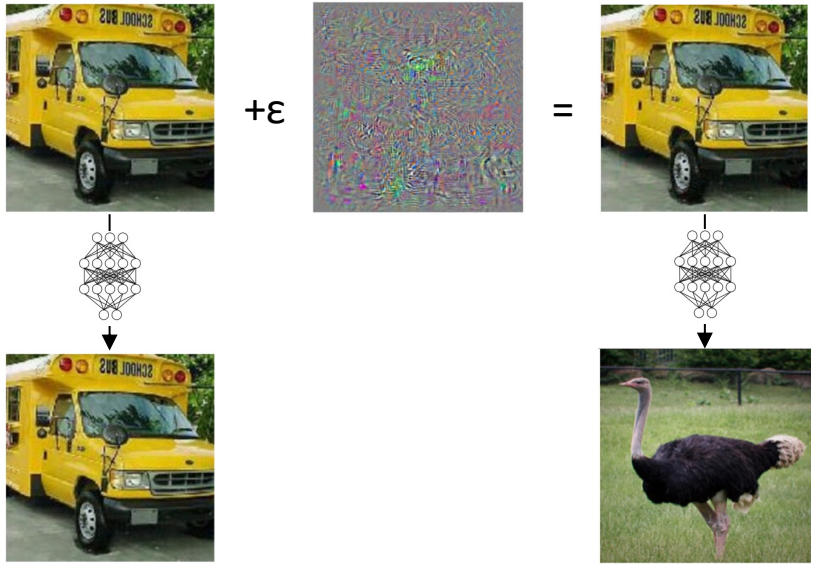
Adversarial Machine Learning

Battista Biggio

The Elephant in the Room: *Adversarial Examples*

- AI/ML successful in many applications
  - Computer Vision
  - Speech Recognition
  - Cybersecurity
  - Healthcare

- ... but extremely *fragile* against *adversarial examples*
  - Carefully-perturbed inputs that mislead classification



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 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/

Attacks against AI are Pervasive!

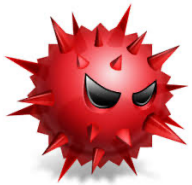
Sharif et al., *Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,* ACM CCS 2016

"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/

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

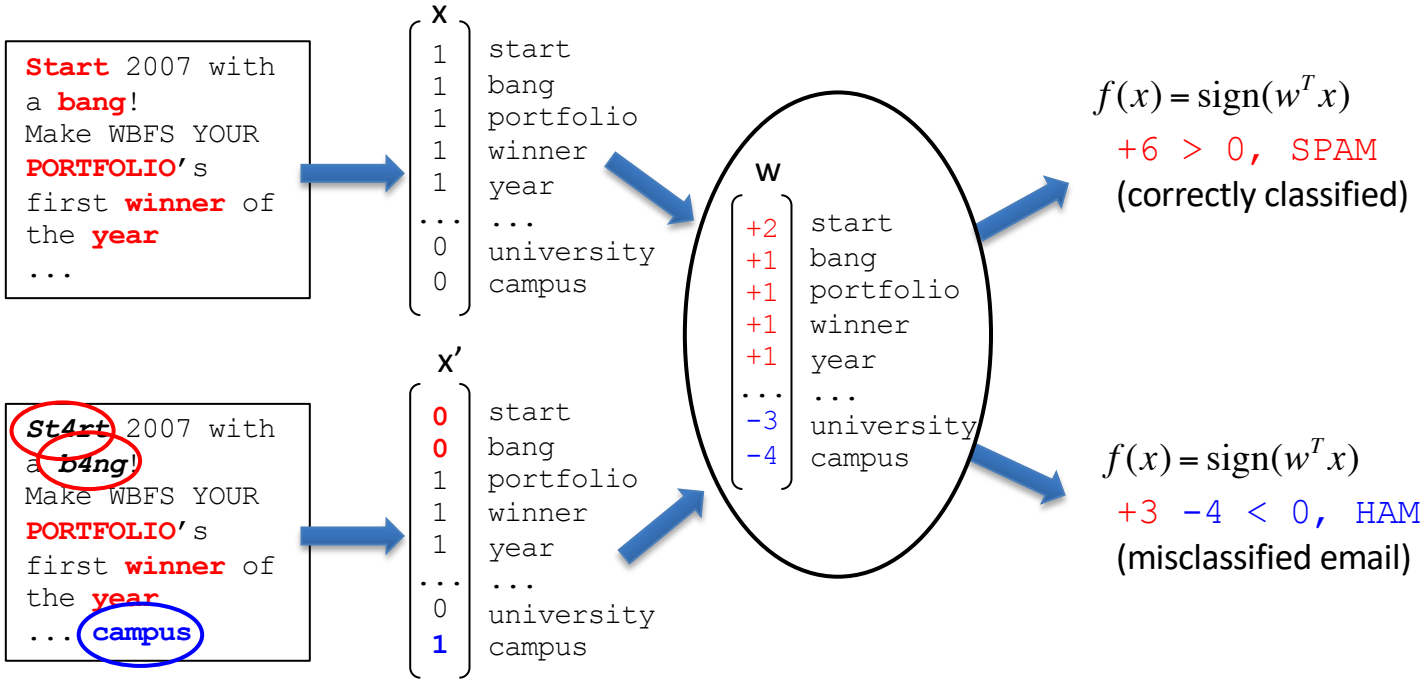


- Demetrio, Biggio, Roli et al., *Adversarial EXEmples: ...*, ACM TOPS 2021
- Demetrio, Biggio, Roli et al., *Functionality-preserving black-box optimization of adversarial windows malware*, IEEE TIFS 2021
- Demontis, Biggio, Roli et al., *Yes, Machine Learning Can Be More Secure!...*, IEEE TDSC 2019

How Do These Attacks Work?

Evasion of Linear Classifiers

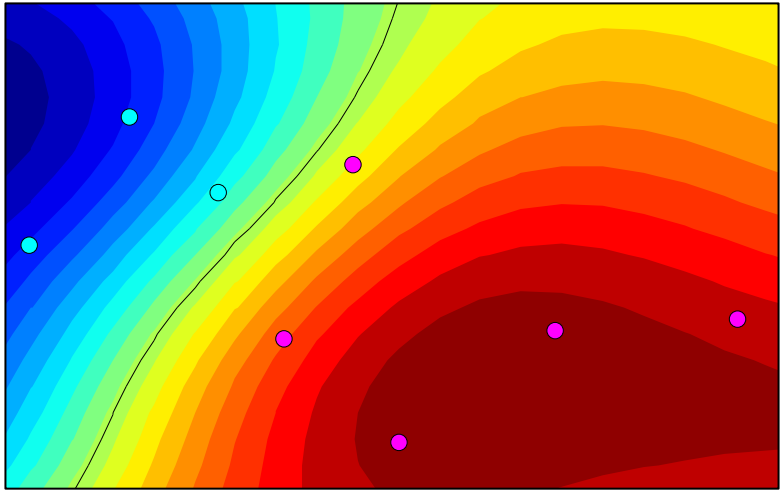
- **Problem:** how to evade a linear (trained) classifier?



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 ML at Test Time

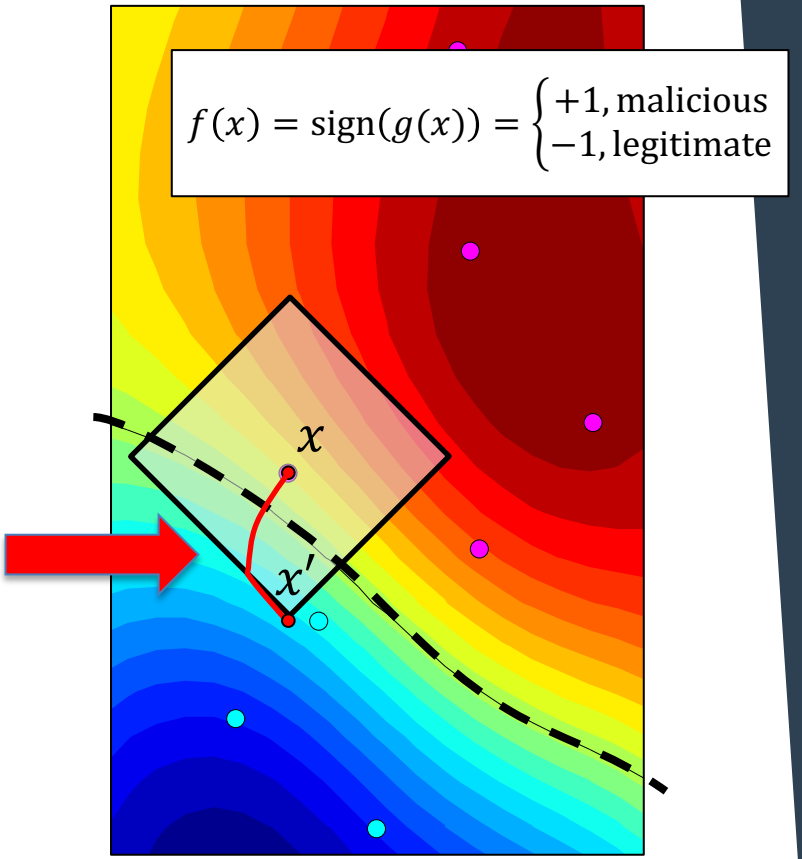
Biggio et al., Evasion Attacks Against Machine Learning at Test Time, **ECML 2013**

- **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



Computing Descent Directions

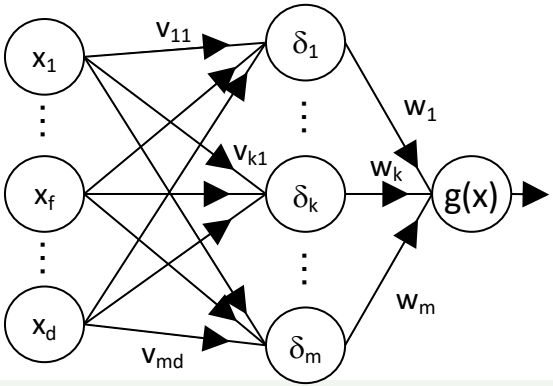
Biggio et al., Evasion Attacks Against Machine Learning at Test Time? ECML 2013

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



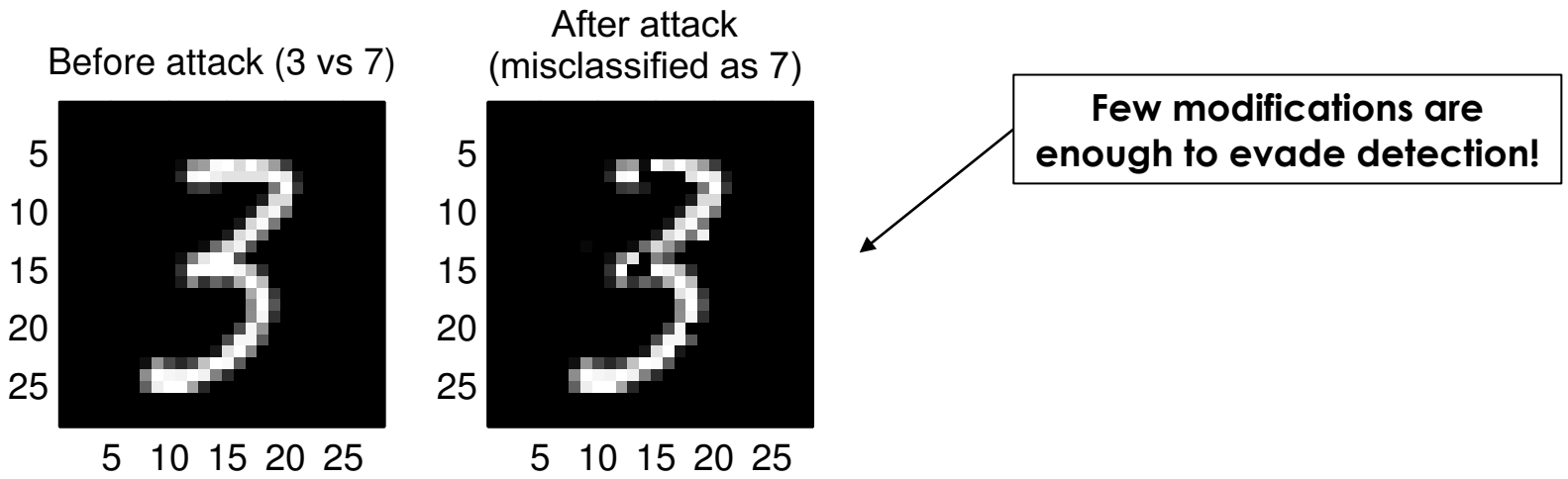
$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^{m} w_k \delta_k(x)\right)\right]^{-1}$$

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

An Example on Handwritten Digits

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

- Nonlinear SVM (RBF kernel) to discriminate between '3' and '7'
- **Features**: gray-level pixel values (28 x 28 image = 784 features)



Experiments on PDF Malware Detection

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

- **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)!

