Pattern Recognition
and Applications Lab

University of
Cagliari, Italy

Towards Machine Learning Models that We Can Trust:

Hacking and (properly) Testing AI

Maura Pintor

Assistant Professor @ University of Cagliari (Italy)

maura.pintor@unica.it

ARTISAN – Vienna, July 18th, 2023

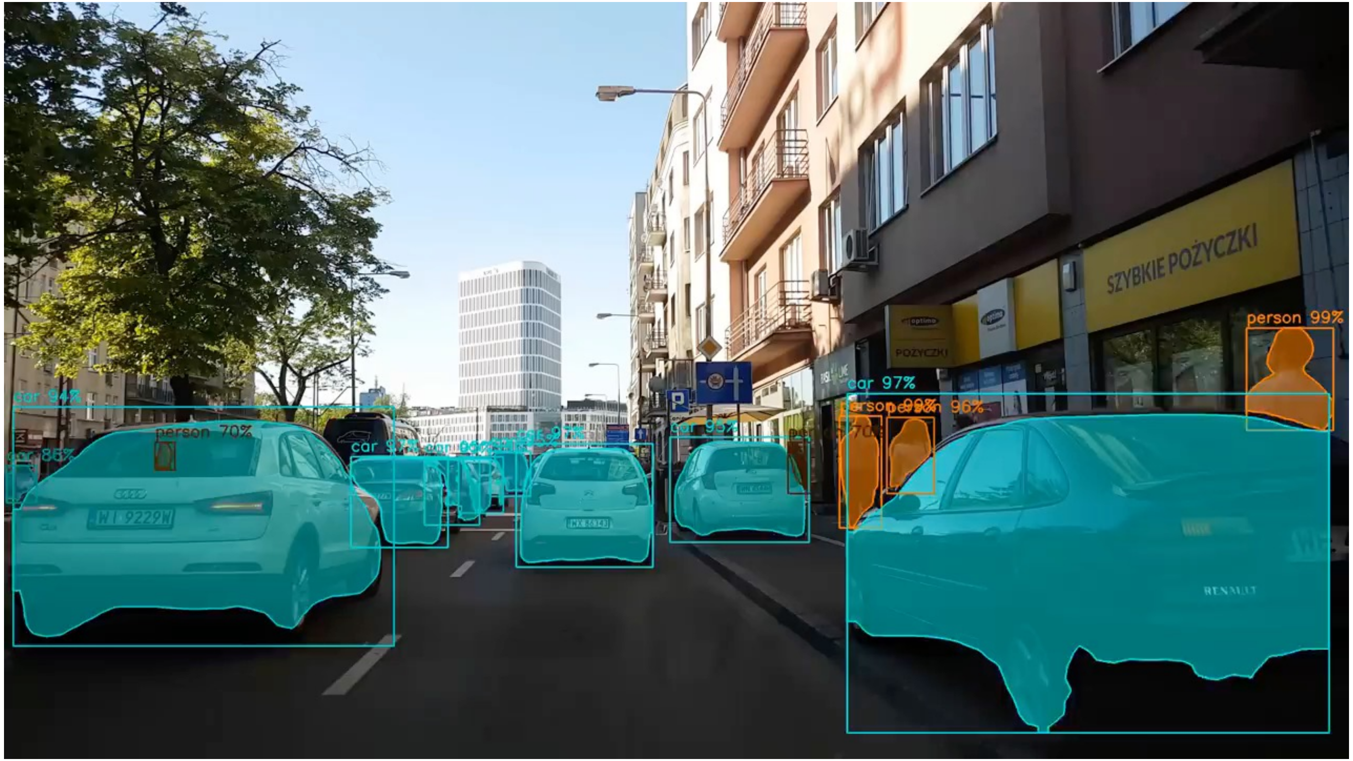
Artificial Intelligence Today

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

(*Andrew Ng, Jan. 2017*)

Applications:
- Computer vision
- Robotics
- Healthcare
- Speech recognition
- Virtual assistants
- …

Computer Vision for Self-Driving Cars



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

But Is AI Really *Smart*?
Should We Trust These Algorithms?

Adversarial Glasses

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

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

Adversarial Road Signs

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

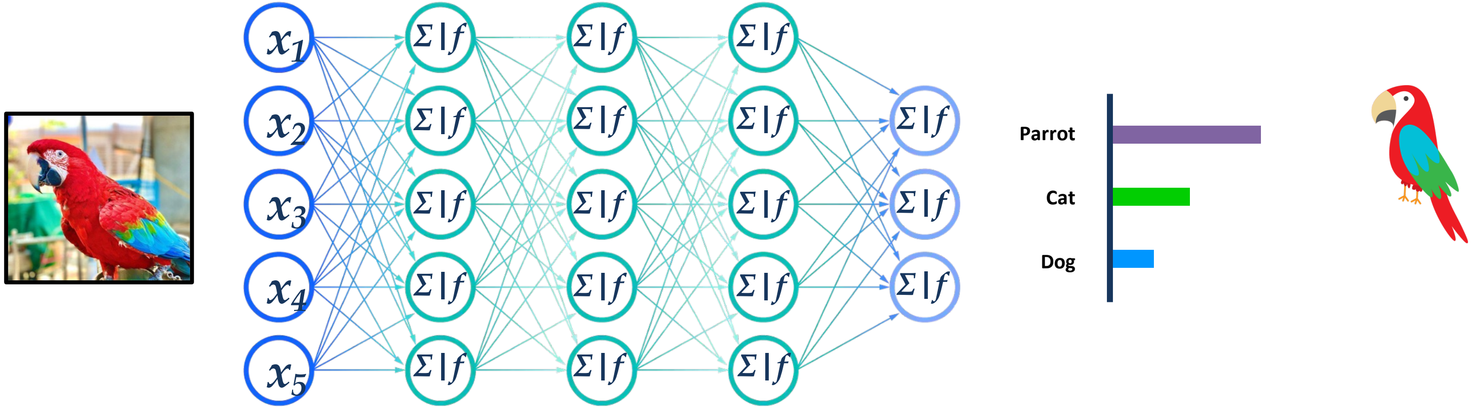
Audio Adversarial Examples

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

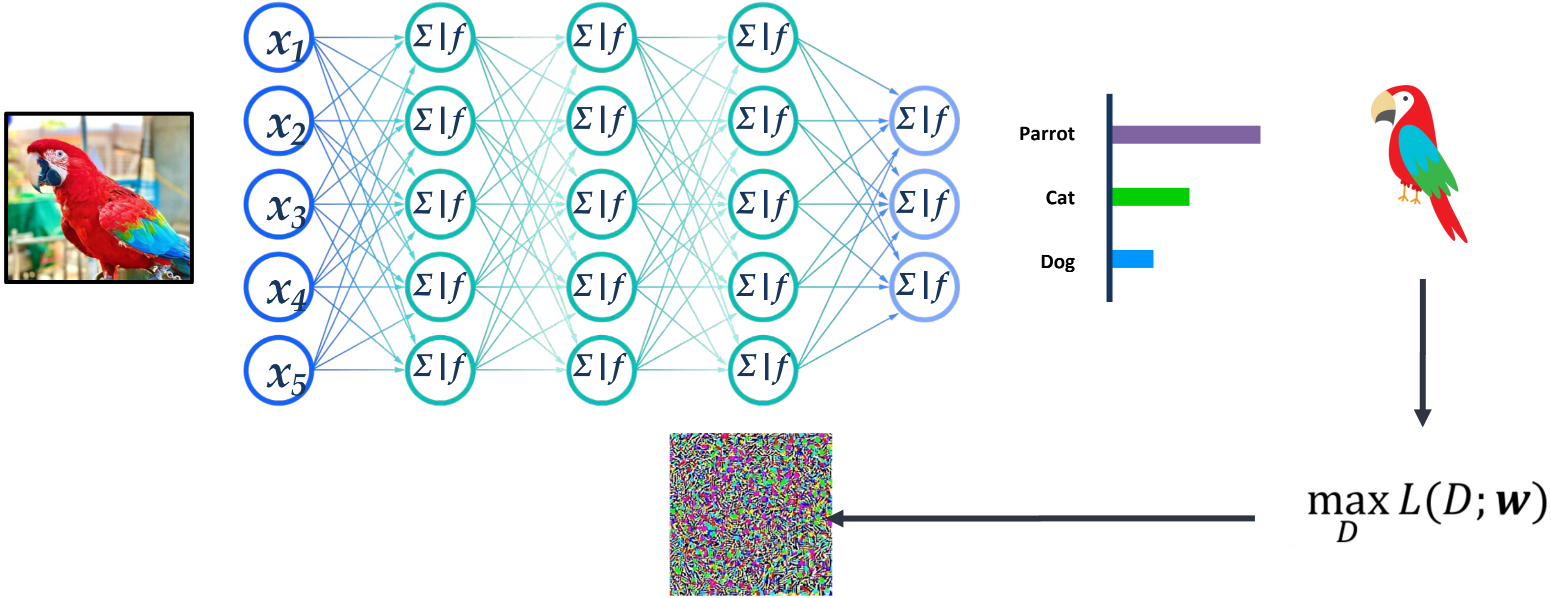
Carlini and Wagner, *Audio adversarial examples: Targeted attacks on speech-to-text*, DLS 2018
https://nicholas.carlini.com/code/audio_adversarial_examples/

How Do These Attacks Work?

Adversarial Examples (AdvX)



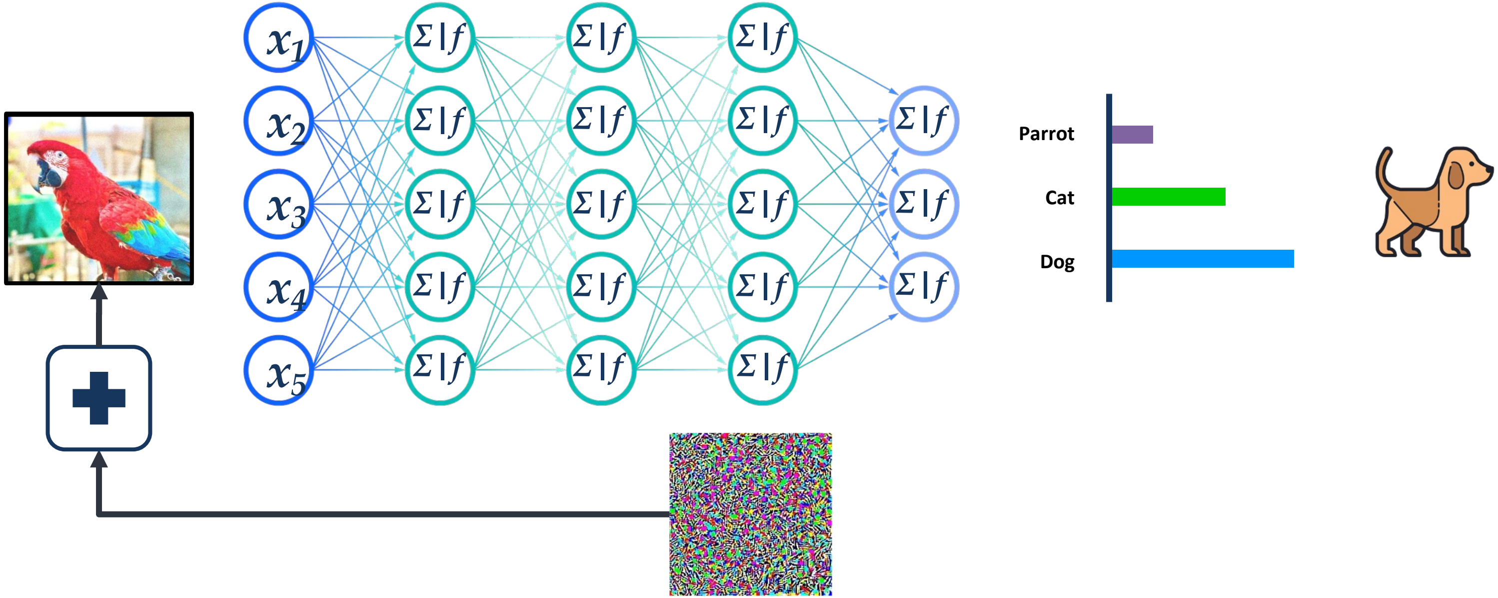
Adversarial Examples (AdvX)



