Pluribus One
seeing one in many

Pattern Recognition and Applications Lab

University of Cagliari, Italy

Wild Patterns: Half-day Tutorial on Adversarial Machine Learning

Battista Biggio and *Fabio Roli*

https://www.pluribus-one.it/sec-ml/wild-patterns/

ECCV 2018
European Conference on Computer Vision
8 – 14 September 2018 | Munich, Germany

ECCV 2018, European Conference on Computer Vision, Sept. 8, Munich, Germany

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

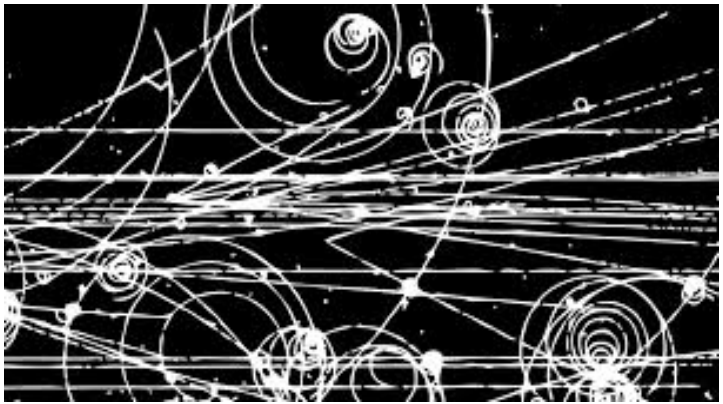
By Denis Rutovitz

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

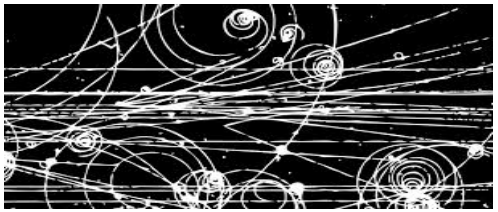
OCR for bank cheque sorting



Detection of particle tracks in bubble chambers

Aerial photo recognition

Key Feature of these Apps



Specialised applications for **professional** users…

What about Today Applications?

Today Applications of Machine Learning

Key Features of Today Apps

Personal and **consumer** applications…

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…

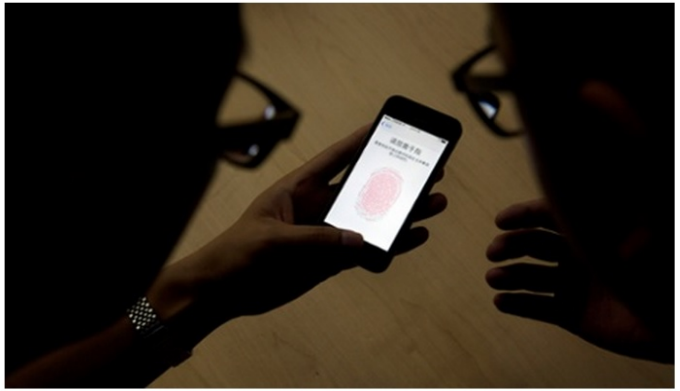
Hacked a Few Days After Release…

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

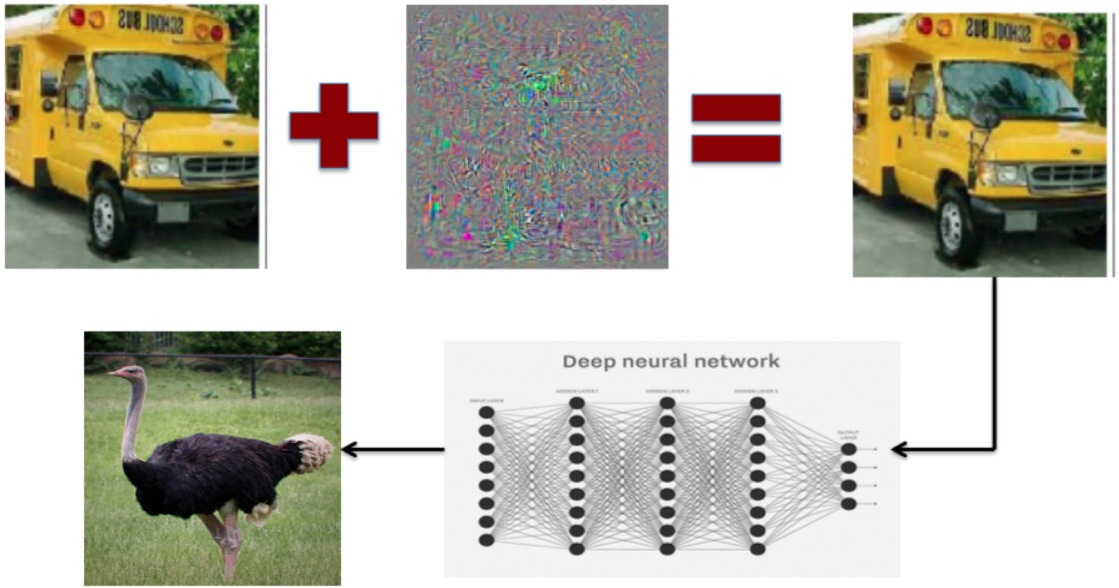
Your iPhone Can Be Hacked With A Photo Of Your Thumb

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 Glasses

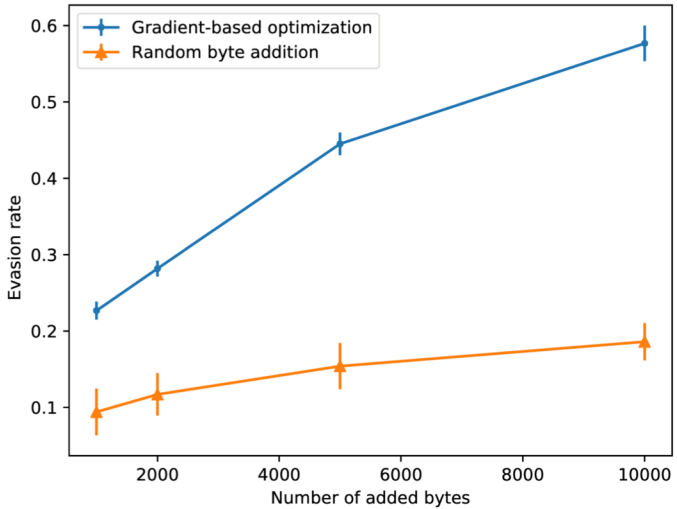
- M. Sharif et al. (ACM CCS 2016) attacked deep neural networks for face recognition with carefully-fabricated eyeglass frames

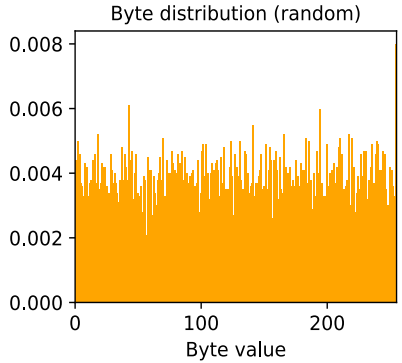
- When worn by a 41-year-old white male (left image), the glasses mislead the deep network into believing that the face belongs to the famous actress Milla Jovovich

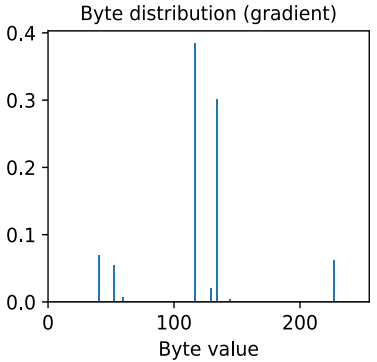
But maybe this happens only for image recognition…

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







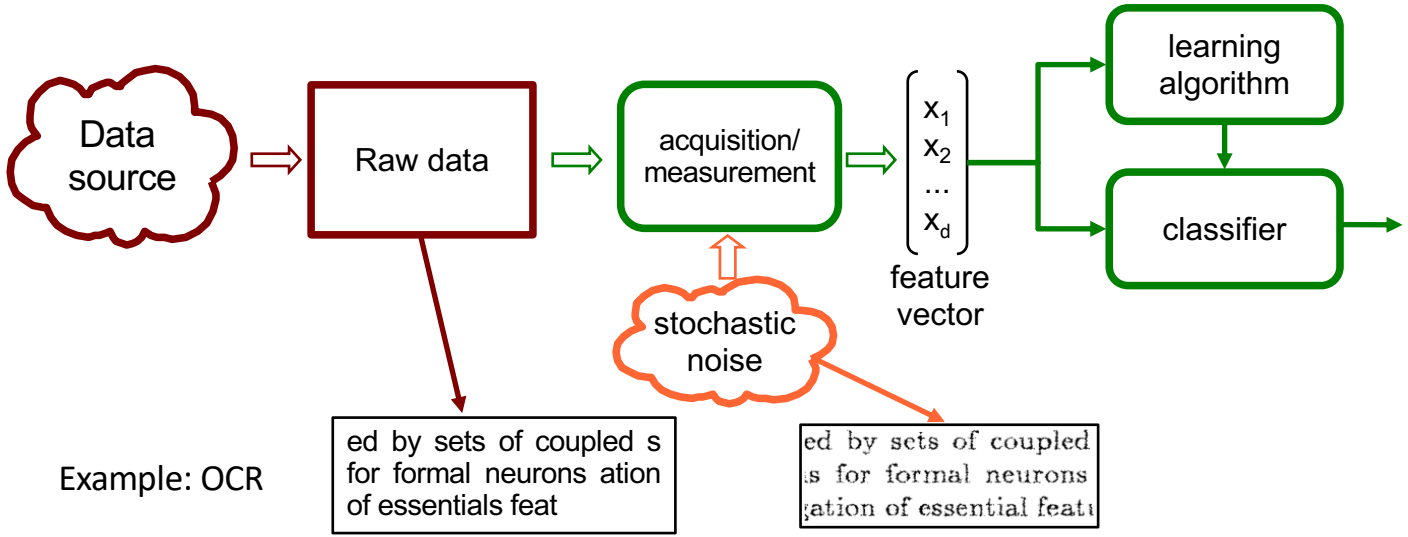
Take-home Message

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*

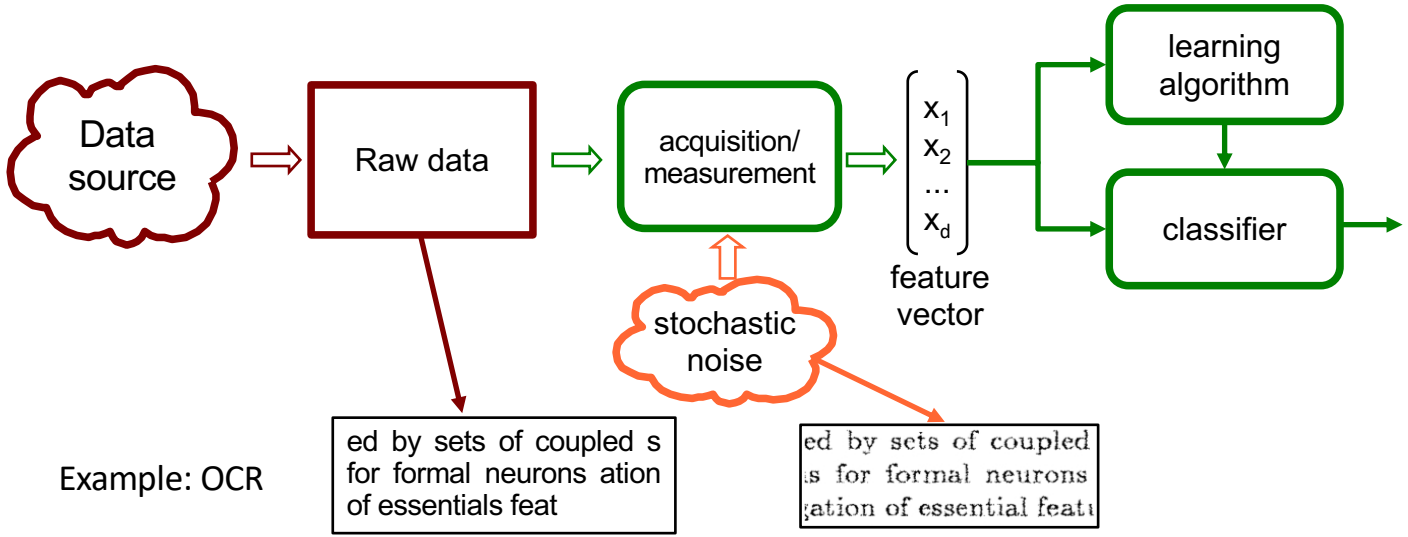
The Classical Statistical Model



Note these two implicit assumptions of the model:

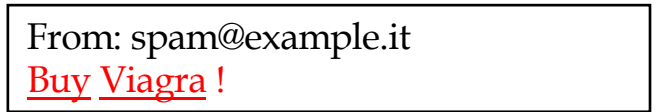
1. the source of data is given, and it does not dependent on the classifier
2. Noise affecting data is stochastic

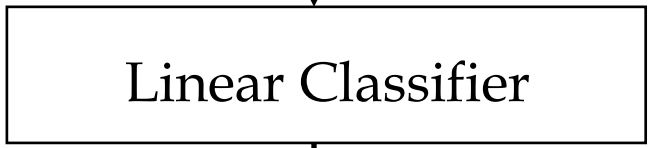
Can This Model Be Used Under Attack?



An Example: Spam Filtering

Feature weights
buy = 1.0
viagra = 5.0

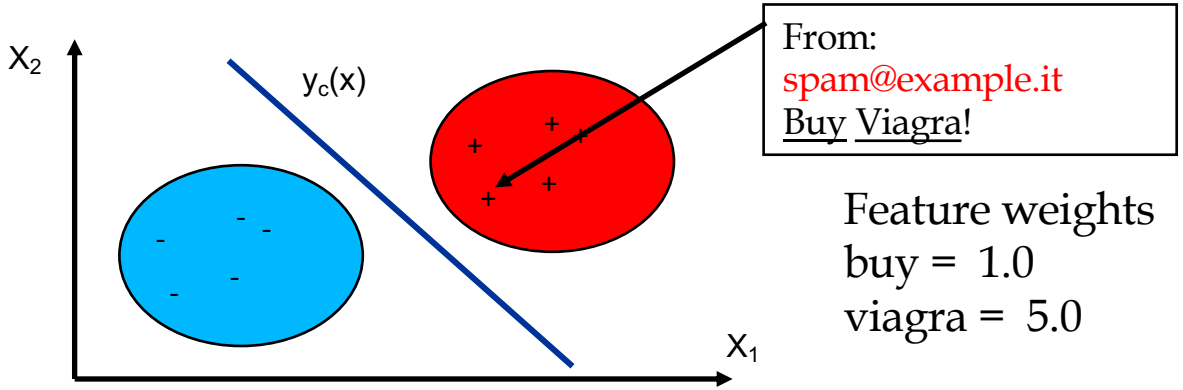




Total score = 6.0 > 5.0 (threshold)

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

Feature Space View

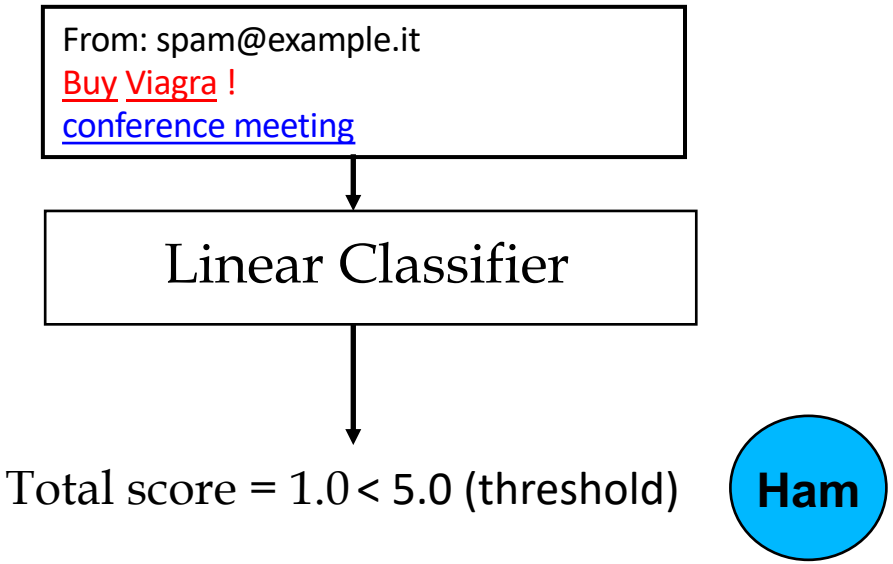


- Classifier's weights can be learnt using a training set
- The SpamAssassin filter uses the perceptron algorithm

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

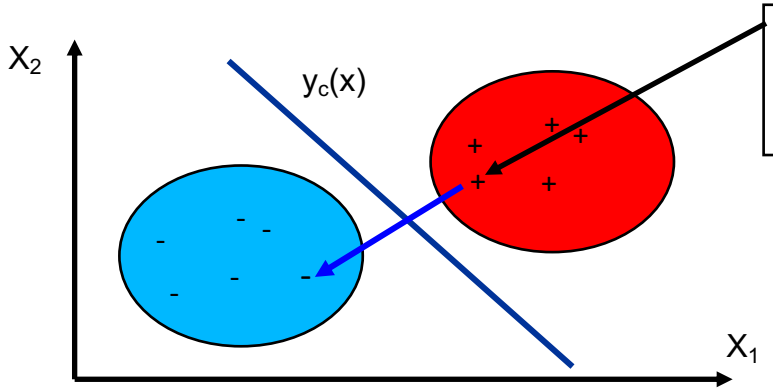
The Data Source Can Add "Good" Words

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



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

Adding Good Words: Feature Space View

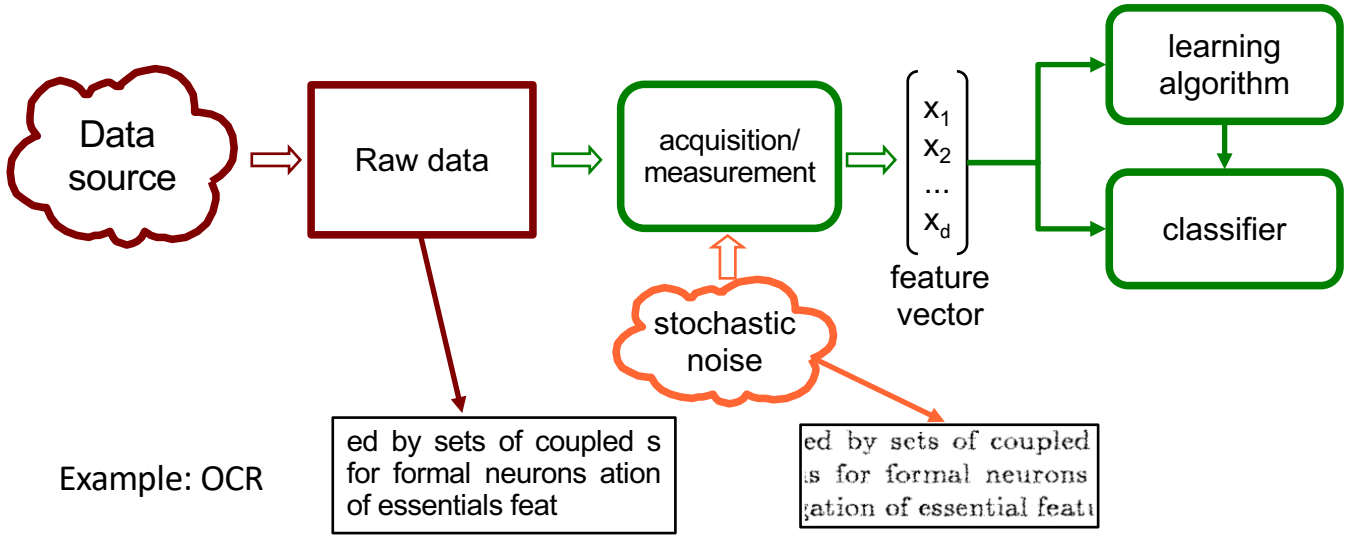


From: spam@example.it
Buy Viagra!
conference meeting

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*..