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

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

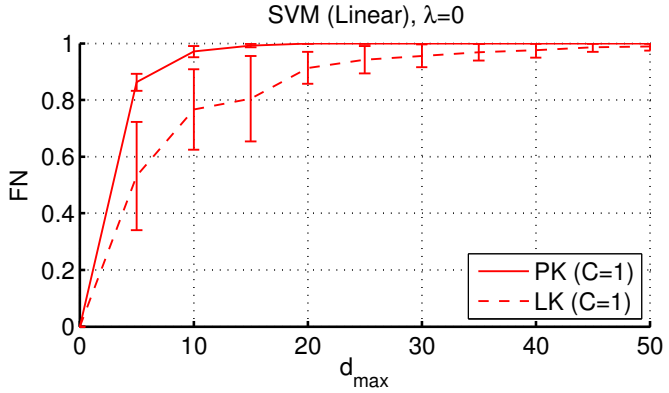
$$x \preceq x'$$

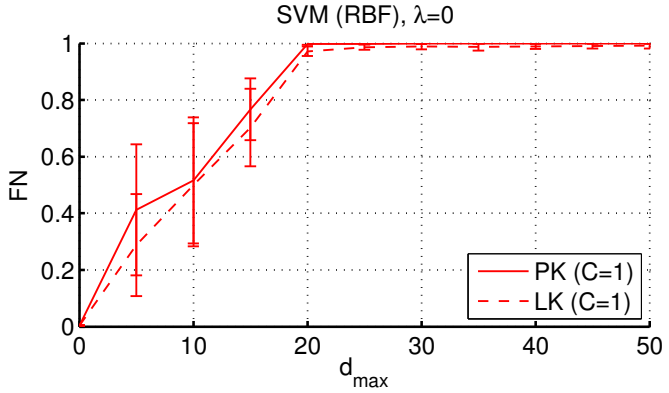
Experiments on PDF Malware Detection

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

- **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)

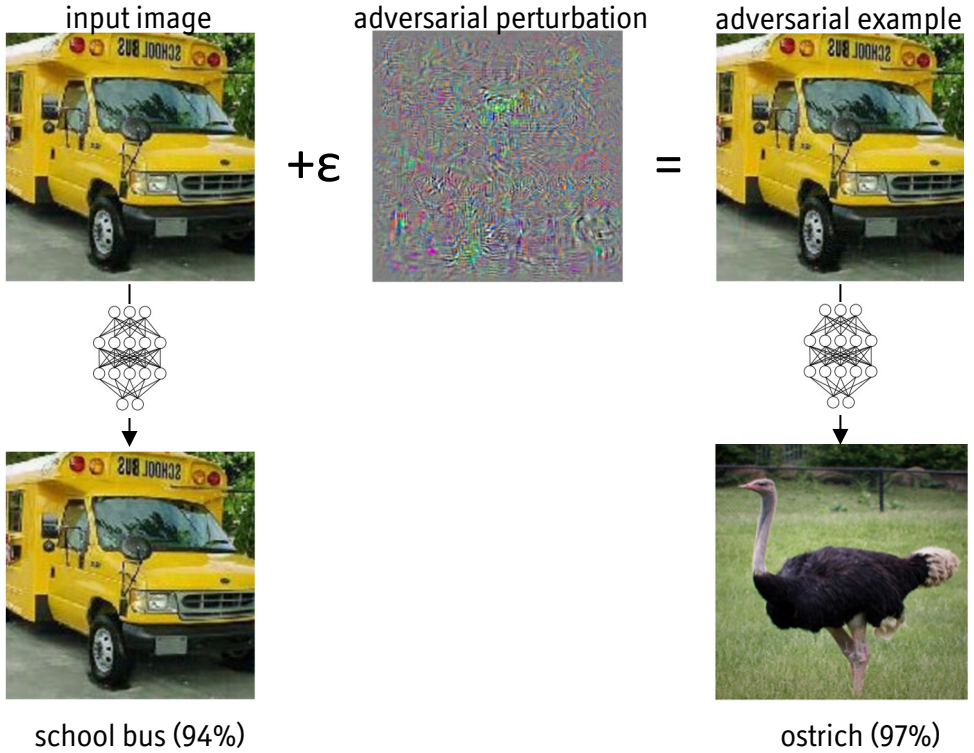




Adversarial Examples against Deep Neural Networks

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

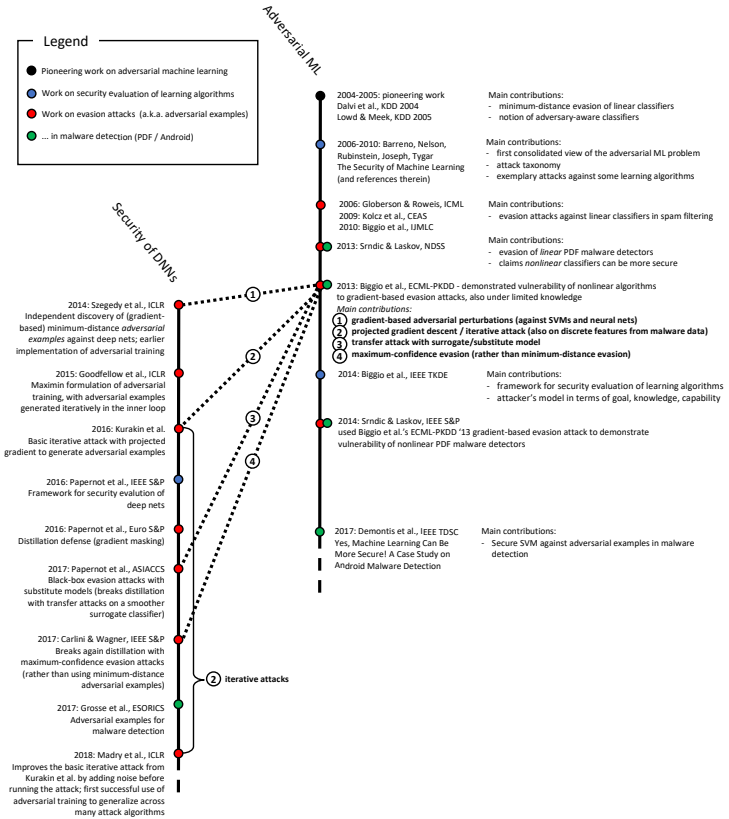
- Szegedy et al. (2014) independently developed gradient-based attacks against DNNs
- They were investigating model interpretability, trying to understand at which point a DNN prediction changes
- They found that the minimum perturbations required to trick DNNs were really small, even imperceptible to humans



Timeline of Learning Security

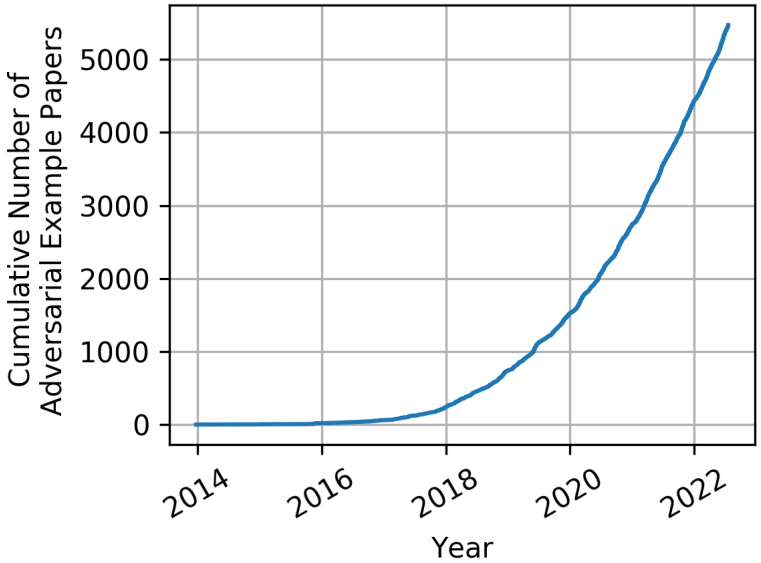
Biggio and Roli, **Wild Patterns**: *Ten Years After The Rise of Adversarial Machine Learning*, Pattern Recognition, 2018

2021 Best Paper Award and Pattern Recognition Medal



ML Security Boomed…

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

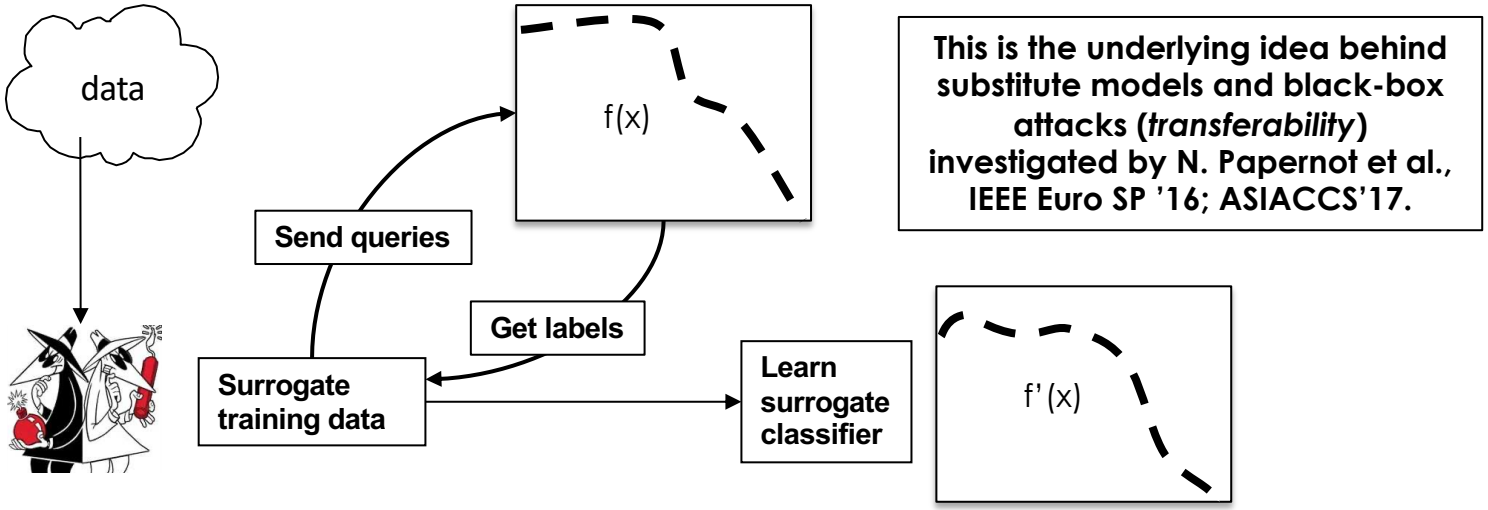


From White-Box to Black-Box Attacks

From White-box to Black-box *Transfer* Attacks

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

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



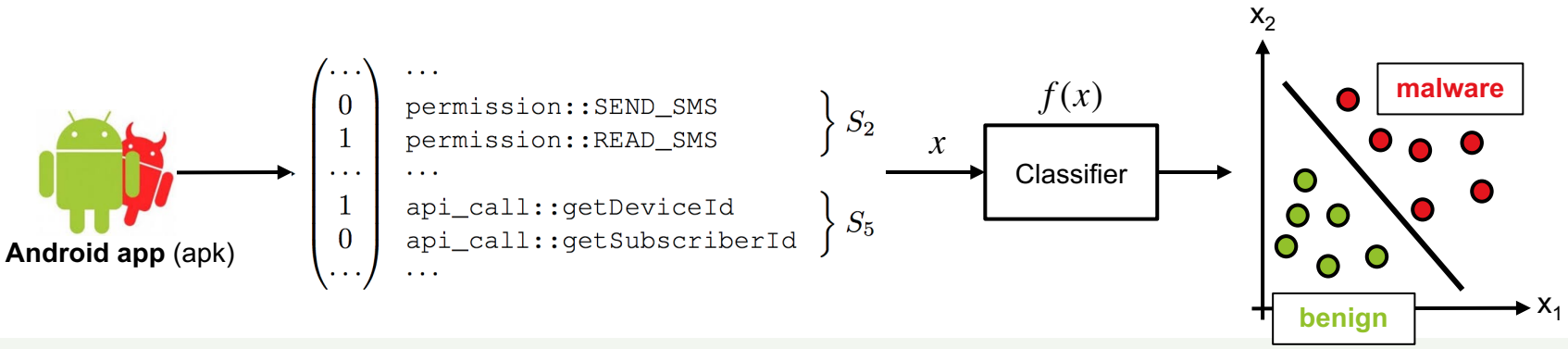
This is the underlying idea behind substitute models and black-box attacks (*transferability*) investigated by N. Papernot et al., IEEE Euro SP '16; ASIACCS'17.

