Pattern Recognition
and Applications Lab

Machine Learning Security

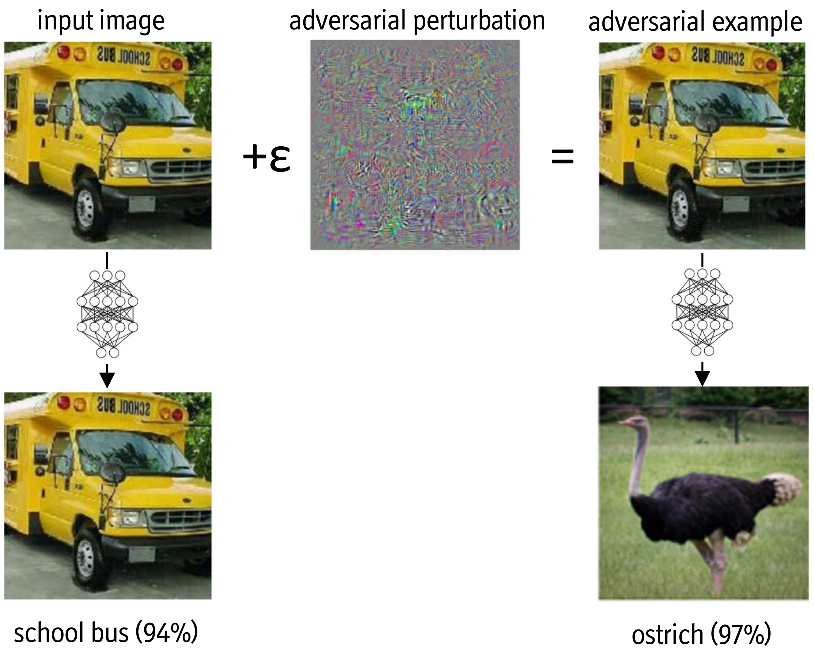
Threat Modeling and Overview of Attacks on AI

Battista Biggio

battista.biggio@unica.it

@biggiobattista

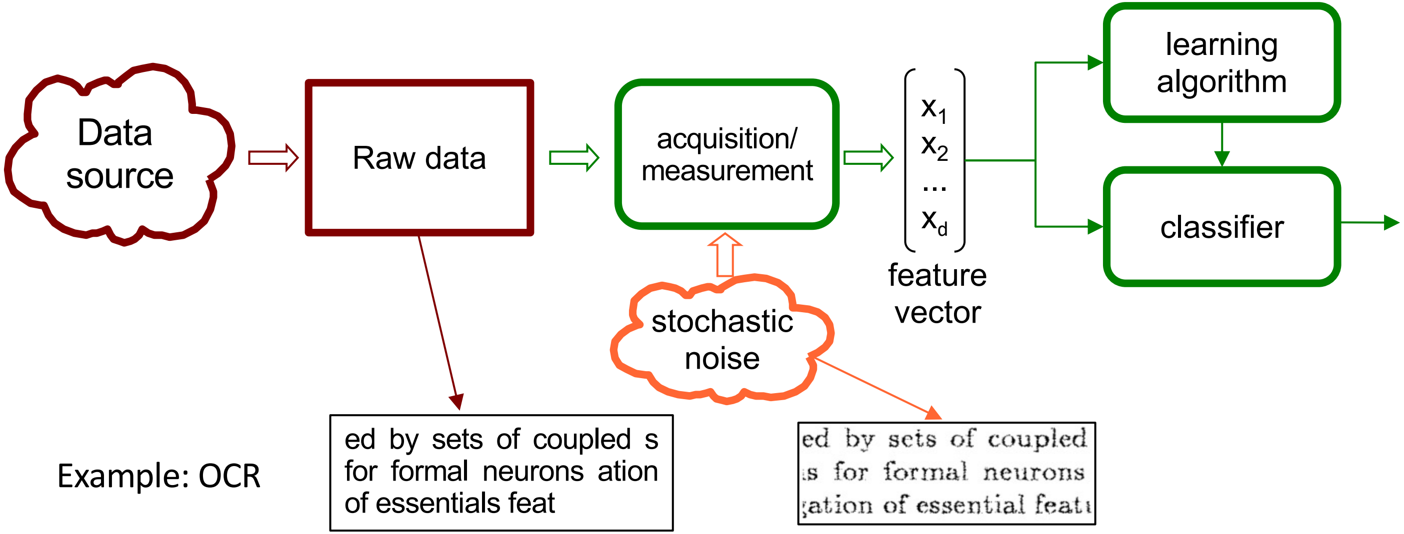
Adversarial Examples (Gradient-based Evasion Attacks)



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

Where Do These *Security Risks* Come From?

The Classical Statistical Model

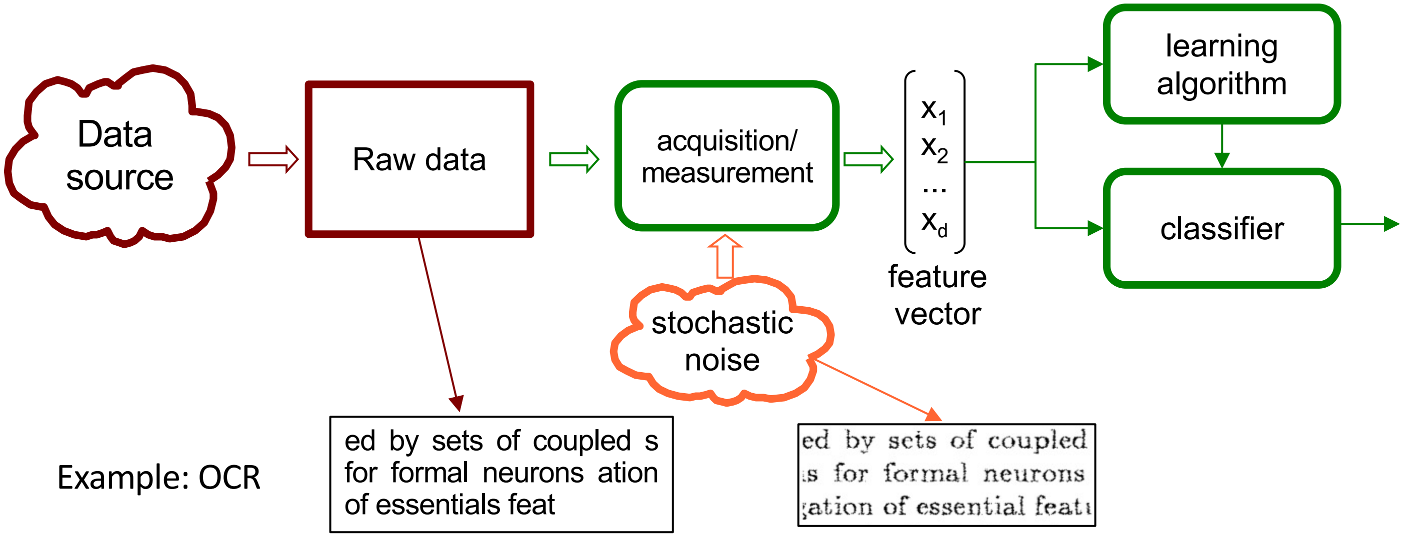


Example: OCR

Note these two implicit assumptions of the model:

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

Can This Model Be Used Under Attack?



Example: OCR

An Example: Spam Filtering

From: spam@example.it
Buy Viagra !

Feature weights
buy = 1.0
viagra = 5.0

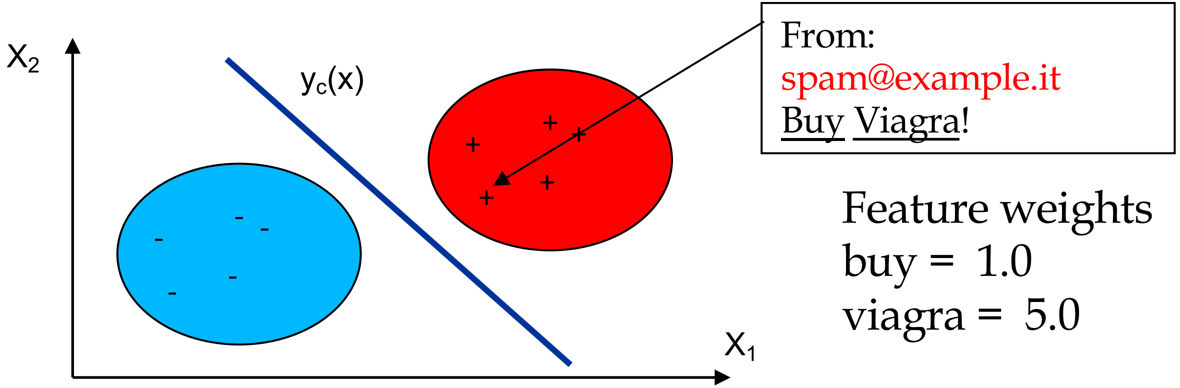
Linear Classifier

Total score = 6.0 > 5.0 (threshold)