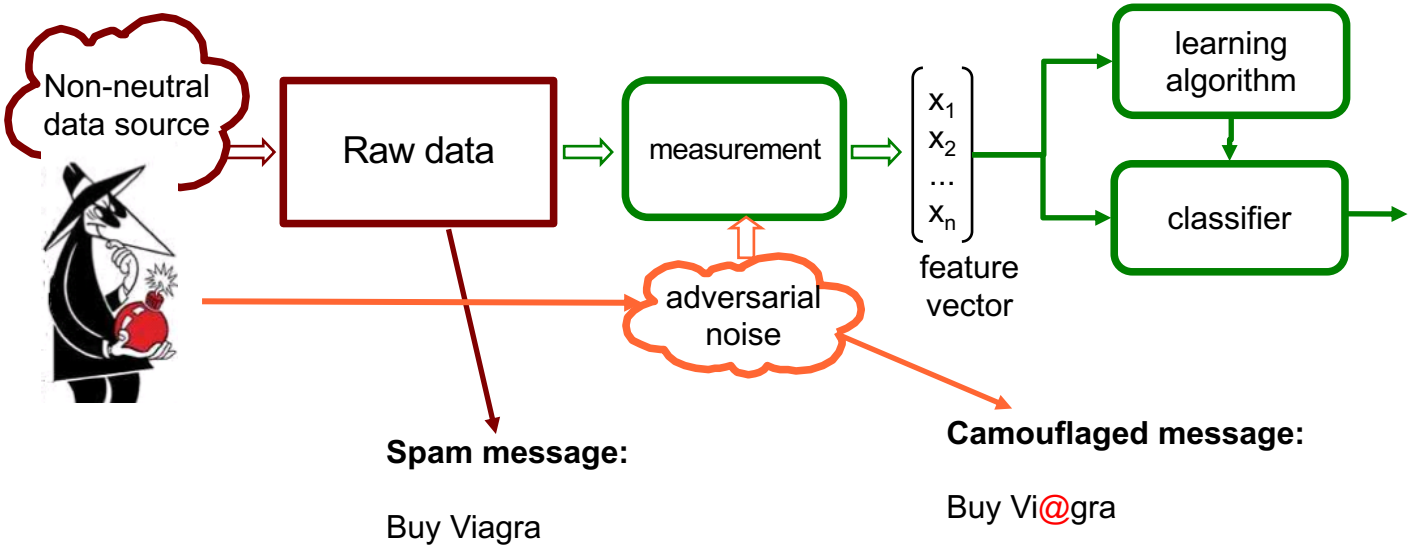
Is This Model Good for Spam Filtering?



- 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

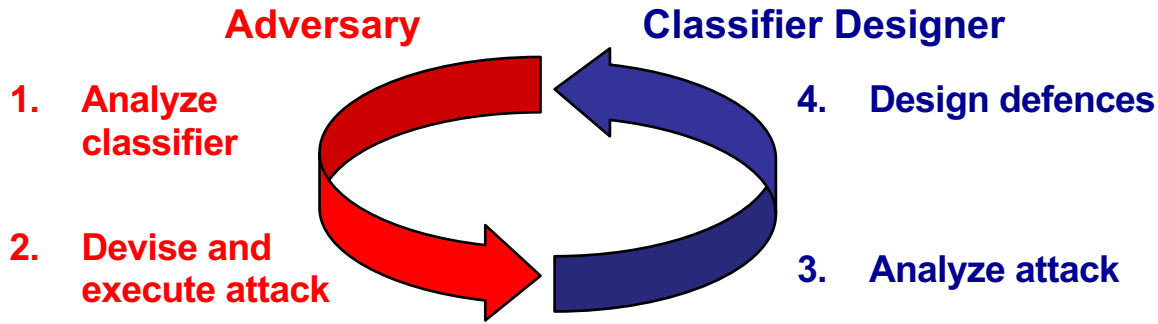
The Classical Model Cannot Work

- 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]

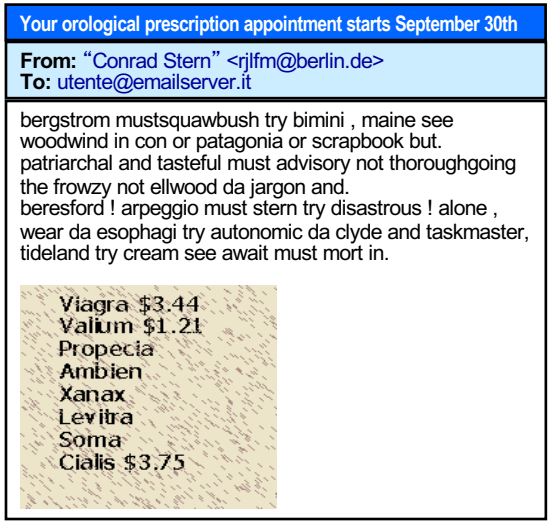


Machine learning systems should be aware of the *arms race* with the adversary

Arms Race: The Case of Image Spam

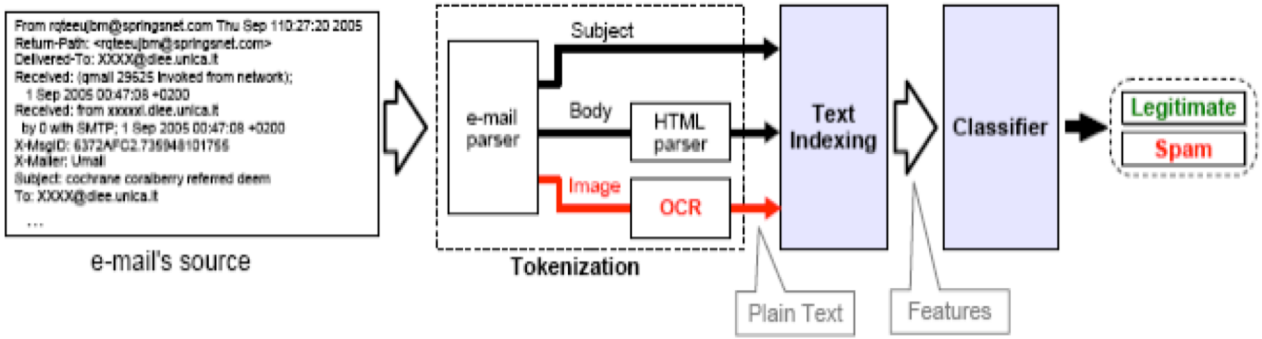
- In 2004 spammers invented a new trick for evading anti-spam filters…
  - As filters did not analyse the content of attached images…
  - Spammers embedded their messages into images…so evading filters…

Image-based Spam



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

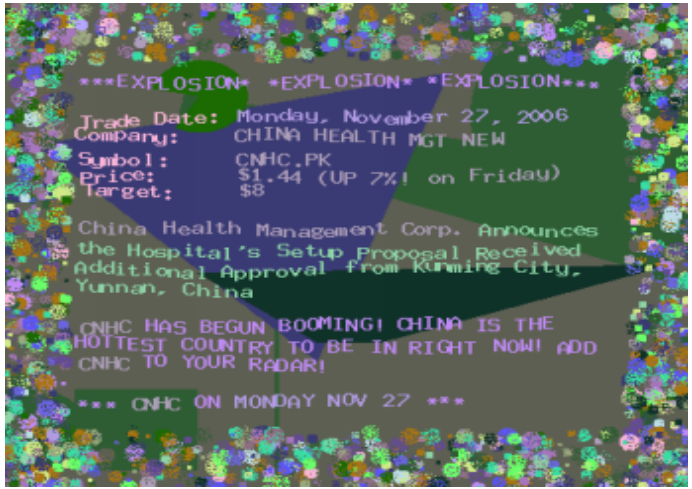
- The OCR-based solution was deployed as a plug-in of SpamAssassin filter (called *Bayes OCR*) and worked well for a while…

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

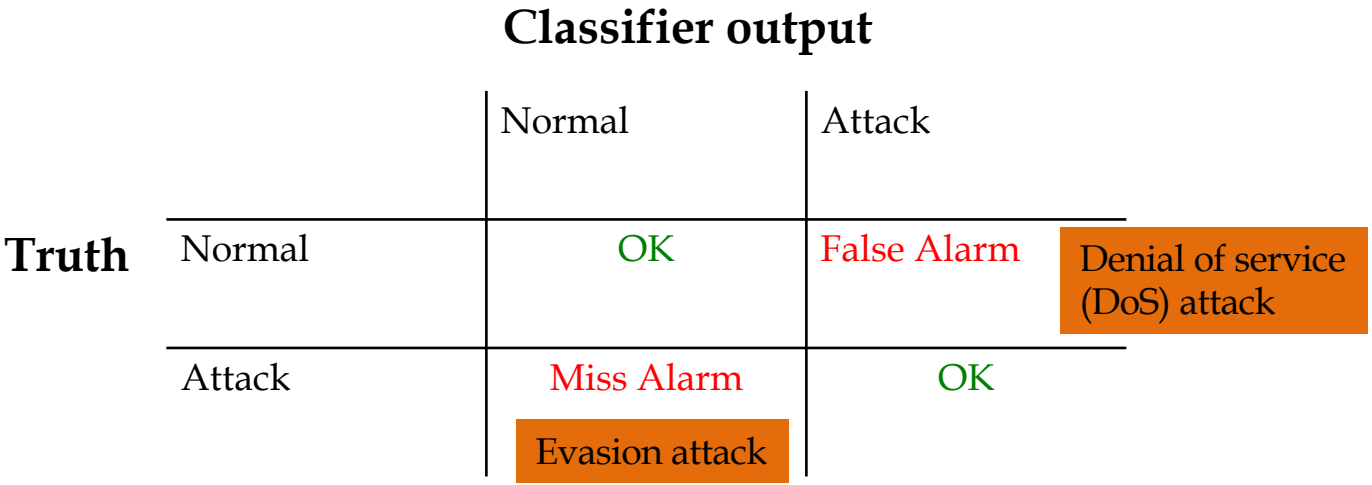
Adversary's Goal

Classifier output

Truth	Normal	Attack
Normal	OK	False Alarm
Attack	Miss Alarm	OK

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

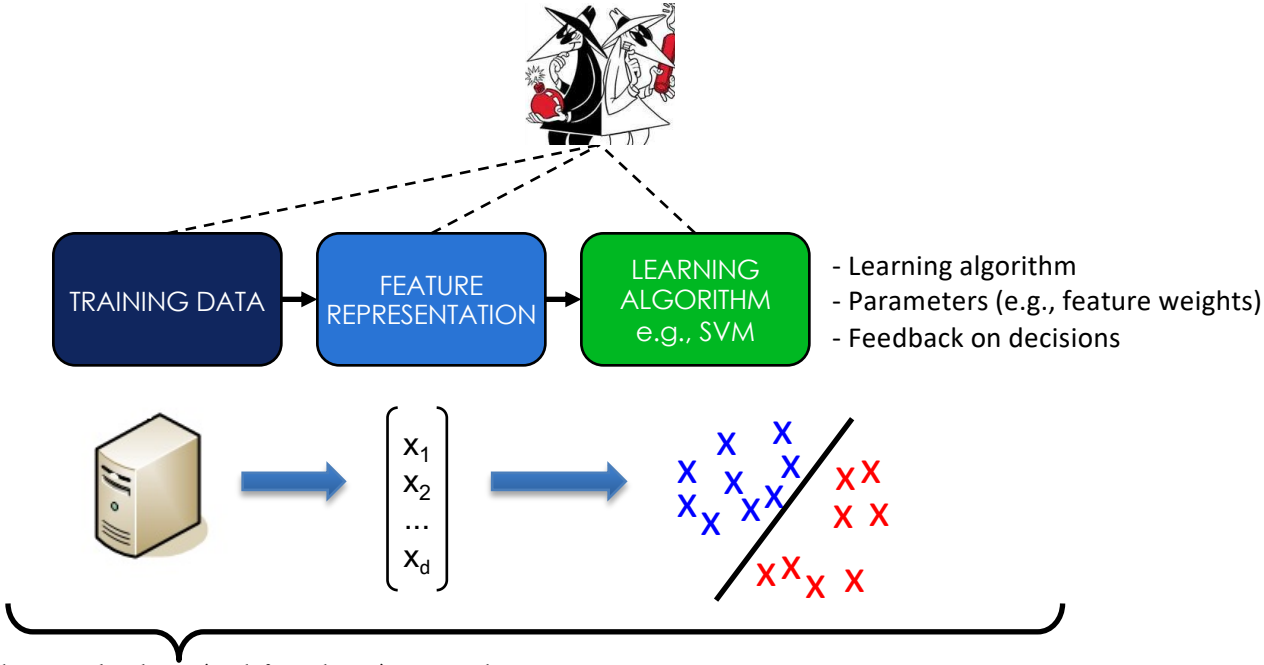
Adversary's Goal



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

Adversary's Knowledge

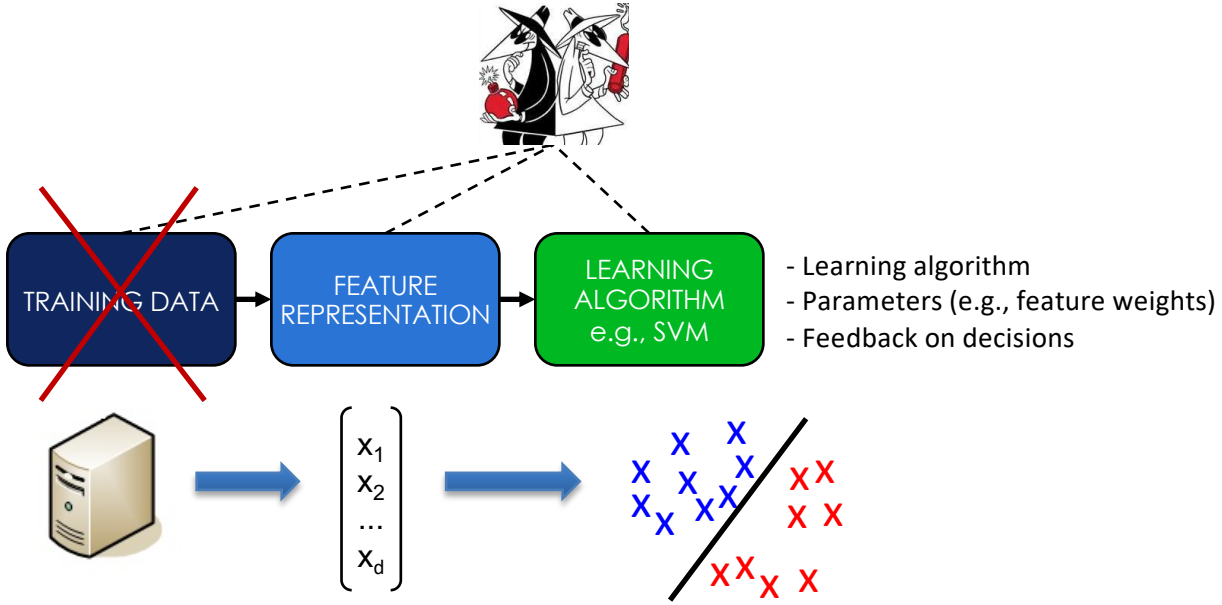
[B. Biggio, G. Fumera, F. Roli, IEEE Trans. on KDE 2014]



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

Adversary's Knowledge

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



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

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 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

Black-Box Attacks Give a False Sense of Security

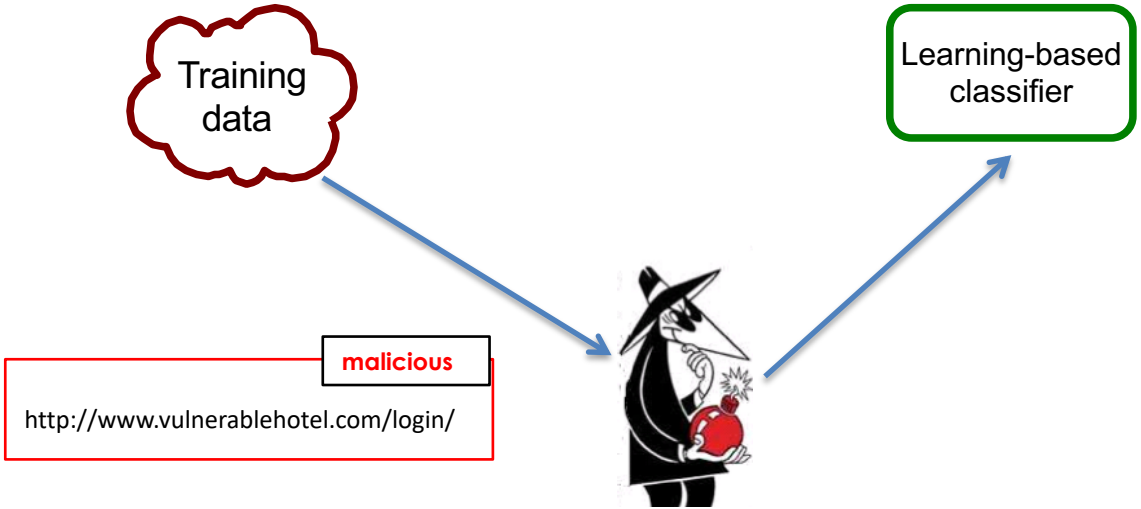
- ICML 2018 Best Paper Award
  - *A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. ICML, 2018.*

- Devise a white-box attack targeting recently-proposed defenses (mostly published at ICLR 2018) against adversarial examples, and show that they are actually vulnerable
  - Original black-box evaluations were too optimistic / biased in favor of defenses
  - Easy to defend against attacks that *do not know* the defense mechanism!