Results on Android Malware Detection

Demontis, Biggio et al., IEEE TDSC 2019

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

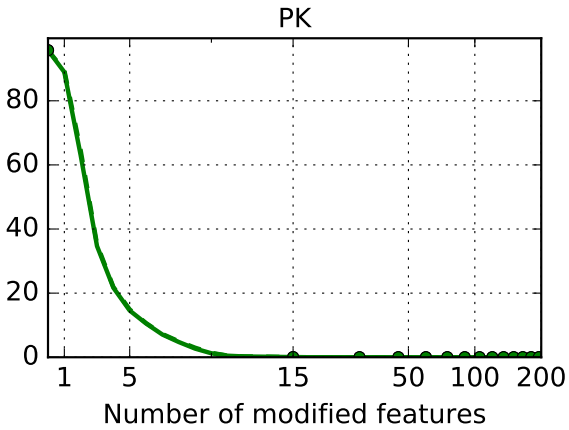
Feature sets		
`manifest`	S_1	Hardware components
	S_2	Requested permissions
	S_3	Application components
	S_4	Filtered intents
`dexcode`	S_5	Restricted API calls
	S_6	Used permission
	S_7	Suspicious API calls
	S_8	Network addresses

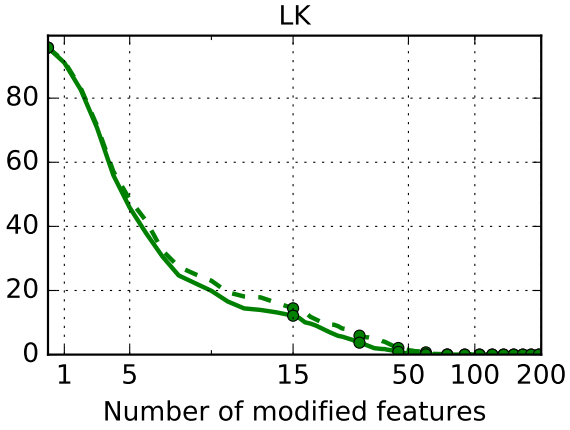


Results on Android Malware Detection

Demontis, Biggio et al., IEEE TDSC 2019

- **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

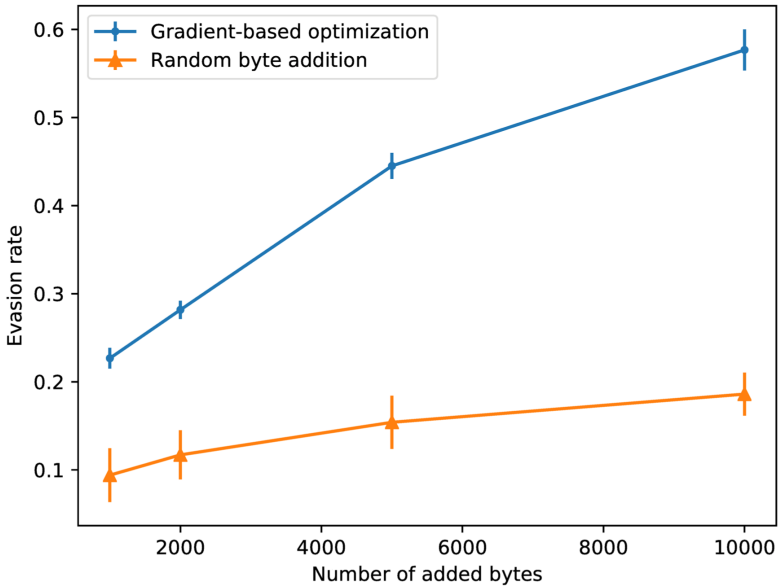


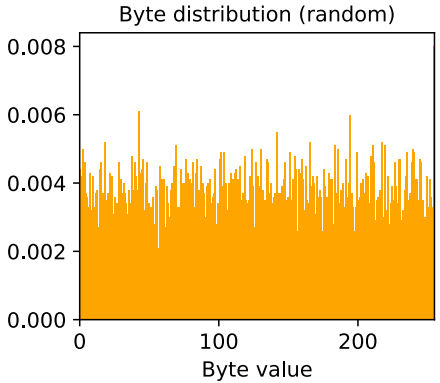


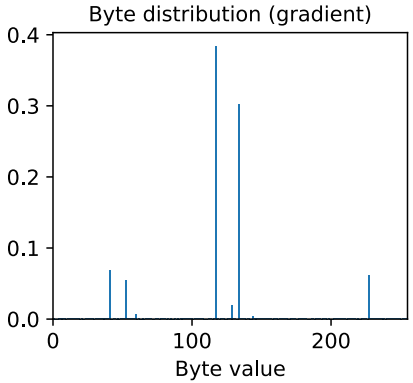
Attacks on EXE Malware

Evasion of Deep Networks for EXE Malware Detection

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



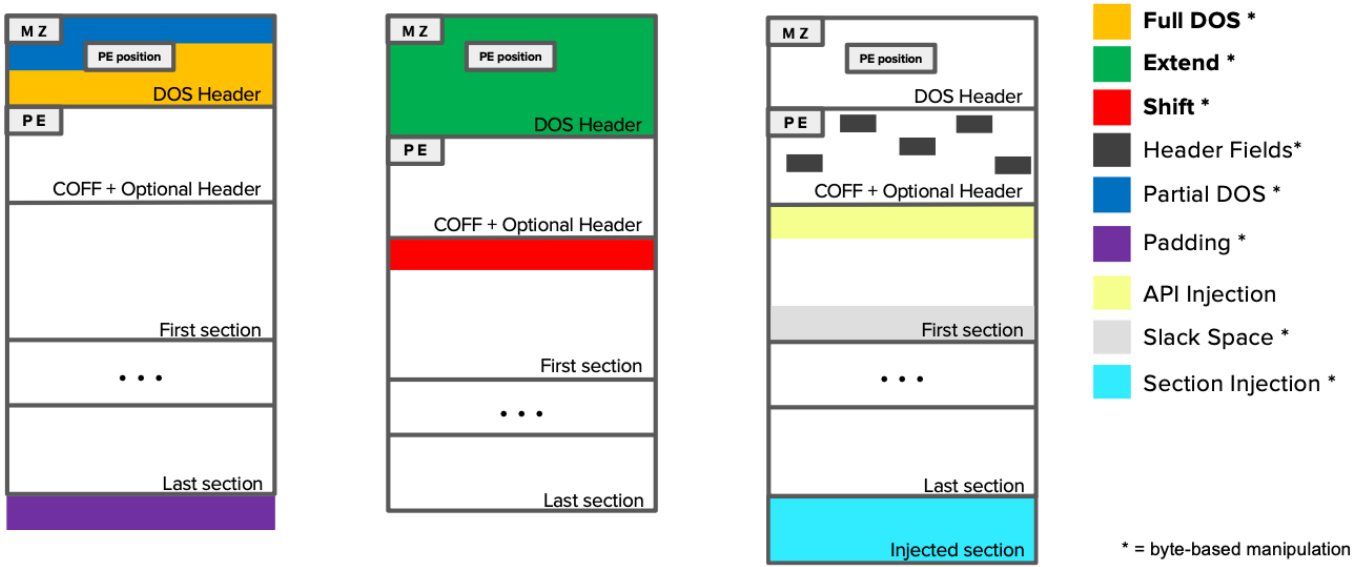




Kolosniaji, Biggio et al., *Adversarial Malware Binaries*, EUSIPCO2018
Demetrio, Biggio et al., *Explaining Vulnerability of DL*, ITASEC 2019

Adversarial EXEmples: Practical Attacks on Machine Learning for Windows Malware Detection

- **Problem-space attacks:** crafting evasive malware programs that preserve functionality!



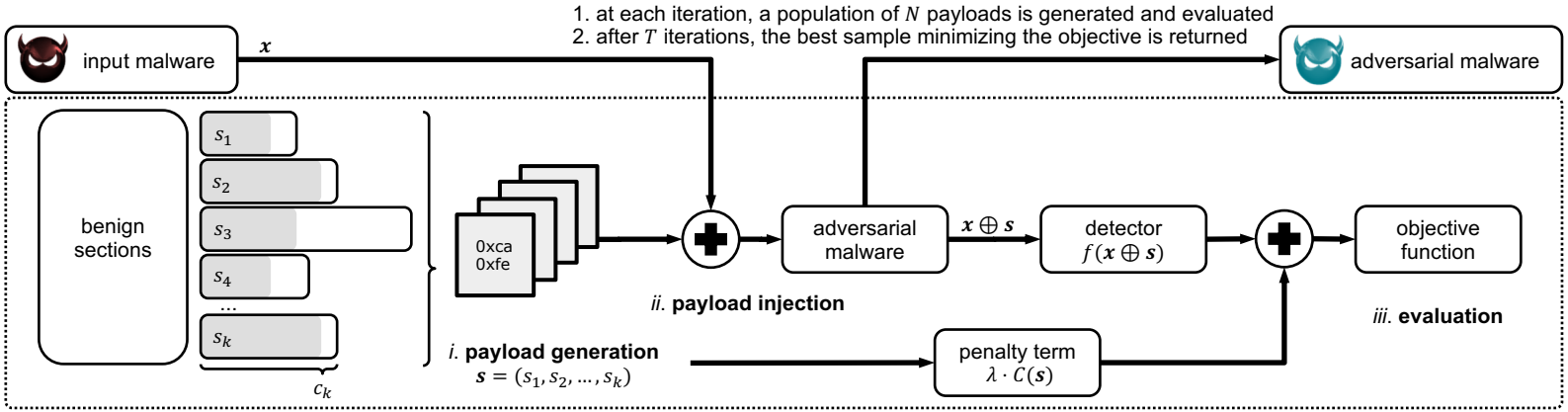
Demetrio, Biggio, et al., Adversarial EXEmples, ACM TOPS 2021
Demetrio, Biggio, et al., Functionality-preserving ..., IEEE TIFS 2021

Black-box Attacks on EXE Malware

Functionality-preserving Black-box Optimization of Adversarial Windows Malware

- Black-box genetic algorithm optimizing the injection of benign sections into malicious PE files

$$s^* = \arg\min_{s \in \mathcal{S}_k} f(x \oplus \boldsymbol{s}) + \lambda \mathcal{C}(\boldsymbol{s})$$
$$\text{subject to} \quad \mathcal{Q}(\boldsymbol{s}) \leq T$$

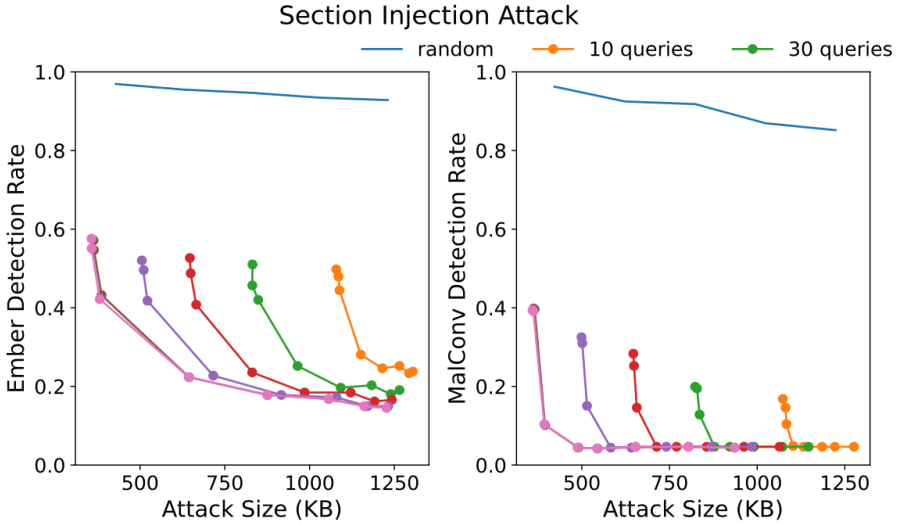


Demetrio, Biggio et al., https://arxiv.org/pdf/2003.13526.pdf

Black-box Attacks on EXE Malware

Functionality-preserving Black-box Optimization of Adversarial Windows Malware

- Our attack bypasses state-of-the-art machine learning-based detectors also with very small payload sizes



- Surprisingly, it also works against some commercial anti-malware solutions available from VirusTotal!