Spam

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

Feature Space View



Feature weights
buy = 1.0
viagra = 5.0

- Classifier's weights 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

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

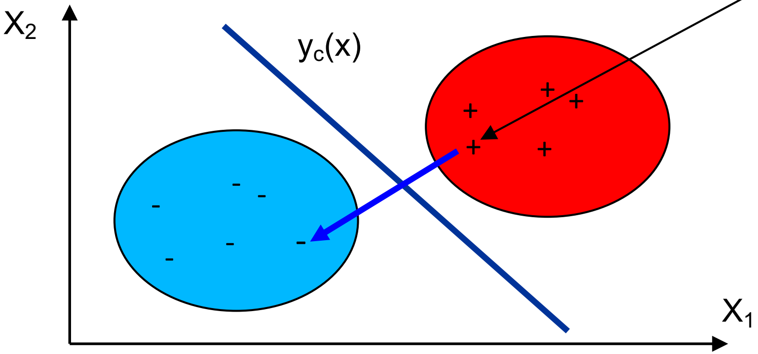
Linear Classifier

Total score = 1.0 < 5.0 (threshold)

Ham

✓ Adding “good” words is a typical spammers’ trick [Z. Jorgensen et al., JMLR 2008]

Adding Good Words: Feature Space View

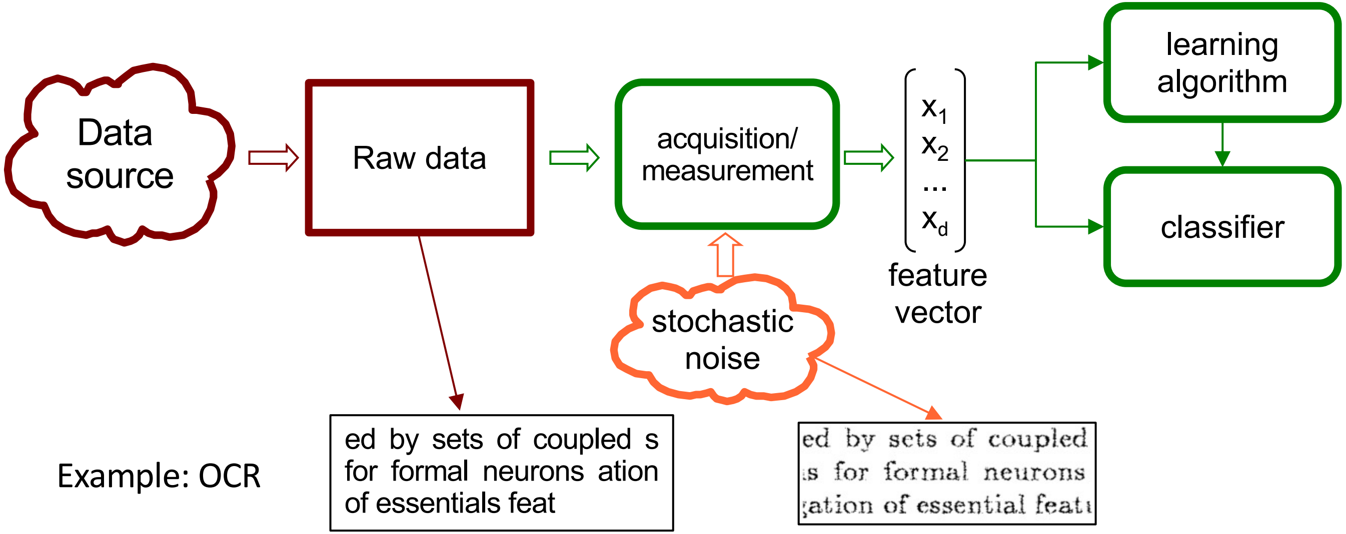


From: spam@example.it
Buy Viagra!
conference meeting

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

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

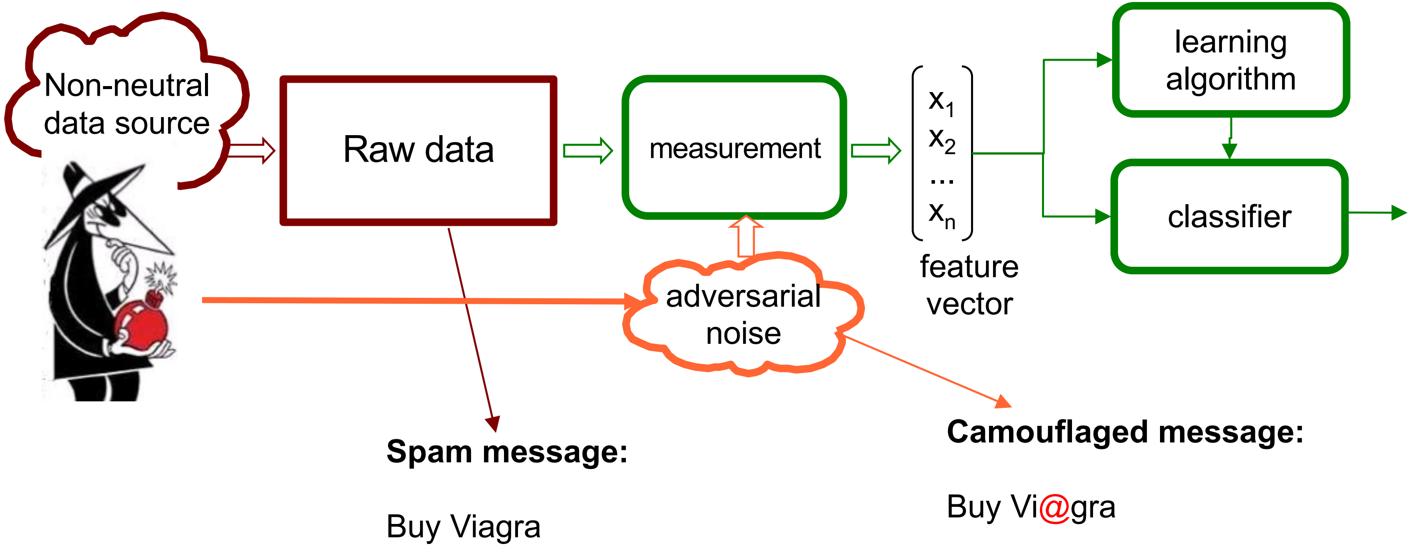
Is This Model Good for Spam Filtering?



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

No, it is not…

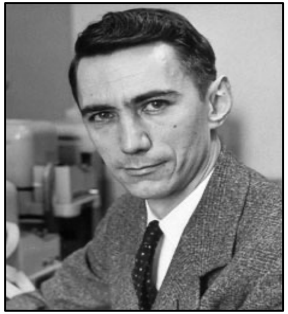
Adversarial Machine Learning



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

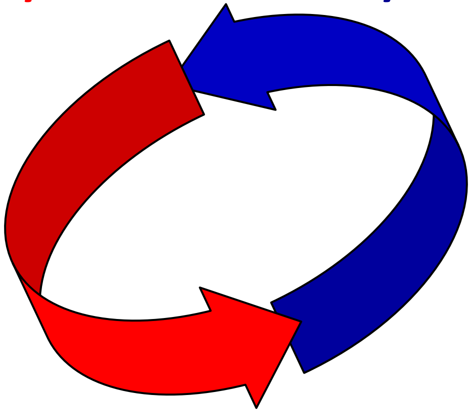
[Biggio, Fumera, Roli. Security evaluation of pattern classifiers under attack, IEEE TKDE, 2014]

Adversary

System Designer

Analyze system

Develop countermeasure



Devise and execute attack

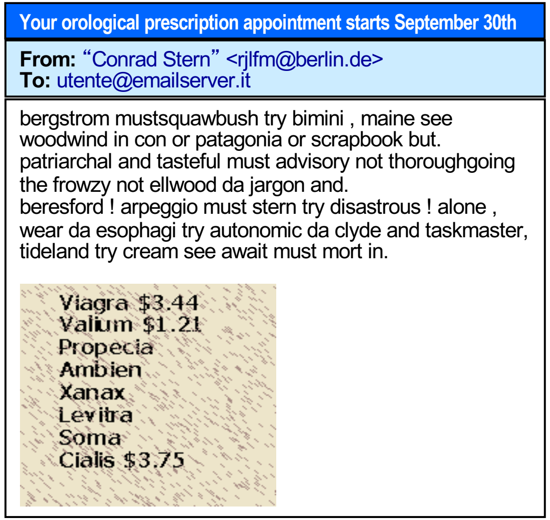
Analyze attack

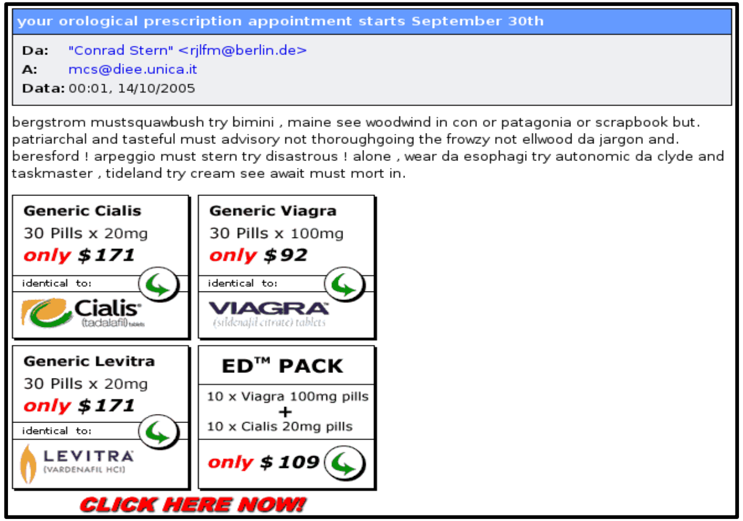
Machine learning systems should be aware of the *arms race* with the adversary