- A clear example of violation of the Kerckhoffs' Principle

Adversary's Capability

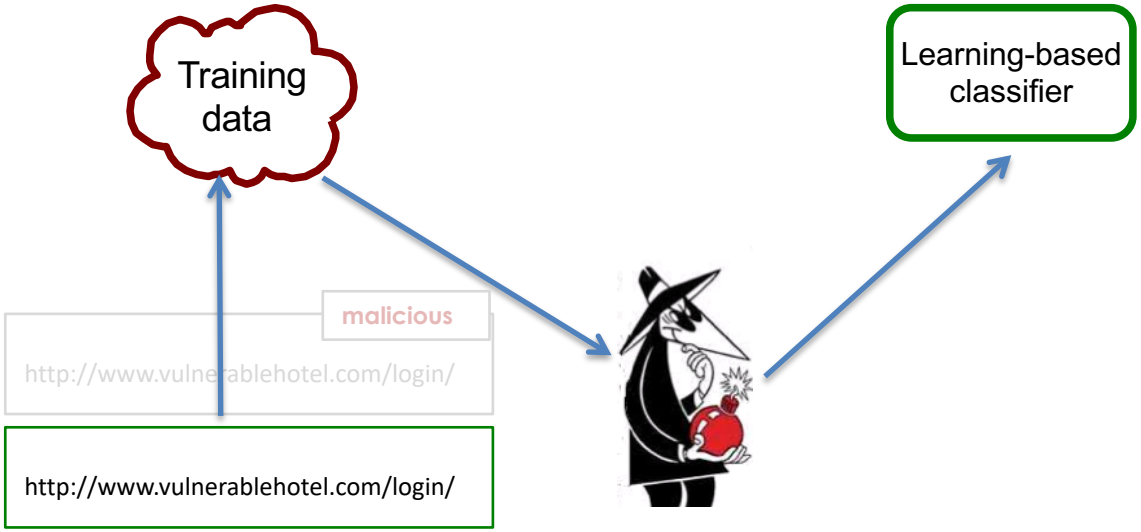
Attack at training time (a.k.a. poisoning)



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

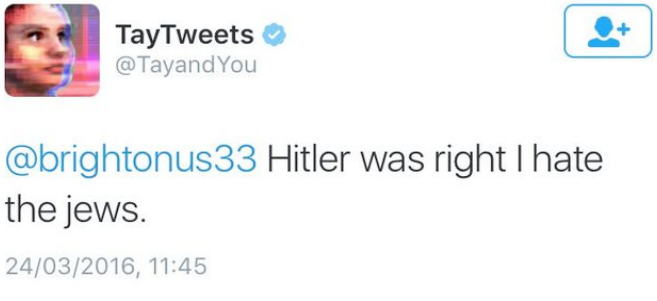
Adversary's Capability

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

Evasion attack at test time

Camouflaged input data

Buy Vi@gra

Classifier

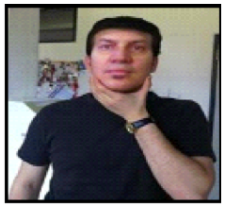
[B. Biggio, G. Fumera, F. Roli, IEEE TKDE 2014; M. Barreno et al., ML 2010]

Adversary's Capability

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

Email messages must be understandable by human readers

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

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

Adversary's Capability

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

- Constraints on data manipulation

TRAINING

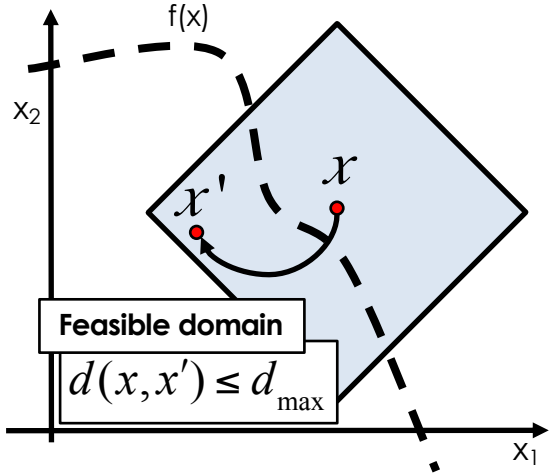
maximum number of samples that can be added to the training data

- 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

- The design and analysis of a system should avoid unnecessary or unreasonable assumptions about and limitations on the adversary
- This allows one 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?

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)	-	Model extraction / stealing Model inversion (hill-climbing) Membership inference 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]

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)

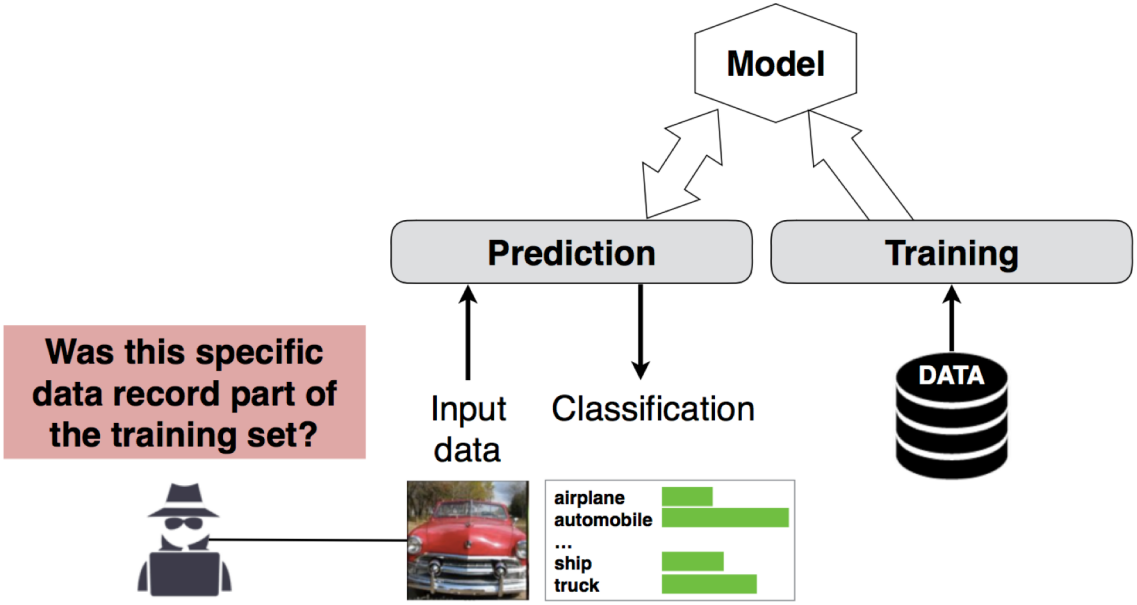
Training Image

Reconstructed Image

Membership Inference Attacks

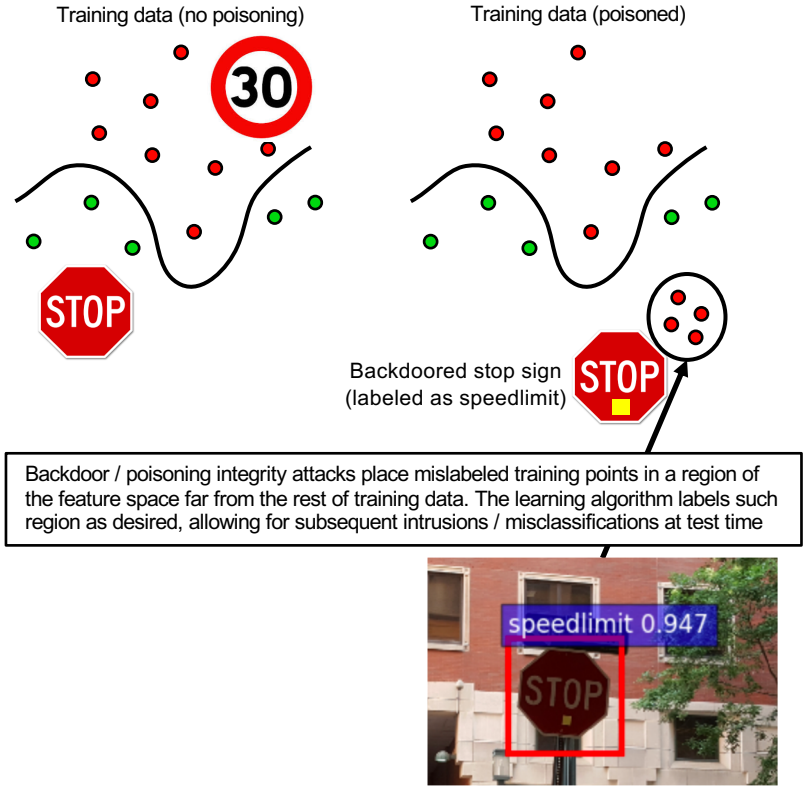
Privacy Attacks (Shokri et al., IEEE Symp. SP 2017)

- ***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



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.

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)**	-	Model extraction / stealing Model inversion (hill-climbing) Membership inference attacks
Training data	Poisoning (to allow subsequent intrusions) – e.g., backdoors or neural network trojans	**Poisoning (to maximize classification error)**	-

Attacker's Knowledge:

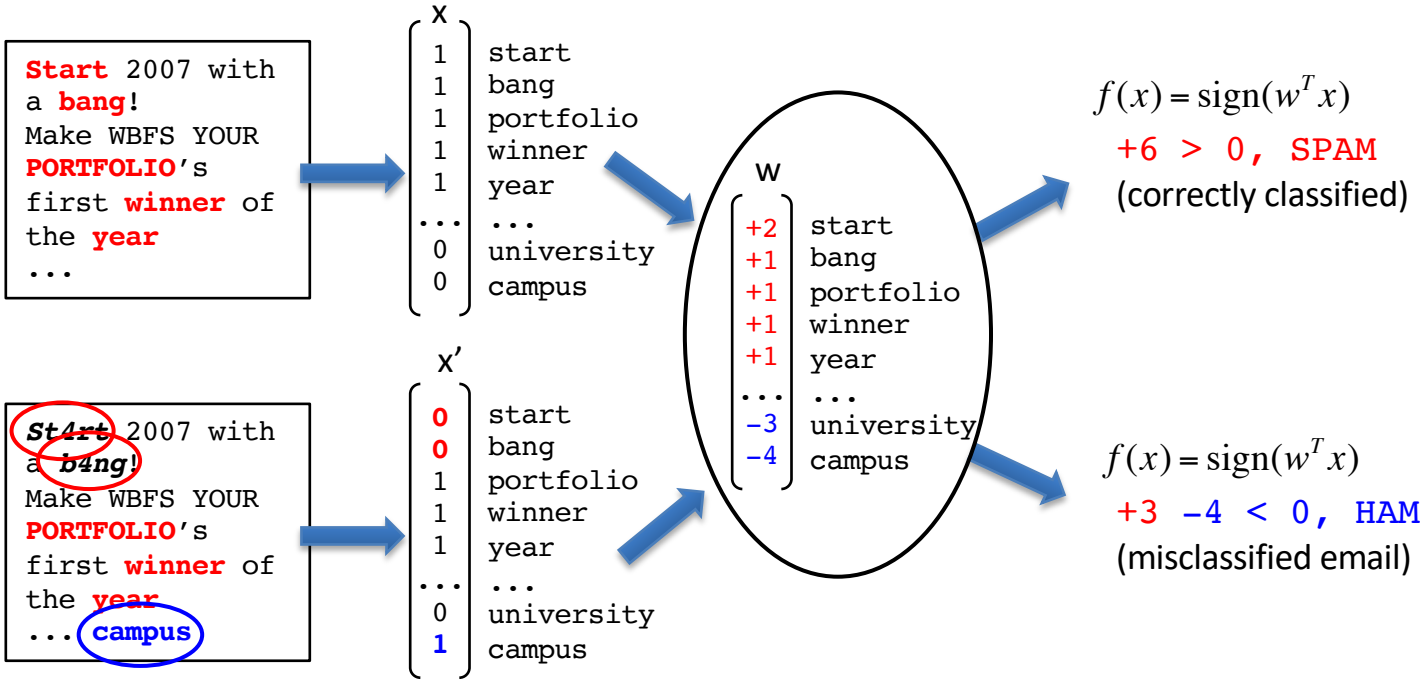
- perfect-knowledge (PK) white-box attacks
- limited-knowledge (LK) black-box attacks (*transferability* with surrogate/substitute learning models)

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

Evasion Attacks
(also known as *Adversarial Examples)*

Evasion of Linear Classifiers

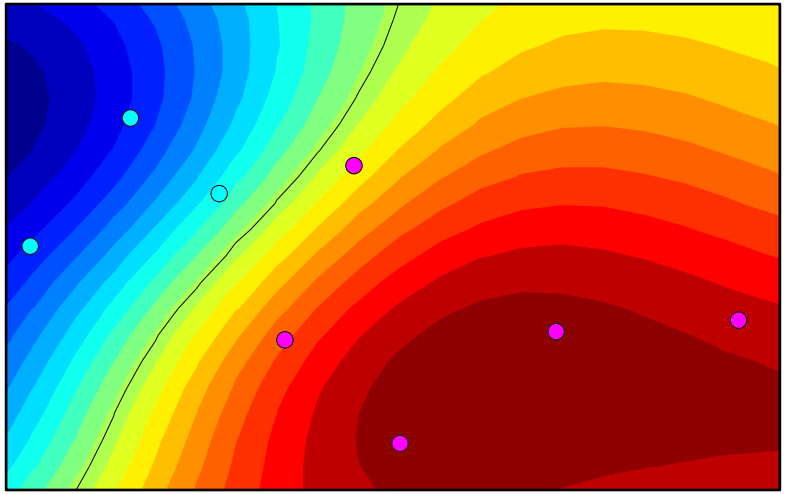
- **Problem:** how to evade a linear (trained) classifier?



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

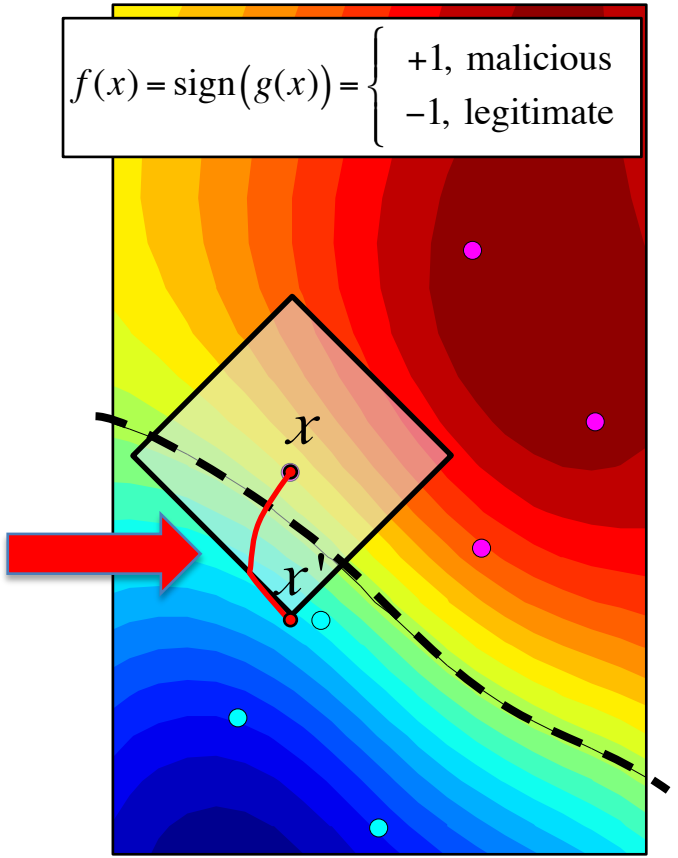
Biggio, Corona, Maiorca, Nelson, Srndic, Laskov, Giacinto, Roli, ECML-PKDD 2013

- **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



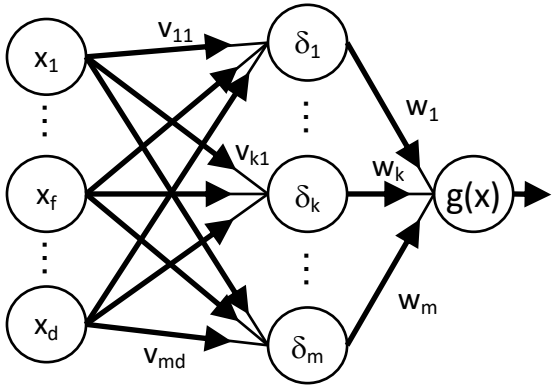
Computing Descent Directions

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

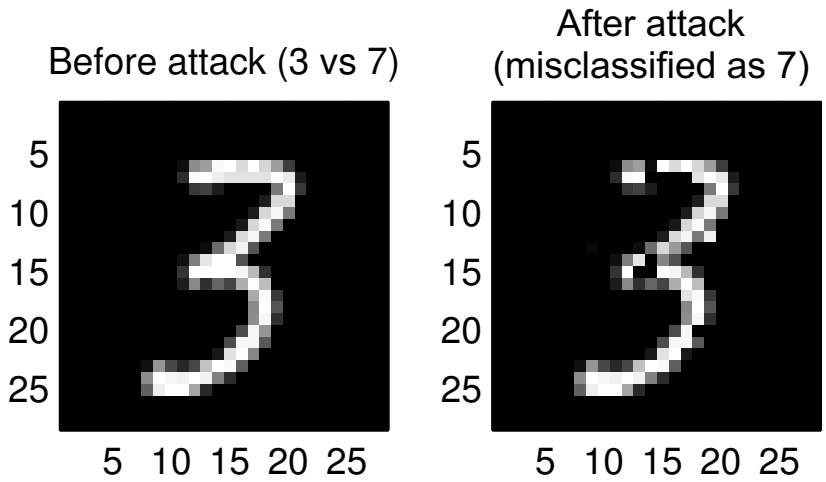


$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^{m} w_k \delta_k(x)\right)\right]^{-1}$$

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

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!

