Machine Learning Security

Battista Biggio
battista.biggio@unica.it

@biggiobattista

This course is inspired from the tutorial *Wild Patterns* held by Battista Biggio and Fabio Roli
https://www.pluribus-one.it/research/sec-ml/wild-patterns/

A Question to Start…

What is the oldest survey article on machine learning that you have ever read?

What is the publication year?

This Is Mine… Year 1966

Pattern Recognition

By DENIS RUTOVITZ

Medical Research Council

[Read before the ROYAL STATISTICAL SOCIETY on Wednesday, May 18th, 1966, the President, Mr L. H. C. TIPPETT, in the Chair]

1. INTRODUCTION

DURING the past 10 years about 200 articles and several books have appeared, dealing with machine recognition of optical and other patterns (mainly alphabetic characters and numerals). About half of these have described methods not linked to a specific

Credits: Dr Gavin Brown for showing me this article

Applications in the Old Good Days…

What applications do you think that this paper dealt with?

Pattern Recognition

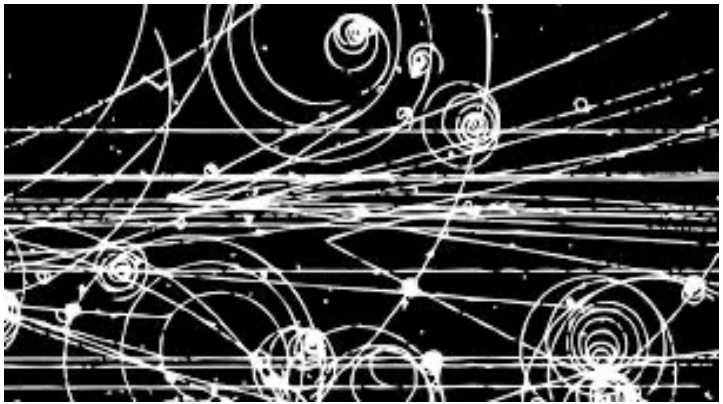
By DENIS RUTOVITZ

Medical Research Council

[Read before the ROYAL STATISTICAL SOCIETY on Wednesday, May 18th, 1966, the President, Mr L. H. C. TIPPETT, in the Chair]

Popular Applications in the Sixties

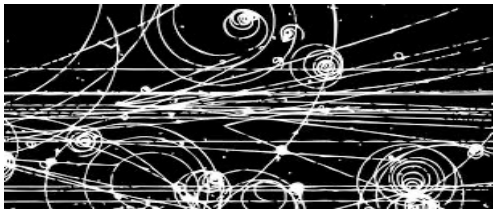
OCR for bank cheque sorting



Detection of particle tracks in bubble chambers

Aerial photo recognition

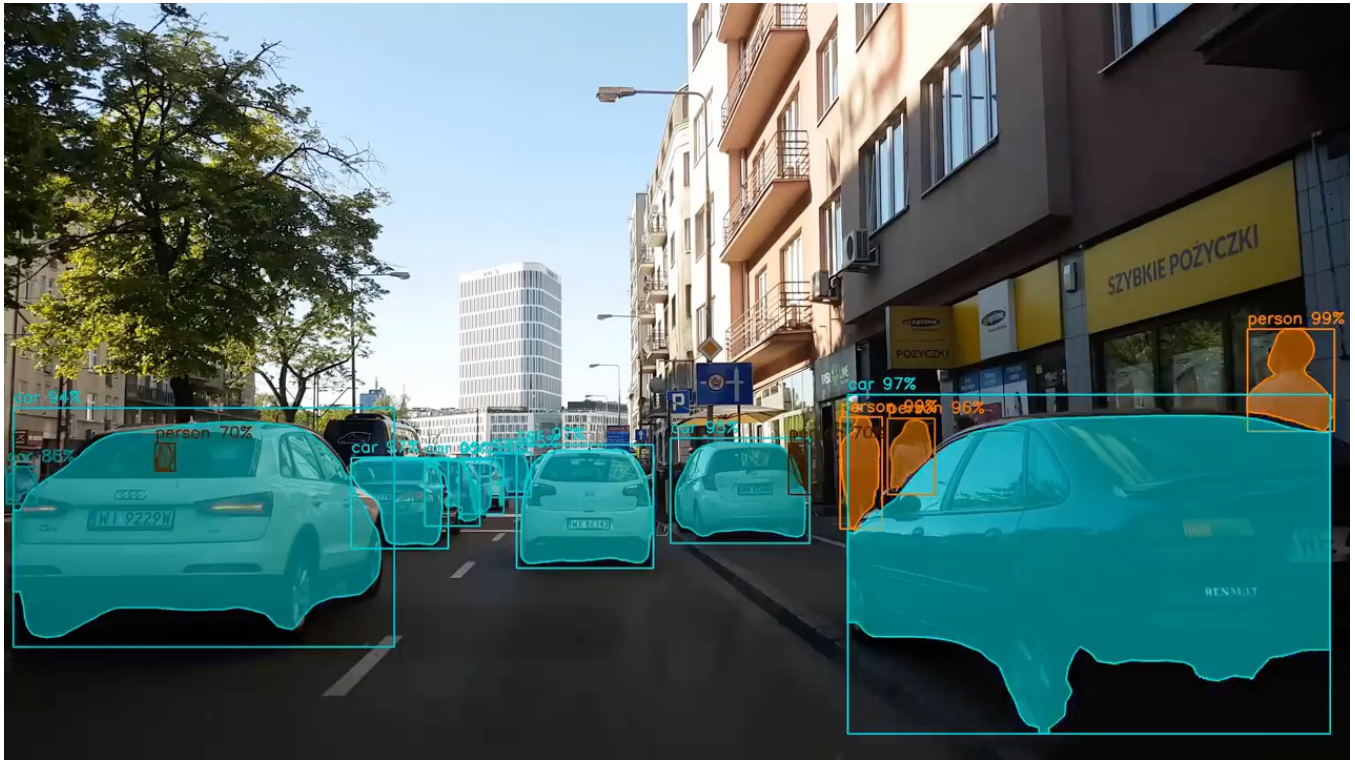
Key Feature of these Apps



Specialised applications for **professional** users…

What about Today Applications?

Computer Vision for Self-Driving Cars



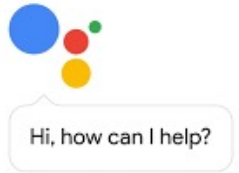
He et al., Mask R-CNN, ICCV '17, https://arxiv.org/abs/1703.06870
Video from: https://www.youtube.com/watch?v=OOT3UIXZztE

Automatic Speech Recognition for Virtual Assistants

Amazon Alexa

Apple Siri

Microsoft Cortana



Google Assistant

Today Applications of Machine Learning

Key Features of Today Apps

Personal and **consumer** applications…

Artificial Intelligence Today

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

(*Andrew Ng, Jan. 2017*)

Applications:

- Cybersecurity
- Robotics
- Healthcare
- Speech recognition
- Virtual assistants
- ...

But… What's the Difference between AI/ML?

Mat Velloso
@matvelloso

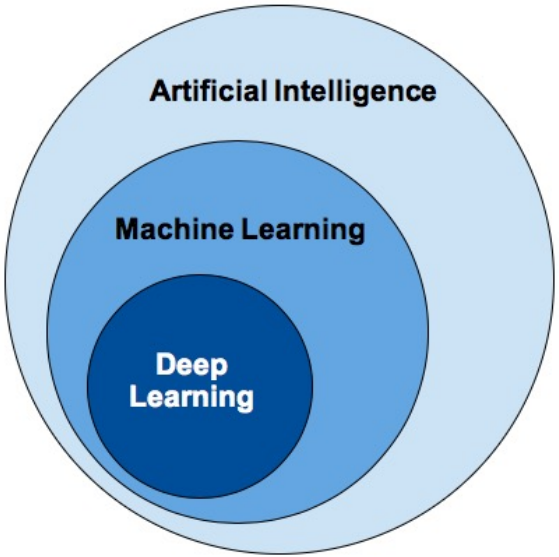
Difference between machine learning and AI:

If it is written in Python, it's probably machine learning

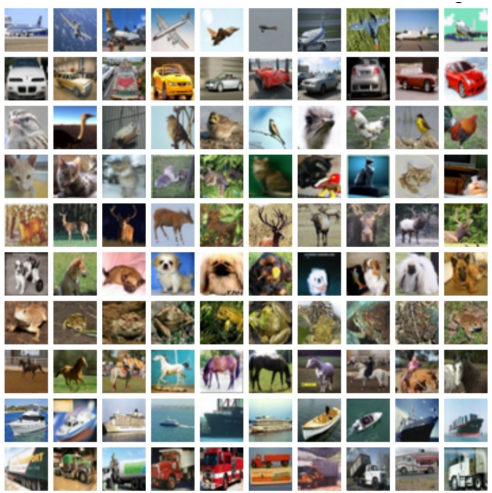
If it is written in PowerPoint, it's probably AI

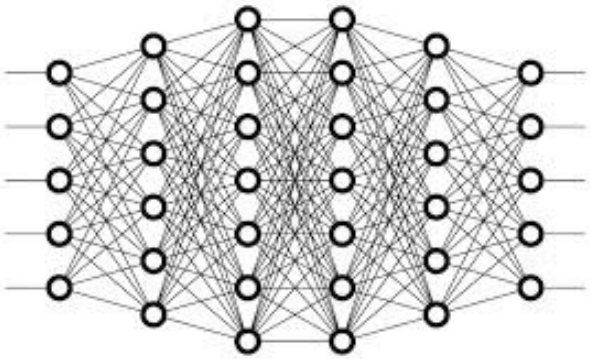
2:25 AM · Nov 23, 2018 · Twitter Web Client

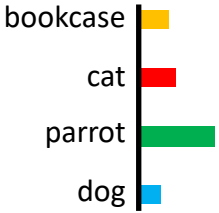
8.6K Retweets **24.1K** Likes



Modern AI is Numerical Optimization + Big Data







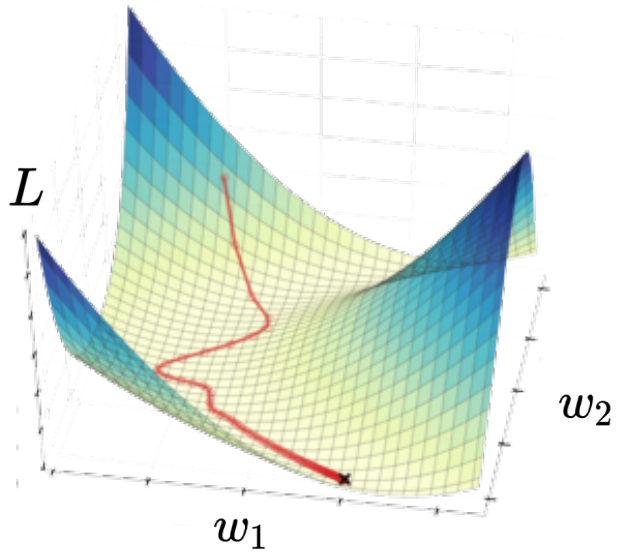
$$\min_{\boldsymbol{w}} L(D; \boldsymbol{w})$$

The goal is to minimize the fraction of *classification errors*

... by iteratively updating the classifier parameters $\boldsymbol{w}$ along the gradient direction $\nabla_{\boldsymbol{w}} L(D; \boldsymbol{w})$

The Workhorse of Machine Learning: *Gradient Descent*

$\mathbf{w} \leftarrow \mathbf{w}_0$
$i \leftarrow 0$
while $i < maxiter$ **do**
    $\mathbf{w} \leftarrow \mathbf{w} - \eta \, \nabla_{\mathbf{w}} L(\mathbf{X}, \mathbf{y})$
    $i \leftarrow i + 1$
end while
return $\mathbf{w}$



All Right? All Good?

iPhone 5s and 6s with Fingerprint Reader…

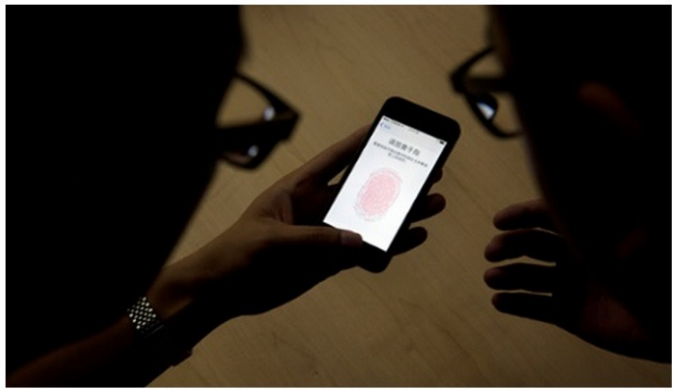
Hacked a Few Days After Release…

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

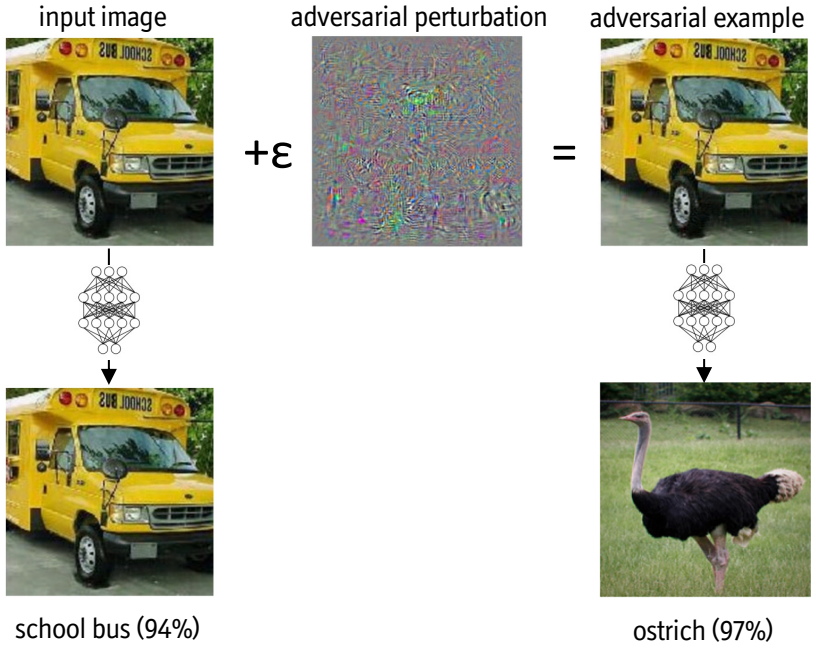
Charles Arthur
theguardian.com, Monday 23 September 2013 08.50 BST
Jump to comments (306)



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

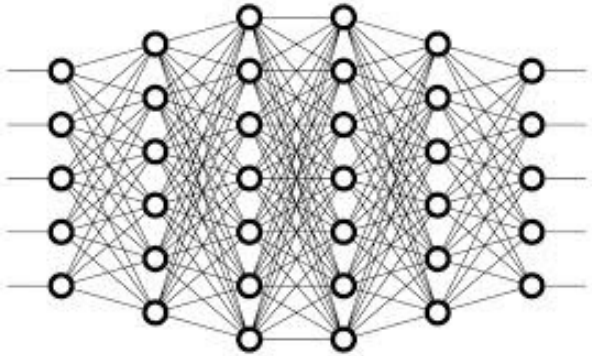
End-to-end deep learning is another story…

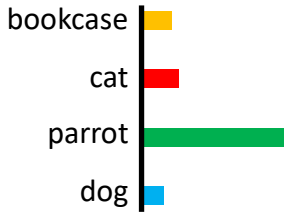
Adversarial Examples (Gradient-based Evasion Attacks)



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

Adversarial Attacks

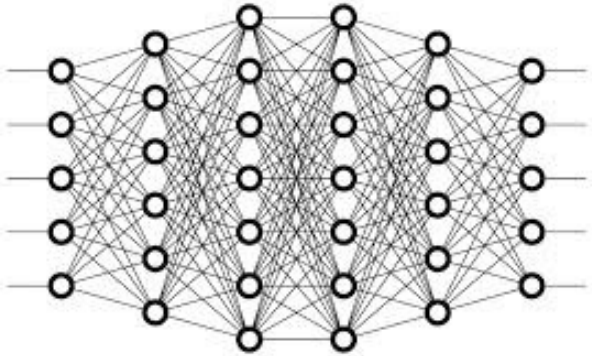




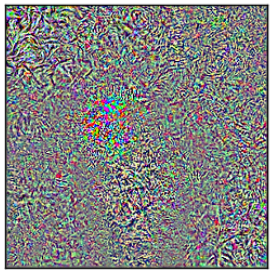
Adversarial attacks exploit the same underlying mechanism of learning, but aim to maximize the probability of error on the input data: $\max_{D} L(D; \boldsymbol{w})$

This problem can also be solved with gradient-based optimizers
(Biggio et al., ICML 2012; Biggio et al., ECML 2013; Szegedy et al., ICLR 2014)

How Do These Attacks Work?

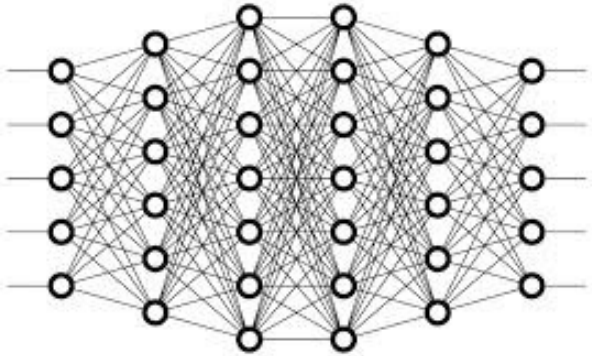


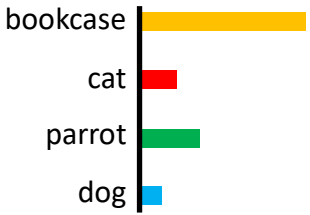
$$\max_{D} L(D; \boldsymbol{w})$$

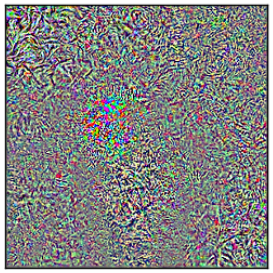


The gradient of the objective allows us to compute an *adversarial perturbation...*

How Do These Attacks Work?







... which is then added to the input image to cause misclassification

... not only in the digital domain!

