Pluribus One
seeing one in many

Pattern Recognition
and Applications Lab

University of
Cagliari, Italy

# Adversarial Machine Learning

*Fabio Roli*

MLA 2018, Nanjing, China, Nov. 4th 2018

# We Are Living in the Best of the Worlds…

AI is going to transform industry and business as electricity did about a century ago

(*Andrew Ng, Jan. 2017*)

# All Right? All Good?

# iPhone 5s and 6s with Fingerprint Reader…

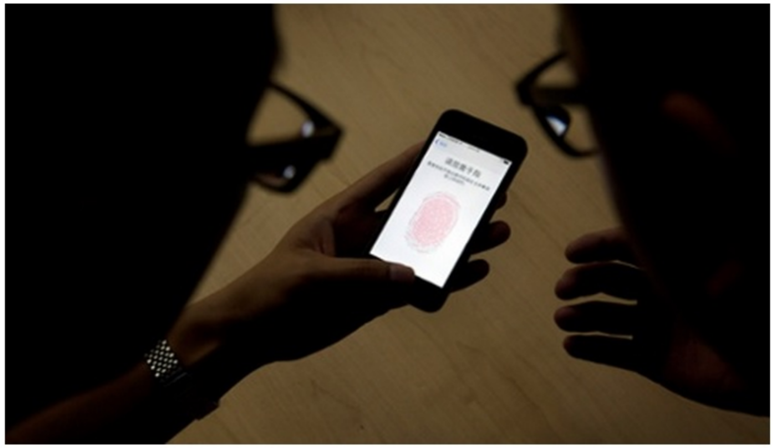
# Hacked a Few Days After Release…

## iPhone 5S fingerprint sensor hacked by Germany's Chaos Computer Club

Biometrics are not safe, says famous hacker team who provide video showing how they could use a fake fingerprint to bypass phone's security lockscreen

Follow Charles Arthur by email BETA

**Charles Arthur**
theguardian.com, Monday 23 September 2013 08.50 BST
Jump to comments (306)



Home › iPhone 6 › Your iPhone Can Be Hacked With A Photo Of Your Thumb

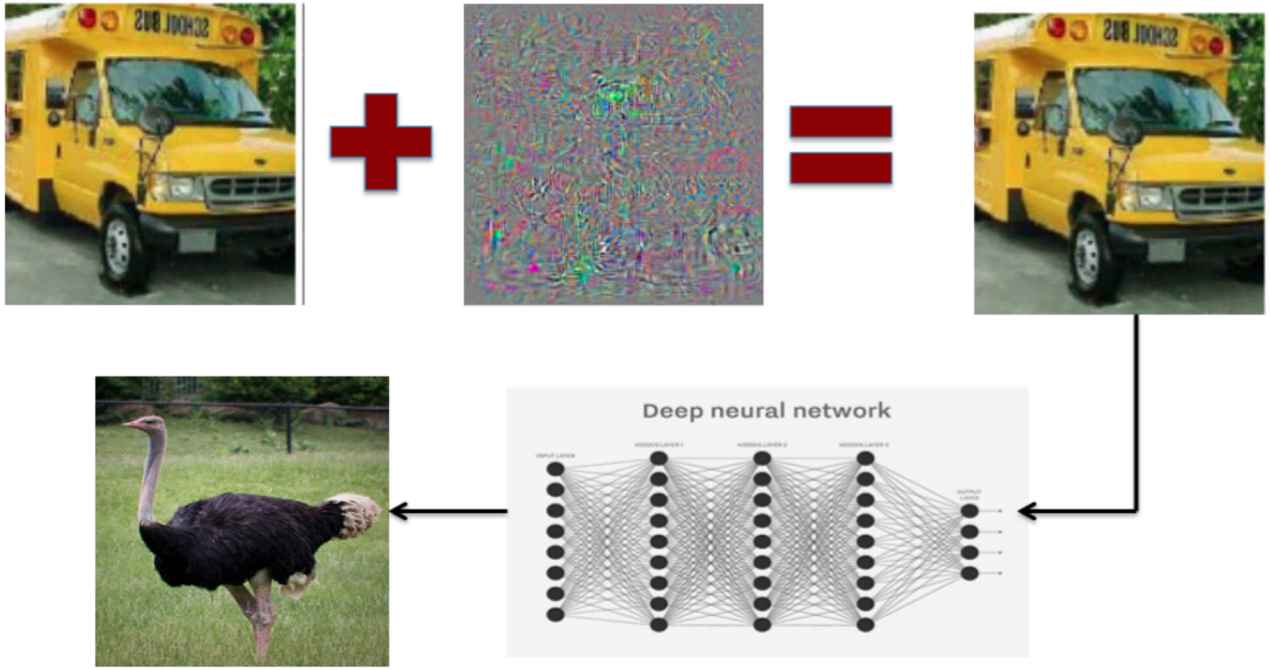
## Your iPhone Can Be Hacked With A Photo Of Your Thumb

Your fingerprint may not keep your iPhone safe any more. Someone has figured out how to use photos and commercially available software to break through an iPhone 6's fingerprint sensor, known as Touch ID.

**But maybe this happens only for old, shallow machine learning…**

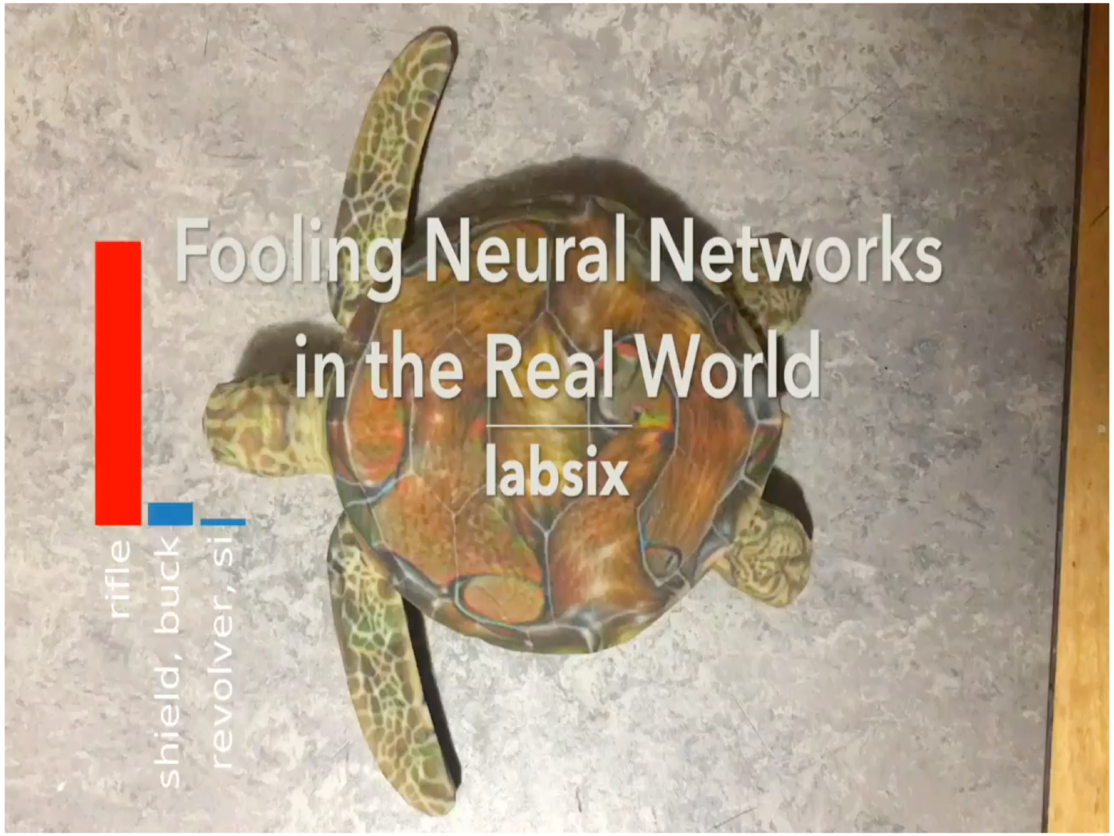
**end-to-end deep learning is another story…**

