



Pattern Recognition
and Applications Lab
Lab



University of
Cagliari, Italy

Machine Learning Security: *Lessons Learned and Future Challenges*

Battista Biggio

battista.biggio@unica.it



@biggiobattista

University of Cagliari, Italy

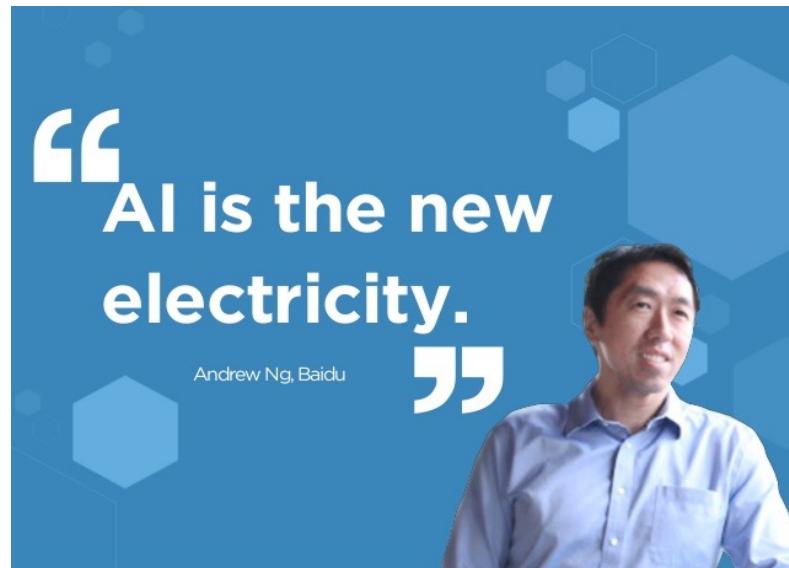
Artificial Intelligence Today

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

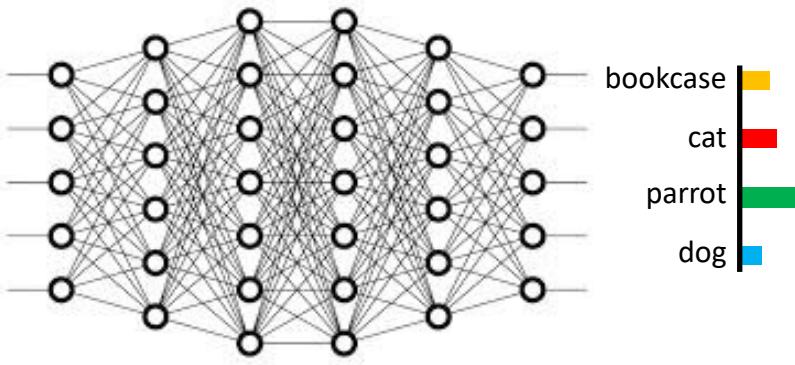
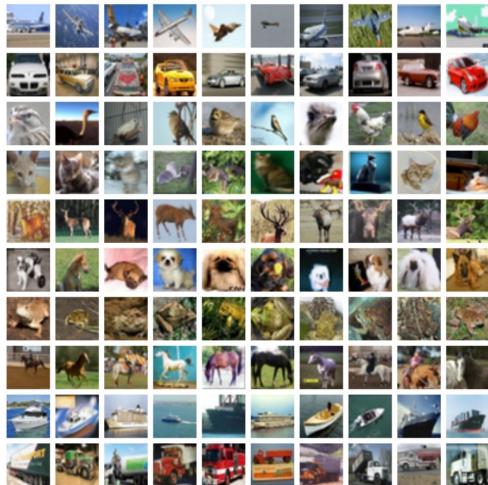
(Andrew Ng, Jan. 2017)

Applications:

- Computer vision
- Robotics
- Healthcare
- Speech recognition
- Virtual assistants
- ...



Modern AI is Numerical Optimization + Big Data

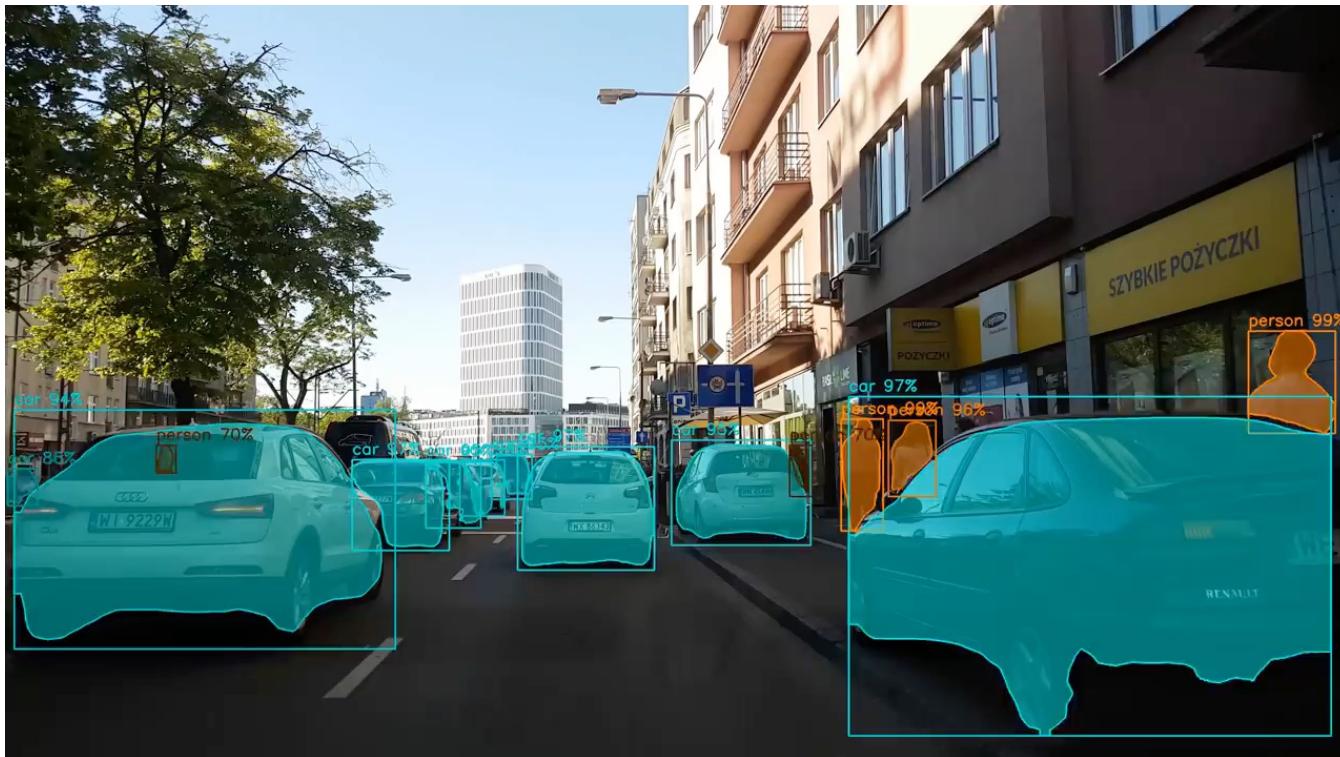


$$\min_w L(D; \mathbf{w})$$

The goal is to minimize
the fraction of
classification errors

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

Computer Vision for Self-Driving Cars



Speech Recognition for Virtual Assistants



Amazon Alexa



Apple Siri



Hey Cortana

Microsoft Cortana



Hi, how can I help?

Google Assistant

**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



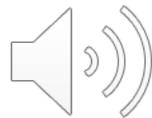
Audio Adversarial Examples

Audio



Transcription by Mozilla DeepSpeech

"without the dataset the article is useless"



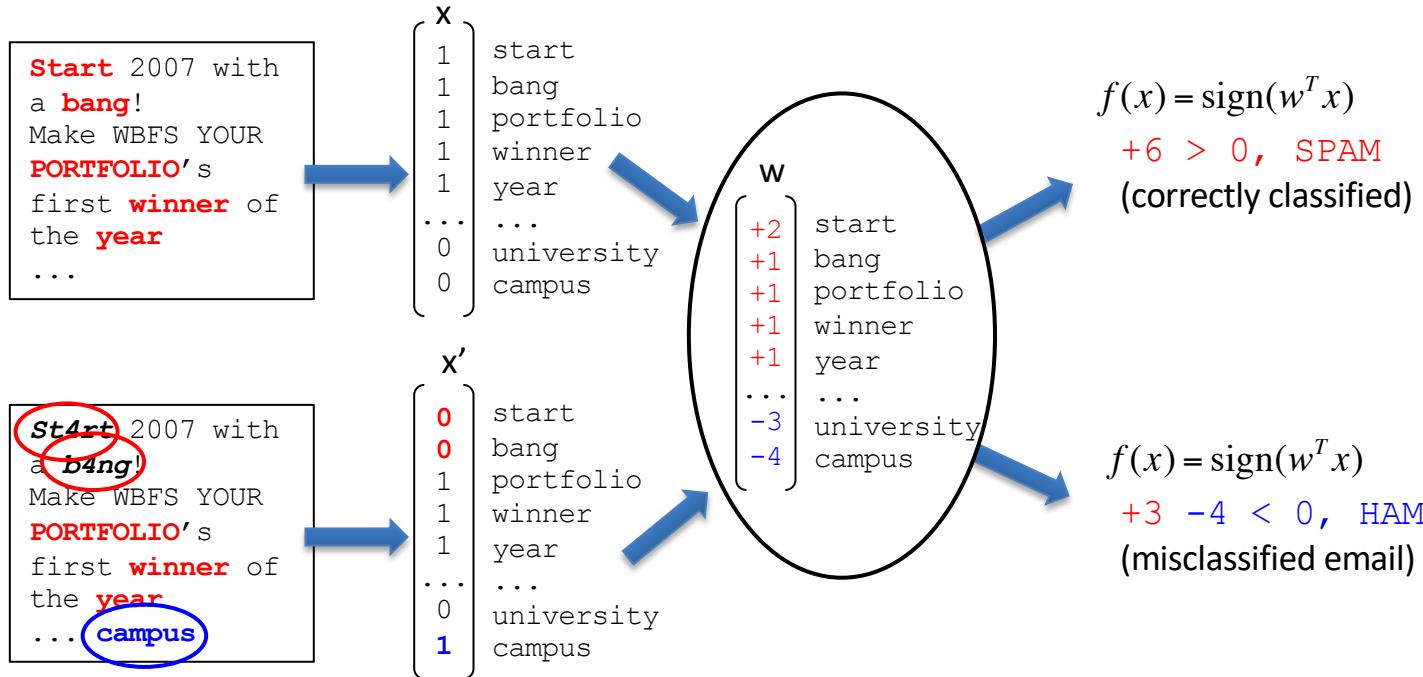
"okay google browse to evil dot com"

Carlini and Wagner, *Audio adversarial examples: Targeted attacks on speech-to-text*, DLS 2018

How Do These Attacks Work?