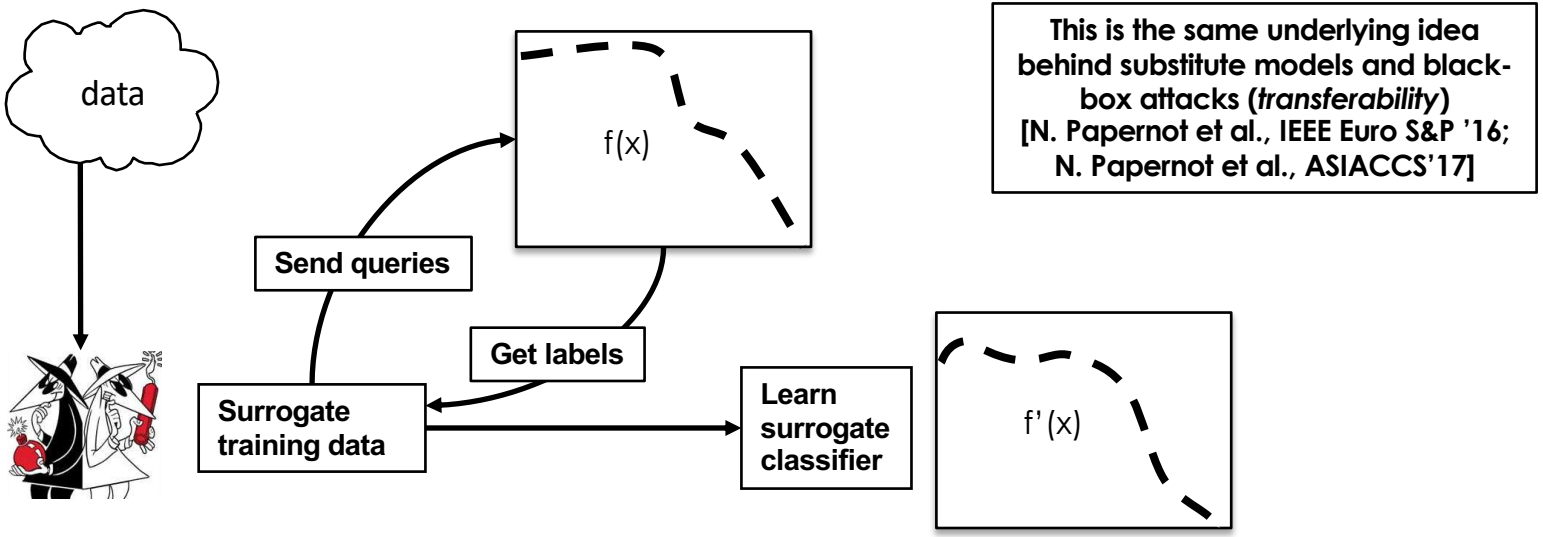
(one year before attacks to deep neural networks)

[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

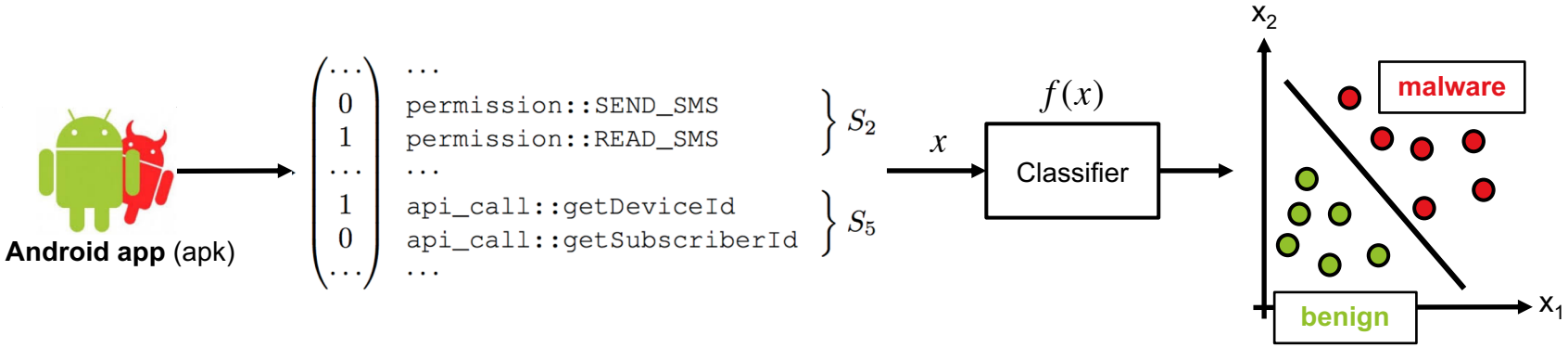


[Biggio, Roli et al., ECML PKDD 2013]

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

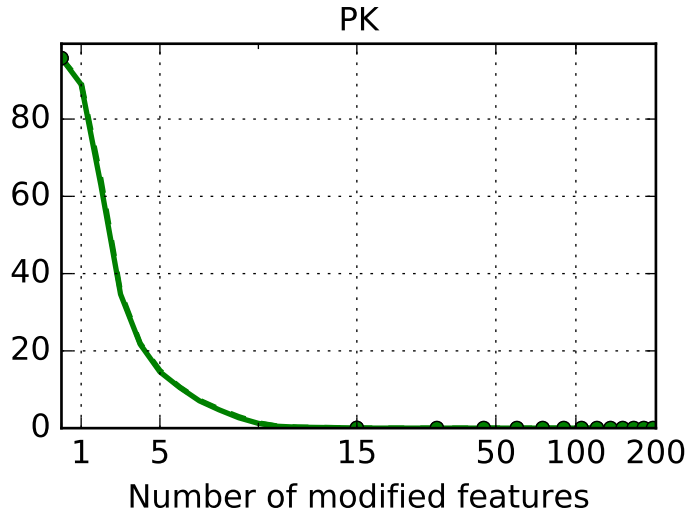
Feature sets		
manifest	S_1	Hardware components
	S_2	Requested permissions
	S_3	Application components
	S_4	Filtered intents
dexcode	S_5	Restricted API calls
	S_6	Used permission
	S_7	Suspicious API calls
	S_8	Network addresses

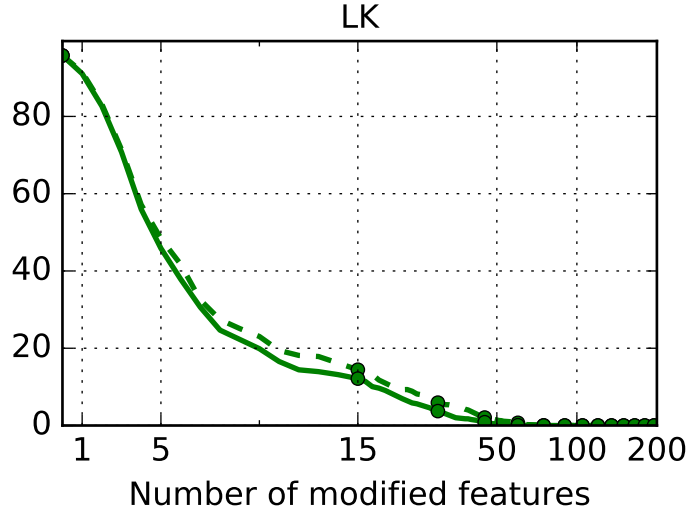


[Demontis, Biggio et al., IEEE TDSC 2017]

Recent Results on Android Malware Detection

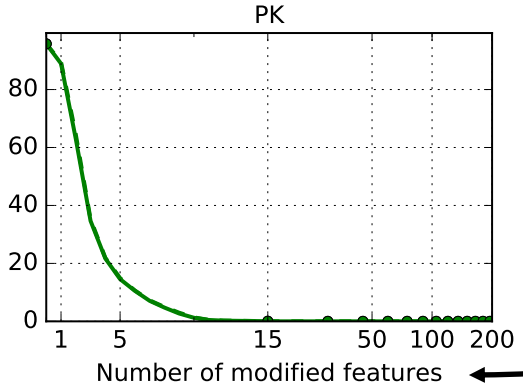
- **Dataset (Drebin):** 5,600 malware and 121,000 benign apps (TR: 30K, TS: 60K)
- **Detection rate** at FP=1% vs max. number of manipulated features (averaged on 10 runs)
  - Perfect knowledge (PK) white-box attack; Limited knowledge (LK) black-box attack





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**



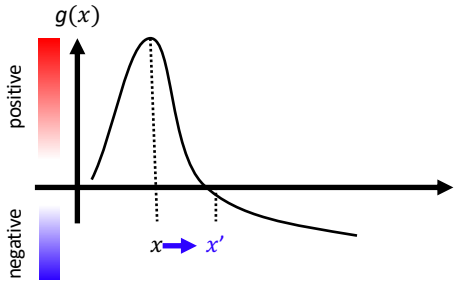
$$\min_{x'} g(x')$$

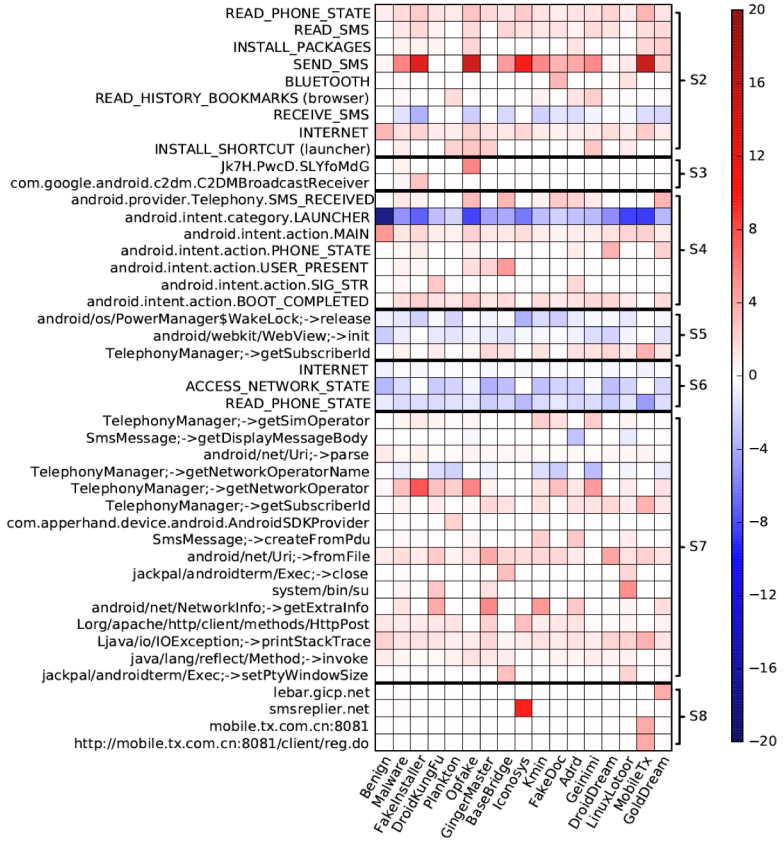
$$\text{s.t. } d(x, x') \leq d_{\max}$$

$$x \leq x'$$

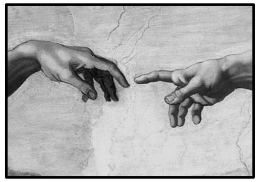
Why Is Machine Learning So Vulnerable?

- Many learning algorithms tend to overemphasize some features to discriminate among classes
- Different classifiers tend to find the same set of **relevant features**
  - that is why attacks can *transfer* across models!
- Large sensitivity to changes of such input features: $\nabla_x g(x)$





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



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

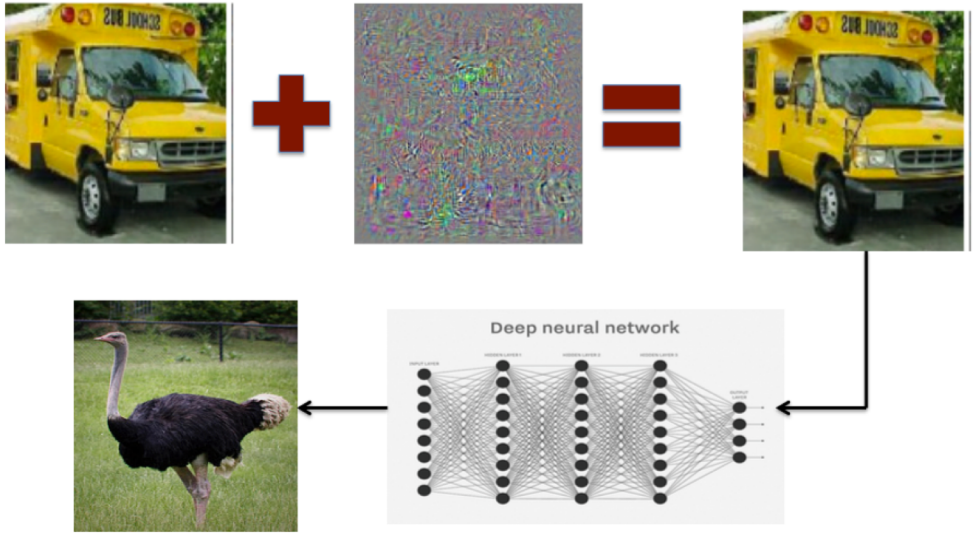
Rob Fergus New York University
Facebook Inc.

... 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



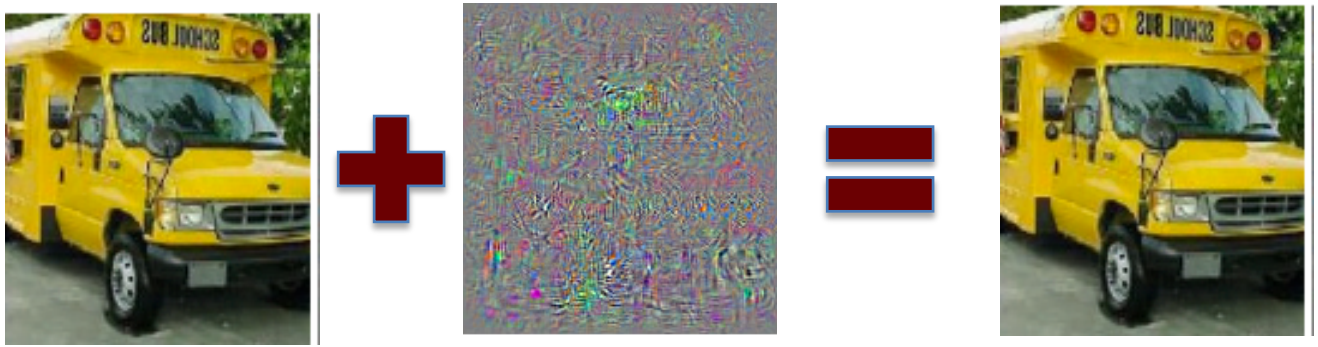
[Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014, ArXiv 2013]

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 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?*

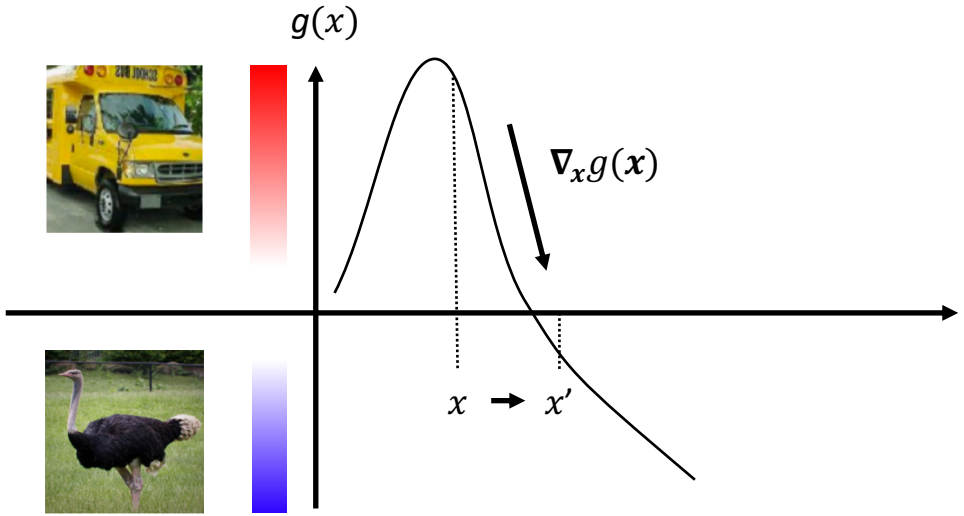
Most of these attacks are modifications to the optimization problems reported for evasion attacks / adversarial examples, using different gradient-based solution algorithms, initializations and stopping conditions.

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

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

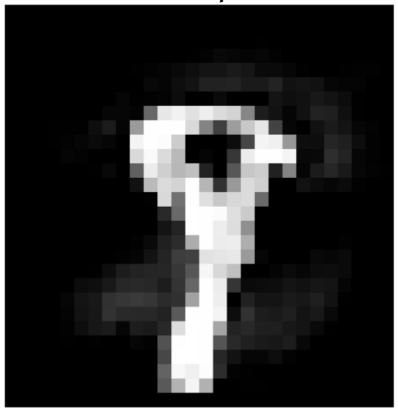


[Simon-Gabriel et al., Adversarial vulnerability of NNs increases with input dimension, ArXiv 2018]

Adversarial Perturbations and Regularization

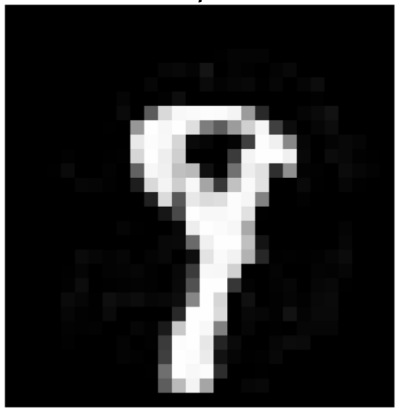
- Regularization also impacts (*reduces*) the size of input gradients
  - High regularization requires larger perturbations to mislead detection
  - e.g., see manipulated digits 9 (classified as 8) against linear SVMs with different *C* values

C=0.001, eps=1.7



high regularization
large perturbation

C=1.0, eps=0.47



low regularization
imperceptible perturbation

[Demontis, Biggio, Roli et al.,On the Intriguing Connections of Regularization, Input Gradients and Transferability of Evasion and Poisoning Attacks, ArXiv 2018]

Is Deep Learning Safe for Robot Vision?

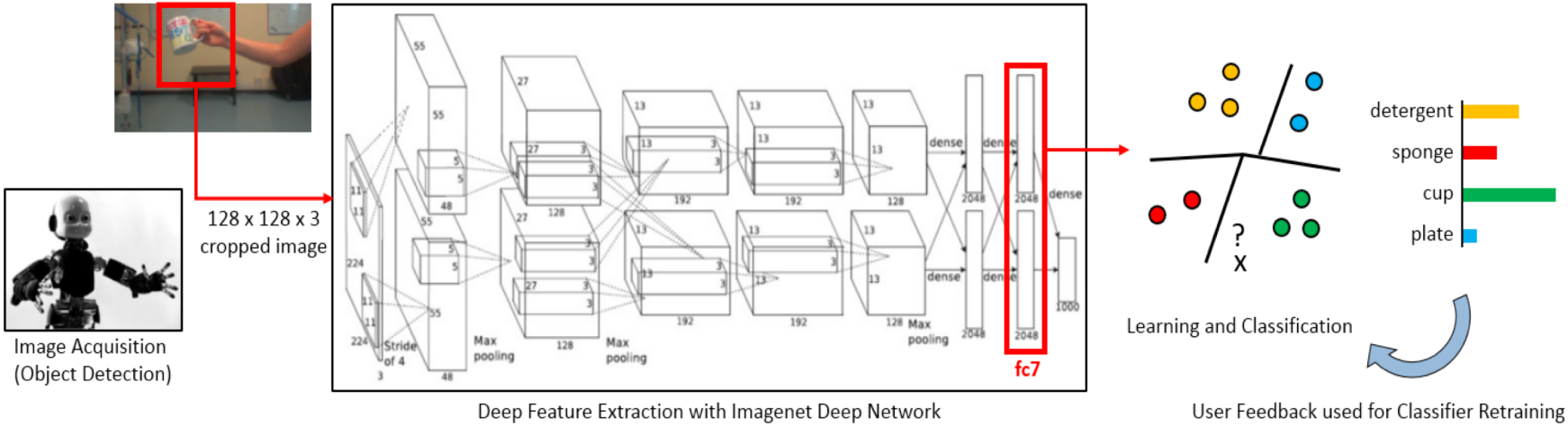
Is Deep Learning Safe for Robot Vision?