Adversarial Road Signs

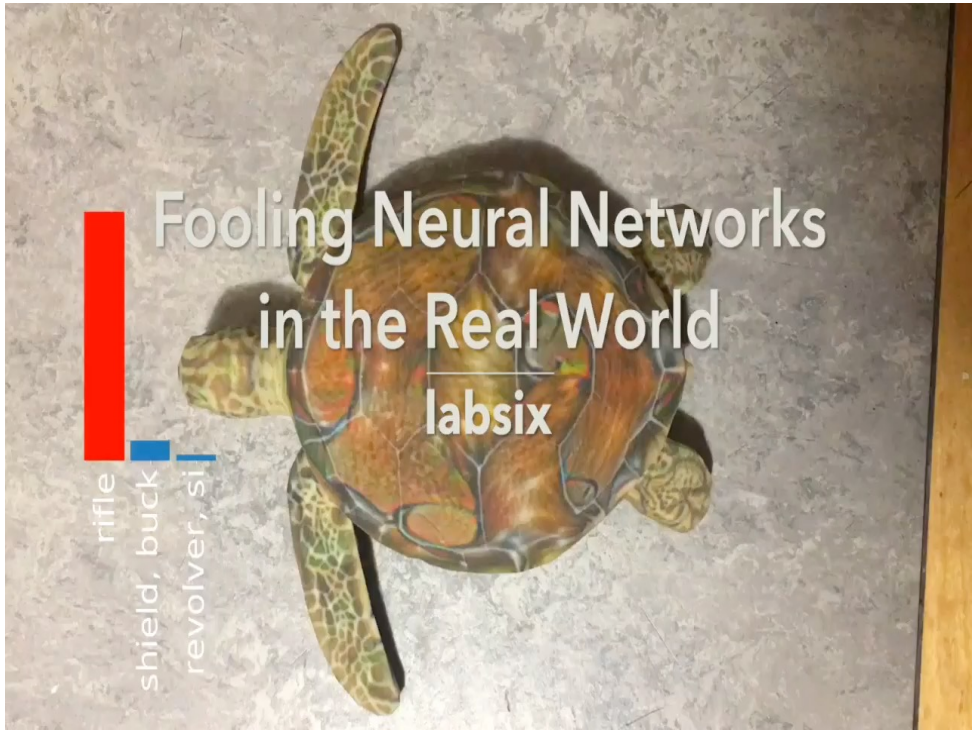
Eykholt et al., *Robust physical-world attacks on deep learning visual classification*, CVPR 2018

Adversarial Glasses

- Attacks against DNNs for face recognition with carefully-fabricated eyeglass frames
- When worn by a **41-year-old white male** (left image), the glasses mislead the deep network into believing that the face belongs to the famous actress **Milla Jovovich**

Sharif et al., *Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition*, ACM CCS 2016

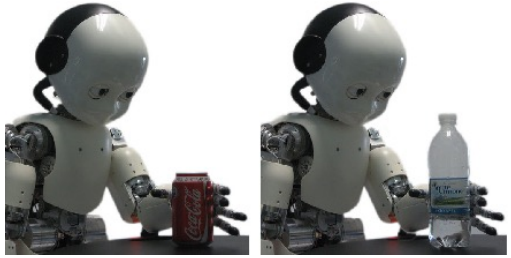
Adversarial Turtles

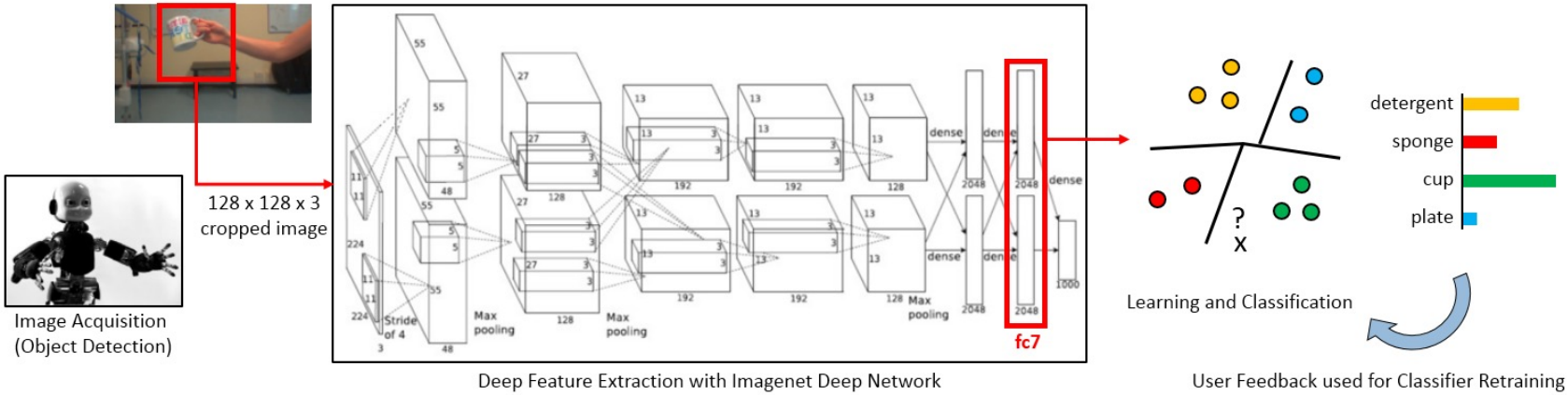


Athalye et al., *Synthesizing robust adversarial examples*. ICLR, 2018

Fooling the iCub Humanoid Robot

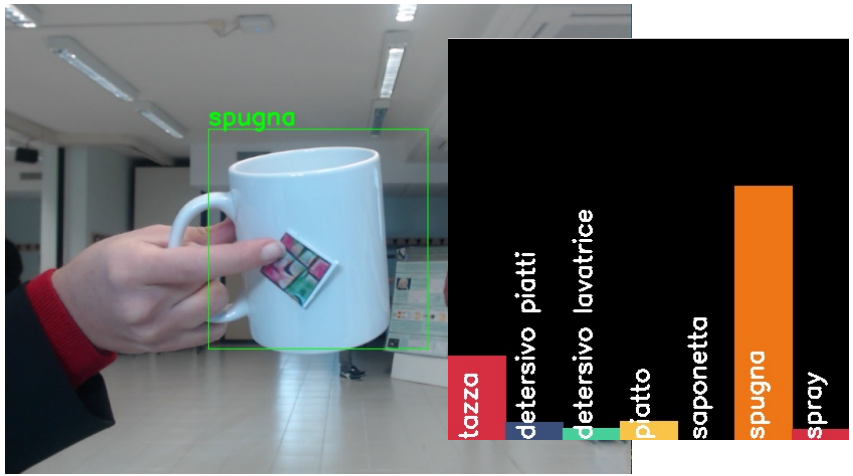
- Attacks against the iCub humanoid robot
  - Deep Neural Network used for visual object recognition

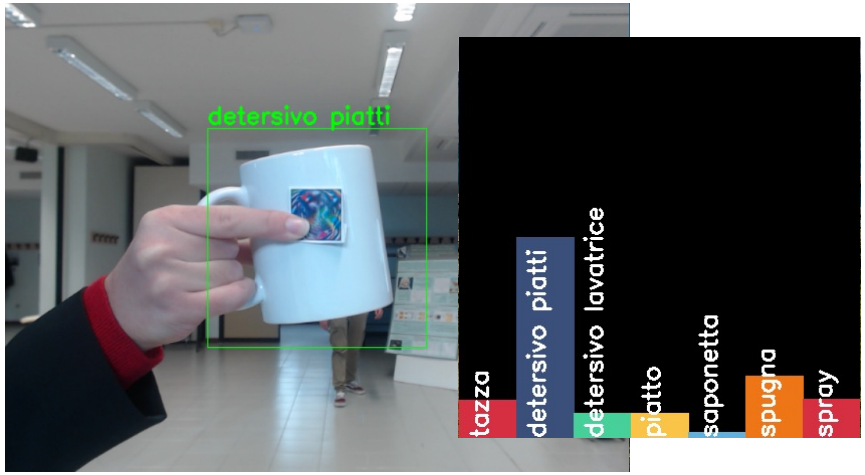




Melis, Biggio et al., ViPAR 2017

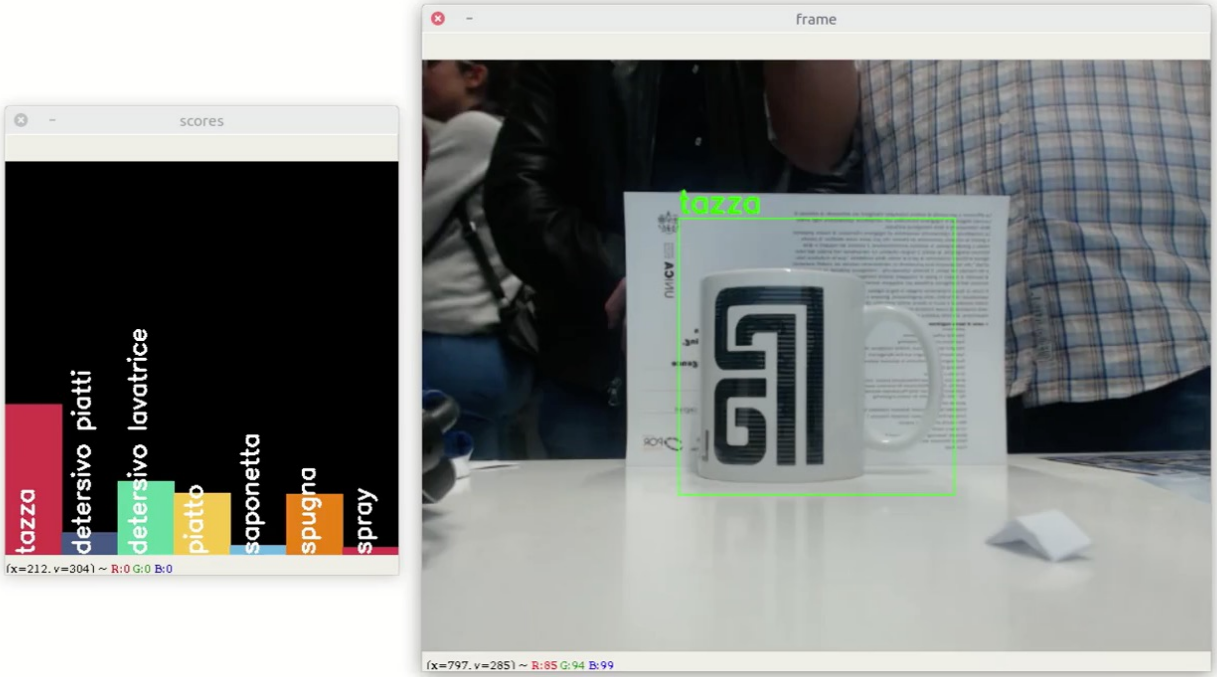
Adversarial Stickers against the iCub Humanoid Robot





ongoing work by: Sotgiu, Demontis, Biggio, Roli

Adversarial Stickers against the iCub Humanoid Robot



ongoing work by: Sotgiu, Demontis, Biggio, Roli

But maybe this happens only for image recognition…

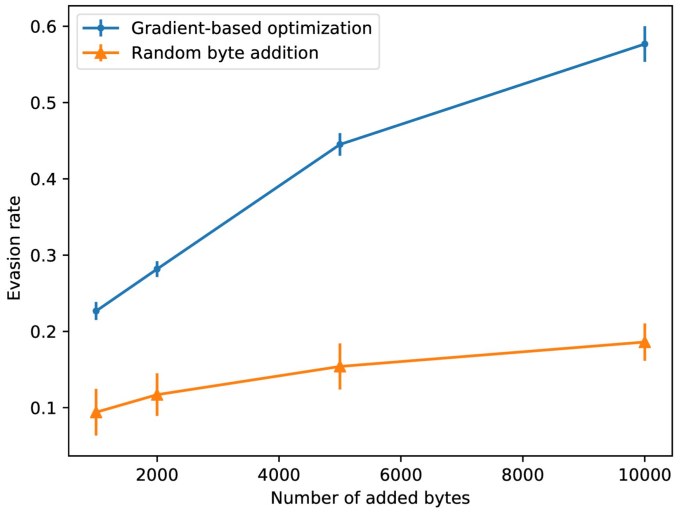
Audio Adversarial Examples

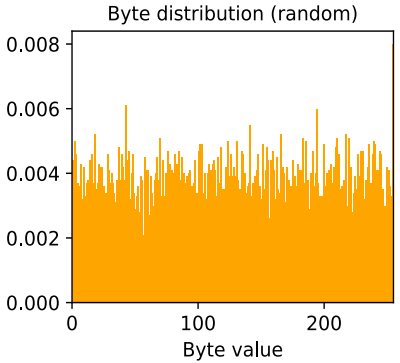
Audio	Transcription by Mozilla DeepSpeech
	“without the dataset the article is useless”
	“okay google browse to evil dot com”

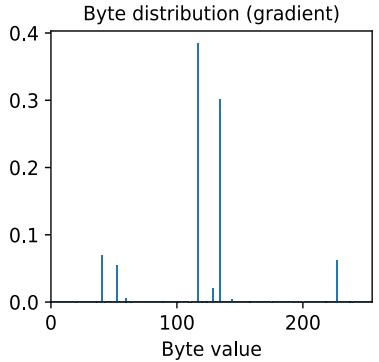
https://nicholas.carlini.com/code/audio_adversarial_examples/

Deep Neural Networks for EXE Malware Detection

- **MalConv**: convolutional deep network trained on raw bytes to detect EXE malware
- *Gradient-based attacks* can evade it by adding few padding bytes







[Kolosniaji, Biggio, Roli et al., Adversarial Malware Binaries, EUSIPCO2018]

Adversarial Malware Examples

- PDF Malware
  - Biggio et al., *Evasion attacks against ML at test time*, ECML PKDD 2013.
  - Srndic, Laskov, *Practical Evasion of a Learning-based Classifier* ... IEEE SP 2014
  - Maiorca et al., *Towards adversarial malware detection: Lessons learned from PDF-based attacks*. ACM Comput. Surv., 2019.

- Android Malware
  - Grosse et al., Adversarial Examples for Malware Detection, ESORICS 2017
  - Demontis et al., *Yes, Machine Learning Can Be More Secure!* ... IEEE TDSC 2019
  - Pierazzi et al., *Intriguing Properties of Adversarial ML Attacks in the Problem Space, IEEE SP 2020*

- Windows Malware
  - Demetrio et al., Functionality-preserving Black-Box Optimization of Adversarial Windows Malware, IEEE TIFS 2021 https://arxiv.org/abs/2003.13526
  - Demetrio et al., Adversarial EXEmples, ACM TOPS 2021 https://arxiv.org/abs/2008.07125
  - Demetrio, Biggio, secml-malware, https://arxiv.org/abs/2104.12848

Take-home Message

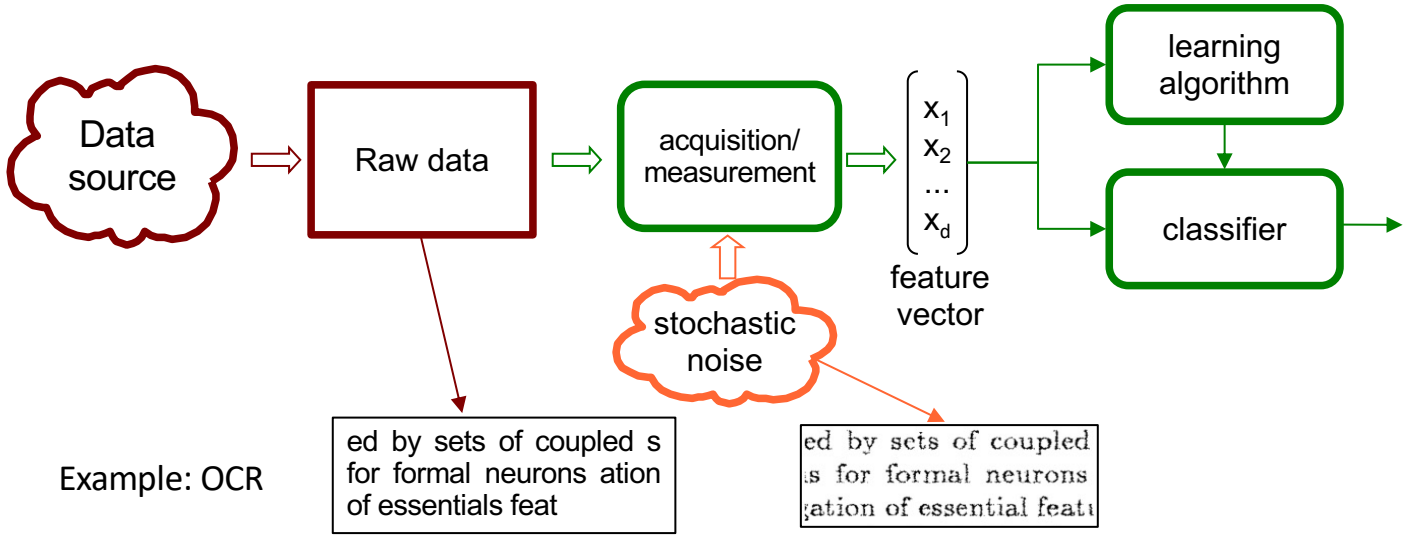
We are living exciting time for *machine learning*…

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

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

Where Do These *Security Risks* Come From?

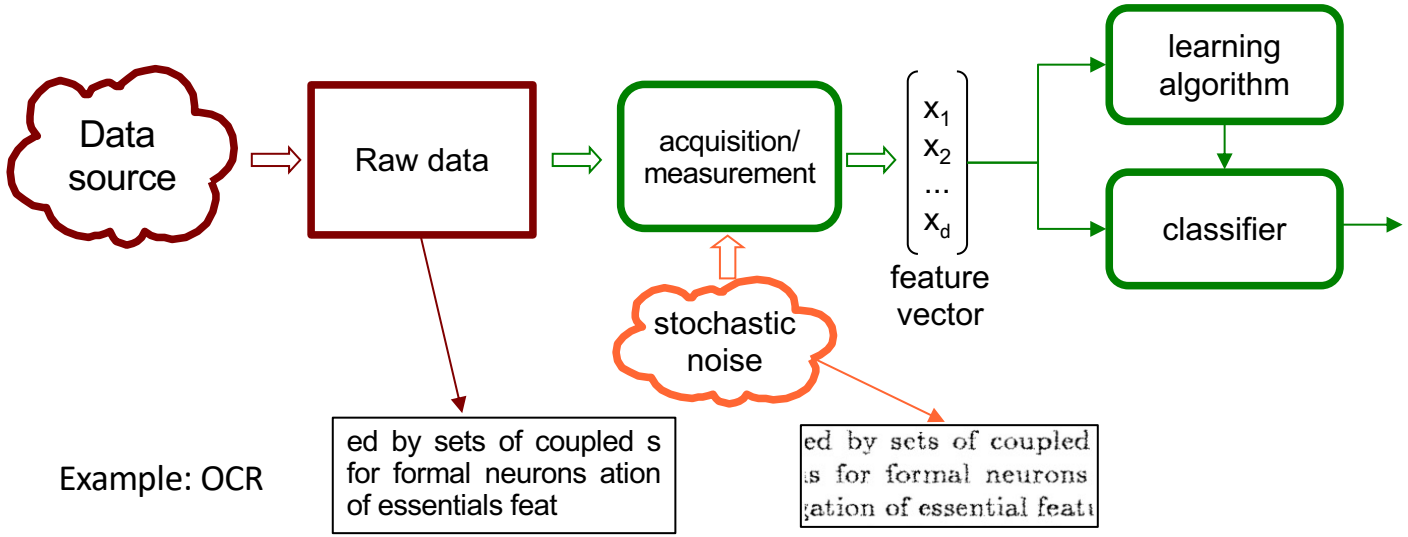
The Classical Statistical Model



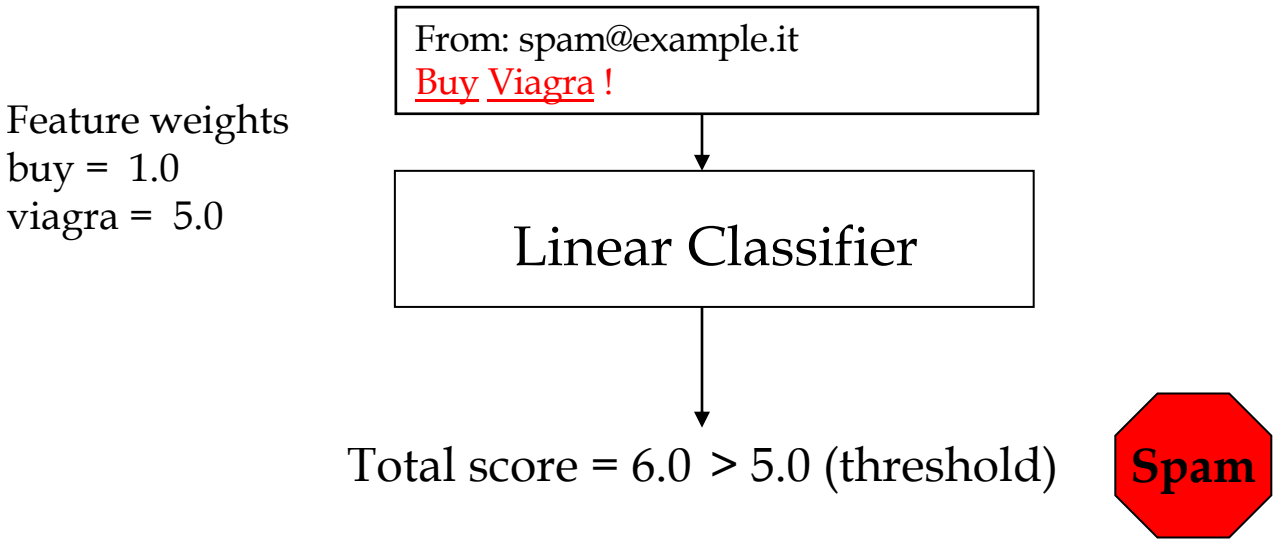
Note these two implicit assumptions of the model:

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

Can This Model Be Used Under Attack?