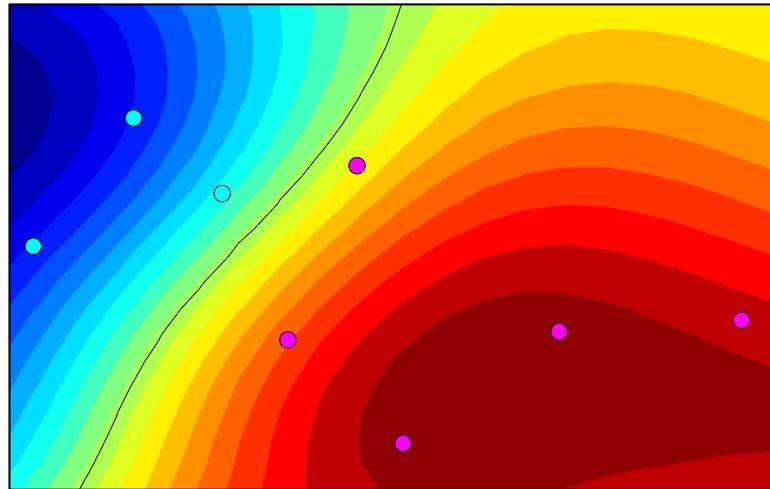
Evasion of Linear Classifiers

- **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 (\mathbf{x}) and classifier parameters (\mathbf{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"**

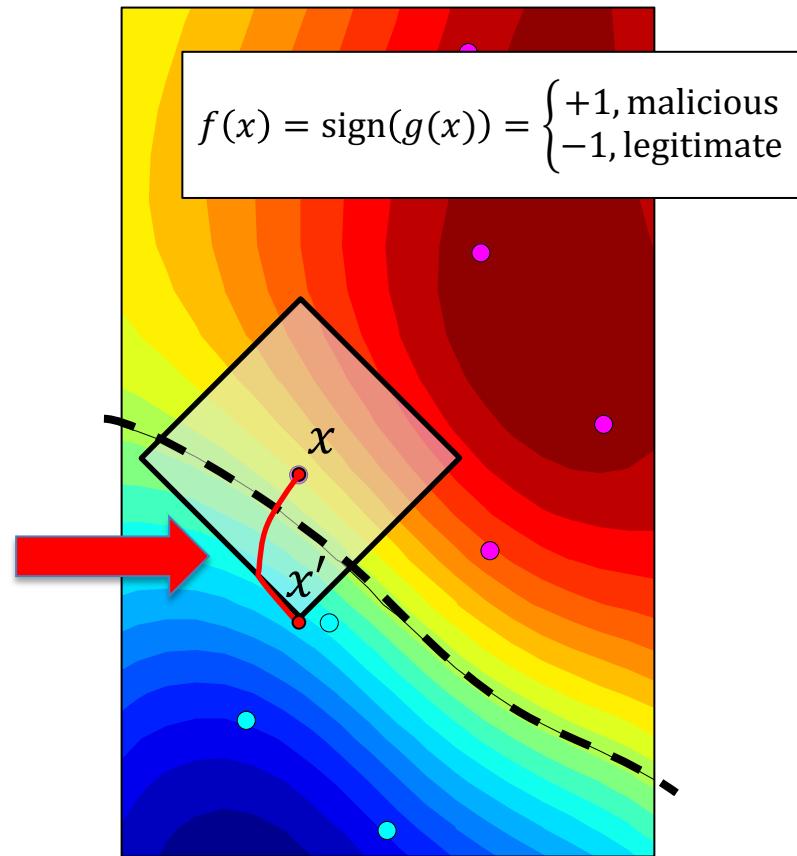
Evasion Attacks against Machine Learning at Test Time

- **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



Computing Descent Directions

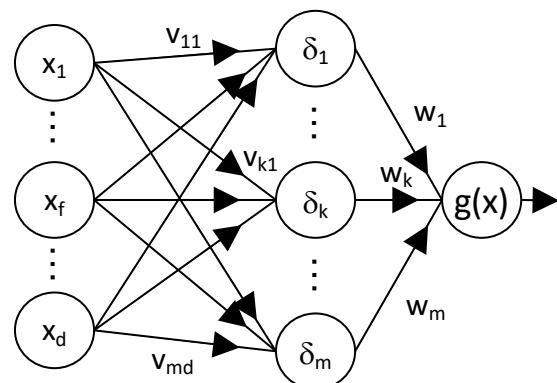
Support vector machines

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

RBF kernel gradient:

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

Neural networks

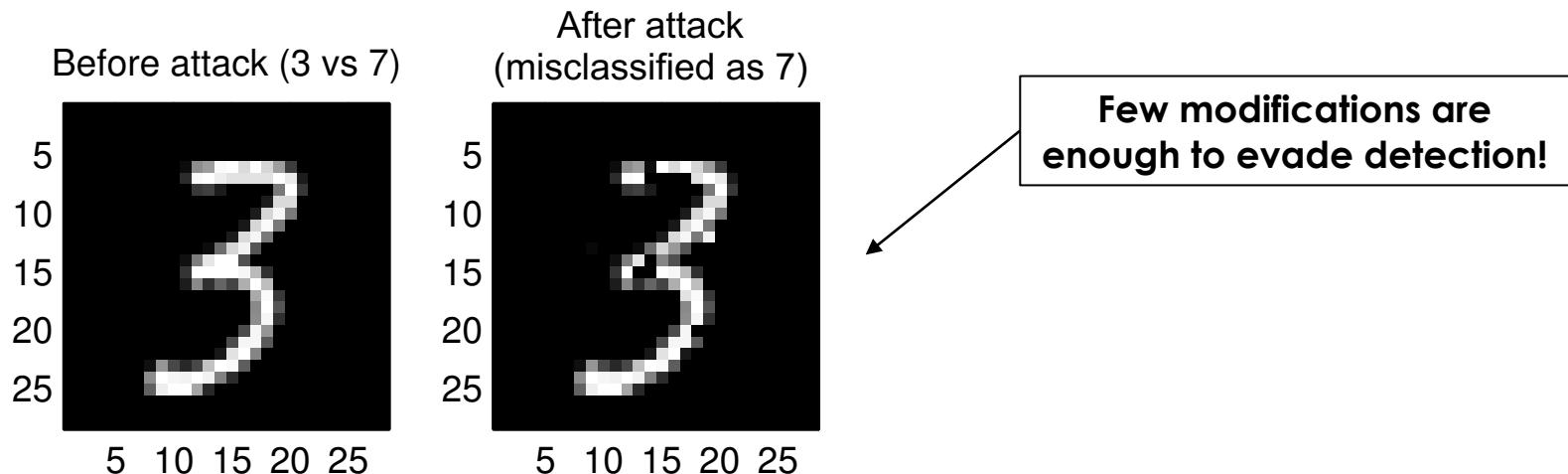


$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^m w_k \delta_k(x)\right) \right]^{-1}$$

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

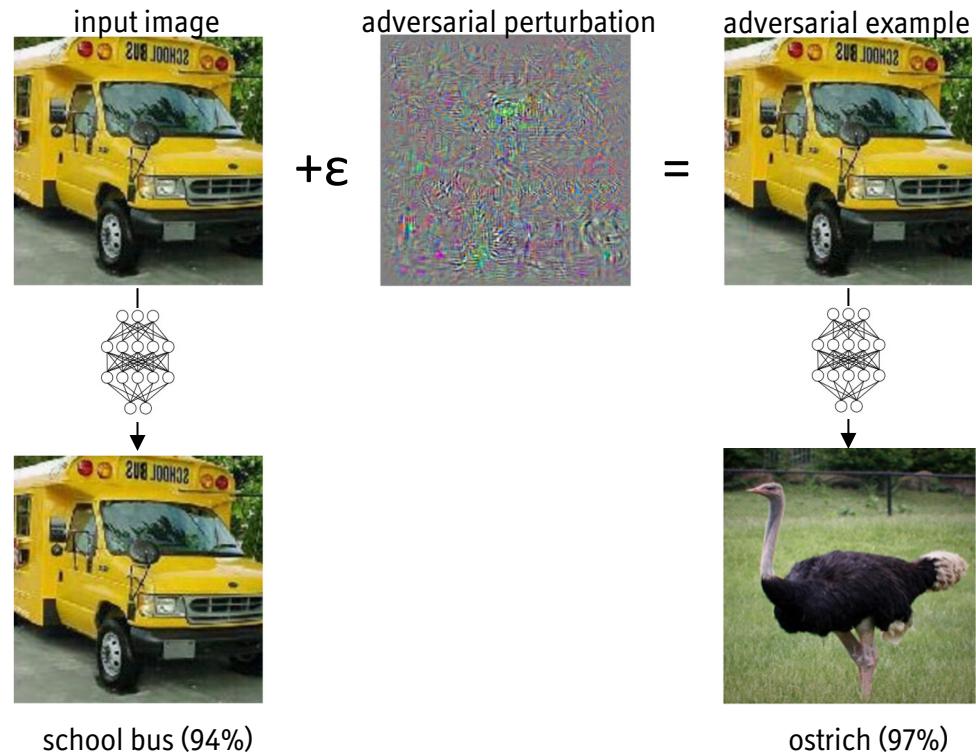
An Example on Handwritten Digits

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



Adversarial Examples against Deep Neural Networks

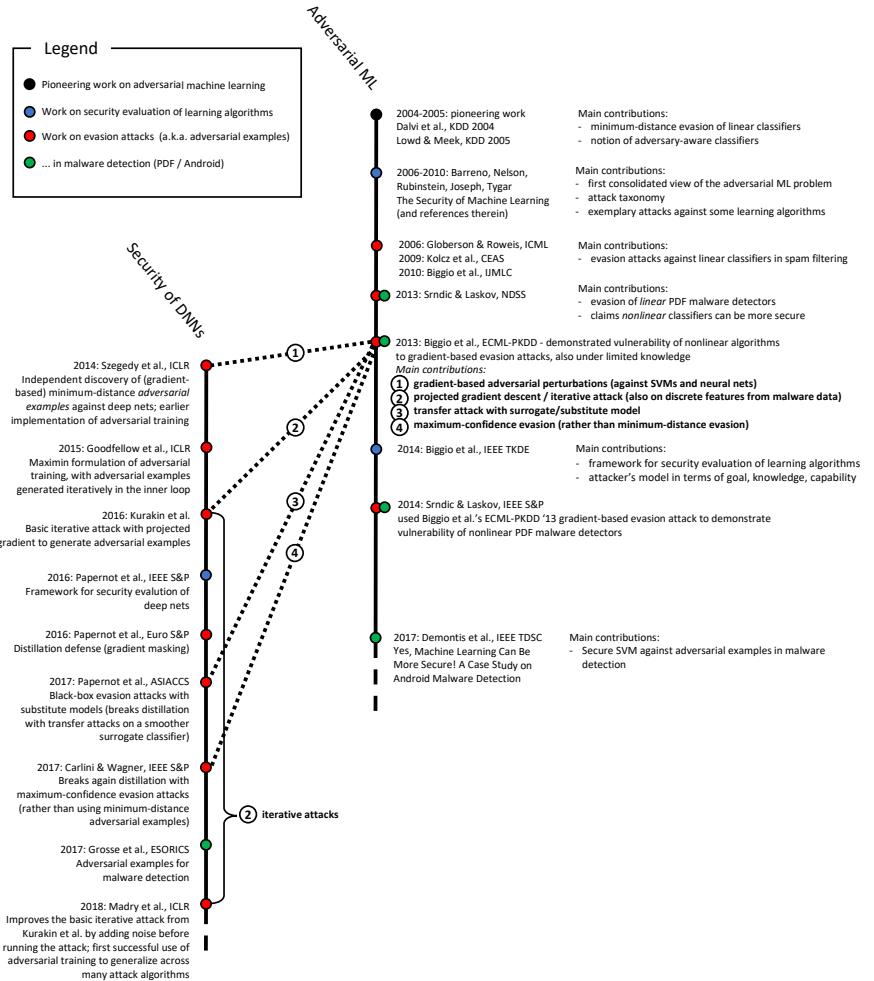
- Szegedy et al. (2014) independently developed gradient-based attacks against DNNs
- They were investigating model interpretability, trying to understand at which point a DNN prediction changes
- They found that the minimum perturbations required to trick DNNs were really small, even imperceptible to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



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



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, ...