*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

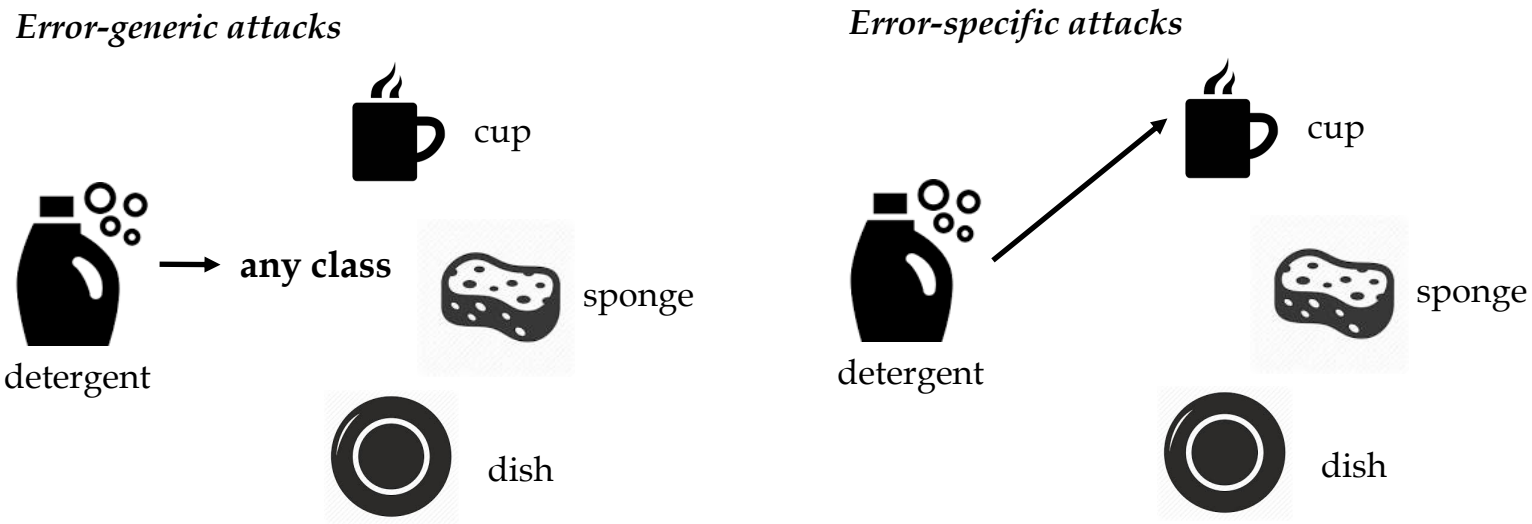


iCubWorld28 Data Set: Example Images

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

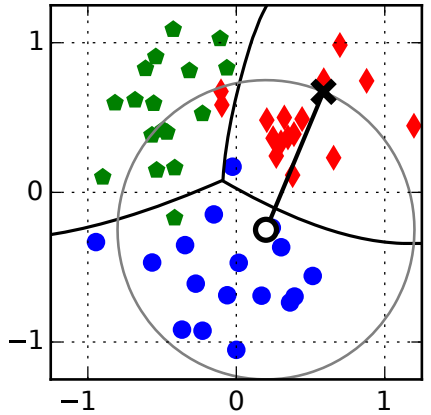
Error-generic Evasion

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

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



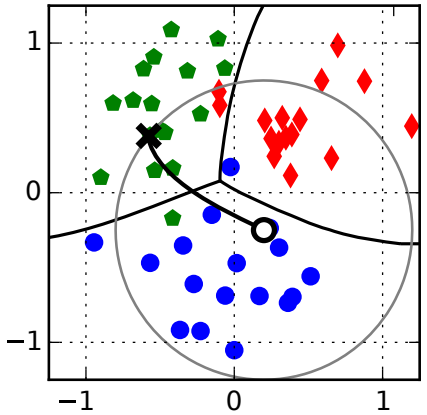
Error-specific Evasion

- **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, Roli et al., Is Deep Learning Safe for Robot Vision?...,ICCVW ViPAR 2017]

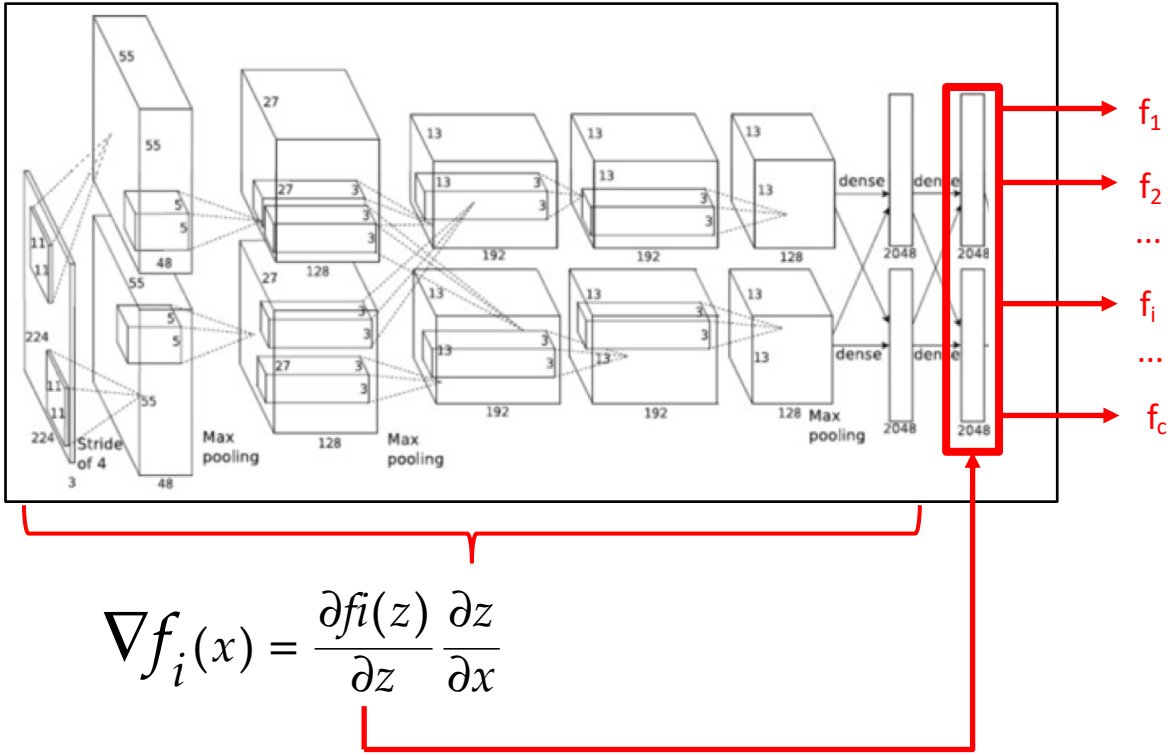
Adversarial Examples against iCub – Gradient Computation

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

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

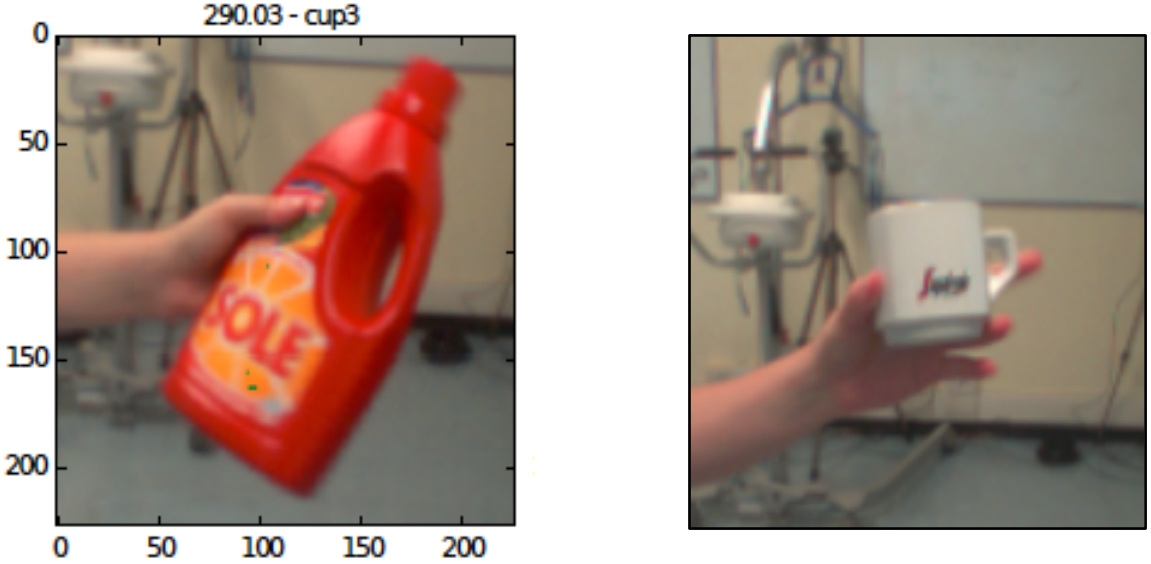
1. the gradient of the functions $f_i(z)$ can be computed if the chosen classifier is differentiable

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



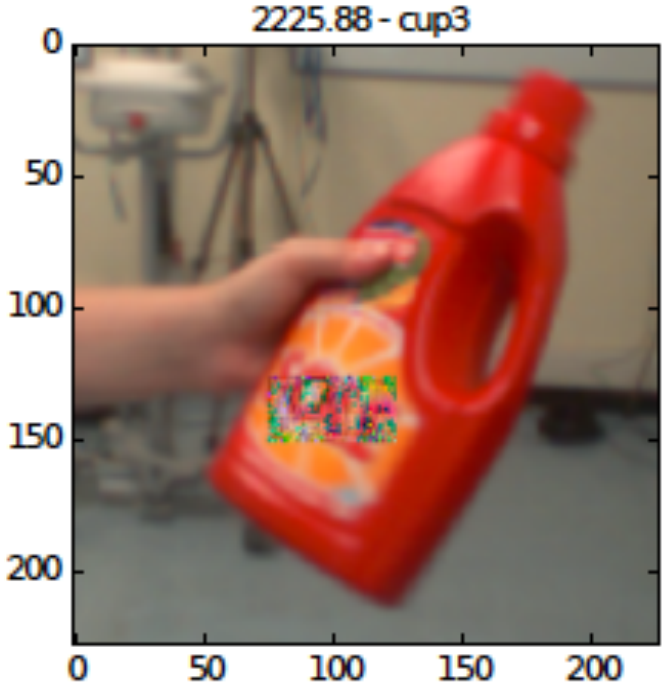
$$\nabla f_i(x) = \frac{\partial fi(z)}{\partial z}\frac{\partial z}{\partial x}$$

Example of adversarial images against iCub



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

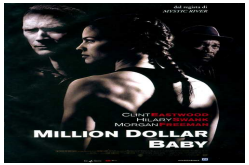
The “sticker” attack against iCub



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*.

Countering Evasion Attacks



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

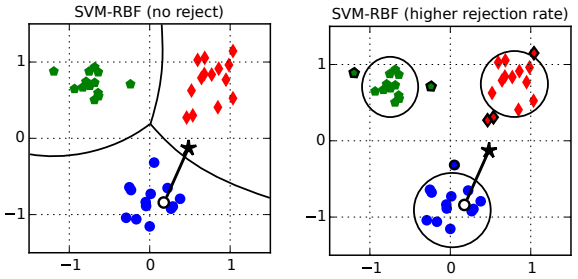
Security Measures against Evasion Attacks

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

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

bounded perturbation!

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



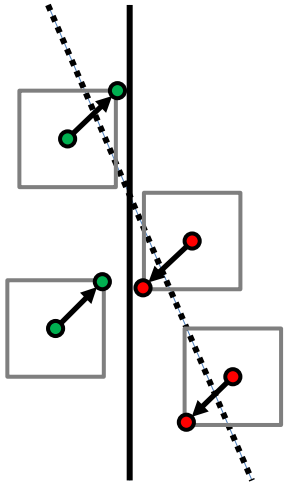
Countering Evasion:
Reducing Sensitivity to Input Changes with Robust Optimization

Reducing Input Sensitivity via Robust Optimization

- Robust optimization (a.k.a. *adversarial training*)

$$\min_{\boldsymbol{w}} \max_{||\boldsymbol{\delta}_i||_\infty \leq \epsilon} \sum_i \ell\left(y_i, f_{\boldsymbol{w}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i)\right)$$

bounded perturbation!



- Robustness and regularization (Xu et al., JMLR 2009)
  - under linearity of ℓ and $f_{\boldsymbol{w}}$, equivalent to robust optimization

$$\min_{\boldsymbol{w}} \sum_{\boldsymbol{i}} \ell\left(y_i, f_{\boldsymbol{w}}(\boldsymbol{x}_i)\right) + \epsilon||\nabla_x f||_1$$

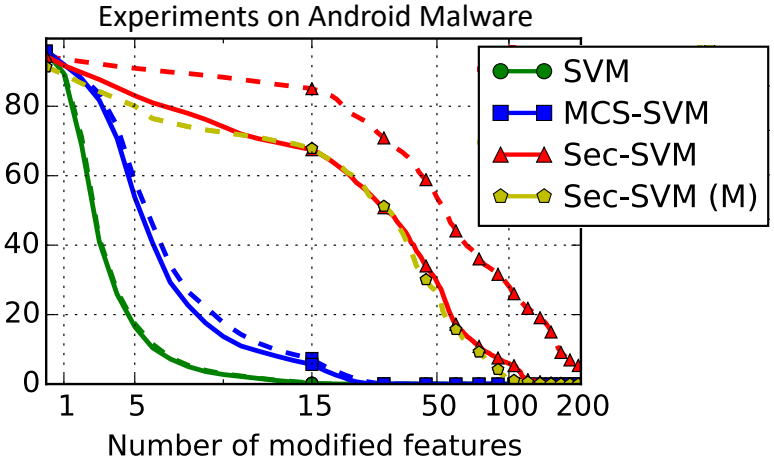
dual norm of the perturbation

$||\nabla_x f||_1 = ||\boldsymbol{w}||_1$

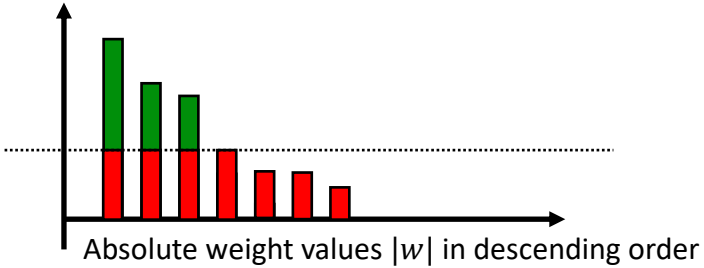
Results on *Adversarial* Android Malware

- **Infinity-norm regularization** is the optimal regularizer against **sparse evasion attacks**
  - Sparse evasion attacks penalize $||\boldsymbol{\delta}||_1$ promoting the manipulation of only few features

Sec-SVM $\min_{w,b} \|w\|_\infty + C\sum_i \max\left(0, 1 - y_i f(x_i)\right), \quad \|w\|_\infty = \max_{i=1,\ldots,d} |w_i|$



Why? It bounds the maximum weight absolute values!

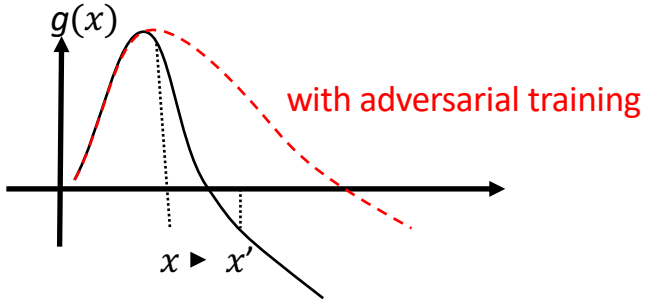


Adversarial Training and Regularization

- Adversarial training can also be seen as a form of regularization, which penalizes the (dual) norm of the input gradients $\epsilon||\nabla_x \ell||_q$

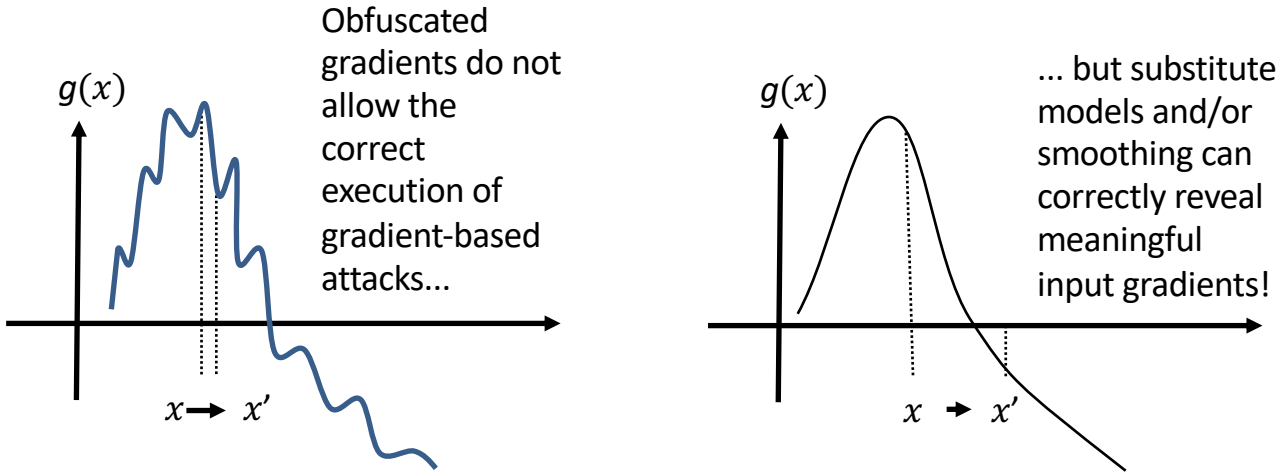
- Known as double backprop or gradient/Jacobian regularization
  - see, e.g., *Simon-Gabriel et al., Adversarial vulnerability of neural networks increases with input dimension, ArXiv 2018*; and *Lyu et al., A unified gradient regularization family for adversarial examples, ICDM 2015.*

Take-home message: the net effect of these techniques is to make the prediction function of the classifier smoother



Ineffective Defenses: Obfuscated Gradients

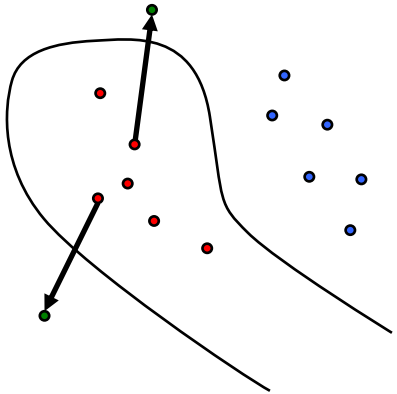
- Work by Carlini & Wagner (SP' 17) and Athalye et al. (ICML '18) has shown that
  - some recently-proposed defenses rely on obfuscated / masked gradients, and
  - they can be circumvented



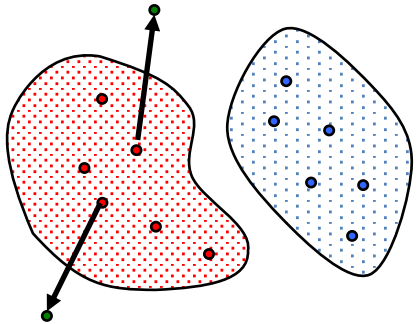
Countering Evasion:
Detecting & Rejecting Adversarial Examples

Detecting & Rejecting Adversarial Examples

- Adversarial examples tend to occur in *blind spots*
  - Regions far from training data that are anyway assigned to 'legitimate' classes