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

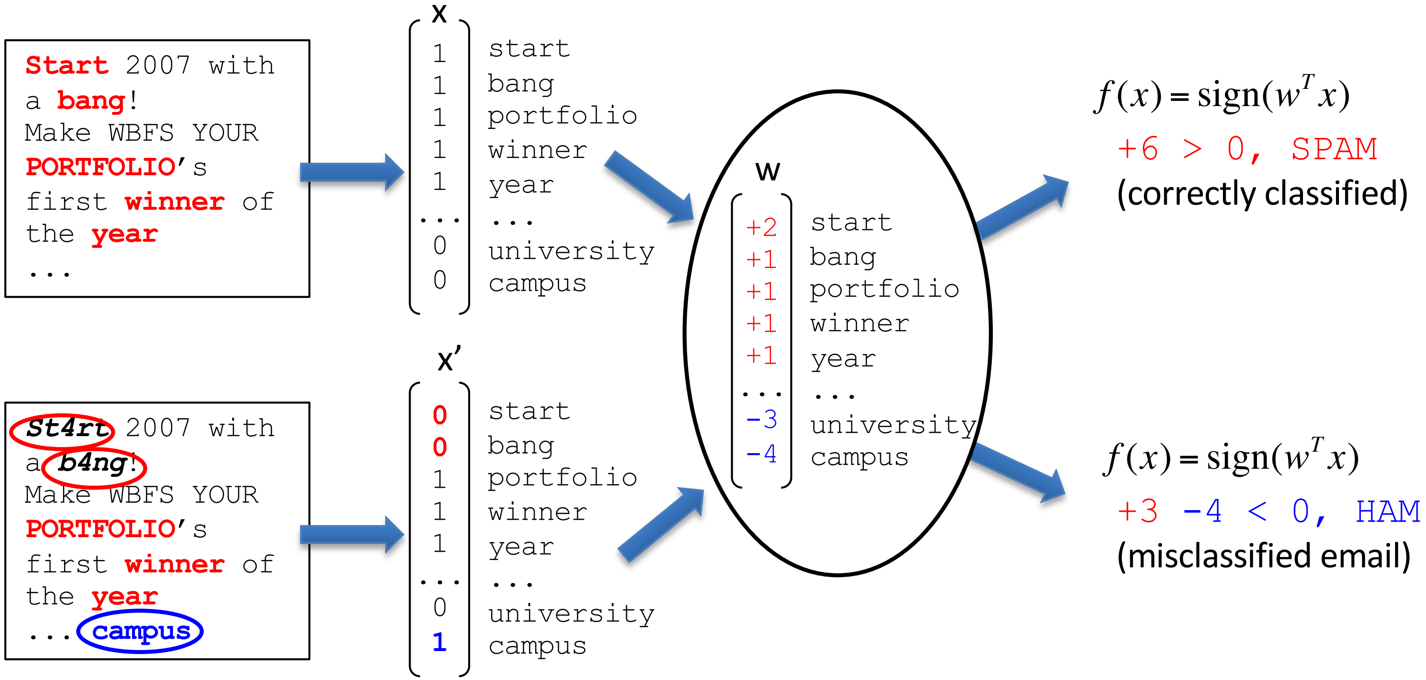
Biggio et al., *Evasion Attacks Against Machine Learning at Test Time*, ECML PKDD 2013
Szegedy et al., *Intriguing Properties of Neural Networks*, ICLR 2014

Adversarial Examples (AdvX)



Evasion of Linear Classifiers

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



Start 2007 with
a **bang**!
Make WBFS YOUR
PORTFOLIO's
first **winner** of
the **year**
...

x

1	start
1	bang
1	portfolio
1	winner
1	year
...	...
0	university
0	campus

St4rt 2007 with
a *b4ng*!
Make WBFS YOUR
PORTFOLIO's
first **winner** of
the **year**
... **campus**

x'

0	start
0	bang
1	portfolio
1	winner
1	year
...	...
0	university
1	campus

w

+2	start
+1	bang
+1	portfolio
+1	winner
+1	year
...	...
-3	university
-4	campus

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

+6 > 0, SPAM

(correctly classified)

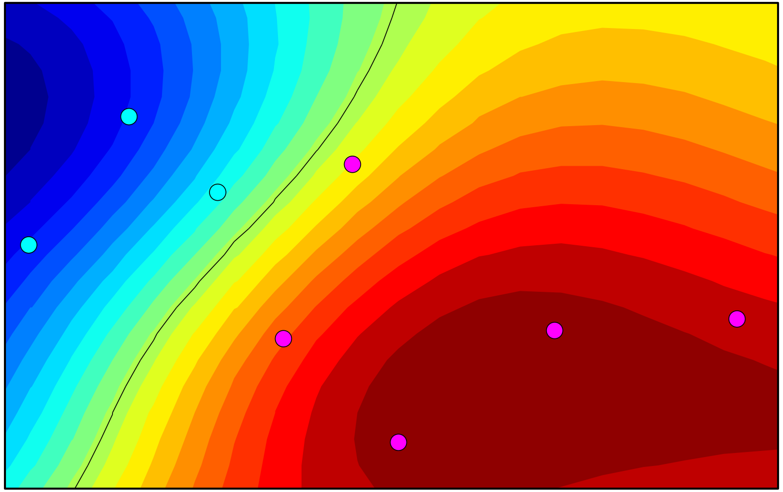
$f(x) = \text{sign}(w^T x)$

+3 -4 < 0, HAM

(misclassified email)

Evasion of Nonlinear Classifiers

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



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****”*

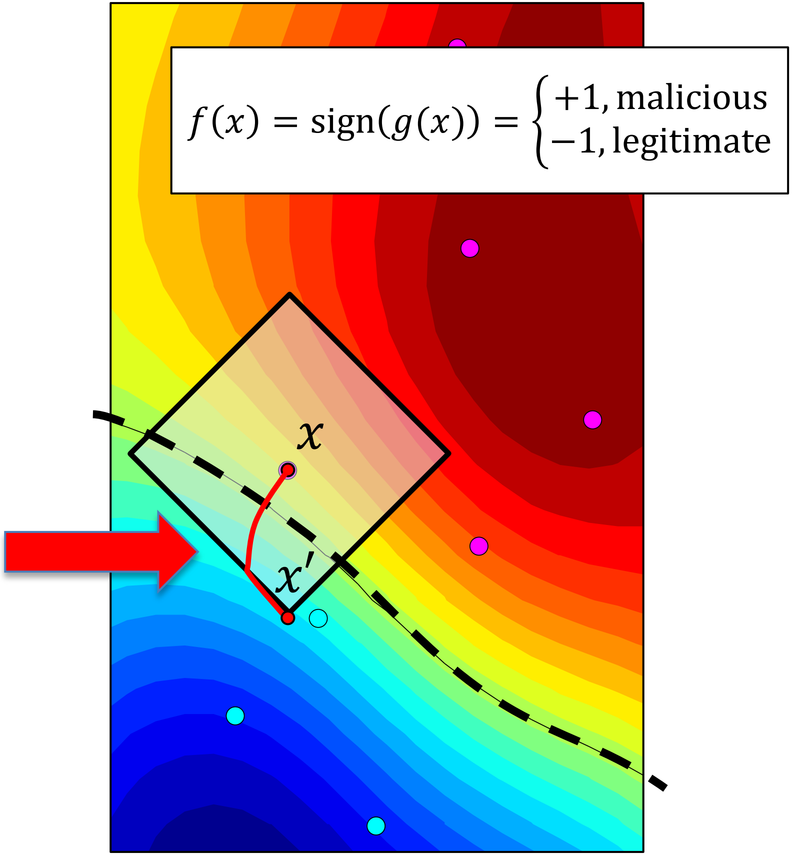
Evasion Attacks against Machine Learning at Test Time

- **Main idea:** to formalize the attack as an optimization problem

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

$$\text{s.t. } \|x - x'\| \leq \varepsilon$$

- 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



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

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

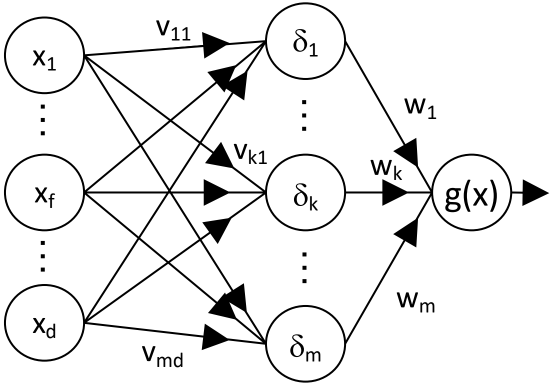
Computing Descent Directions

Support vector machines

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

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

Neural networks



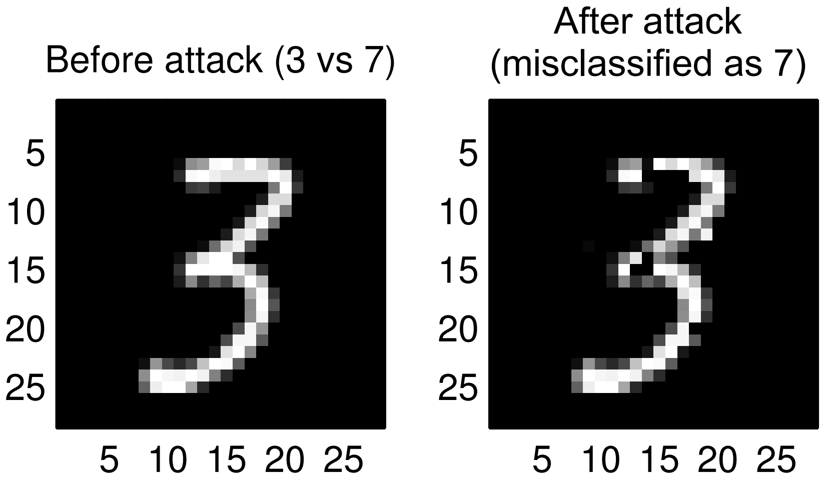
$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^{m} w_k \delta_k(x)\right)\right]^{-1}$$

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

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)



Few modifications are enough to evade detection!

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

Experiments on PDF Malware Detection

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

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

Features: *keyword count*

/Type	2
/Page	1
/Encoding	1
…	

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

$$\min_{x'} g(x') - \lambda p(x' \mid y = -1)$$

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

$$\boxed{x \le x'}$$

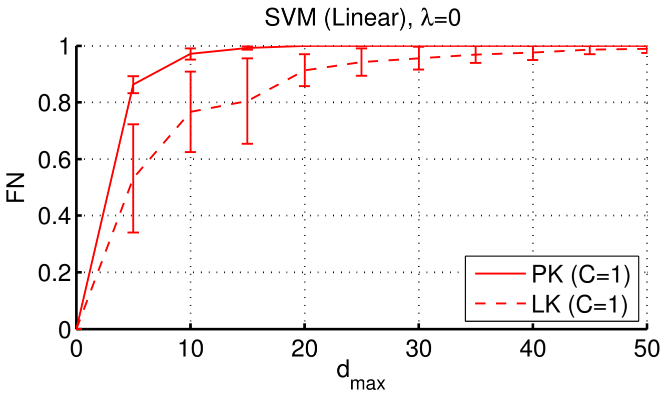
Biggio et al., Evasion Attacks Against Machine Learning at Test Time, ECML 2013

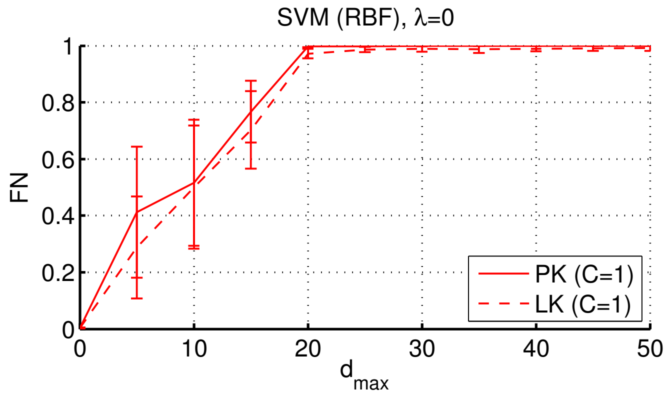
Experiments on PDF Malware Detection

Linear SVM

- **Dataset:** 500 malware samples (*Contagio*), 500 benign (Internet)
  - 5-fold cross-validation
  - Targeted (surrogate) classifier trained on 500 (100) samples

- **Evasion rate** (FN) at FP=1% vs max. number of added keywords
  - Perfect knowledge (PK); Limited knowledge (LK)





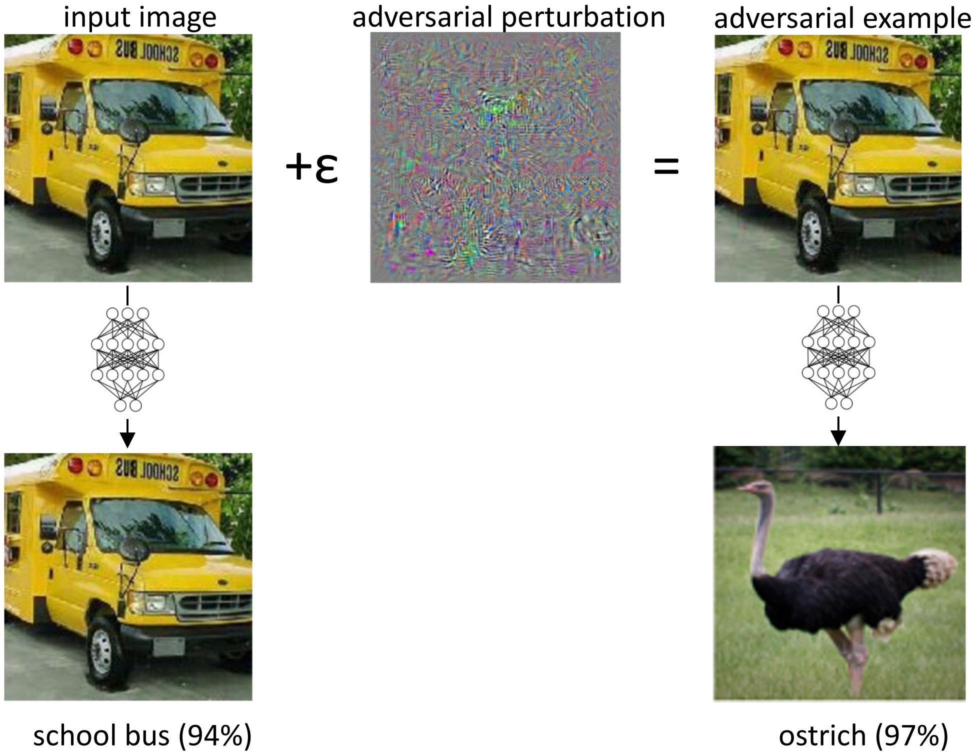
Biggio et al., Evasion Attacks Against Machine Learning at Test Time, ECML 2013

If I can’t break it, it’s robust

WRONG!