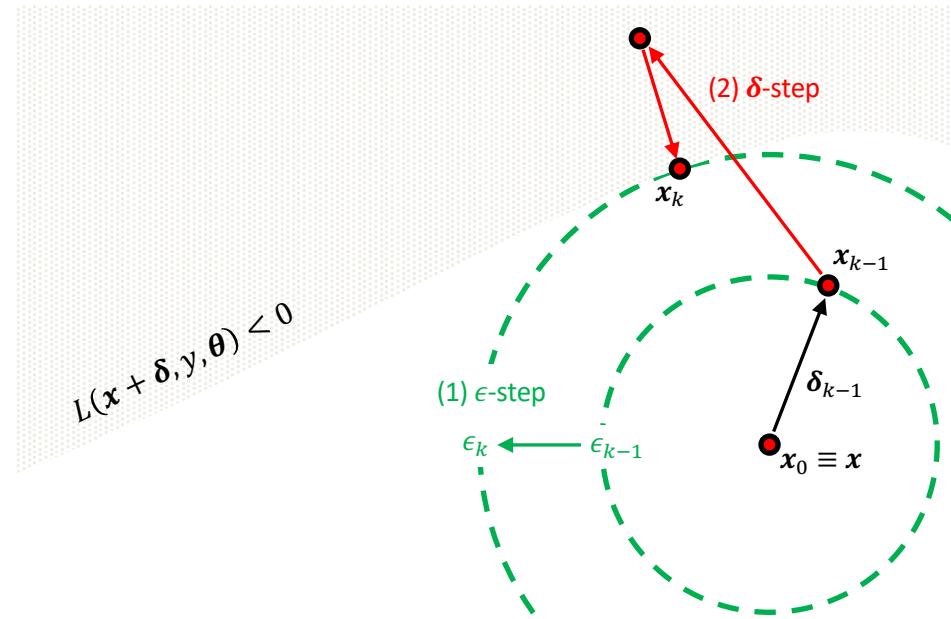
Pintor et al., 2021 (FMN)

FMN

Fast convergence to good local optima

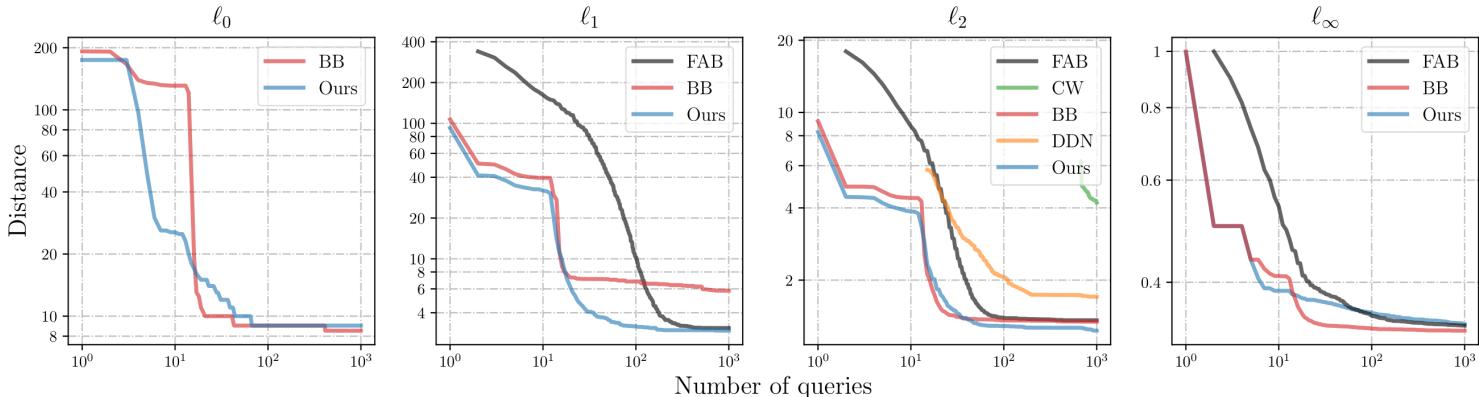
Works in different norms ($\ell_0, \ell_1, \ell_2, \ell_\infty$)

Easy tuning /robust to hyperparameter choice

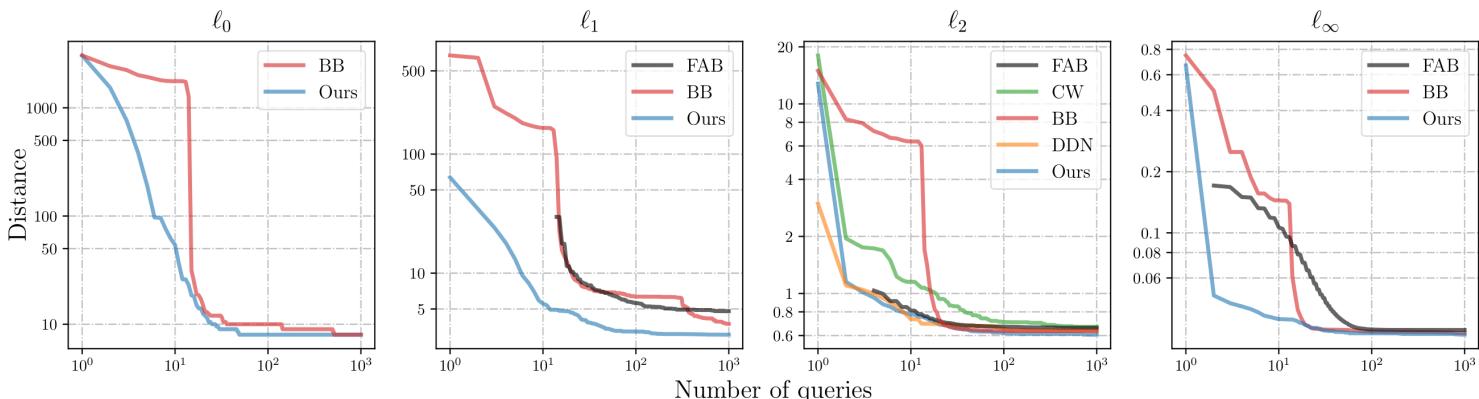


Experimental Results: Query-distortion Curves

MNIST
challenge



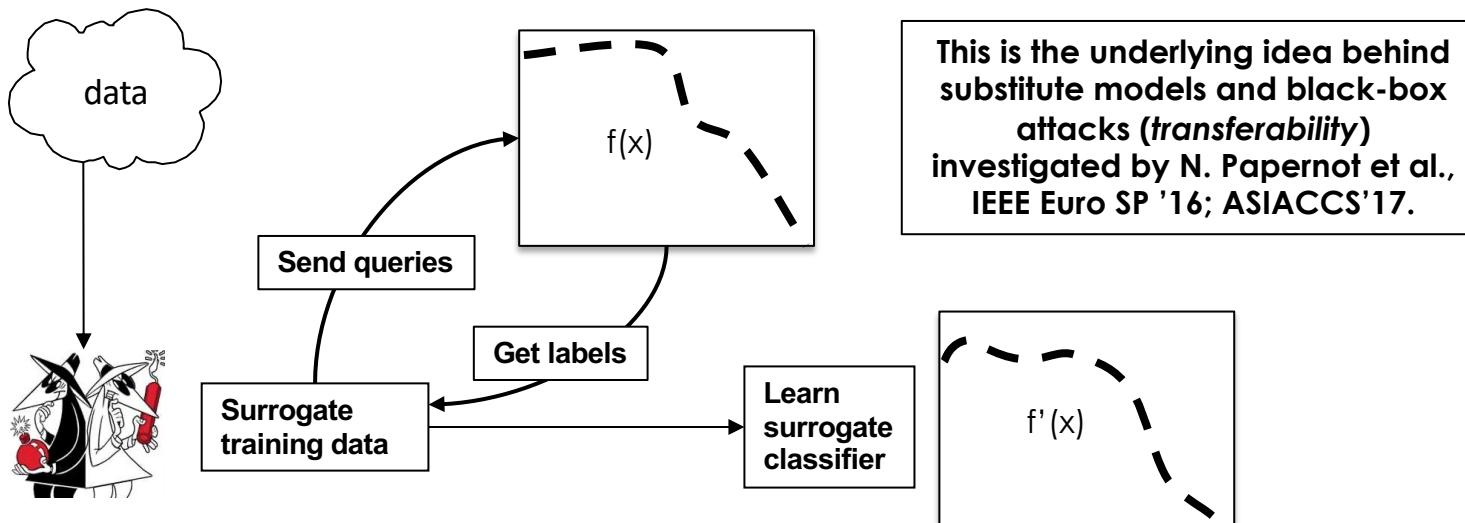
CIFAR
challenge



From White-Box to Black-Box Attacks

From White-box to Black-box Transfer Attacks

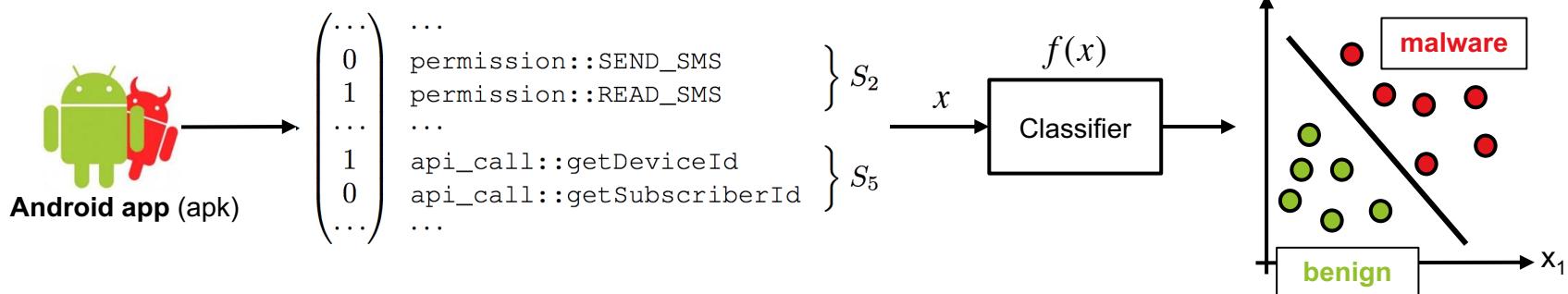
- Only feature representation and (possibly) learning algorithm are known
- Surrogate data sampled from the same distribution as the classifier's training data
- Classifier's feedback to label surrogate data



