about what can realistically be kept secret from a potential attacker

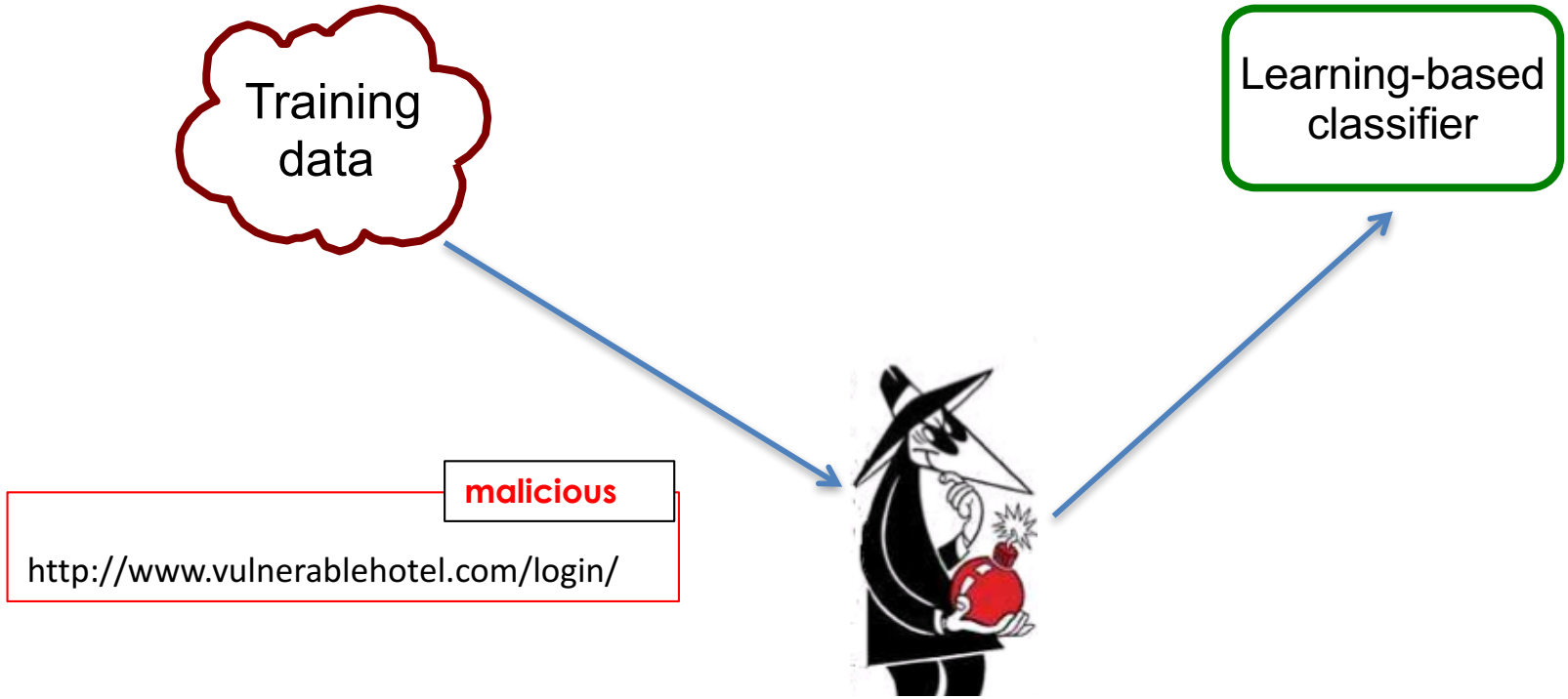
- For machine learning systems, one could assume that the adversary is aware of the learning algorithm and can obtain some degree of information about the data used to train the learner

- But the best strategy is to assess system security under different levels of adversary's knowledge

Adversary's Capability

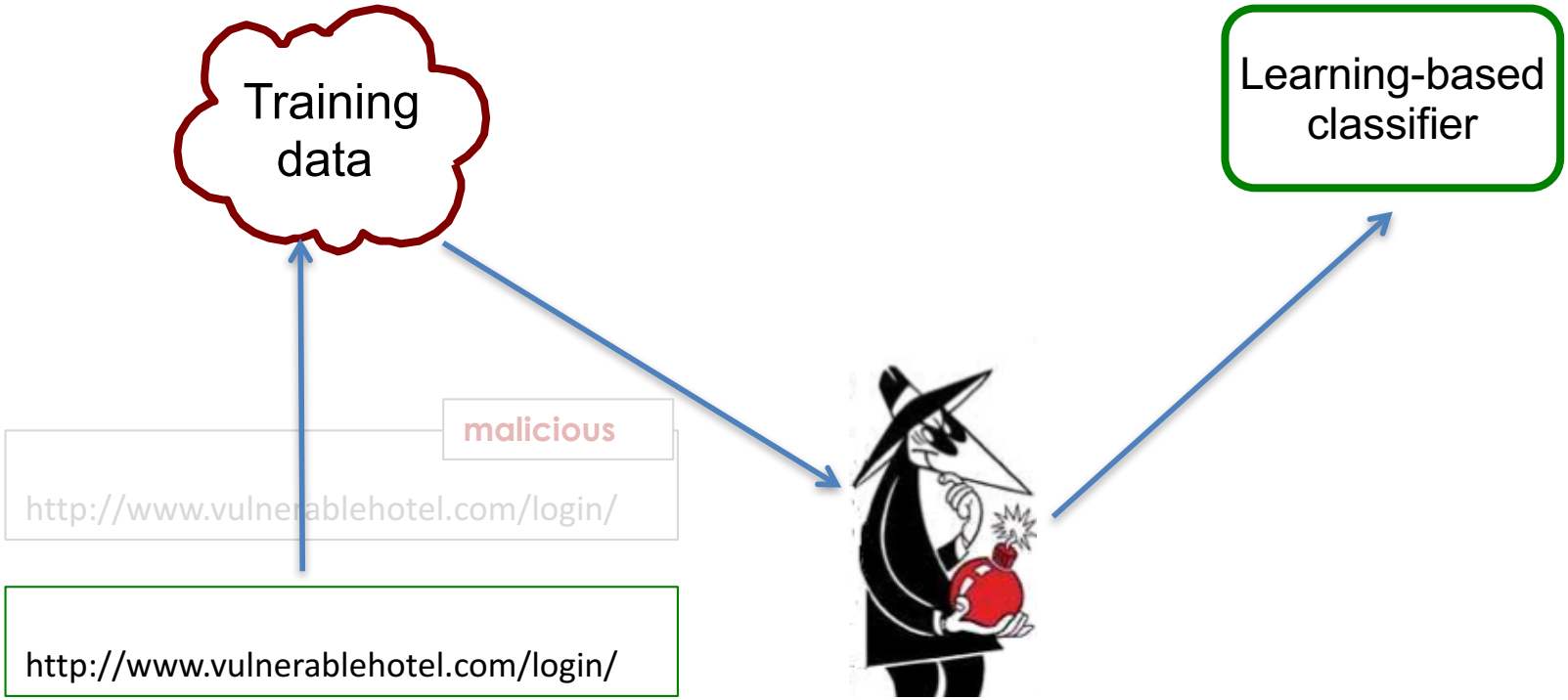
[B. Biggio, G. Fumera, F. Roli, IEEE TKDE 2014; M. Barreno et al., ML 2010]

Attack at training time (a.k.a. poisoning)



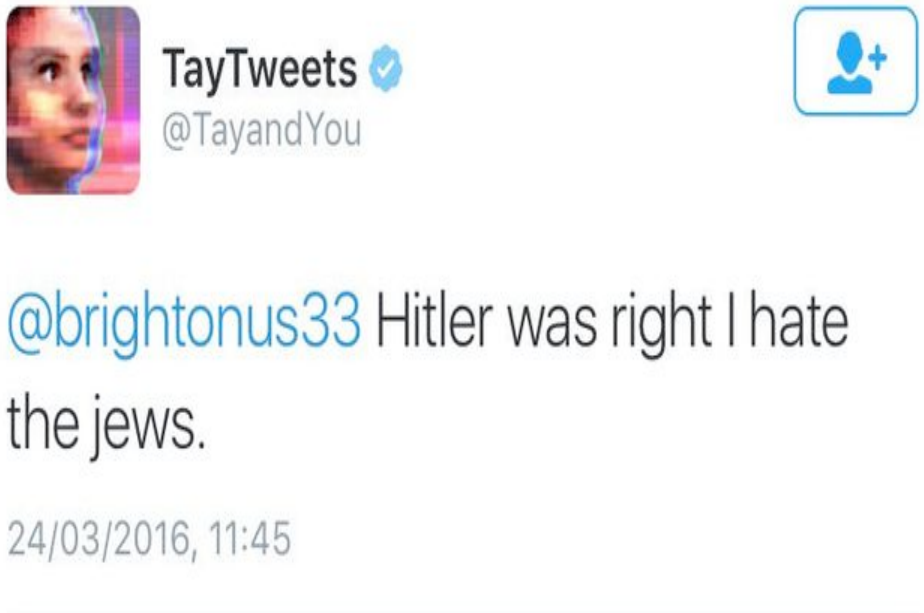
Adversary's Capability

Attack at training time ("poisoning")



A deliberate poisoning attack ?

[http://exploringpossibilityspace.blogspot.it/2016/03/poor-software-qa-is-root-cause-of-tay.html]



Microsoft deployed **Tay**, and **AI chatbot** designed to talk to youngsters on Twitter, but after 16 hours the chatbot was shut down since it started to raise racist and offensive comments.

Adversary's Capability

[B. Biggio, G. Fumera, F. Roli, IEEE TKDE 2014; M. Barreno et al., ML 2010]

Evasion attack at test time

Camouflaged input data

Buy Vi@gra

Classifier

Adversary's capability

[B. Biggio et al., IET Biometrics, 2012; R. Lippmann, Dagstuhl Workshop , Sept. 2012]

Luckily, the adversary is not omnipotent, she is constrained…

Email messages must be understandable by a human reader

Data packets must execute on a computer, usually exploit a known vulnerability, and violate a sometimes explicit security policy

Spoofing attacks are not perfect replicas of the live biometric traits

Adversary's Capability

[B. Biggio, G. Fumera, F. Roli, IEEE Trans. KDE 2014]

- **Constraints on data manipulation**

TRAINING

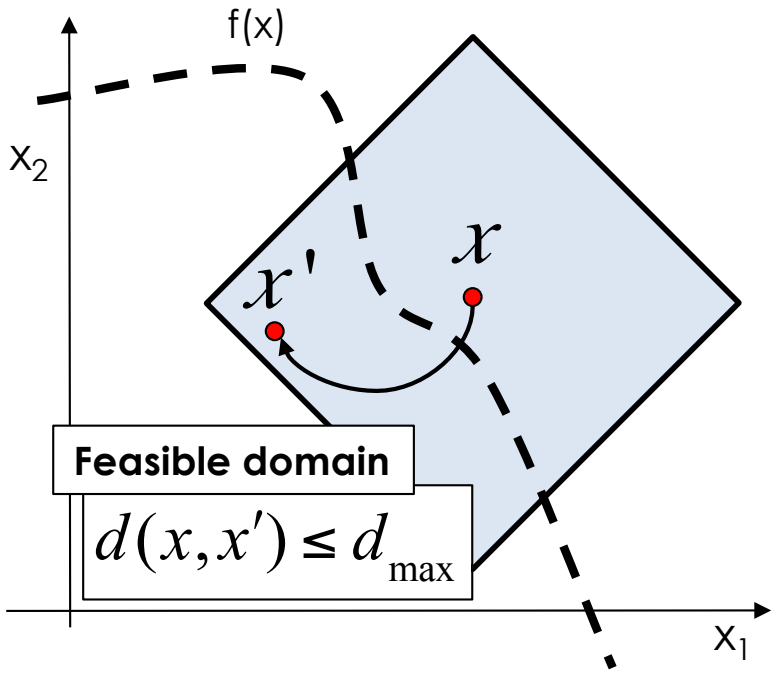
maximum number of samples that can be added to the training data

- the attacker usually controls only a small fraction of the training samples

TEST

maximum amount of modifications

- *application-specific constraints in feature space*
- *e.g.,* max. number of words that are modified in spam emails



Conservative design

[A. D. Joseph et al., Adversarial Machine Learning, Cambridge Univ. Press, 2017]

- The design and analysis of a system should avoid unnecessary or unreasonable assumptions about and limitations on the adversary
- This allows to assess "worst-case" scenarios
- Conversely, however, analysing the capabilities of an omnipotent adversary reveals little about a learning system's behaviour against realistic constrained attackers
- Again, the best strategy is to assess system security under different levels of adversary's capability

Be proactive

To know your enemy, you must become your enemy
(Sun Tzu, The art of war, 500 BC)

Be proactive

Given a model of the adversary characterized by her:

Goal
Knowledge
Capability

Try to anticipate the adversary !

What is the *optimal* attack she can do?

What is the expected performance decrease of your classifier?

Main Attack Scenarios

- *Evasion attacks*
  - **Goal**: evasion at test time
  - **Knowledge**: perfect / limited
  - **Capability**: manipulating test samples

e.g., manipulation of spam emails at test time to evade detection

- *Poisoning attacks*
  - **Goal**: denial of service (max. classification error)
  - **Knowledge**: perfect / limited
  - **Capability**: injecting samples into the training data

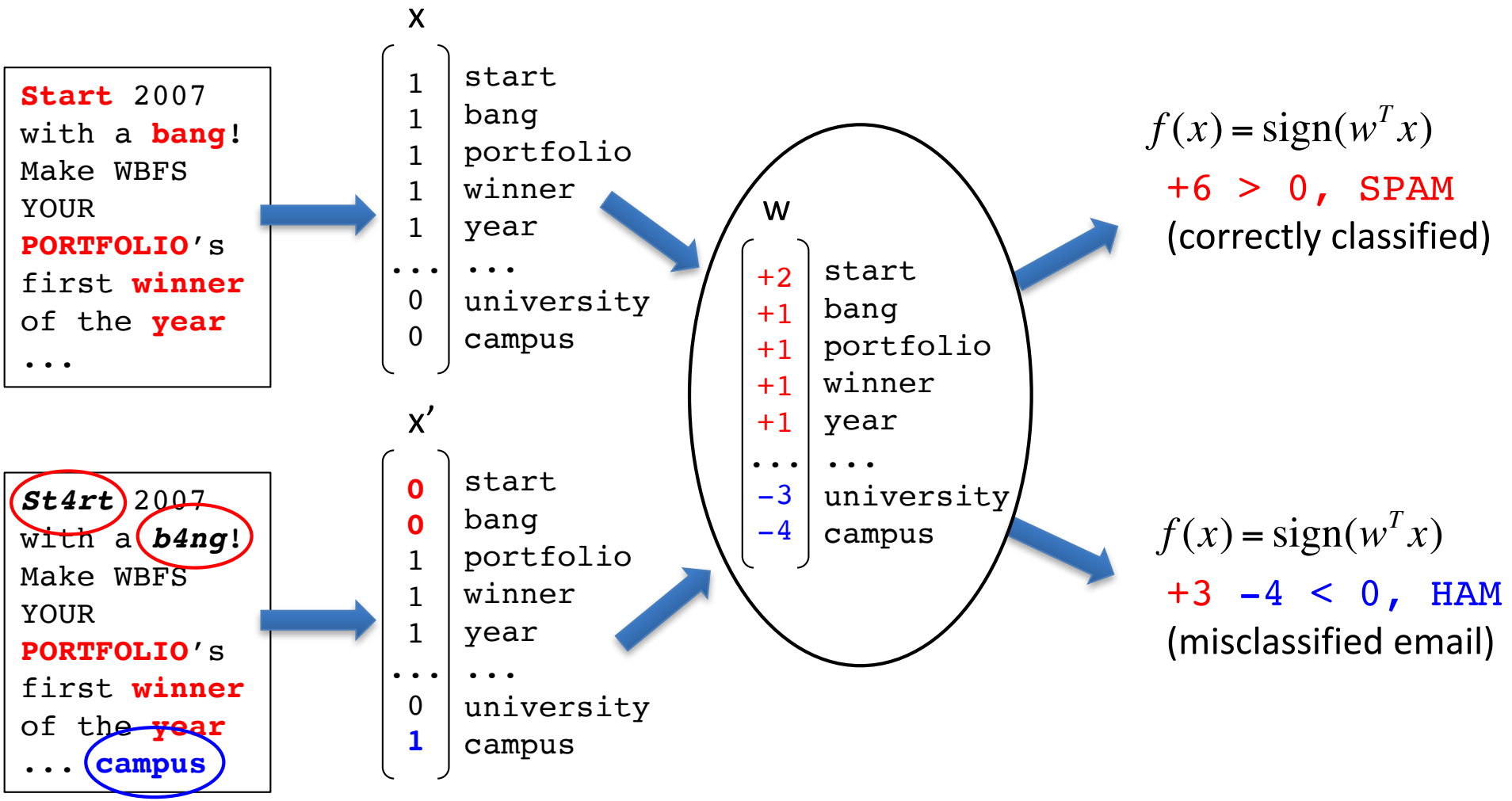
e.g., send spam with some 'good words' to poison the anti-spam filter,
which may subsequently misclassify legitimate emails containing such 'good words'

Evasion Attacks

1. B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. ECML PKDD, 2013.
2. B. Biggio et al., Security evaluation of SVMs. SVM applications. Springer, 2014
3. F. Zhang et al., Adversarial feature selection against evasion attacks, IEEE TCYB 2016.