# Adversarial School Bus




*Biggio, Roli et al.*, Evasion attacks against machine learning at test time, **ECML-PKDD 2013**
*Szegedy et al.*, Intriguing properties of neural networks, **ICLR 2014**

# Adversarial Turtle…



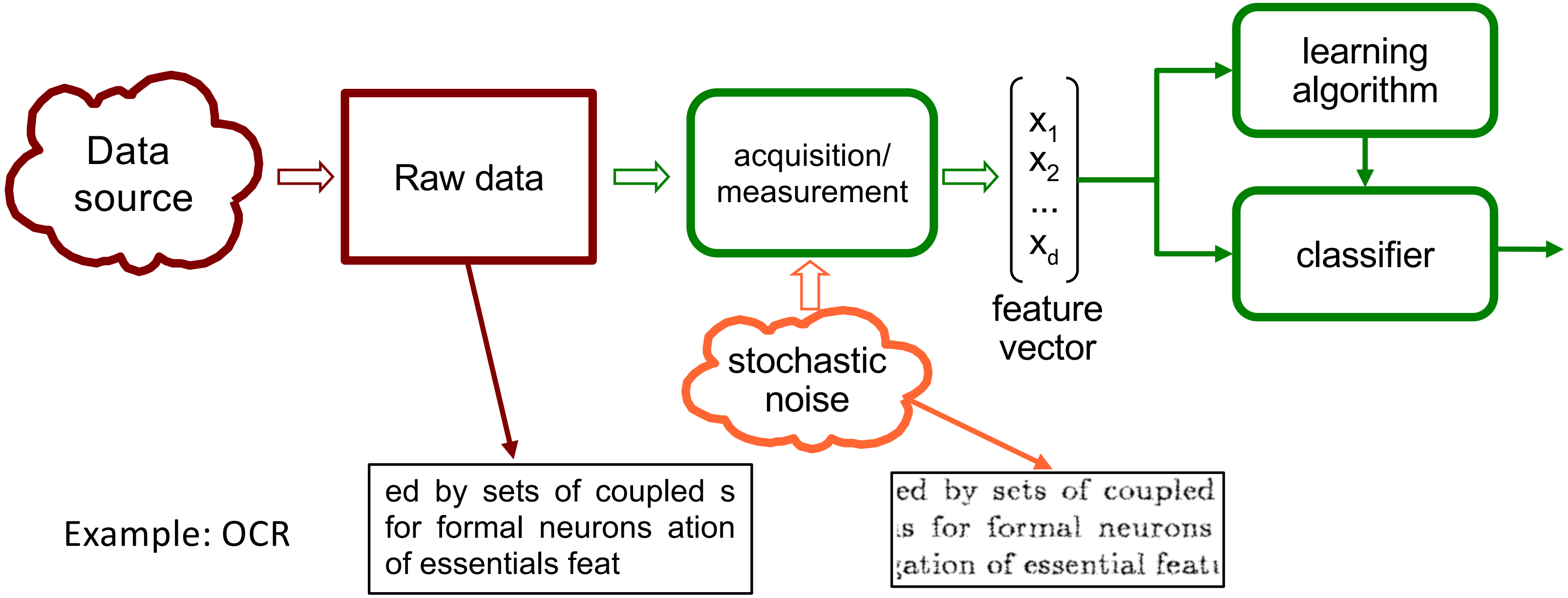
# Take-home Message

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*

# The Classical Statistical Model

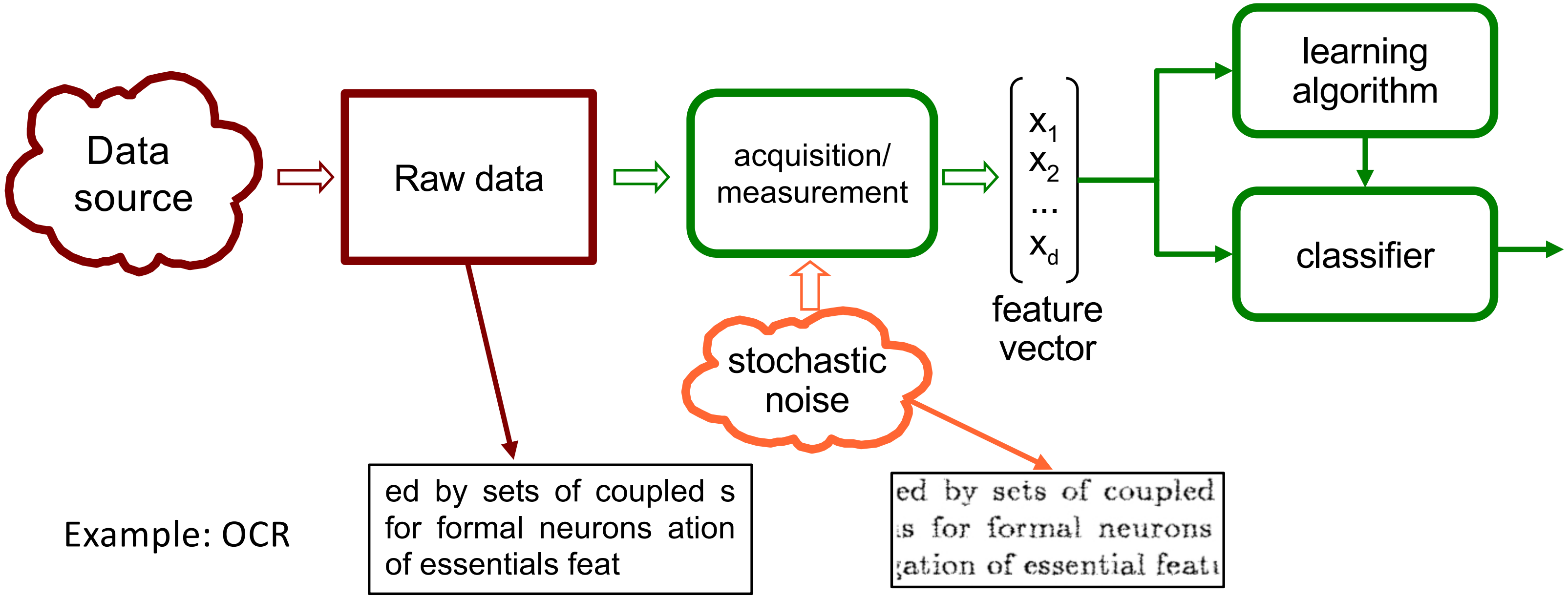



Example: OCR

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?