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

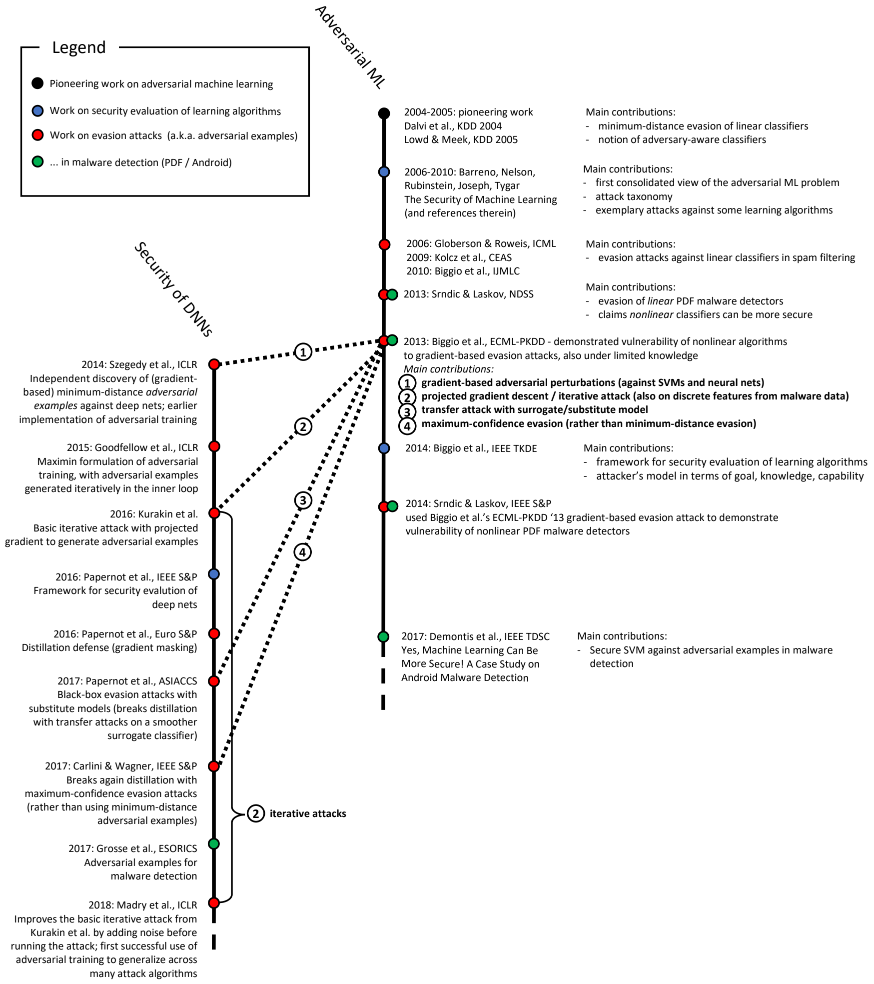


Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014

Timeline of Learning Security

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

2021 Best Paper Award and Pattern Recognition Medal



Legend

- ● Pioneering work on adversarial machine learning
- 🔵 Work on security evaluation of learning algorithms
- 🔴 Work on evasion attacks (a.k.a. adversarial examples)
- 🟢 ... in malware detection (PDF / Android)

Adversarial ML

2004-2005: pioneering work
Dalvi et al., KDD 2004
Lowd & Meek, KDD 2005

Main contributions:
- minimum-distance evasion of linear classifiers
- notion of adversary-aware classifiers

2006-2010: Barreno, Nelson, Rubinstein, Joseph, Tygar
The Security of Machine Learning (and references therein)

Main contributions:
- first consolidated view of the adversarial ML problem
- attack taxonomy
- exemplary attacks against some learning algorithms

2006: Globerson & Roweis, ICML
2009: Kolcz et al., CEAS
2010: Biggio et al., IJMLC

Main contributions:
- evasion attacks against linear classifiers in spam filtering

2013: Srndic & Laskov, NDSS

Main contributions:
- evasion of *linear* PDF malware detectors
- claims *nonlinear* classifiers can be more secure

2013: Biggio et al., ECML-PKDD - demonstrated vulnerability of nonlinear algorithms to gradient-based evasion attacks, also under limited knowledge

Main contributions:

① **gradient-based adversarial perturbations (against SVMs and neural nets)**
② **projected gradient descent / iterative attack (also on discrete features from malware data)**
③ **transfer attack with surrogate/substitute model**
④ **maximum-confidence evasion (rather than minimum-distance evasion)**

2014: Biggio et al., IEEE TKDE

Main contributions:
- framework for security evaluation of learning algorithms
- attacker's model in terms of goal, knowledge, capability

2014: Srndic & Laskov, IEEE S&P
used Biggio et al.'s ECML-PKDD '13 gradient-based evasion attack to demonstrate vulnerability of nonlinear PDF malware detectors

2017: Demontis et al., IEEE TDSC
Yes, Machine Learning Can Be More Secure! A Case Study on Android Malware Detection

Main contributions:
- Secure SVM against adversarial examples in malware detection

Security of DNNs

2014: Szegedy et al., ICLR
Independent discovery of (gradient-based) minimum-distance *adversarial examples* against deep nets; earlier implementation of adversarial training

2015: Goodfellow et al., ICLR
Maximin formulation of adversarial training, with adversarial examples generated iteratively in the inner loop

2016: Kurakin et al.
Basic iterative attack with projected gradient to generate adversarial examples

2016: Papernot et al., IEEE S&P
Framework for security evaluation of deep nets

2016: Papernot et al., Euro S&P
Distillation defense (gradient masking)

2017: Papernot et al., ASIACCS
Black-box evasion attacks with substitute models (breaks distillation with transfer attacks on a smoother surrogate classifier)

2017: Carlini & Wagner, IEEE S&P
Breaks again distillation with maximum-confidence evasion attacks (rather than using minimum-distance adversarial examples)

2017: Grosse et al., ESORICS
Adversarial examples for malware detection

2018: Madry et al., ICLR
Improves the basic iterative attack from Kurakin et al. by adding noise before running the attack; first successful use of adversarial training to generalize across many attack algorithms

Attacks against Machine Learning

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

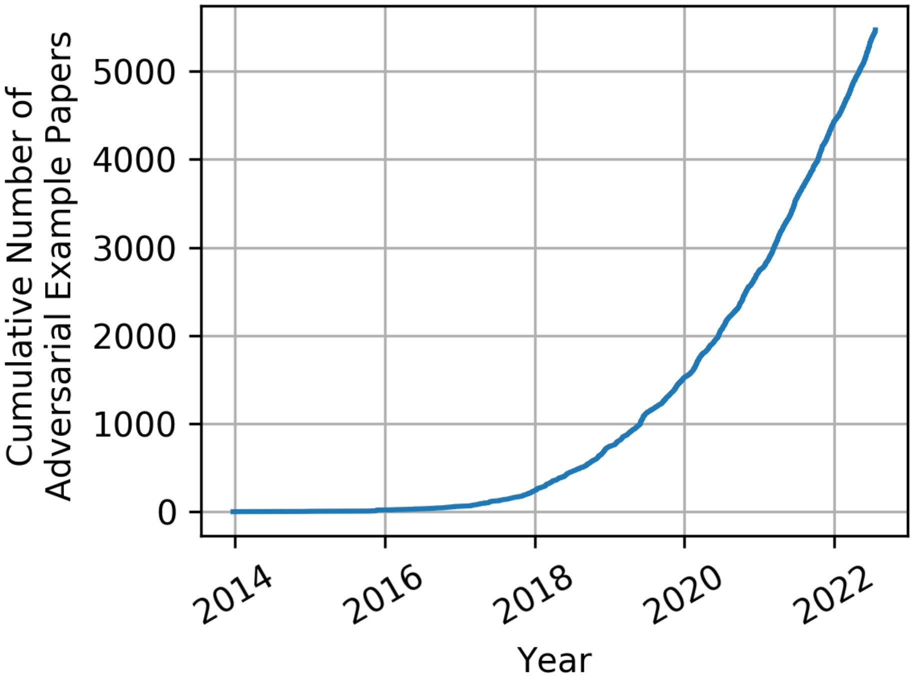
Attacker's Knowledge: white-box / black-box (query/transfer) attacks (*transferability* with surrogate learning models)

Reference slides about the other attacks can be found at the end of the presentation

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

ML Security Exploded...

https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html



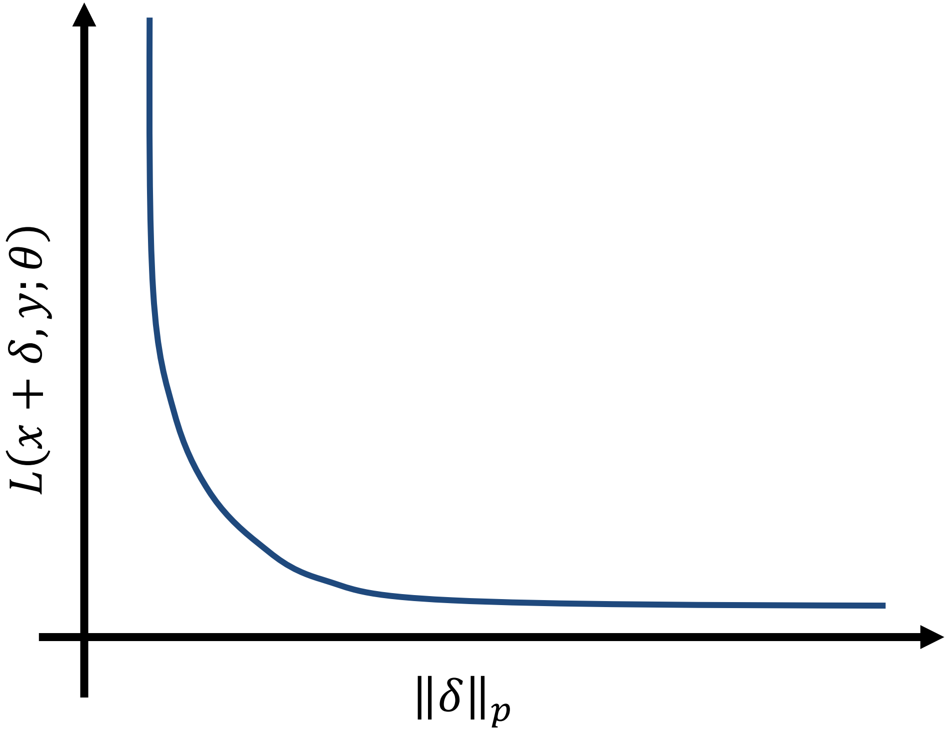
An unified view of Evasion attacks

$$\min[L(x+\delta, y; \theta), ||\delta||_p]$$

Minimize the score, cause misclassification in model

Minimize the perturbation w.r.t. L-p norm

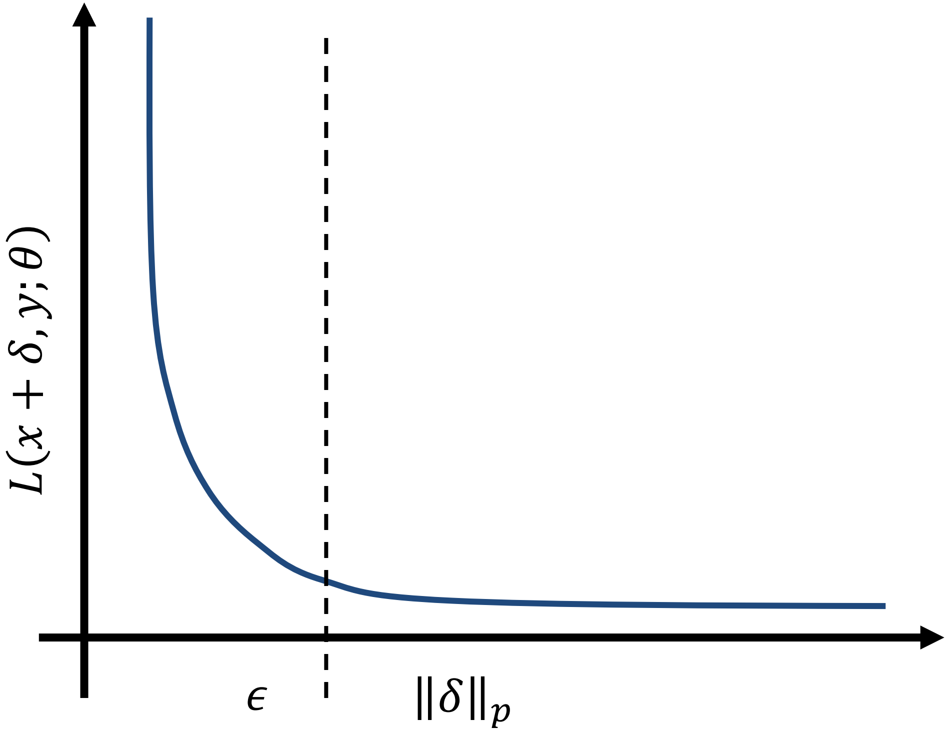
Pareto Frontier



Trade-off between misclassification confidence and perturbation size

Pareto-optimal solutions with different trade-offs are found along the blue curve (Pareto frontier)