	Malware	Random	Sect. Injection
AV1	93.5%	85.5%	30.5%
AV2	85.0%	78.0%	68.0%
AV3	85.0%	46.0%	43.5%
AV4	84.0%	83.5%	63.0%
AV5	83.5%	79.0%	73.0%
AV6	83.5%	82.5%	69.5%
AV7	83.5%	54.5%	52.5%
AV8	76.5%	71.5%	60.5%
AV9	67.0%	54.5%	16.5%

Detection rates of AV products from VirusTotal, including AVs in the Gartner's leader quadrant. Our **section-injection attack** evades detection with high probability. We are in touch with some AV companies for responsible disclosure of such a vulnerability.

Demetrio al., IEEE TIFS 2021 https://arxiv.org/pdf/2003.13526.pdf

Countering Evasion Attacks

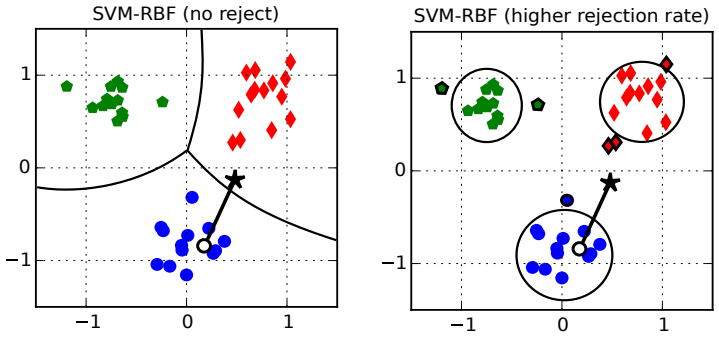
Security Measures against Evasion Attacks

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

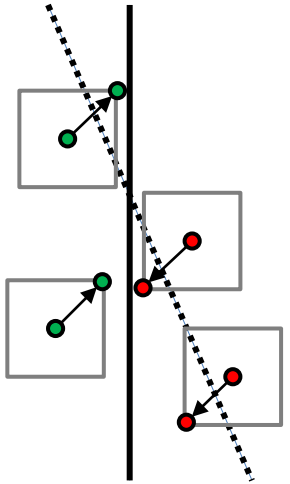


Increasing Input Margin via Robust Optimization

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

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

bounded perturbation!



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

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

dual norm of the perturbation

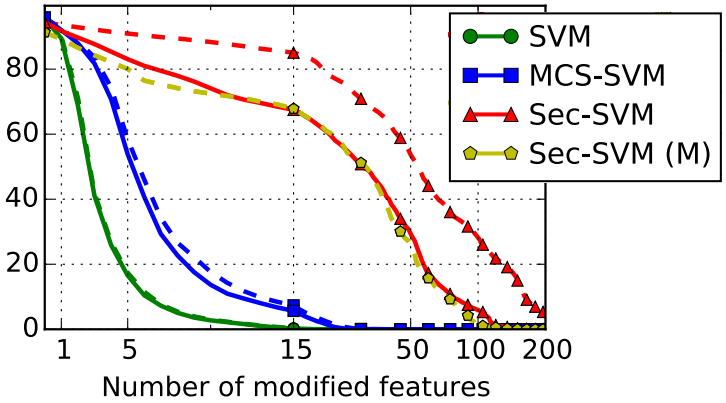
Yes, Machine Learning Can Be More Secure!
A Case Study on Android Malware Detection

- **Infinity-norm regularization** is optimal against **adversarial Android malware** samples
  - Sparse attacks penalize $||\delta||_1$ promoting the manipulation of few features

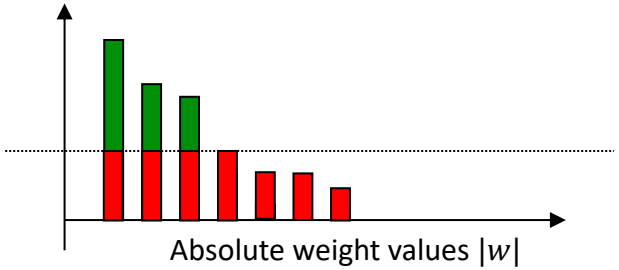
Sec-SVM

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

Experiments on Android Malware



Why? It bounds the maximum absolute weight values!



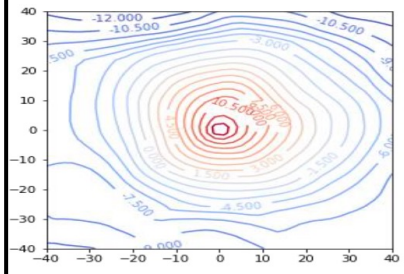
Demontis, Biggio et al., IEEE TDSC 2019

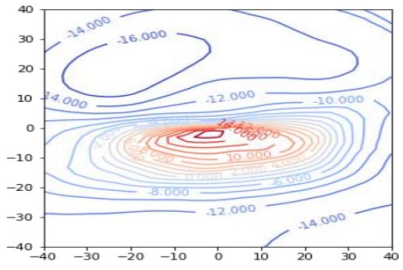
Why Does Robust Optimization Work?

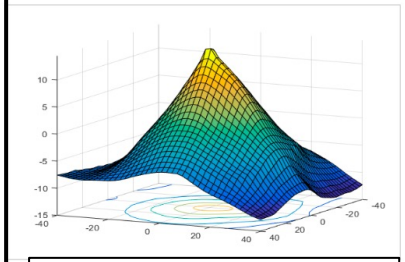
Yu et al., Interpreting and Evaluating NN Robustness, IJCAI 2019

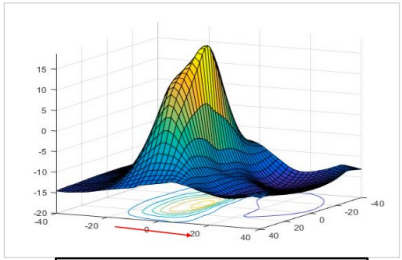
CIFAR-10

Undefended model – Adversarial accuracy: 0.3%





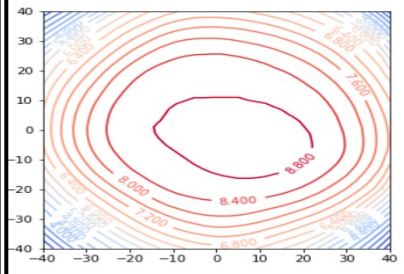


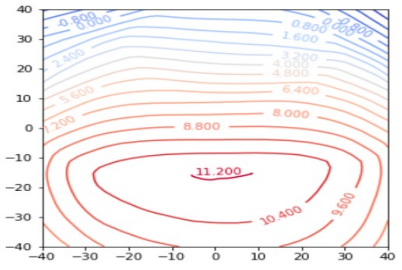


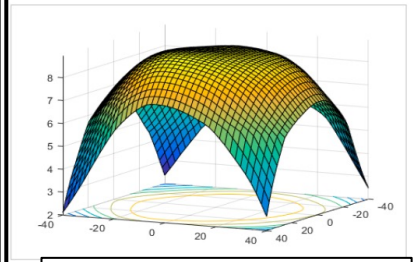
random perturbation

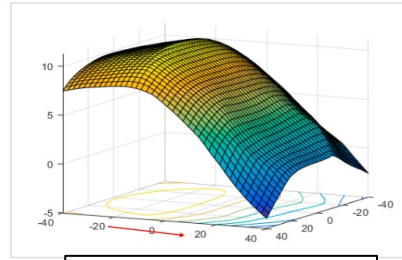
adv. perturbation

Defended model – Adversarial accuracy: 44.7%







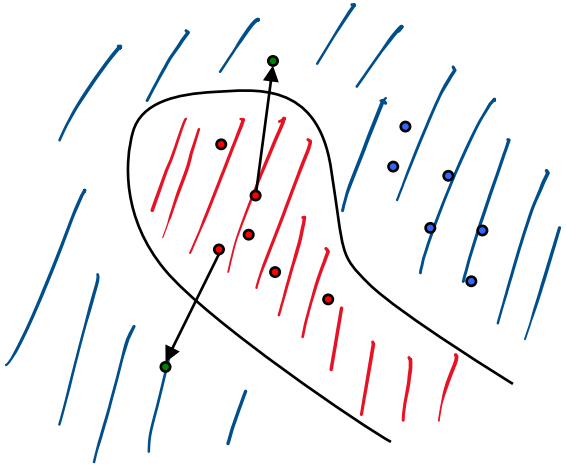


random perturbation

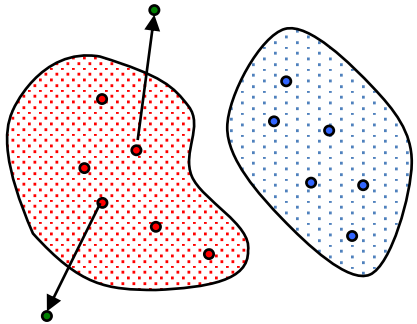
adv. perturbation

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

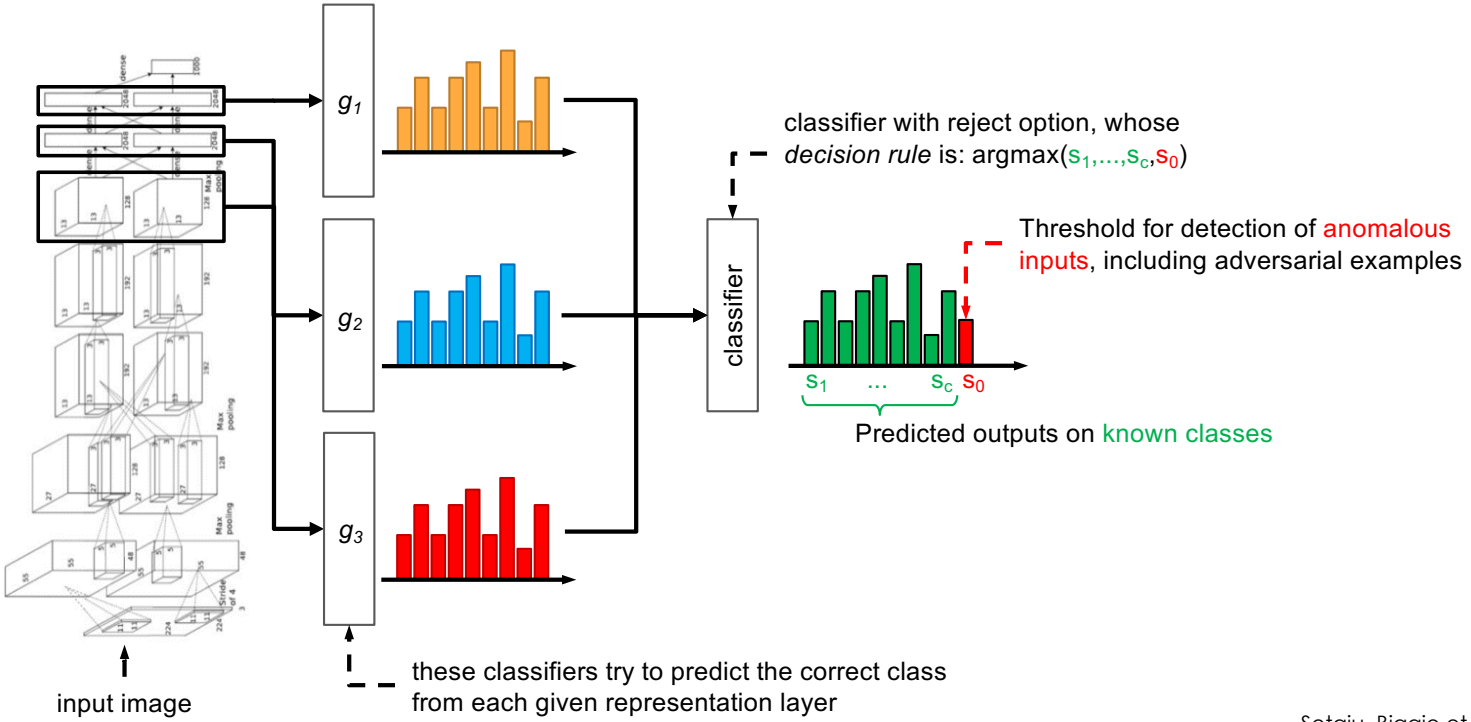


blind-spot evasion
(not even required to mimic the target class)