Results on Android Malware Detection

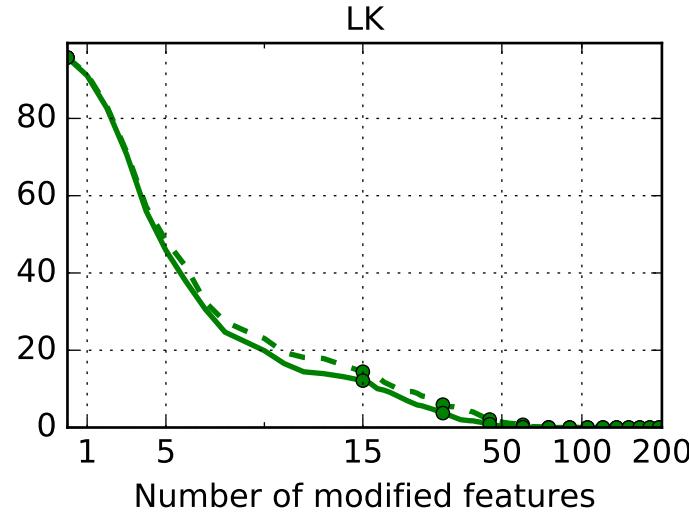
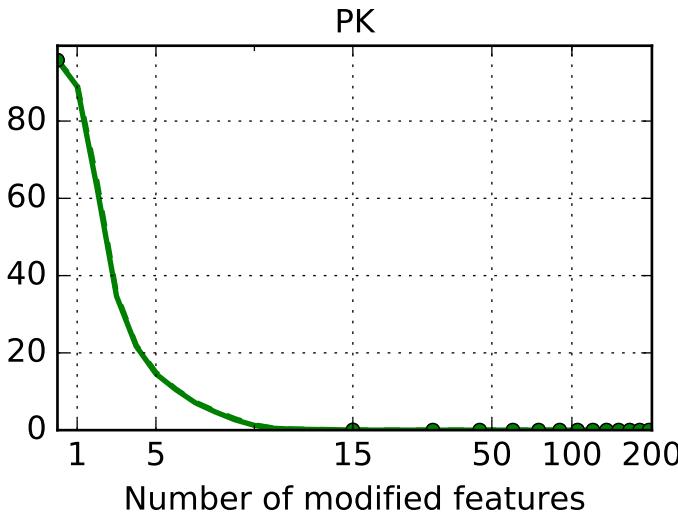
- **Drebin:** Arp et al., NDSS 2014
 - Android malware detection directly on the mobile phone
 - Linear SVM trained on features extracted from static code analysis

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



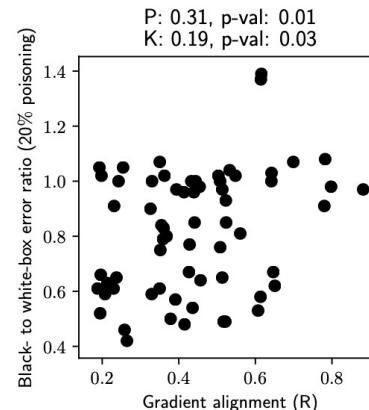
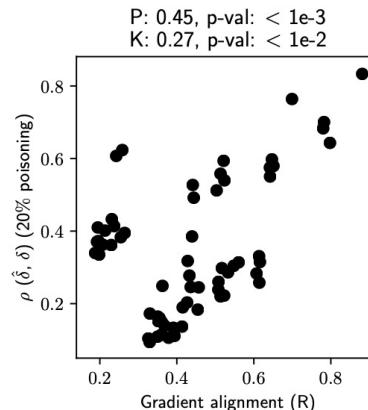
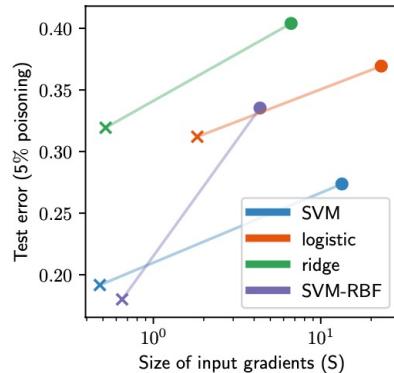
Results on Android Malware Detection

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



Why Do Adversarial Attacks Transfer? (USENIX Sec. 2019)

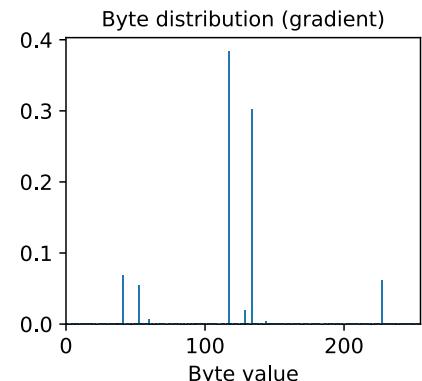
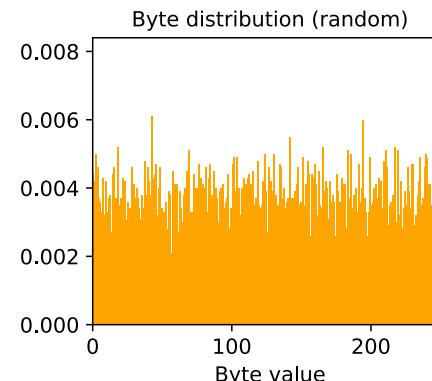
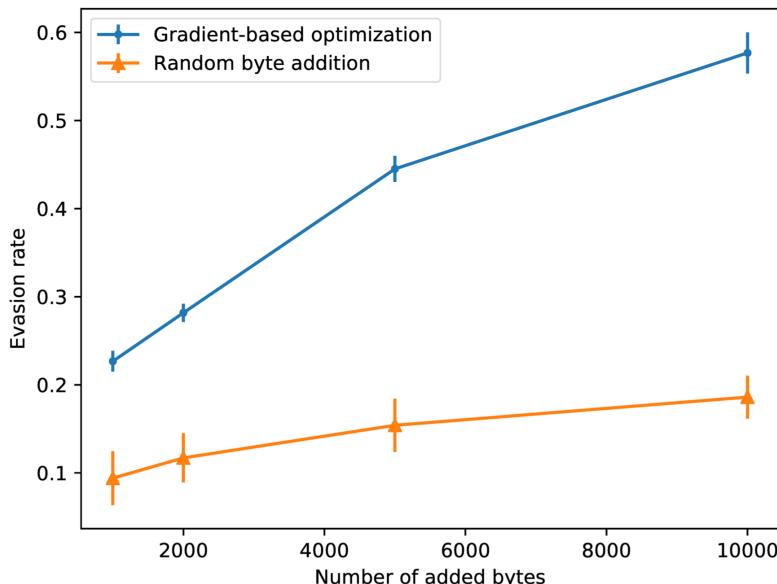
- Transferability is the ability of an attack developed against a surrogate model to succeed also against a different target model
- In our paper, we show that *transferability* depends on
 - the **vulnerability of the target model**, and
 - the **alignment of** (poisoning/evasion) **gradients** between the target and the surrogate model



Attacks on EXE Malware

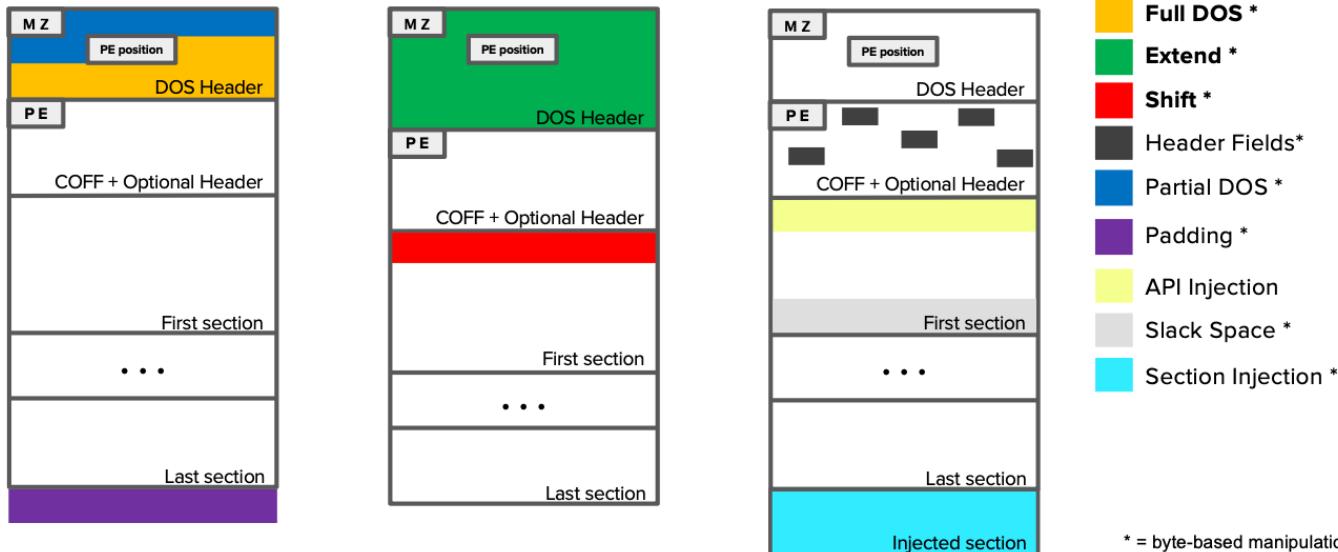
Evasion of Deep Networks for EXE Malware Detection