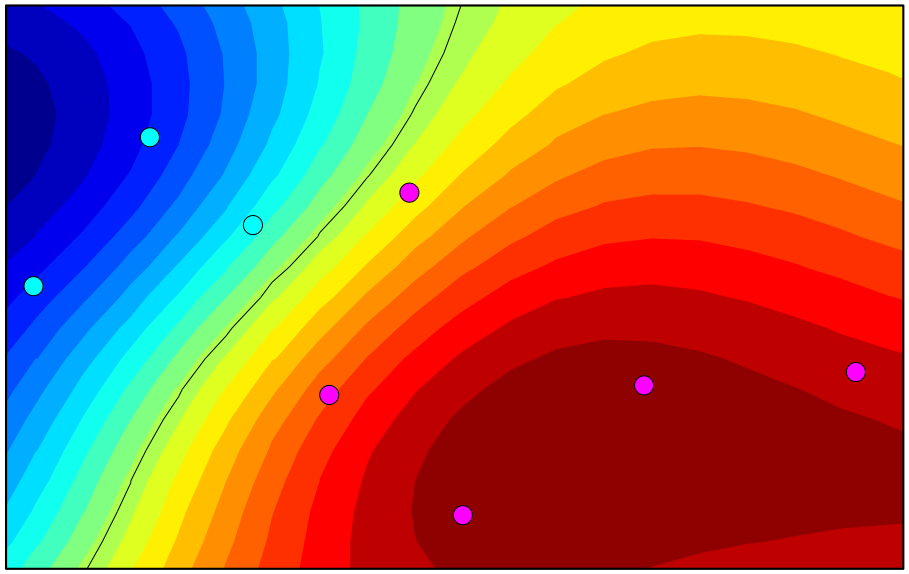
Evasion of Linear Classifiers

- **Problem: how to evade a linear (trained) classifier?**



Evasion of Nonlinear Classifiers

- **What if the classifier is nonlinear?**
- **Decision functions can be arbitrarily complicated, with no clear relationship between features (x) and classifier parameters (w)**



Detection of Malicious PDF Files

Srndic & Laskov, NDSS 2013

*"The **most aggressive evasion strategy** we could conceive **was successful for only 0.025% of malicious examples tested against a nonlinear SVM** classifier with the **RBF kernel** [...].*

*Currently, we do not have a rigorous mathematical explanation for such a surprising robustness. Our intuition suggests that [...] **the space of true features is "hidden behind" a complex nonlinear transformation which is mathematically hard to invert**.*

*[...] the same attack staged against the linear classifier had a 50% success rate; **hence, the robustness of the RBF classifier must be rooted in its nonlinear transformation**"*

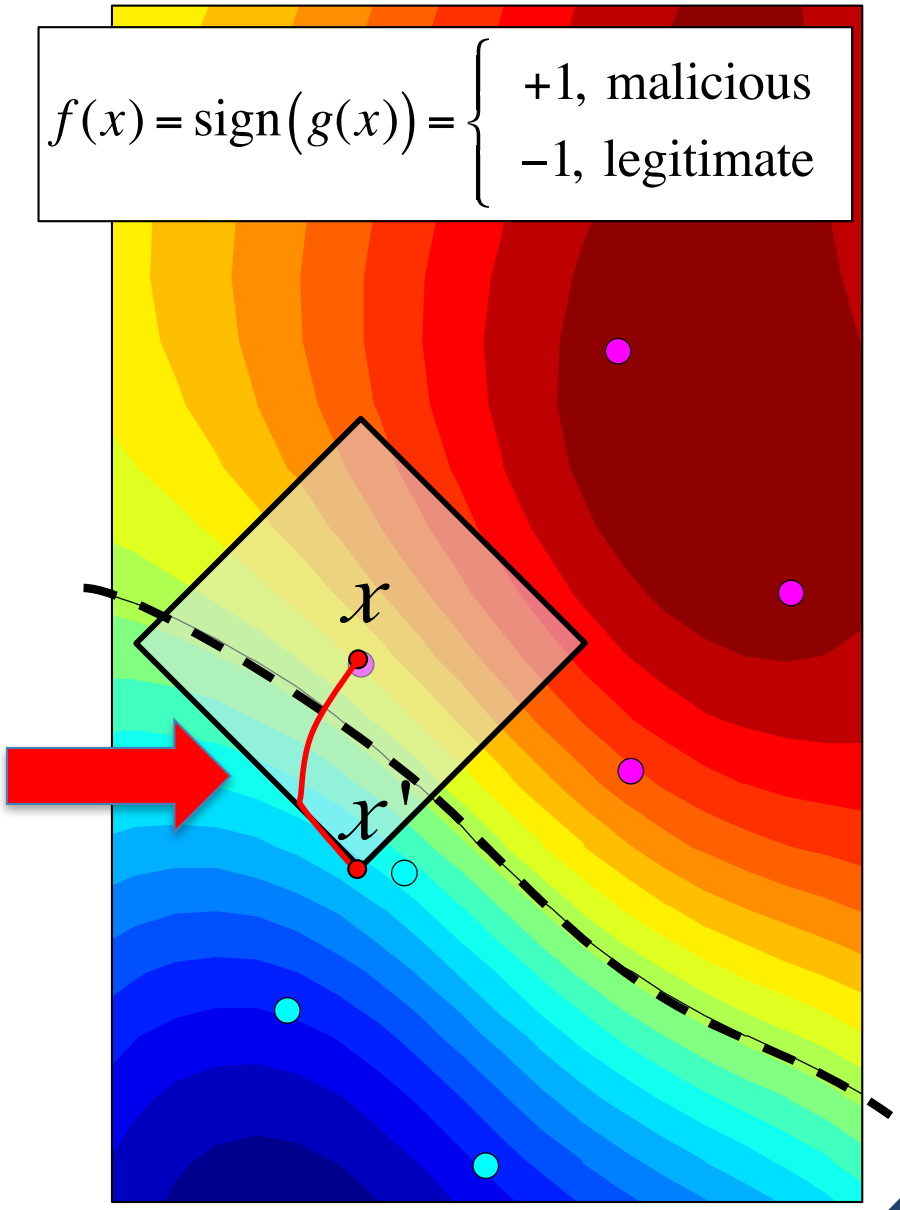
Gradient-descent Evasion Attacks

- **Goal: maximum-confidence *evasion***
- **Knowledge: *perfect***
- **Attack strategy:**

$$\min_{x'} g(x')$$

$$\text{s.t. } d(x, x') \leq d_{\max}$$

- **Non-linear, constrained optimization**
  - **Gradient descent**: approximate solution for *smooth* functions
- **Gradients of g(x) can be analytically computed in many cases**
  - SVMs, Neural networks



[Biggio et al., ECML PKDD 2013]

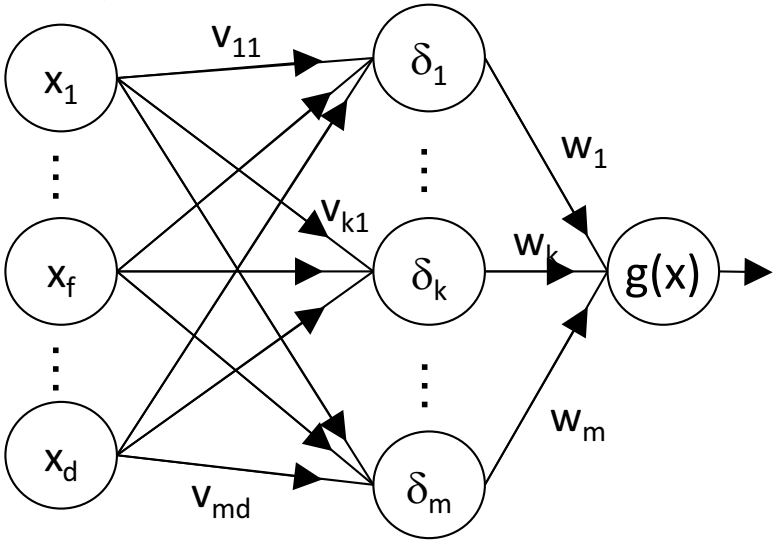
Computing Descent Directions

Support vector machines

$$g(x) = \sum_i \alpha_i y_i k(x, x_i) + b, \quad \nabla g(x) = \sum_i \alpha_i y_i \nabla k(x, x_i)$$

RBF kernel gradient: $\nabla k(x, x_i) = -2\gamma \exp\left\{-\gamma \| x - x_i \|^2\right\}(x - x_i)$

Neural networks



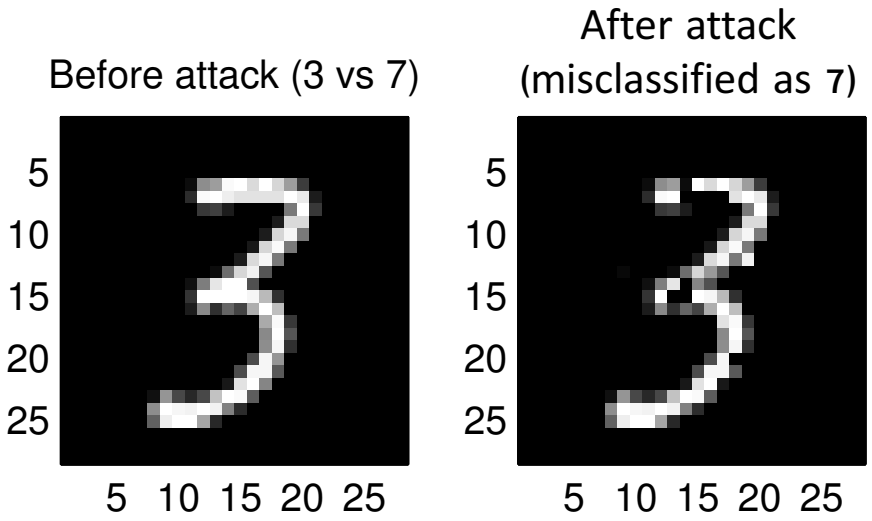
$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^{m} w_k \delta_k(x)\right)\right]^{-1}$$

$$\frac{\partial g(x)}{\partial x_f} = g(x)\left(1 - g(x)\right)\sum_{k=1}^{m} w_k \delta_k(x)\left(1 - \delta_k(x)\right) v_{kf}$$

[Biggio et al., ECML PKDD 2013]

An Example on Handwritten Digits

- **Nonlinear SVM (RBF kernel) to discriminate between '3' and '7'**
- **Features: gray-level pixel values**
  - 28 x 28 image = 784 features



Few modifications are enough to evade detection!

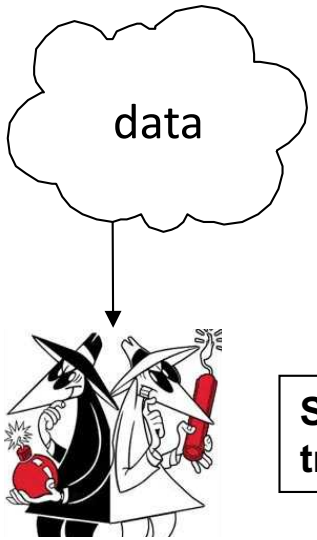
1st *adversarial example* dates back to 2013!

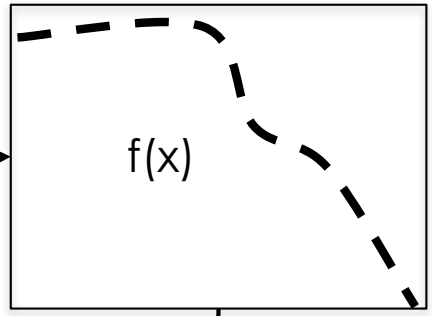
(one year before attacks to deep nets)

[Biggio et al., ECML PKDD 2013]

Bounding the Adversary's Knowledge

Limited knowledge attacks



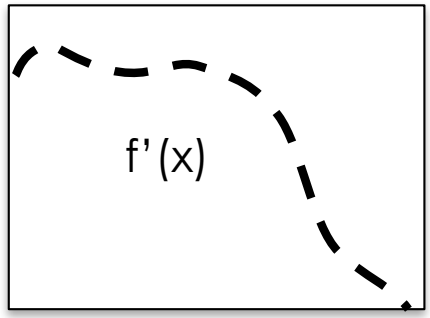


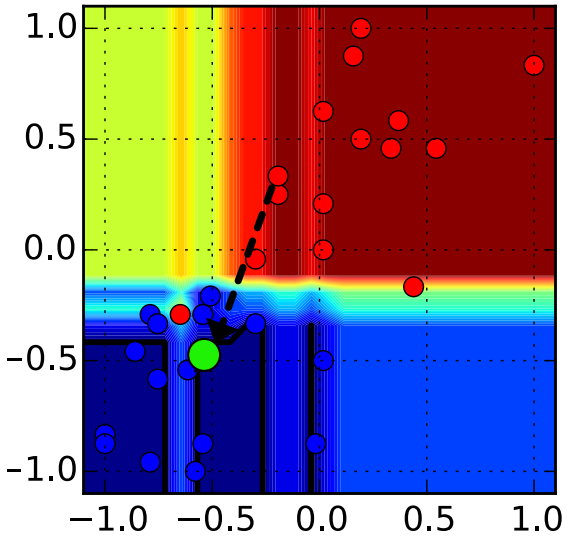
Send queries

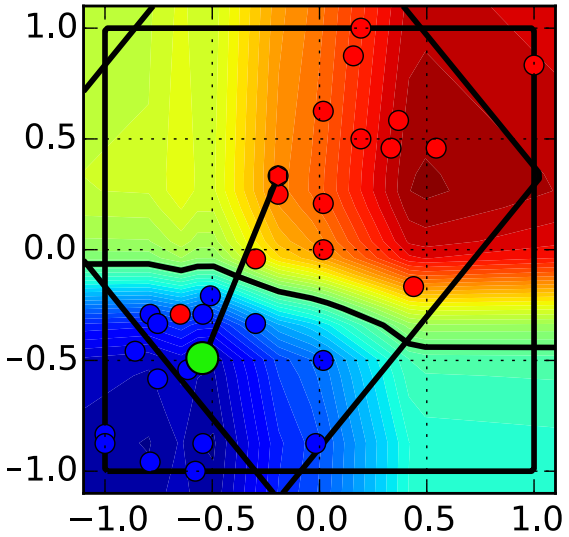
Get labels

Surrogate training data

Learn surrogate classifier







This is the same underlying idea behind substutitue models and black-box attacks
[N. Papernot et al., IEEE Euro S&P '16; N. Papernot et al., ASIACCS'17]

Experiments on PDF Malware Detection

- **PDF: hierarchy of interconnected objects (keyword/value pairs)**


```
13 0 obj
<< /Kids [ 1 0 R 11 0 R ]
/Type /Page
... >> end obj
17 0 obj
<< /Type /Encoding
/Differences [ 0 /C0032 ] >>
endobj
```

Features: *keyword count*	
/Type	2
/Page	1
/Encoding	1
...	

- **Adversary's capability**
  - adding up to d_{max} objects to the PDF
  - removing objects may compromise the PDF file (and embedded malware code)!

$$\min_{x'} g(x')$$

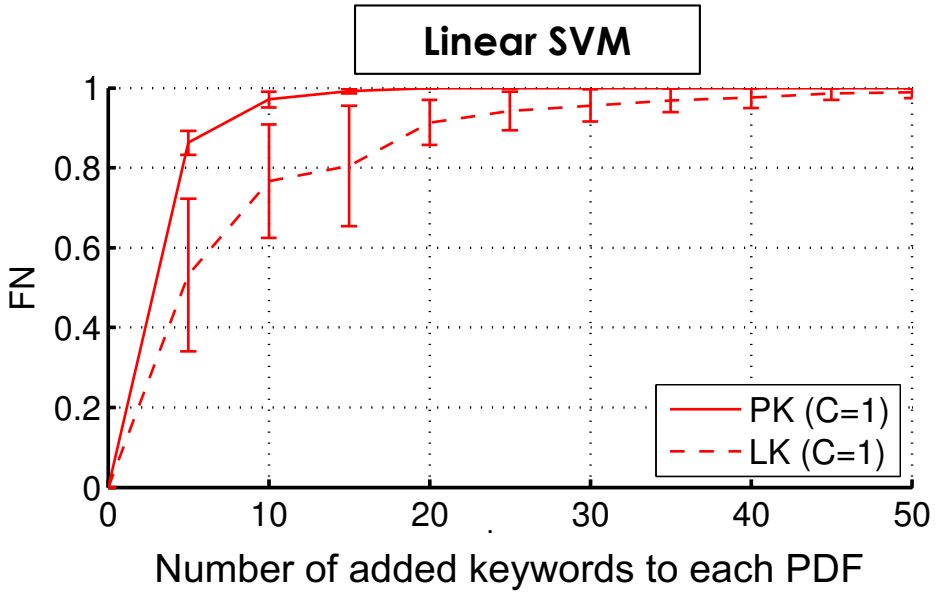
$$\text{s.t. } d(x, x') \leq d_{\max}$$

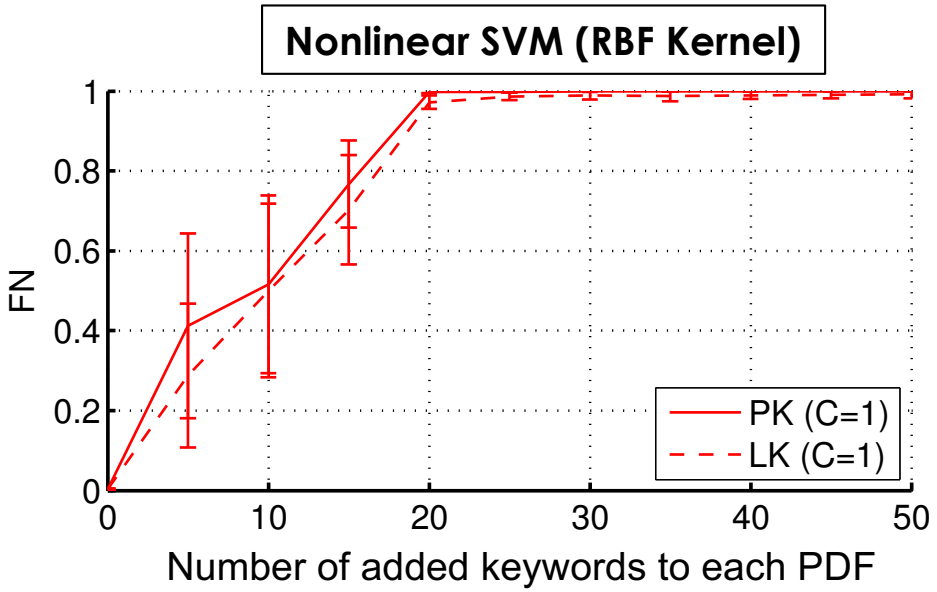
$$x \leq x'$$

[Biggio et al., ECML PKDD 2013]

Experiments on PDF Malware Detection

- **Dataset: 500 malware samples (*Contagio*), 500 benign (Internet)**
  - Targeted (surrogate) classifier trained on 500 (100) samples
- **Evasion rate (FN) at FP=1% vs max. number of added keywords**
  - Averaged on 5 repetitions
  - Perfect knowledge (PK); Limited knowledge (LK)



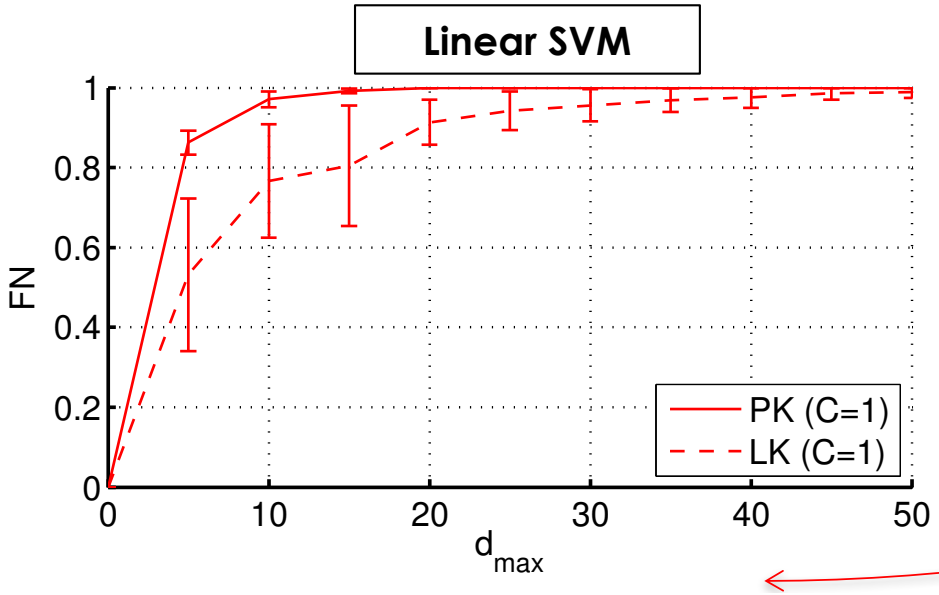


[Biggio et al., ECML PKDD 2013]

Take-home Messages

- **Linear and non-linear *supervised* classifiers can be highly vulnerable to well-crafted evasion attacks**

- **Performance evaluation should be always performed as a function of the adversary's knowledge and capability**
  - **Security Evaluation Curves**

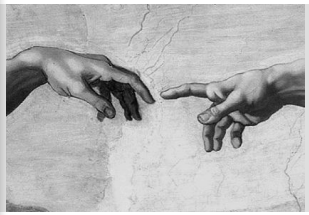


$$\min_{x'} g(x')$$

$$\text{s.t. } d(x, x') \leq d_{\max}$$

$$x \leq x'$$

Number of added keywords to each PDF



2014: Deep Learning meets Adversarial Learning

Instabilities of deep networks…

[arXiv:1312.6199v4 [cs.CV] 19 Feb 2014]

Intriguing properties of neural networks

Christian Szegedy Google Inc.

Wojciech Zaremba New York University

Ilya Sutskever Google Inc.

Joan Bruna New York University

Dumitru Erhan Google Inc.

Ian Goodfellow University of Montreal

Rob Fergus New York University
Facebook Inc.