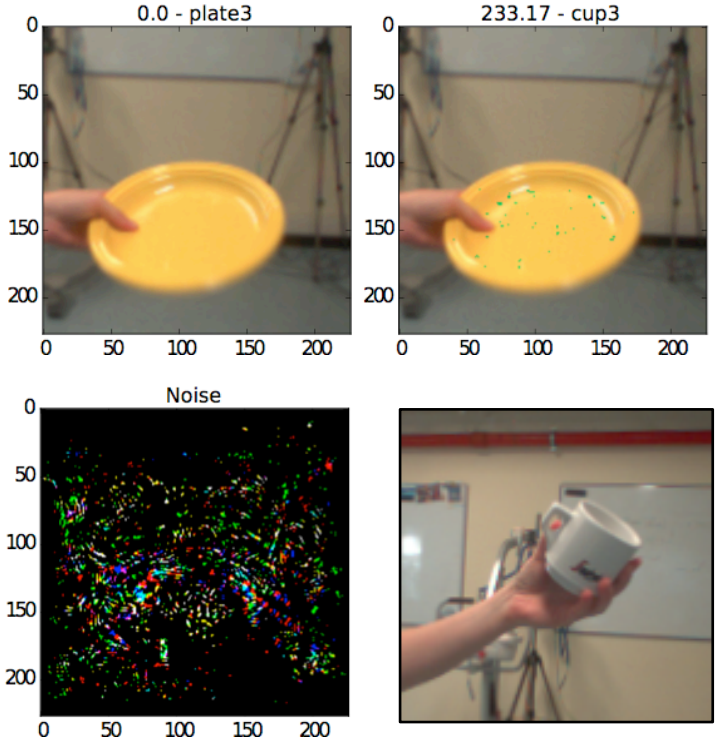


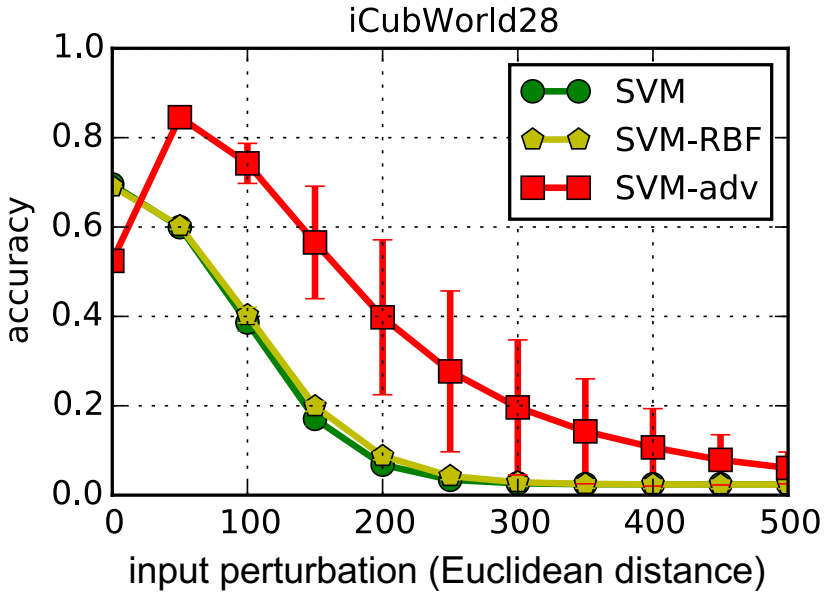
blind-spot evasion
(not even required to mimic the target class)



rejection of adversarial examples through enclosing of legitimate classes

Detecting & Rejecting Adversarial Examples





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

Adversarial Examples against Machine Learning

Web Demo

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

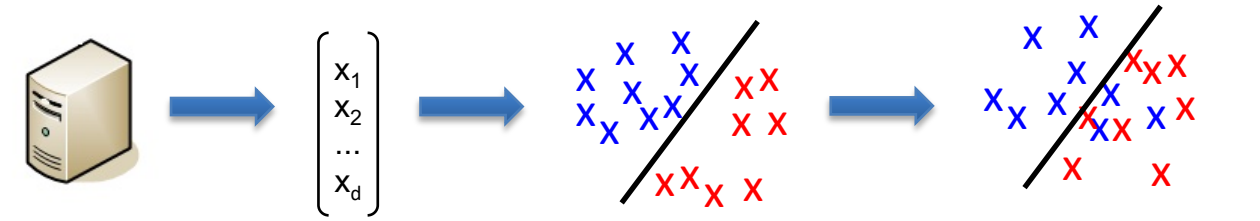
EU H2020 Project ALOHA

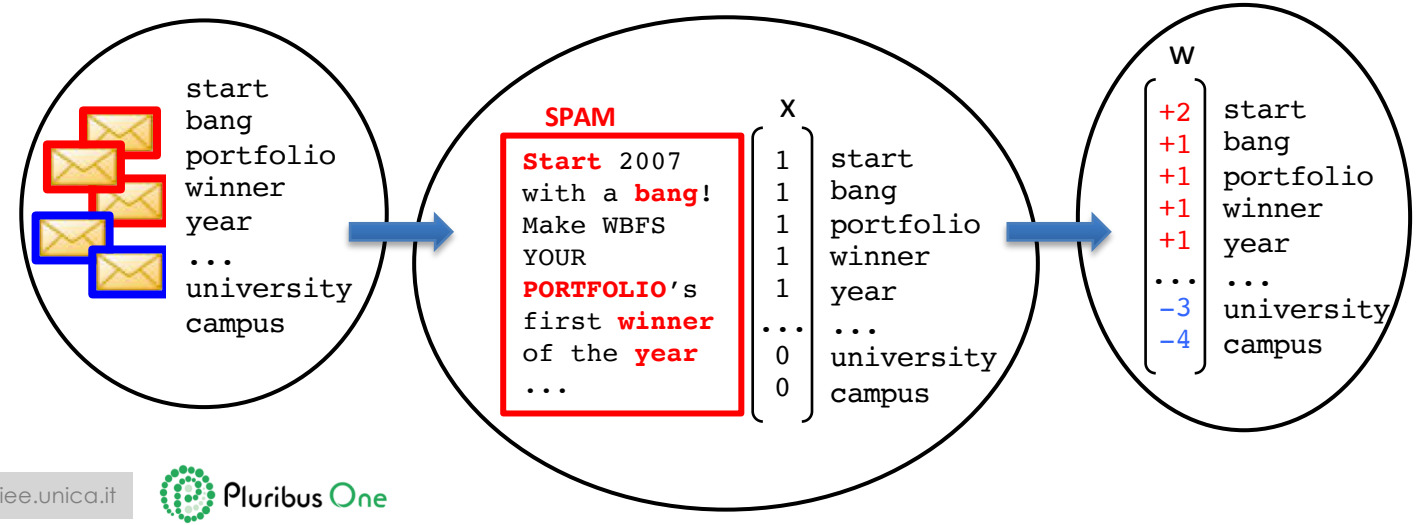
- **ALOHA** – software framework for runtime-Adaptive and secure deep Learning On Heterogeneous Architectures
- **Project goal**: to facilitate implementation of deep learning algorithms on heterogeneous low-energy computing platforms
- **Project website:** www.aloha-h2020.eu
- *Pluribus One* is in charge of evaluating and improving security of deep learning algorithms under attack

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No. 780788

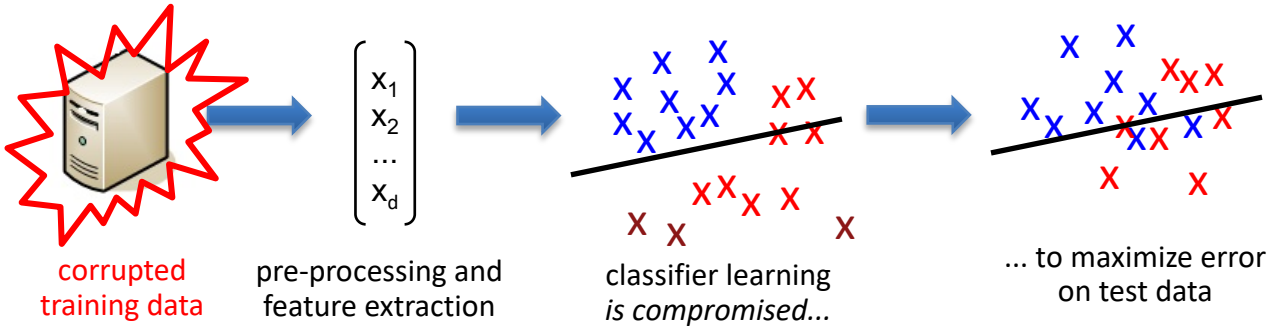
Poisoning Machine Learning

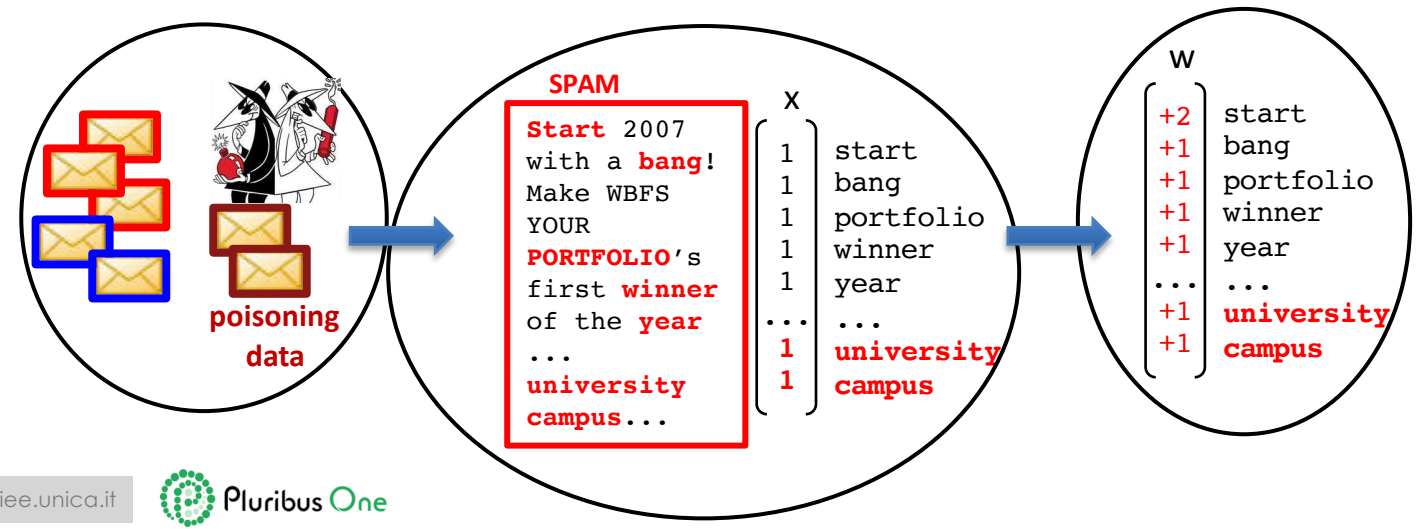
Poisoning Machine Learning





Poisoning Machine Learning

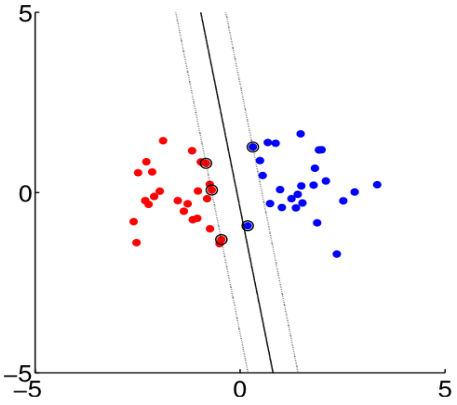




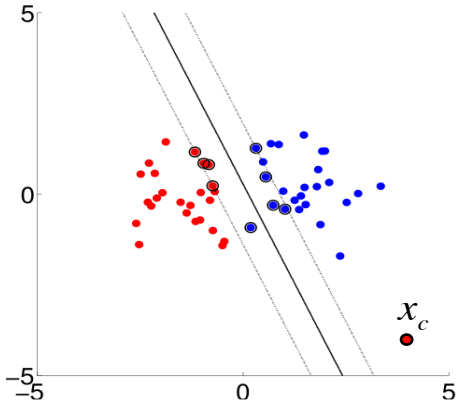
Poisoning Attacks against Machine Learning

- **Goal**: to maximize classification error
- **Knowledge:** perfect / white-box attack
- **Capability:** injecting poisoning samples into TR
- **Strategy:** find an *optimal* attack point x_c in TR that maximizes classification error

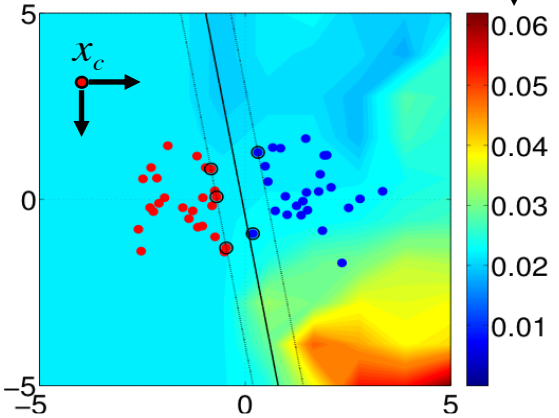
classification error = 0.022



classification error = 0.039



classification error as a function of x_c



[Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012]

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:

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

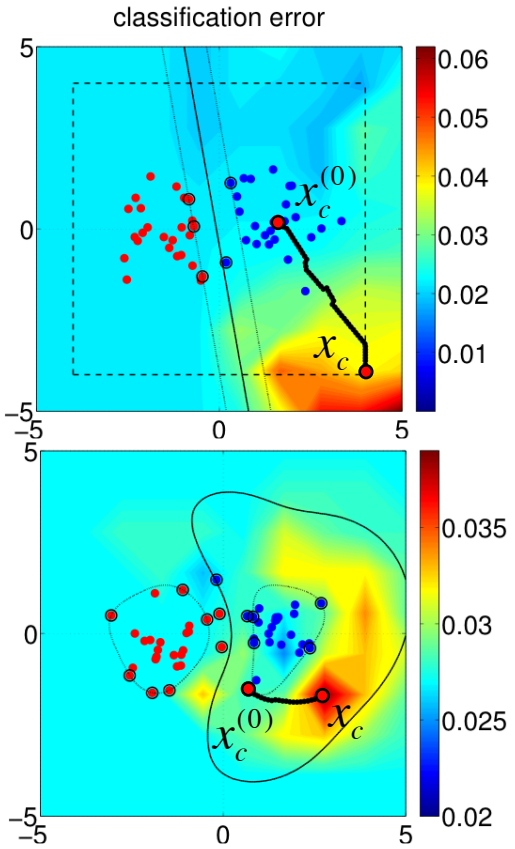
$$\text{s.t. } f^* = \text{argmin}_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + \mathrm{C}\sum_{i=1}^{n} \max(0, 1 - y_i f(\boldsymbol{x}_i)) + \mathrm{C}\max(0, 1 - y_c f(\boldsymbol{x}_c))$$

[Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012]
[Xiao, Biggio, Roli et al., Is feature selection secure against training data poisoning? ICML, 2015]
[Munoz-Gonzalez, Biggio, Roli et al., Towards poisoning of deep learning..., AISec 2017]

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.

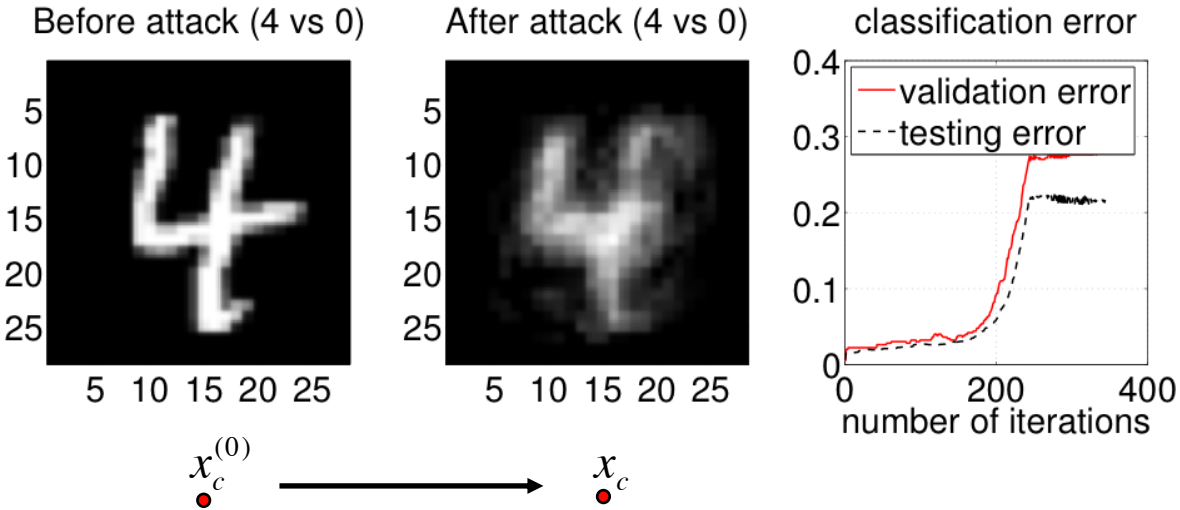
$$\nabla_{\boldsymbol{x}_c}\mathcal{A} = -\boldsymbol{y}_k^\top \frac{\partial \boldsymbol{k}_{kc}}{\partial \boldsymbol{x}_c}\alpha_c + \boldsymbol{y}_k^\top \underbrace{[\mathbf{K}_{ks} \quad \mathbf{1}]}_{k\times s+1} \underbrace{\begin{bmatrix} \boldsymbol{K}_{ss} & \mathbf{1} \\ \mathbf{1}^\top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



[Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012]
[Xiao, Biggio, Roli et al., Is feature selection secure against training data poisoning? ICML, 2015]

Experiments on MNIST digits
Single-point attack

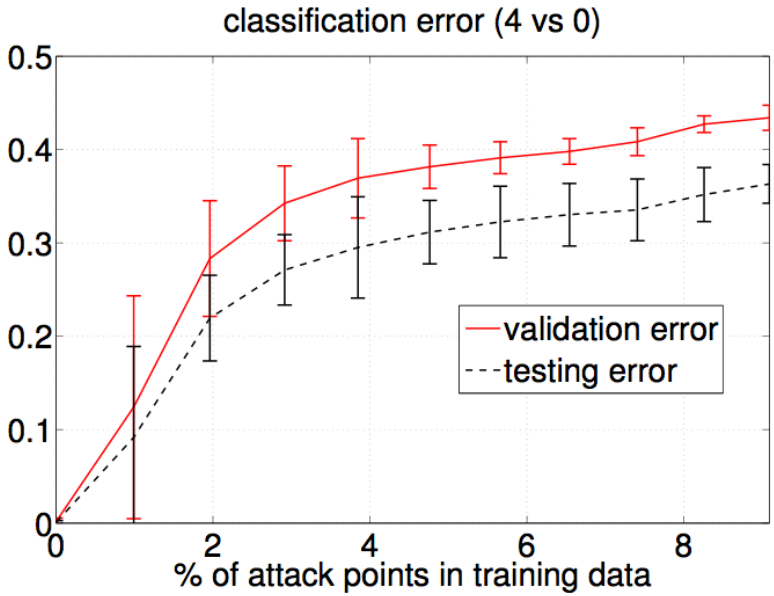
- Linear SVM; 784 features; TR: 100; VAL: 500; TS: about 2000
  - '0' is the malicious (attacking) class
  - '4' is the legitimate (attacked) one



[Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012]