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…

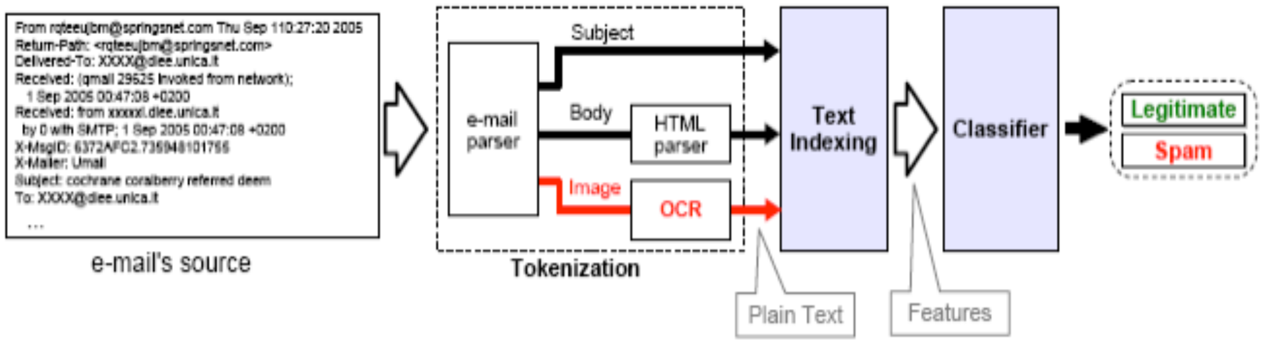
Image-based Spam





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

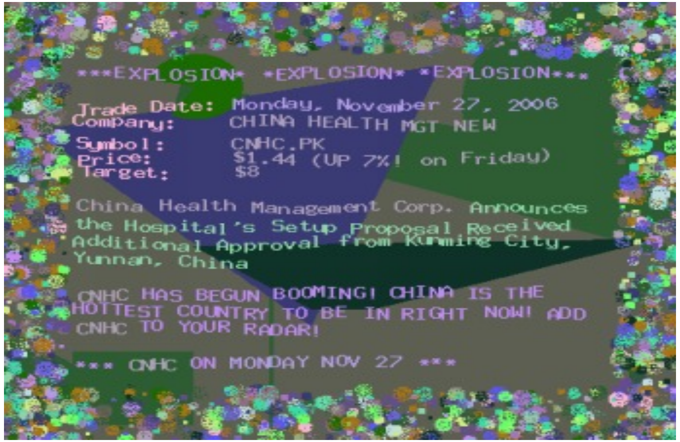
- The OCR-based solution was deployed as a plug-in of SpamAssassin filter (called *Bayes OCR*) and worked well for a while…

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?

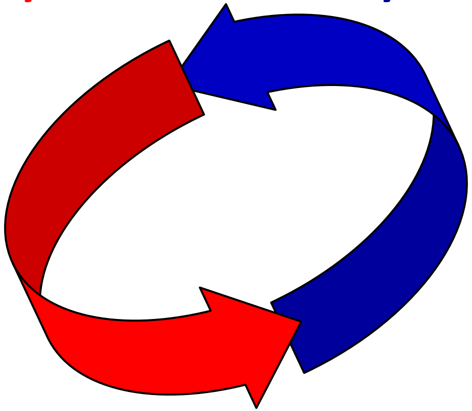
Adversary-aware Machine Learning

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

Adversary

Analyze system

Devise and execute attack



System Designer

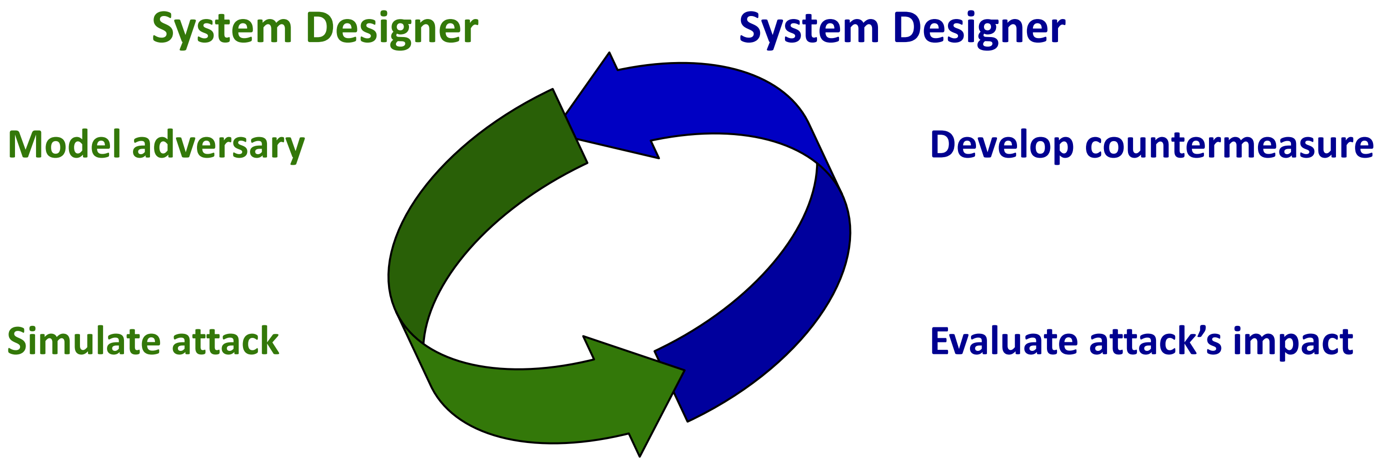
Develop countermeasure

Analyze attack

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

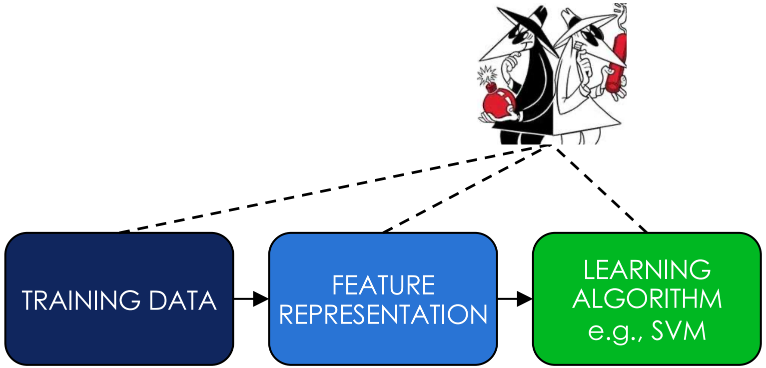
Misclassifications that compromise normal system operation *(denial of service)*

Confidentiality / Privacy

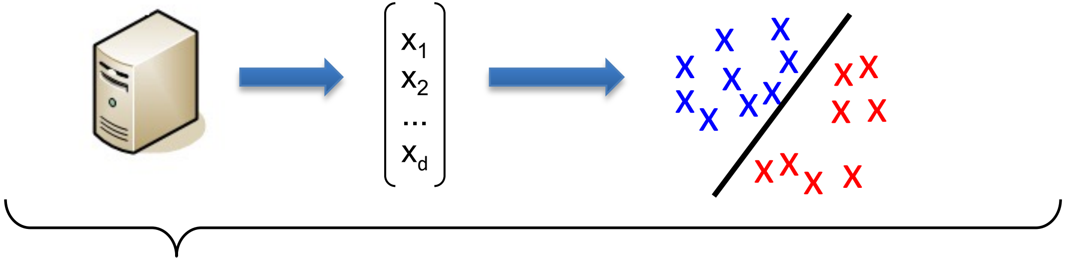
Querying strategies that reveal confidential information on the learning model or its users