# An Example: Spam Filtering

Feature weights
buy = 1.0
viagra = 5.0

From: spam@example.it
Buy Viagra !

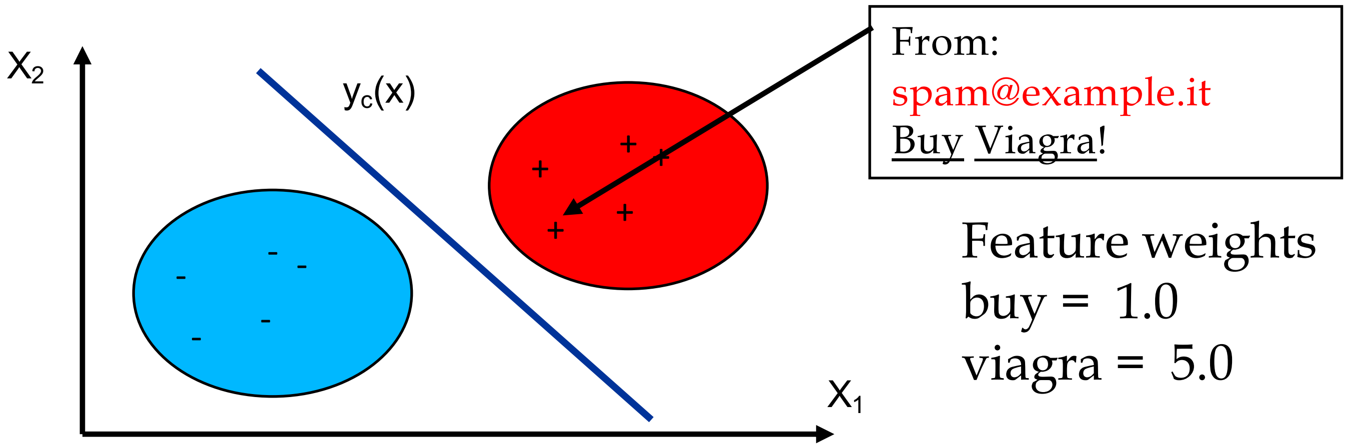
Linear Classifier

Total score = 6.0 > 5.0 (threshold)

**Spam**

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

# Feature Space View




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
conference = -2.0
meeting = -3.0

From: spam@example.it
Buy Viagra !
conference meeting

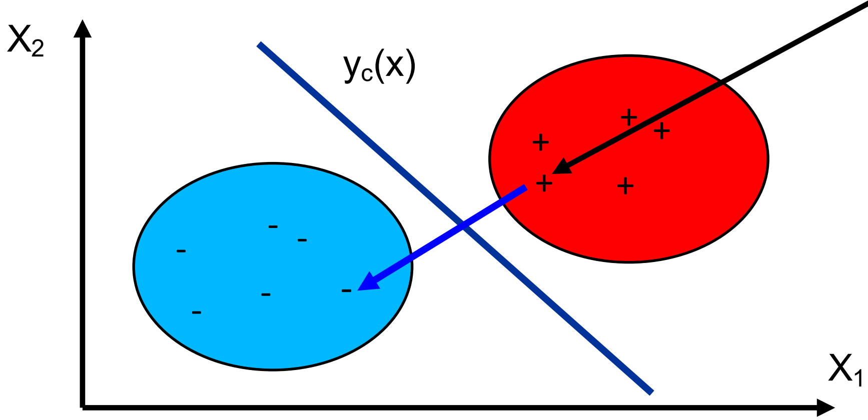
Linear Classifier

Total score = 1.0 < 5.0 (threshold)