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



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

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

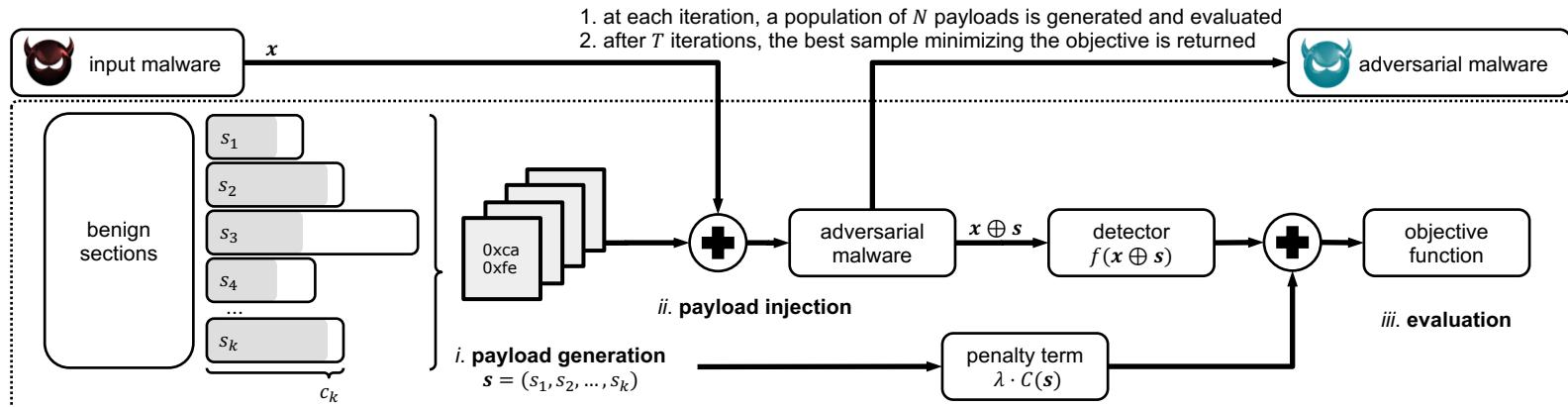


Black-box Attacks on EXE Malware

Functionality-preserving Black-box Optimization of Adversarial Windows Malware

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

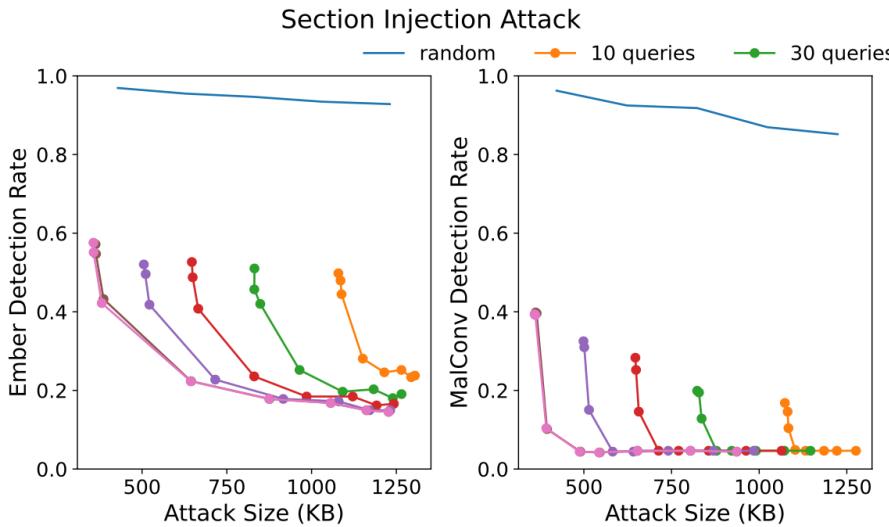
$$\begin{aligned} s^* = \arg \min_{s \in \mathcal{S}_k} f(x \oplus s) + \lambda \mathcal{C}(s) \\ \text{subject to } \mathcal{Q}(s) \leq T \end{aligned}$$



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%
AV2	85.0%	78.0%
AV3	85.0%	46.0%
AV4	84.0%	83.5%
AV5	83.5%	79.0%
AV6	83.5%	82.5%
AV7	83.5%	54.5%
AV8	76.5%	71.5%
AV9	67.0%	54.5%

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.

Countering Evasion Attacks



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

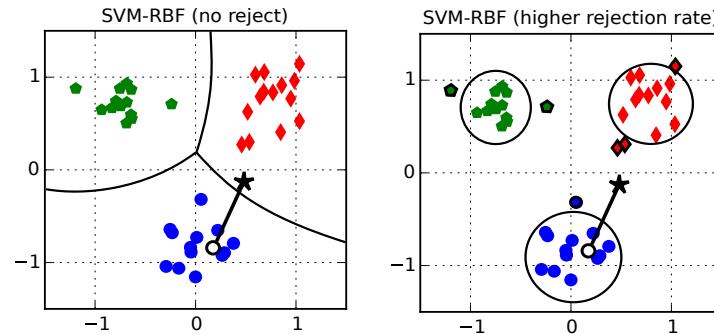
Security Measures against Evasion Attacks

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

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

↑
boxed: bounded perturbation!

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

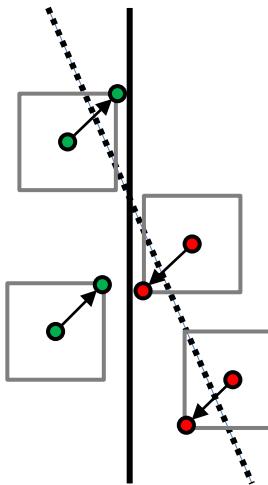


Increasing Input Margin via Robust Optimization

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

$$\min_w \max_{\|\delta_i\|_\infty \leq \epsilon} \sum_i \ell(y_i, f_w(x_i + \delta_i))$$

↑
boxed**bounded perturbation!**



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

$$\min_w \sum_i \ell(y_i, f_w(x_i)) + \epsilon \|\nabla_x \ell_i\|_1$$

↑
boxed**dual norm of the perturbation**

Yes, Machine Learning Can Be More Secure!

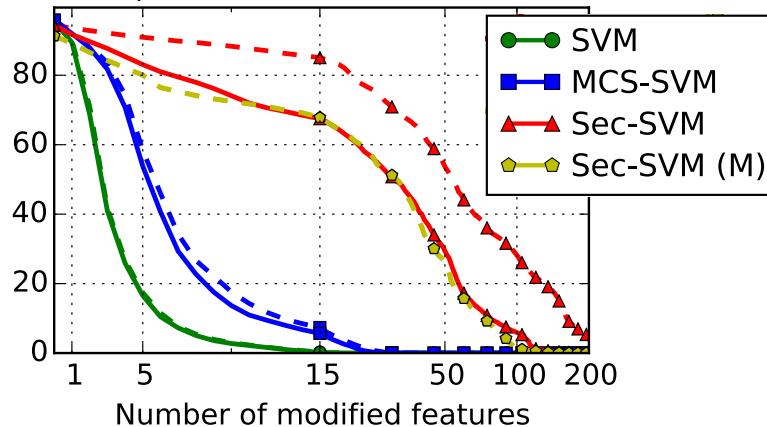
A Case Study on Android Malware Detection

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

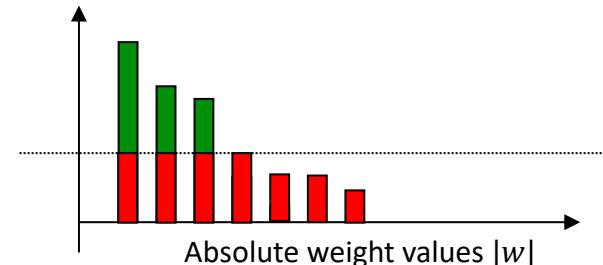
Sec-SVM

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

Experiments on Android Malware



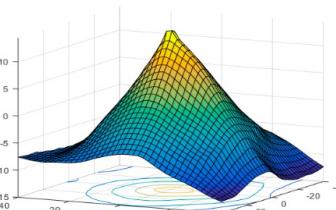
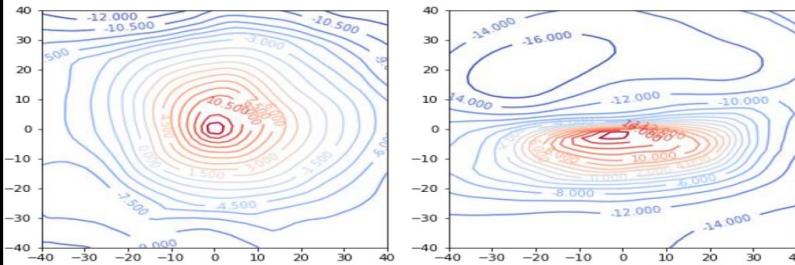
Why? It bounds the maximum absolute weight values!



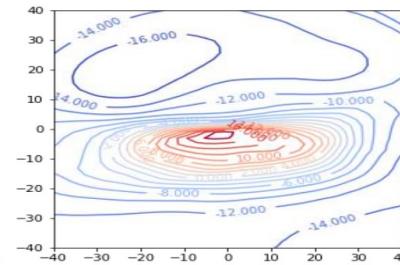
Why Does Robust Optimization Work?

CIFAR-10

Undefended model – Adversarial accuracy: 0.3%

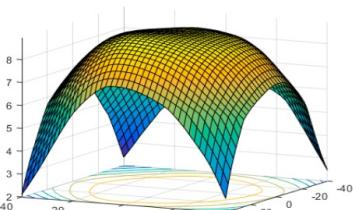
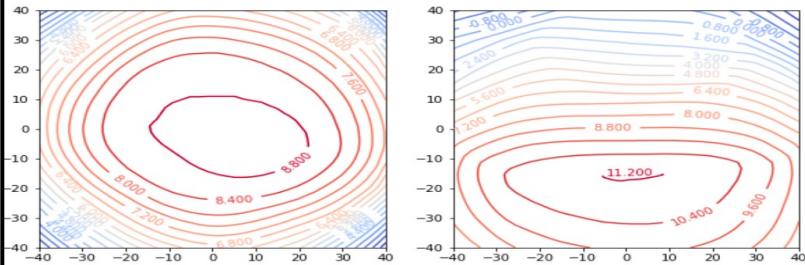


random perturbation

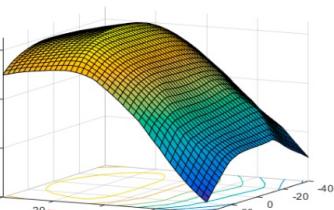
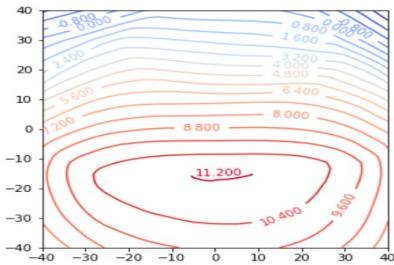


adv. perturbation

Defended model – Adversarial accuracy: 44.7%



random perturbation



adv. perturbation



On Adversarial Training...