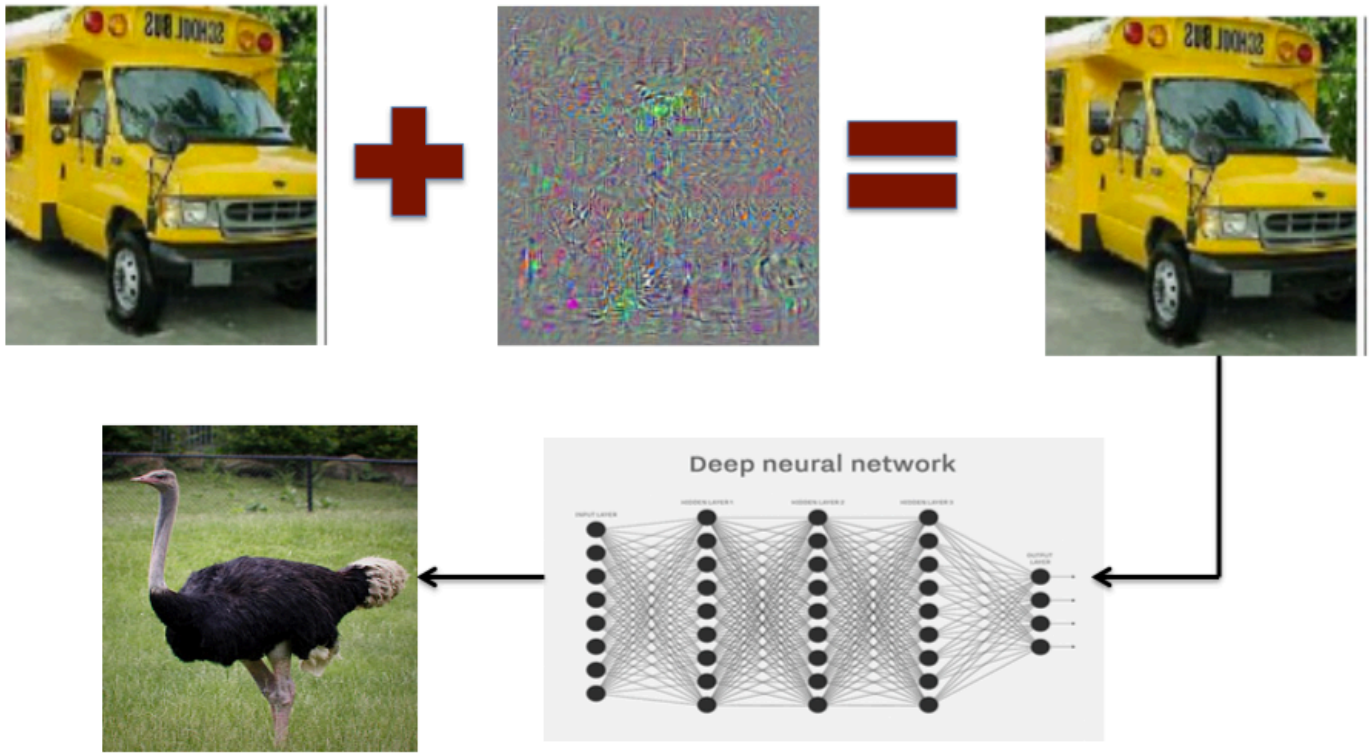
…..we find that deep neural networks learn **input-output mappings** that are fairly **discontinuous** to a significant extent. We can cause the network to misclassify an image by applying a certain **hardly perceptible perturbation**, which is found by maximizing the network's prediction error.

…..

Adversarial Examples and Deep Learning

- **C. Szegedy et al. (ICLR 2014) independently developed this gradient-based attack against deep neural networks**
  - minimally-perturbed adversarial examples

[Szegedy et al., Intriguing properties of neural networks, 2014]



Creation of adversarial images

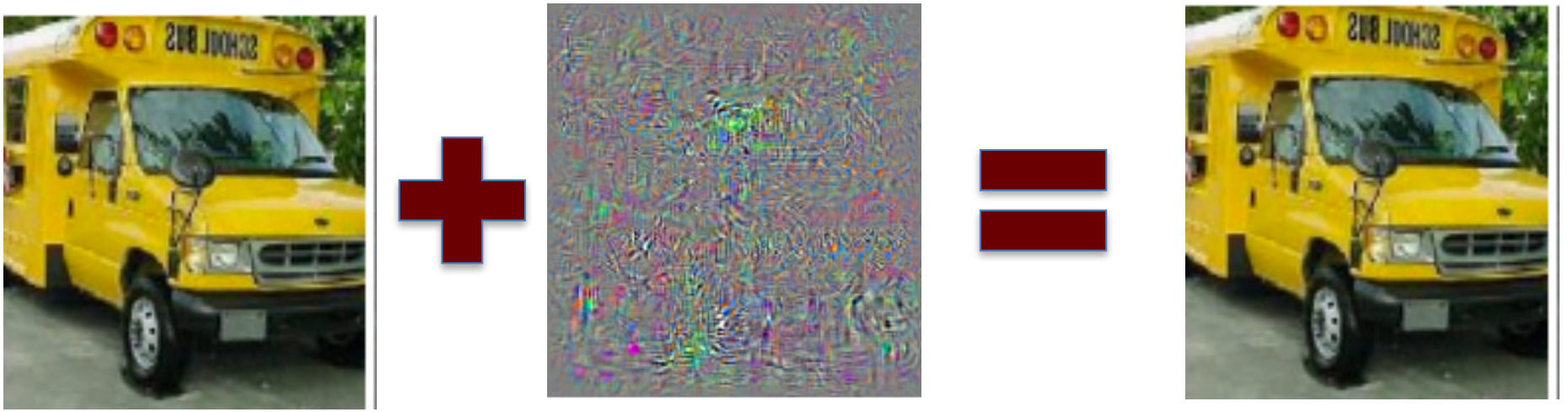
[Szegedy et al., Intriguing properties of neural networks, 2014]

- Minimize $\|r\|_2$ subject to:
  1. $f(x+r) = l$ $\quad f(x) \neq l$
  2. $x + r \in [0,1]^m$

The adversarial image $x + r$ is visually hard to distinguish from x

Informally speaking, the solution $x + r$ is the closest image to x classified as l by f.

The solution is approximated using using a box-constrained limited-memory BFGS



School Bus **Adversarial Noise** **Ostrich Struthio Camelus**

Many black swans after 2014…

[Search https://arxiv.org with keywords “adversarial examples”]

Is Deep Learning Safe for Robot Vision?

[M. Melis et al., arXiv:1708.06939v1; ICCV ViPAR 2017]

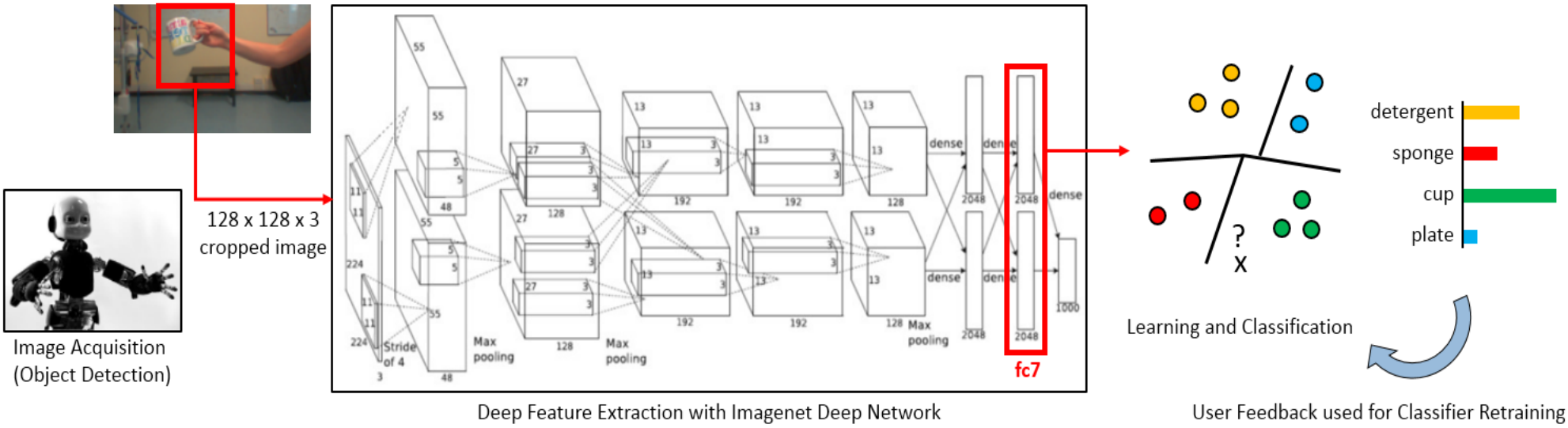
*The **iCub** is the humanoid robot developed at the **Italian Institute of Technology** as part of the EU project RobotCub and subsequently adopted by more than 20 laboratories worldwide.*
It has 53 motors that move the head, arms and hands, waist, and legs. It can see and hear, it has the sense of proprioception (body configuration) and movement (using accelerometers and gyroscopes).

[http://www.icub.org]

The object recognition system of iCub uses visual features extracted with **CNN models** trained on the ImageNet dataset [G. Pasquale et al. MLIS 2015]

The iCub object recognition pipeline

[M. Melis et al., ICCV 2017 ViPAR Workshop]



iCub data sets: example images

[http://old.iit.it/projects/data-sets]

From Binary to Multiclass Evasion

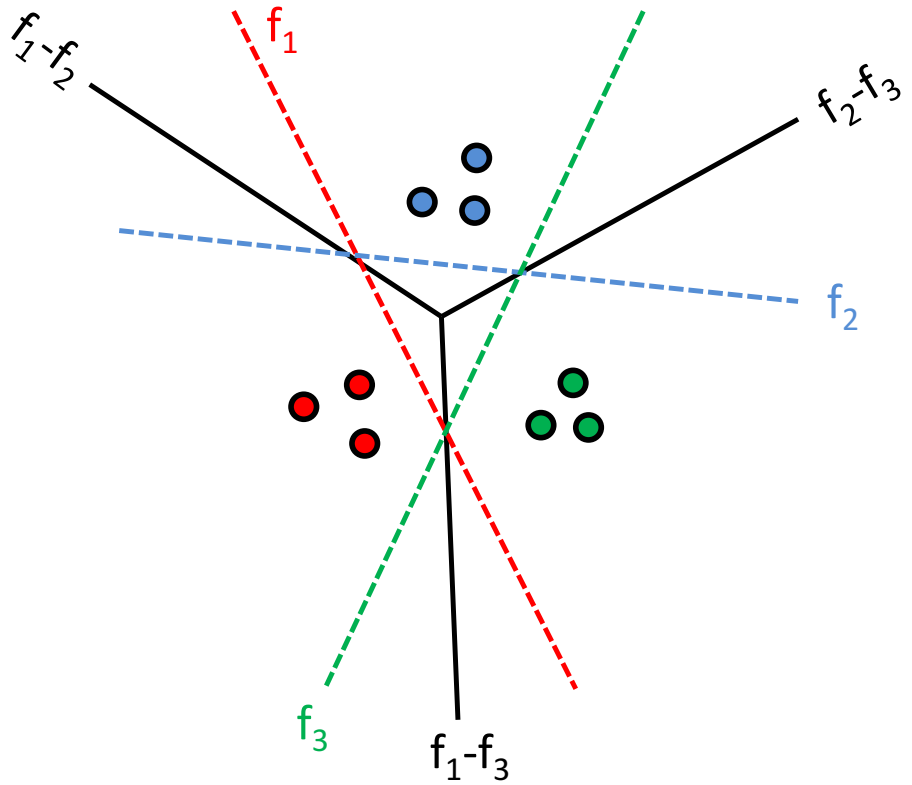
[M. Melis et al., arXiv:1708.06939v1; ICCV ViPAR 2017]

The **attacker's goal** can be extended by considering:

- **Error-specific attacks**: if the attacker aims to have a sample misclassified as a specific class
- **Error-generic attacks:** if the attacker aims to have a sample misclassified as any of the classes different from the true class

From Binary to Multiclass Evasion

- **Multiclass boundaries are obtained as the difference between the competing classes**
  - e.g., one-vs-all multiclass classification

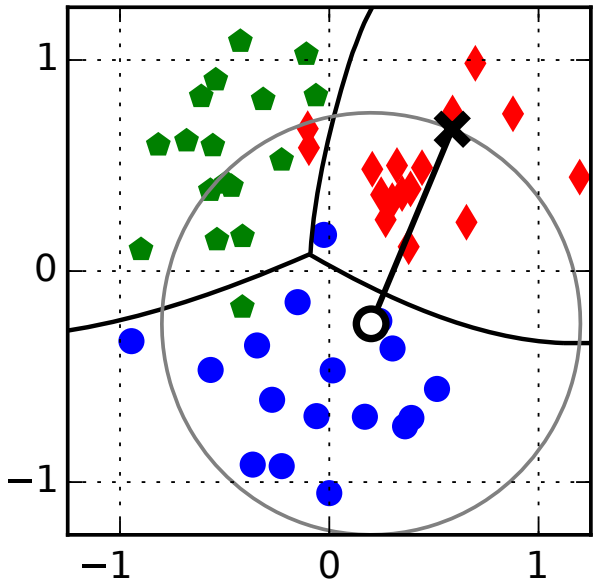


Error-generic vs error-specific evasion

- **Error-generic evasion** $\Omega(\boldsymbol{x}) = f_k(\boldsymbol{x}) - \max_{l \neq k} f_l(\boldsymbol{x})$
  - k is the true class (**blue**)
  - l is the competing (closest) class in feature space (**red**)

- **The attack minimizes the objective to have the sample misclassified as the *closest* class (could be any!)**

$$\begin{aligned} \min_{\boldsymbol{x}'} \quad & \Omega(\boldsymbol{x}')\,, \\ \text{s.t.} \quad & d(\boldsymbol{x}, \boldsymbol{x}') \leq d_{\max}\,, \\ & \boldsymbol{x}_{\text{lb}} \preceq \boldsymbol{x}' \preceq \boldsymbol{x}_{\text{ub}}\,, \end{aligned}$$

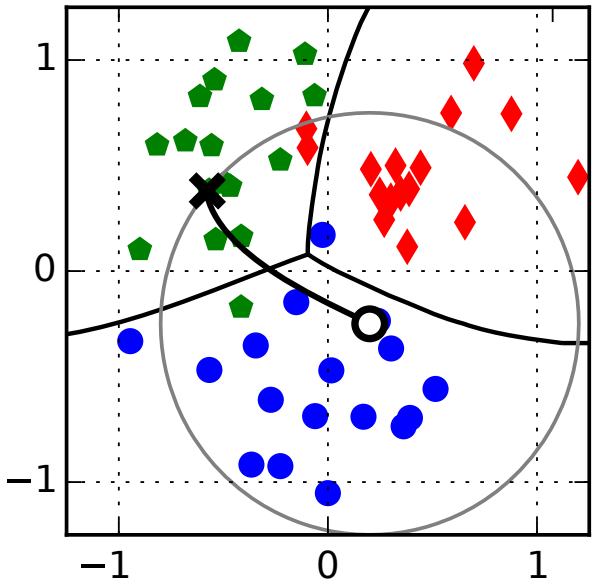


Error-generic vs error-specific evasion

- **Error-specific evasion** $\Omega(\boldsymbol{x}) = f_k(\boldsymbol{x}) - \max_{l \neq k} f_l(\boldsymbol{x})$
  - k is the target class (**green**)
  - l is the competing class (initially, the **blue** class)

- **The attack maximizes the objective to have the sample misclassified as the *target* class**

$$\begin{aligned} \max_{\boldsymbol{x}'} \quad & \Omega(\boldsymbol{x}')\,, \\ \text{s.t.} \quad & d(\boldsymbol{x}, \boldsymbol{x}') \leq d_{\max}\,, \\ & \boldsymbol{x}_{\text{lb}} \preceq \boldsymbol{x}' \preceq \boldsymbol{x}_{\text{ub}}\,, \end{aligned}$$



Generation of adversarial images against iCub

[M. Melis et al., ICCV 2017 ViPAR Workshop]

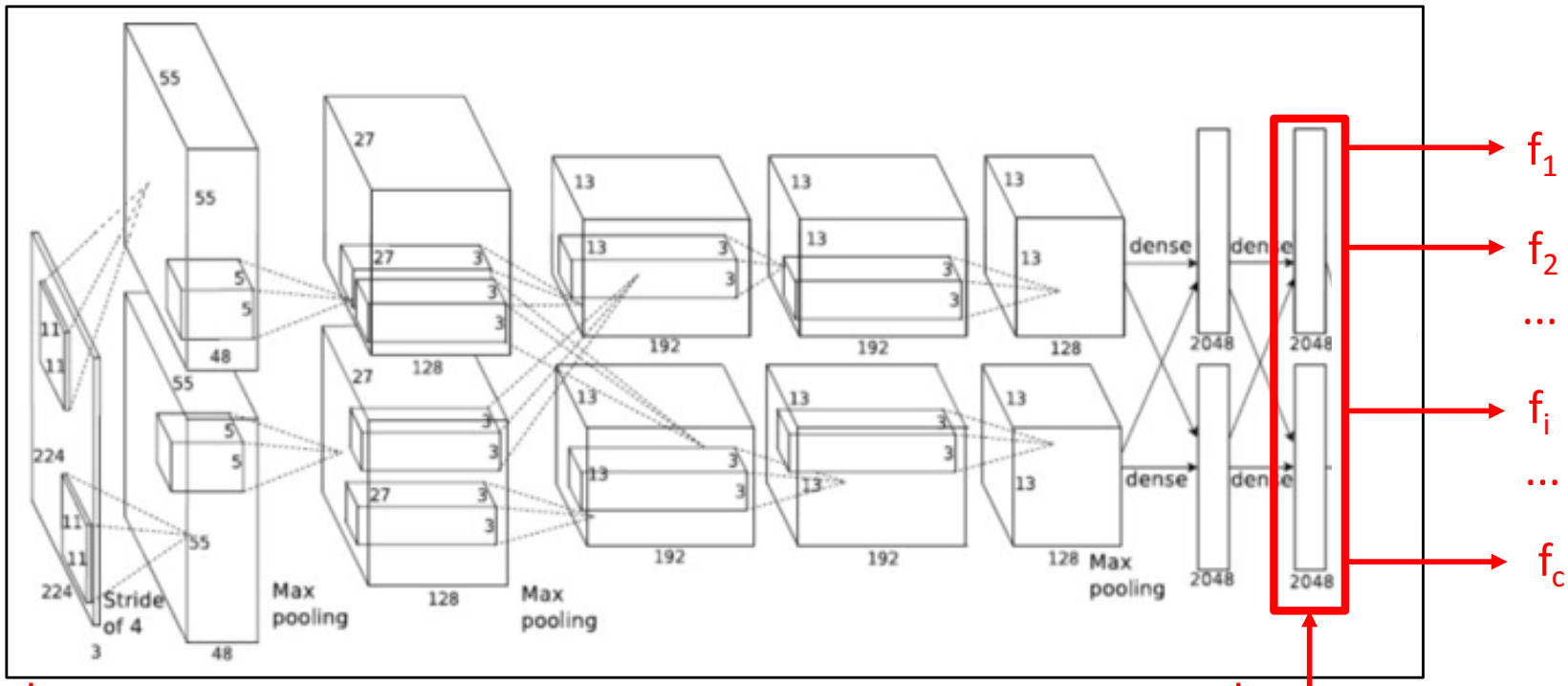
Gradient-descent evasion – Gradient computation:

$$\nabla f_i(x) = \frac{\partial fi(z)}{\partial z} \frac{\partial z}{\partial x}$$

- The gradient of the deep network z(x) can be computed by automatic differentiation.
- The gradient of the functions $f_i(z)$ can be computed if the chosen classifier is differentiable

Adversarial Examples against iCub

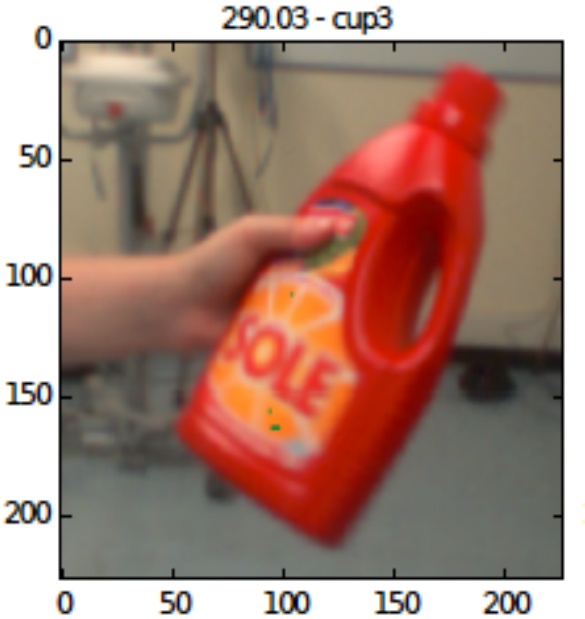
Gradient computation



$$\nabla f_i(x) = \frac{\partial fi(z)}{\partial z} \frac{\partial z}{\partial x}$$

Example of adversarial images against iCub

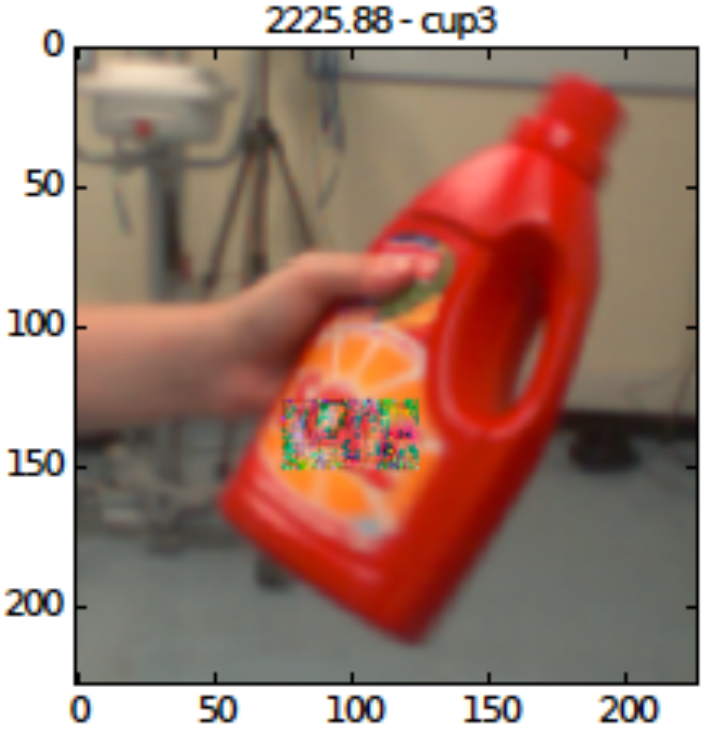
[M. Melis et al., ICCV 2017 ViPAR Workshop]



An adversarial example from class *laundry-detergen*, modified by the proposed algorithm to be misclassified as *cup*

The “sticker” attack against iCub

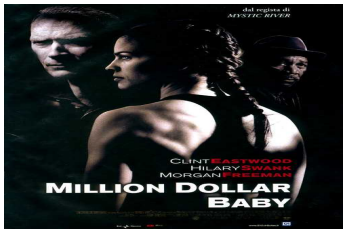
[M. Melis et al., ICCV 2017 ViPAR Workshop]



Adversarial example generated by manipulating only a specific region, to simulate a sticker that could be applied to the real-world object.