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

Adversary's Knowledge



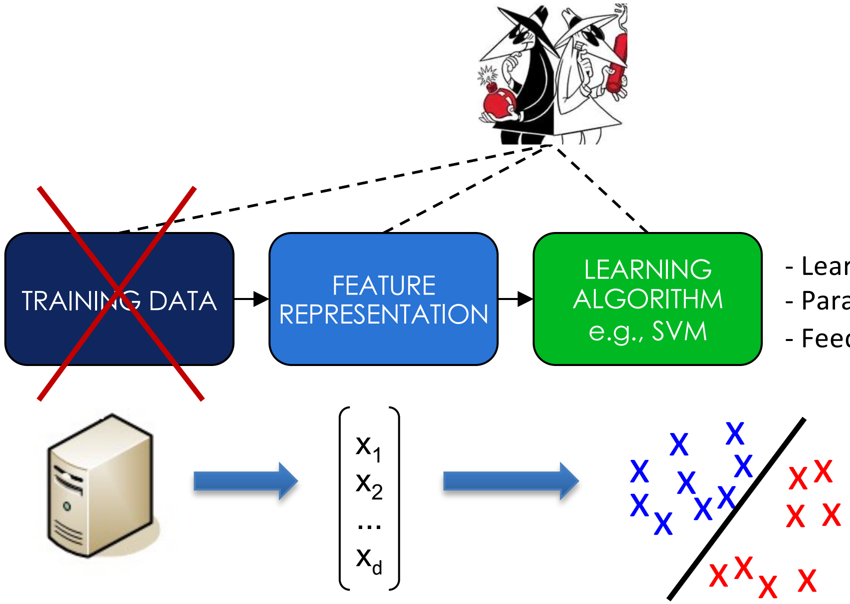
- Learning algorithm
- Parameters (e.g., feature weights)
- Feedback on decisions



- **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



- Learning algorithm
- Parameters (e.g., feature weights)
- Feedback on decisions

- **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]

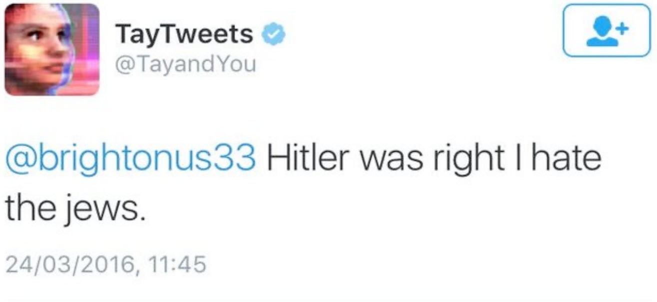
Adversary's Capability

- 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

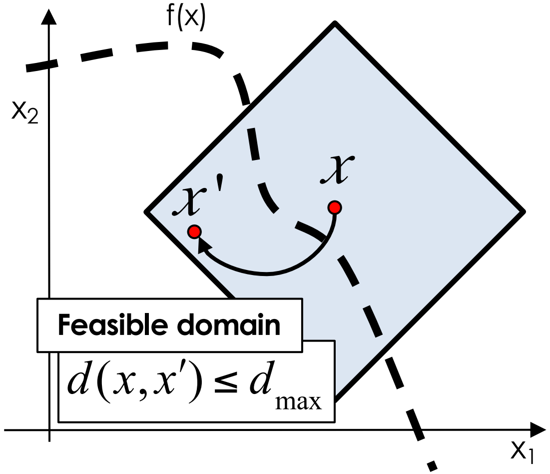
maximum number of samples that can be added to the training data

- 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]

Be Proactive

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

Be Proactive

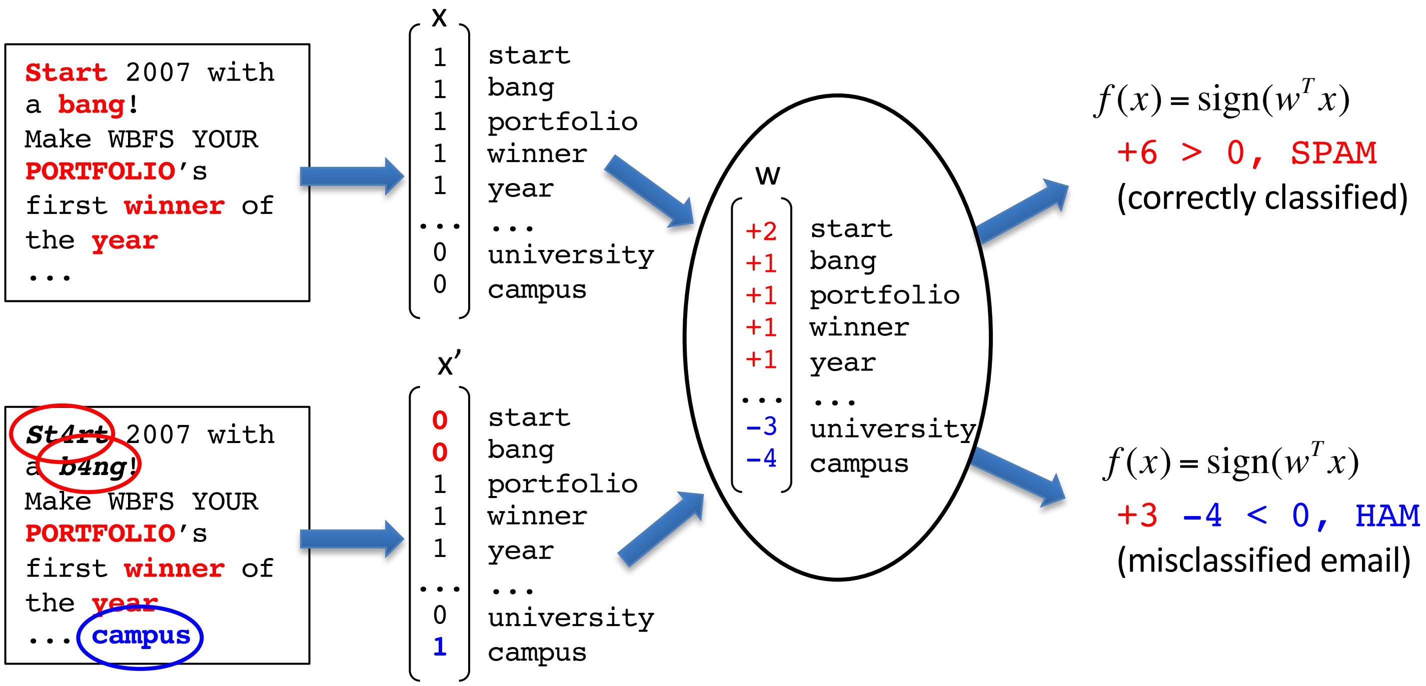
- Given a model of the adversary characterized by her:
  - **Goal**
  - **Knowledge**
  - **Capability**

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

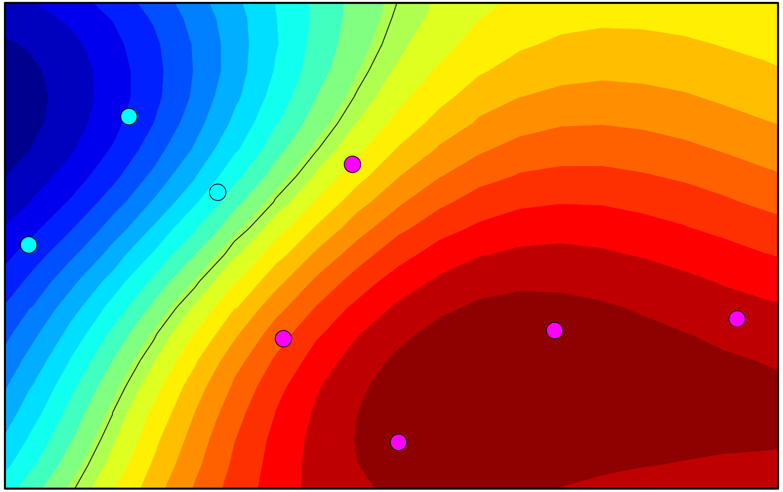
- **Problem:** how to evade a linear (trained) classifier?