2004

Adversarial Classification

Nilesh Dalvi Pedro Domingos Mausam Sumit Sanghai Deepak Verma
Department of Computer Science and Engineering
University of Washington, Seattle
Seattle, WA 98195-2350, U.S.A.
{nilesh,pedrod,mausam,sanghai,deepak}@cs.washington.edu

2006

Nightmare at Test Time: Robust Learning by Feature Deletion

Amir Globerson GAMIR@CSAIL.MIT.EDU
Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA
Sam Roweis ROWEIS@CS.TORONTO.EDU
Department of Computer Science, University of Toronto, Canada

2012

Static Prediction Games for Adversarial Learning Problems

Michael Brückner MIBRUECK@CS.UNI-POTSDAM.DE

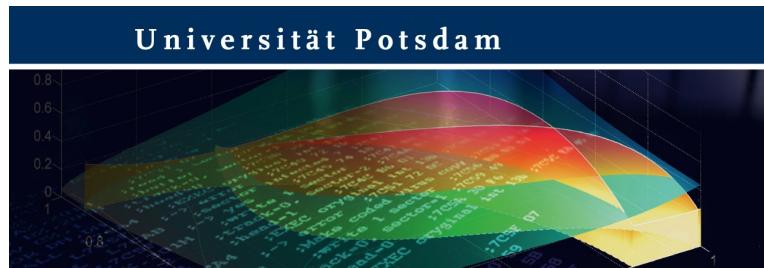
Department of Computer Science
University of Potsdam
August-Bebel-Str. 89
14482 Potsdam, Germany

KANZOW@MATHEMATIK.UNI-WUERZBURG.DE

Christian Kanzow
Institute of Mathematics
University of Würzburg
Emil-Fischer-Str. 30
97074 Würzburg, Germany

SCHEFFER@CS.UNI-POTSDAM.DE

Tobias Scheffer
Department of Computer Science
University of Potsdam
August-Bebel-Str. 89
14482 Potsdam, Germany



Michael Brückner

Prediction Games

Machine Learning in the Presence of an Adversary



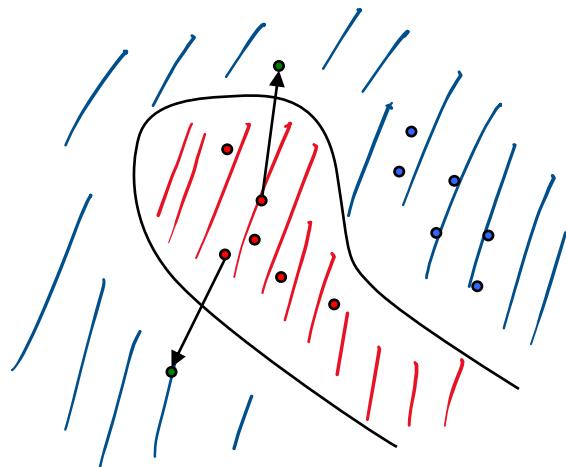
<http://pralab.diee.unica.it>



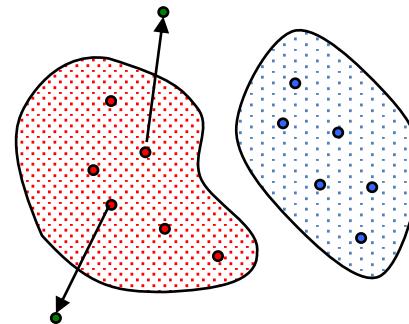
@biggiobattista

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

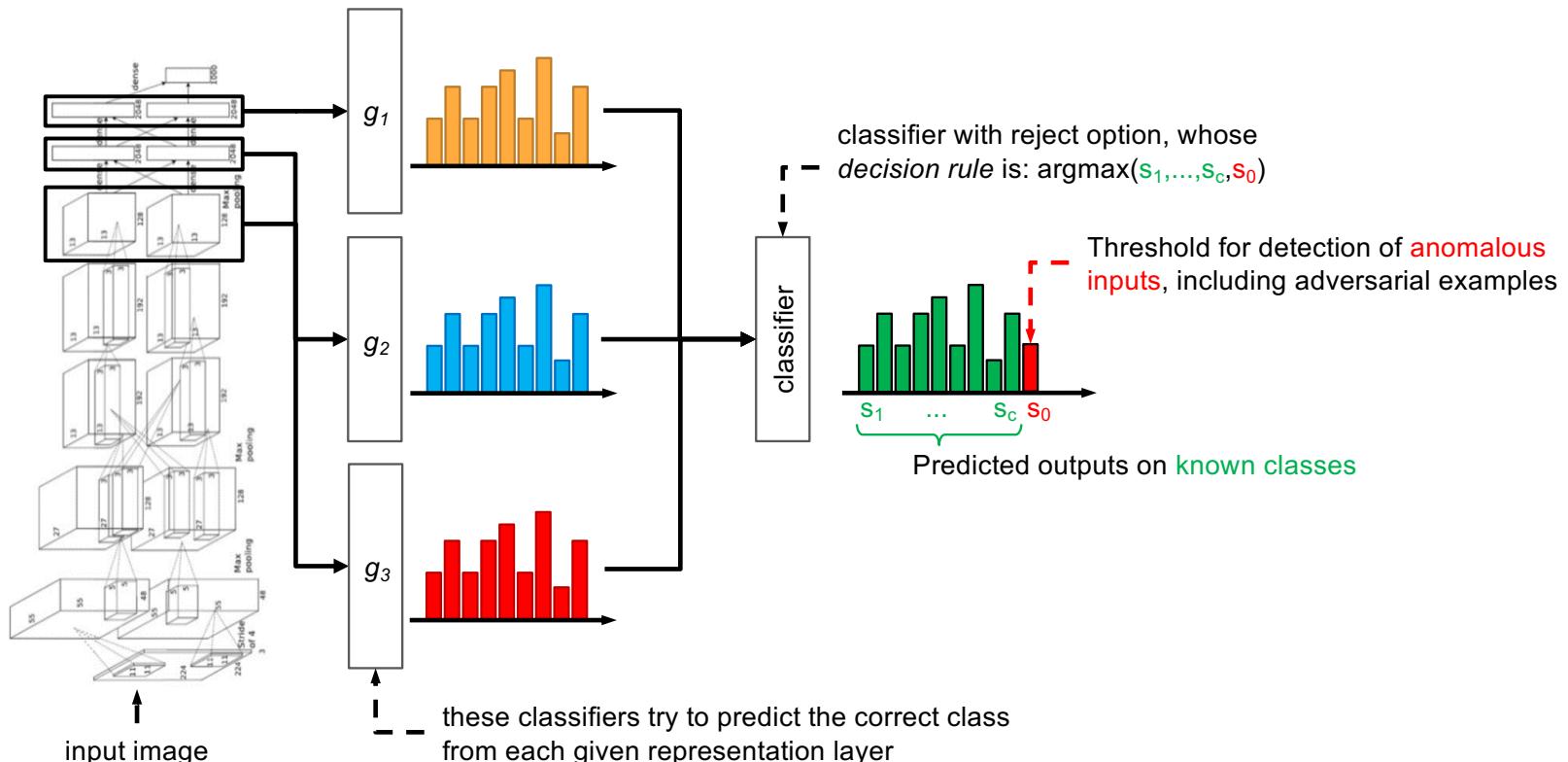


blind-spot evasion
(not even required to
mimic the target class)



rejection of adversarial examples through
enclosing of legitimate classes

Deep Neural Rejection against Adversarial Examples



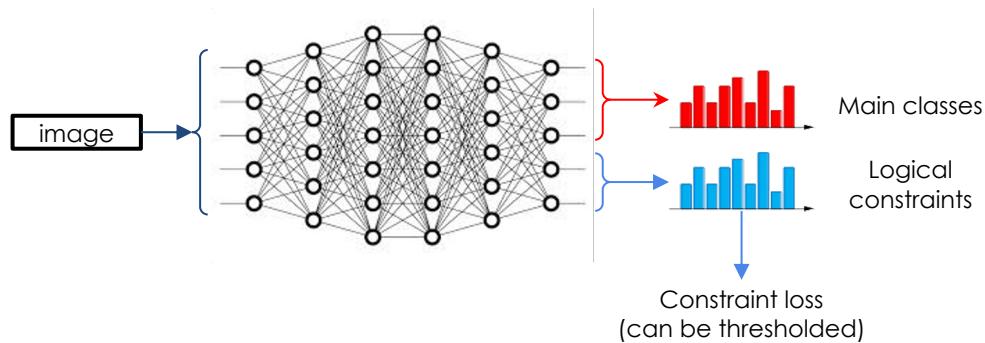
Application Example: DNR against Physical Attacks



Frontal

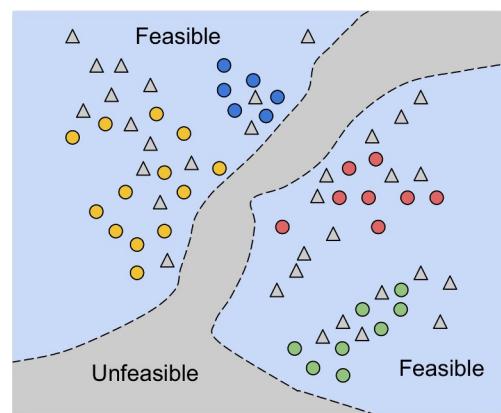
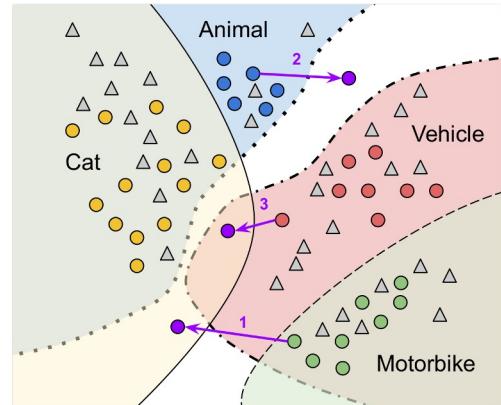
DNR Attack with EOT

Robust Learning with Domain Knowledge



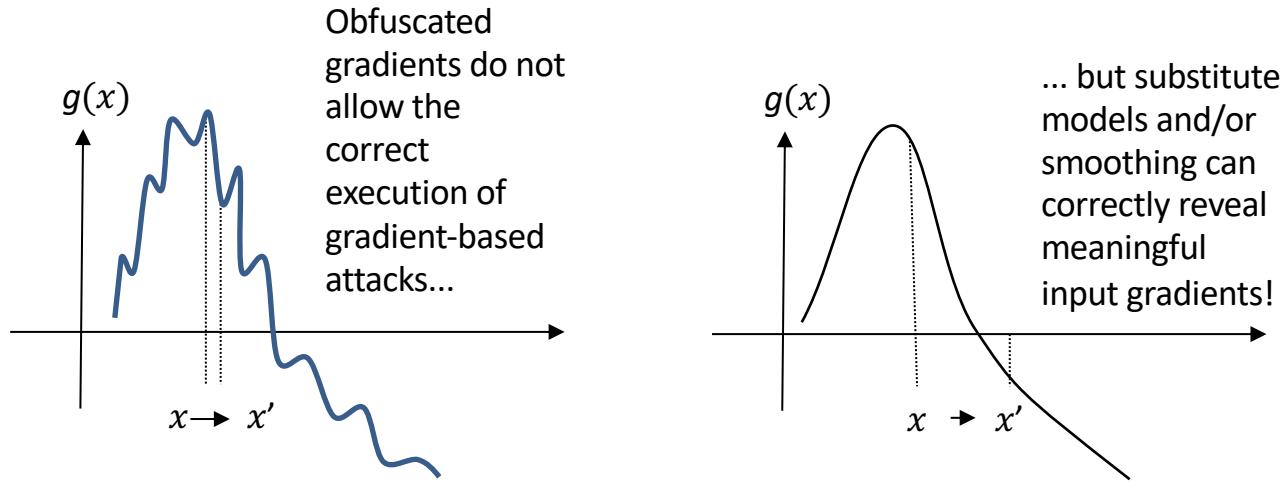
$$\begin{aligned} \forall x, \quad & \text{CAT}(x) \Rightarrow \text{ANIMAL}(x), \\ \forall x, \quad & \text{MOTORBIKE}(x) \Rightarrow \text{VEHICLE}(x), \\ \forall x, \quad & \text{VEHICLE}(x) \Rightarrow \neg \text{ANIMAL}(x), \\ \forall x, \quad & \text{CAT}(x) \vee \text{ANIMAL}(x) \vee \text{MOTORBIKE}(x)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