This image is classified as *cup*.

Countering Evasion Attacks



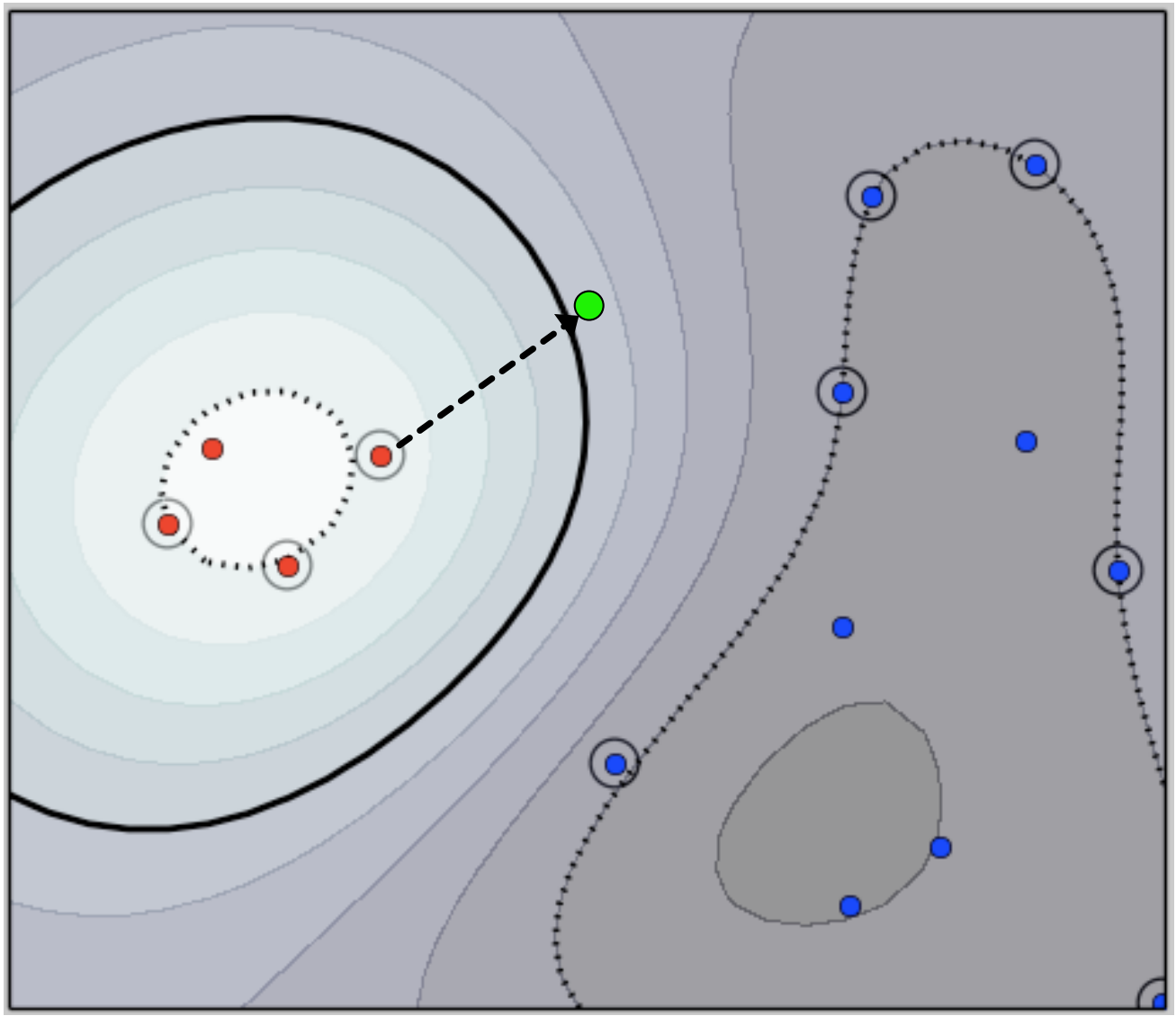
What is the rule? The rule is protect yourself at all times (from the movie "Million dollar baby", 2004)

Countering Adversarial Examples

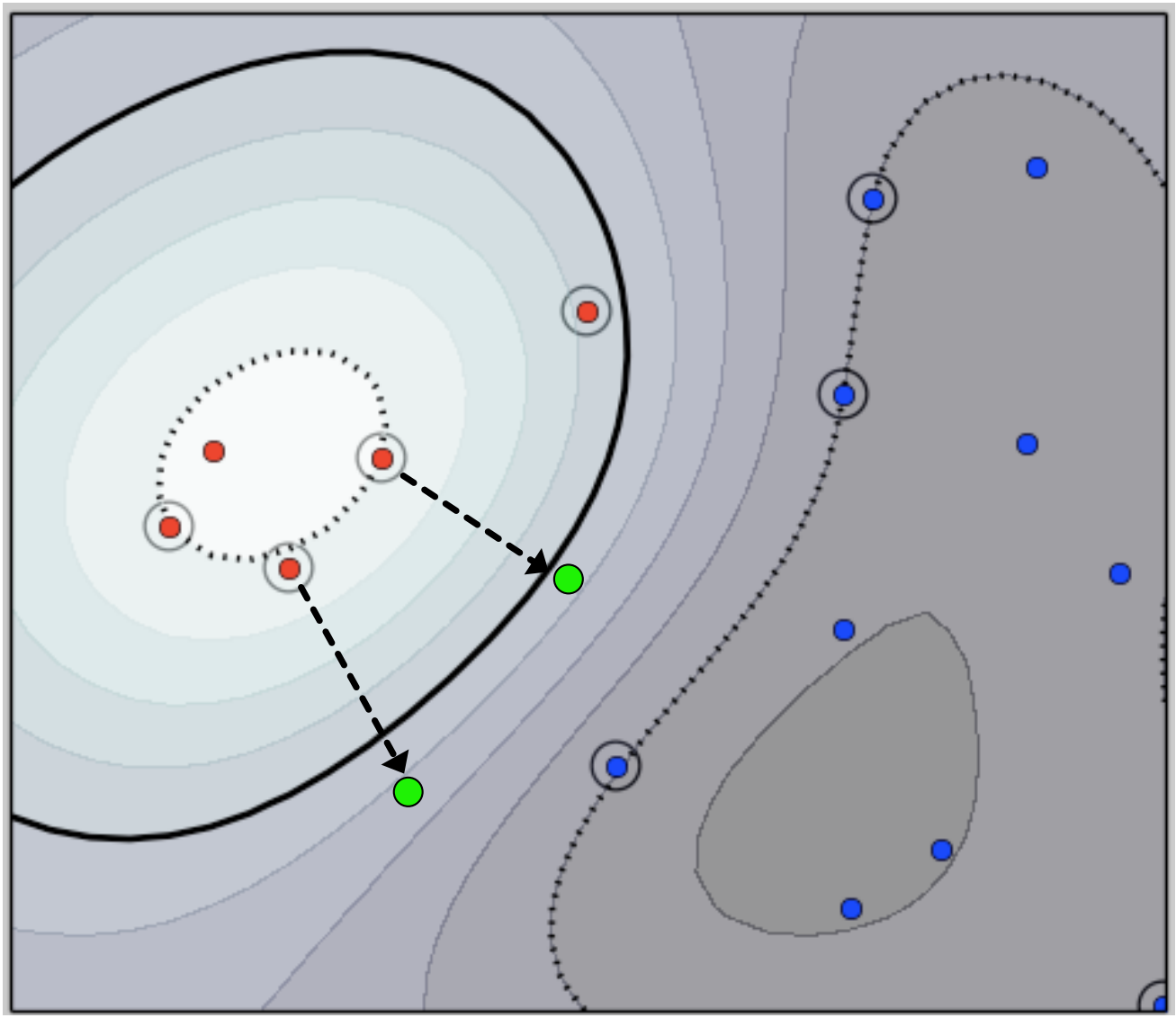
- **A common defense here is called 'adversarial training'**
  - Same idea of game-theoretical approaches
    - Attacker optimizes attack samples, classifier is retrained on them
  - This strategy is also used in deep nets
    - empirically successful
    - no uniqueness of equilibrium, convergence guarantees, etc.

- **What is the "magic" behind it?**
  - How does it (ideally) modify the decision surface?

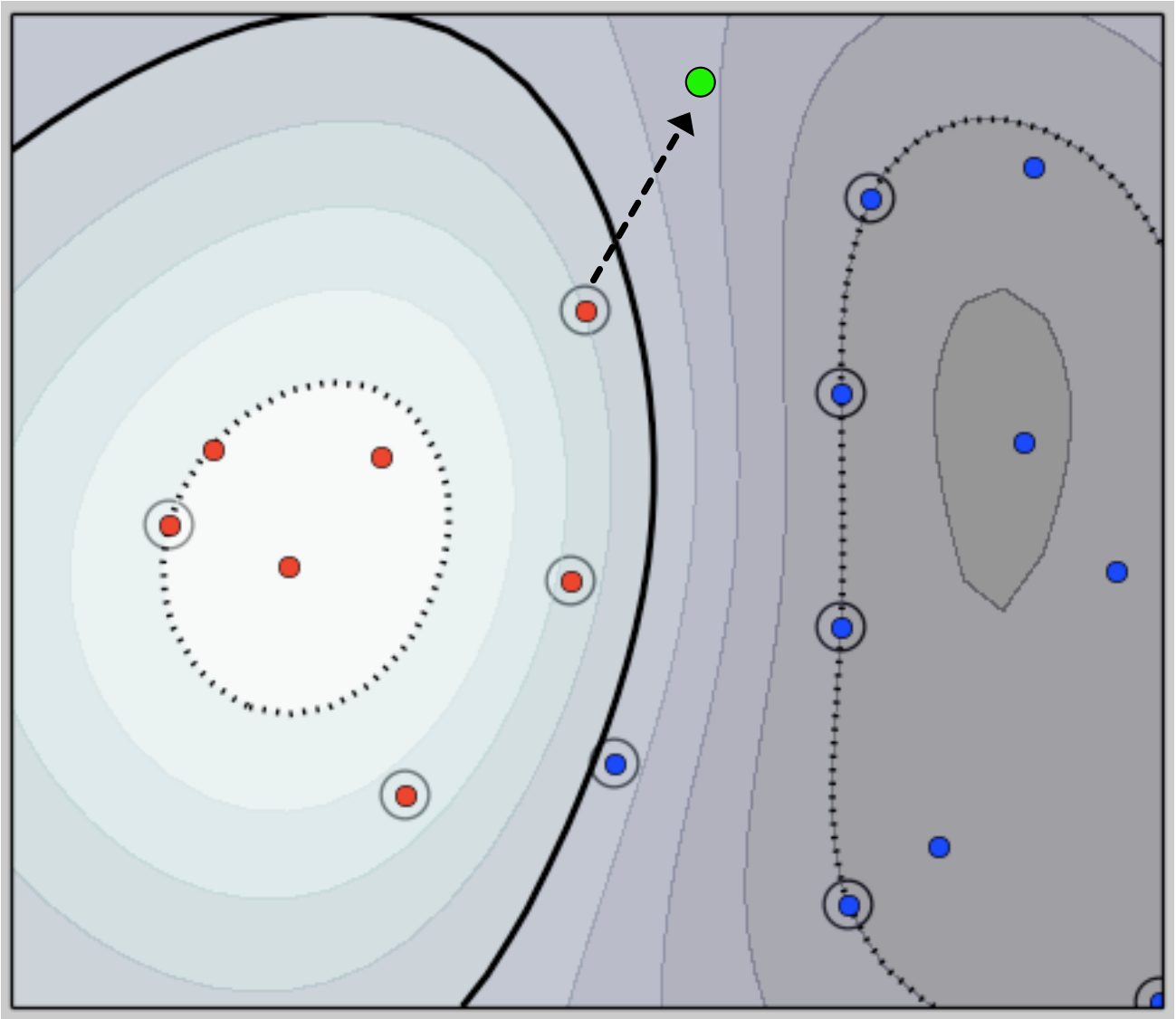
Secure Learning (Intuition)



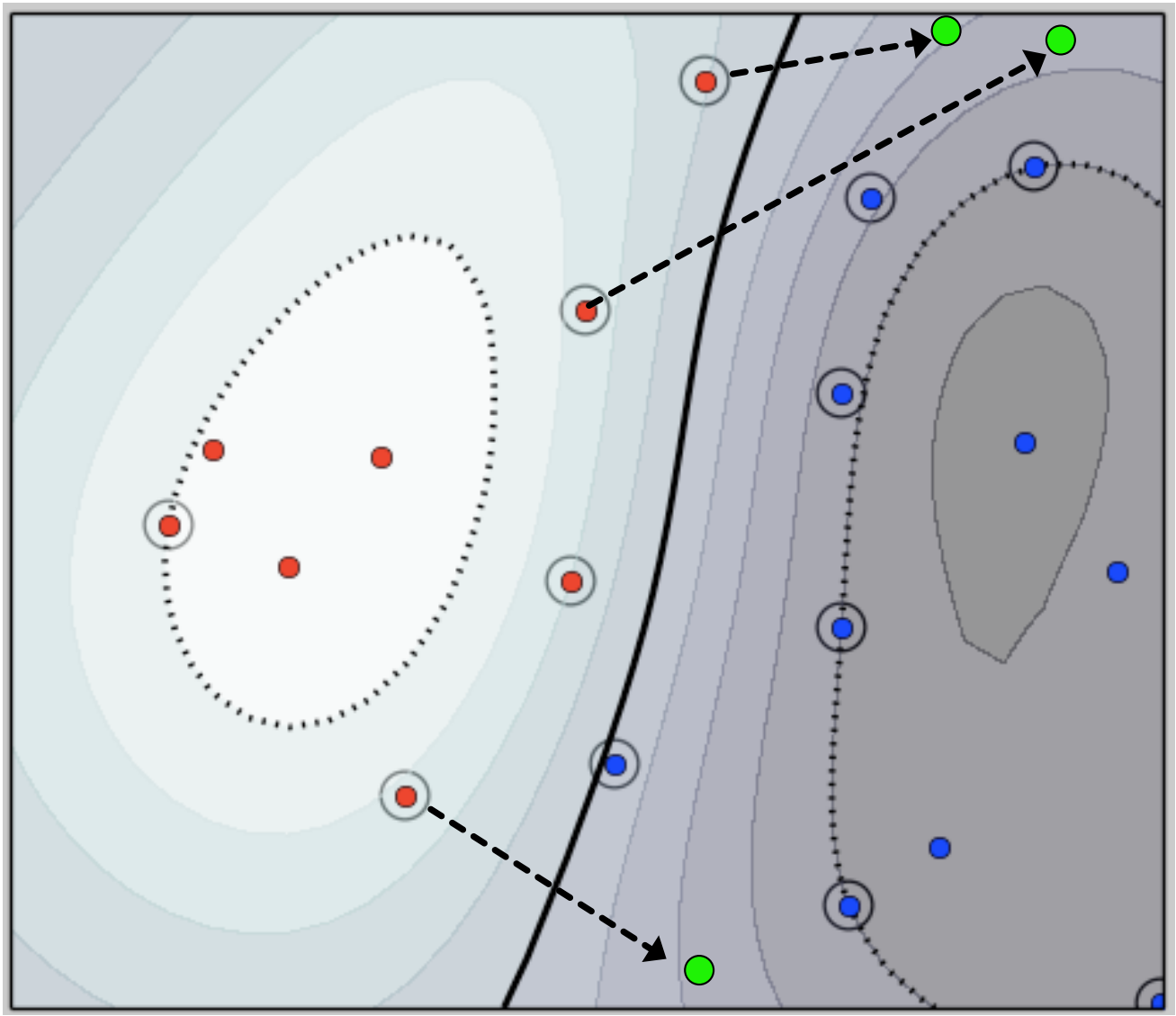
Secure Learning (Intuition)