**Ham**

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

# Adding Good Words: Feature Space View

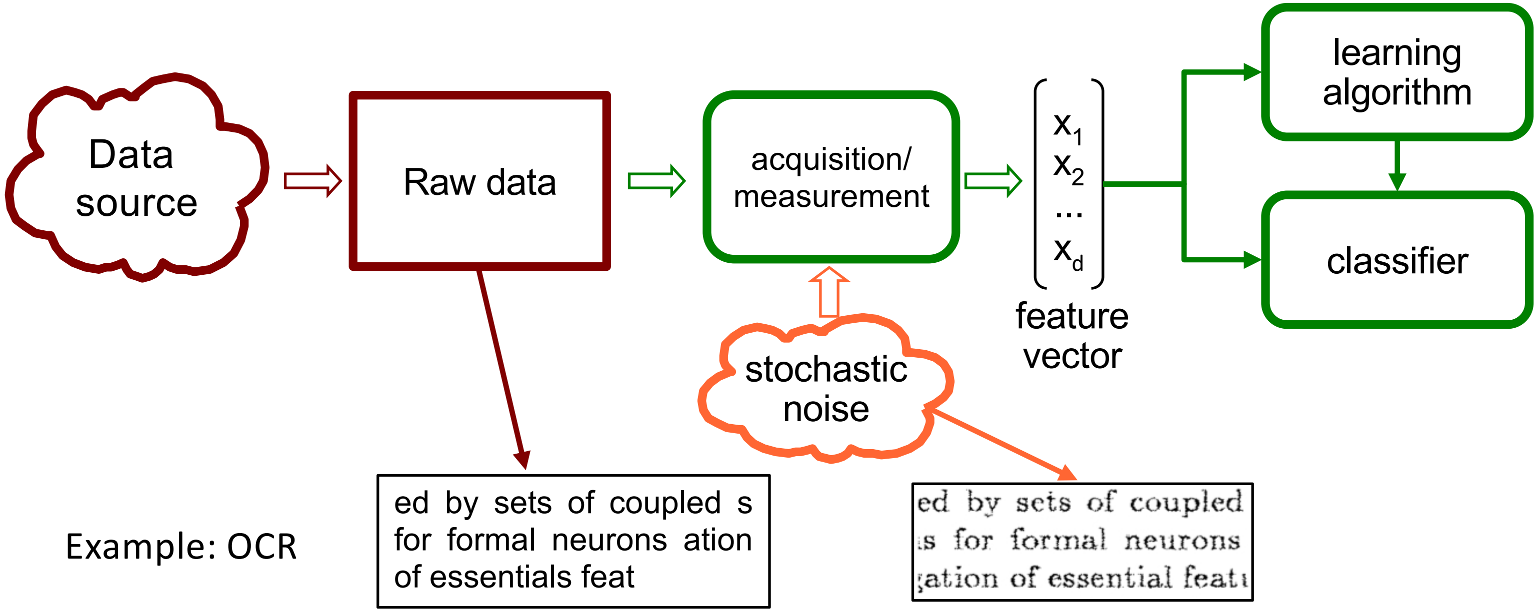



From: spam@example.it
Buy Viagra!
conference meeting

***Feature weights***
buy = 1.0
viagra = 5.0
conference = -2.0
meeting = -3.0

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

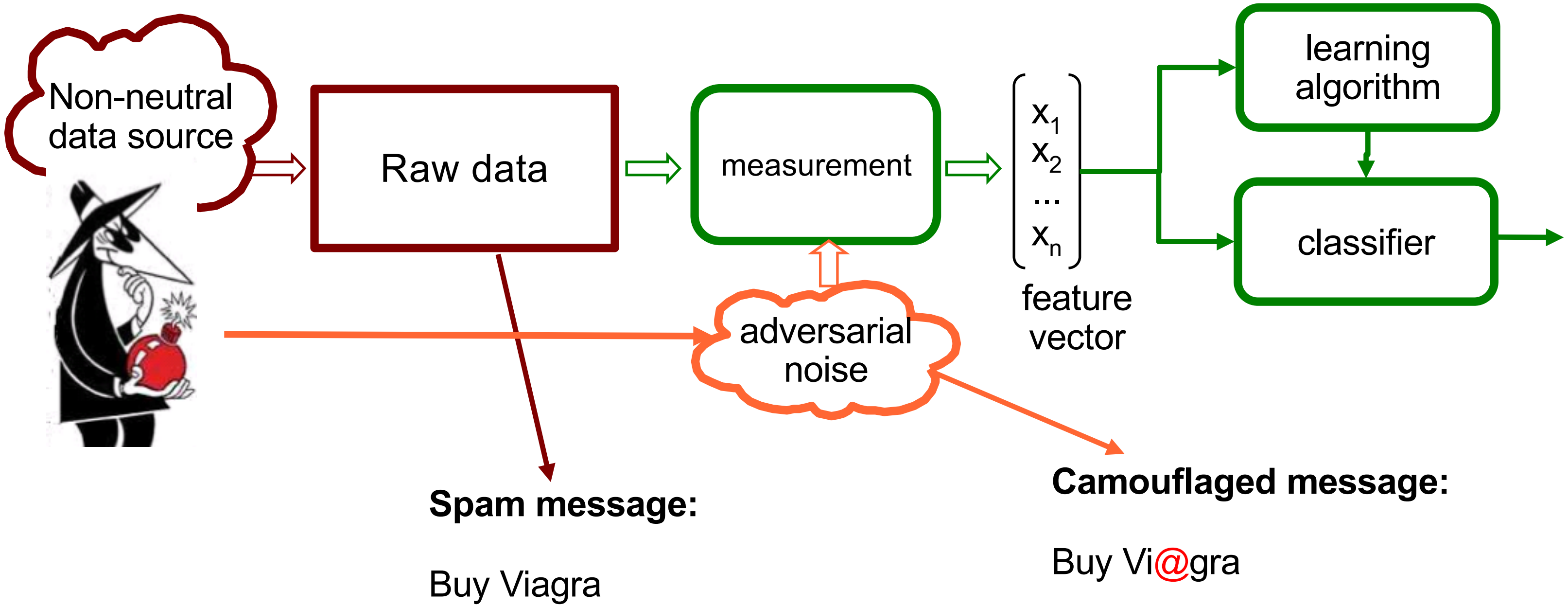
# Is This Model Good for Spam Filtering?




- 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

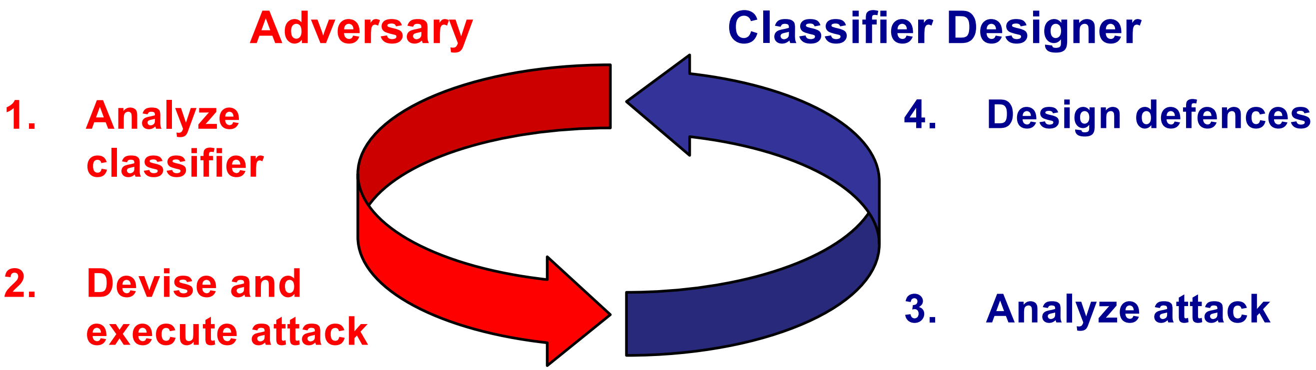
- Easy to see that classifier performance will degrade quickly if the adversarial noise is not taken into account

- Adversarial tasks are a mission impossible for the classical model

## How Should We Design Pattern Classifiers Under Attack?

# Adversary-aware Machine Learning

[Biggio, Fumera, Roli. Security evaluation of pattern classifiers under attack, IEEE TKDE, 2014]




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

## 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

# Attacks against Machine Learning

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

**Attacker's Knowledge:**

- perfect-knowledge (PK) white-box attacks
- limited-knowledge (LK) black-box attacks (*transferability* with surrogate/substitute learning models)



[Biggio & Roli, Wild Patterns, 2018 https://arxiv.org/abs/1712.03141]