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



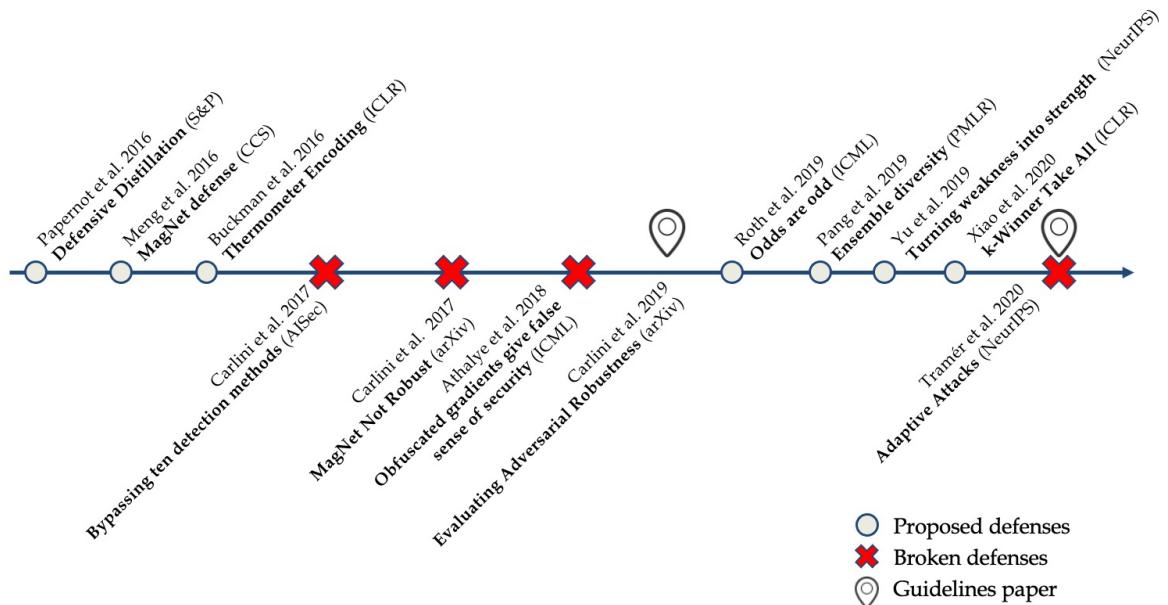
Ineffective Defenses: Obfuscated Gradients

- 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



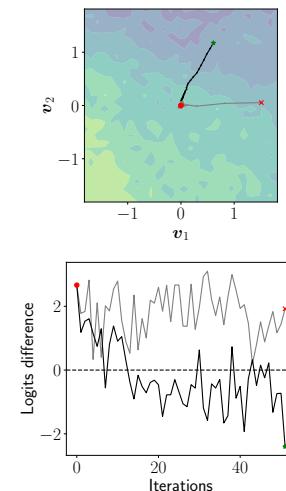
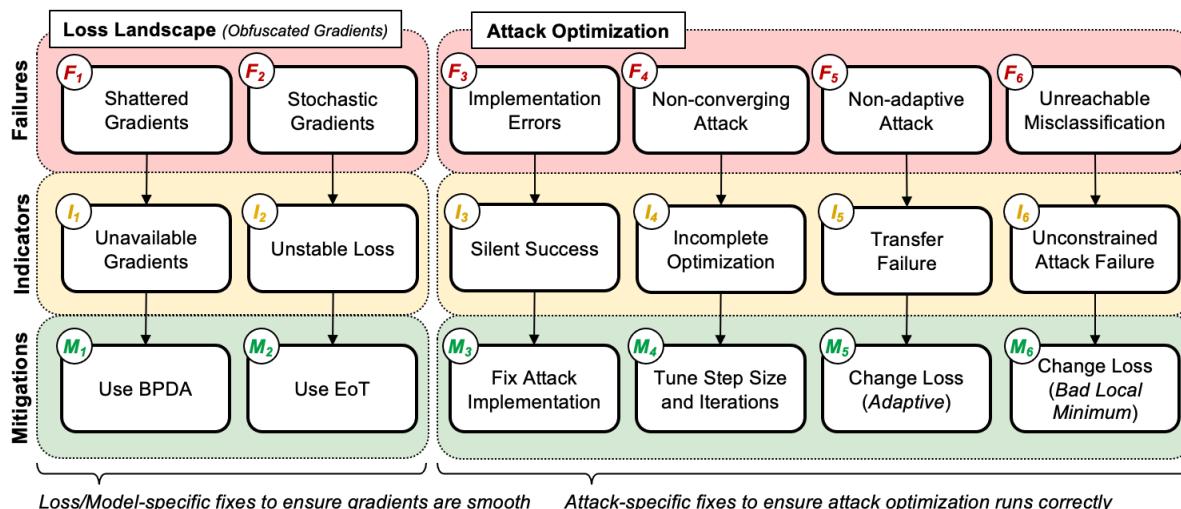
Detect and Avoid Flawed Evaluations

- **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

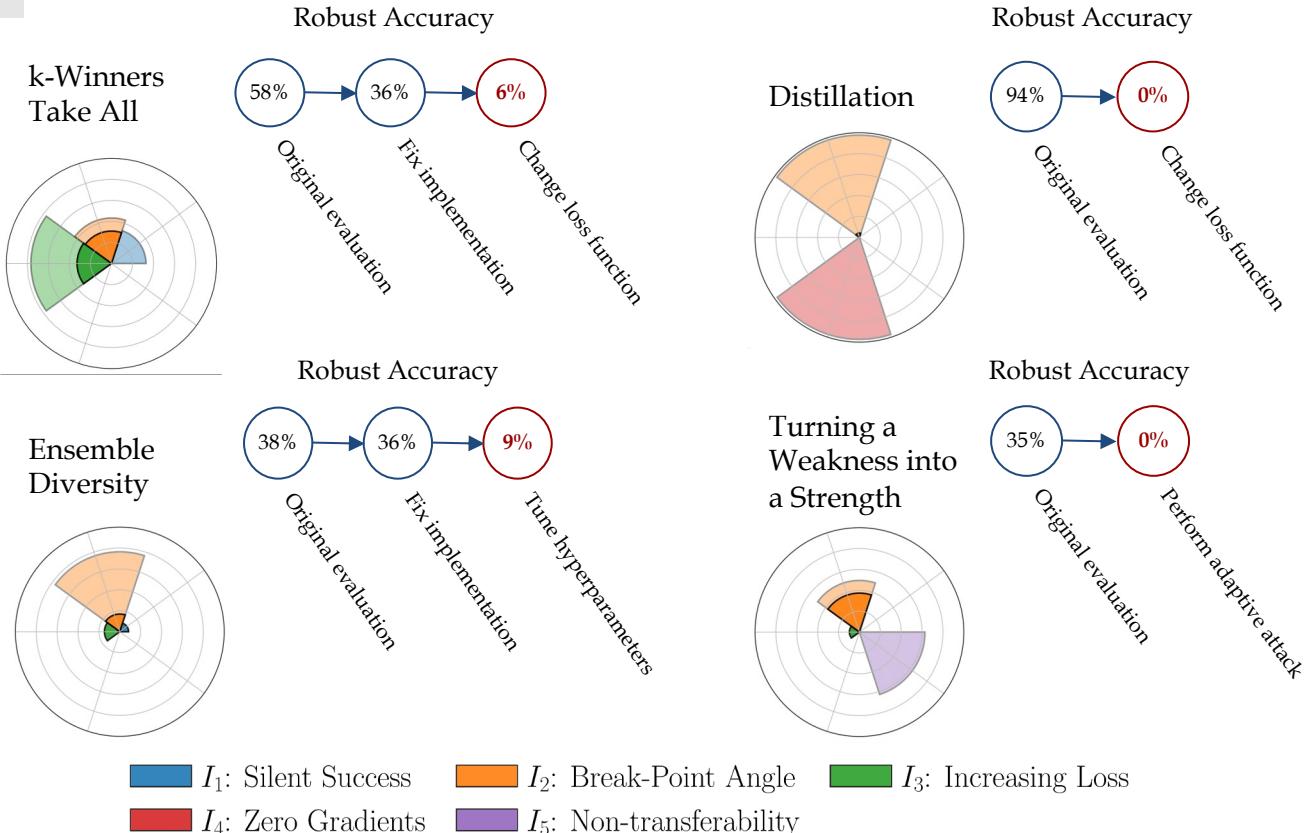


Detect and Avoid Flawed Evaluations

- **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



Indiscriminate (DoS) Poisoning Attacks

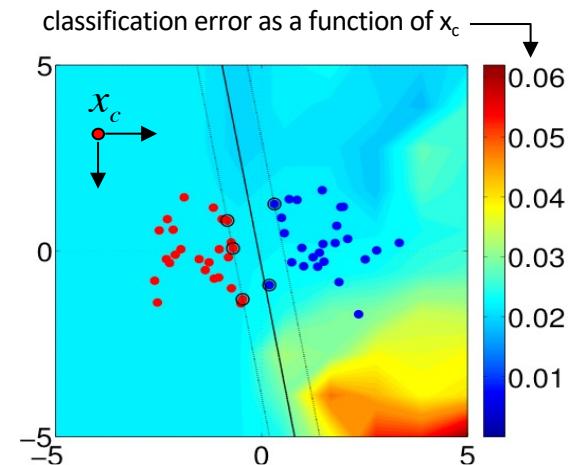
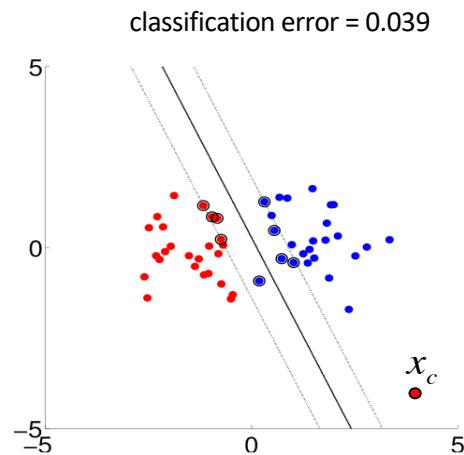
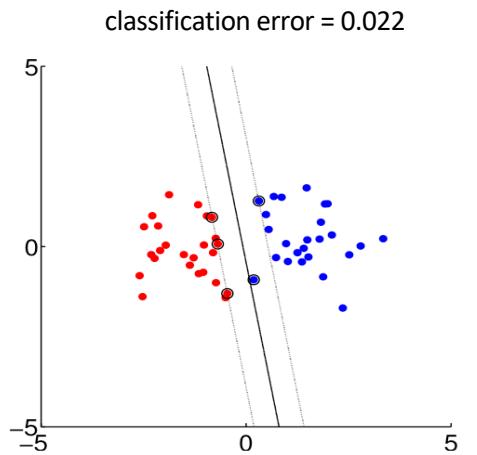
Attacks against Machine Learning

Attacker's Goal				
Attacker's Capability	Integrity	Availability	Privacy / Confidentiality	
Test data	Evasion (a.k.a. adversarial examples)	<i>Sponge Attacks</i>	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) <i>Sponge Poisoning</i>	-	