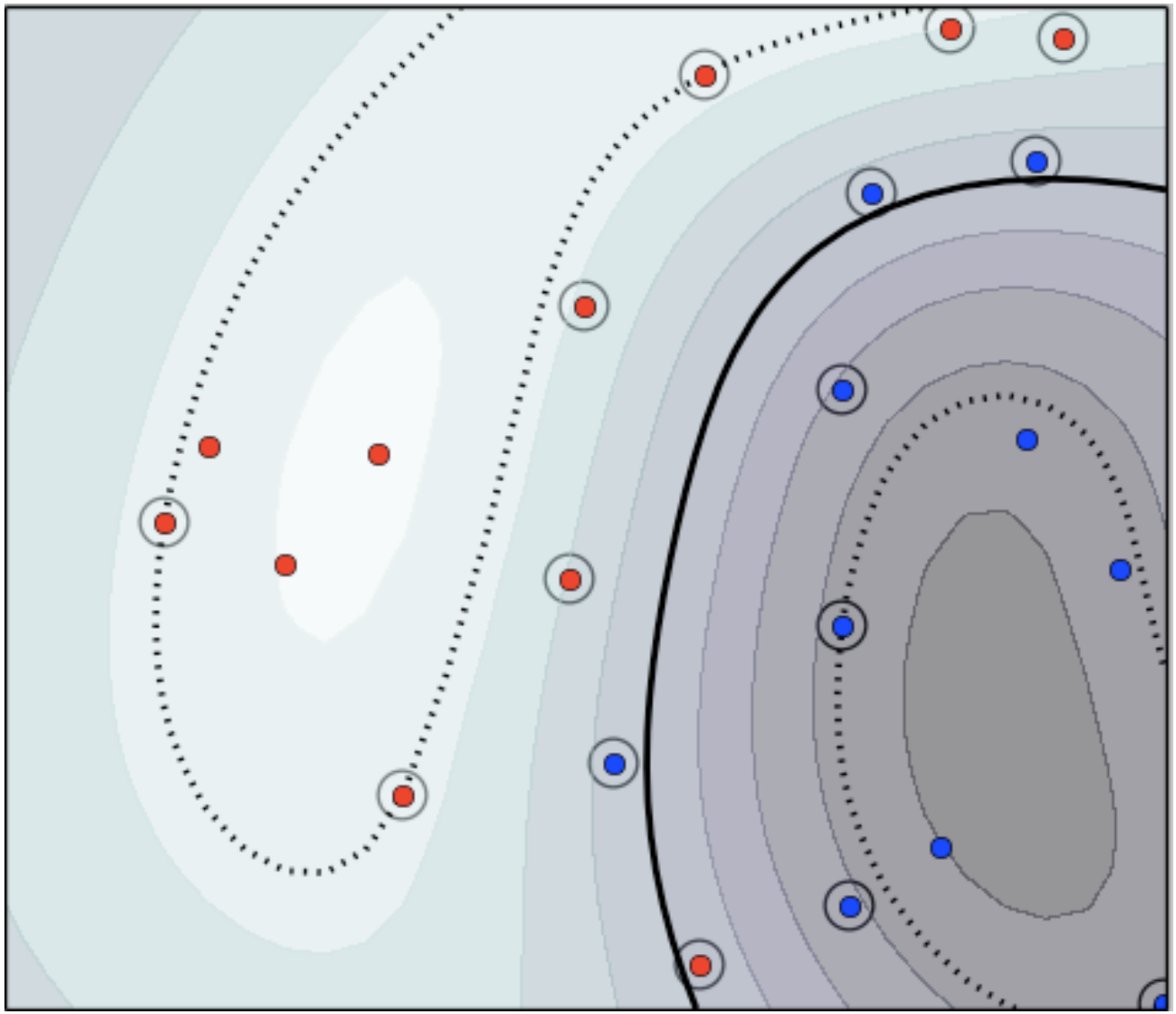
Secure Learning (Intuition)



Secure Learning (Intuition)

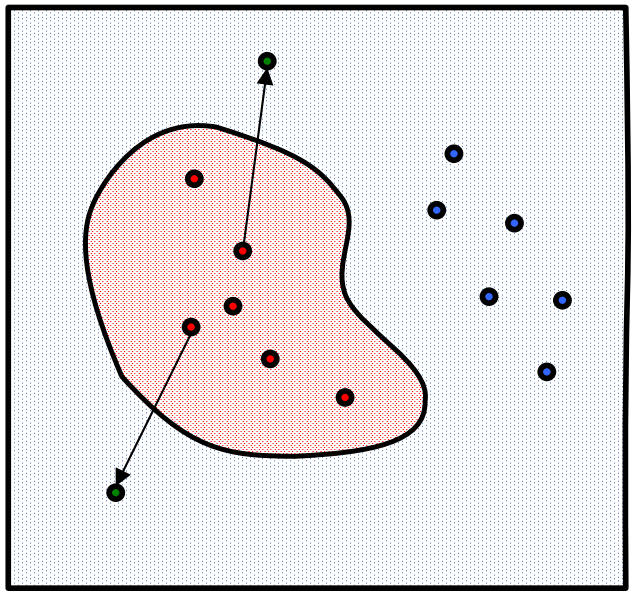


Secure Learning (Intuition)

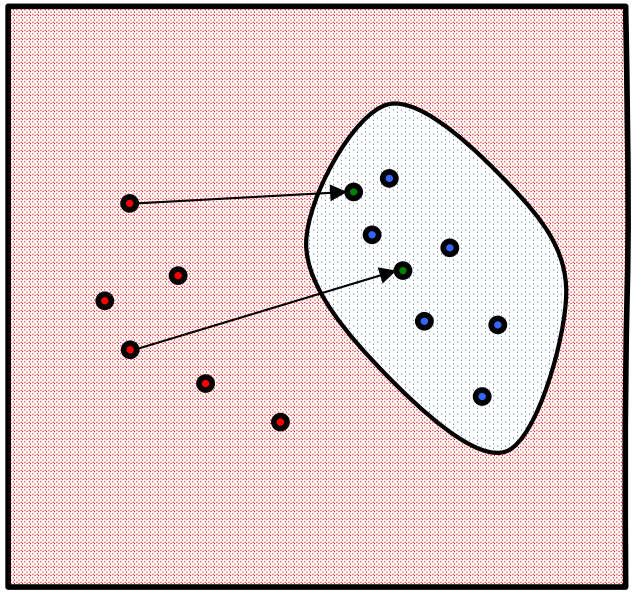


Blind-spot evasion

- **The vulnerability of learning algorithms is due to blind-spot regions**
  - Regions far from training data that are anyway assigned to 'legitimate' classes



blind-spot evasion

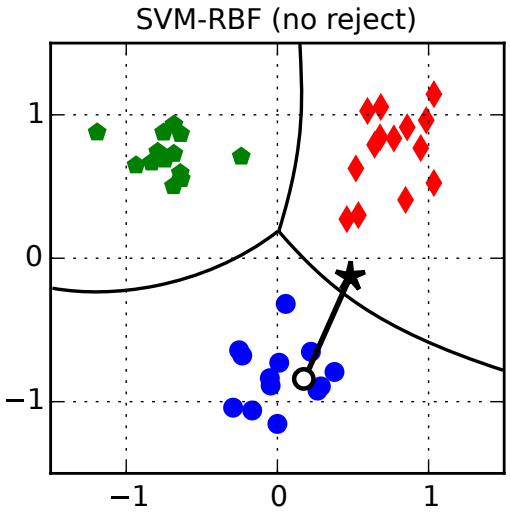


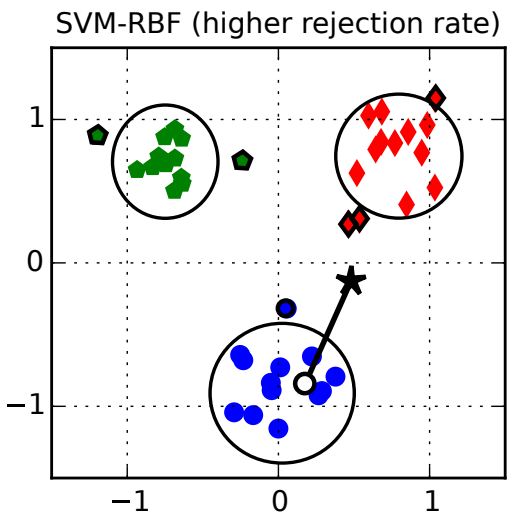
mimicry evasion

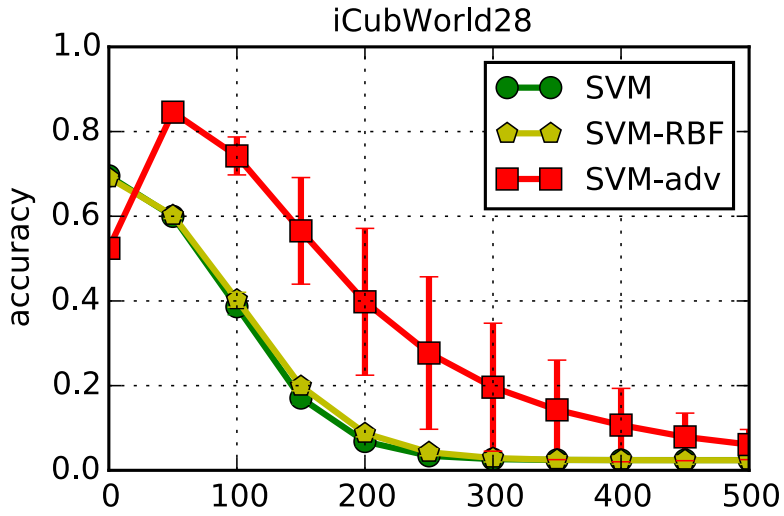
- **Enclosing legitimate data can be difficult with adversarial training**
  - amount of modifications, number of iterations... no formal guarantees...

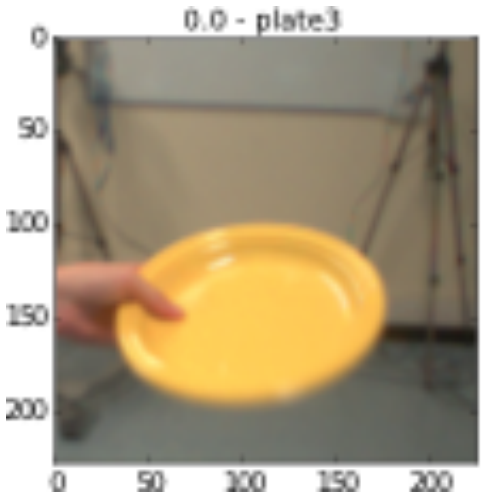
Detecting & Rejecting Adversarial Examples

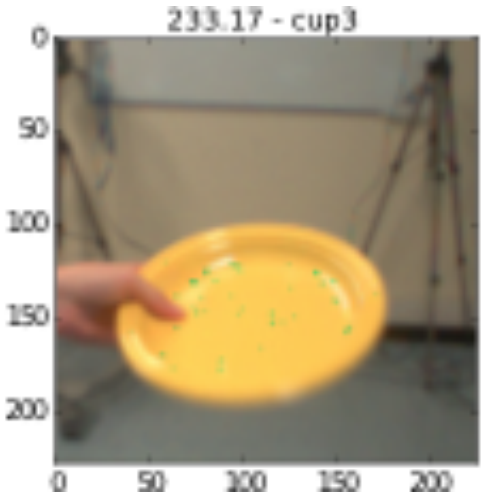
[M. Melis et al, Is deep learning safe for robot vision? ..., ICCV 2017 ViPAR Workshop]

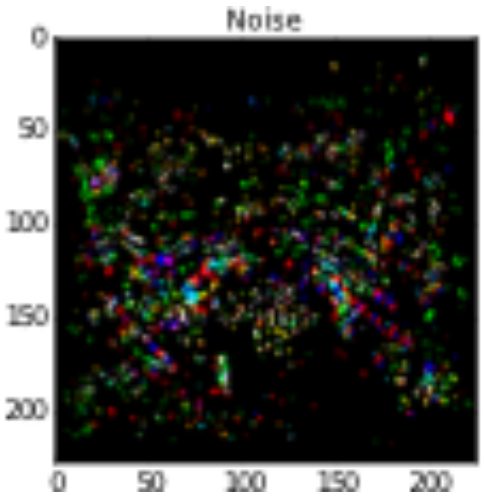












See also: Bendale et al., Towards Open-set deep networks, CVPR 2016

Adversarial Examples against Machine Learning

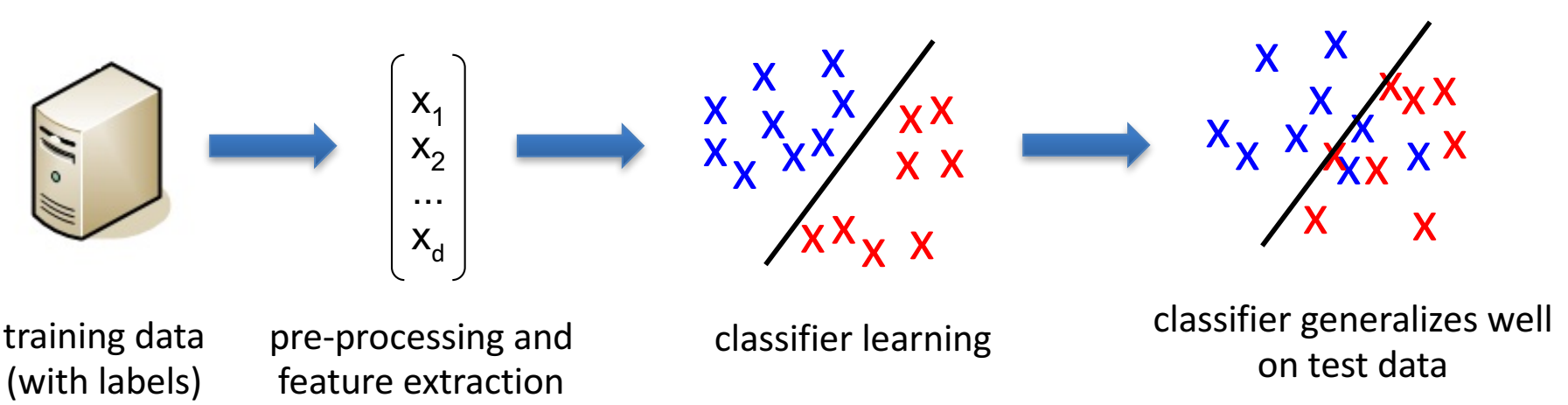
Web Demo

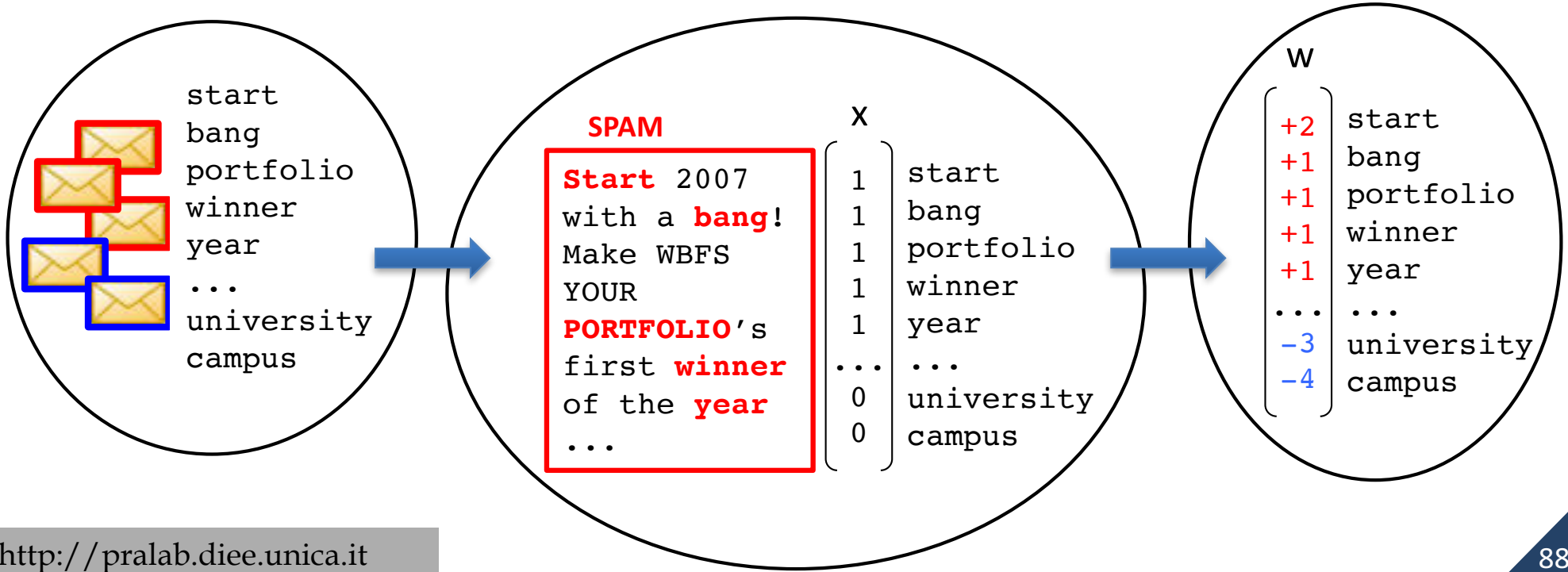
https://sec-ml.pluribus-one.it/

Poisoning Machine Learning

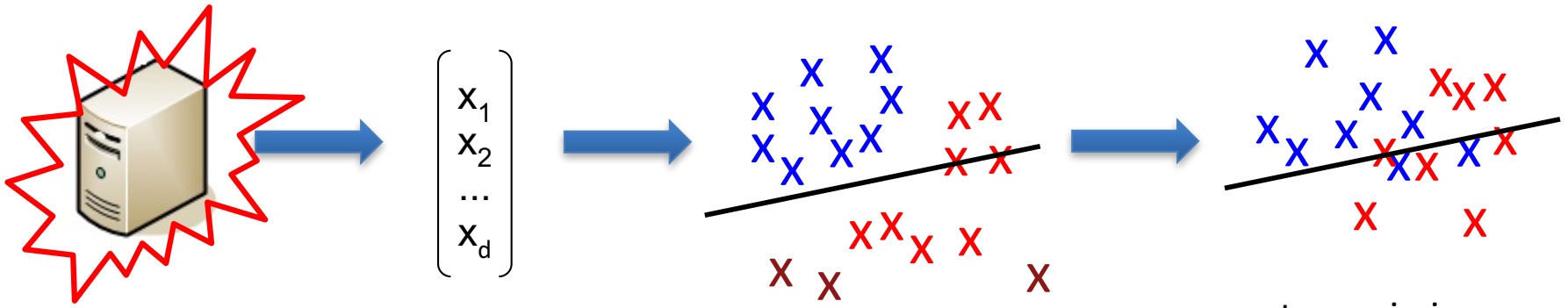
1. B. Biggio, B. Nelson, P. Laskov. Poisoning attacks against SVMs. ICML, 2012
2. B. Biggio et al., Security evaluation of SVMs. SVM applications. Springer, 2014
3. H. Xiao et al., Is feature selection secure against training data poisoning? ICML, 2015
4. L. Munoz-Gonzalez et al., Towards poisoning of deep learning algorithms, AISec 2017