# Be Proactive

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

# Be Proactive

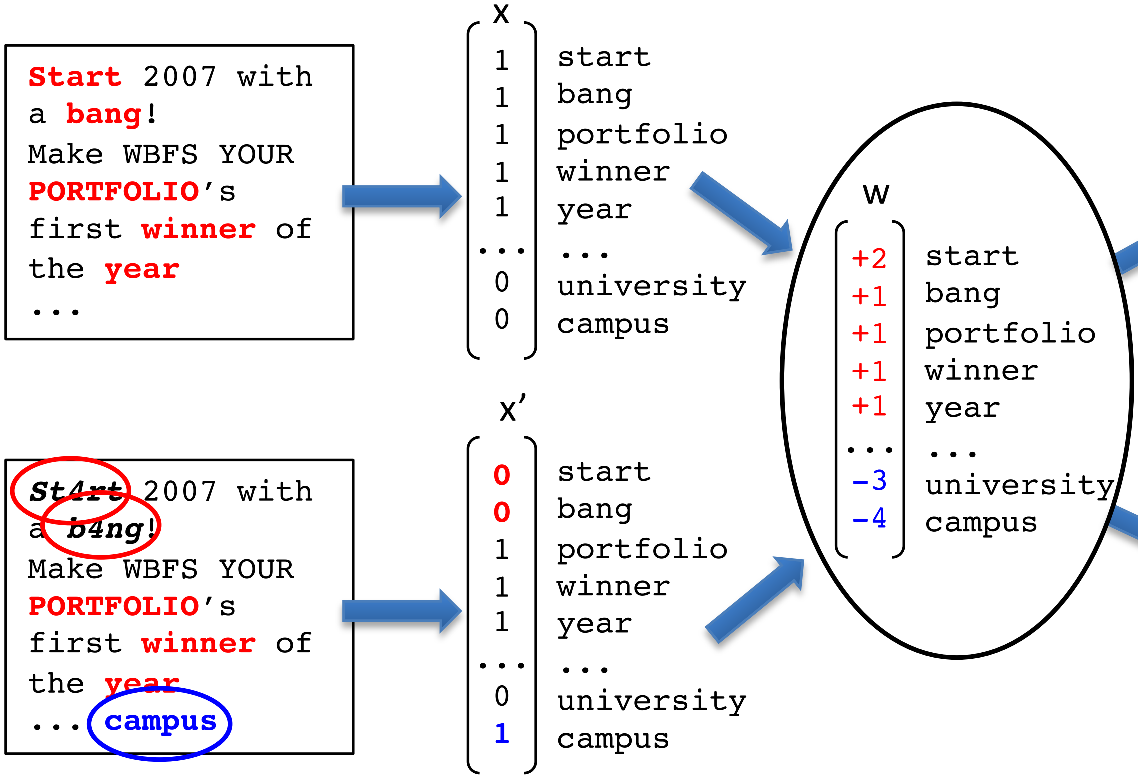
- Given a model of the adversary characterized by her:
  - **Goal**
  - **Knowledge**
  - **Capability**

*Try to anticipate the adversary!*

- What is the optimal attack she can do?
- What is the expected performance decrease of your classifier?

# Evasion Attacks
## (also known as *Adversarial Examples*)

# Evasion of Linear Classifiers




$f(x) = \mathrm{sign}(w^T x)$

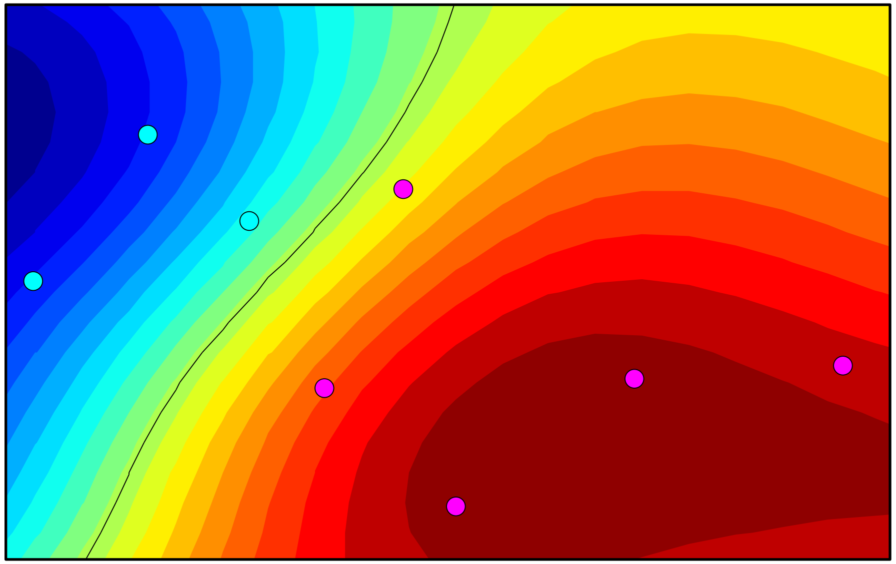
+6 > 0, SPAM
(correctly classified)

$f(x) = \mathrm{sign}(w^T x)$

+3 -4 < 0, HAM
(misclassified email)

# 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**)



# Evasion Attacks against Machine Learning at Test Time

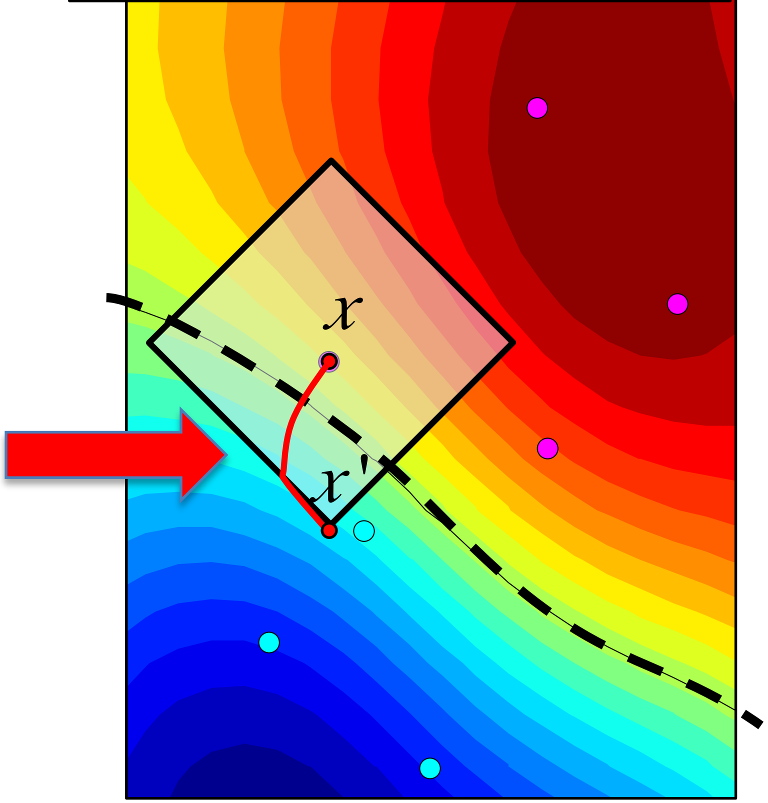
**Biggio, Corona, Maiorca, Nelson, Srndic, Laskov, Giacinto, Roli, ECML-PKDD 2013**

- **Goal:** maximum-confidence *evasion*
- **Knowledge:** *perfect (white-box attack)*
- **Attack strategy:**

$$\min_{x'} g(x')$$
$$\text{s.t.} \; \|x - x'\|_p \leq d_{\max}$$

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

$$f(x) = \text{sign}\left(g(x)\right) = \begin{cases} +1, & \text{malicious} \\ -1, & \text{legitimate} \end{cases}$$