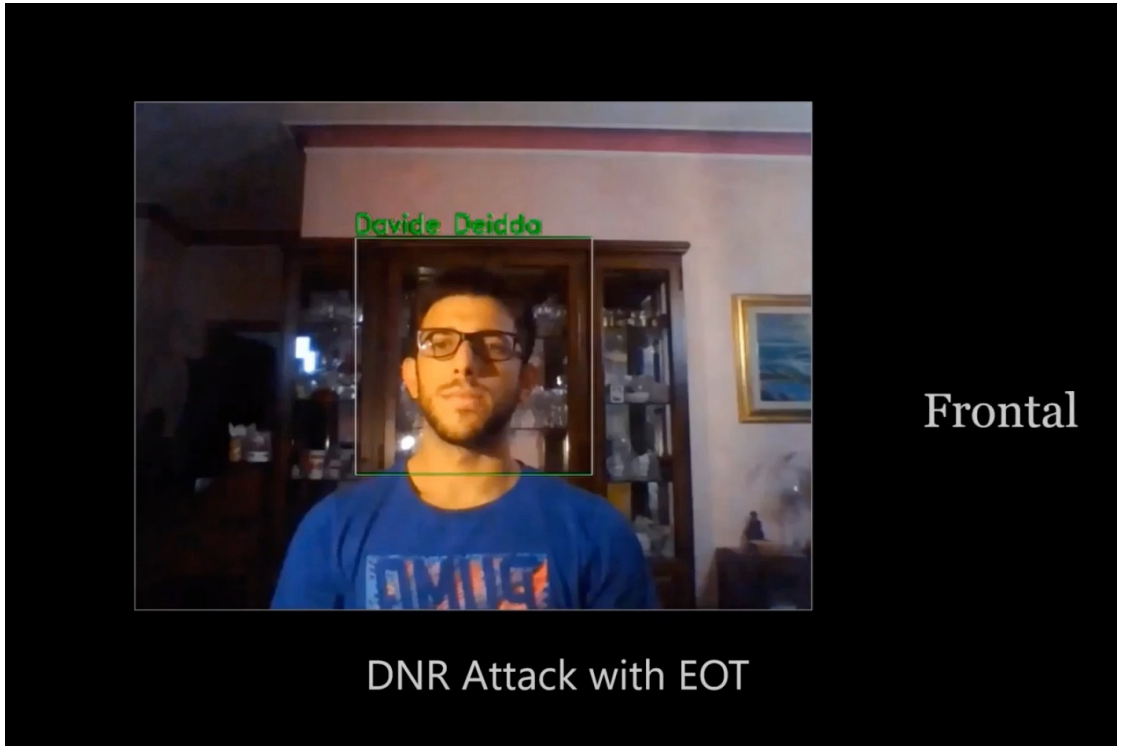
rejection of adversarial examples through enclosing of legitimate classes

Deep Neural Rejection against Adversarial Examples

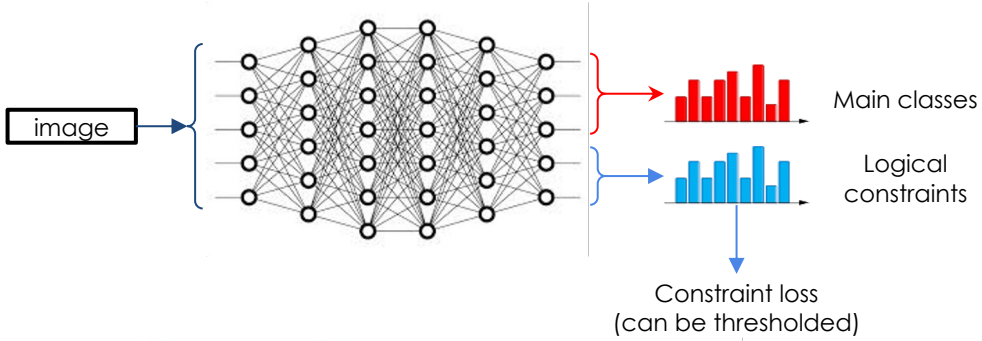


Sotgiu, Biggio et al., EURASIP JIS, 2020
Crecchi, Biggio et al., *FADER:* ..., Neurocomputing 2021

Application Example: DNR against Physical Attacks

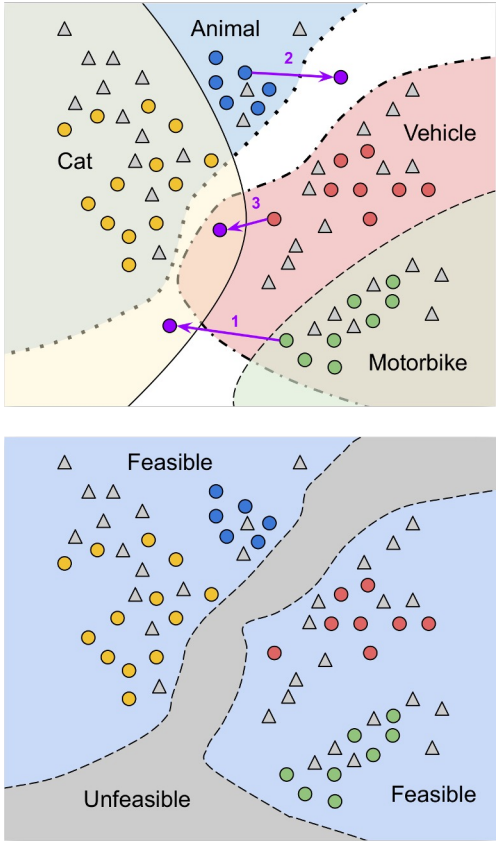


Robust Learning with Domain Knowledge



$$
\begin{aligned}
&\forall x, \quad \mathrm{CAT}(x) \Rightarrow \mathrm{ANIMAL}(x)\,,\\
&\forall x, \quad \mathrm{MOTORBIKE}(x) \Rightarrow \mathrm{VEHICLE}(x)\,,\\
&\forall x, \quad \mathrm{VEHICLE}(x) \Rightarrow \neg\mathrm{ANIMAL}(x)\,,\\
&\forall x, \quad \mathrm{CAT}(x) \vee \mathrm{ANIMAL}(x) \vee \mathrm{MOTORBIKE}(x) \vee \mathrm{VEHICLE}(x)
\end{aligned}
$$

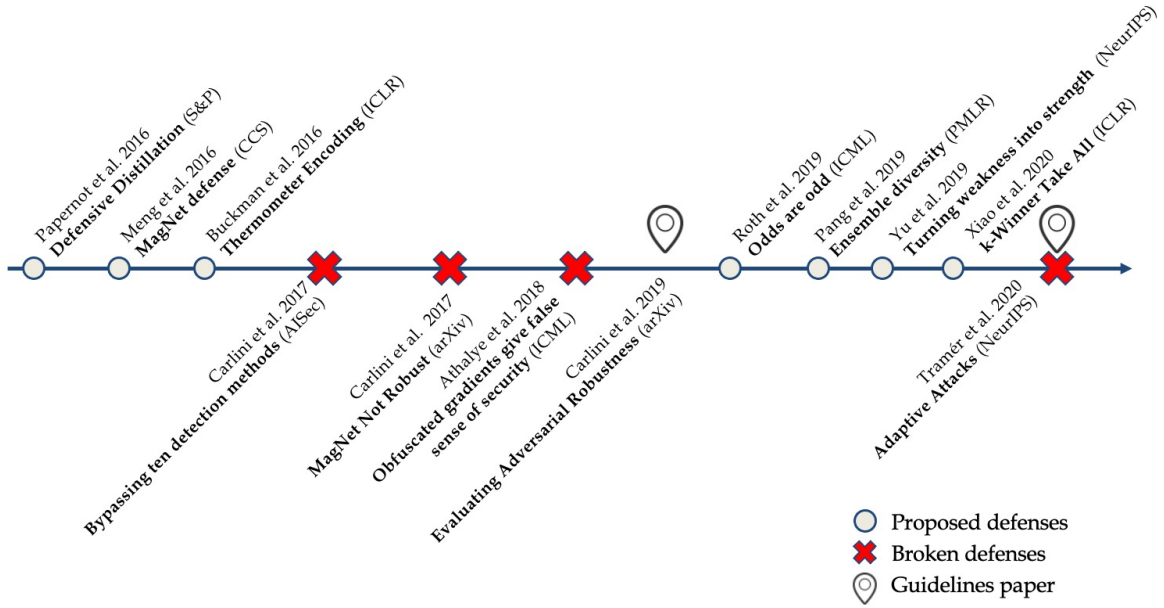
$$
\min_{\mathbf{f}} = \frac{1}{n}\sum_{i=1}^{l} L_y(\mathbf{f}(\mathbf{x}_i), \mathbf{y}_i) + \sum_{j=1}^{l+u}\sum_{h=1}^{m} \lambda_m \cdot L_\phi(\phi_h(\mathbf{f}(\mathbf{x}_j))) + \lambda\|\mathbf{f}\|
$$



Melacci et al., *Domain Knowl. Alleviates Adv. Ex.*, IEEE TPAMI 2021

Detect and Avoid Flawed Evaluations

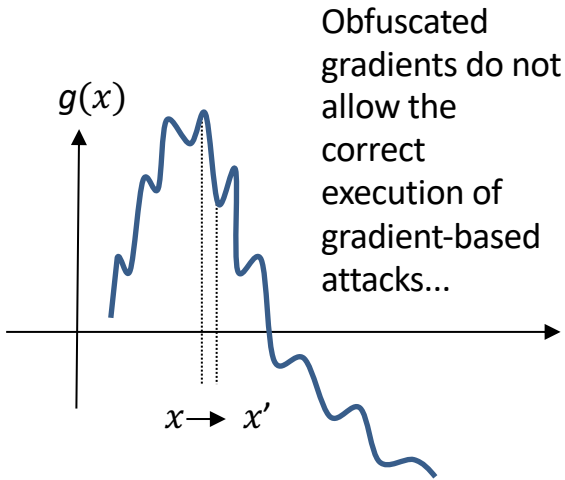
- **Problem**: formal evaluations do not scale, adversarial robustness evaluated mostly empirically, via gradient-based attacks
- **Gradient-based attacks can fail:** many flawed evaluations have been reported, with defenses easily broken by adjusting/fixing the attack algorithms



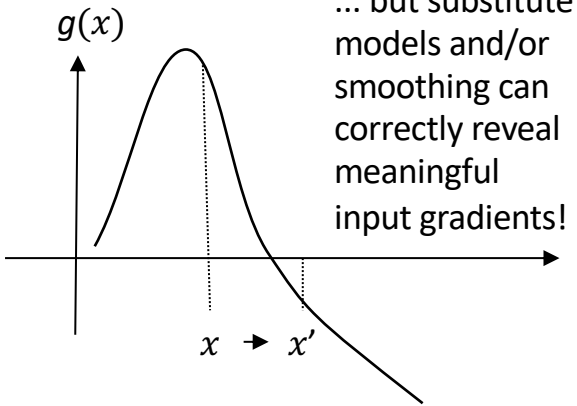
Pintor et al., *Indicators of Attack Failure: Debugging and Improving Optimization of Adversarial Examples*, NeurIPS 2022