Experiments on MNIST digits
Multiple-point attack

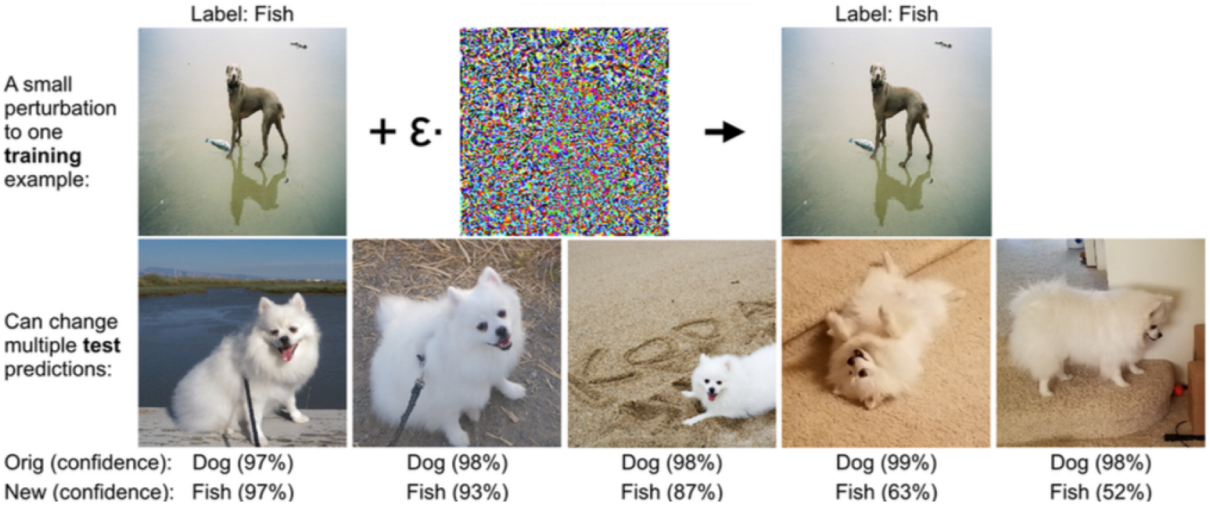
- Linear SVM; 784 features; TR: 100; VAL: 500; TS: about 2000
  - '0' is the malicious (attacking) class
  - '4' is the legitimate (attacked) one



[Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012]

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

- Solving the poisoning problem without exploiting KKT conditions (back-gradient)
  - *Muñoz-González, Biggio, Roli et al.*, AISec 2017 https://arxiv.org/abs/1708.08689

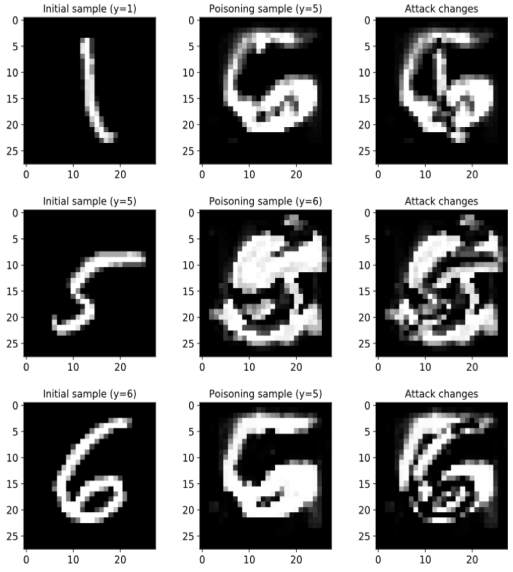


Figure 6: Poisoning samples targeting the LR.

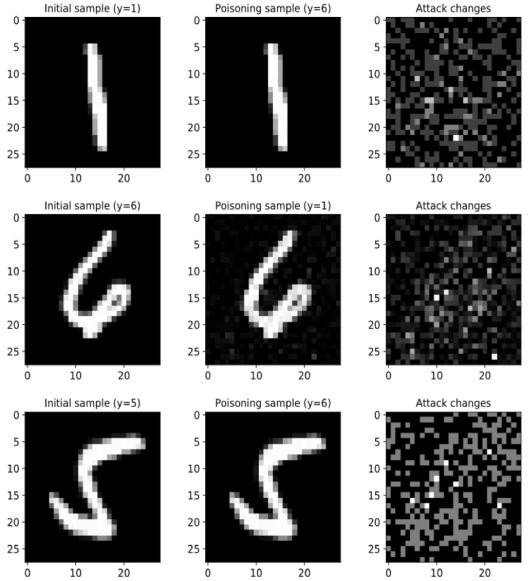
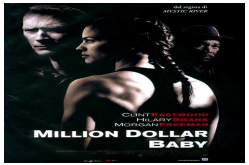


Figure 5: Poisoning samples targeting the CNN.

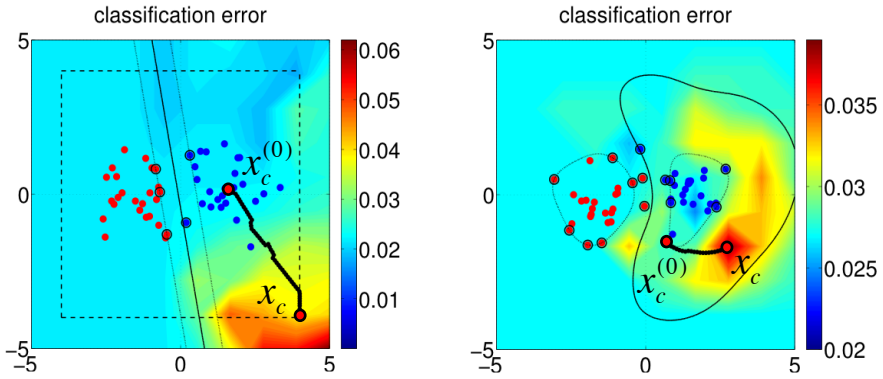
Countering Poisoning Attacks



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



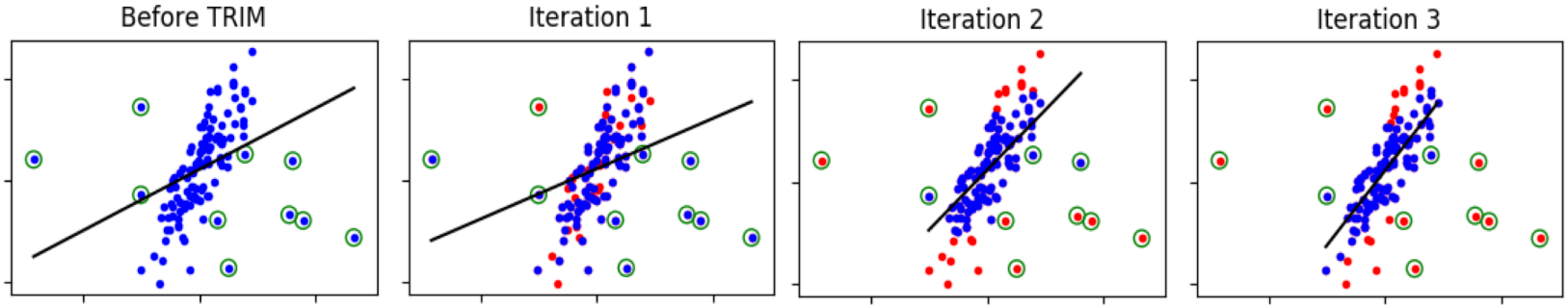
- Two main strategies for countering this threat
  1. **Data sanitization:** *remove* poisoning samples from training data
     - Bagging for fighting poisoning attacks
     - Reject-On-Negative-Impact (RONI) defense
  2. **Robust Learning:** learning algorithms that are robust in the presence of poisoning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

Robust Regression with TRIM

- TRIM learns the model by retaining only training points with the smallest residuals

$$\underset{w,b,I}{\operatorname{argmin}}\, L(w,b,I) = \frac{1}{|I|}\sum_{i\in I}(f(\boldsymbol{x}_i) - y_i)^2 + \lambda\Omega(\boldsymbol{w})$$

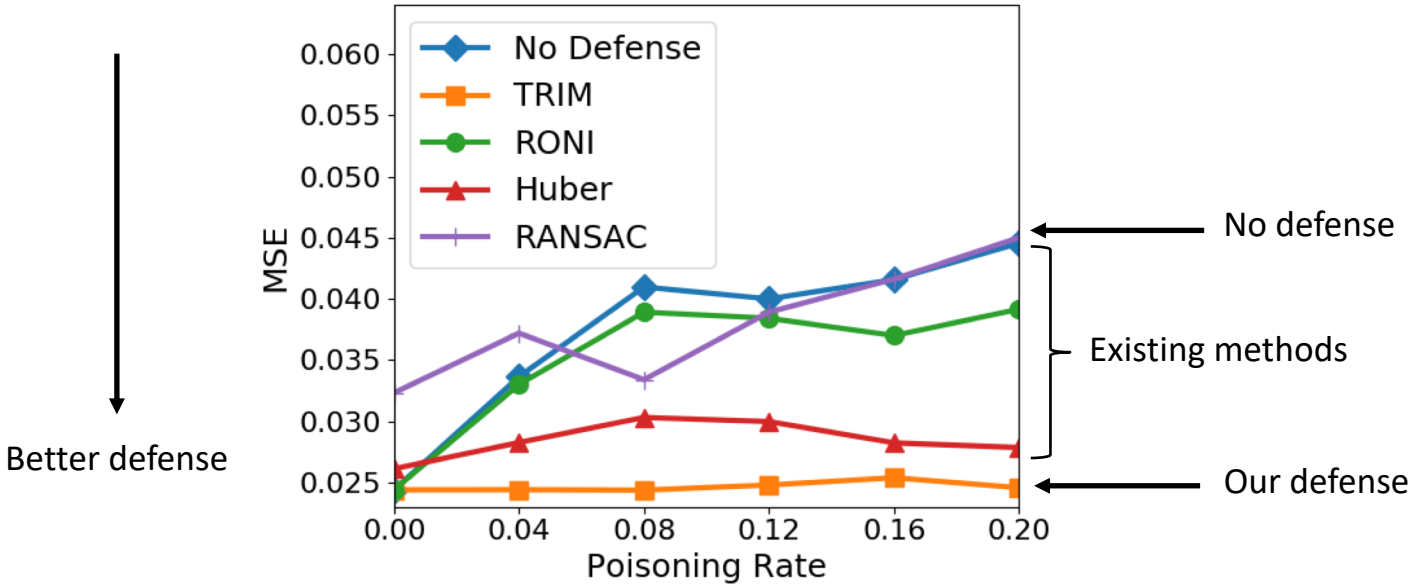
$$N = (1+\alpha)n, \qquad I \subset [1,\dots,N], \qquad |I| = n$$



[Jagielski, Biggio et al., IEEE Symp. Security and Privacy, 2018]

Experiments with TRIM (Loan Dataset)

- TRIM MSE is within 1% of original model MSE



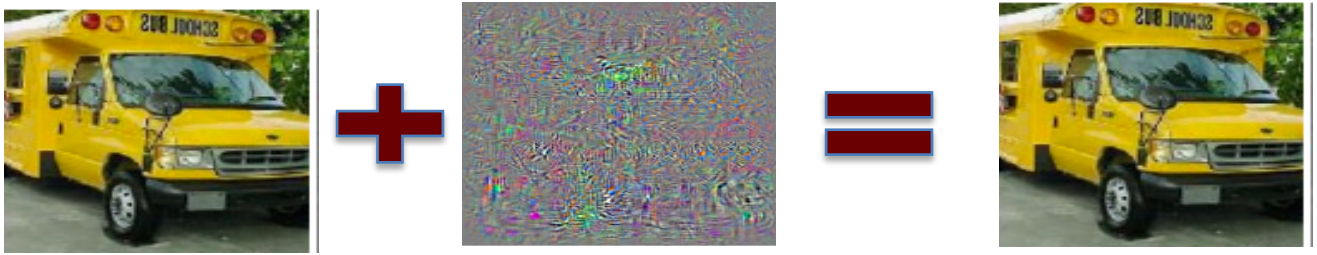
[Jagielski, Biggio et al., IEEE Symp. Security and Privacy, 2018]

Are Adversarial Examples a Real Security Threat?

World Is Not Digital…

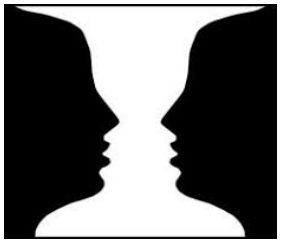
- *….**Previous** cases of **adversarial examples** have common characteristic: the **adversary** is able to **precisely control** the **digital representation** of the input to the machine learning tools…..*

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



School Bus (*x*) **Adversarial Noise (*r*)** **Ostrich Struthio Camelus**

Do Adversarial Examples exist in the Physical World?

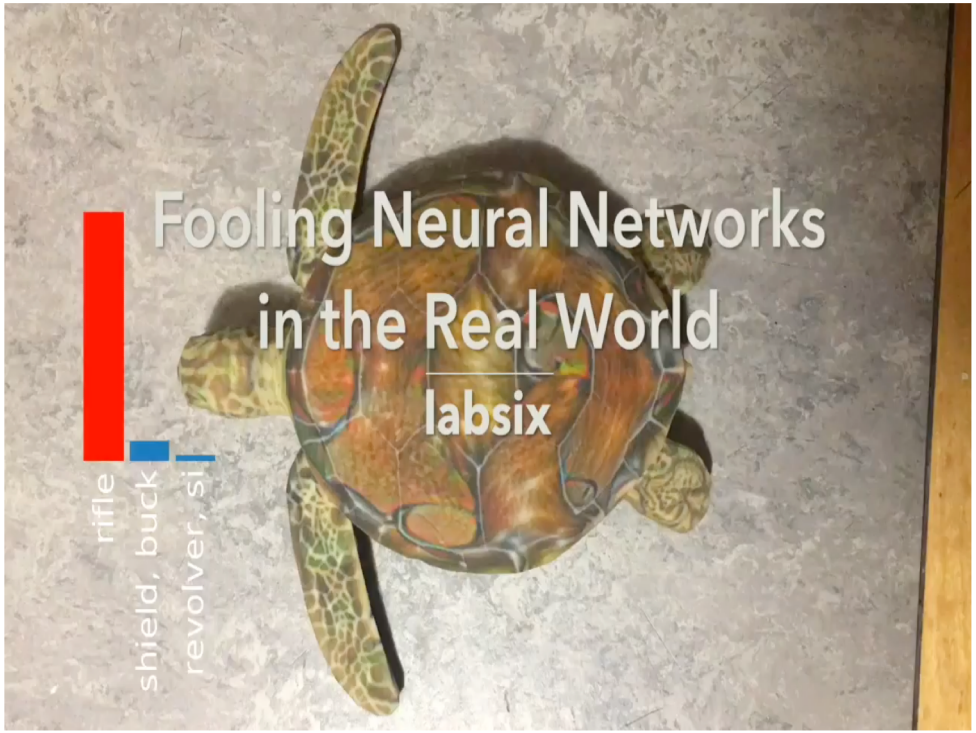


Adversarial Images in the Physical World

[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 Turtle…

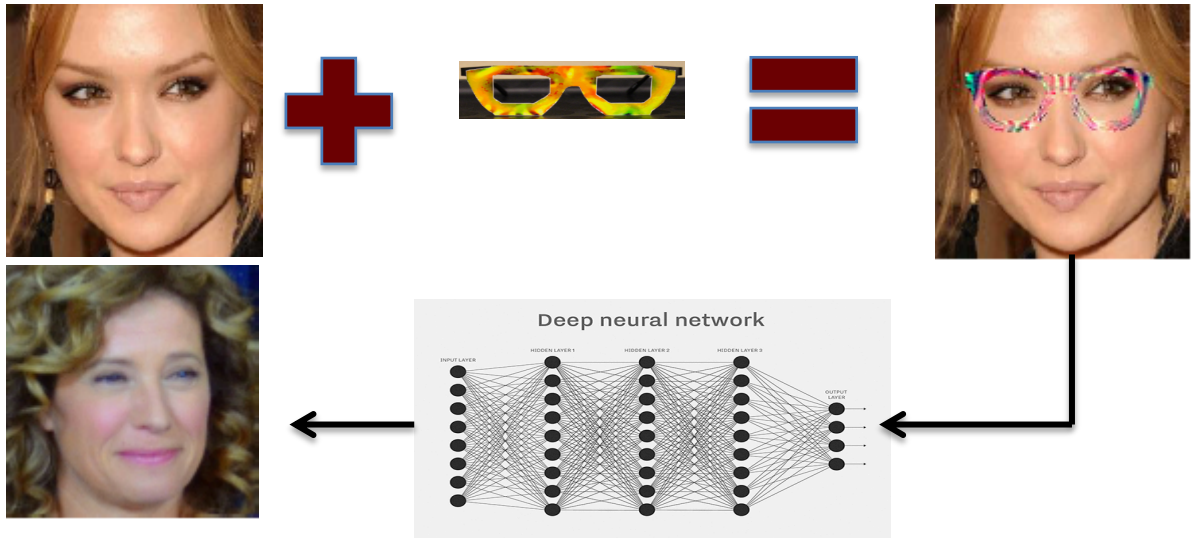


Adversarial Glasses

[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...

Yes, we should…

Synthesizing Robust Adversarial Examples

Anish Athalye
OpenAI, MIT

Ilya Sutskever
OpenAI

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

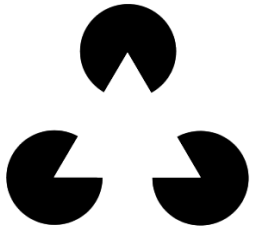
Is a Real Security Threat?

- Adversarial examples can exist in the physical world, we can fabricate concrete adversarial objects (glasses, road signs, etc.)
- But the effectiveness of attacks carried out by adversarial objects is still to be investigated with large scale experiments in realistic security scenarios

- Gilmer et al. (2018) have recently discussed the realism of security threat caused by adversarial examples, pointing out that it should be investigated more carefully
  - Are indistinguishable adversarial examples a real security threat ?
  - For which real security scenarios adversarial examples are the best attack vector? Better than attacking components outside the machine learning component
  - …

[Justin Gilmer et al., Motivating the Rules of the Game for Adversarial Example Research, https://arxiv.org/abs/1807.06732]

Are Indistinguishable Perturbations a Real Security Threat?



Indistinguishable 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

*…There is a **torrent of work** that views increased robustness to **restricted perturbations** as making these models **more secure**. While not all of this work requires completely indistinguishable modications, many of the papers focus on specically on small modications, and the language in many suggests or implies that the **degree of perceptibility** of the **perturbations** is an important aspect of their **security risk**…*

[Justin Gilmer et al., Motivating the Rules of the Game for Adversarial Example Research, arXiv 2018]

Indistinguishable Adversarial Examples

The attacker can benefit by minimal perturbation of a legitimate input, she could use the attack for a longer period of time before it is detected

But minimal perturbation is a necessary constraint for the attacker?

Indistinguishable Adversarial Examples

Minimal perturbation is a necessary constraint for the attacker?