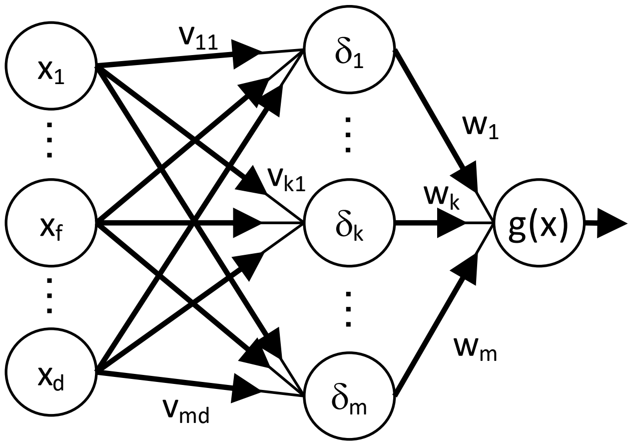
# Computing Descent Directions

## 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

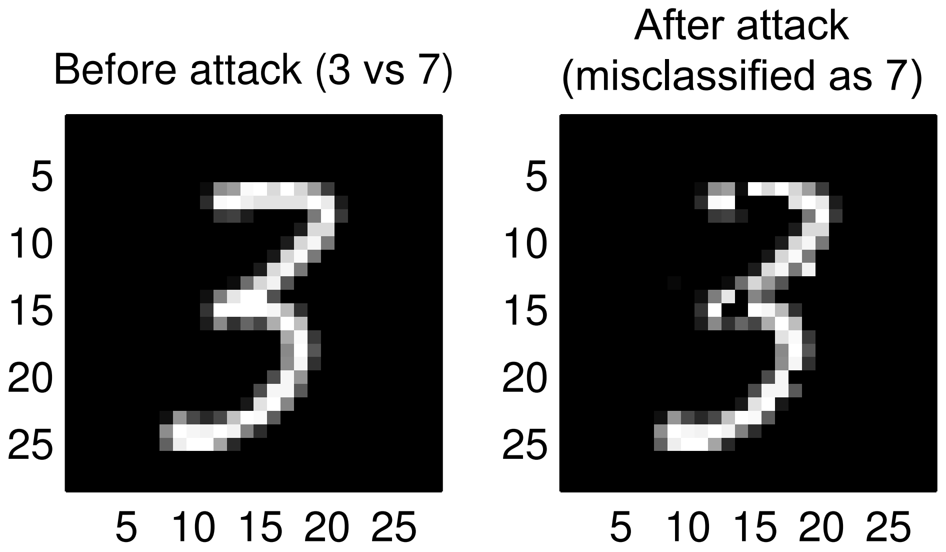



$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^{m} w_k \delta_k(x)\right)\right]^{-1}$$

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

# 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 examples* generated with gradient-based attacks date back to 2013!**

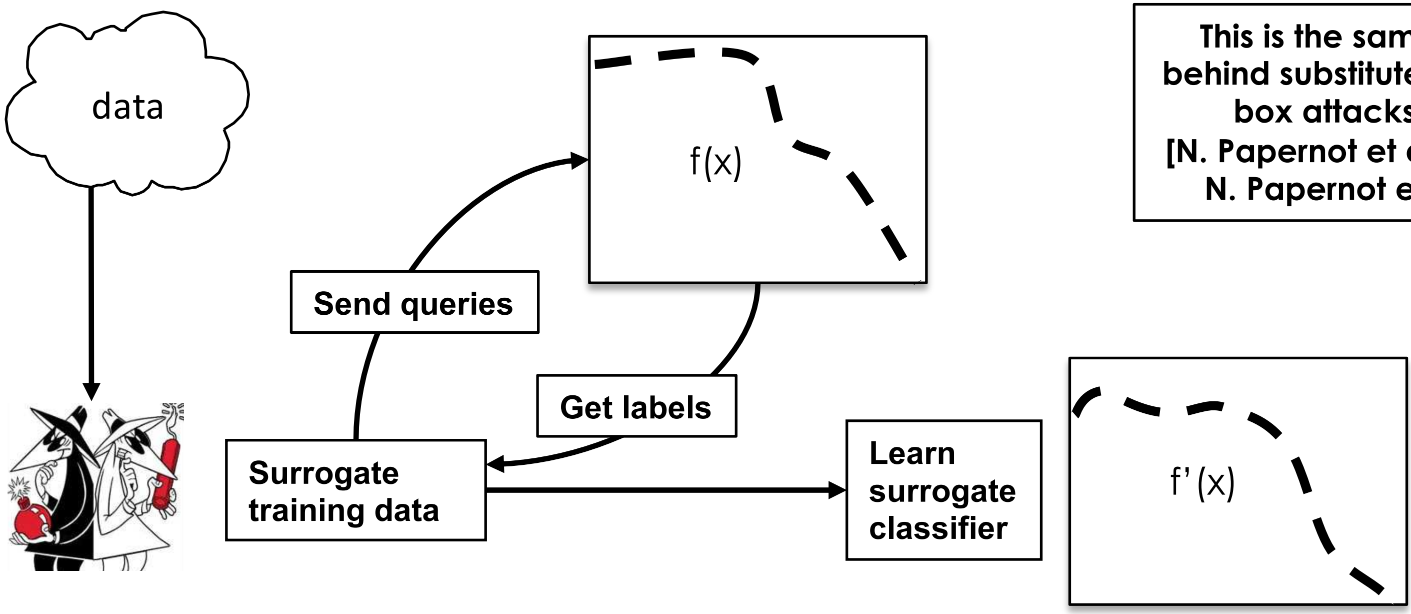
**(one year before attacks to deep neural networks)**



[Biggio, Roli et al., ECML PKDD 2013]

# Bounding the Adversary's Knowledge
## Limited-knowledge (black-box) attacks

- Only feature representation and (possibly) learning algorithm are known
- Surrogate data sampled from the same distribution as the classifier's training data
- Classifier's feedback to label surrogate data



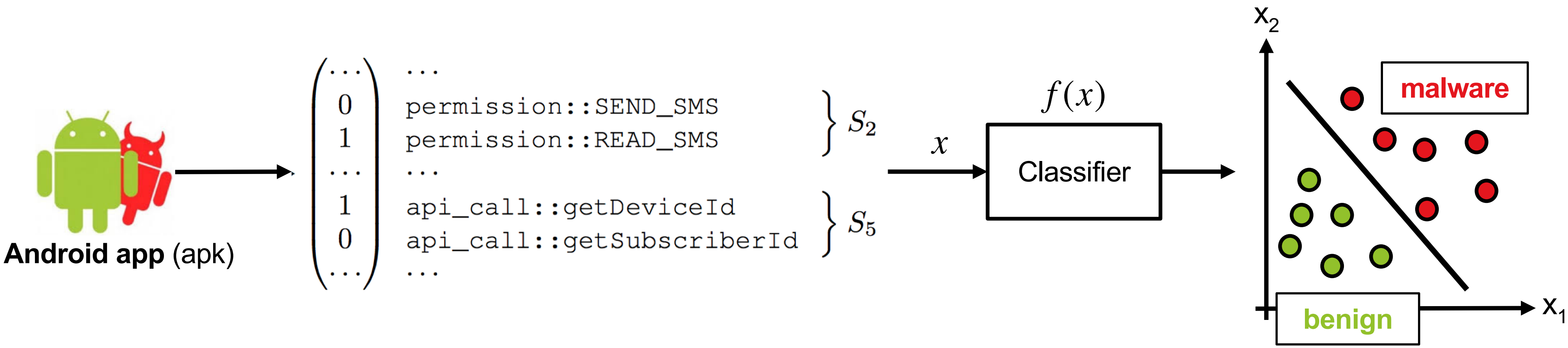

**This is the same underlying idea behind substitute models and black-box attacks (*transferability*)**
**[N. Papernot et al., IEEE Euro S&P '16; N. Papernot et al., ASIACCS'17]**