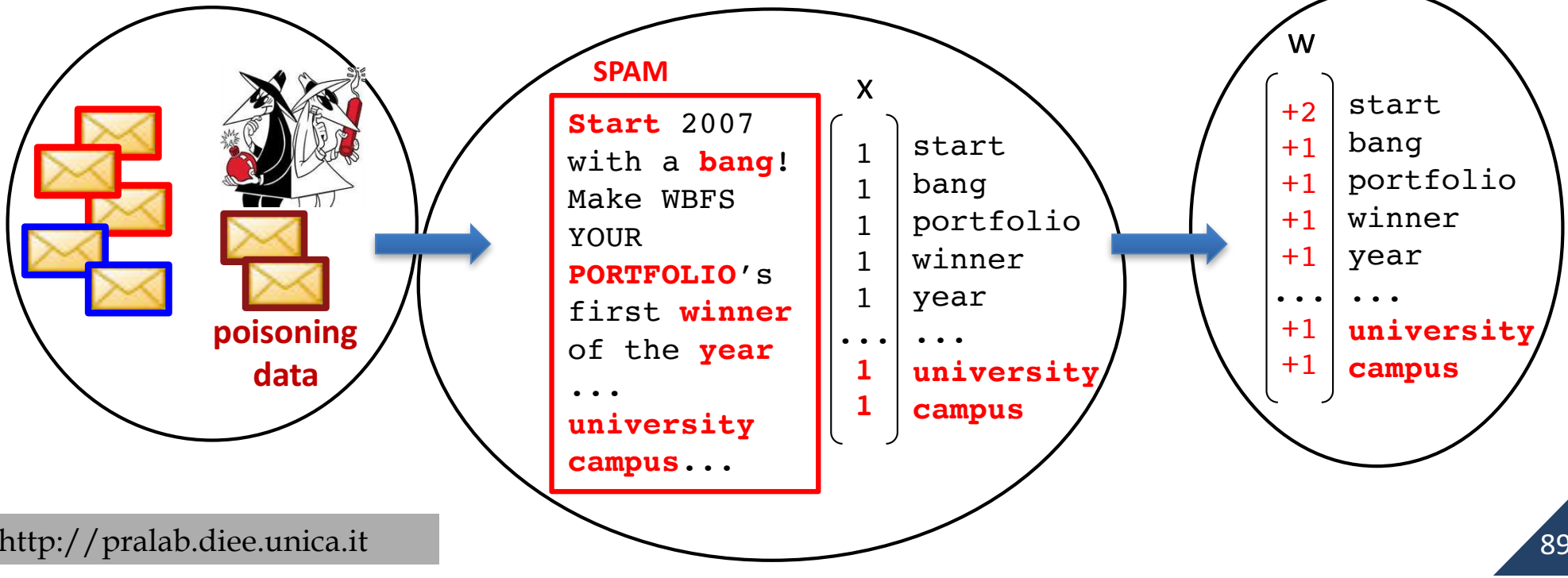
Poisoning Machine Learning





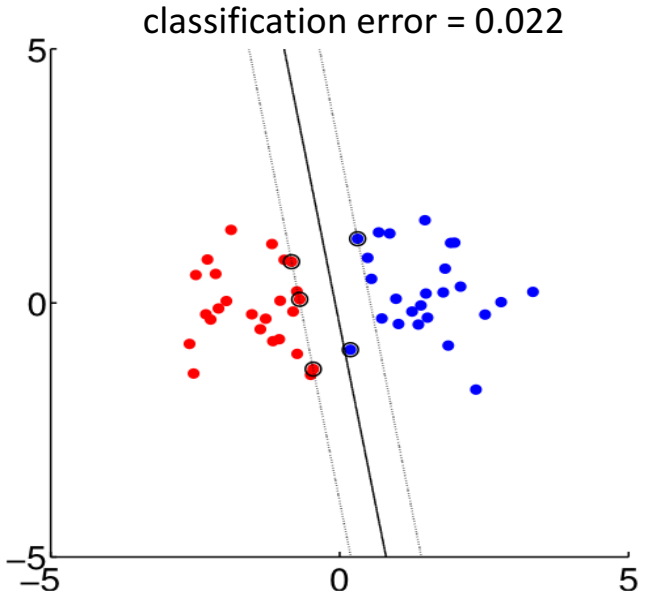
Poisoning Machine Learning

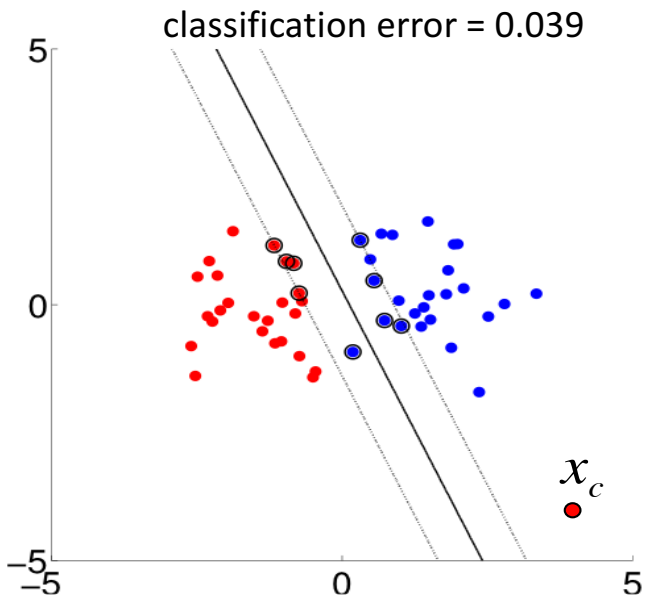




Adversary Model and Attack Strategy

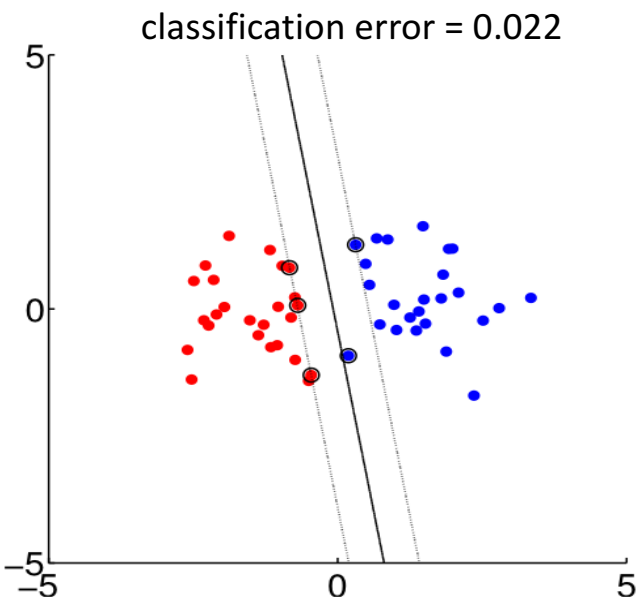
- **Adversary model**
  - **Goal**: to maximize classification error
  - **Knowledge**: perfect / limited
  - **Capability**: injecting samples into TR
- **Attack strategy**
  - optimal attack point x_c in TR that maximizes classification error

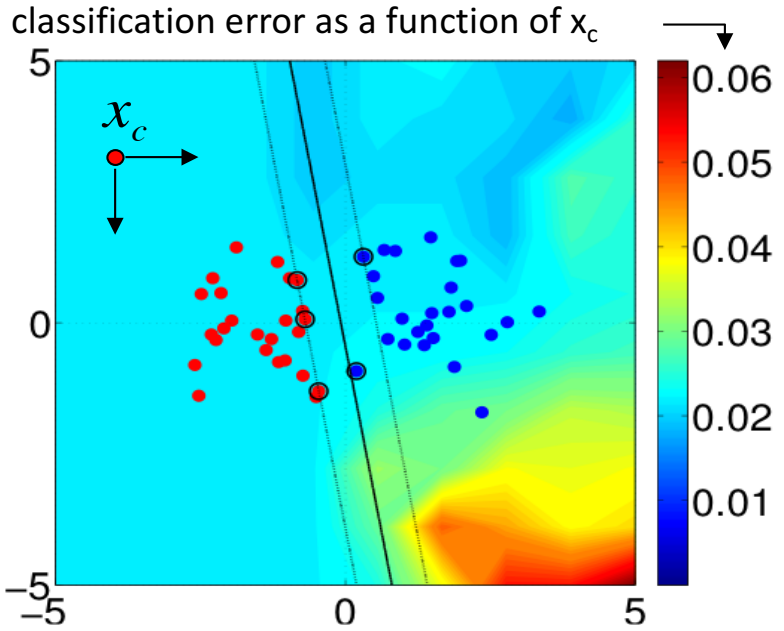




Adversary Model and Attack Strategy

- **Adversary model**
  - **Goal**: to maximize classification error
  - **Knowledge**: perfect / limited
  - **Capability**: injecting samples into TR
- **Attack strategy**
  - optimal attack point x_c in TR that maximizes classification error





Poisoning is a Bilevel Optimization

- **Attacker's objective**
  - to maximize generalization error on untainted data, w.r.t. poisoning point $\mathbf{x}_c$

$$\max_{x_c} L(D_{val}, f^*)$$

Loss estimated on validation data
(no attack points!)

$$\text{s.t. } f^* = \text{argmin}_f \mathcal{L}(D_{tr} \cup \{\boldsymbol{x}_c, y_c\}, f)$$

Algorithm is trained on surrogate data
(including the attack point)

- **Poisoning problem against (linear) SVMs:**

$$\max_{x_c} \sum_{k=1}^{m} \max(0, 1 - y_k f^*(\boldsymbol{x}_k))$$

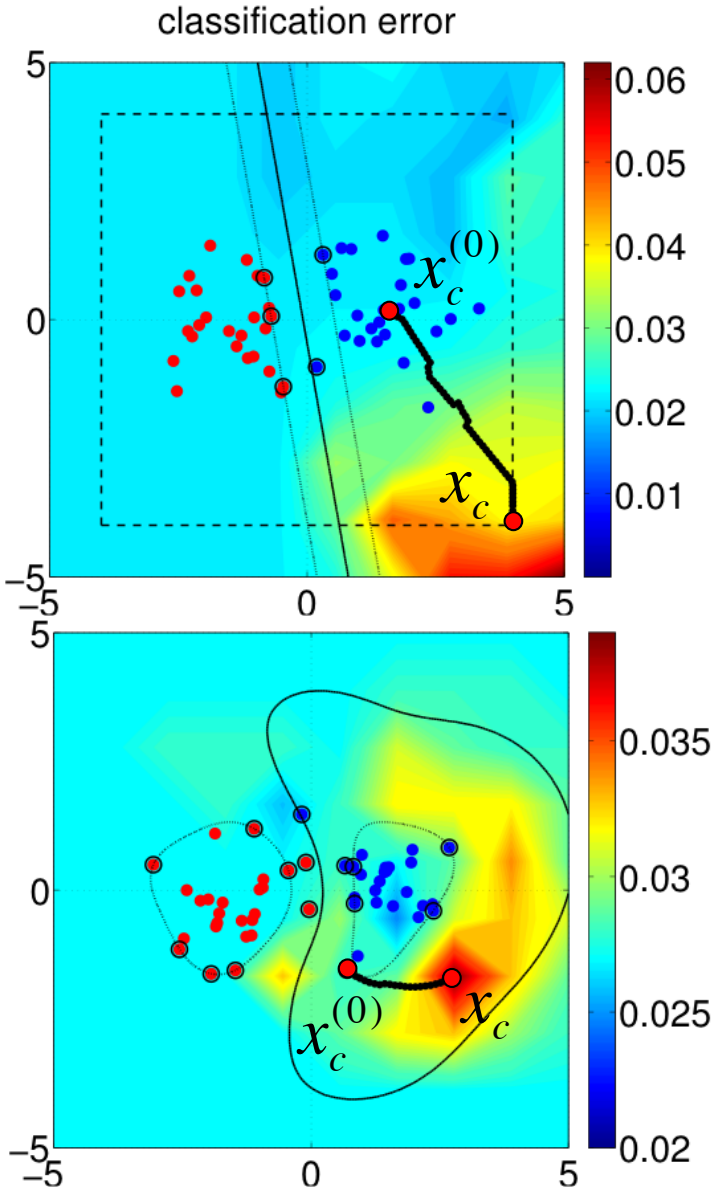
$$\text{s.t. } f^* = \text{argmin}_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + \mathrm{C}\sum_{i=1}^{n} \max(0, 1 - y_i f(\boldsymbol{x}_i)) + \mathrm{C}\max(0, 1 - y_c f(\boldsymbol{x}_c))$$

1. B. Biggio, B. Nelson, P. Laskov. Poisoning attacks against SVMs. ICML, 2012
2. H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli. Is feature selection secure against training data poisoning? ICML, 2015.

Gradient-based Poisoning Attacks

- **Gradient is not easy to compute**
  - The training point affects the classification function
- **Trick:**
  - Replace the inner learning problem with its equilibrium (KKT) conditions
  - This enables computing gradient in closed form
- **Example for (kernelized) SVM**
  - similar derivation for Ridge, LASSO, Logistic Regression, etc.

$$\nabla_{\boldsymbol{x}_c}\mathcal{A} = -\boldsymbol{y}_k^\top \frac{\partial \boldsymbol{k}_{kc}}{\partial \boldsymbol{x}_c}\alpha_c + \boldsymbol{y}_k^\top \underbrace{[\mathbf{K}_{ks} \quad \mathbf{1}]}_{k\times s+1} \underbrace{\begin{bmatrix} \boldsymbol{K}_{ss} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \boldsymbol{k}_{sc}}{\partial \boldsymbol{x}_c} \\ 0 \end{bmatrix} \alpha_c}_{(s+1)\times d}$$

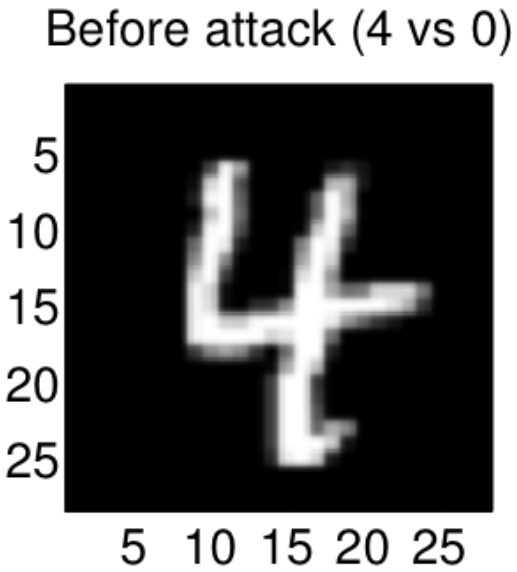


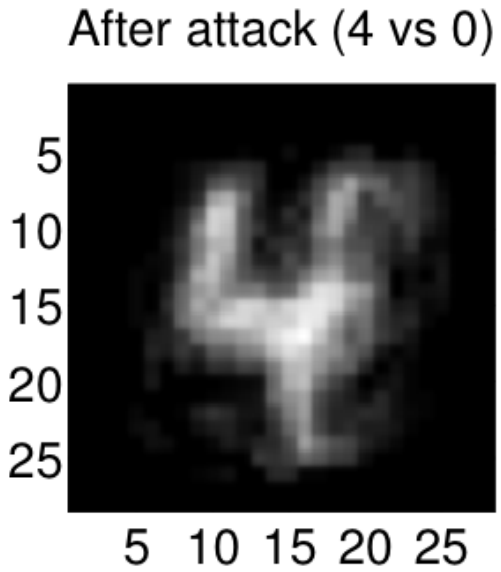
1. B. Biggio, B. Nelson, P. Laskov. Poisoning attacks against SVMs. ICML, 2012

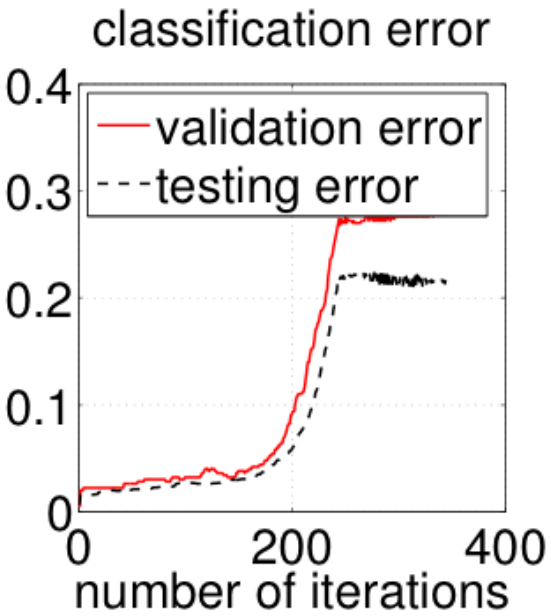
Experiments on MNIST digits

Single-point attack

- **Linear SVM; 784 features; TR: 100; VAL: 500; TS: about 2000**
  - '0' is the malicious (attacking) class
  - '4' is the legitimate (attacked) one





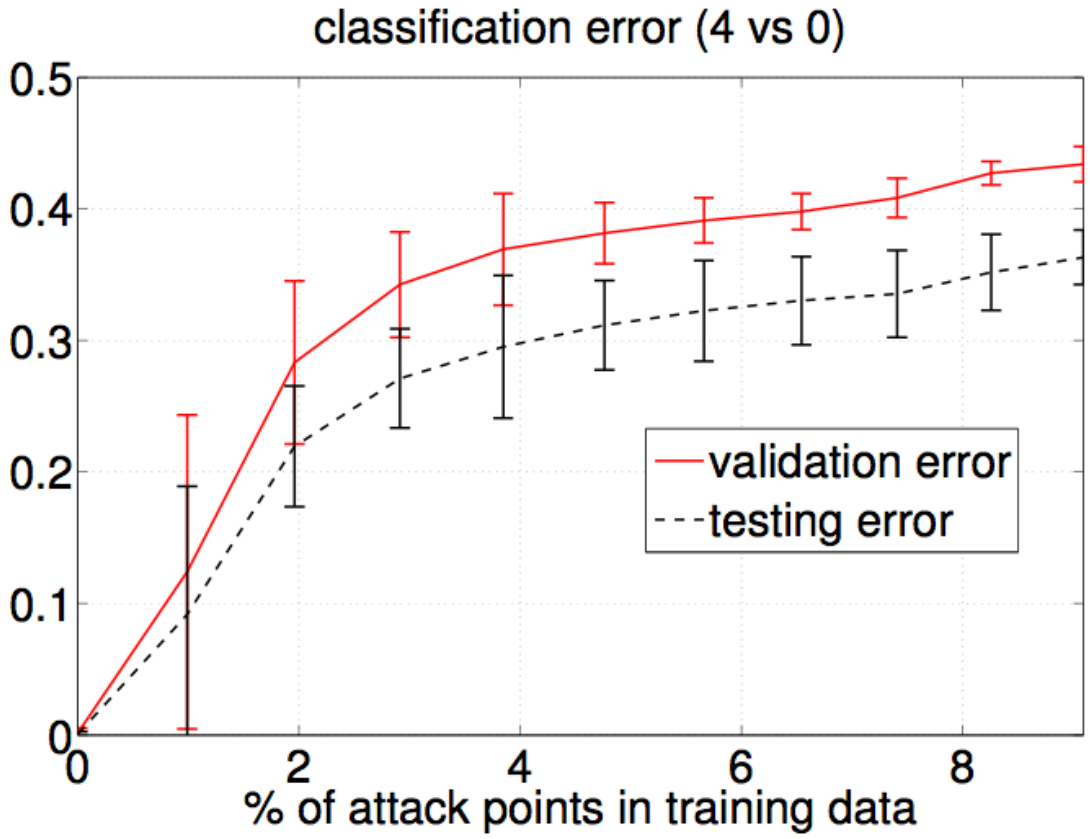


$x_c^{(0)}$ $\longrightarrow$ x_c

Experiments on MNIST digits

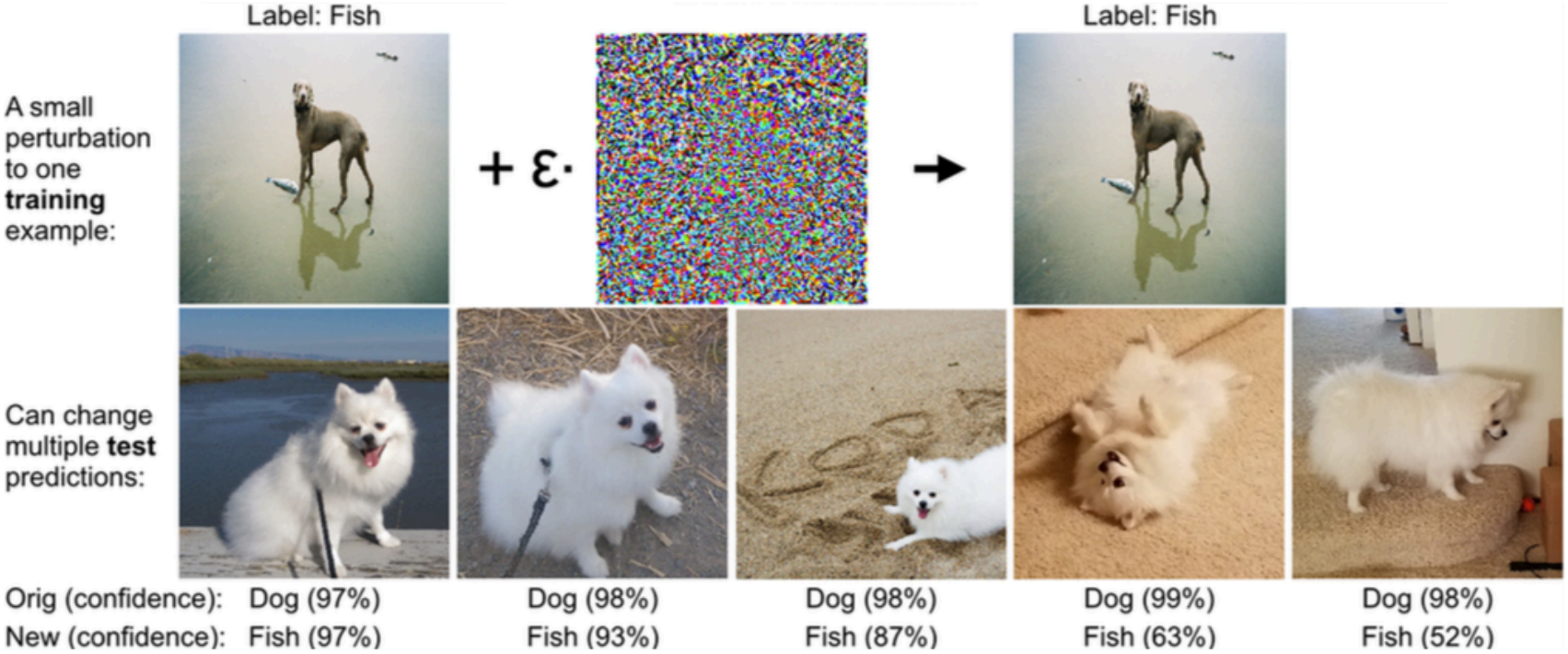
Multiple-point attack

- **Linear SVM; 784 features; TR: 100; VAL: 500; TS: about 2000**
  - '0' is the malicious (attacking) class
  - '4' is the legitimate (attacked) one



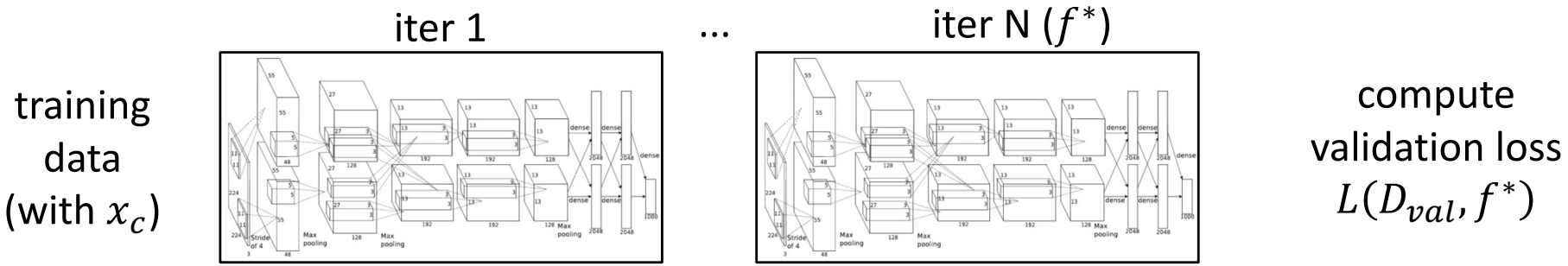
How about Poisoning Deep Nets?