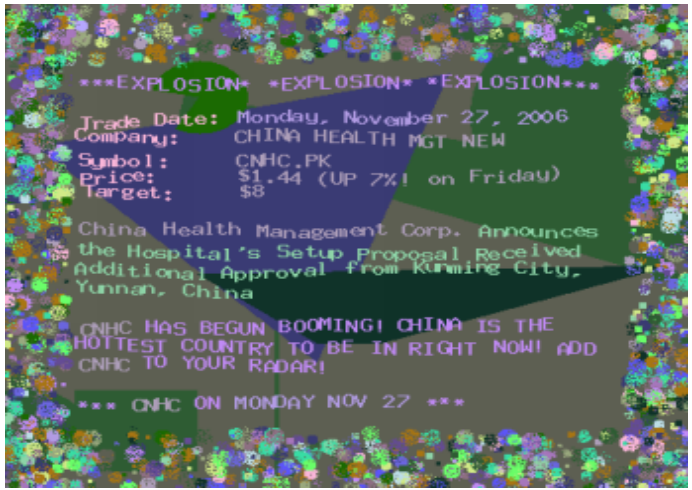
Indistinguishable Adversarial Examples

Minimal perturbation is a necessary constraint for the attacker?

[Flickr user \faungg". Stop? Accessed: 2018-7-18. Aug. 2014. url : https://www.flickr.com/photos/44534236@N00/15536855528/]

Attacks with content preservation

There are well known security applications where minimal perturbations and indistinguishability of adversarial inputs are not required at all…



Are Indistinguishable Perturbations a Real Security Threat?

*…At the time of writing, we were **unable** to find a **compelling example** that **required indistinguishability**…*

*To have the largest impact, we should both recast future adversarial example research as a **contribution** to **core machine learning** and develop new abstractions that capture **realistic threat models**.*

[Justin Gilmer et al., Motivating the Rules of the Game for Adversarial Example Research, arXiv 2018]

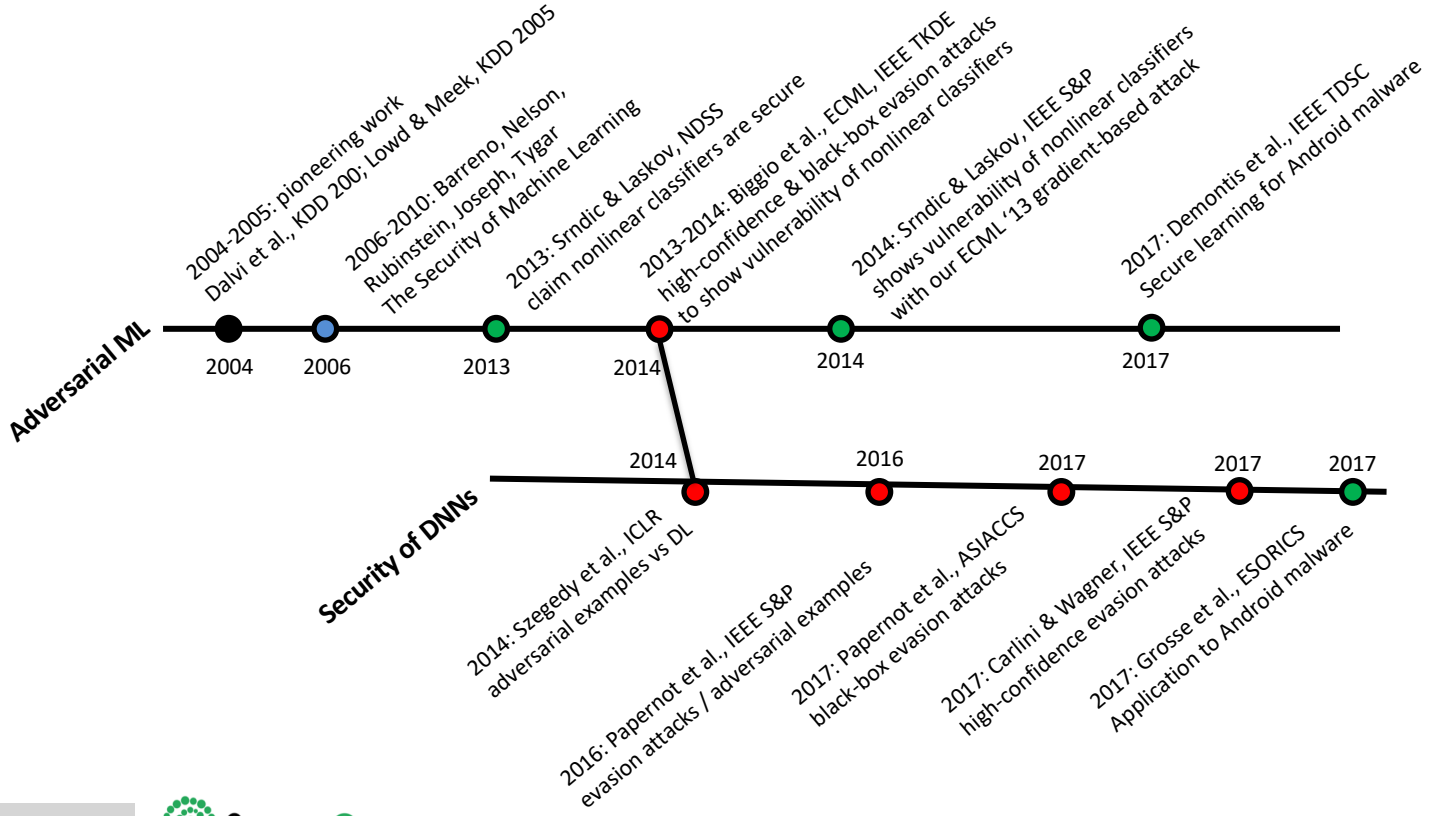
To Conclude…

This is a recent research field…

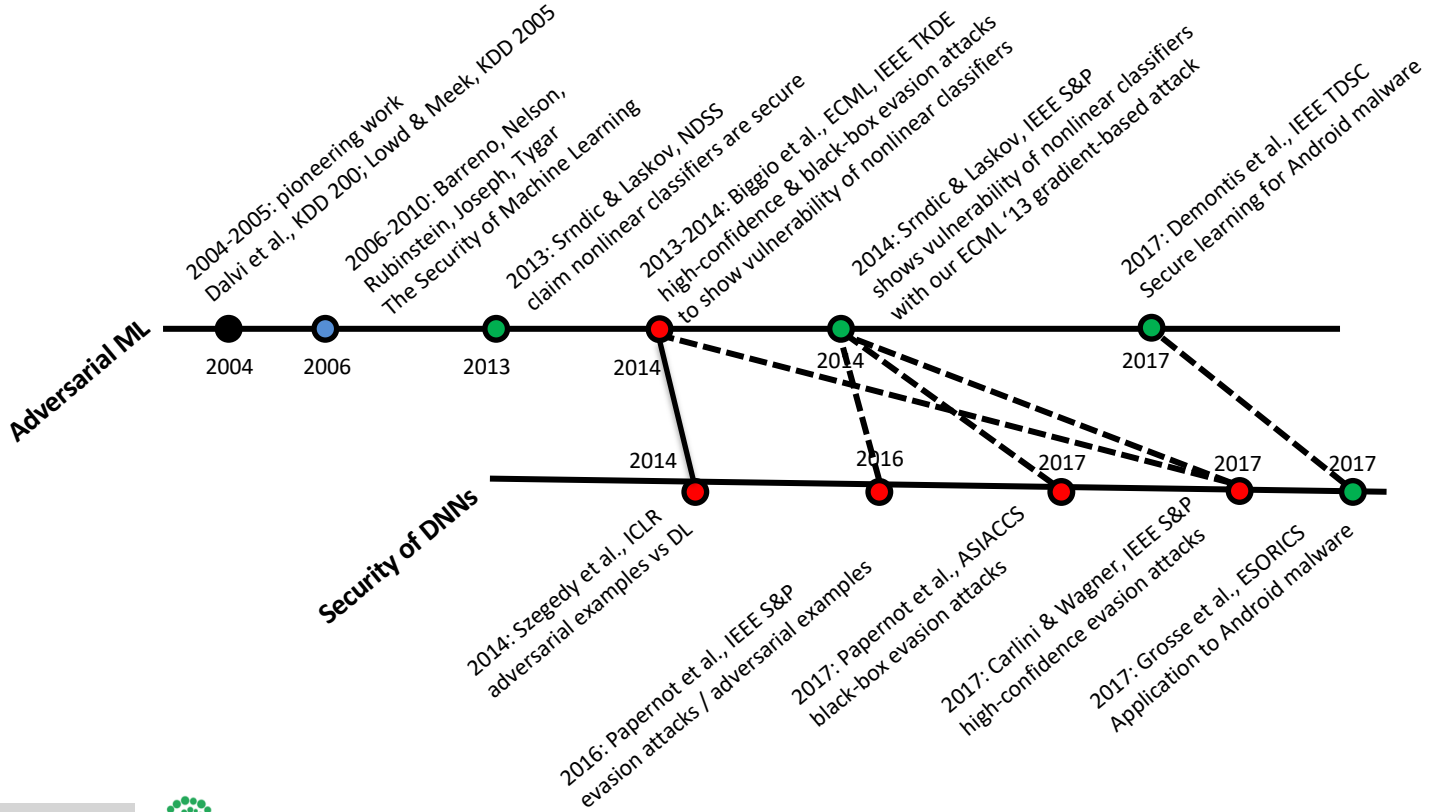
Dagstuhl Perspectives Workshop on
"Machine Learning in Computer Security"
Schloss Dagstuhl, Germany, Sept. 9th-14th, 2012

SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik

Timeline of Learning Security

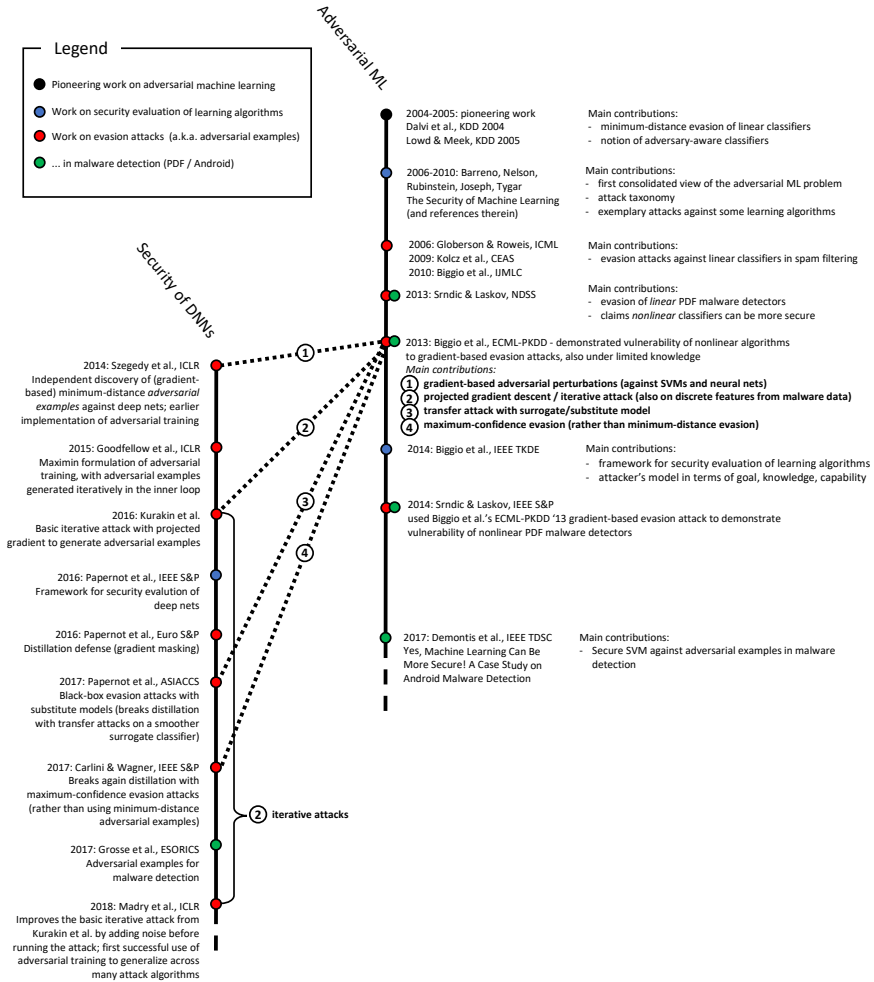


Timeline of Learning Security



Timeline of Learning Security

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



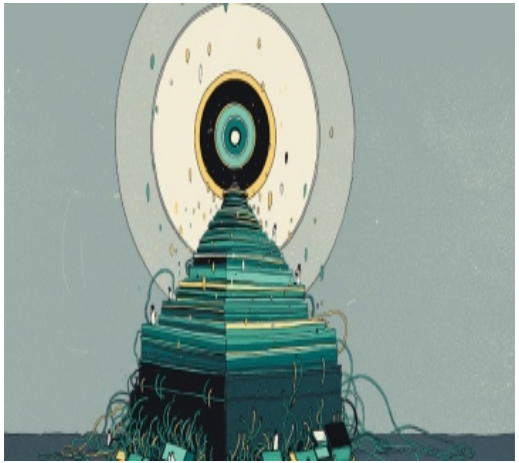
Black Swans to the Fore

[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?

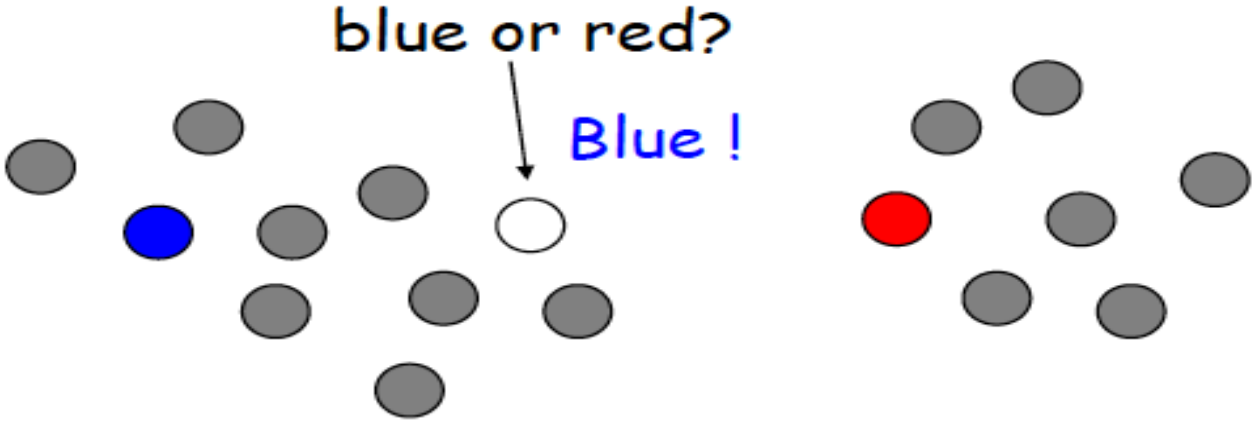
Why So Much Interest?

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

The smoothness assumption

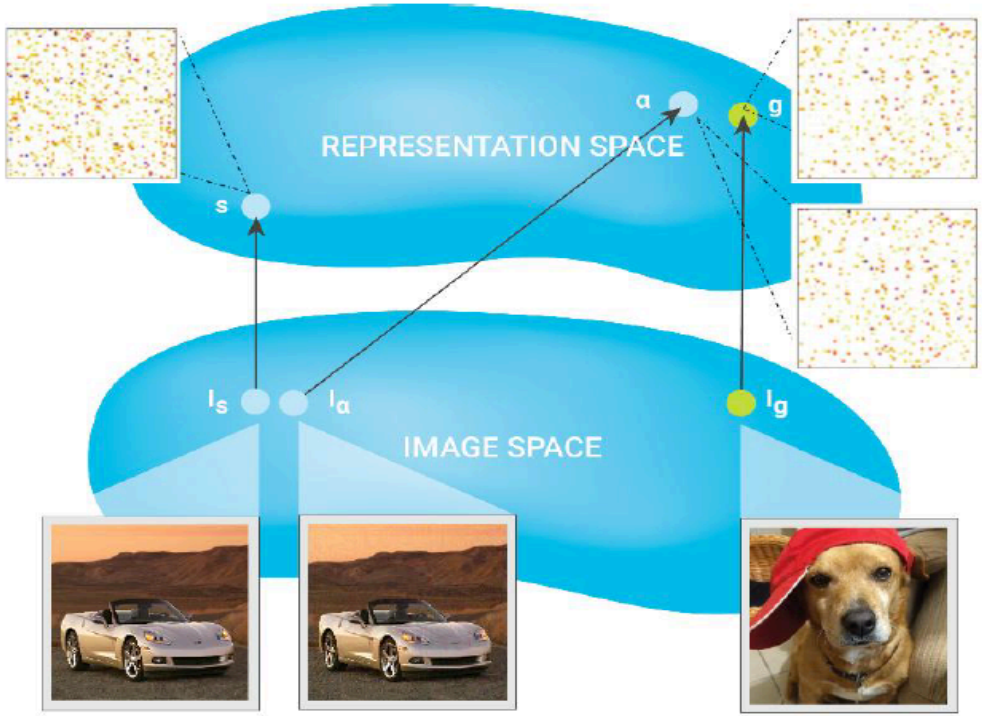


- *Points close to each other are more likely to share a label*

- This is generally assumed in supervised learning and yields a preference for geometrically simple decision boundaries.
- But deep networks seem to violate this assumption…

Instability Is One of the Fundamental Problems!

[S. Sabour at al., ICLR 2016]



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

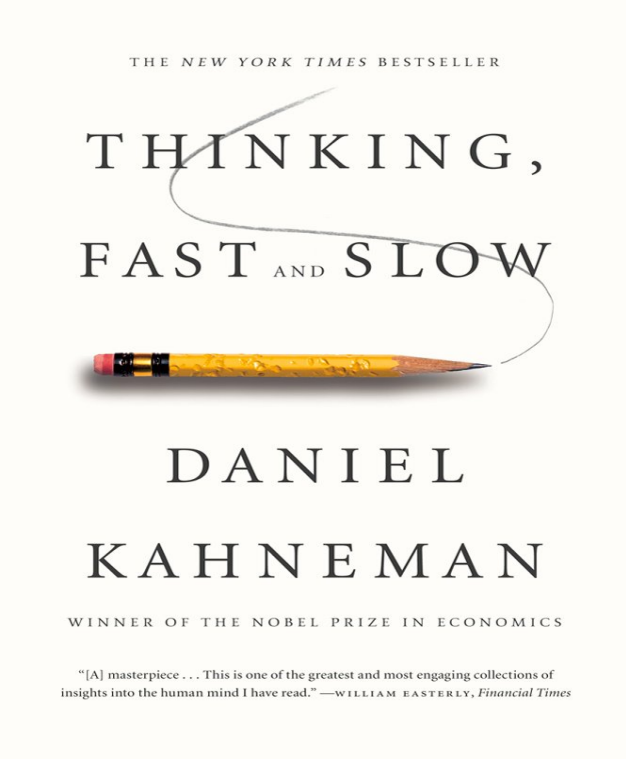
$$\begin{cases} bat+ball=\$1.10 \\ bat=ball+\$1.0 \end{cases}$$

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?

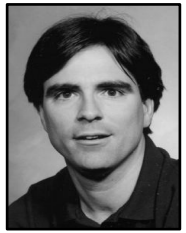
Learning Comes at a Price!

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)