[Biggio, Roli et al., ECML PKDD 2013]

# Recent Results on Android Malware Detection

- **Drebin:** Arp et al., NDSS 2014
  - Android malware detection directly on the mobile phone
  - Linear SVM trained on features extracted from static code analysis

| Feature sets | | |
|---|---|---|
| manifest | $S_1$ | Hardware components |
| | $S_2$ | Requested permissions |
| | $S_3$ | Application components |
| | $S_4$ | Filtered intents |
| dexcode | $S_5$ | Restricted API calls |
| | $S_6$ | Used permission |
| | $S_7$ | Suspicious API calls |
| | $S_8$ | Network addresses |

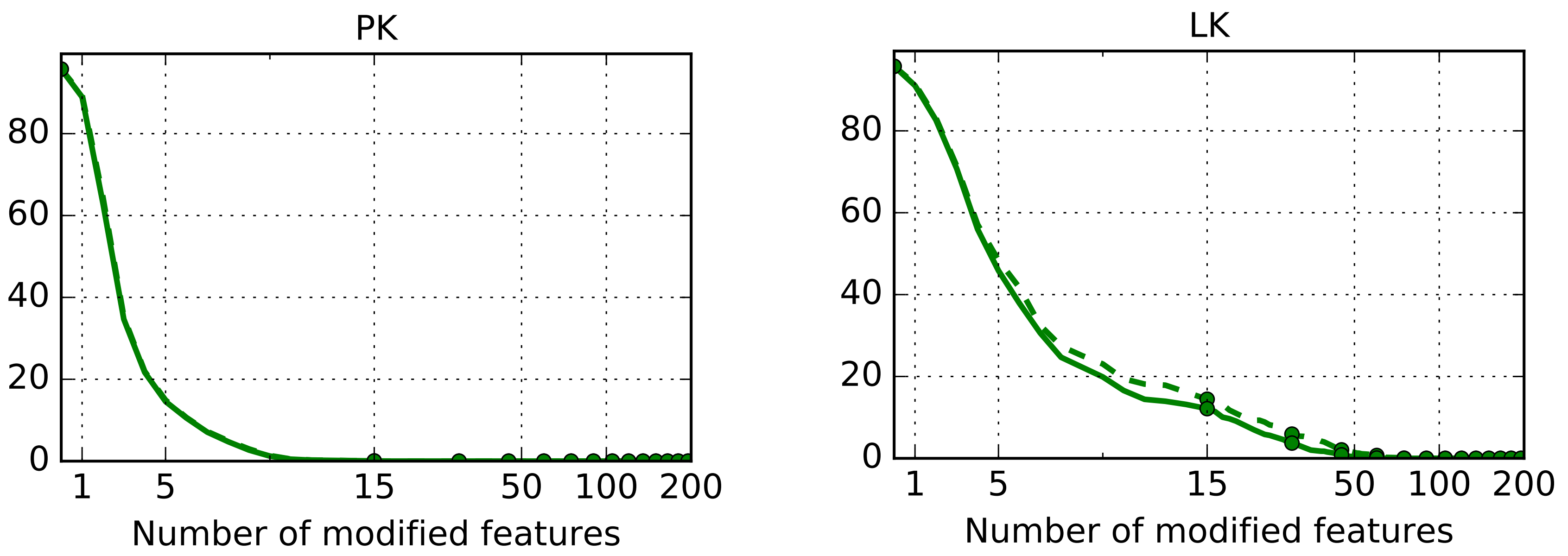





[Demontis, Biggio et al., IEEE TDSC 2017]

# Recent Results on Android Malware Detection

- **Dataset (Drebin):** 5,600 malware and 121,000 benign apps (TR: 30K, TS: 60K)
- **Detection rate** at FP=1% vs max. number of manipulated features (averaged on 10 runs)
  - Perfect knowledge (PK) white-box attack; Limited knowledge (LK) black-box attack



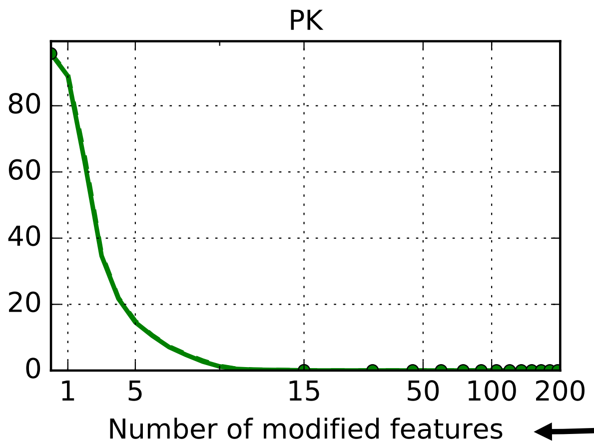



[Demontis, Biggio et al., IEEE TDSC 2017]