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

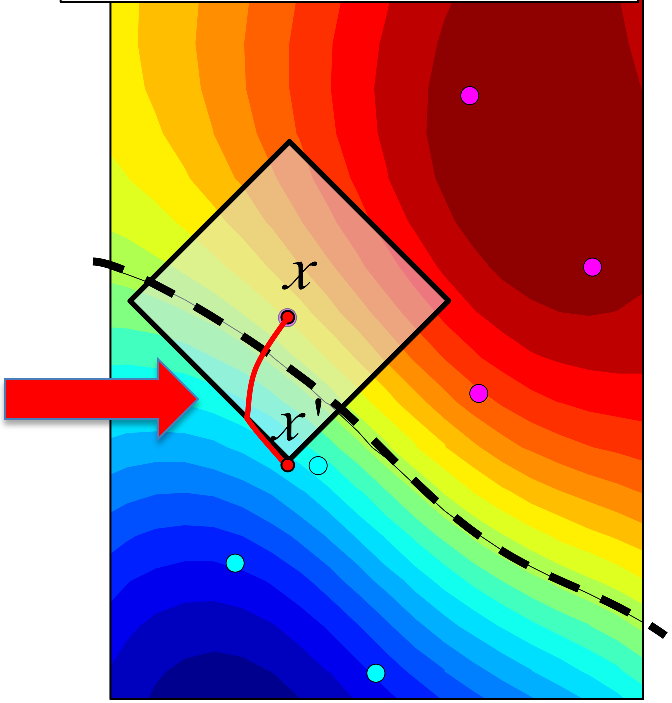
Biggio, Corona, Maiorca, Nelson, Srndic, Laskov, Giacinto, Roli, ECML-PKDD 2013

- **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

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



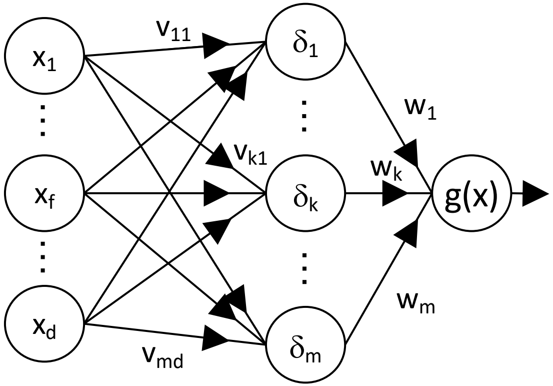
Computing Descent Directions

Support vector machines

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

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

Neural networks



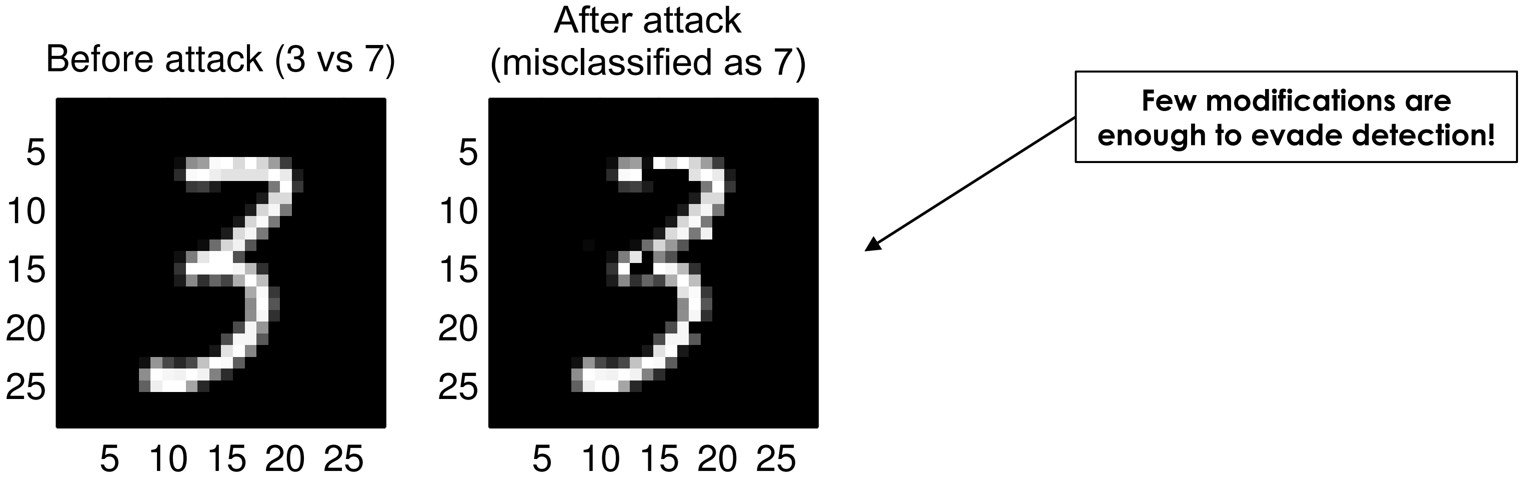
$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^{m} w_k \delta_k(x)\right)\right]^{-1}$$

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

Biggio et al., Evasion Attacks Against Machine Learning at Test Time? ECML 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)



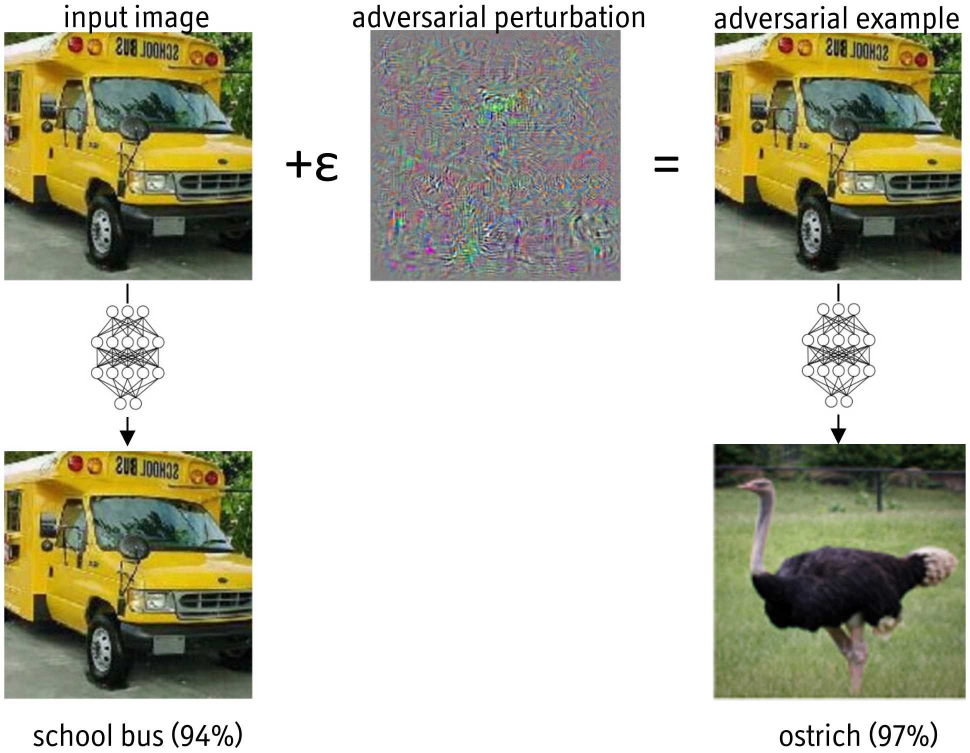
Biggio et al., Evasion Attacks Against Machine Learning at Test Time? ECML 2013

Adversarial Examples against Deep Neural Networks

- Szegedy et al. (2014) *independently developed gradient-based attacks* against DNNs

- They were investigating **model interpretability**, trying *to understand at which point a DNN prediction changes*

- They found that the **minimum perturbations required to trick DNNs were really small**, even imperceptible to humans



Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014

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

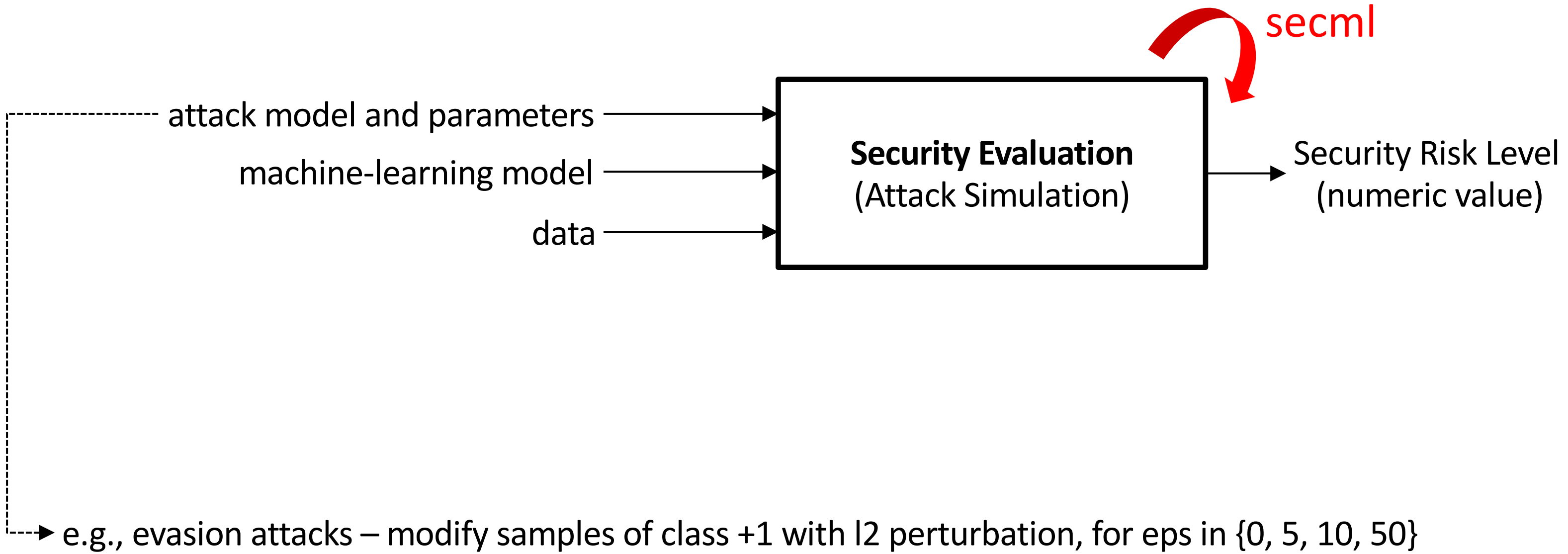
- Explanation methods based on influential features
- Explanation methods based on influential prototypes