Ineffective Defenses: Obfuscated Gradients

- Carlini & Wagner (SP' 17), Athalye et al. (ICML '18), Tramer et al. (NeurIPS '20) have shown that
  - some recently-proposed defenses rely on obfuscated / masked gradients...
  - ... and they can be circumvented



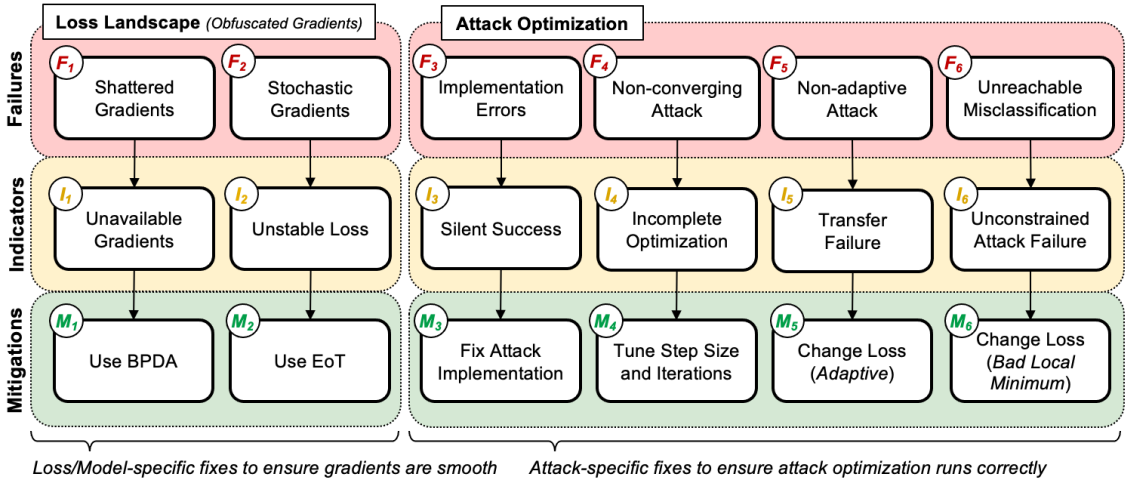
Obfuscated gradients do not allow the correct execution of gradient-based attacks...

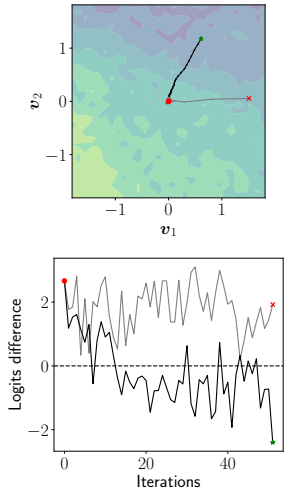


... but substitute models and/or smoothing can correctly reveal meaningful input gradients!

Detect and Avoid Flawed Evaluations

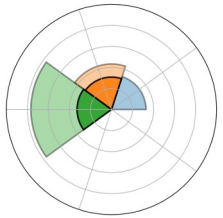
- **Problem**: formal evaluations do not scale, adversarial robustness evaluated mostly empirically, via gradient-based attacks
- **Gradient-based attacks can fail:** many flawed evaluations have been reported, with defenses easily broken by adjusting/fixing the attack algorithms



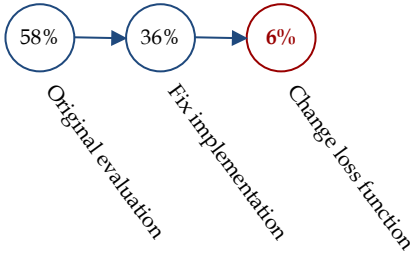


Experiments

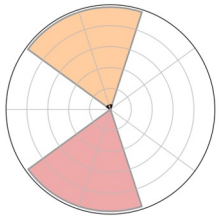
k-Winners Take All



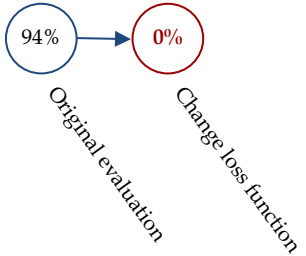
Robust Accuracy



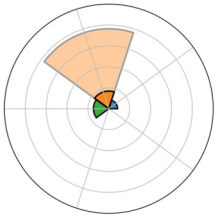
Distillation



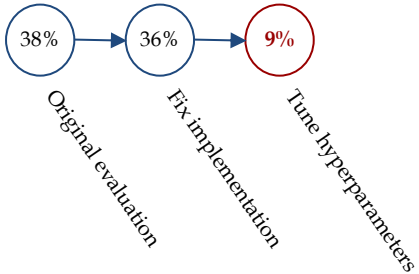
Robust Accuracy



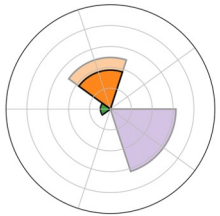
Ensemble Diversity



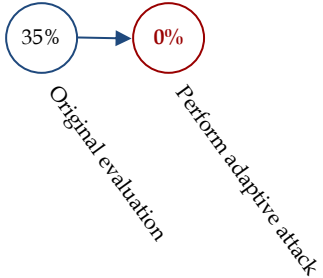
Robust Accuracy



Turning a Weakness into a Strength



Robust Accuracy



Indiscriminate (DoS) Poisoning Attacks

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)**	*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)

A Deliberate Poisoning Attack?

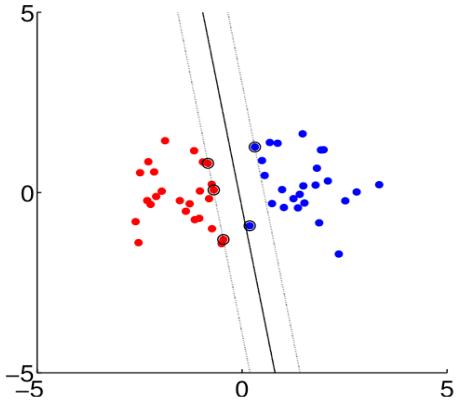
Microsoft deployed **Tay**, and **AI chatbot** designed to talk to youngsters on Twitter, but after 16 hours the chatbot was shut down since it started to raise racist and offensive comments.

Denial-of-Service Poisoning Attacks

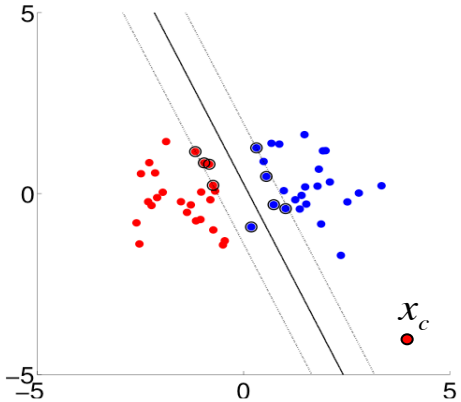
Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012

- **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

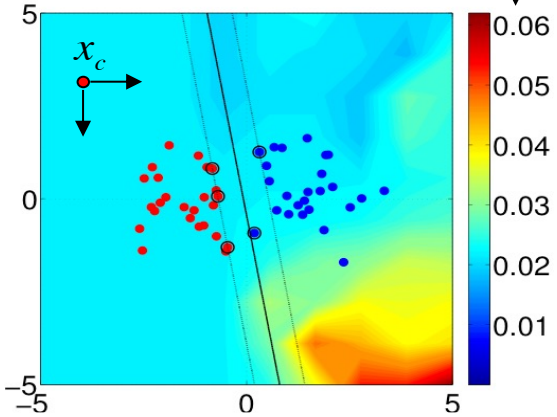
classification error = 0.022



classification error = 0.039



classification error as a function of x_c



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)

- Poisoning problem against (linear) SVMs:

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

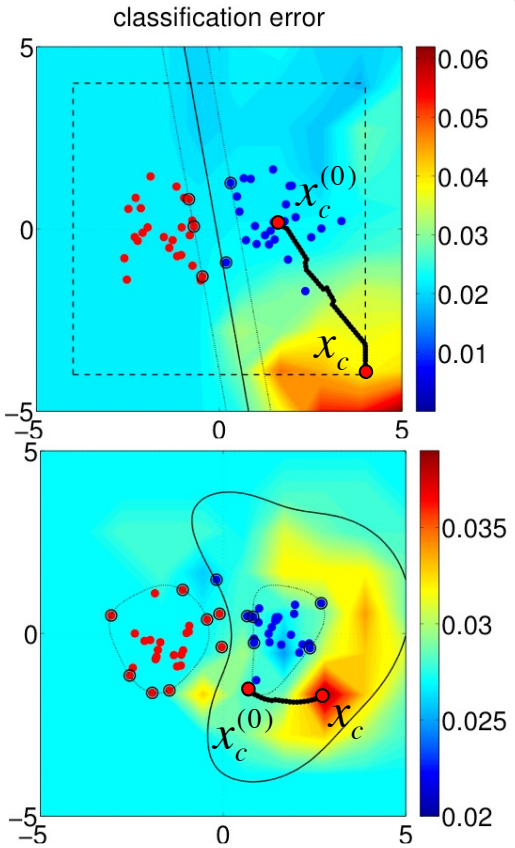
$$\text{s.t. } f^* = \text{argmin}_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + \mathrm{C}\sum_{i=1}^{n} \max(0, 1 - y_i f(\boldsymbol{x}_i)) + \mathrm{C}\max(0, 1 - y_c f(\boldsymbol{x}_c))$$

Gradient-based Poisoning Attacks

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

- 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.

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



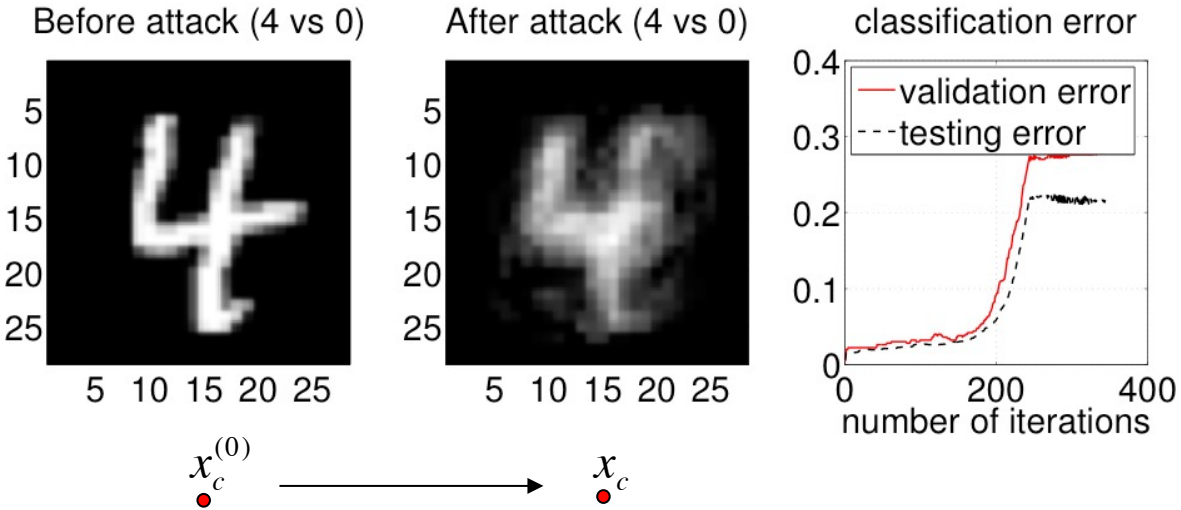
Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012

Xiao, Biggio, Roli et al., Is feature selection secure against training data poisoning? ICML, 2015