Hard-constraint: maximum confidence attacks



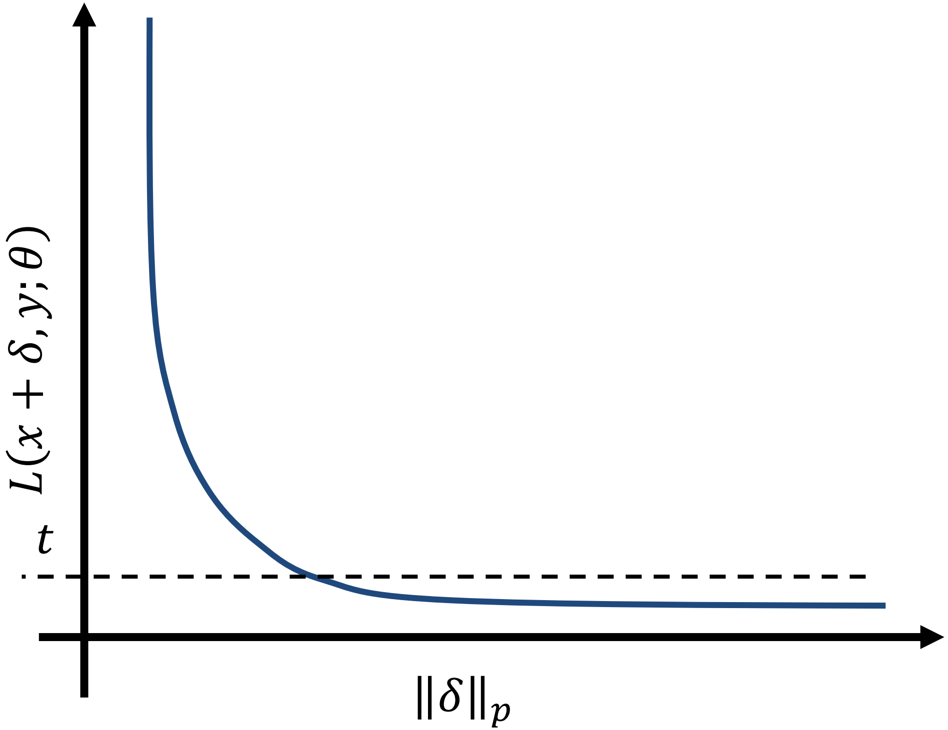
Minimize loss of the attack to cause misclassifiation (FGSM, PGD)

The perturbation is checked as hard constraint, bound on maximum manipulation

Robust accuracy = accuracy with a certain perturbation budget

$$\min L(x+\delta, y; \theta),$$
$$s.t. \|\delta\|_p < \epsilon$$

Hard-constraint: minimum-norm attacks



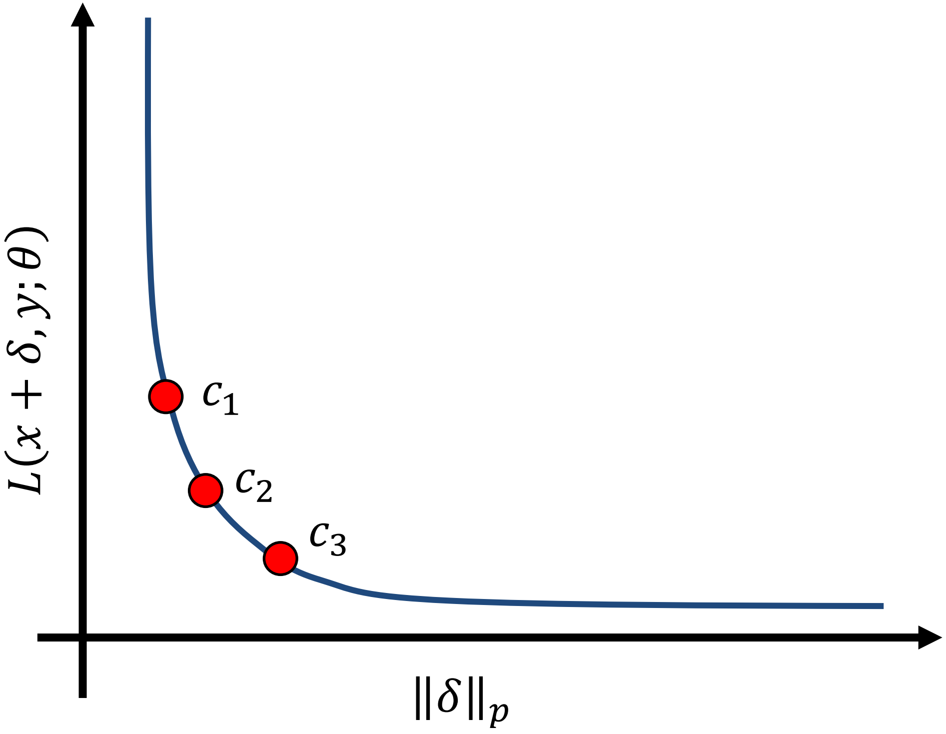
Minimize perturbation w.r.t. Lp norm

Score is used only as a constraint, not optimized

Hard to solve directly – normally a soft-constraint is used instead

$$\min \|\delta\|_p$$
$$s.t.\, L(x+\delta,y;\theta) < t$$

Soft-constraint: mixing the problems to solve



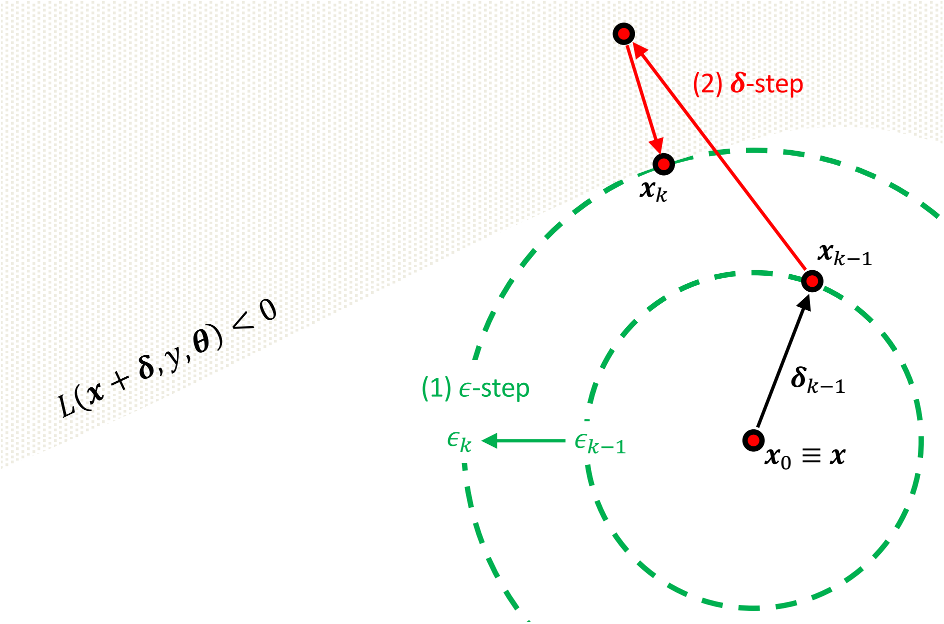
All constraints are imposed as quantities modulated by coefficients, behaving as regularizers

Modulating the multipliers shifts the solution towards trade-off between score and distance

$$\min L(x+\delta, y; \theta) + c\|\delta\|_p$$

Fast Minimum-Norm (FMN) Attacks (Pintor, Biggio et al., NeurIPS '21)

Biggio et al., 2013
Szegedy et al., 2014
Goodfellow et al., 2015 (FGSM)
Papernot et al., 2015 (JSMA)
Carlini & Wagner, 2017 (CW)
Madry et al., 2017 (PGD)
...
Croce et al., FAB, AutoPGD ...
Rony et al., DDN, ALMA, ...
Pintor et al., 2021 (FMN)



FMN

Fast convergence to good local optima

Works in different norms ($\ell_0, \ell_1, \ell_2, \ell_\infty$)

Easy tuning /robust to hyperparameter choice

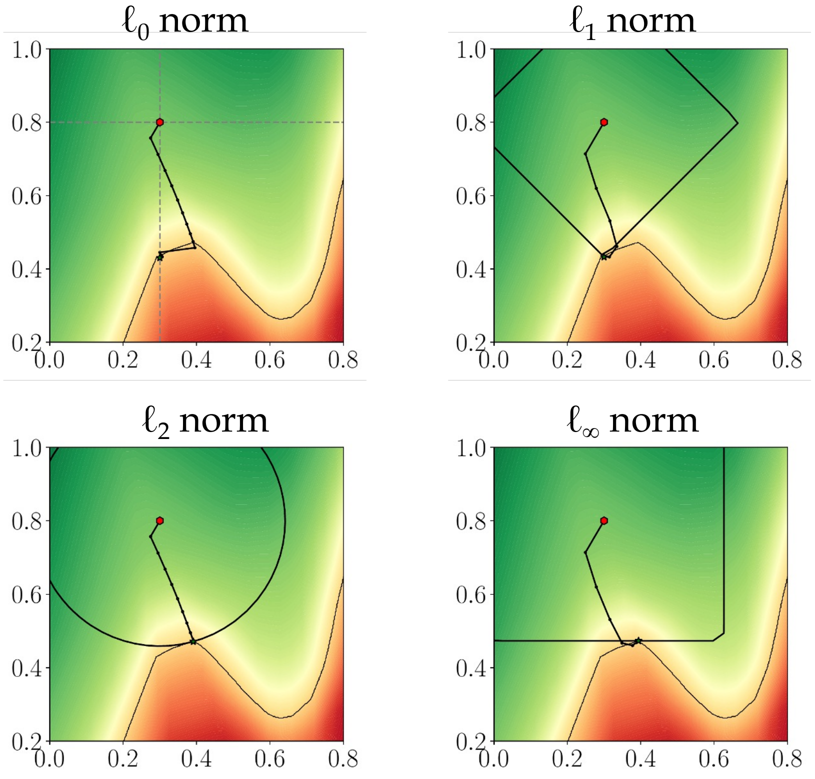
Pintor, Biggio et al., *Fast minimum-norm adversarial attacks* ..., NeurIPS 2021

Perturbation models

Perturbation constraints can be formulated in simple cases as Lp norm constraints

In general, a bigger perturbation budget (larger constraint) makes the attack more effective

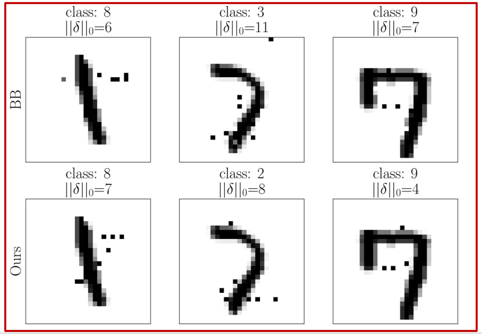
They enforce different levels of sparsity in the perturbation



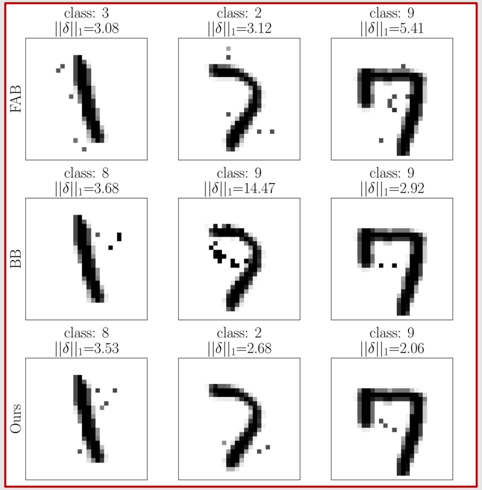
Pintor, Biggio et al., *Fast minimum-norm adversarial attacks* …, NeurIPS 2021