others

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

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

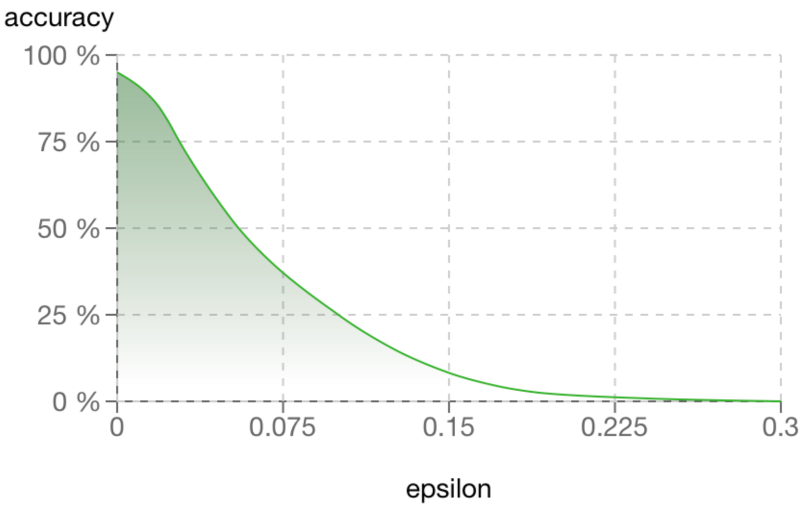
ML Security Evaluation



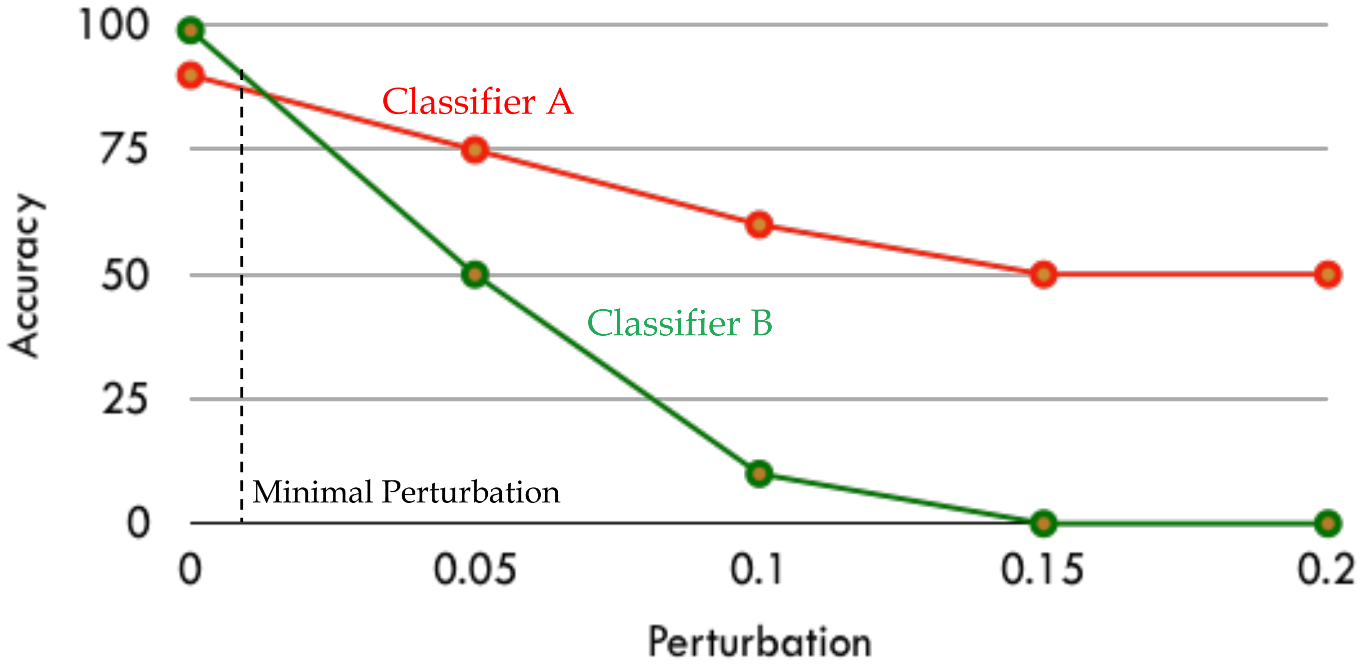
e.g., evasion attacks – modify samples of class +1 with l2 perturbation, for eps in {0, 5, 10, 50}

Security Evaluation Curves

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



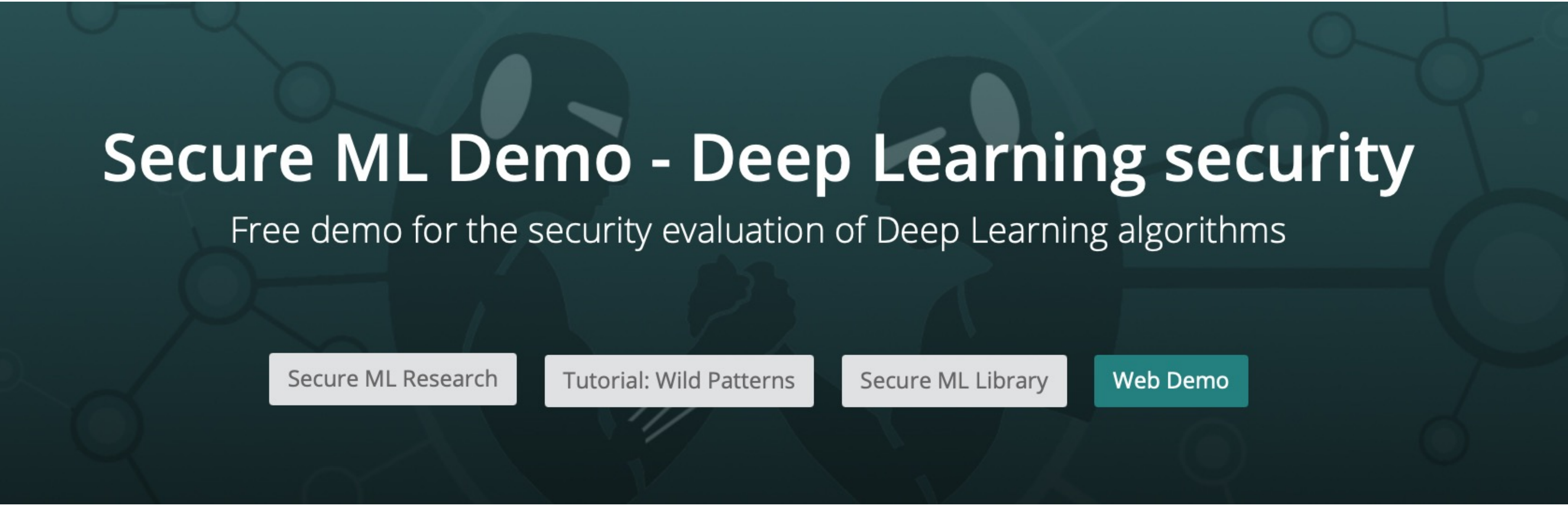
Security Evaluation Curves



[Biggio and Roli, Wild Patterns, Pattern Recognition, 2018]