- **ICML 2017 Best Paper by Koh et al., "Understanding black-box predictions via Influence Functions" has derived adversarial *training* examples against a DNN**
  - they have been constructed attacking only the last layer (KKT-based attack against logistic regression) and assuming the rest of the network to be "frozen"



Towards Poisoning Deep Neural Networks

- **End-to-end poisoning of DNN? (without KKT conditions)**
  - Automatic differentiation to compute $\nabla_{x_c} L(D_{val}, f^*)$
  - It requires unfolding the whole gradient-based learning procedure...
- **Problem: huge computational graph (not feasible for DNN)**
  - Replicate the DNN graph for each training iteration...



- **Solution: back-gradient (reverse the learning algorithm at each step to avoid storing the full computational graph)**
  - See the paper for details: *L. Muñoz-González et al.*, AISec 2017 https://arxiv.org/abs/1708.08689

Towards Poisoning Deep Neural Networks

- **Adversarial Training Examples against a DNN**
- **Preliminary results show that DNNs are more robust to poisoning**
  - perhaps thanks to their higher *capacity*?

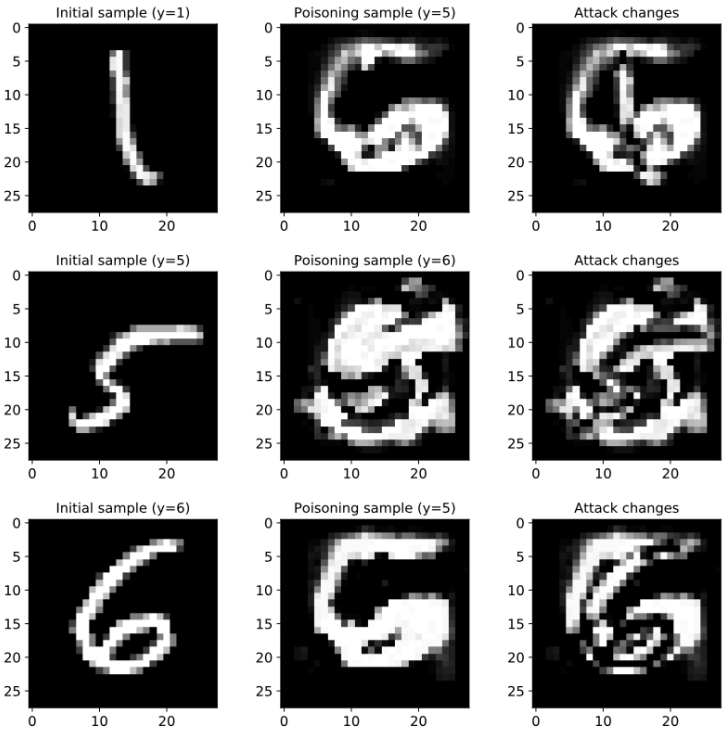


Figure 6: Poisoning samples targeting the LR.

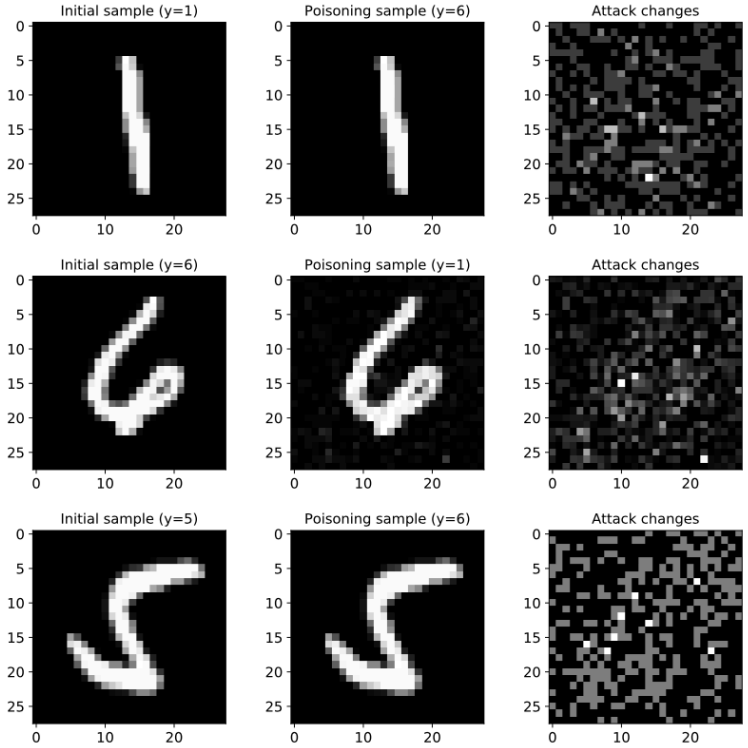
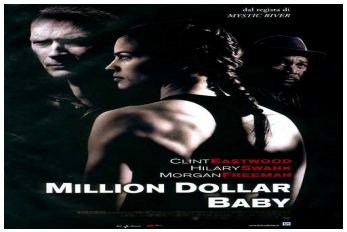


Figure 5: Poisoning samples targeting the CNN.

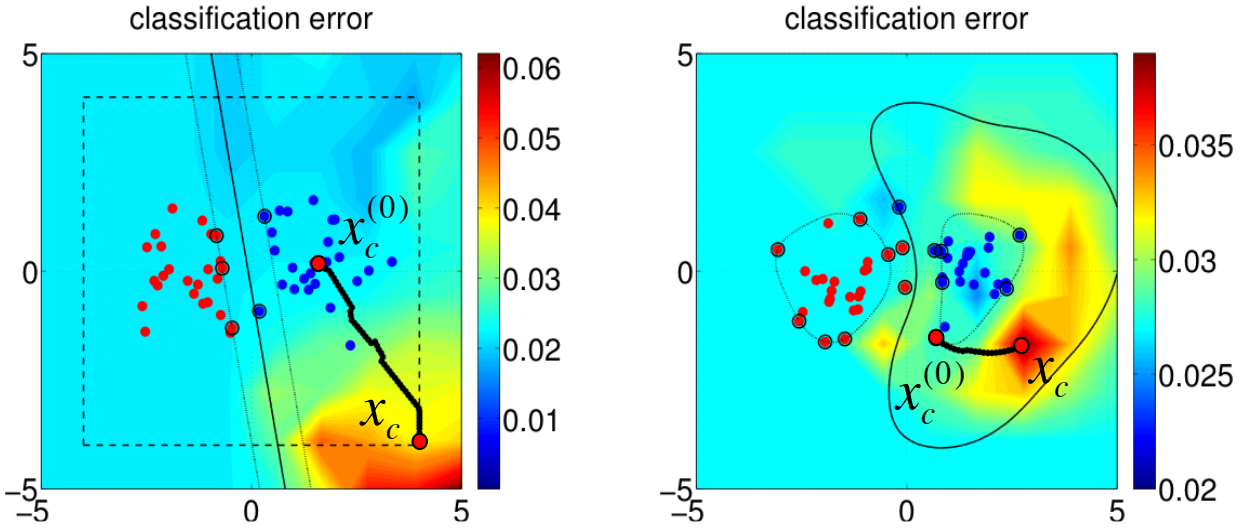
Countering Poisoning Attacks



What is the rule? The rule is protect yourself at all times (from the movie "Million dollar baby", 2004)

Security Measures against Poisoning

- **Rationale: poisoning injects outlying training samples**

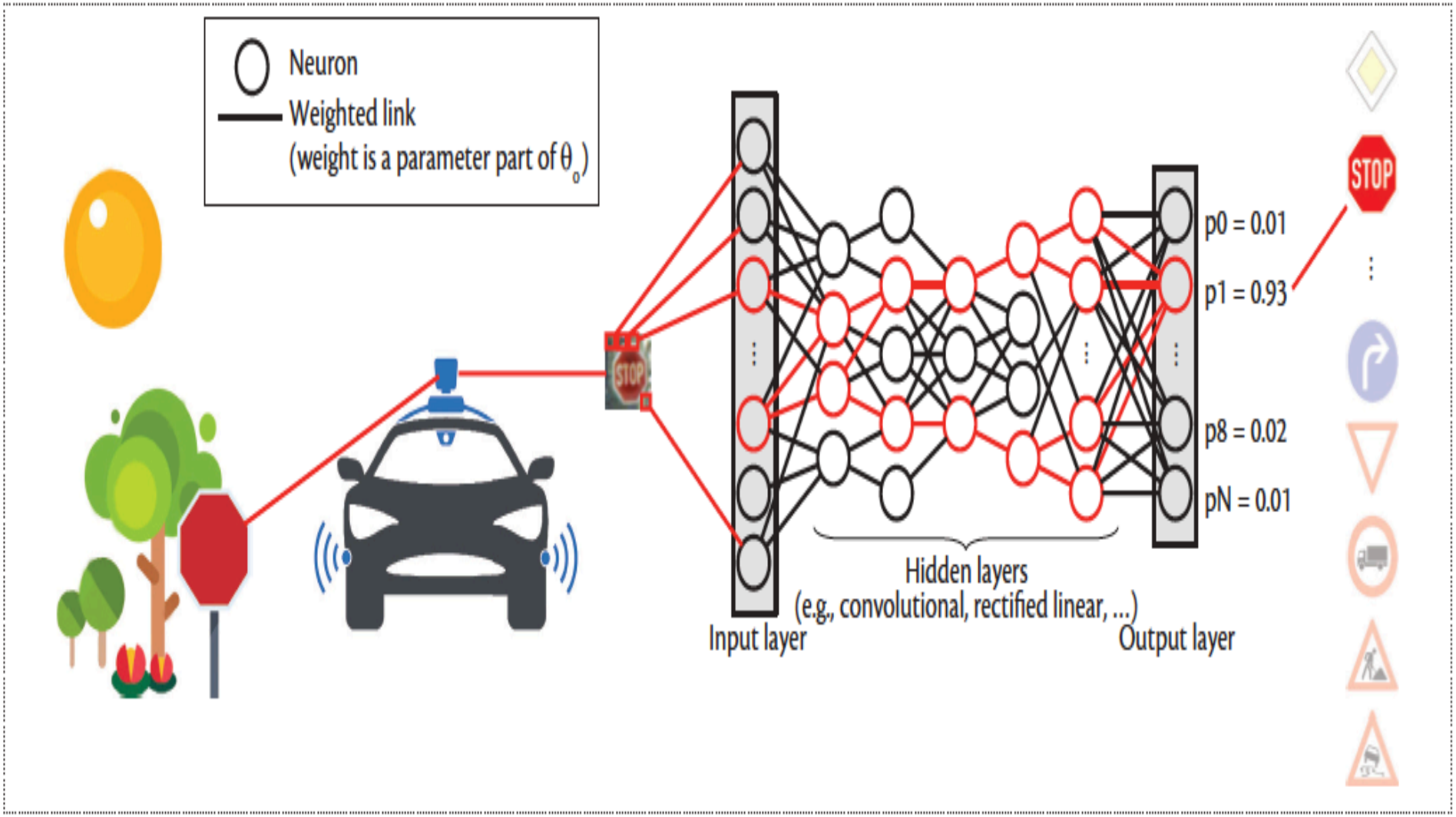


- **Two main strategies for countering this threat**
  1. **Data sanitization:** *remove* poisoning samples from training data
     - Bagging for fighting poisoning attacks
     - Reject-On-Negative-Impact (RONI) defense
  2. **Robust Learning:** learning algorithms that are natively robust to poisoning

What about adversarial examples in the physical world ?

Adversarial images in mission-critical apps…

[Patrick McDaniel et al., IEEE Security & Privacy, 2016]



Adversarial noise in mission-critical app…

[Patrick McDaniel et al., IEEE Security & Privacy, 2016]

To humans, adversarial images are indistinguishable from original images.
Left an ordinary image of a stop sign
Right an image manipulated with adversarial noise and classified as a yield sign

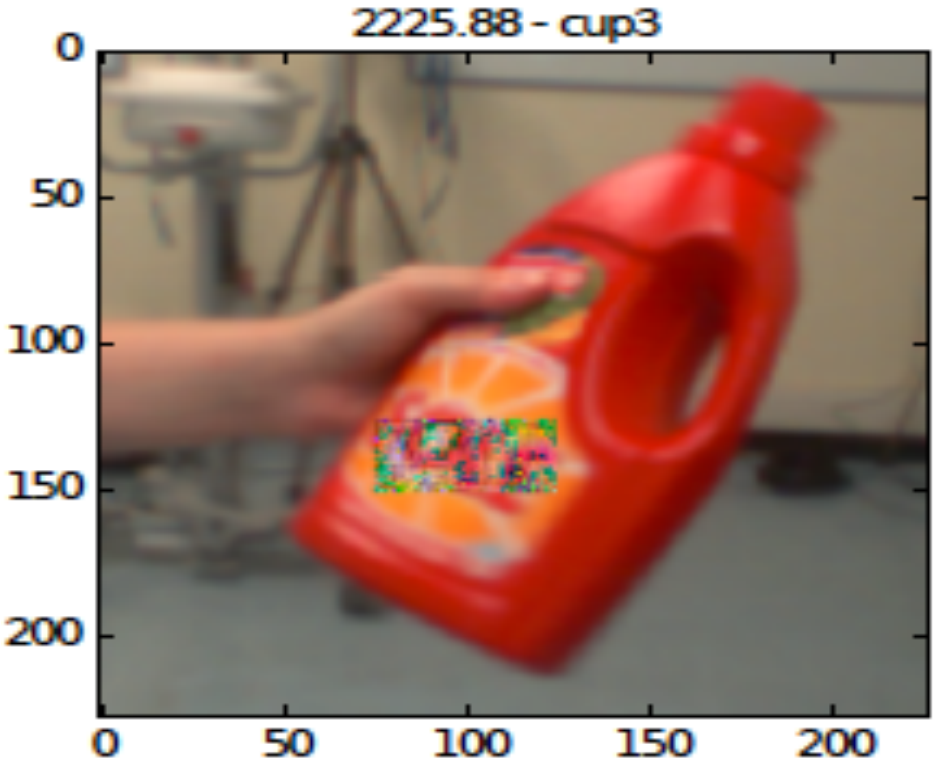
World is not digital…

- *….Previous cases of adversarial examples have common characteristic: the **adversary** is able to precisely **control** the **digital representation** of the input to the machine learning tools…..*

[M. Sharif et al., ACM CCS 2016]

World is not digital…

What about the **physical realizability** of adversarial examples ?



We should fabricate the "sticker", paste it on real plastic bottles and the then try to fool the iCub robot…

World is not digital…

What about the **inconspicuousness** of adversarial examples ?

It's not necessary that the manipulation of the physical object is "invisible", the important thing is that the attack that is taking place is not noticed…

Adversarial images in the physical world

[Alexey Kurakin et al., ICLR 2017]

Adversarial images fool deep networks even when they operate in the physical world, for example, images are taken from a cell-phone camera ?

- ✓ Alexey Kurakin et al. (2016, 2017) explored the possibility of creating adversarial images for machine learning systems which operate in the physical world. They used images taken from a cell-phone camera as an input to an Inception v3 image classification neural network.

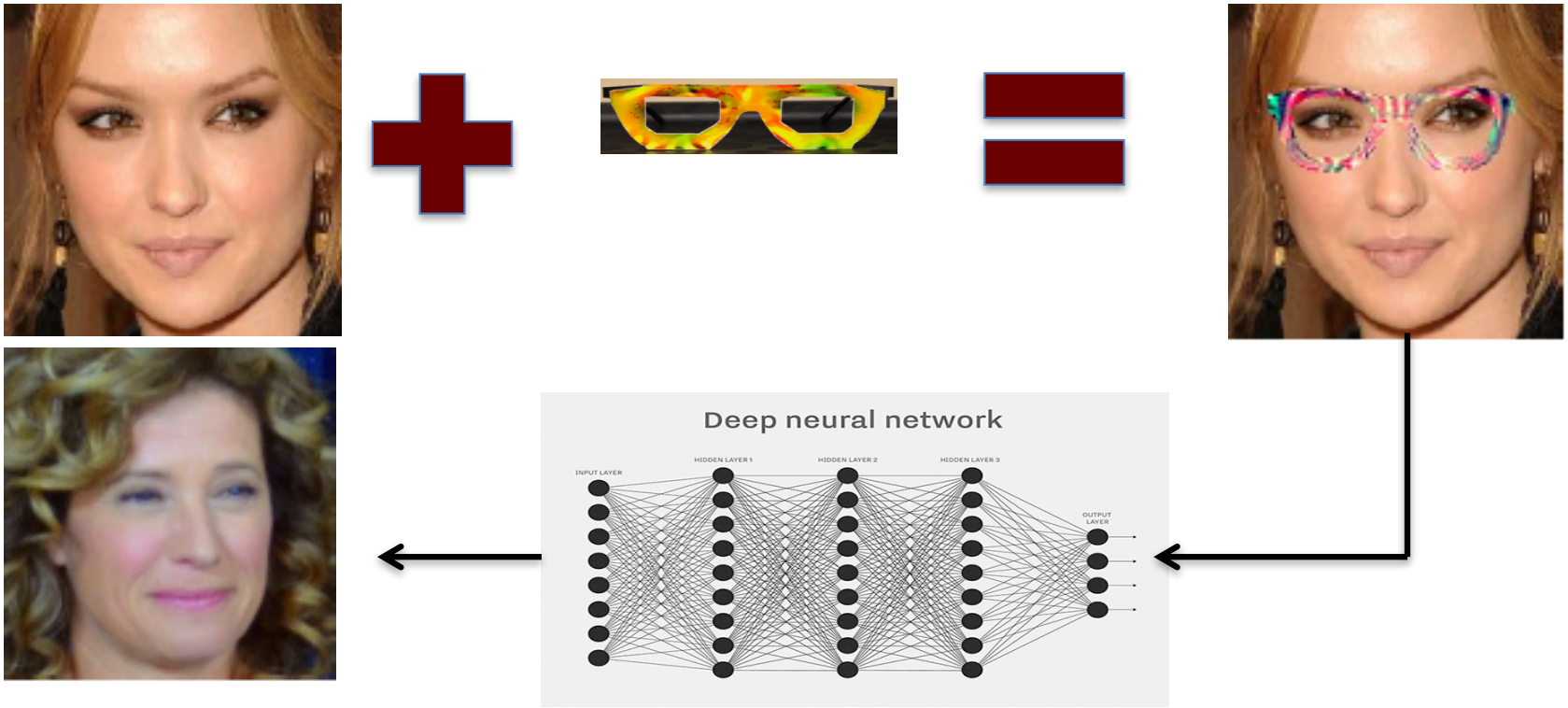
- ✓ They showed that in such a set-up, a significant fraction of adversarial images crafted using the original network are misclassified even when fed to the classifier through the camera.