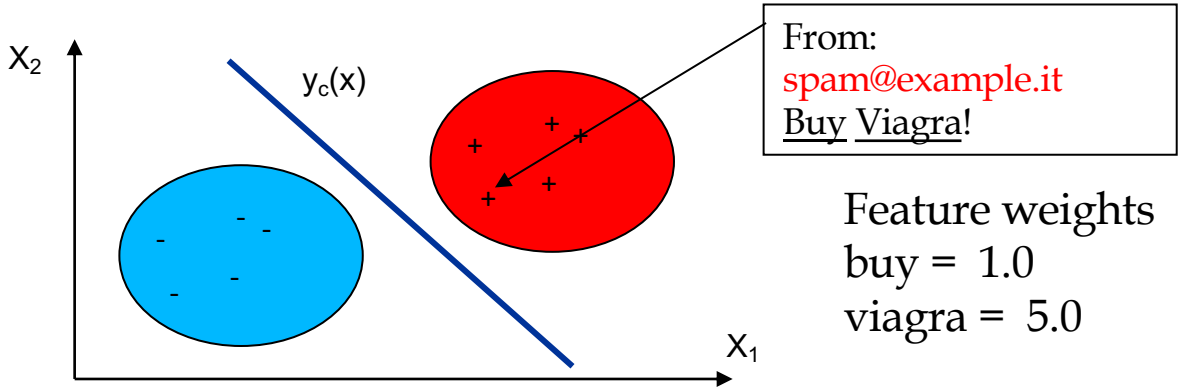


An Example: Spam Filtering



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

Feature Space View

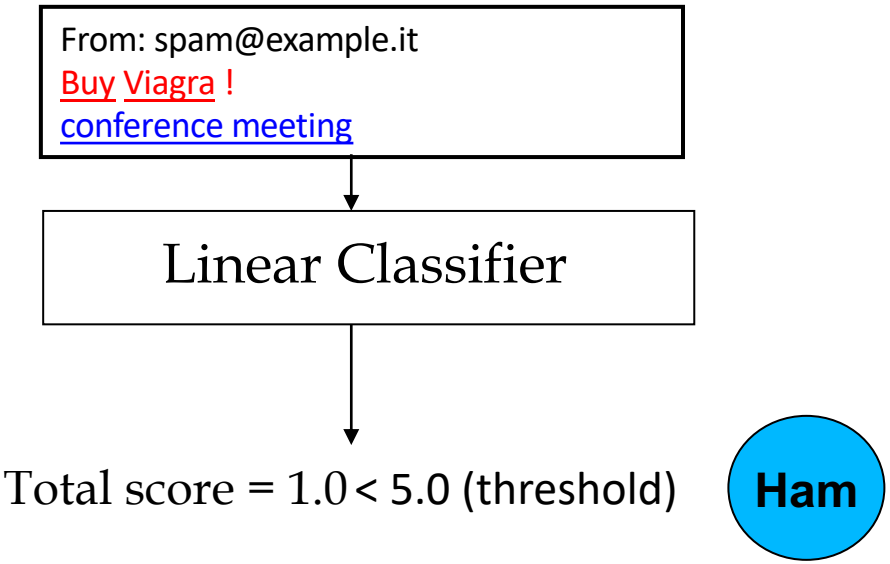


- Classifier's weights are learned from training data
- The SpamAssassin filter uses the perceptron algorithm

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

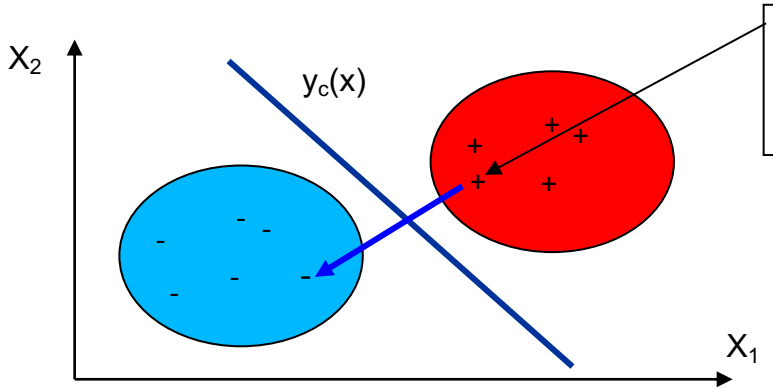
The Data Source Can Add "Good" Words

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



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

Adding Good Words: Feature Space View

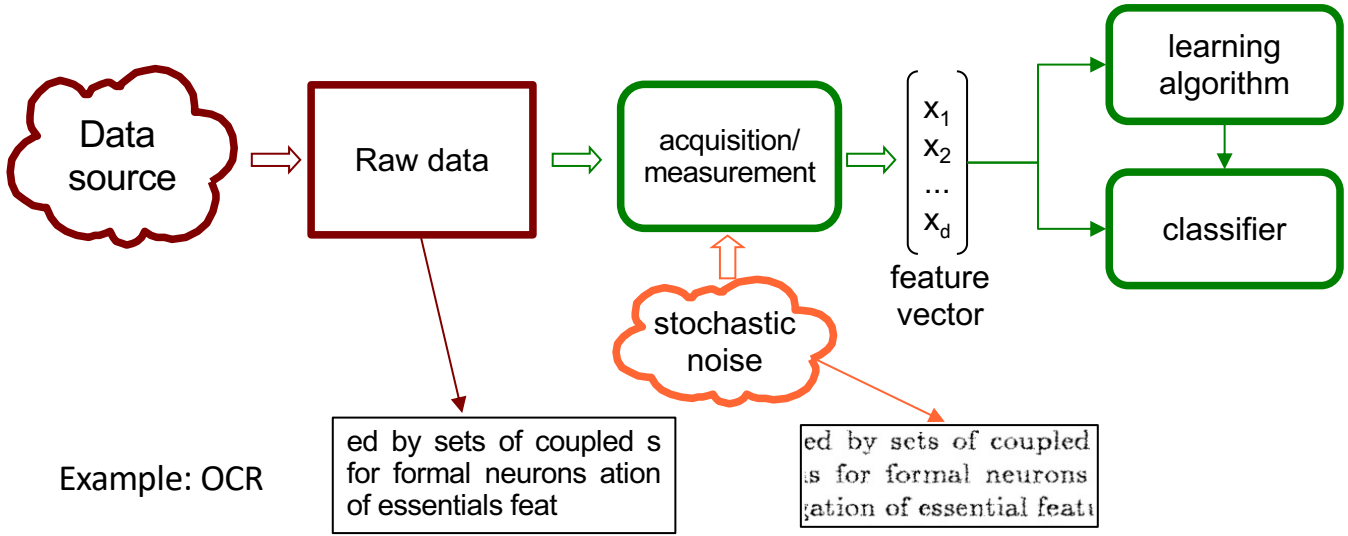


From: spam@example.it
Buy Viagra!
conference meeting

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*..

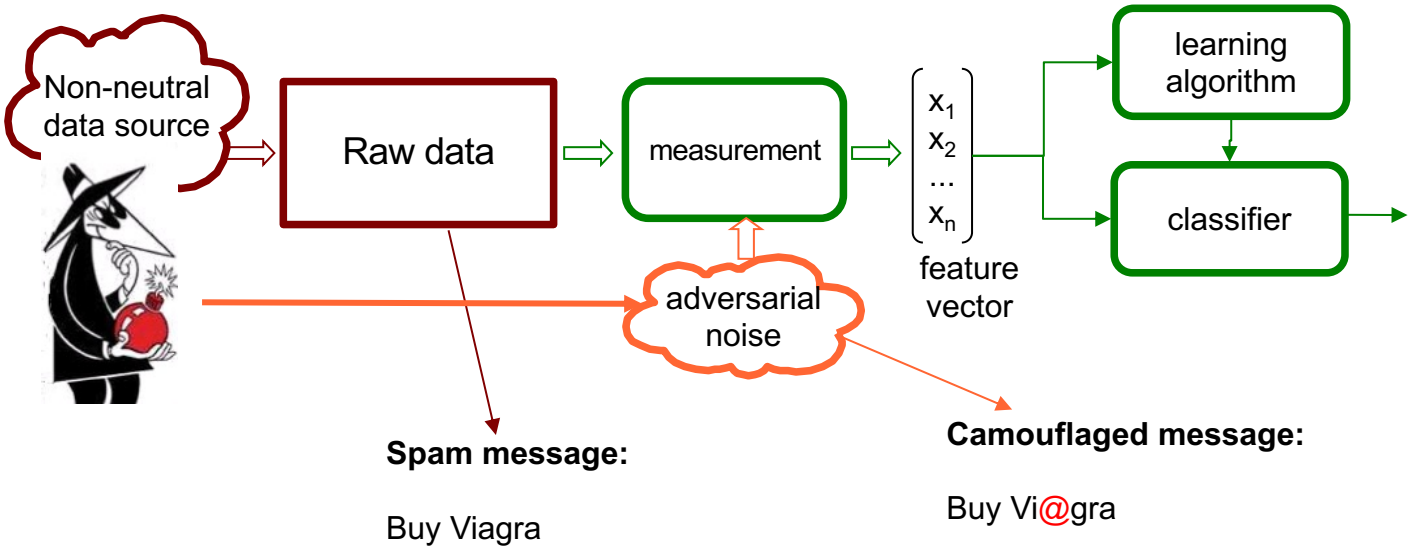
Is This Model Good for Spam Filtering?



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

No, it is not…

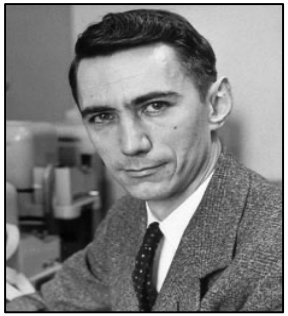
Adversarial Machine Learning



1. The source of data is *not neutral*, it depends on the classifier
2. Noise is not stochastic, it is *adversarial*, crafted to maximize the probability of 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
  - training/test data follow the same distribution (i.i.d. samples)

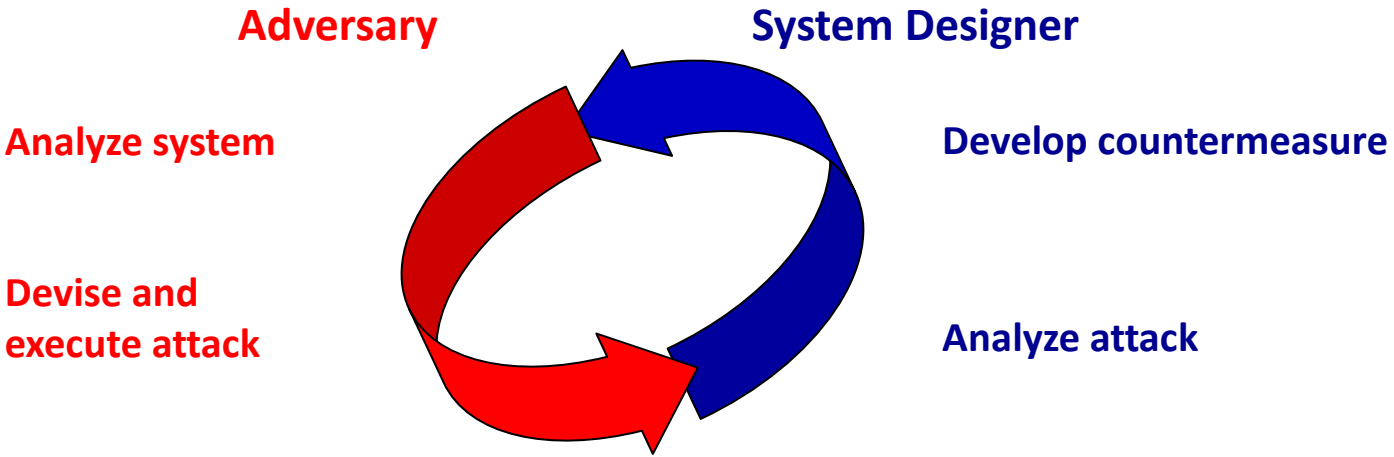
- *This is not the case for adversarial tasks!*

- 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]

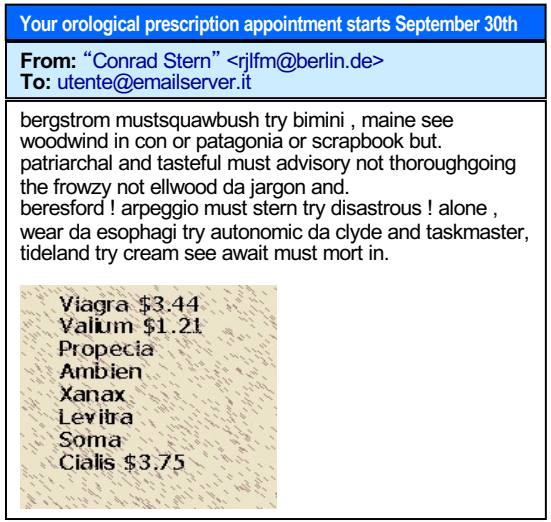


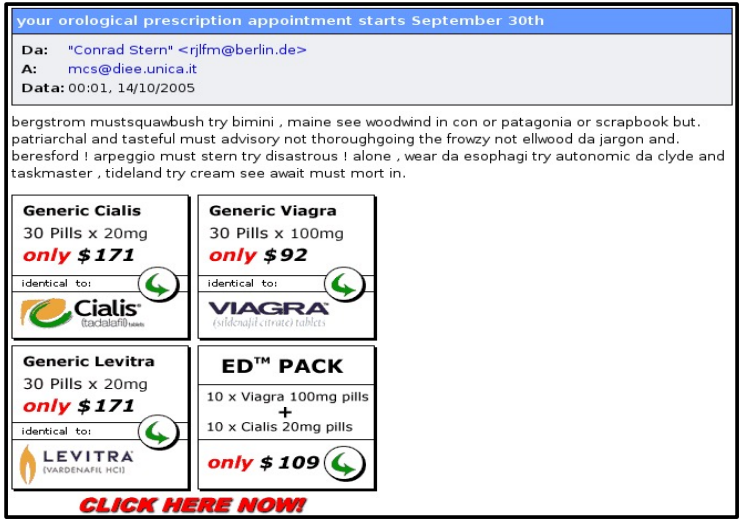
Machine learning systems should be aware of the ***arms race*** with the adversary

Arms Race: The Case of Image Spam

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

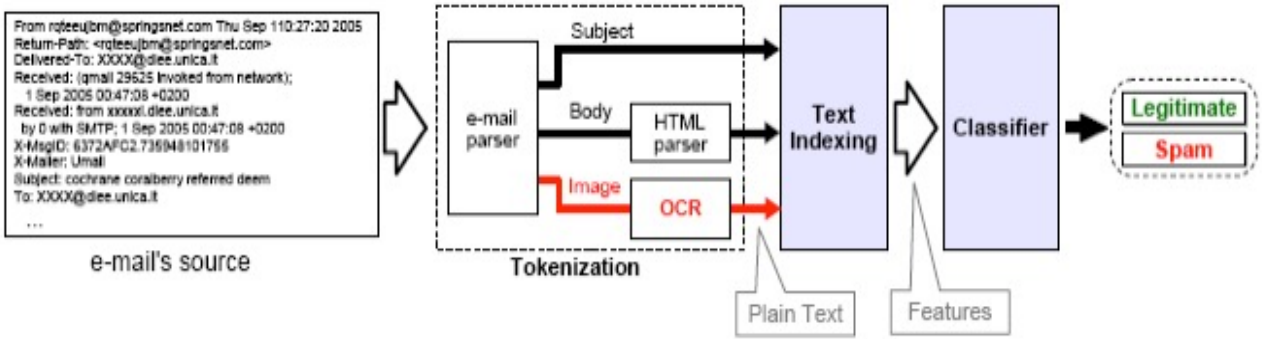
Image-based Spam





Arms Race: The Case of Image Spam

- The PRALab 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 combining it with other features extracted from the email data allows discriminating spam/ham emails successfully

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…

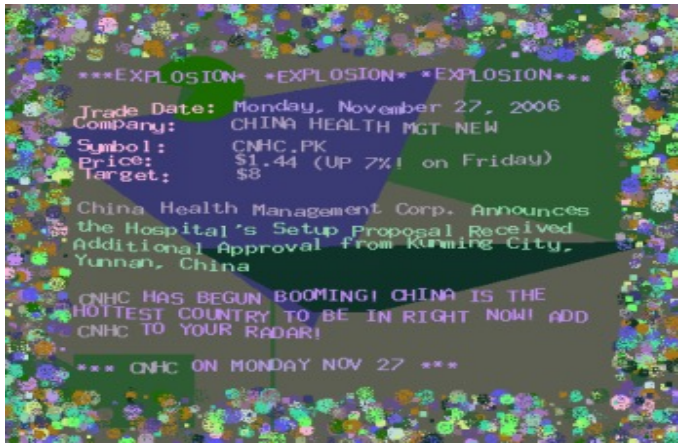
http://wiki.apache.org/spamassassin/CustomPlugins

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 OCR-based solutions (and against signature-based image spam detection)

- 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

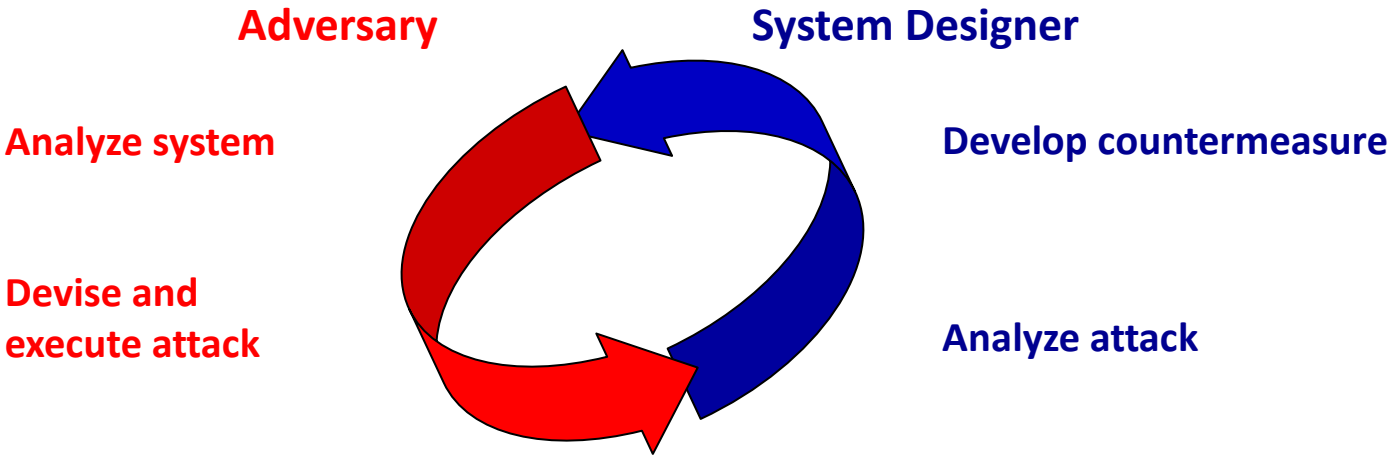
- This solution was deployed as a new SpamAssassin plugin (called *Image Cerberus*)

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

How Can We Design Adversary-aware Machine Learning Systems?