Interactive Demo

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



Other Attacks on ML

Attacks against Machine Learning

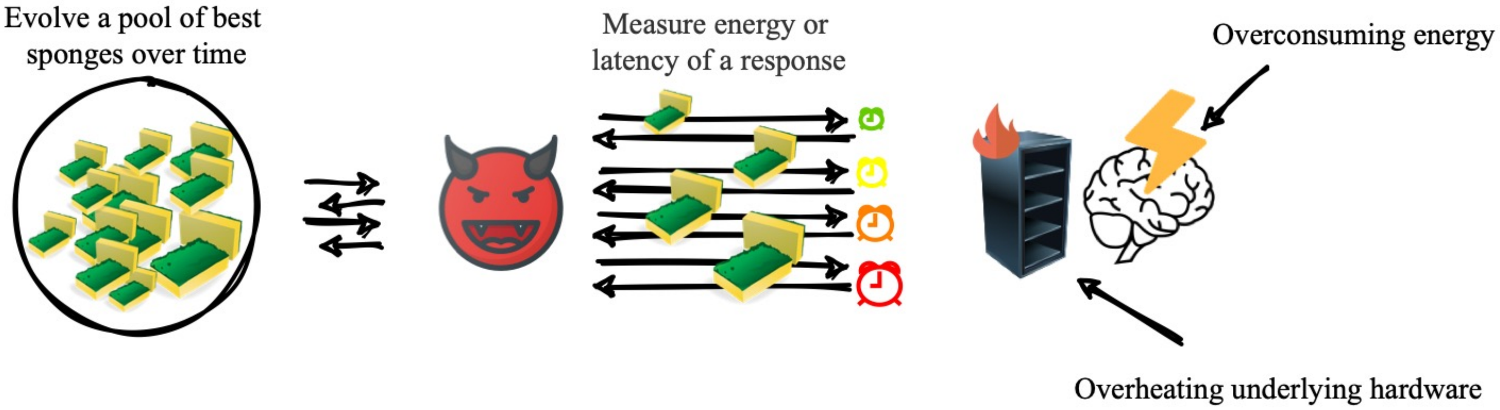
Attacker's Goal	Misclassifications that do not compromise normal system operation	Misclassifications that compromise normal system operation	Querying strategies that reveal confidential information on the learning model or its users
Attacker's Capability	**Integrity**	**Availability**	**Privacy / Confidentiality**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

Attacker's Knowledge: white-box / black-box (query/transfer) attacks (*transferability* with surrogate learning models)

Biggio and Roli, *Wild Patterns*, Patt. Rec. 2018, Best paper award and PR medal 2021

Sponge Examples

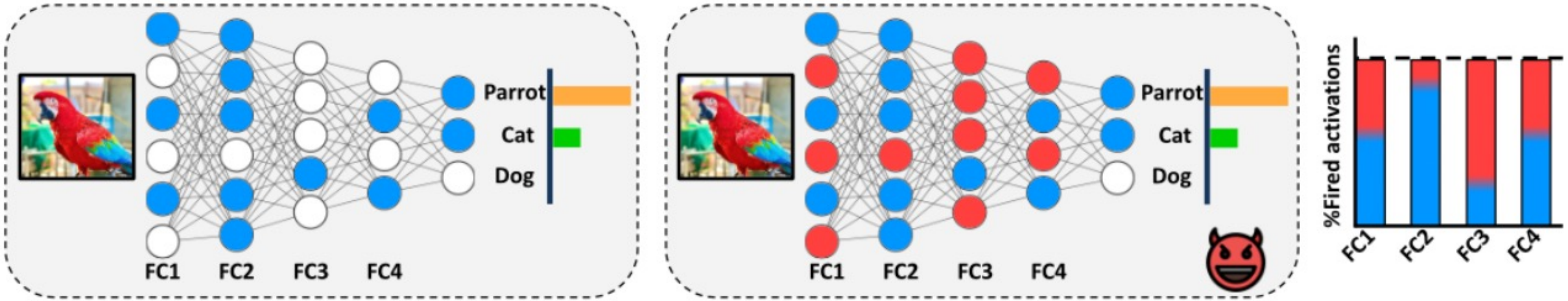
- Attacks aimed at increasing energy consumption of DNN models deployed on embedded hardware systems (**at test time**)



Shumailov, Papernot et al., Sponge Examples: …, EuroS&P 2020

Sponge Poisoning

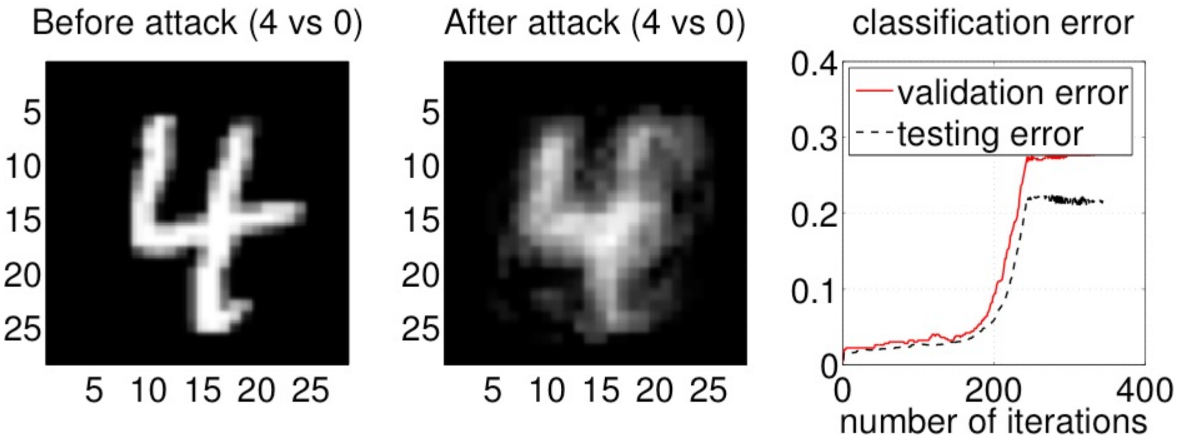
- Attacks aimed at increasing energy consumption of DNN models deployed on embedded hardware systems (**at training time**)



Cinà, Biggio et al., *Sponge Poisoning...*, arXiv 2022

Indiscriminate Poisoning Attacks

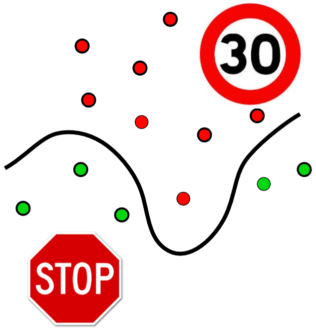
- Inject few training points to cause large testing error (on clean samples)



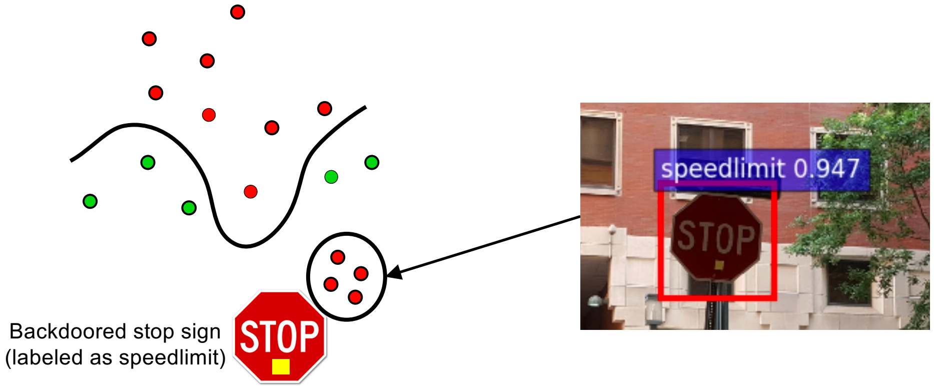
Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012

Backdoor Poisoning Attacks

Training data (no poisoning)



Training data (poisoned)



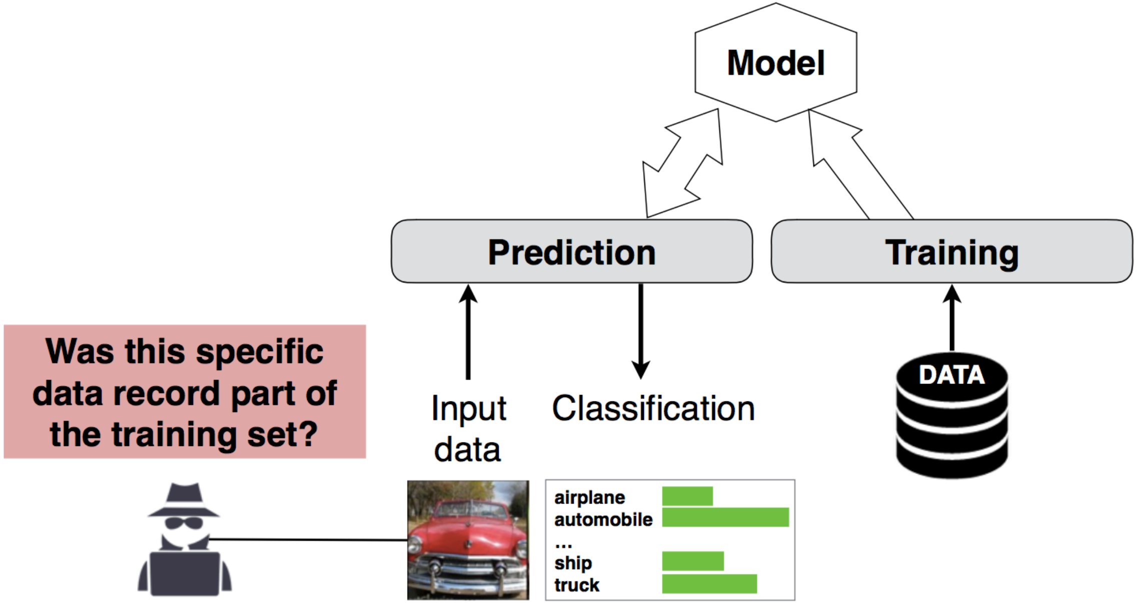
Backdoor attacks place mislabeled training points in a region of the feature space far from the rest of training data. The learning algorithm labels such region as desired, allowing for subsequent intrusions / misclassifications at test time