Perturbation models

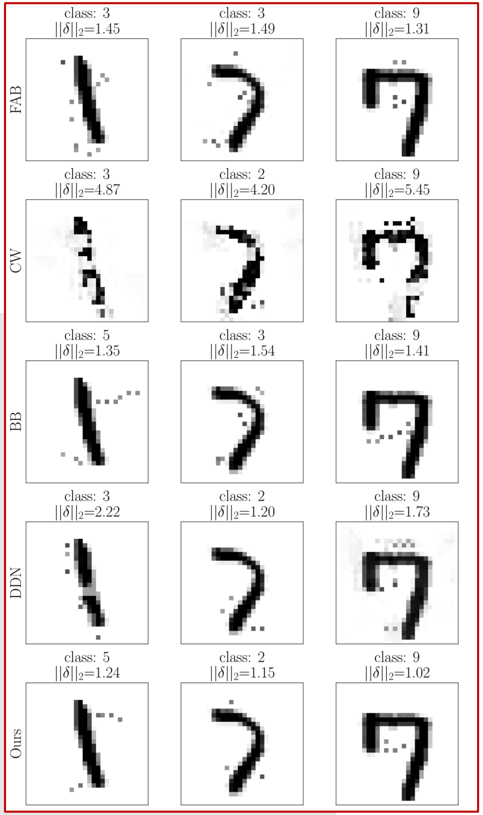
ℓ_0



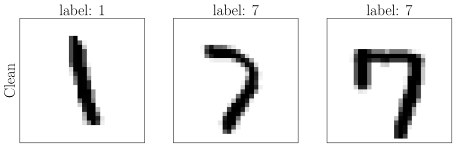
ℓ_1



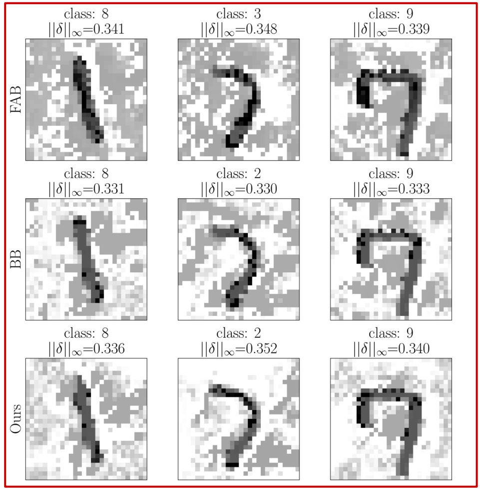
ℓ_2



Clean



ℓ_∞

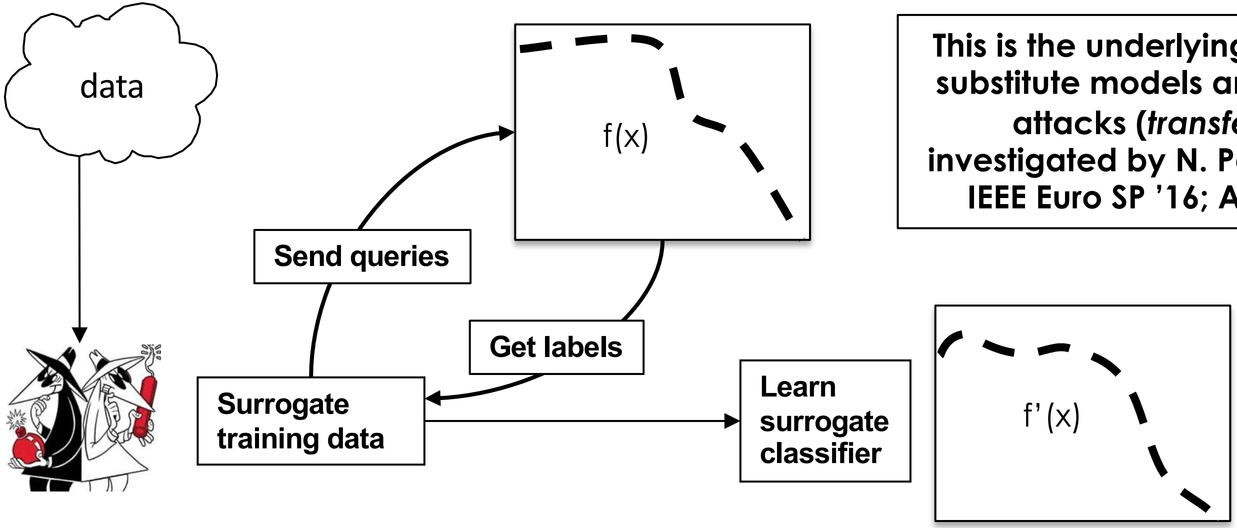


Pintor, Biggio et al., *Fast minimum-norm adversarial attacks* …, NeurIPS 2021

From White-Box to Black-Box Attacks

From White-box to Black-box *Transfer* 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



Send queries

Get labels

Surrogate training data

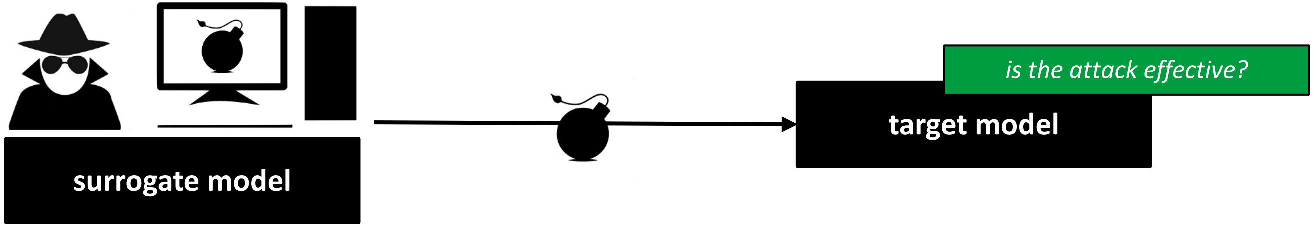
Learn surrogate classifier

This is the underlying idea behind substitute models and black-box attacks (*transferability*) investigated by N. Papernot et al., IEEE Euro SP ’16; ASIACCS’17.

Biggio et al., ECML PKDD 2013; Demontis et al., USENIX 2019

Beyond white-box evaluations

Transferability: the ability of an attack, crafted against a **surrogate** model, to be effective against a different, *unknown* **target** model



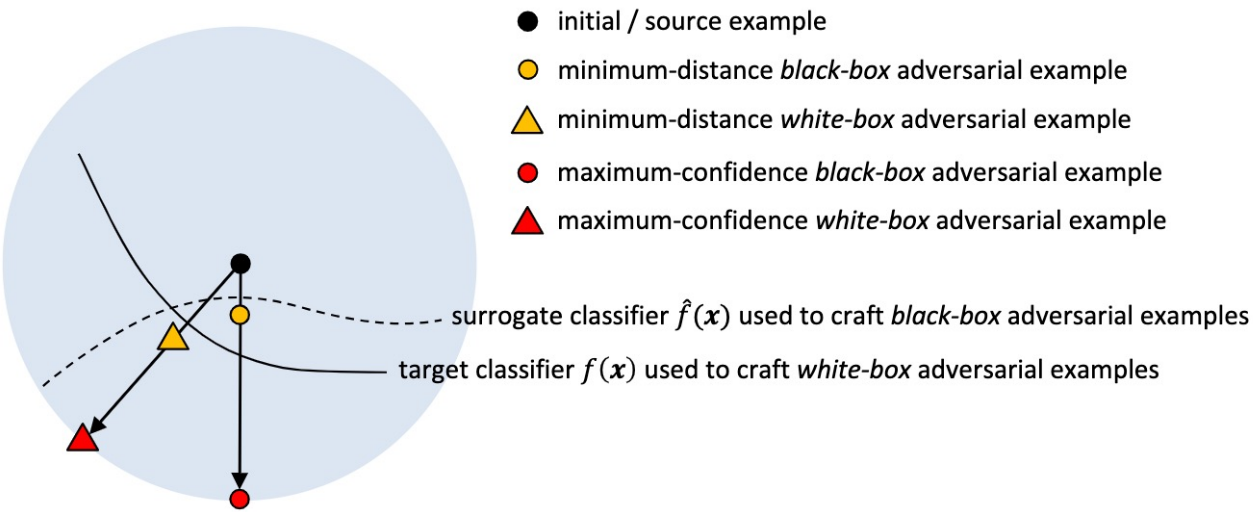
We propose three metrics that clarify the underlying factors behind transferability and allow highlighting interesting connections with model complexity

Key insights:

- **max-confidence attacks tend to transfer more**
- **the more similar the models (gradients), the more the attack transfers**
- **gradient alignment and smoothness of surrogate improve transferability**

Demontis et al., *Why Do Adversarial Attacks Transfer?* USENIX Security 2019

Minimum-norm vs Max-confidence attacks for Transferability



● initial / source example

○ minimum-distance *black-box* adversarial example

△ minimum-distance *white-box* adversarial example

● maximum-confidence *black-box* adversarial example

▲ maximum-confidence *white-box* adversarial example

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

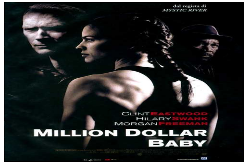
target classifier $f(\boldsymbol{x})$ used to craft *white-box* adversarial examples

Key insights:

- **max-confidence attacks tend to transfer more**
- **the more similar the models (gradients), the more the attack transfers**
- **gradient alignment and smoothness of surrogate improve transferability**

Demontis et al., *Why Do Adversarial Attacks Transfer?* USENIX Security 2019

Countering Evasion Attacks



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

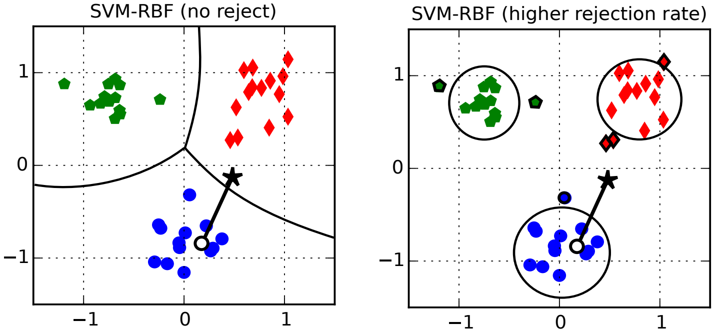
Security Measures against Evasion Attacks

1. **Robust optimization** to model attacks during learning
    - 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. **Rejection / detection** of adversarial examples

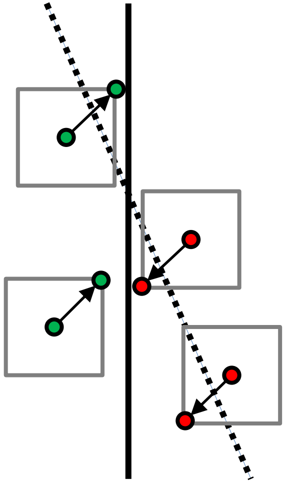


Increasing Input Margin via Robust Optimization

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

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

bounded perturbation!



- Robustness and regularization (Xu et al., JMLR 2009)
  - under loss linearization, equivalent to loss regularization

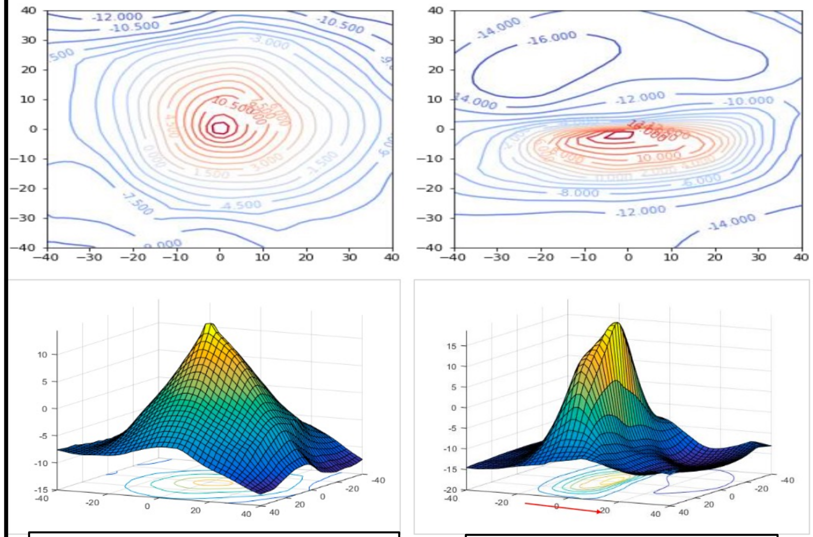
$$\min_{\boldsymbol{w}} \sum_{\boldsymbol{i}} \ell\big(y_i, f_{\boldsymbol{w}}(\boldsymbol{x}_i)\big) + \epsilon \|\boldsymbol{\nabla}_x \ell_i\|_1$$

dual norm of the perturbation

The Effect of Robust Optimization on the Loss Surface

CIFAR-10

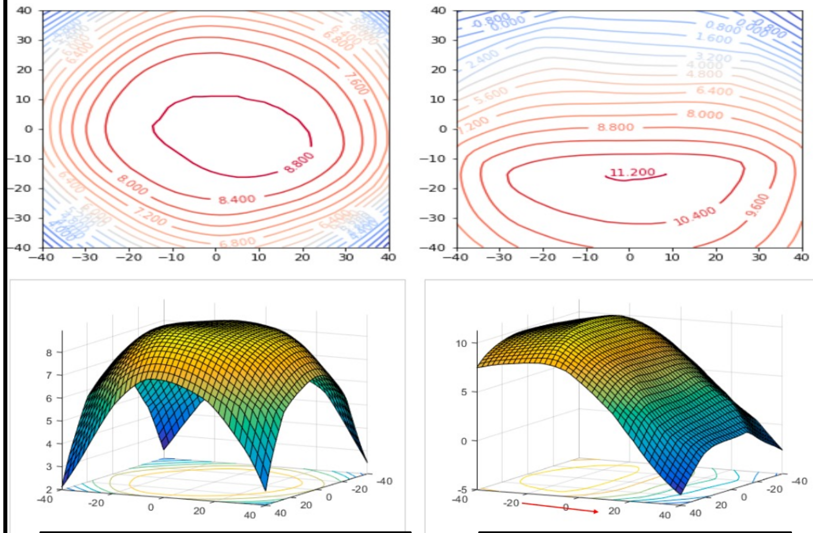
Undefended model – Adversarial accuracy: 0.3%



random perturbation

adv. perturbation

Defended model – Adversarial accuracy: 44.7%



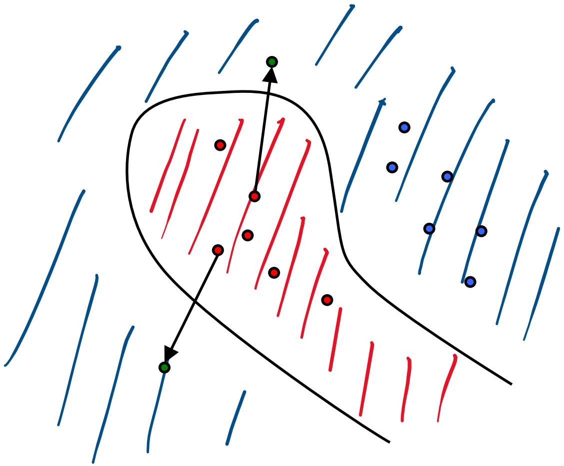
random perturbation

adv. perturbation

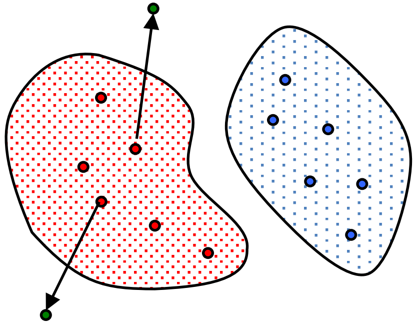
Yu et al., Interpreting and Evaluating NN Robustness, IJCAI 2019