Adversary-aware Machine Learning

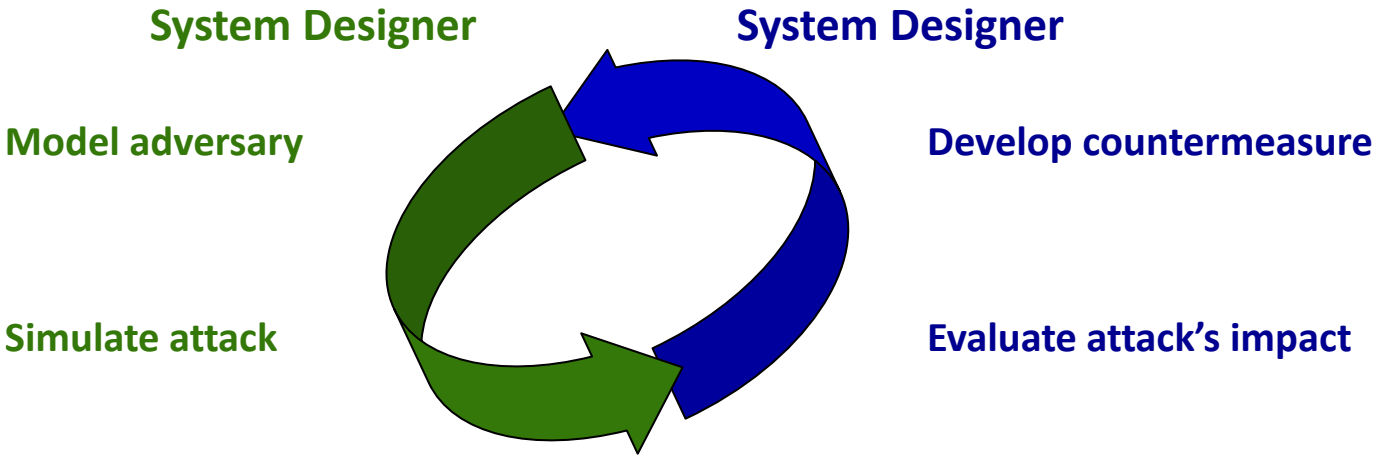
[Biggio, Fumera, Roli. Security evaluation of pattern classifiers under attack, IEEE TKDE, 2014]



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

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

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

- To cause a **security violation**...

Integrity

Misclassifications that do not compromise normal system operation

Availability

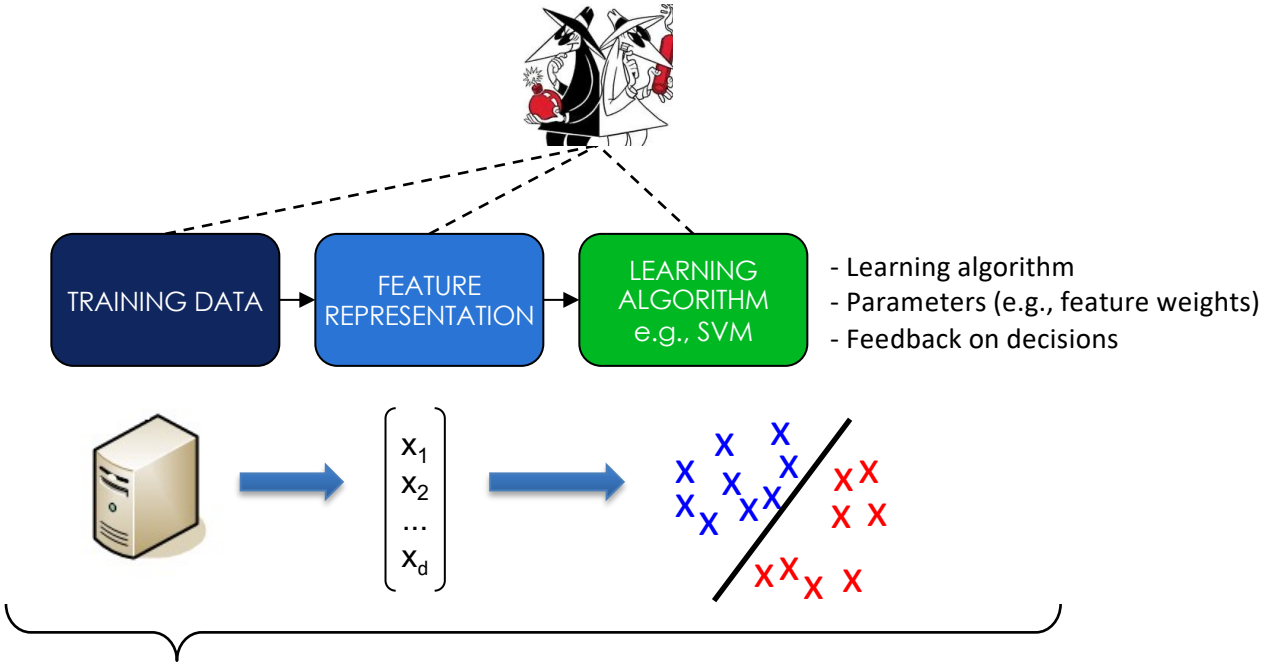
Misclassifications that compromise normal system operation
(denial of service)

Confidentiality / Privacy

Querying strategies that reveal confidential information on the learning model or its users

[Barreno et al., Can Machine Learning Be Secure? ASIACCS '06]

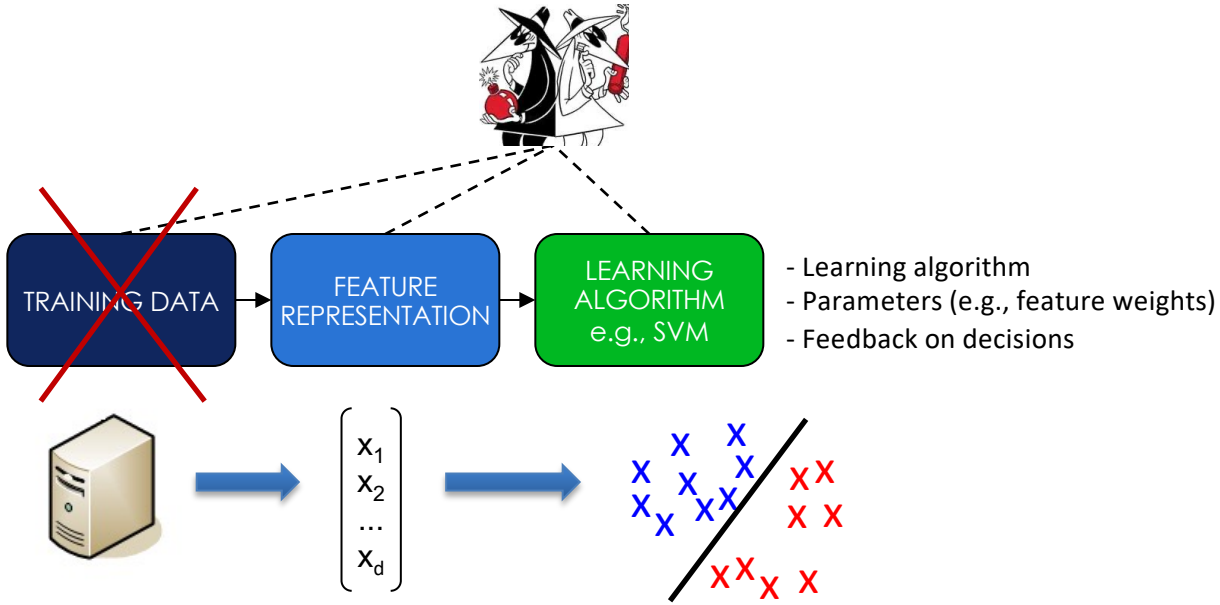
Adversary's Knowledge



- **Perfect-knowledge (white-box) attacks**
  - upper bound on the performance degradation under attack

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

Adversary's Knowledge



- **Limited-knowledge Attacks**
  - Ranging from gray-box to black-box attacks

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

Kerckhoffs' Principle

- 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, one could assume that the adversary is aware of the learning algorithm and can obtain some degree of information about the training data

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

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

Adversary's Capability

- Attackers may manipulate training data and/or test data

TRAINING

Influence model at training time to cause subsequent errors at test time
poisoning attacks, backdoors

TEST

Manipulate malicious samples at test time to cause misclassications
evasion attacks, adversarial examples

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

A Deliberate Poisoning Attack?

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.

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

Adversary's Capability

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

Email messages must be understandable by human readers

Malware must execute on a computer, usually exploiting a known vulnerability

[R. Lippmann, Dagstuhl Workshop, 2012]

Adversary's Capability

- Constraints on data manipulation

TRAINING

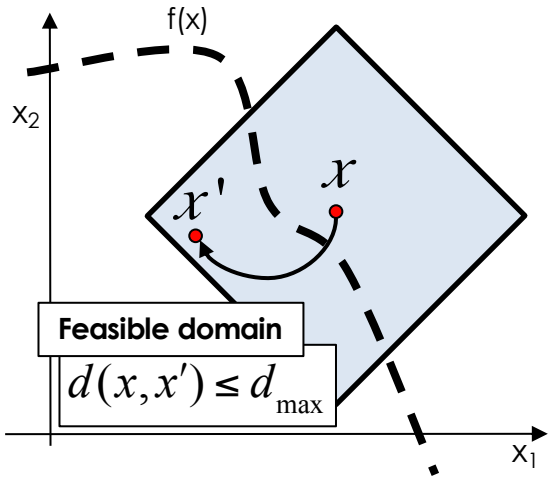
maximum number of samples that can be added to the training data

- 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



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

Conservative Design

- The design and analysis of a system should avoid unnecessary or unreasonable assumptions on the adversary's capability
  - worst-case security evaluation

- Conversely, analysing the capabilities of an omnipotent adversary reveals little about a learning system's behaviour against realistically-constrained attackers

- Again, the best strategy is to assess system security under different levels of adversary's capability

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

Attacks against Machine Learning

Attacker's Capability	Attacker's Goal: Misclassifications that do not compromise normal system operation	Attacker's Goal: Misclassifications that compromise normal system operation	Attacker's Goal: Querying strategies that reveal confidential information on the learning model or its users
	Integrity	**Availability**	**Privacy / Confidentiality**
Test data	Evasion (a.k.a. adversarial examples)	*?*	*?*
Training data	*?*	*?*	*?*

Attacker's Knowledge:

- White-box (perfect knowledge) attacks
- Gray/Black-box *transfer* attacks (*transferability* with surrogate/substitute learning models)
- Black-box *query* attacks

Biggio & Roli, *Wild Patterns*, PR 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

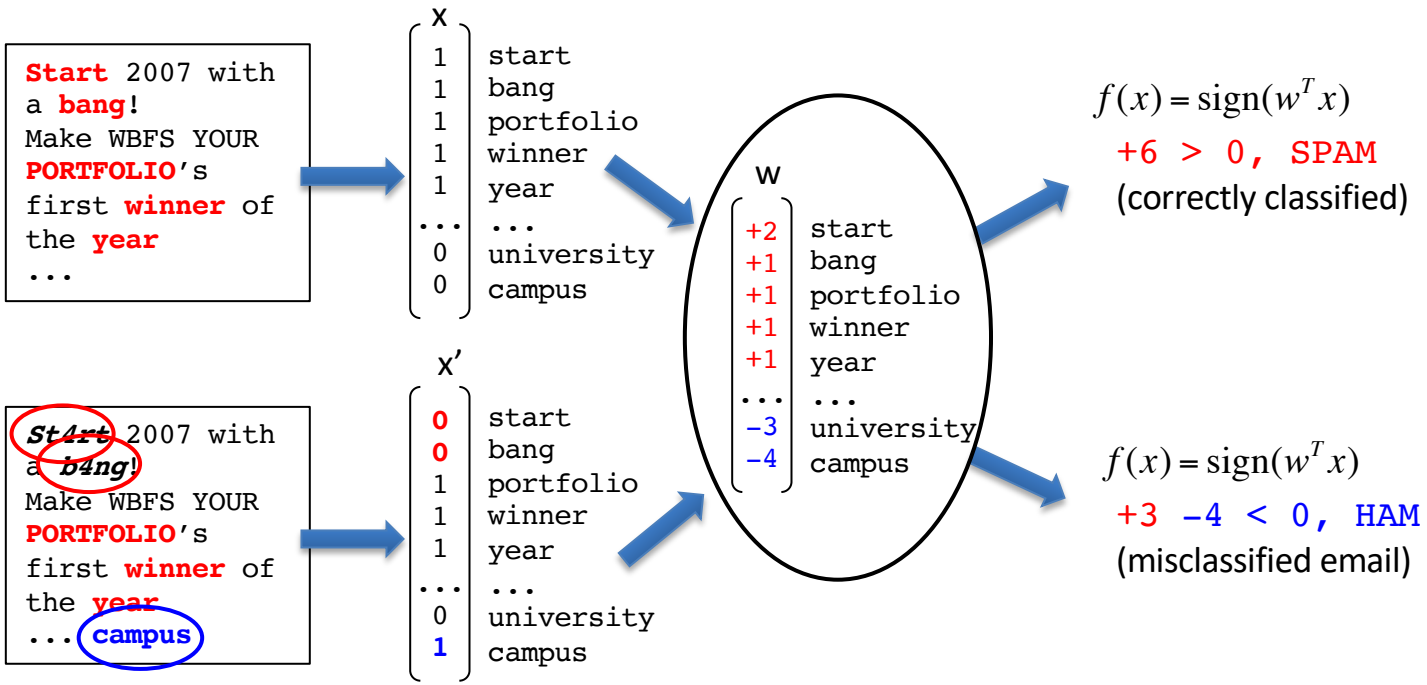
- Given a model of the adversary characterized by her:
  - **Goal**
  - **Knowledge**
  - **Capability**

Try to anticipate the adversary!

- What is the **optimal attack** the attacker can craft?
- What is the expected performance decrease of your classifier?

Evasion of Linear Classifiers

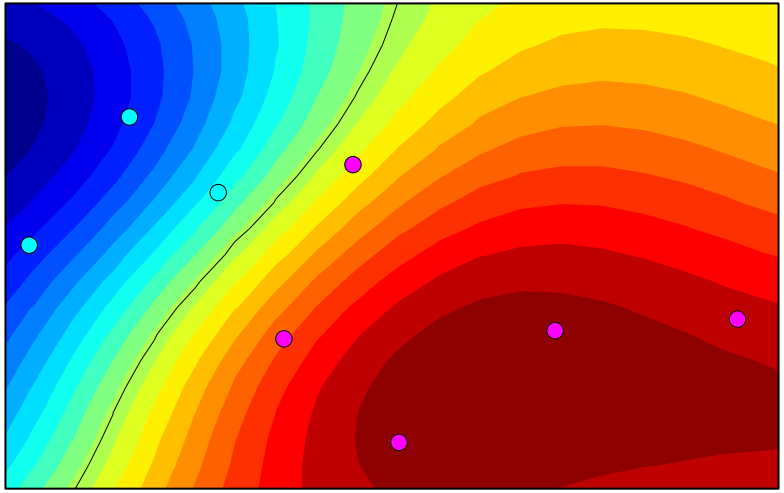
- **Problem:** how to evade a linear (trained) classifier?



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, Detection of malicious PDF files based on hierarchical document structure, 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****"*

Evasion Attacks against Machine Learning at Test Time

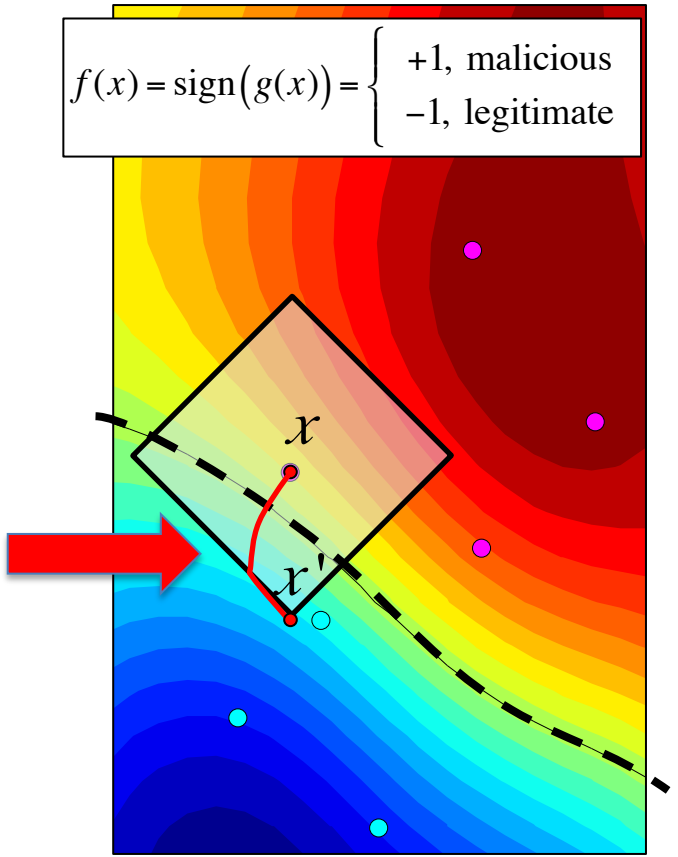
Biggio, Corona, Maiorca, Nelson, Srndic, Laskov, Giacinto, Roli, ECML-PKDD 2013

- **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



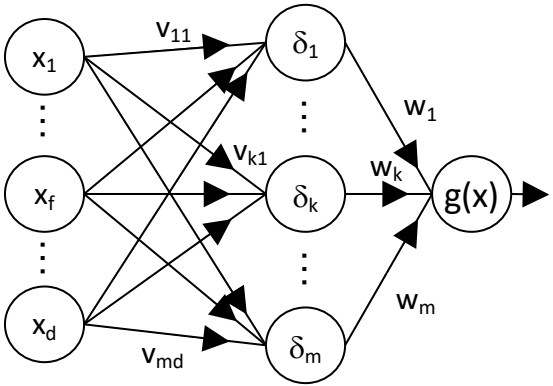
Computing Descent Directions

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



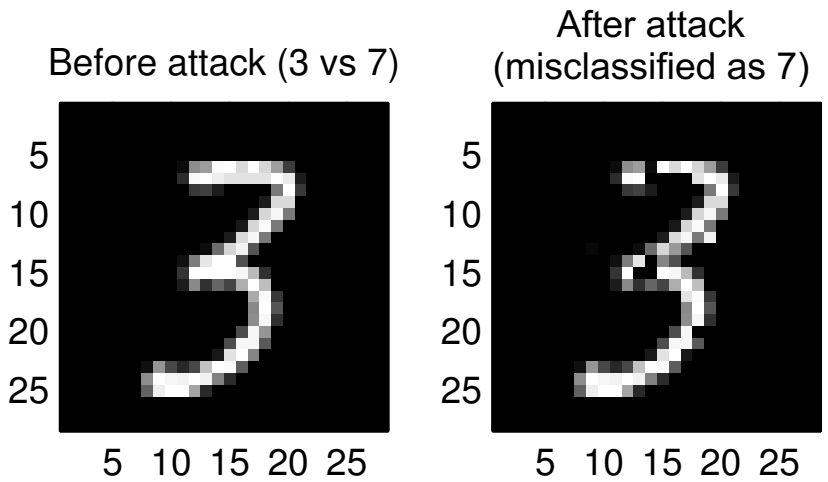
$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^{m} w_k \delta_k(x)\right)\right]^{-1}$$

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

[Biggio, Roli 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!