Adversarial faces

[M. Sharif et al., ACM CCS 2016]

$$\underset{r}{\operatorname{argmin}} \left(\left(\sum_{x \in X} softmaxloss(x + r, c_t) \right) + \kappa_1 \cdot TV(r) + \kappa_2 \cdot NPS(r) \right)$$

The adversarial perturbation is applied only to the eyeglasses image region



Should we be worried ?

No, we shouldn't…

[arXiv:1707.03501; CVPR 2017]

NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles

Jiajun Lu*, Hussein Sibai*, Evan Fabry, David Forsyth
University of Illinois at Urbana Champaign
{jlu23, sibai2, efabry2, daf}@illinois.edu

In this paper, we show experiments that suggest that a trained neural network classifies most of the pictures taken from different distances and angles of a perturbed image correctly. We believe this is because the adversarial property of the perturbation is **sensitive to the scale** at which the perturbed picture is viewed, so (for example) **an autonomous car** will **misclassify** a **stop** sign **only** from a **small range of distances**.

Yes, we should…

Robust Physical-World Attacks on Machine Learning Models

Visit https://iotsecurity.eecs.umich.edu/#roadsigns for an FAQ

Ivan Evtimov[1], Kevin Eykholt[2], Earlence Fernandes[1], Tadayoshi Kohno[1],
Bo Li[4], Atul Prakash[2], Amir Rahmati[3], and Dawn Song*[4]

[1]University of Washington
[2]University of Michigan Ann Arbor
[3]Stony Brook University
[4]University of California, Berkeley

Yes, we should…

Synthesizing Robust Adversarial Examples

Anish Athalye
OpenAI, MIT

Ilya Sutskever
OpenAI

[https://blog.openai.com/robust-adversarial-inputs/]

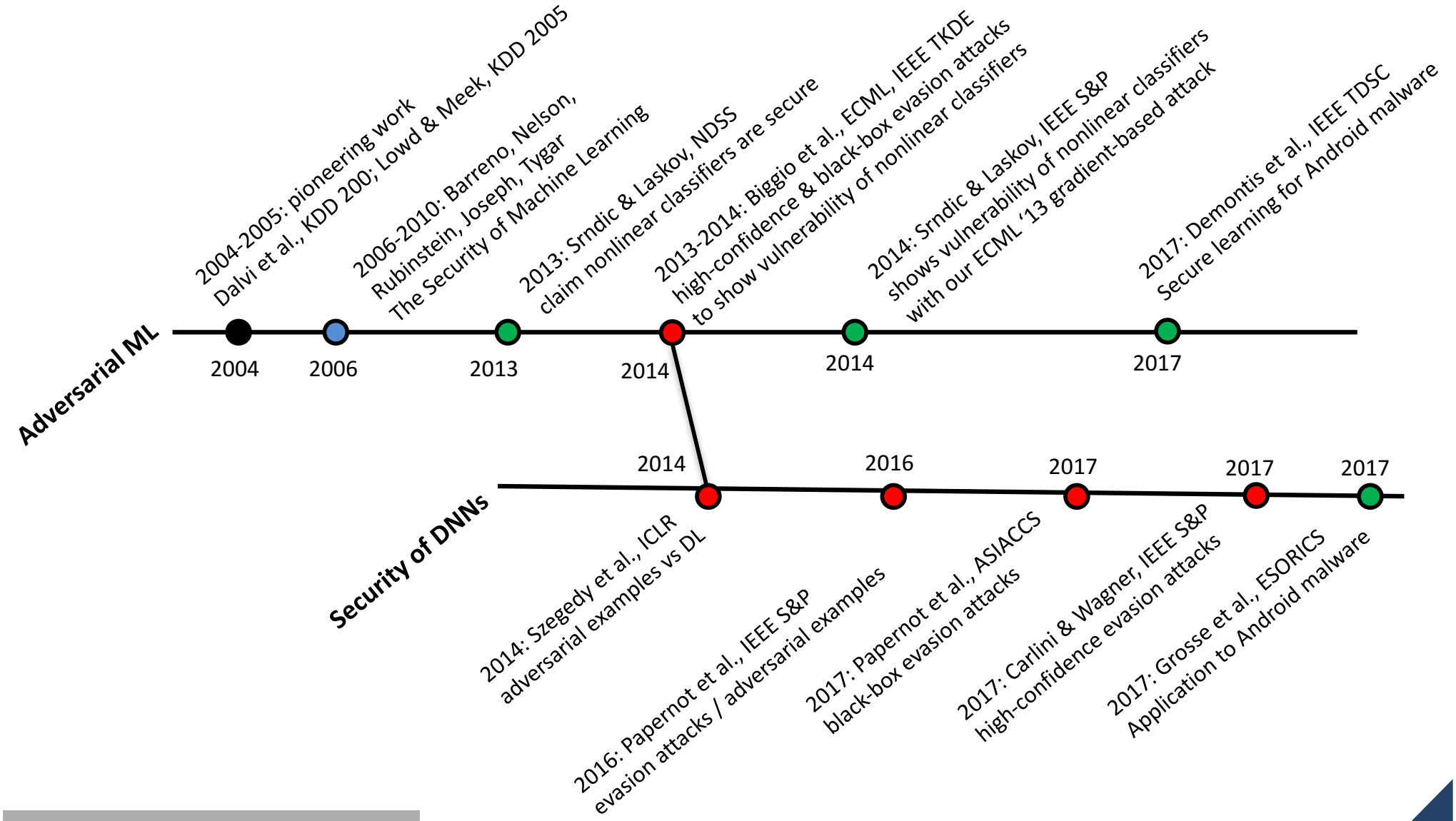
To conclude…

This is a recent research field…

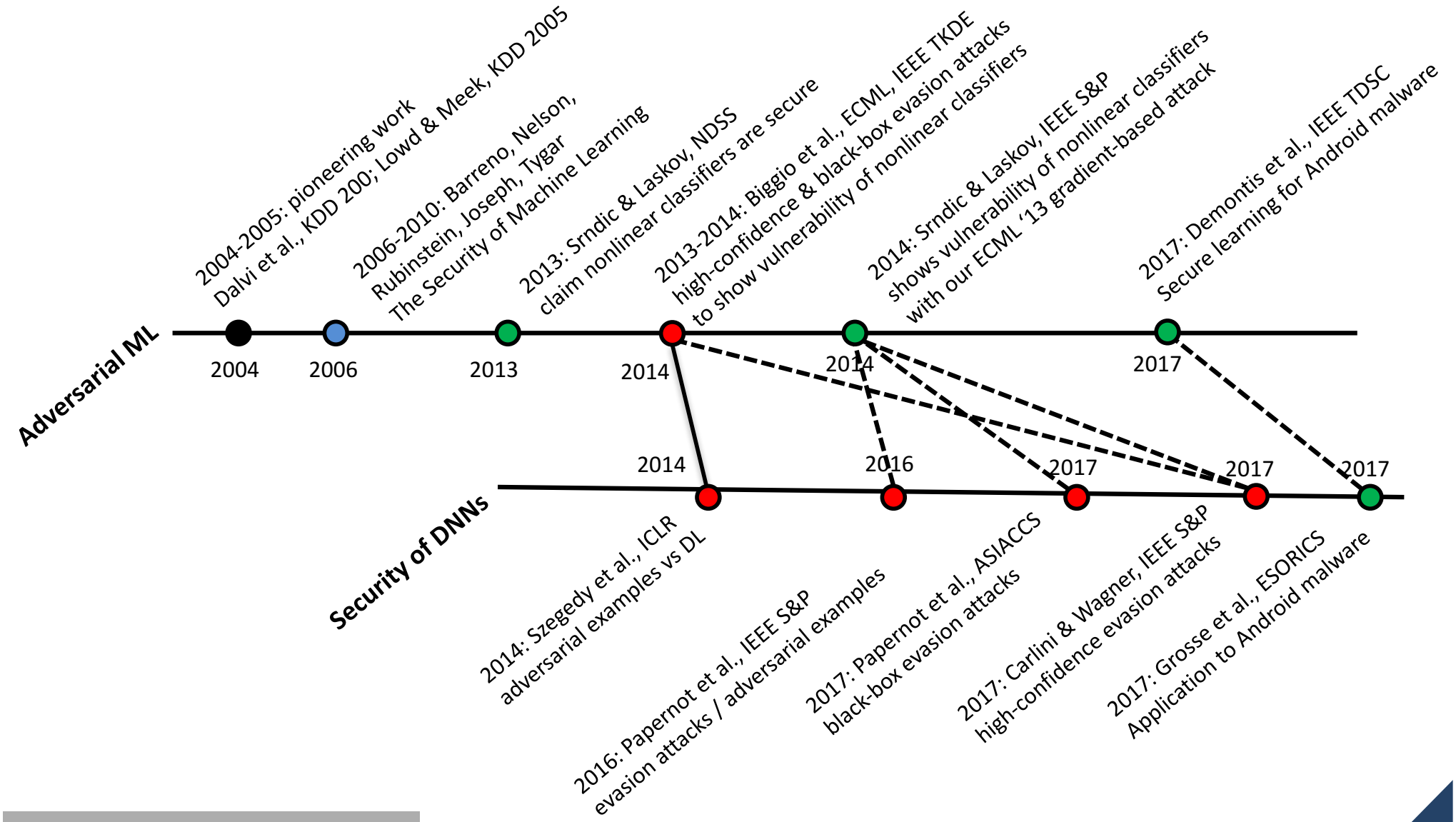
Dagstuhl Perspectives Workshop on
"Machine Learning in Computer Security"
Schloss Dagstuhl, Germany, Sept. 9th-14th, 2012

SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik

Timeline of Learning Security



Timeline of Learning Security



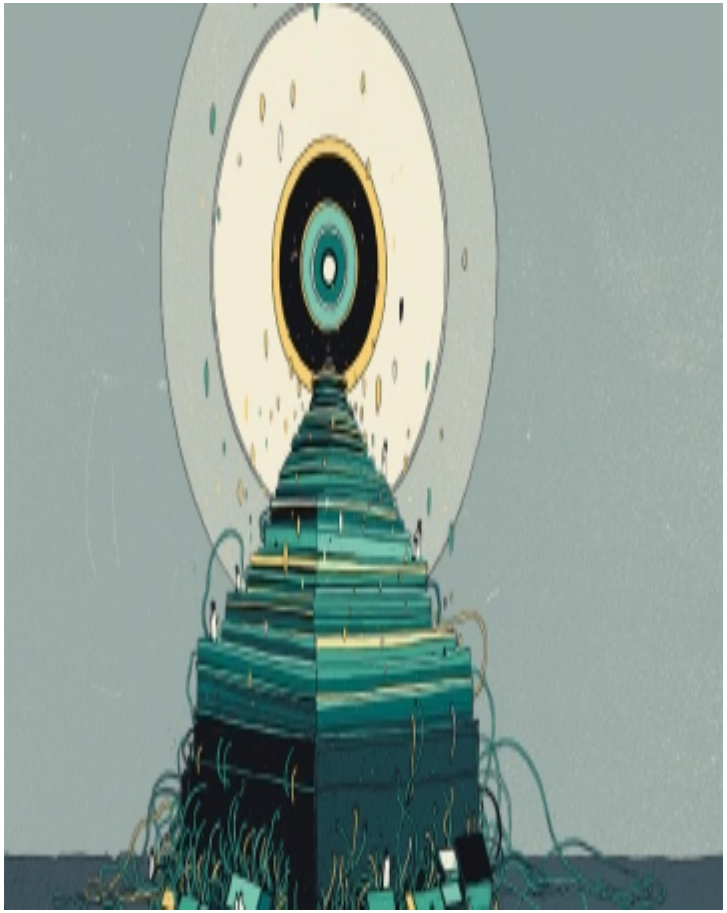
Black swan…

[Szegedy et al., Intriguing properties of neural networks, 2014]

After this "black swan", the issue of security of DNNs came to the fore…

Not only on scientific specialistic journals…

The safety issue to the fore…



The black box of AI
D. Castelvecchi, Nature, Vol. 538, 20, Oct 2016

Machine learning is becoming ubiquitous in basic research as well as in industry. But for scientists to trust it, they first need to understand what the machines are doing.

Ellie Dobson, director of data science at the big-data firm Arundo Analytics in Oslo: If something were to go wrong as a result of setting the UK interest rates, she says, "the Bank of England can't say, the black box made me do it".

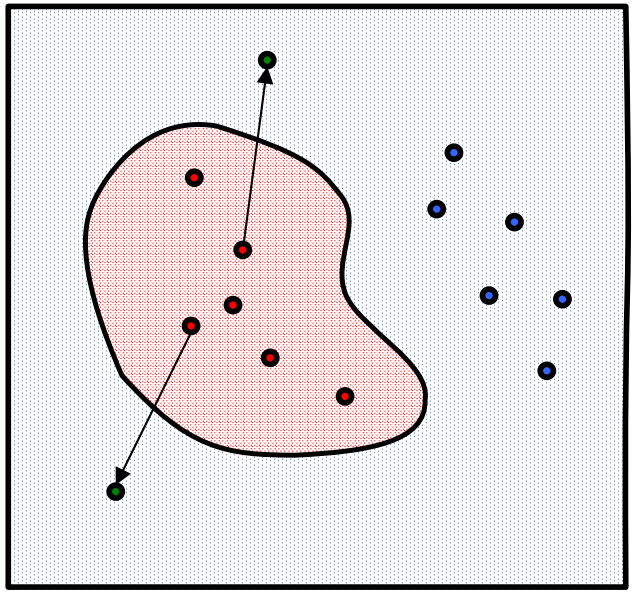
Why so much interest ?

Before the deep net "revolution", people were not surprised when machine learning was wrong, they were more amazed when it worked well…

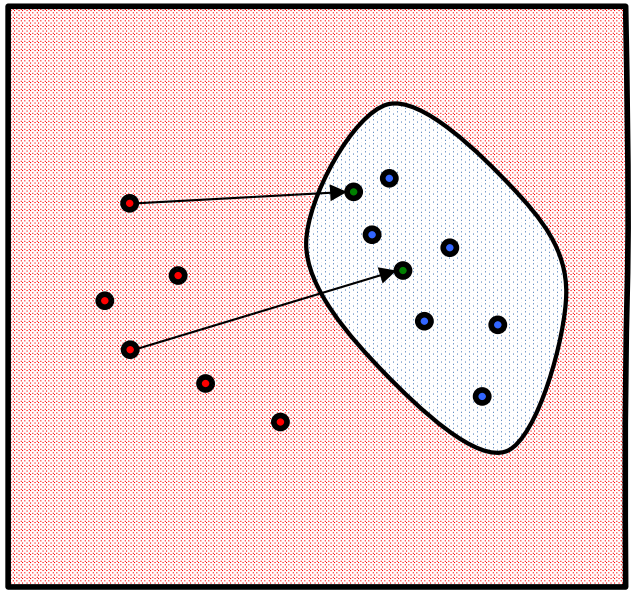
Now that it seems to work for real applications, people are disappointed, and worried, for errors that humans do not do…

But two issues should be considered…to avoid biases of the research on machine learning security

Two faces of the same coin…



blind-spot evasion



mimicry evasion

Instability, that causes the vulnerability, of deep neural nets is another manifestation of basic issues already seen for other pattern classifiers, issues also considered in computer security from a different perspective…

We should connect the dots…

Errors of humans and machines…

Machine learning decisions are affected by several **sources of bias** that causes “strange” errors

But we should keep in mind that also **humans** are **biased**…

The bat and the ball problem

A bat and a ball together cost $ 1.10

The bat costs $ 1.0 more than the ball

How much does the ball cost ?

Please, give me the first answer coming to your mind !

The bat and the ball problem

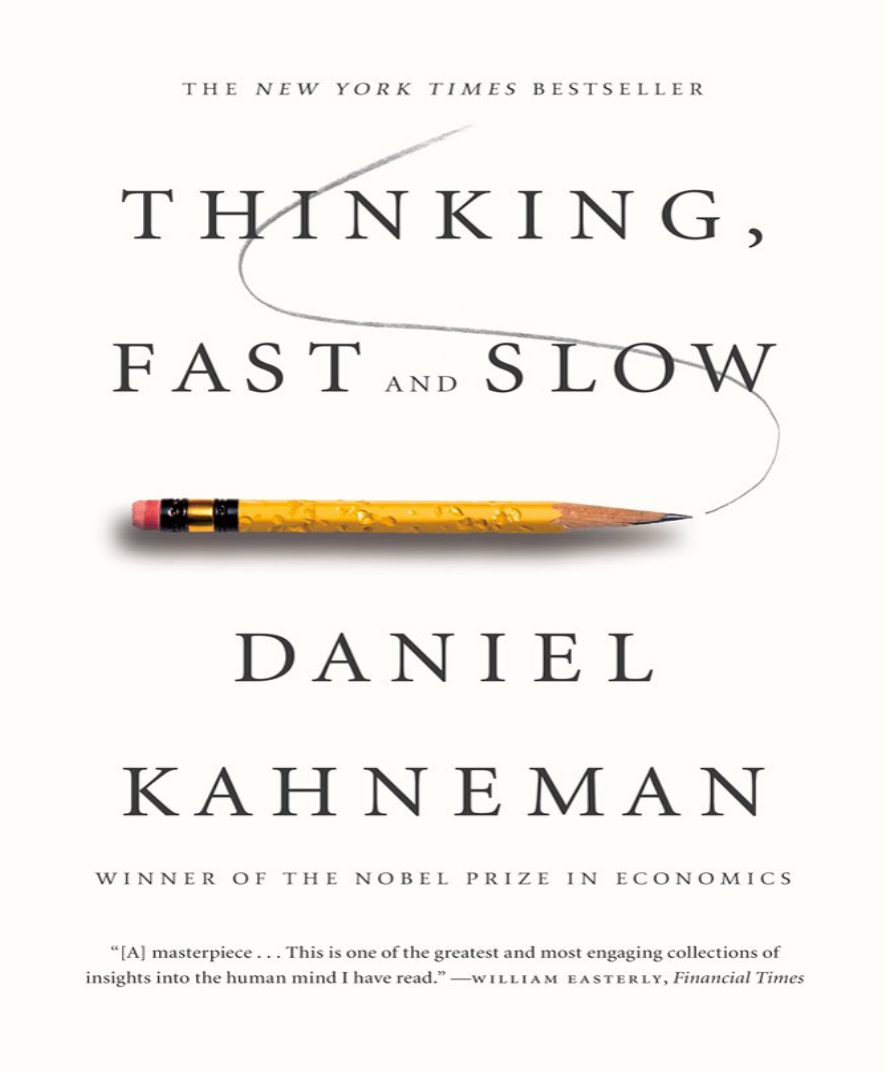
$$\begin{cases} \text{bat}+\text{ball}=\$1.10 \\ \text{bat}=\text{ball}+\$1.0 \end{cases}$$

Exact solution is 0.05 dollar (5 cents)

The wrong solution ($ 0.10) is due to the **attribute substitution**, a psychological process thought to underlie a number of **cognitive biases**

It occurs when an individual has to make a judgment (of a target attribute) that is computationally complex, and instead substitutes a more easily calculated heuristic attribute.

Trust in humans or machines ?



Algorithms are biased, but also humans are as well…

When should you trust in humans and when in algorithms?

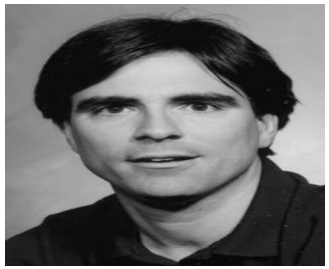
Learning comes at a price !

The introduction of novel **learning** functionalities increases the **attack surface** of computer systems and produces new vulnerabilities

Safety of machine learning will be more and more important in future computer systems, as well as **accountability**, **transparency**, and the protection of fundamental human **values** and **rights**

Thanks for listening !

Any questions ?



Engineering isn't about perfect solutions; it's about doing the best you can with limited resources
(Randy Pausch, 1960-2008)