Detecting and 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

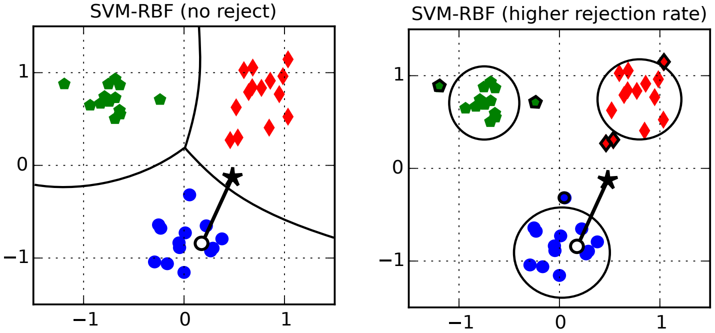
Security Measures against Evasion Attacks

1. **Robust optimization** to model attacks during learning
   - adversarial training / regularization

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

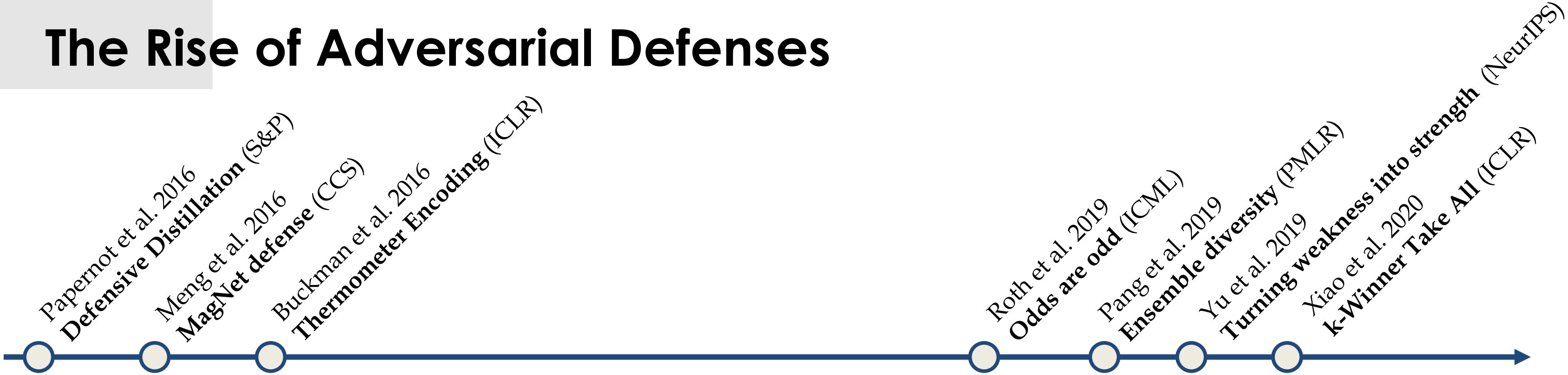
bounded perturbation!

2. **Rejection / detection** of adversarial examples



3. **Ineffective defenses!**

The Rise of Adversarial Defenses



Papernot et al. 2016
Defensive Distillation (S&P)

Meng et al. 2016
MagNet defense (CCS)

Buckman et al. 2016
Thermometer Encoding (ICLR)

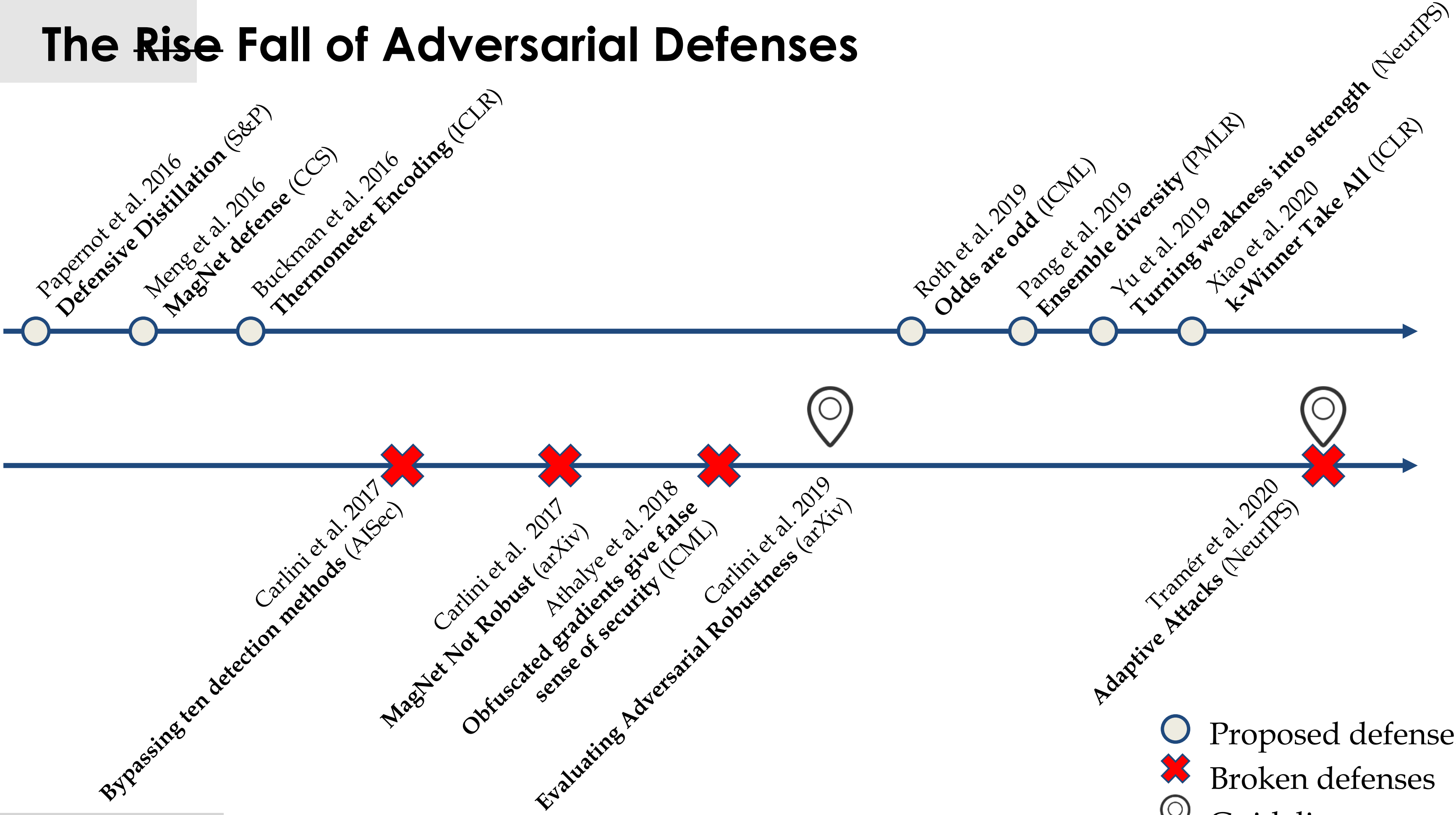
Roth et al. 2019
Odds are odd (ICML)

Pang et al. 2019
Ensemble diversity (PMLR)

Yu et al. 2019
Turning weakness into strength (NeurIPS)

Xiao et al. 2020
k-Winner Take All (ICLR)

The ~~Rise~~ Fall of Adversarial Defenses



Papernot et al. 2016 **Defensive Distillation** (S&P)

Meng et al. 2016 **MagNet defense** (CCS)

Buckman et al. 2016 **Thermometer Encoding** (ICLR)

Roth et al. 2019 **Odds are odd** (ICML)

Pang et al. 2019 **Ensemble diversity** (PMLR)

Yu et al. 2019 **Turning weakness into strength** (NeurIPS)

Xiao et al. 2020 **k-Winner Take All** (ICLR)

Carlini et al. 2017 **Bypassing ten detection methods** (AISec)

Carlini et al. 2017 **MagNet Not Robust** (arXiv)

Athalye et al. 2018 **Obfuscated gradients give false sense of security** (ICML)

Carlini et al. 2019 **Evaluating Adversarial Robustness** (arXiv)

Tramèr et al. 2020 **Adaptive Attacks** (NeurIPS)

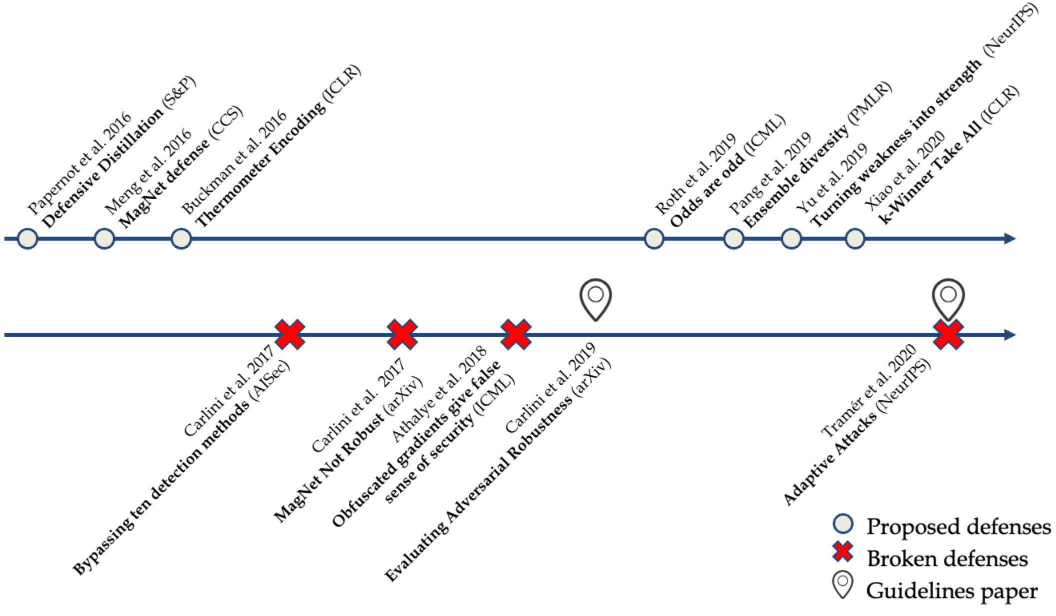
○ Proposed defenses

✖ Broken defenses

📍 Guidelines paper

Detect and Avoid Flawed Evaluations

- **Problem**: formal evaluations do not scale, adversarial robustness evaluated mostly empirically, via gradient-based attacks
- **Gradient-based attacks can fail:** many flawed evaluations have been reported, with defenses easily broken by adjusting/fixing the attack algorithms



Papernot et al. 2016 **Defensive Distillation** (S&P)
Meng et al. 2016 **MagNet defense** (CCS)
Buckman et al. 2016 **Thermometer Encoding** (ICLR)
Roth et al. 2019 **Odds are odd** (ICML)
Pang et al. 2019 **Ensemble diversity** (PMLR)
Yu et al. 2019 **Turning weakness into strength** (NeurIPS)
Xiao et al. 2020 **k-Winner Take All** (ICLR)

Carlini et al. 2017 **Bypassing ten detection methods** (AISec)
Carlini et al. 2017 **MagNet Not Robust** (arXiv)
Athalye et al. 2018 **Obfuscated gradients give false sense of security** (ICML)
Carlini et al. 2019 **Evaluating Adversarial Robustness** (arXiv)
Tramér et al. 2020 **Adaptive Attacks** (NeurIPS)

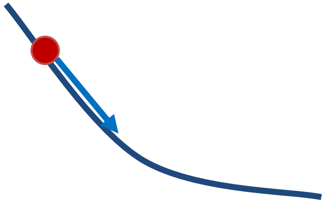
○ Proposed defenses
✖ Broken defenses
📍 Guidelines paper