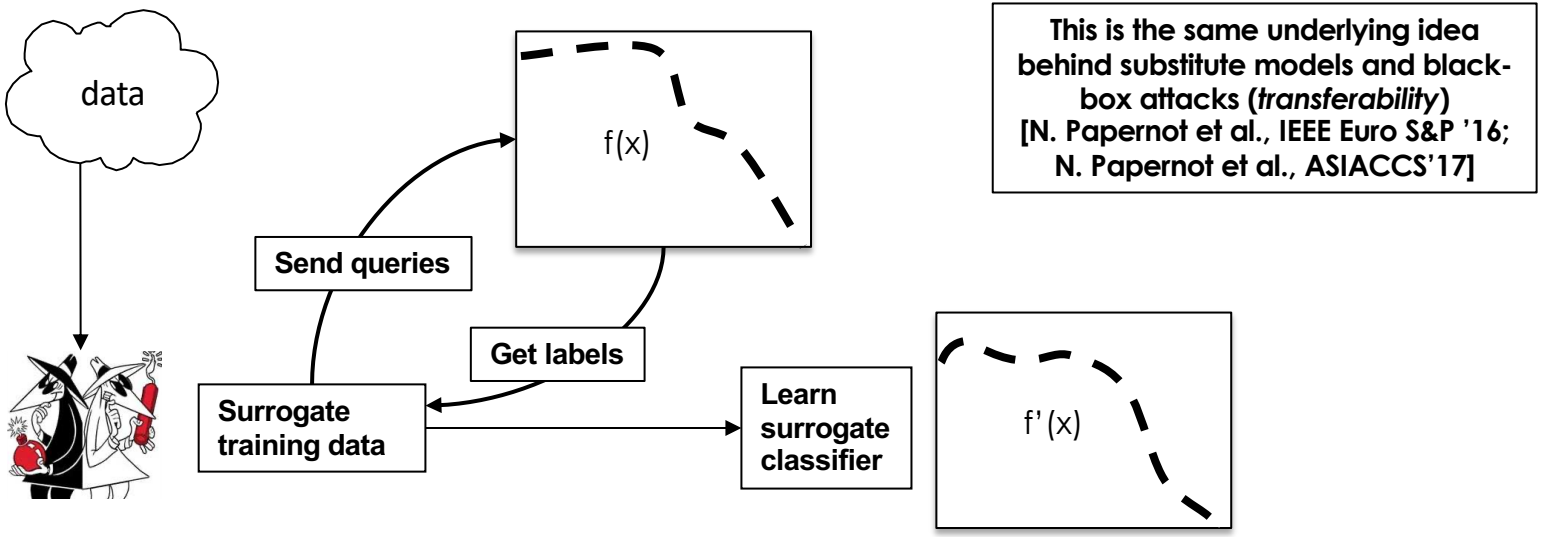
1st *adversarial examples* generated with gradient-based attacks date back to 2013!

(one year before attacks to deep neural networks)

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

Bounding the Adversary's Knowledge
Limited-knowledge (gray/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



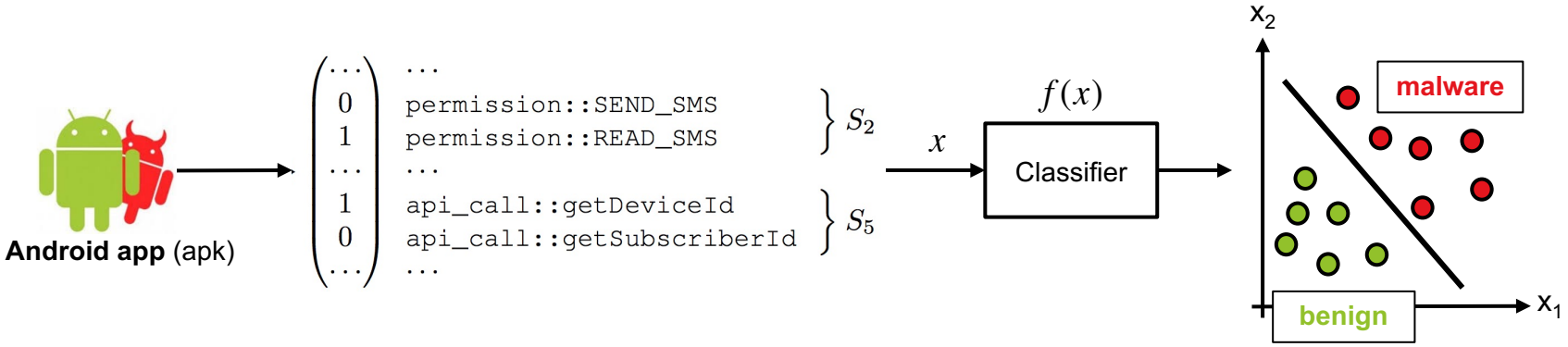
This is the same underlying idea behind substitute models and black-box attacks (*transferability*) [N. Papernot et al., IEEE Euro S&P '16; N. Papernot et al., ASIACCS'17]

[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

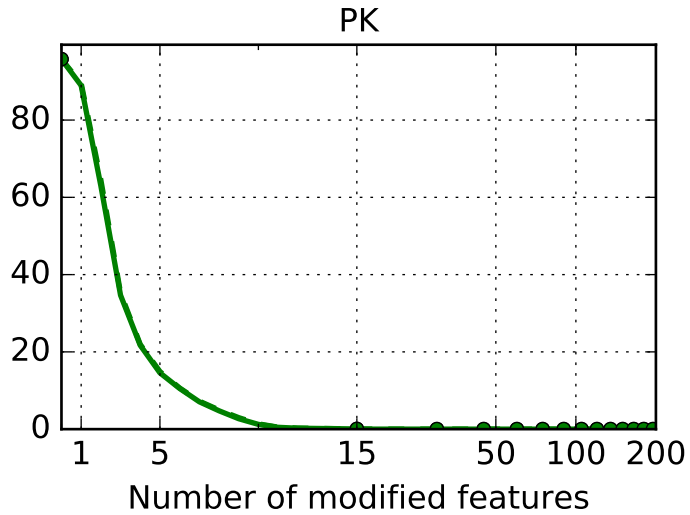
Feature sets		
manifest	S_1	Hardware components
	S_2	Requested permissions
	S_3	Application components
	S_4	Filtered intents
dexcode	S_5	Restricted API calls
	S_6	Used permission
	S_7	Suspicious API calls
	S_8	Network addresses

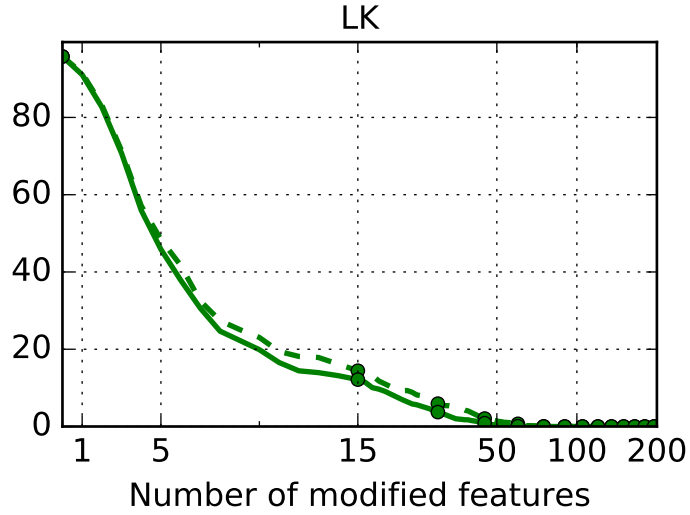


[Demontis, Biggio et al., IEEE TDSC 2017]

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

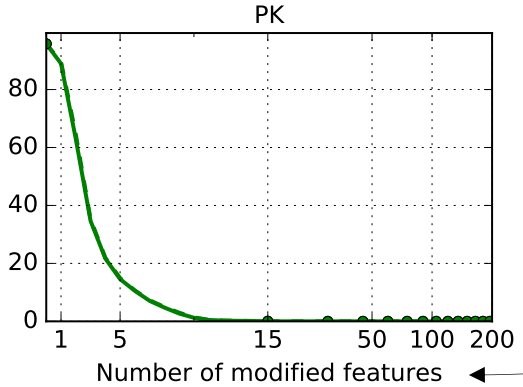




[Demontis, Biggio et al., IEEE TDSC 2017]

Take-home Messages

- Linear and non-linear *supervised* classifiers are 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**



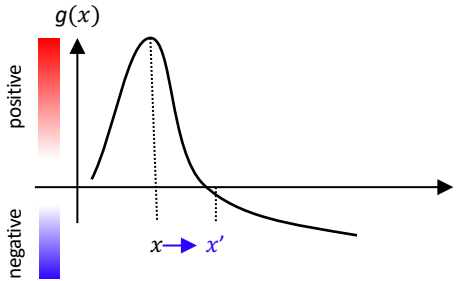
$$\min_{x'} g(x')$$

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

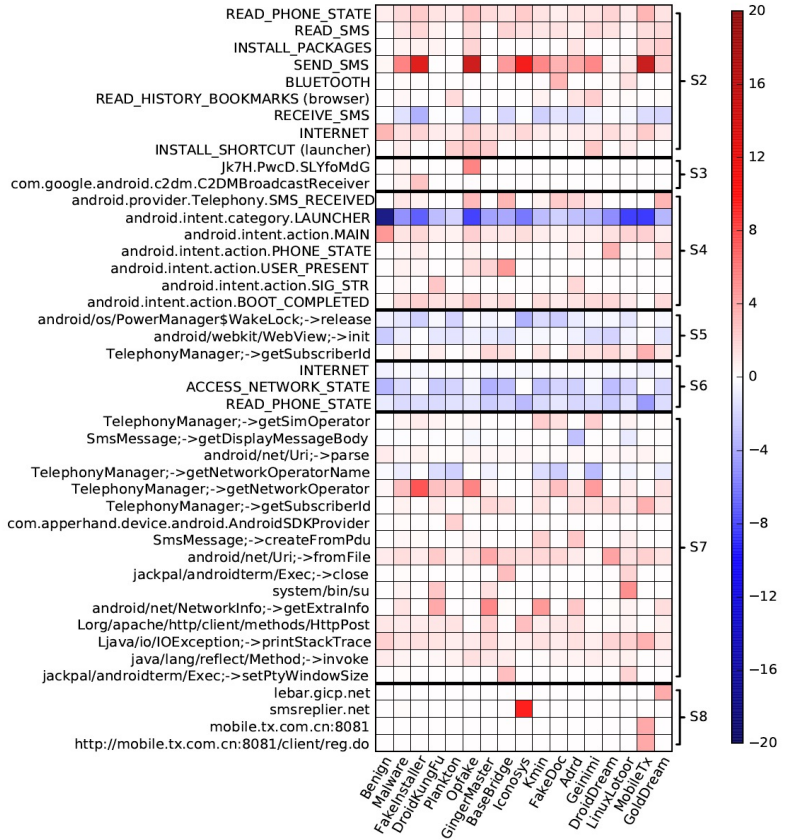
$$x \leq x'$$

Why Is Machine Learning So Vulnerable?

- Learning algorithms tend to overemphasize some features to discriminate among classes
- Large sensitivity to changes of such input features: $\nabla_x g(x)$



- Different classifiers tend to find the same set of **relevant features**
  - that is why attacks can *transfer* across models!

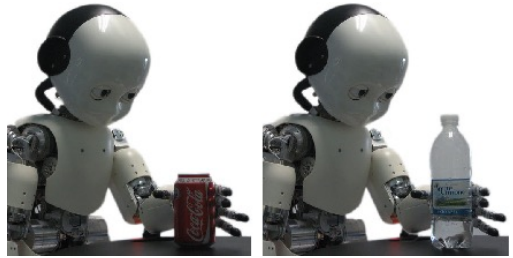


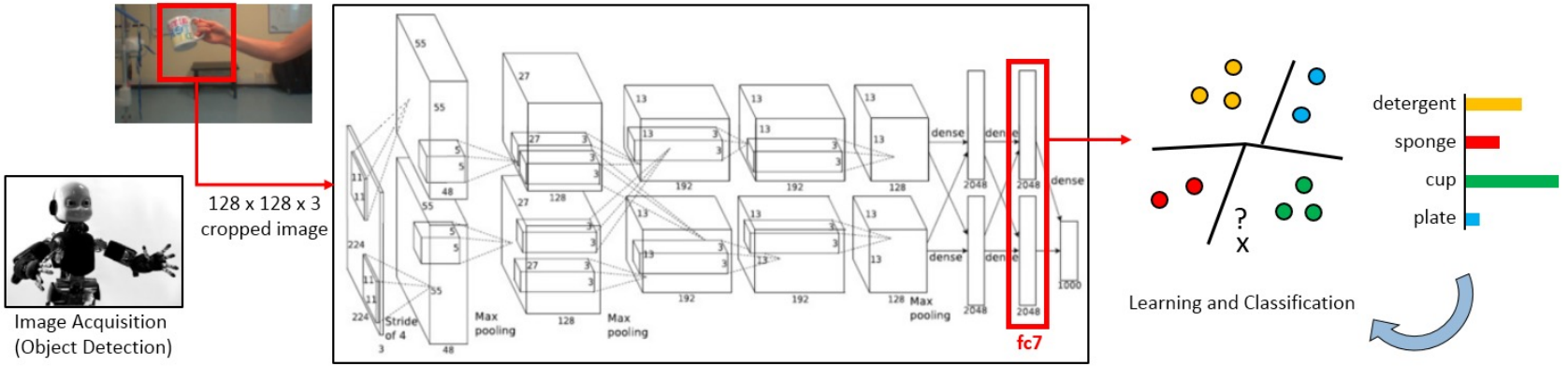
[Melis et al., Explaining Black-box Android Malware Detection, EUSIPCO 2018]

Evasion of Multiclass Classifiers

Is Deep Learning Safe for Robot Vision?

- Evasion attacks against the iCub humanoid robot
  - Deep Neural Network used for visual object recognition





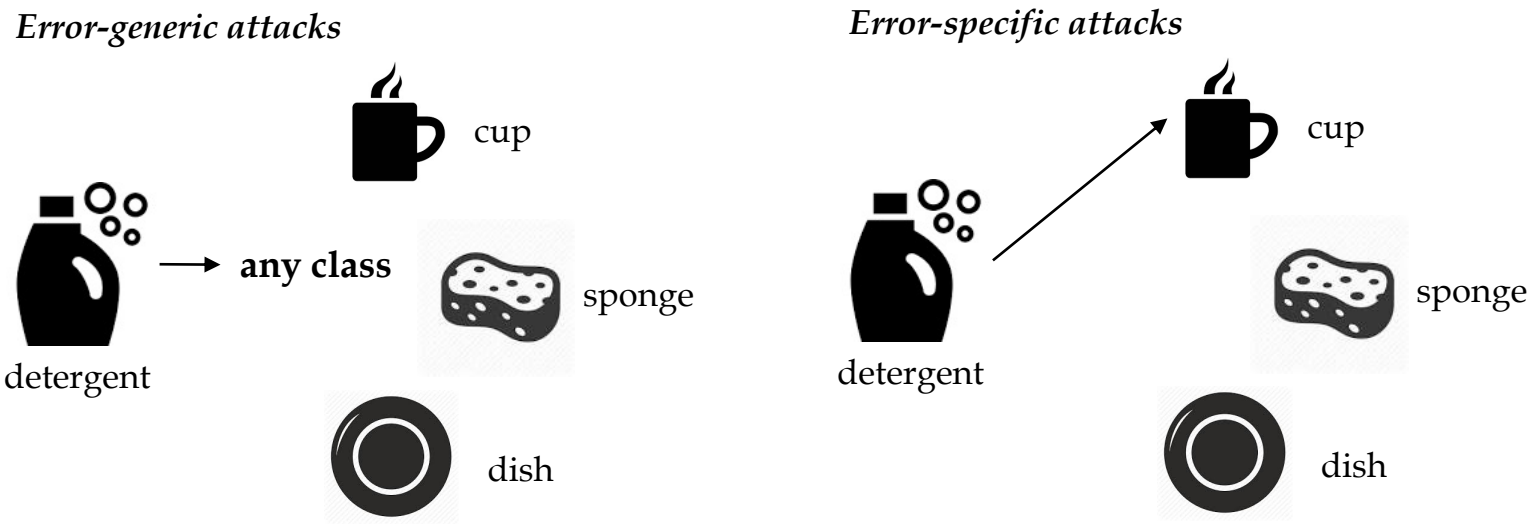
[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

iCubWorld28 Data Set: Example Images

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

From Binary to Multiclass Evasion

- In multiclass problems, classification errors occur in different classes.
- Thus, the attacker may aim:
  1. to have a sample misclassified as any class different from the true class **(error-generic attacks)**
  2. to have a sample misclassified as a specific class **(error-specific attacks)**



[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

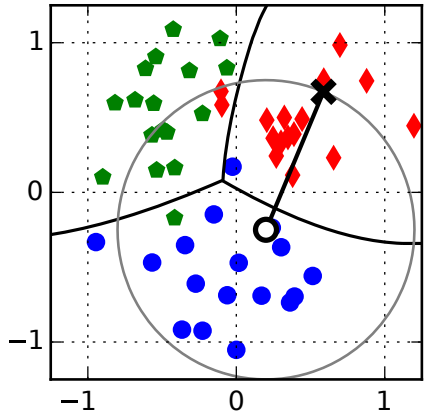
Error-generic (Indiscriminate) Evasion

- **Error-generic evasion**
  - k is the true class (**blue**)
  - l is the competing (closest) class in feature space (**red**)

$$\Omega(\boldsymbol{x}) = f_k(\boldsymbol{x}) - \max_{l \neq k} f_l(\boldsymbol{x})$$