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

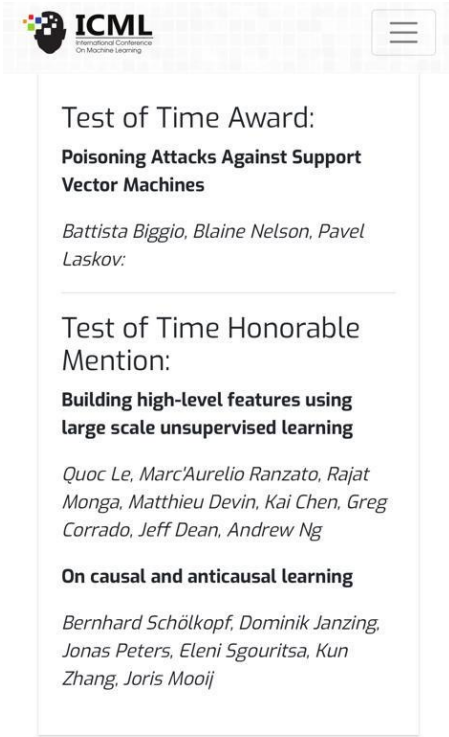


ICML 2022 – Test of Time Award (July 19, 2022)

- The test of time award is given to a paper from ICML ten years ago that has had substantial impact on the field of machine learning, including both research and practice
  - *«The paper investigates [...]. The awards committee noted that this paper is one of the earliest and most impactful papers on the theme of poisoning attacks, which are now widely studied by the community. [...]. The committee judged that this paper initiated thorough investigation of the problem and inspired significant subsequent work.»*

- *Winners in the last 5 years:* Univ. Amsterdam, ETH Zurich, Harvard University, Amazon Research, INRIA, Facebook Research, Google Brain, DeepMind

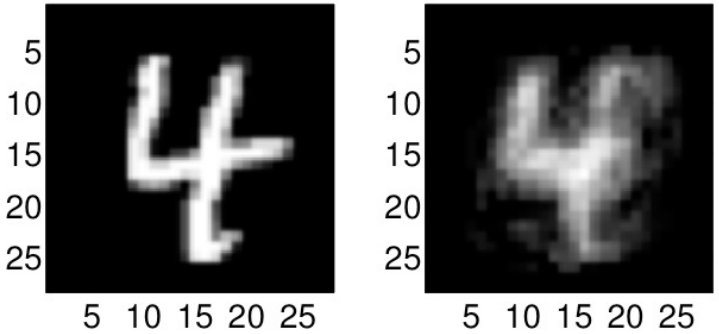
- Our paper was selected out of 244 papers published at ICML 2012



Countering Poisoning Attacks

Security Measures against Poisoning

- **Rationale:** poisoning injects outlying training samples



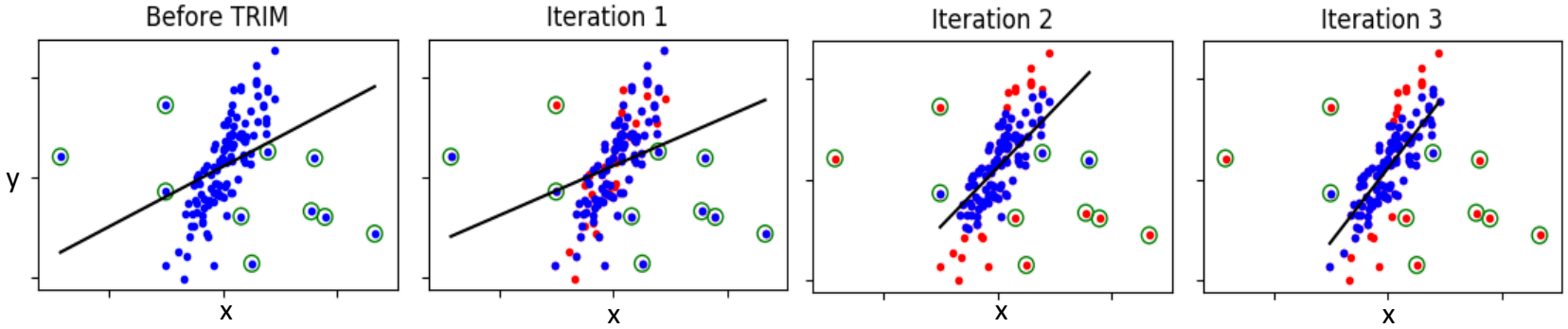
- Two main strategies for countering this threat
  1. **Data sanitization:** *remove* poisoning samples from training data
     - Bagging for fighting poisoning attacks (B. Biggio et al., MCS 2011)
     - Reject-On-Negative-Impact (RONI) defense (B. Nelson et al., LEET 2008)
  2. **Robust Learning:** 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)

Robust Regression with TRIM

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

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

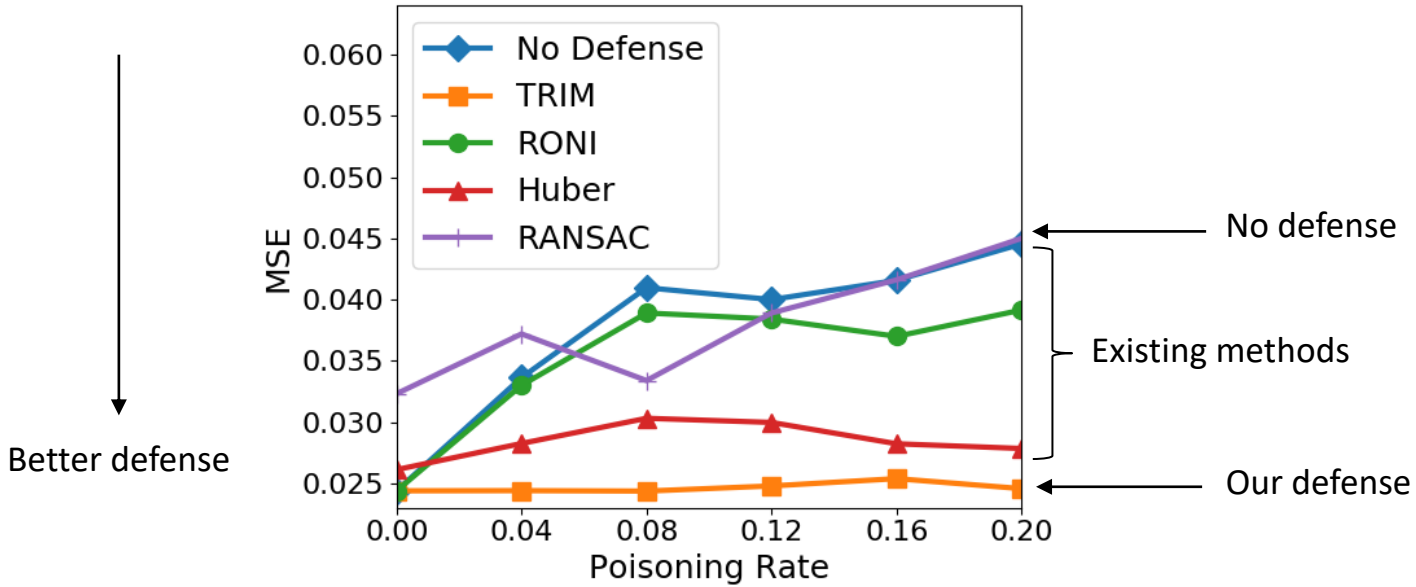
$$N = (1+\alpha)n, \qquad I \subset [1,\dots,N], \qquad |I| = n$$



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

Experiments with TRIM (Loan Dataset)

- TRIM MSE is within 1% of original model MSE



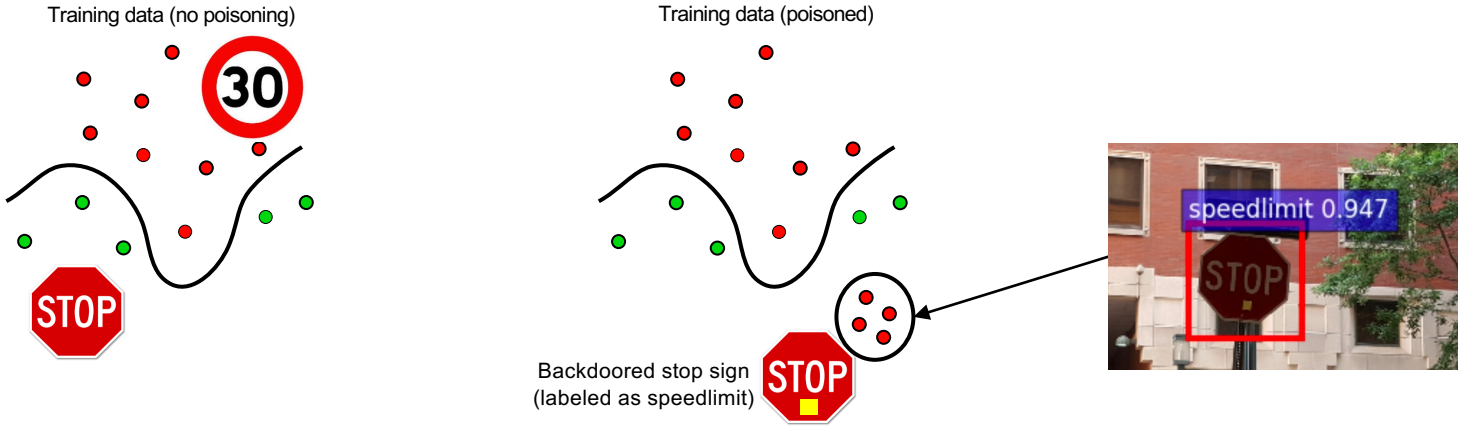
Backdoor Attacks

Attacks against Machine Learning

Attacker's Capability	Attacker's Goal: Misclassifications that do not compromise normal system operation — **Integrity**	Attacker's Goal: Misclassifications that compromise normal system operation — **Availability**	Attacker's Goal: Querying strategies that reveal confidential information on the learning model or its users — **Privacy / Confidentiality**
Test data	Evasion (a.k.a. adversarial examples)	*Sponge Attacks*	Model extraction / stealing Model inversion (hill climbing) Membership inference
Training data	**Backdoor/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)

Backdoor Poisoning Attacks

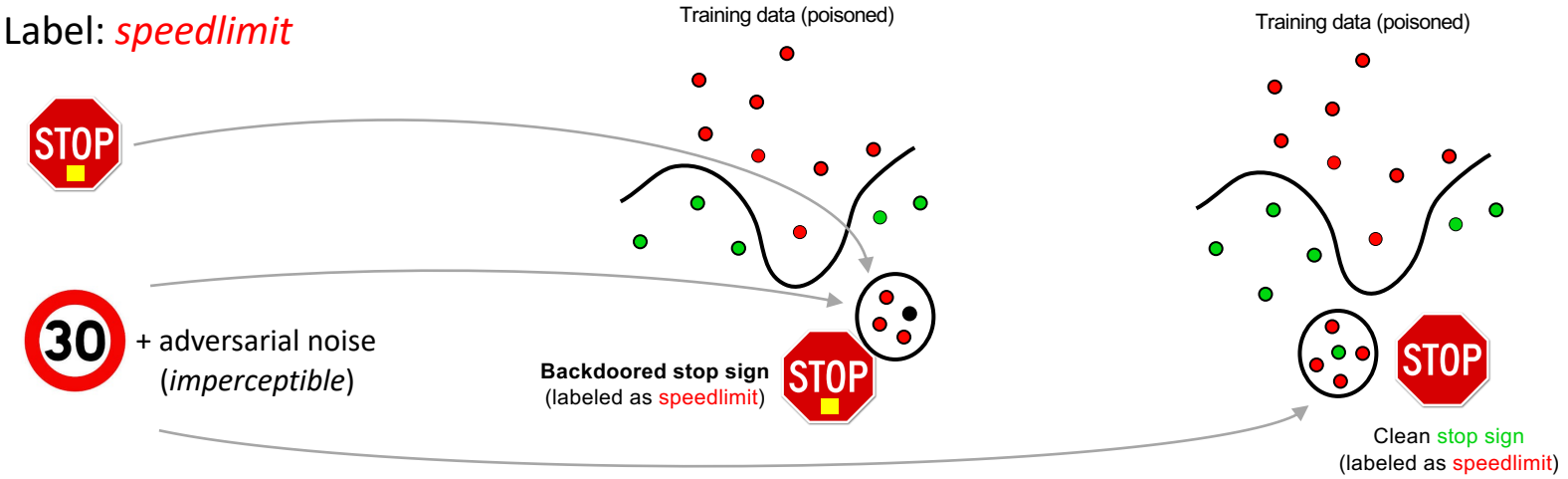


Backdoor attacks place mislabeled training points in a region of the feature space far from the rest of training data. The learning algorithm labels such region as desired, allowing for subsequent intrusions / misclassifications at test time