Pintor, Biggio et al., *Indicators of Attack Failure:* …, NeurIPS 2022

Example: Gradient Obfuscation

When GD works

Smooth function: linear approximation works

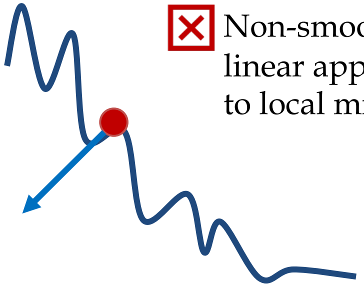


When GD does not work

☒ Zero gradients: impossible to find adversarial direction

Check gradient norm

☒ Non-smooth function: linear approximation leads to local minima



Check variability of loss landscape

Pintor, Biggio et al., *Indicators of Attack Failure:* …, NeurIPS 2022

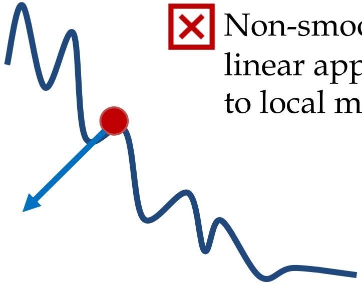
Example: Gradient Obfuscation

When GD does not work

☒ Zero gradients: impossible to find adversarial direction

Check gradient norm

☑ Change loss function



☒ Non-smooth function: linear approximation leads to local minima

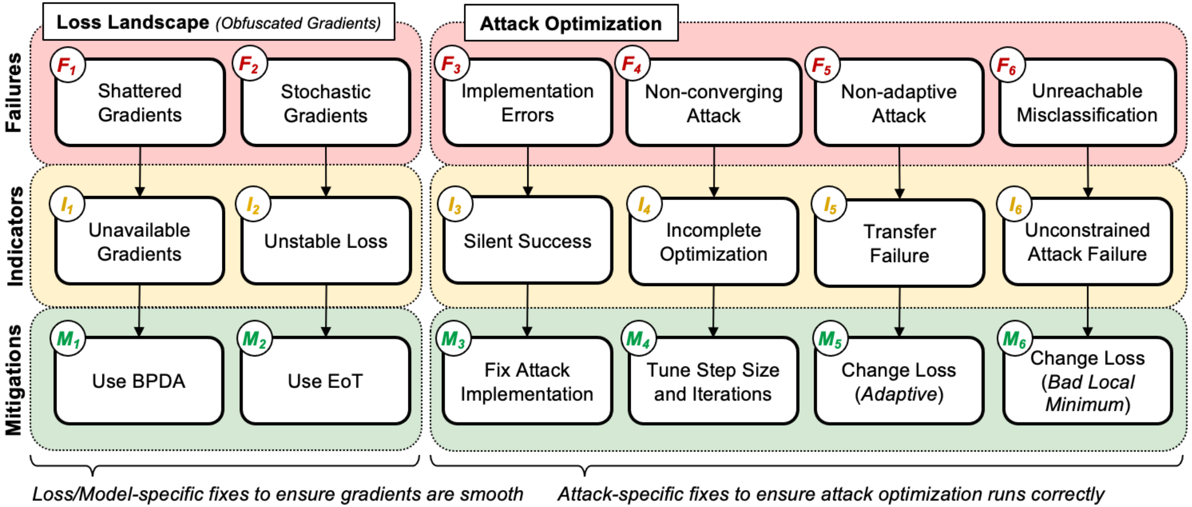
Check variability of loss landscape

☑ Use smooth approximation

Pintor, Biggio et al., *Indicators of Attack Failure:* …, NeurIPS 2022

Detect and Avoid Flawed Evaluations

- **Problem**: formal evaluations do not scale, adversarial robustness evaluated mostly empirically, via gradient-based attacks
- **Gradient-based attacks can fail:** many flawed evaluations have been reported, with defenses easily broken by adjusting/fixing the attack algorithms



Loss Landscape *(Obfuscated Gradients)*

Attack Optimization

Failures

F_1 Shattered Gradients

F_2 Stochastic Gradients

F_3 Implementation Errors

F_4 Non-converging Attack

F_5 Non-adaptive Attack

F_6 Unreachable Misclassification

Indicators

I_1 Unavailable Gradients

I_2 Unstable Loss

I_3 Silent Success

I_4 Incomplete Optimization

I_5 Transfer Failure

I_6 Unconstrained Attack Failure

Mitigations

M_1 Use BPDA

M_2 Use EoT

M_3 Fix Attack Implementation

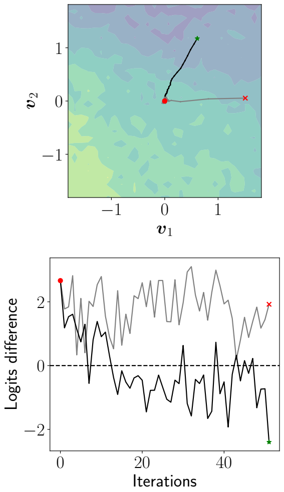
M_4 Tune Step Size and Iterations

M_5 Change Loss *(Adaptive)*

M_6 Change Loss *(Bad Local Minimum)*

Loss/Model-specific fixes to ensure gradients are smooth

Attack-specific fixes to ensure attack optimization runs correctly

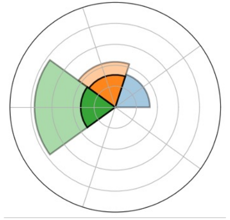


@maurapintor

Pintor, Biggio et al., *Indicators of Attack Failure:* …, NeurIPS 2022

Experiments

k-Winners
Take All

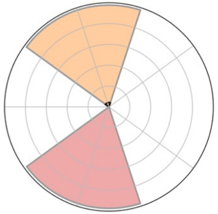


Robust Accuracy

58% → 36% → **6%**

Original evaluation → Fix implementation → Change loss function

Distillation

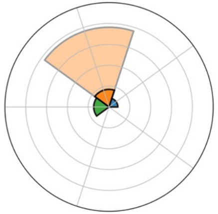


Robust Accuracy

94% → **0%**

Original evaluation → Change loss function

Ensemble
Diversity

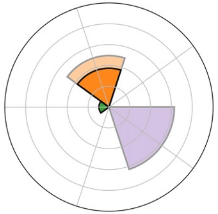


Robust Accuracy

38% → 36% → **9%**

Original evaluation → Fix implementation → Tune hyperparameters

Turning a
Weakness into
a Strength



Robust Accuracy

35% → **0%**

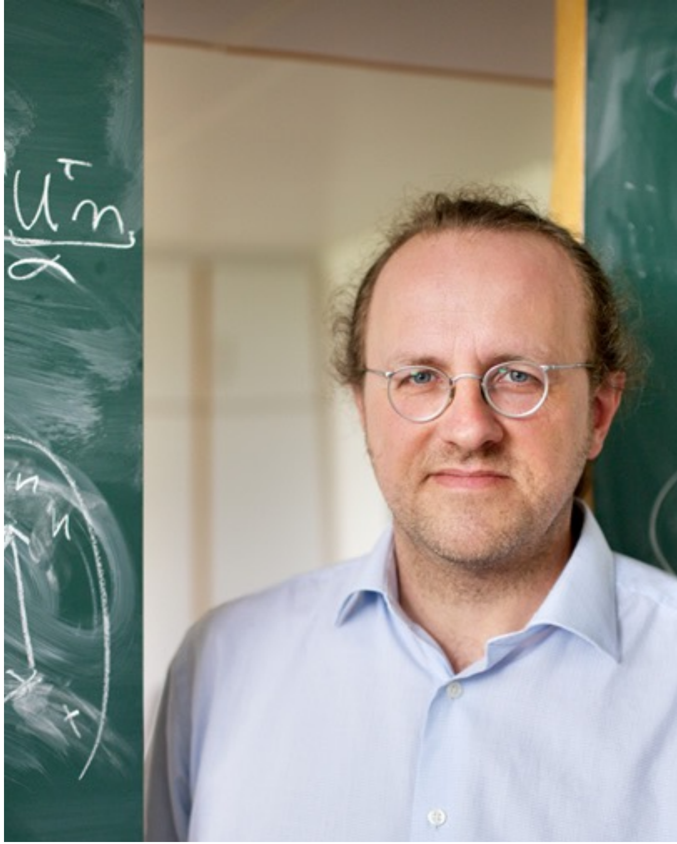
Original evaluation → Perform adaptive attack

🟦 I_1: Silent Success 🟧 I_2: Break-Point Angle 🟩 I_3: Increasing Loss
🟥 I_4: Zero Gradients 🟪 I_5: Non-transferability

Pintor, Biggio et al., *Indicators of Attack Failure:* …, NeurIPS 2022

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



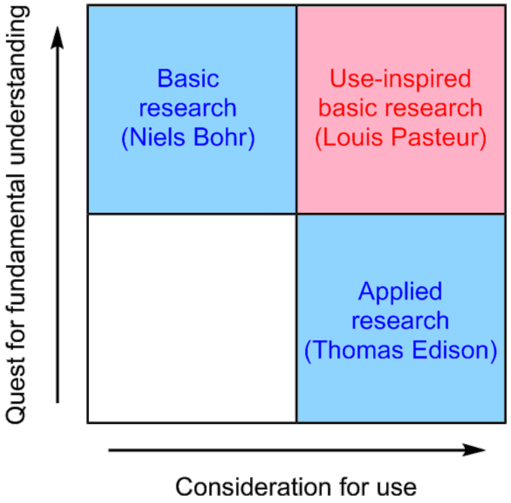
Bernhard Schölkopf

Director, Max Planck Institute, Tuebingen, Germany

What's Next?

Use-Inspired Basic Research Questions from the Pasteur's Quadrant

- Studying ML Security may help understand and debug ML models... but
- ... can we use MLSec to help solve some of today's industrial challenges?
  - To improve robustness/accuracy over time, requiring less frequent retraining
  - To detect OOD examples and provide reliable predictions (confidence values)
  - To improve maintainability and interpretability of deployed models (update procedures)
  - To learn reliably from noisy/incomplete labeled datasets
  - ...
- **Challenge:** to build more reliable and practical ML models using MLSec / AdvML



Practical session!

https://github.com/maurapintor/ARTISAN

University of
Cagliari, Italy

Pattern Recognition
and Applications Lab

Thanks!

Maura Pintor
maura.pintor@unica.it

Open Course on MLSec
https://github.com/unica-mlsec/mlsec

Software Tools
https://github.com/pralab

Machine Learning Security Seminars
https://www.youtube.com/c/MLSec

Indiscriminate (DoS) Poisoning Attacks

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

A Deliberate Poisoning Attack?

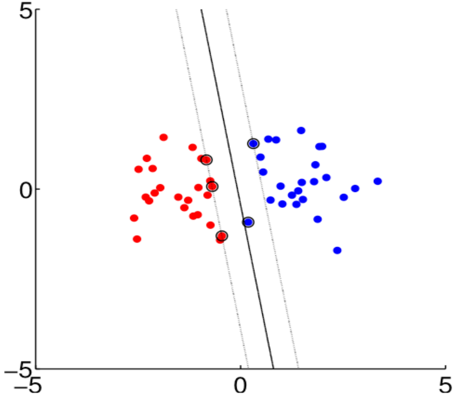
Microsoft deployed **Tay**, and **AI chatbot** designed to talk to youngsters on Twitter

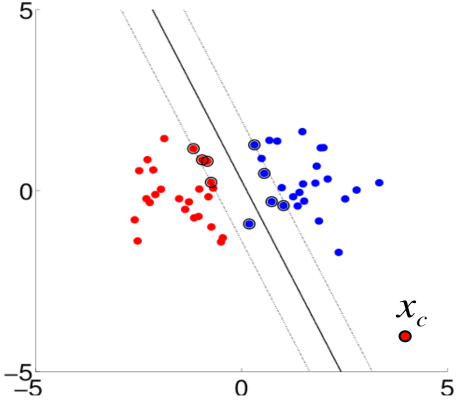
But after 16 hours the chatbot was shut down since it started to raise racist and offensive comments.

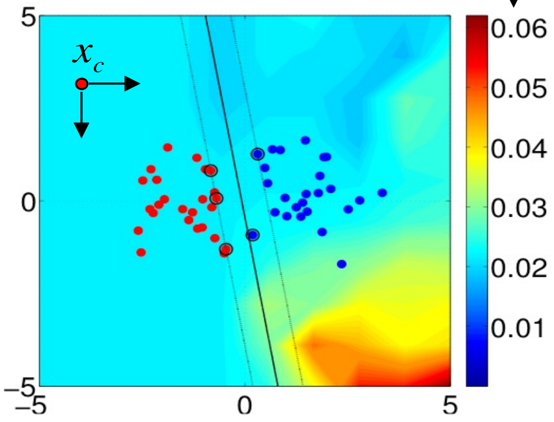
[http://exploringpossibilityspace.blogspot.it/2016/03/poor-software-qa-is-root-cause-of-tay.html]

Denial-of-Service Poisoning Attacks

- **Goal**: to maximize classification error by 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 Problem

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

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

Loss estimated on validation data
(no attack points!)