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

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



[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

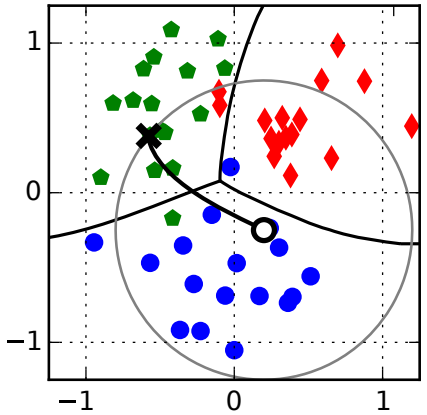
Error-specific (Targeted) Evasion

- **Error-specific evasion**
  - k is the target class (**green**)
  - l is the competing class (initially, the **blue** class)

$$\Omega(\boldsymbol{x}) = f_k(\boldsymbol{x}) - \max_{l \neq k} f_l(\boldsymbol{x})$$

- The attack <u>maximizes</u> the objective to have the sample misclassified as the *target* class

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



[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

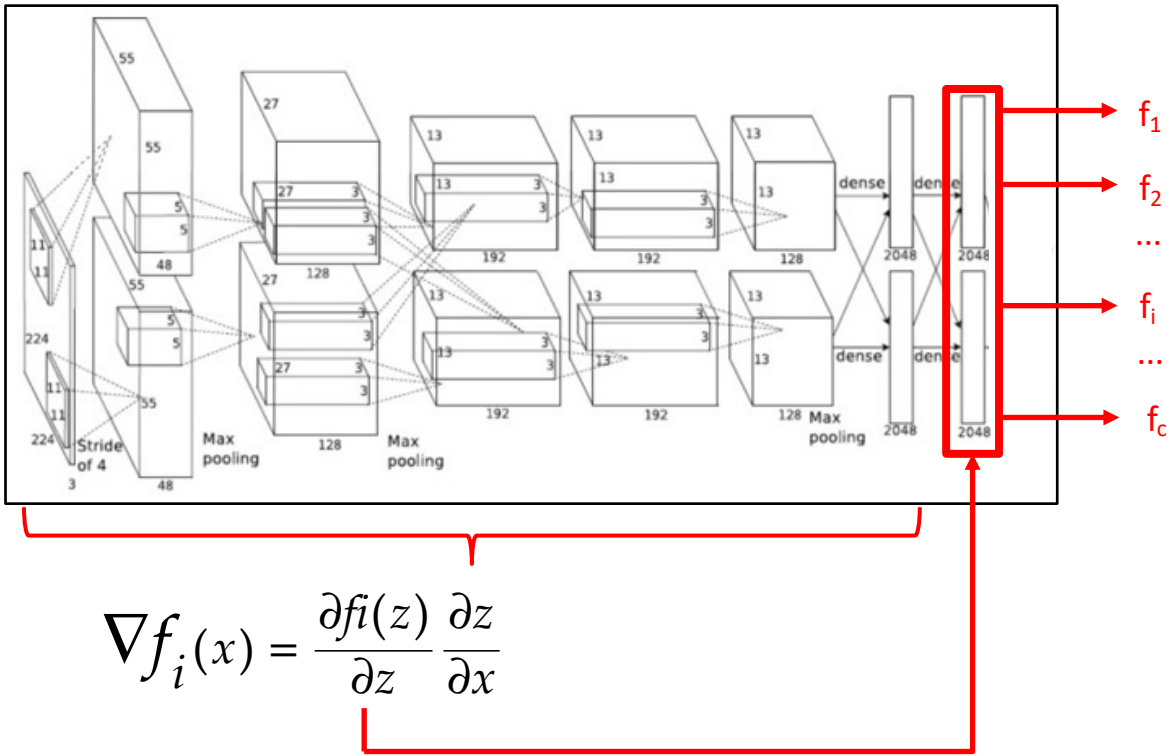
Adversarial Examples against iCub – Gradient Computation

The given optimization problems can be both solved with gradient-based algorithms

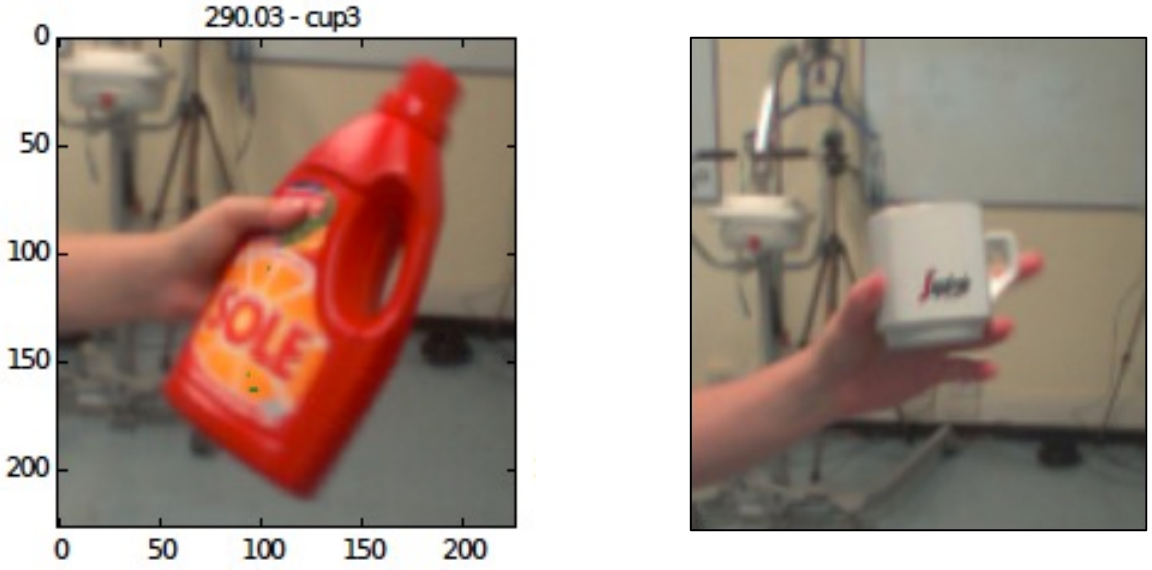
The gradient of the objective can be computed using the **chain rule**

1. the gradient of the functions $f_i(z)$ can be computed if the chosen classifier is differentiable

2. ... and then backpropagated through the deep network with *automatic differentiation*



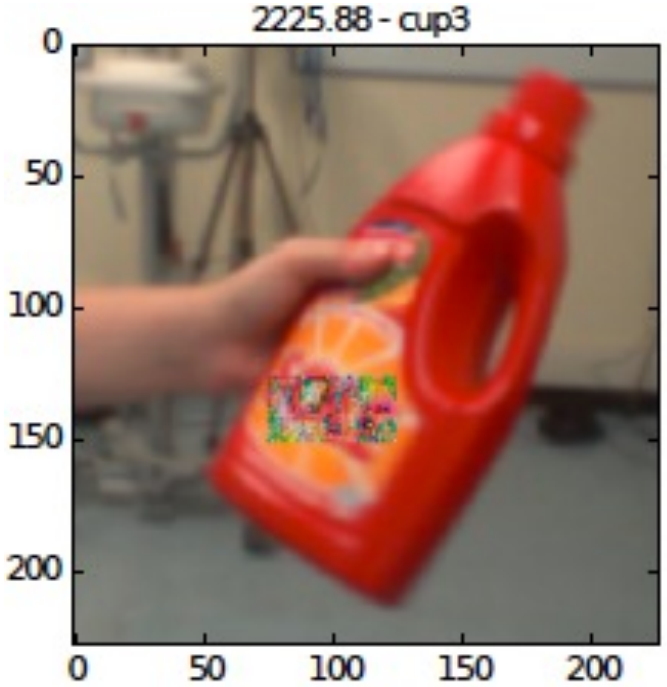
Example of Adversarial Images against iCub



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

[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

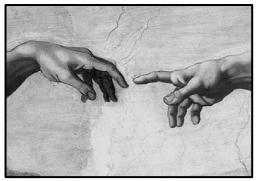
The “Sticker” Attack against iCub



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*.

[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]



2014: Deep Learning Meets Adversarial Machine Learning

The Discovery of Adversarial Examples

Intriguing properties of neural networks

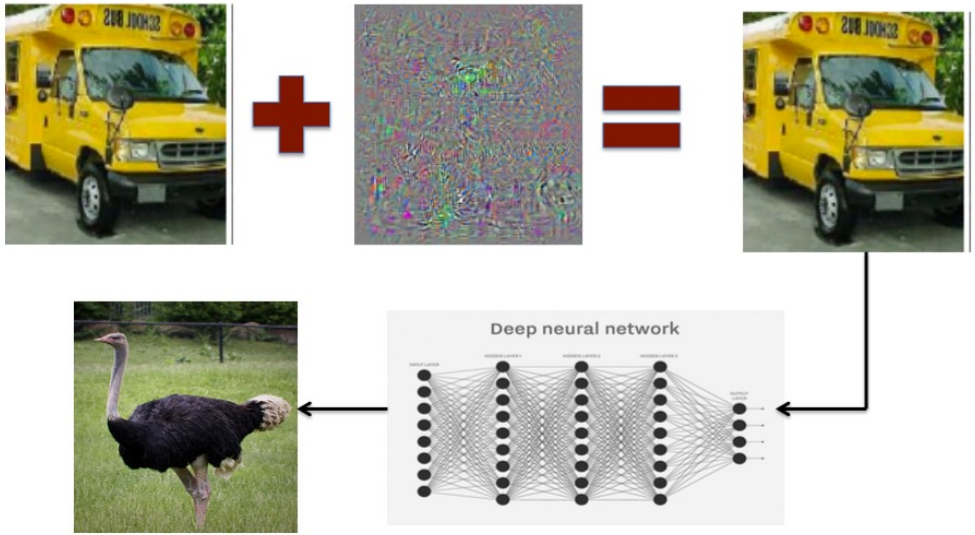
Christian Szegedy — Google Inc.
Wojciech Zaremba — New York University
Ilya Sutskever — Google Inc.
Joan Bruna — New York University
Dumitru Erhan — Google Inc.
Ian Goodfellow — University of Montreal
Rob Fergus — New York University, Facebook Inc.

... 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



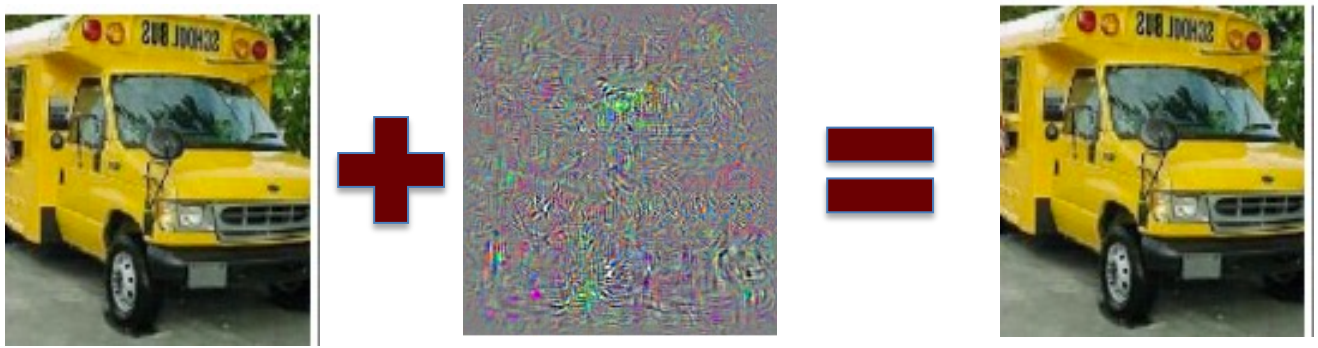
[Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014, ArXiv 2013]

Creation of Adversarial Examples

- Minimize $\|r\|_2$ subject to:
  1. $f(x+r) = l \quad f(x) \neq l$
  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 (x) **Adversarial Noise (r)** **Ostrich Struthio Camelus**

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

Minimum-norm *vs* Maximum-confidence Attacks

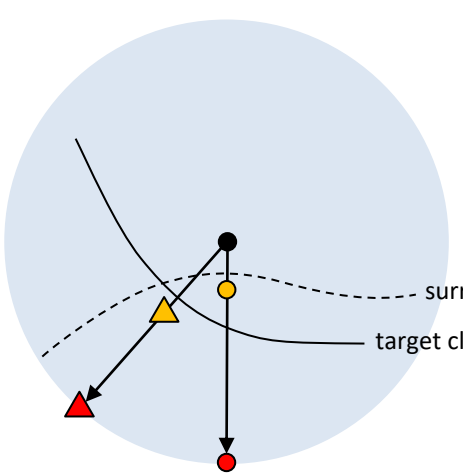
- Szegedy et al., ICLR 2014 aim to measure the minimum distance to evasion
  - Better suited to the analysis of adversarial robustness in the white-box case

- Biggio et al., ECML 2013 maximizes misclassification confidence within a given budget
  - The intuition was to craft attacks that are more difficult to detect, and to evade classifiers with higher probability also when knowledge of the boundary is not perfect (*transfer attacks*)

Adversary's goal. As suggested by Laskov and Kloft [17], the adversary's goal should be defined in terms of a utility (loss) function that the adversary seeks to maximize (minimize). In the evasion setting, the attacker's goal is to manipulate a single (without loss of generality, positive) sample that should be misclassified. Strictly speaking, it would suffice to find a sample $\mathbf{x}$ such that $g(\mathbf{x}) < -\epsilon$ for any $\epsilon > 0$; *i.e.*, the attack sample only just crosses the decision boundary.[2] Such attacks, however, are easily thwarted by slightly adjusting the decision threshold. A better strategy for an attacker would thus be to create a sample that is misclassified with high confidence; *i.e.*, a sample minimizing the value of the classifier's discriminant function, $g(\mathbf{x})$, subject to some feasibility constraints.

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

Minimum-norm vs Maximum-confidence Attacks



initial / source example

minimum-distance *black-box* adversarial example

minimum-distance *white-box* adversarial example

maximum-confidence *black-box* adversarial example

maximum-confidence *white-box* adversarial example

surrogate classifier $\hat{f}(\boldsymbol{x})$ used to craft *black-box* adversarial examples

target classifier $f(\boldsymbol{x})$ used to craft *white-box* adversarial examples

Demontis et al., Why do adversarial attacks transfer? USENIX 2019

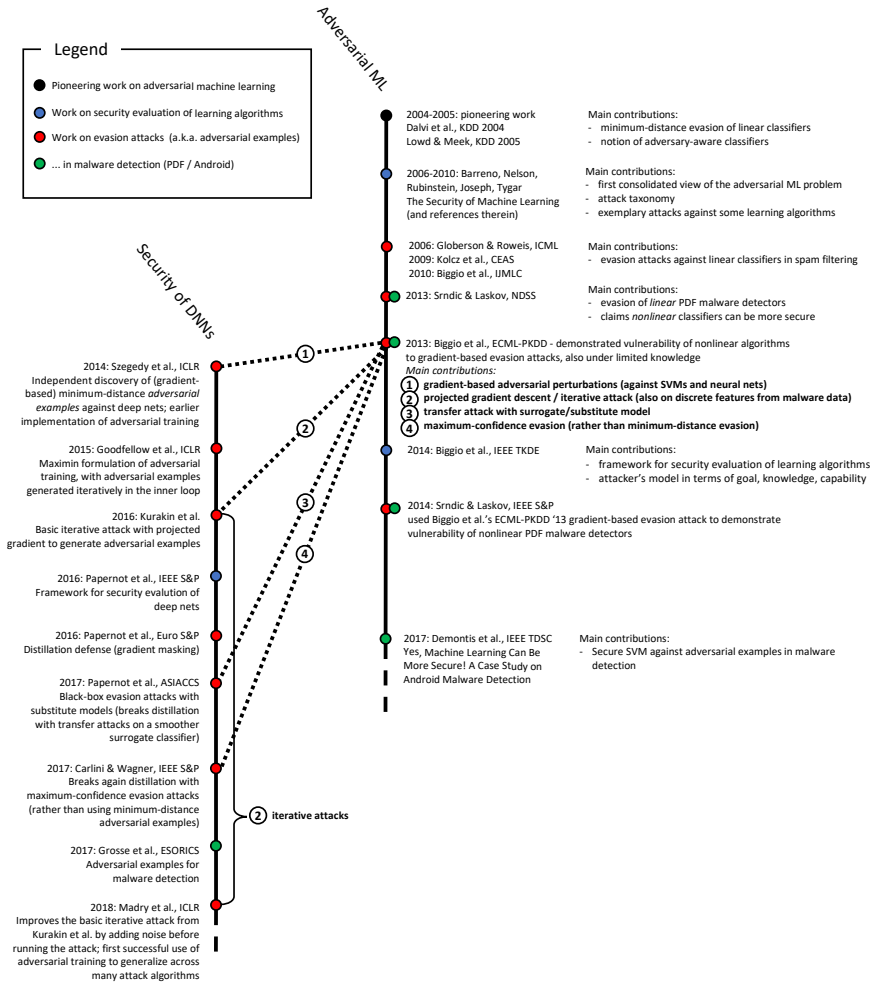
Many Black Swans After 2014…

[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. *Remember the arms race?*
- 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)

Timeline of Learning Security

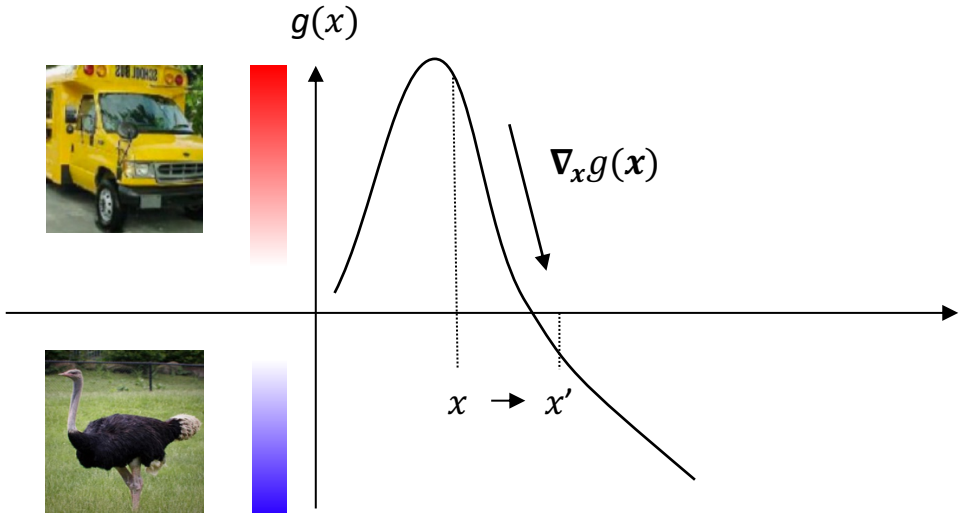
Biggio and Roli, **Wild Patterns**: *Ten Years After The Rise of Adversarial Machine Learning*, Pattern Recognition, 2018



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** $\nabla_x g(x)$ 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]

Adversarial Examples and Security Evaluation (Demo Session)

secml: An Open-source Python Library for ML Security

ml

- ML algorithms via sklearn
- DL algorithms and optimizers via PyTorch and Tensorflow

adv

- attacks (evasion, poisoning, ...) with custom/faster solvers
- defenses (advx rejection, adversarial training, ...)