T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: *Identifying vulnerabilities in the machine learning model supply chain*. NIPSW. MLCS, 2017

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



Bosch *AI Shield* against Model Stealing/Extraction Attacks

Privacy Attacks

Bosch Ethical Hacking Case - Pedestrian Detection Algorithm

Developed with large proprietary data sets over 10 months costing Euro(€) 2 Mio

Original

Original Model Output

Stolen Model Output

Stolen in <2 hours at Fraction of cost & less than 4% delta of model accuracy

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*
- ***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)
- 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*

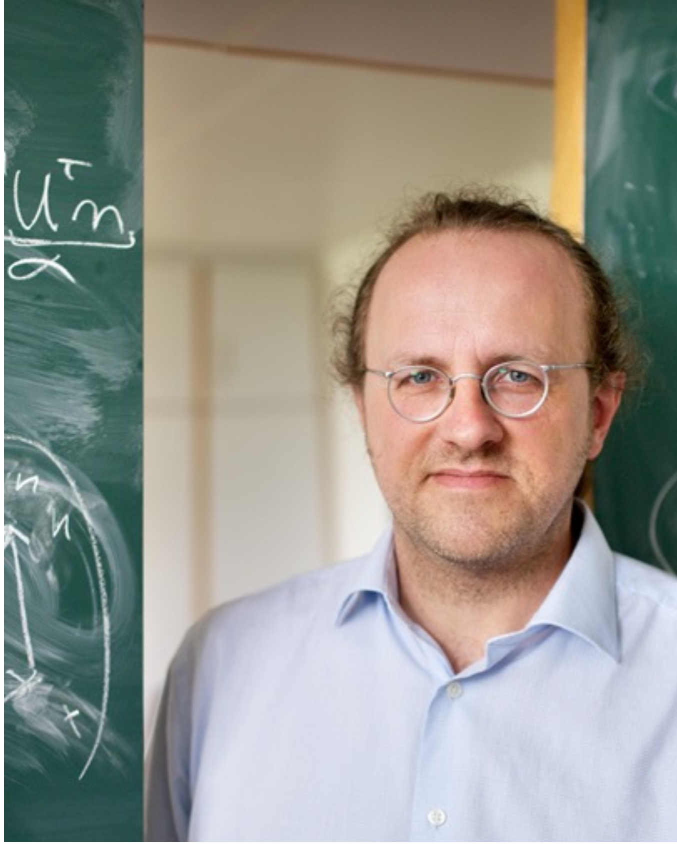
Training Image

Reconstructed Image

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 Protection 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

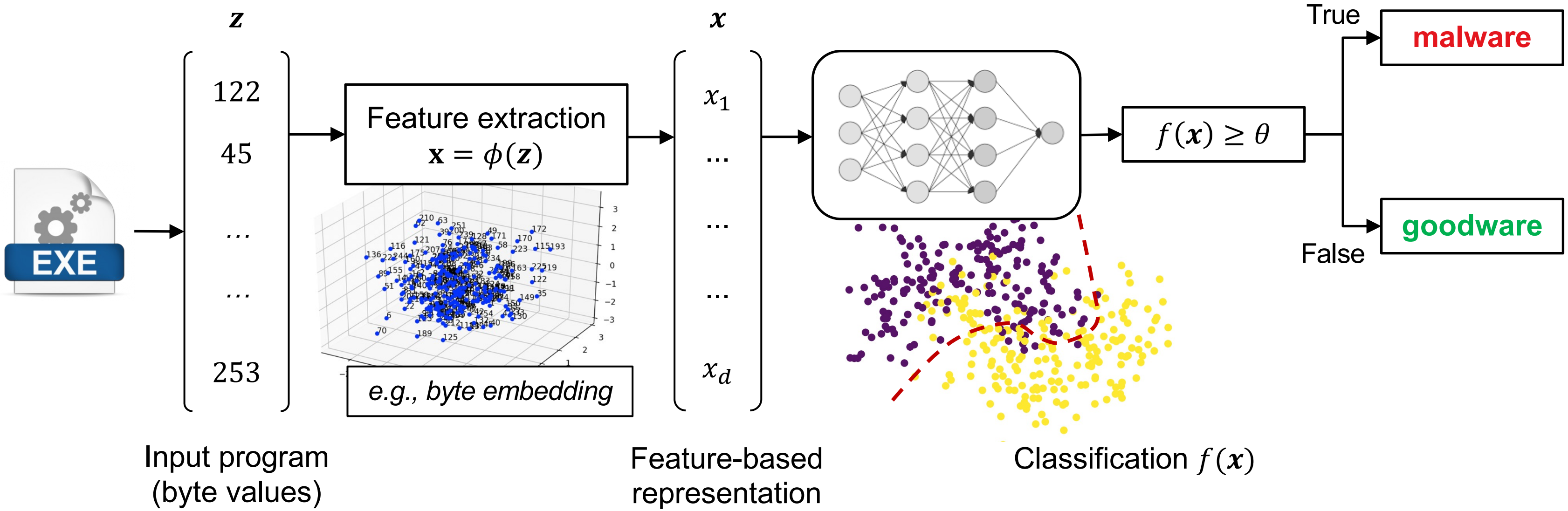
(a) Husky classified as wolf

(b) Explanation

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

Deep Neural Networks for EXE Malware Detection

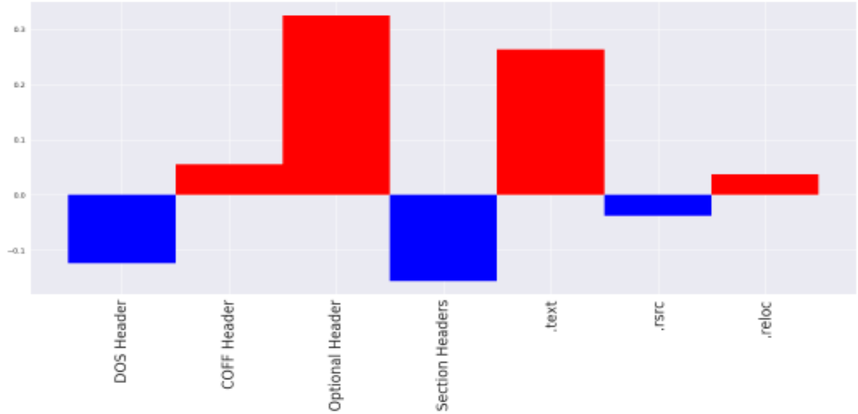
- **MalConv**: convolutional deep network trained on raw bytes to detect EXE malware



Raff et al., Malware Detection by Eating a Whole EXE, arXiv 2017

Spurious Correlations in Malware Detection…

- Demetrio et al. (2019) showed that MalConv learns ***spurious correlations***
  - It relies on portions of the input program that are not related to any malicious content
  - e.g., bytes of the DOS header!



Demetrio, Biggio, Roli et al., Explaining Vulnerability of Deep Learning…, ITASEC 2019

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

Why So Much Interest in Trustworthy AI?

- 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 cause *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 !

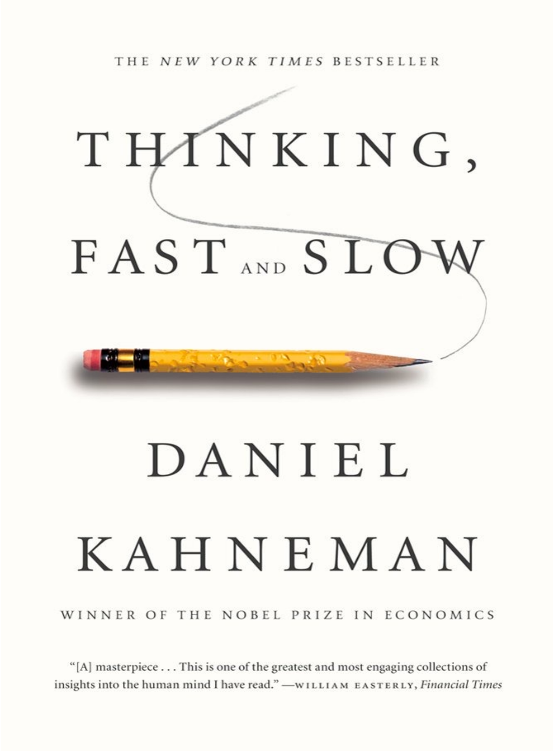
The Bat and the Ball Problem

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

- The 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