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

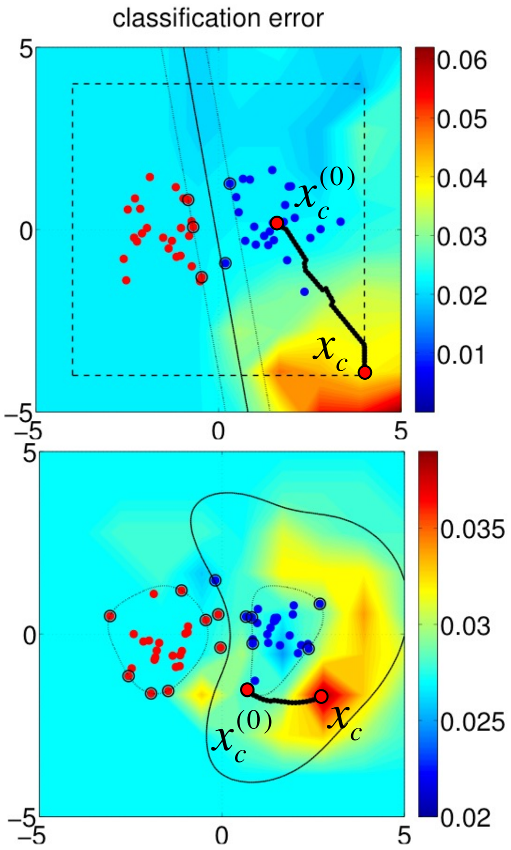
Algorithm is trained on surrogate data
(including the attack point)

Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012
Xiao, Biggio et al., Is feature selection secure against training data poisoning? ICML, 2015
Munoz-Gonzalez, Biggio 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

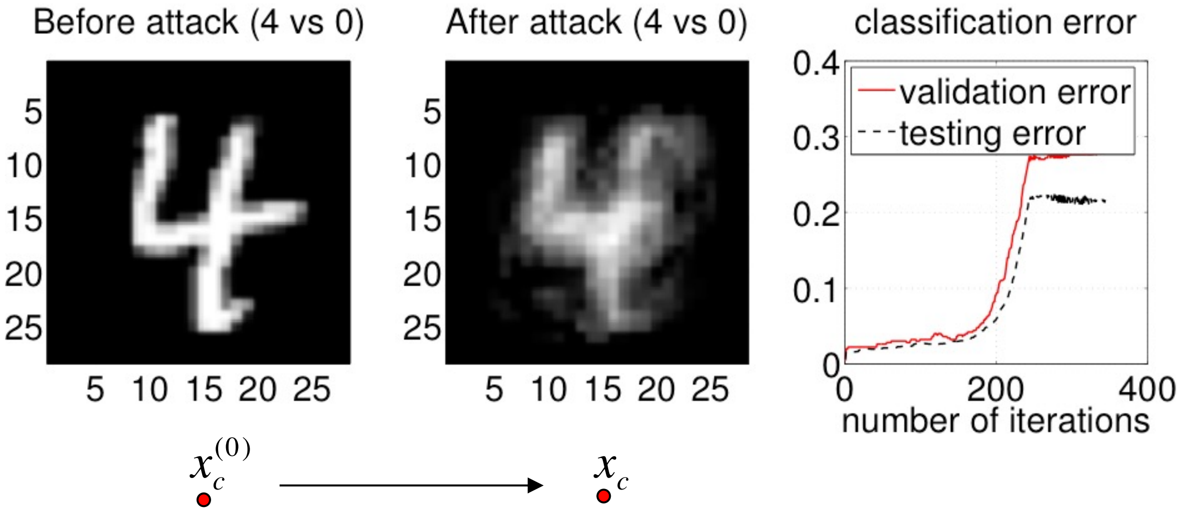


Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012
Xiao, Biggio, Roli et al., Is feature selection secure against training data poisoning? ICML, 2015
Demontis, Biggio et al., Why do Adversarial Attacks Transfer? USENIX 2019

Experiments on MNIST digits

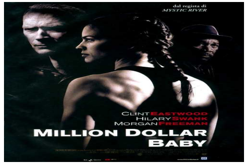
Single-point attack

- 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

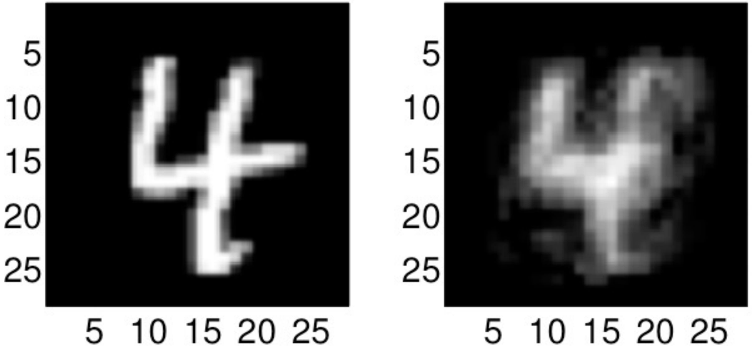
Countering Poisoning Attacks



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 (B. Biggio et al., MCS 2011)
     - Reject-On-Negative-Impact (RONI) defense (B. Nelson et al., LEET 2008)
  2. **Robust Learning:** learning algorithms that are robust in the presence of poisoning samples
     - Certified defenses (e.g., J. Steinhardt, P. W. Koh, and P. Liang, NeurIPS 2017)

Backdoor Attacks

Attacks against Machine Learning

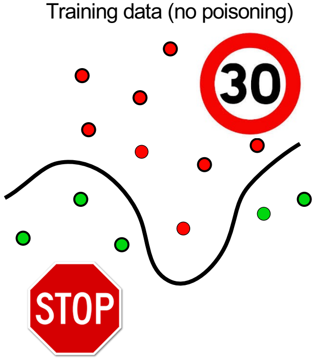
Attacker's Goal

	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*	-

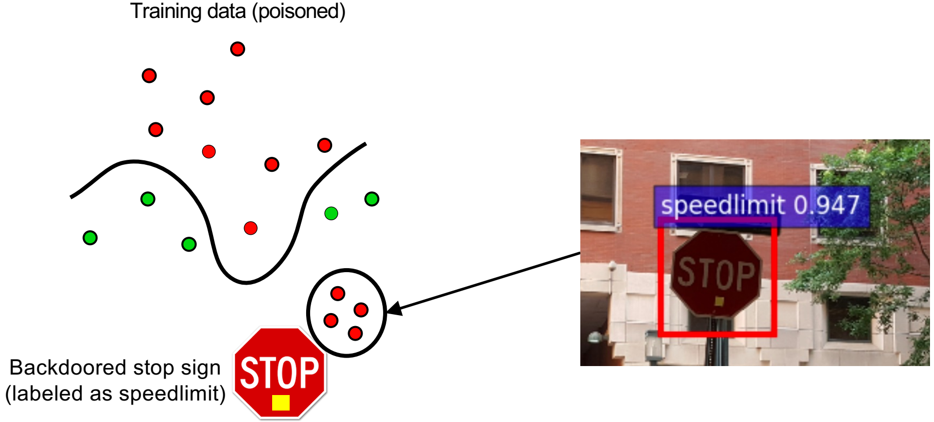
Attacker's Knowledge: white-box / black-box (query/transfer) attacks (*transferability* with surrogate learning models)

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

Backdoor Poisoning Attacks



Training data (no poisoning)



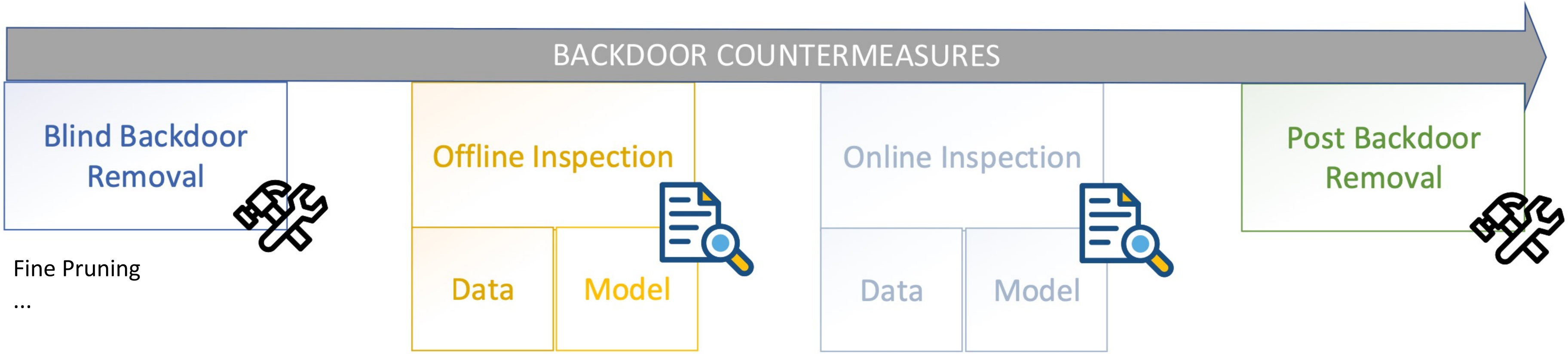
Training data (poisoned)

Backdoored stop sign
(labeled as speedlimit)

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

Defending against Backdoor Poisoning Attacks



BACKDOOR COUNTERMEASURES

Blind Backdoor Removal

Offline Inspection

Data

Model

Online Inspection

Data

Model

Post Backdoor Removal

Fine Pruning
...

Spectral Signature
...
Neural Cleanse
DeepInspect
...

SentiNet (GradCAM)
ABS
NIC
...

Gao et al., *Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review*, arXiv 2007.10760

Other Attacks on Machine Learning Models

Attacks against Machine Learning

Attacker's Goal

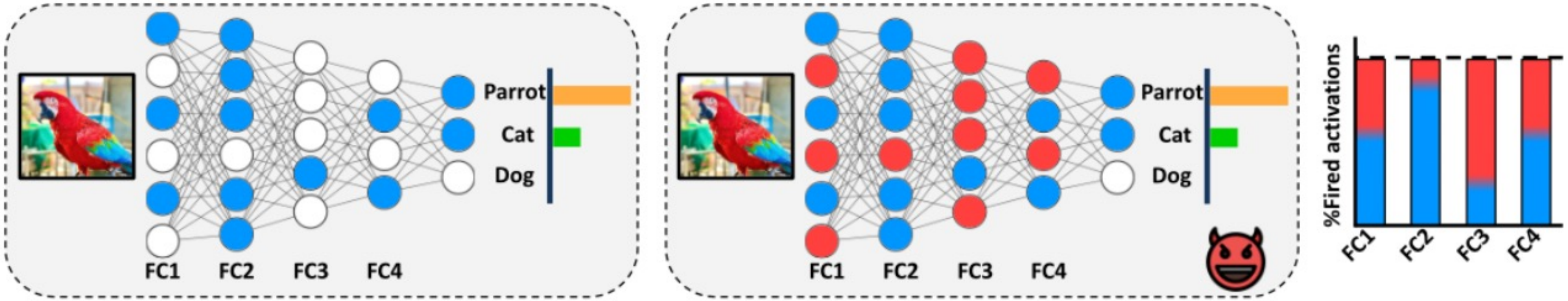
	Misclassifications that do not compromise normal system operation	Misclassifications that compromise normal system operation	Querying strategies that reveal confidential information on the learning model or its users
Attacker's Capability	**Integrity**	**Availability**	**Privacy / Confidentiality**
Test 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 Poisoning

- Attacks aimed at increasing energy consumption of DNN models deployed on embedded hardware systems

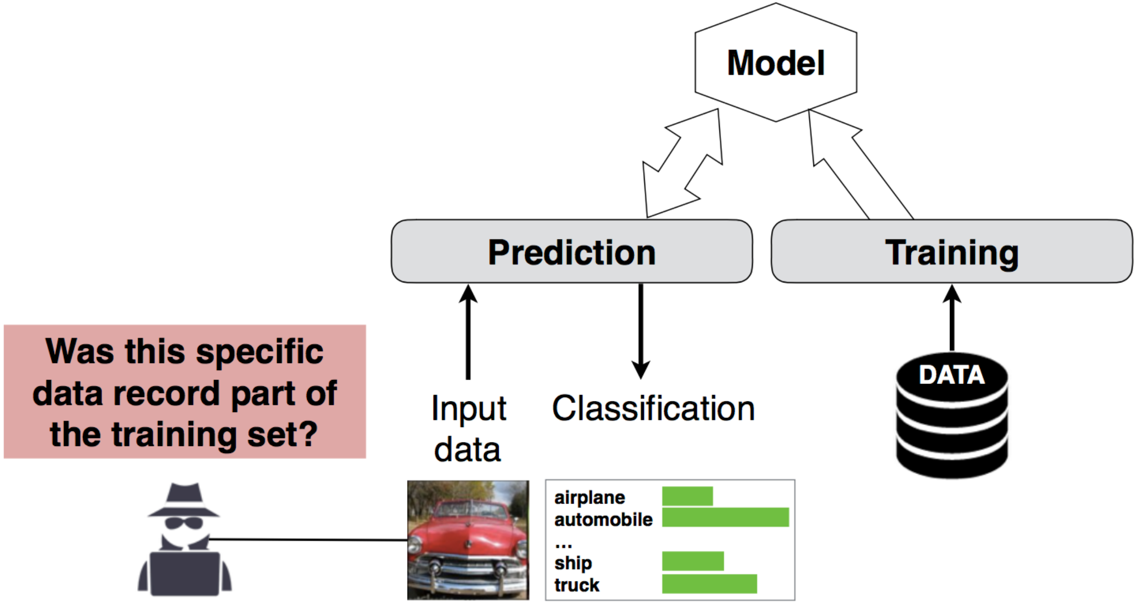


Shumailov et al., *Sponge Examples...*, EuroSP 2021
Cinà, Biggio et al., *Sponge Poisoning...*, arXiv 2022

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



Model

Prediction

Training

Was this specific data record part of the training set?

Input data

Classification

DATA

airplane
automobile
...
ship
truck

Bosch *AI Shield* against Model Stealing/Extraction 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*
- 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

Machine Learning **Defenses** in a Nutshell

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 Membership inference
Training data	Backdoor/Targeted poisoning (to allow subsequent intrusions)	Indiscriminate (DoS) poisoning 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