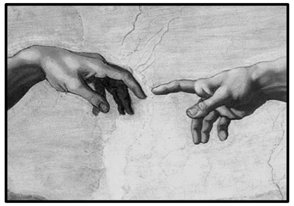
# Take-home Messages

- 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**




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



# 2014: Deep Learning Meets Adversarial Machine Learning

# The Discovery of Adversarial Examples

## 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.

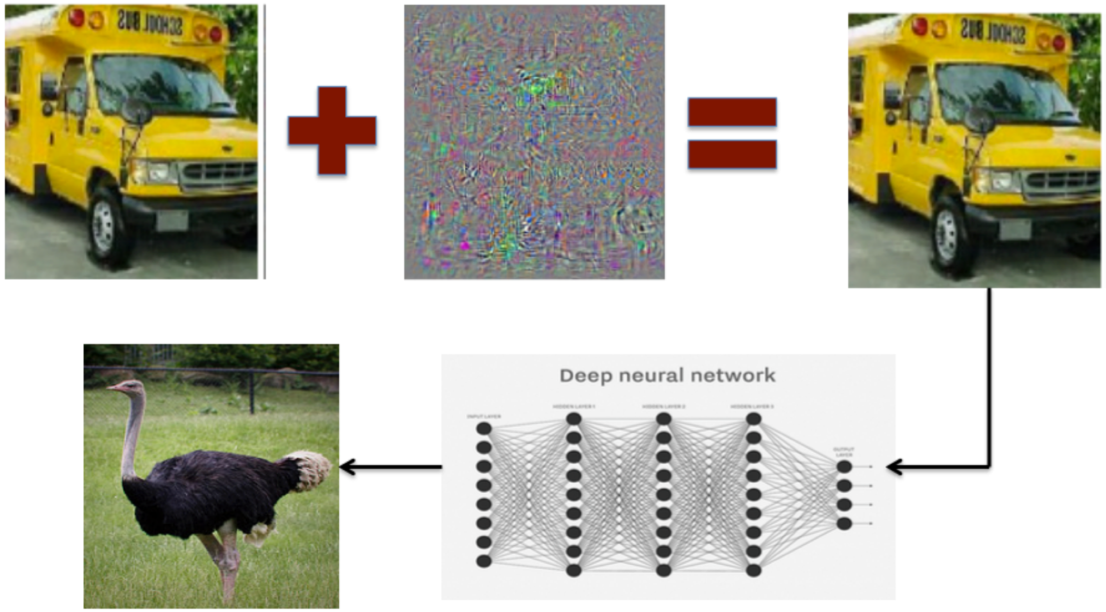
... we find that deep neural networks learn **input-output mappings** that are fairly **discontinuous** to a significant extent. We can cause the network to misclassify an image by applying a certain **hardly perceptible perturbation**, which is found by maximizing the network's prediction error ...



[Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014, ArXiv 2013]

# Adversarial Examples and Deep Learning

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




[Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014, ArXiv 2013]

# Creation of Adversarial Examples

- Minimize $\|r\|_2$ subject to:
  1. $f(x+r) = l \qquad f(x) \neq l$
  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 (*x*)**

**Adversarial Noise (*r*)**

**Ostrich**
**Struthio Camelus**



[Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014, ArXiv 2013]

# Many Adversarial Examples After 2014…

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

Several defenses have been proposed against adversarial examples, and more powerful attacks have been developed to show that they are ineffective.

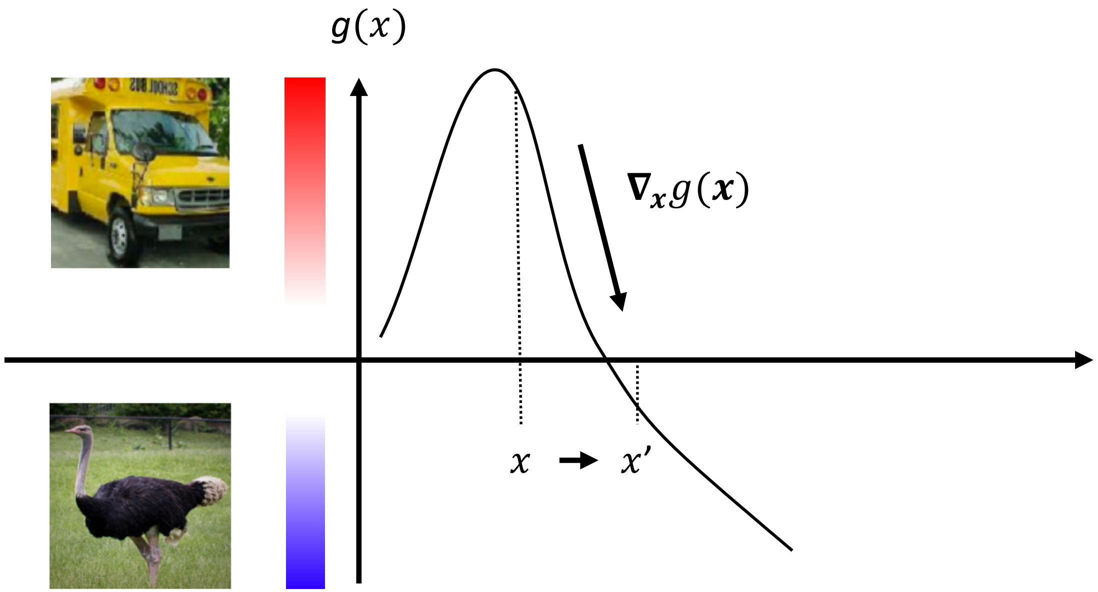
Most of these attacks are modifications to the optimization problems reported for evasion attacks / adversarial examples, using different gradient-based solution algorithms, initializations and stopping conditions.

Most popular attack algorithms: FGSM (Goodfellow et al.), JSMA (Papernot et al.), CW (Carlini & Wagner, and follow-up versions)

# Why Adversarial Perturbations are Imperceptible?

# Why Adversarial Perturbations against Deep Networks are Imperceptible?

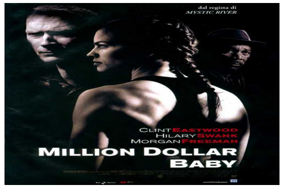
- Large sensitivity of g(**x**) to input changes
  - i.e., the **input gradient** $\nabla_x g(\boldsymbol{x})$ has a large norm (scales with input dimensions!)
  - Thus, even small modifications along that direction will cause large changes in the predictions






[Simon-Gabriel et al., Adversarial vulnerability of NNs increases with input dimension, ArXiv 2018]

# Countering Evasion Attacks




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

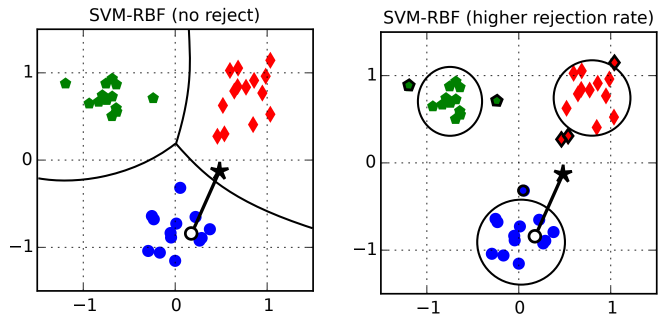
# Main Security Measures against Evasion Attacks

1. Reduce sensitivity to input changes with **robust optimization**
   – Adversarial Training / Regularization

$$\min_{\boldsymbol{w}} \sum_i \max_{||\boldsymbol{\delta}_i|| \le \epsilon} \ell(y_i, f_{\boldsymbol{w}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i))$$

**bounded perturbation!**

2. Introduce ***rejection / detection*** of adversarial examples




[Demontis, Biggio, Roli et al., Yes, Machine Learning Can Be More Secure! ..., IEEE TDSC 2017]
[Melis, Biggio, Roli et al., Is Deep Learning Safe for Robot Vision?...,ICCVW ViPAR 2017]

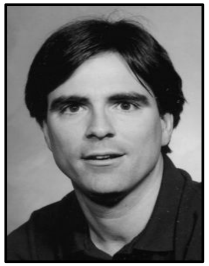
# Learning Comes at a Price!

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)*