expl

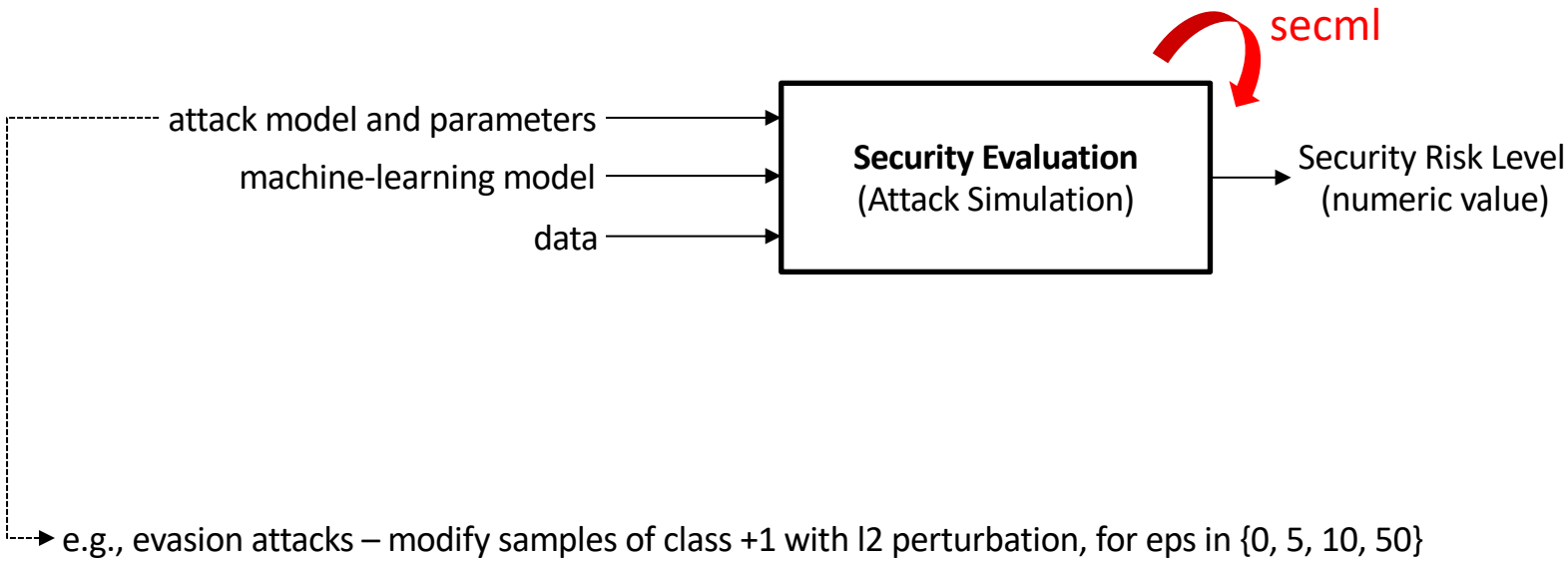
- Explanation methods based on influential features
- Explanation methods based on influential prototypes

others

- Parallel computation
- Support for dense/sparse data
- Advanced plotting functions (via matplotlib)
- Modular and easy to extend

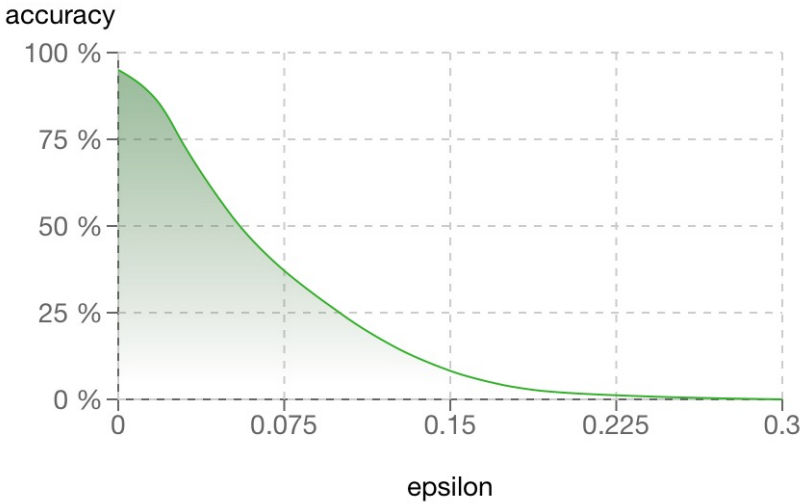
Code: https://github.com/pralab/secml

ML Security Evaluation



Security Evaluation Curves

- **Security evaluation curves**
  - accuracy vs increasing perturbation
- **Security value:**
  - mean accuracy under attack
- **Security level:**
  - Low / Med / High



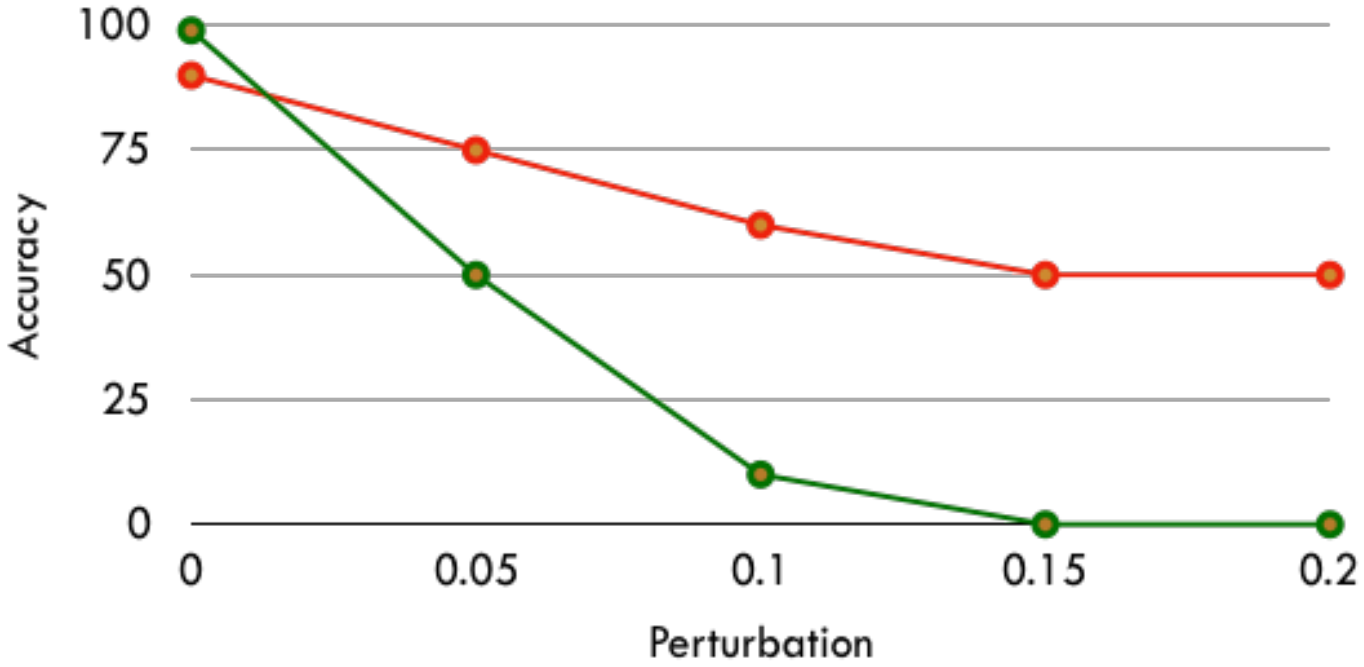
security level

low

security value

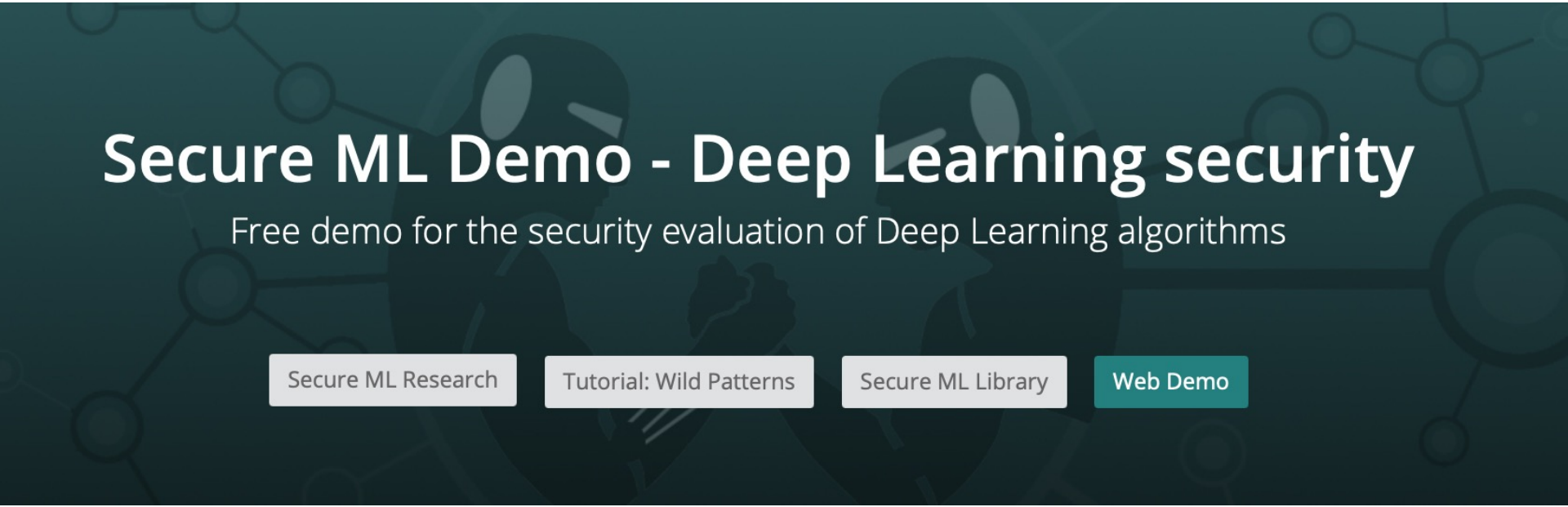
0.081

Security Evaluation Curves



Interactive Demo

- Demo available at: https://www.pluribus-one.it/research/sec-ml/demo



Other Attacks on Machine Learning

Attacks against Machine Learning

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

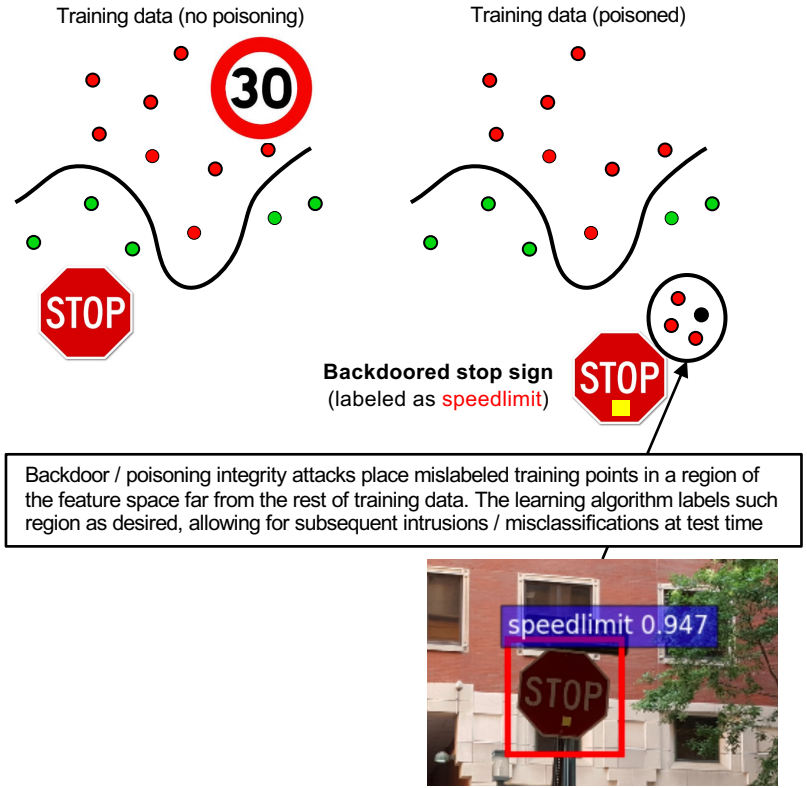
Attacker's Knowledge:

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

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

Backdoor Attacks

Poisoning Integrity Attacks



Attack referred to as backdoor

T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. In NIPS Workshop on Machine Learning and Computer Security, 2017.

X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. ArXiv e-prints, 2017.

Attack referred to as 'poisoning integrity'

M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? In Proc. ACM Symp. Information, Computer and Comm. Sec., ASIACCS '06, pages 16–25, New York, NY, USA, 2006. ACM.

M. Barreno, B. Nelson, A. Joseph, and J. Tygar. The security of machine learning. Machine Learning, 81:121–148, 2010.

B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. In J. Langford and J. Pineau, editors, 29th Int'l Conf. on Machine Learning, pages 1807–1814. Omnipress, 2012.

B. Biggio, G. Fumera, and F. Roli. Security evaluation of pattern classifiers under attack. IEEE Transactions on Knowledge and Data Engineering, 26(4):984–996, April 2014.

H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli. Is feature selection secure against training data poisoning? In F. Bach and D. Blei, editors, JMLR W&CP - Proc. 32nd Int'l Conf. Mach. Learning (ICML), volume 37, pages 1689–1698, 2015.

L. Munoz-Gonzalez, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In 10th ACM Workshop on Artificial Intelligence and Security, AISec '17, pp. 27–38, 2017. ACM.

B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. ArXiv e-prints, 2018.

M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In 39th IEEE Symp. on Security and Privacy, 2018.

Model Inversion Attacks

Privacy Attacks

- ***Goal*:** to extract users' sensitive information (e.g., face templates stored during user enrollment)
  - *Fredrikson, Jha, Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. ACM CCS, 2015*
- Also known as hill-climbing attacks in the biometric community
  - *Adler. Vulnerabilities in biometric encryption systems. 5th Int'l Conf. AVBPA, 2005*
  - *Galbally, McCool, Fierrez, Marcel, Ortega-Garcia. On the vulnerability of face verification systems to hill-climbing attacks. Patt. Rec., 2010*
- ***How***: by repeatedly querying the target system and adjusting the input sample to maximize its output score (e.g., a measure of the similarity of the input sample with the user templates)

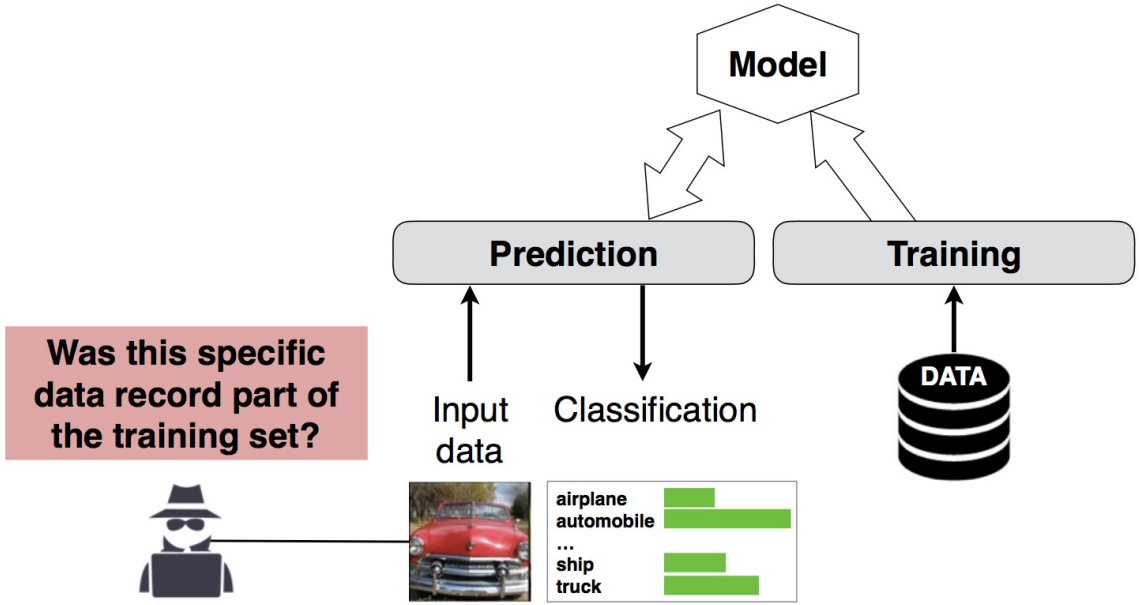
Training Image

Reconstructed Image

Membership Inference Attacks

Privacy Attacks (Shokri et al., IEEE Symp. SP 2017)

- ***Goal:*** to identify whether an input sample is part of the training set used to learn a deep neural network based on the observed prediction scores for each class



Is AI/ML Security really Relevant from a More Practical/Business Perspective?

Industry Survey on AI Security (Microsoft)

- Microsoft has seen a notable increase in attacks on commercial ML systems
- **Market reports:** *Gartner's Top 10 Strategic Technology Trends for 2020*: "Through 2022, 30% of all AI cyberattacks will leverage training-data poisoning, AI model theft, or adversarial samples to attack AI-powered systems."
- Despite these reasons to secure ML systems, Microsoft's survey spanning 28 businesses found that most industry practitioners have yet to come to terms with adversarial machine learning
- 25/28 businesses don't have the right tools in place to secure their ML systems and need guidance

TABLE I
ORGANIZATION SIZE

Organization size	*Count*
Large Organizations (> 1000 employees)	18
Small-and-Medium Size Businesses	10

TABLE II
ORGANIZATION TYPES

Organization	*Count*
Cybersecurity	10
Healthcare	5
Government	4
Consulting	2
Banking	2
Social Media Analytics	1
Publishing	1
Agriculture	1
Urban Planning	1
Food Processing	1
Translation	1

TABLE III
ML STRATEGY

How do you build ML Systems	*Count*
Using ML Frameworks	16
Using ML as a Service	10
Building ML Systems from scratch	2

TABLE IV
STATE OF ADVERSARIAL ML

Do you secure your ML systems today	*Count*
Yes	3
No	22

TABLE V
TOP ATTACK

Which attack would affect your org the most?	*Distribution*
Poisoning (e.g: [21])	10
Model Stealing (e.g: [22])	6
Model Inversion (e.g: [23])	4
Backdoored ML (e.g: [24])	4
Membership Inference (e.g: [25])	3
Adversarial Examples (e.g: [26])	2
Reprogramming ML System (e.g: [27])	0
Adversarial Example in Physical Domain (e.g: [5])	0
Malicious ML provider recovering training data (e.g: [28])	0
Attacking the ML supply chain (e.g: [24])	0
Exploit Software Dependencies (e.g: [29])	0

https://www.microsoft.com/security/blog/2020/10/22/cyberattacks-against-machine-learning-systems-are-more-common-than-you-think/

R.S. Kumar et al., Microsoft, *Adversarial Machine Learning – Industry Perspectives*, AISec 2020

Startups and Standardization Efforts

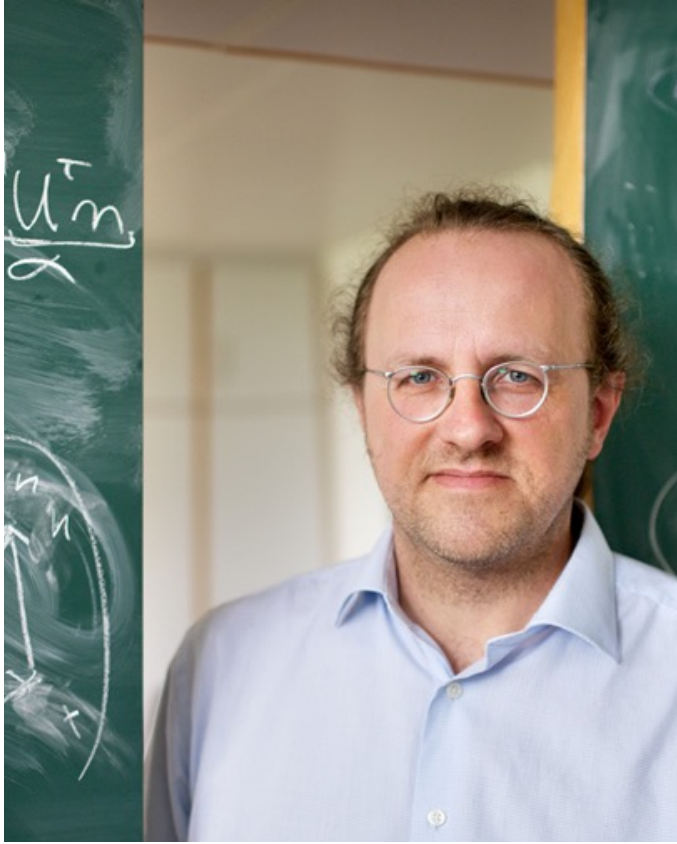
- https://adversa.ai
- https://www.robustintelligence.com
- https://latticeflow.ai
- https://resistant.ai
- https://troj.ai
- https://www.calypsoai.com

- ETSI working group on AI security
  - https://www.etsi.org/technologies/securing-artificial-intelligence

Why Is AI Vulnerable?