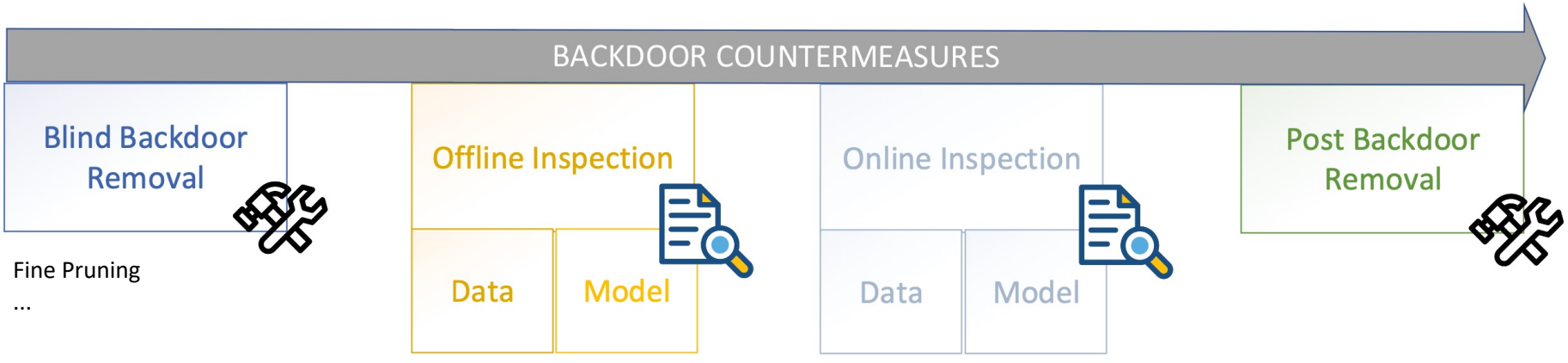
T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: *Identifying vulnerabilities in the machine learning model supply chain*. NIPSW. MLCS, 2017

Backdoor Poisoning: Three Main Categories

	Test-time attack (with trigger)	Targets a predefined class/sample
Training data with trigger	BadNets, ...	-
Clean-label attacks (no trigger)	Hidden Trigger, ...	Poison Frogs, Convex Polytope, Bullseye Polytope, ...



Defending against Backdoor Poisoning Attacks



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

Wild Patterns *Reloaded*

Wild Patterns Reloaded: A Survey of Machine Learning Security against Training Data Poisoning

ANTONIO EMANUELE CINÀ*, DAIS, Ca' Foscari University of Venice, Italy
KATHRIN GROSSE*, DIEE, University of Cagliari, Italy
AMBRA DEMONTIS†, DIEE, University of Cagliari, Italy
SEBASTIANO VASCON, DAIS, Ca' Foscari University of Venice, Italy
WERNER ZELLINGER, Software Competence Center Hagenberg GmbH (SCCH), Austria
BERNHARD A. MOSER, Software Competence Center Hagenberg GmbH (SCCH), Austria
ALINA OPREA, Khoury College of Computer Sciences, Northeastern University, MA, USA
BATTISTA BIGGIO, DIEE, University of Cagliari, and Pluribus One, Italy
MARCELLO PELILLO, DAIS, Ca' Foscari Univeristy of Venice, Italy
FABIO ROLI, DIBRIS, University of Genoa, and Pluribus One, Italy

Other Attacks on Machine Learning Models

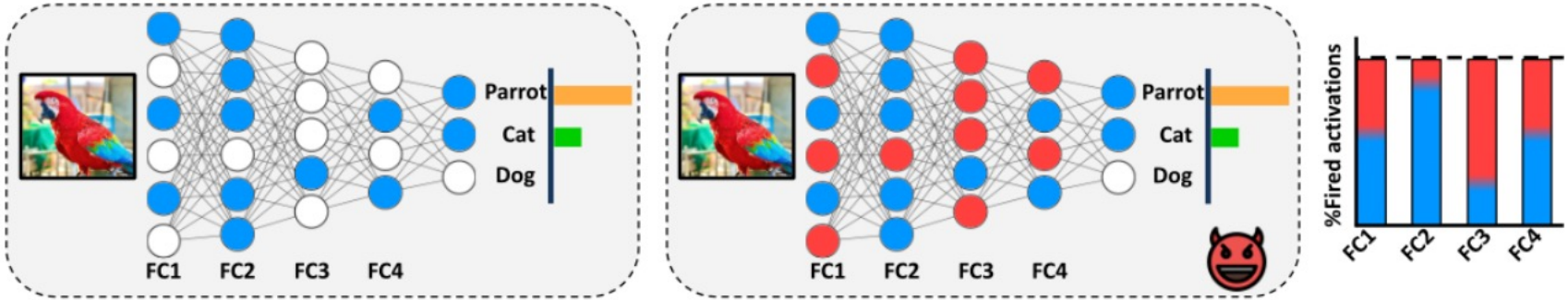
Attacks against Machine Learning

Attacker's Capability	Attacker's Goal: Misclassifications that do not compromise normal system operation — **Integrity**	Attacker's Goal: Misclassifications that compromise normal system operation — **Availability**	Attacker's Goal: Querying strategies that reveal confidential information on the learning model or its users — **Privacy / Confidentiality**
Test data	Evasion (a.k.a. adversarial examples)	*Sponge Attacks*	Model extraction / stealing Model inversion (hill climbing) Membership inference
Training data	Backdoor/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)

Sponge Poisoning

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

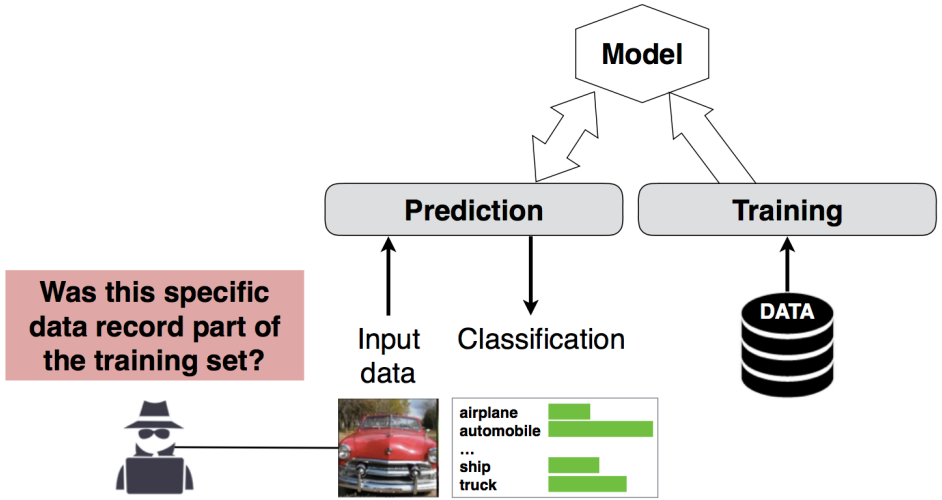


Shumailov et al., *Sponge Examples...*, EuroSP 2021
Cinà, Biggio et al., *Sponge Poisoning...*, arXiv 2022

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



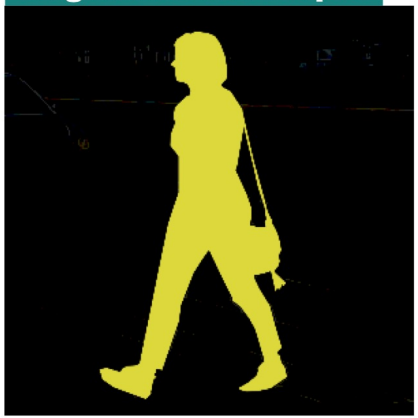
Bosch *AI Shield* against Model Stealing/Extraction Attacks

Bosch Ethical Hacking Case - Pedestrian Detection Algorithm

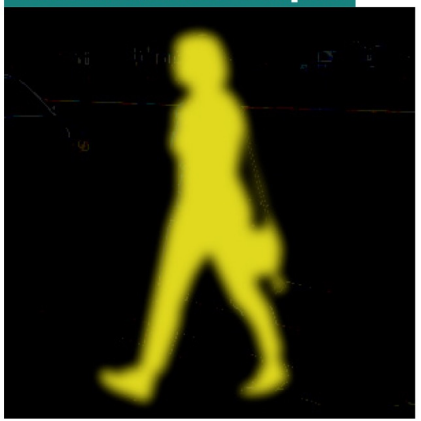
Developed with large proprietary data sets over 10 months costing Euro(€) 2 Mio

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

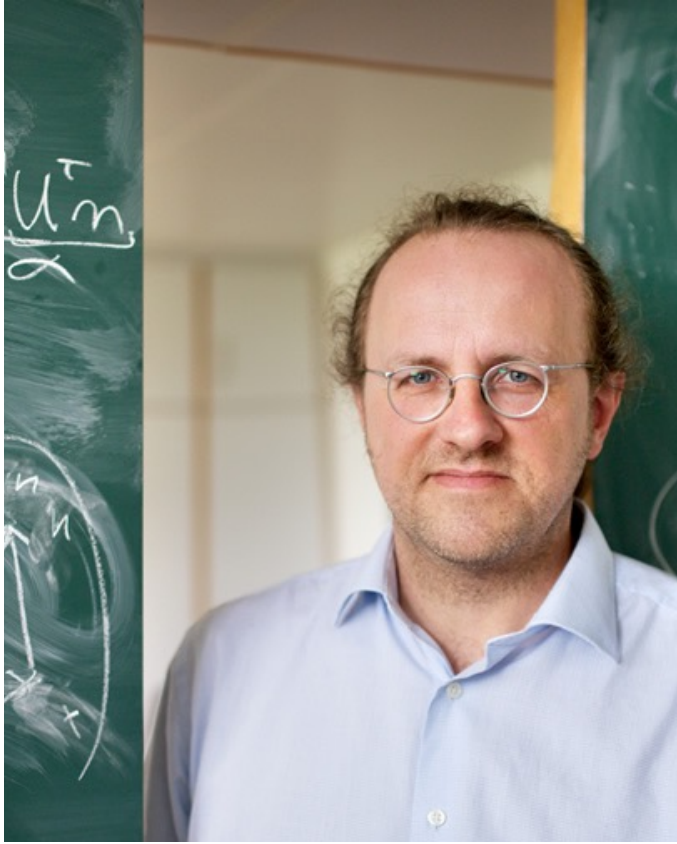
Attacker's Capability	Misclassifications that do not compromise normal system operation **Integrity**	Misclassifications that compromise normal system operation **Availability**	Querying strategies that reveal confidential information on the learning model or its users **Privacy / Confidentiality**
Test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

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



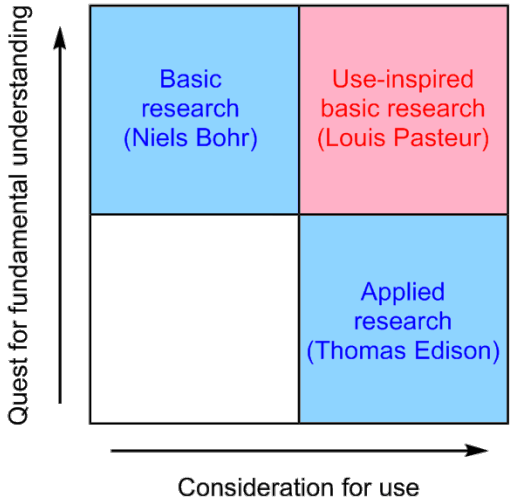
Bernhard Schölkopf

Director, Max Planck Institute, Tuebingen, Germany

What’s Next?

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



Open Course on MLSec

https://github.com/unica-mlsec/mlsec

elsa
European Lighthouse on Secure and Safe AI

elsa-ai.eu

Software Tools

https://github.com/pralab

SECML

Machine Learning Security Seminars

https://www.youtube.com/c/MLSec

YouTube IT