Why Is AI Vulnerable?

- **Underlying assumption:** past data is *representative* of future data (IID data)
- The success of modern AI is on tasks for which we collected enough representative training data
- **We cannot build AI models for each task an agent is ever going to encounter,** but there is a whole world out there where the IID assumption is violated
- **Adversarial attacks** point exactly at this lack of robustness which comes from IID specialization



Bernhard Schölkopf

Director, Max Planck Institute, Tuebingen, Germany

Why Is AI Safety an Important Concern?

- We learn how to break machine learning and AI not just because it is fun, but...
  - to understand the limits of these technologies
  - to be able to design more robust algorithms and systems

- Systems that can be used in safety-critical applications
  - e.g., self-driving cars, monitoring / controlling nuclear plants

- Knowing when to ***trust*** automated decisions in these contexts is extremely important
  - Should I use the autopilot of my self-driving car or not? Can I trust it?

Hacking Tesla Autopilot

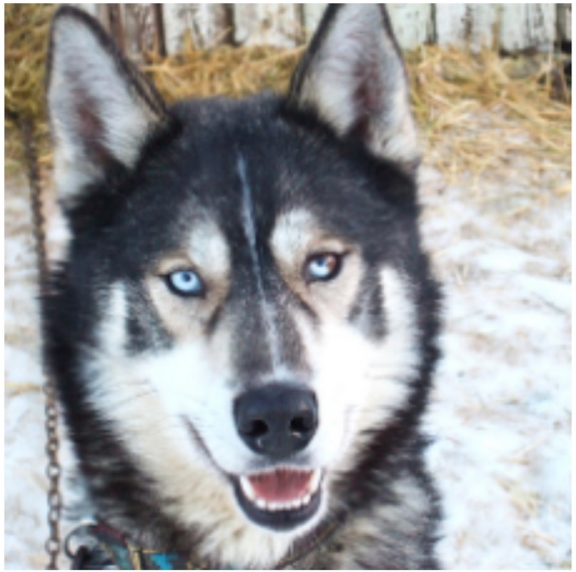
Explainability Is Another Important Asset for AI Safety

- How can we trust a black-box algorithm providing *opaque* decisions?
  - *Why did my car decide to turn left rather than right?*
  - *Why is this application considered malicious / harmful?*

- The right to explanation (https://en.wikipedia.org/wiki/Right_to_explanation)
  - EU on General Data Privacy Regulation (GDPR), Art. 22

- Important concept
  - to build trust in machines and automated algorithms
  - to understand if the algorithm has properly learned meaningful notions/abstractions from data
  - to uncover potential biases encountered during the learning process

An Example on Image Classification



(a) Husky classified as wolf

(b) Explanation

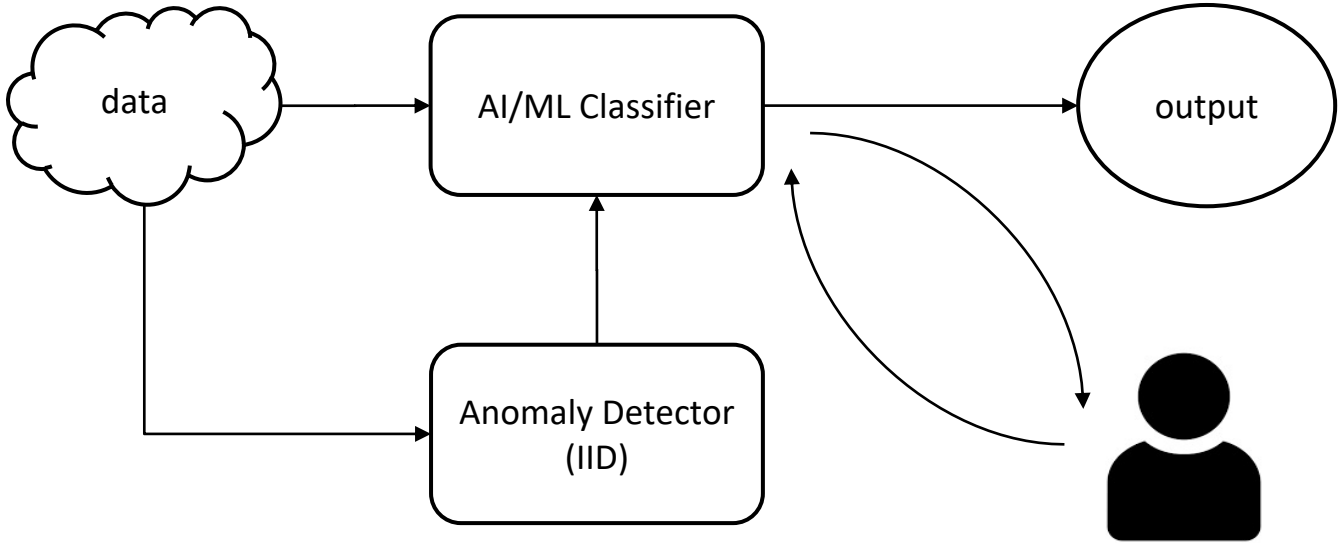
Ribeiro et al., Why Should I Trust You?... KDD 2016

The Pillars of Trustworthy AI

- **Safety, Robustness and Reliability**
  - AI systems should perform reliably and safely
- **Transparency, Interpretability and Explainability**
  - AI systems should be understandable
- **Accountability**
  - AI systems should have algorithmic accountability
- **Security and Privacy**
  - AI systems should be secure and respect privacy
- **Fairness**
  - AI systems should treat all people fairly
- **Inclusiveness**
  - AI systems should empower everyone and engage people

https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

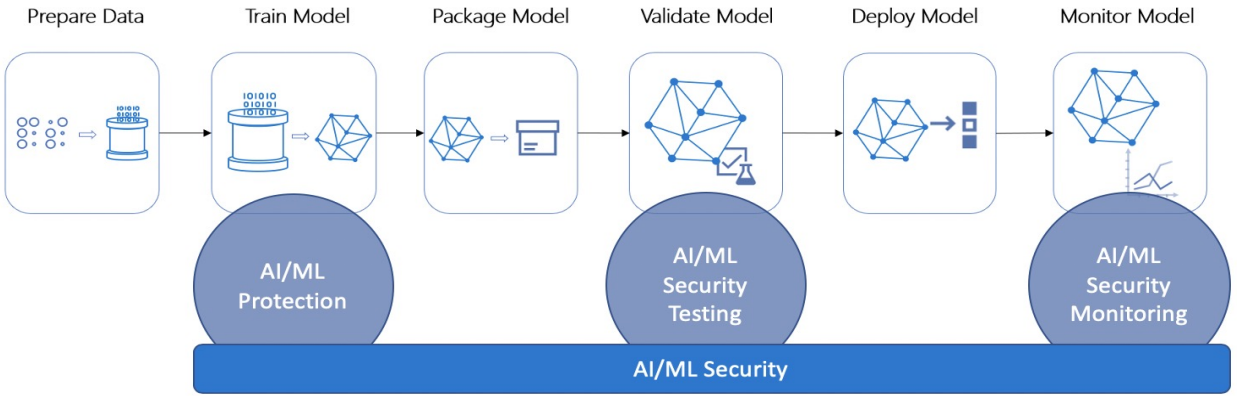
What Can We Do, Next? Explainability/Humans in the Loop



Human in the loop, giving feedback on *explainable* decisions

What Can We Do, Next? From MLOps to MLSecOps

- MLOps aims at automating development & operations of ML
- ML***Sec***Ops aims at adding ML security to the MLOps development & deployment cycle



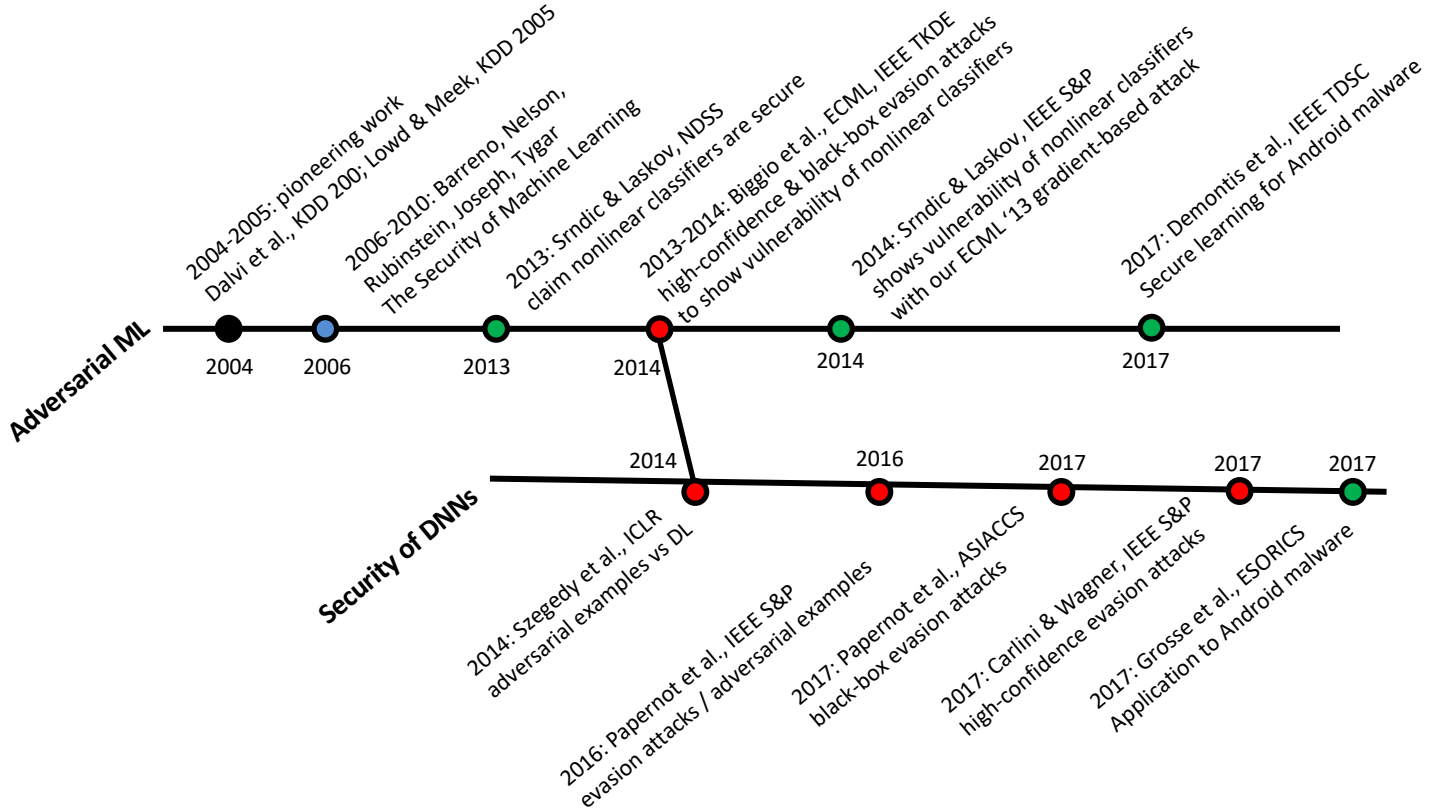
To Conclude…

This is a recent research field…

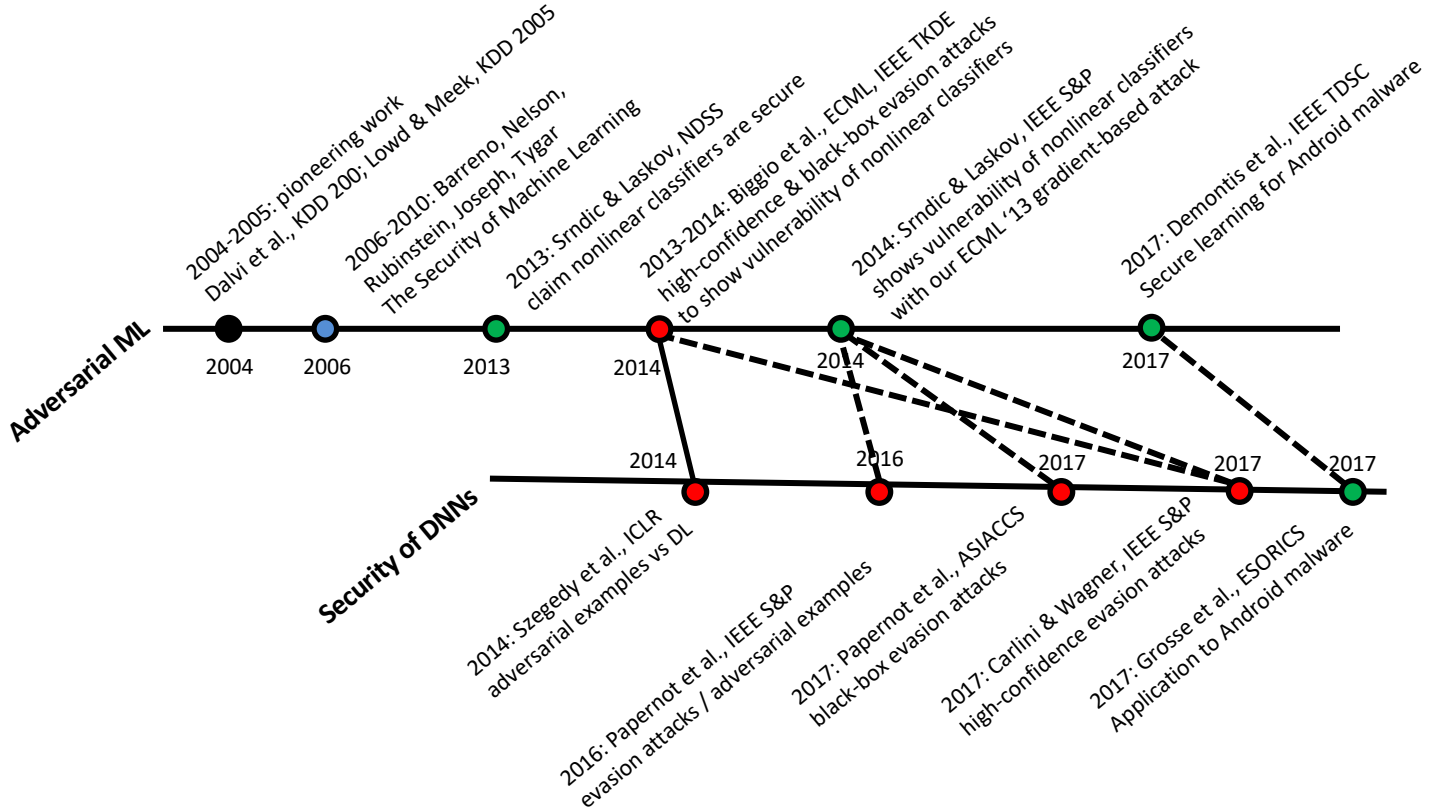
Dagstuhl Perspectives Workshop on
"Machine Learning in Computer Security"
Schloss Dagstuhl, Germany, Sept. 9th-14th, 2012

SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik

Timeline of Learning Security



Timeline of Learning Security



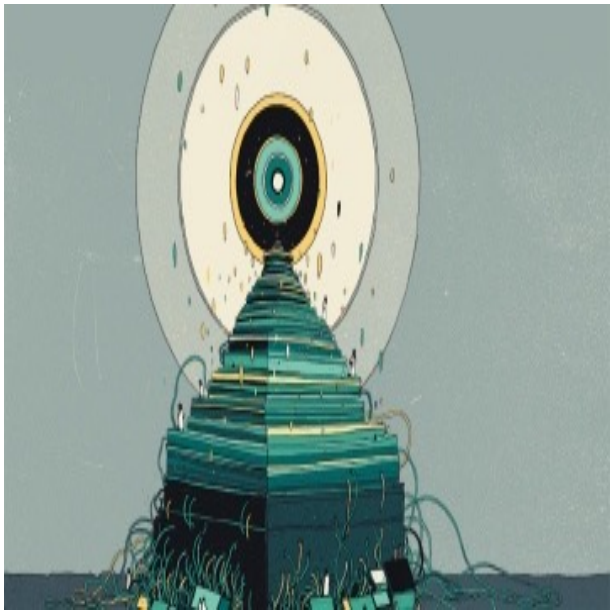
Black Swans to the Fore

[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…

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

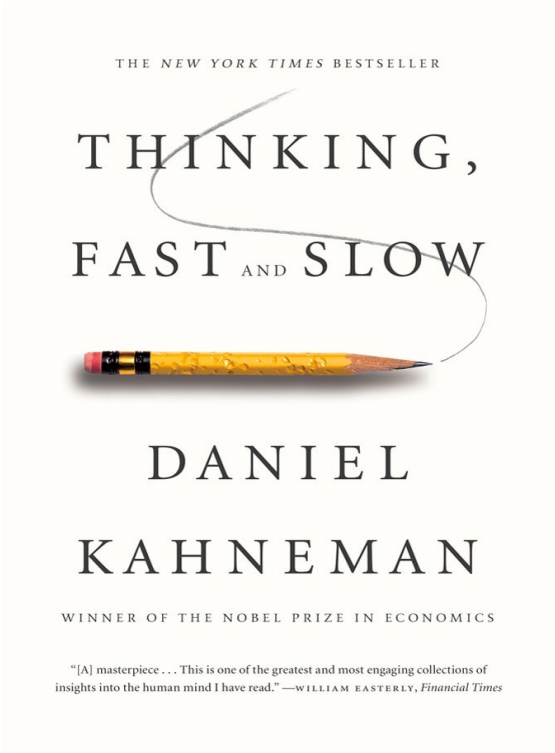
$$\begin{cases} \text{bat}+\text{ball}=\$1.10 \\ \text{bat}=\text{ball}+\$1.0 \end{cases}$$

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 humans and when 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**

Battista Biggio
battista.biggio@unica.it
@biggiobattista

Thanks!

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