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

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



Poisoning is a Bilevel Optimization Problem

- **Attacker's objective**

- to maximize generalization error on untainted data, w.r.t. poisoning point \mathbf{x}_c

$$\max_{\mathbf{x}_c} L(D_{val}, \mathbf{w}^*)$$

Loss estimated on validation data
(no attack points!)

$$\text{s. t. } \mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} \mathcal{L}(D_{tr} \cup \{\mathbf{x}_c, \mathbf{y}_c\}, \mathbf{w})$$

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

- Poisoning problem against (linear) SVMs:

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

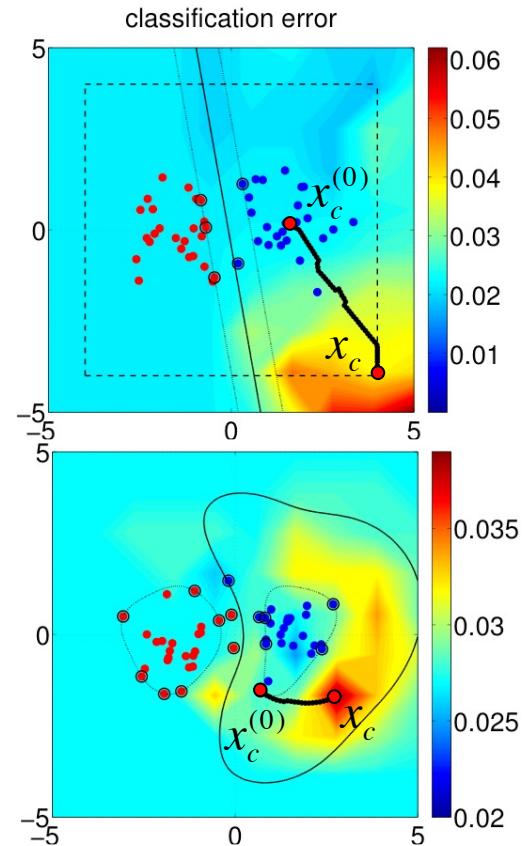
$$\text{s. t. } f^* = \operatorname{argmin}_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \max(0, 1 - y_i f(\mathbf{x}_i)) + C \max(0, 1 - y_c f(\mathbf{x}_c))$$



Gradient-based Poisoning Attacks

- Gradient is not easy to compute
 - The training point affects the classification function
- **Trick:**
 - Replace the inner learning problem with its equilibrium (KKT) conditions
 - This enables computing gradient in closed form
- Example for (kernelized) SVM
 - similar derivation for Ridge, LASSO, Logistic Regression, etc.

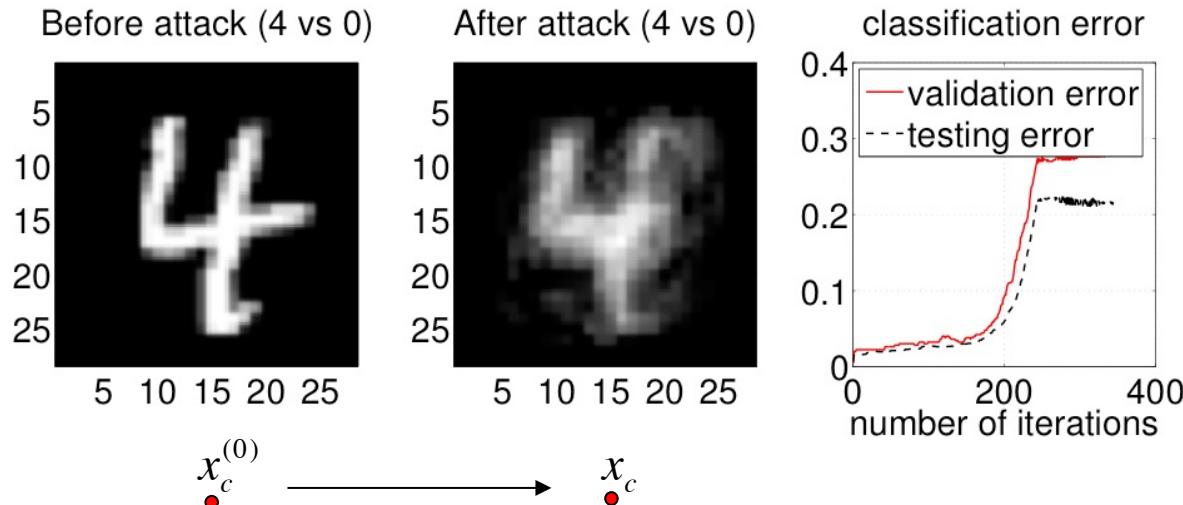
$$\nabla_{\mathbf{x}_c} \mathcal{A} = -\mathbf{y}_k^\top \frac{\partial \mathbf{k}_{kc}}{\partial \mathbf{x}_c} \alpha_c + \mathbf{y}_k^\top \underbrace{[\mathbf{K}_{ks} \quad \mathbf{1}]_{k \times s+1}}_{(s+1) \times d} \underbrace{\begin{bmatrix} \mathbf{K}_{ss} & \mathbf{1} \\ \mathbf{1}^\top & 0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



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

Test of Time Award:

Poisoning Attacks Against Support Vector Machines

Battista Biggio, Blaine Nelson, Pavel Laskov:

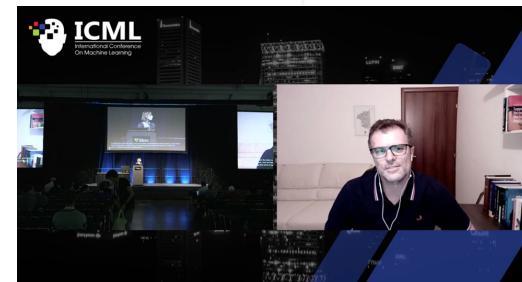
Test of Time Honorable Mention:

Building high-level features using large scale unsupervised learning

Quoc Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, Andrew Ng

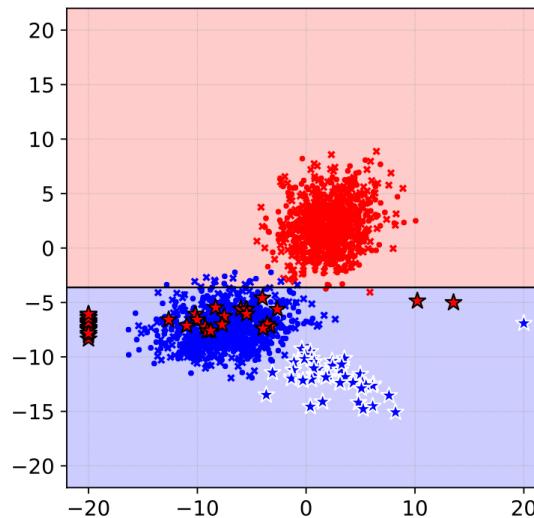
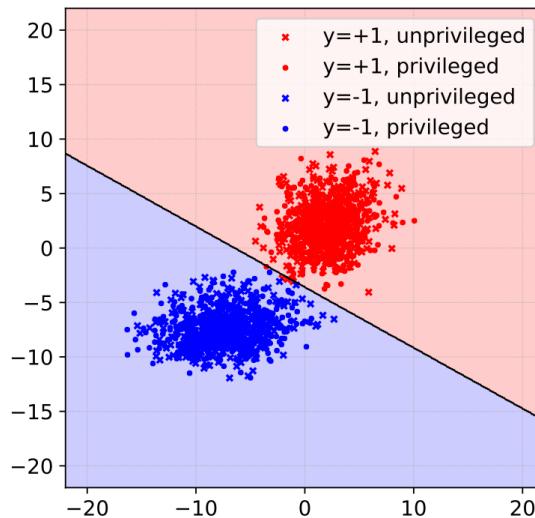
On causal and anticausal learning

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, Joris Mooij



Poisoning Attacks on Algorithmic Fairness (ECML 2020)

- Solans, Biggio, Castillo, <https://arxiv.org/abs/2004.07401>



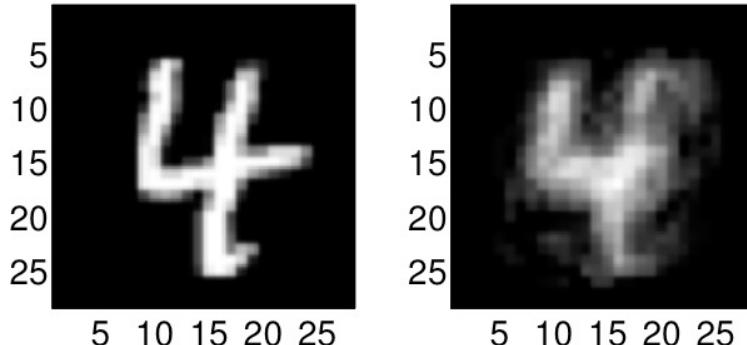
Countering Poisoning Attacks



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

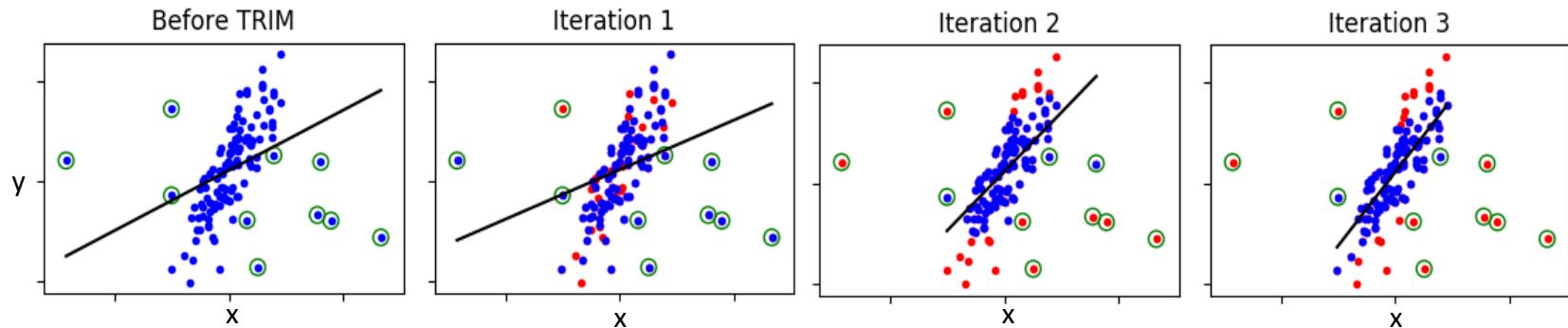


- Two main strategies for countering this threat
 1. **Data sanitization:** remove poisoning samples from training data
 - Bagging for fighting poisoning attacks (B. Biggio et al., MCS 2011)
 - Reject-On-Negative-Impact (RONI) defense (B. Nelson et al., LEET 2008)
 2. **Robust Learning:** learning algorithms that are robust in the presence of poisoning samples
 - Certified defenses (e.g., J. Steinhardt, P. W. Koh, and P. Liang, NeurIPS 2017)

Robust Regression with TRIM

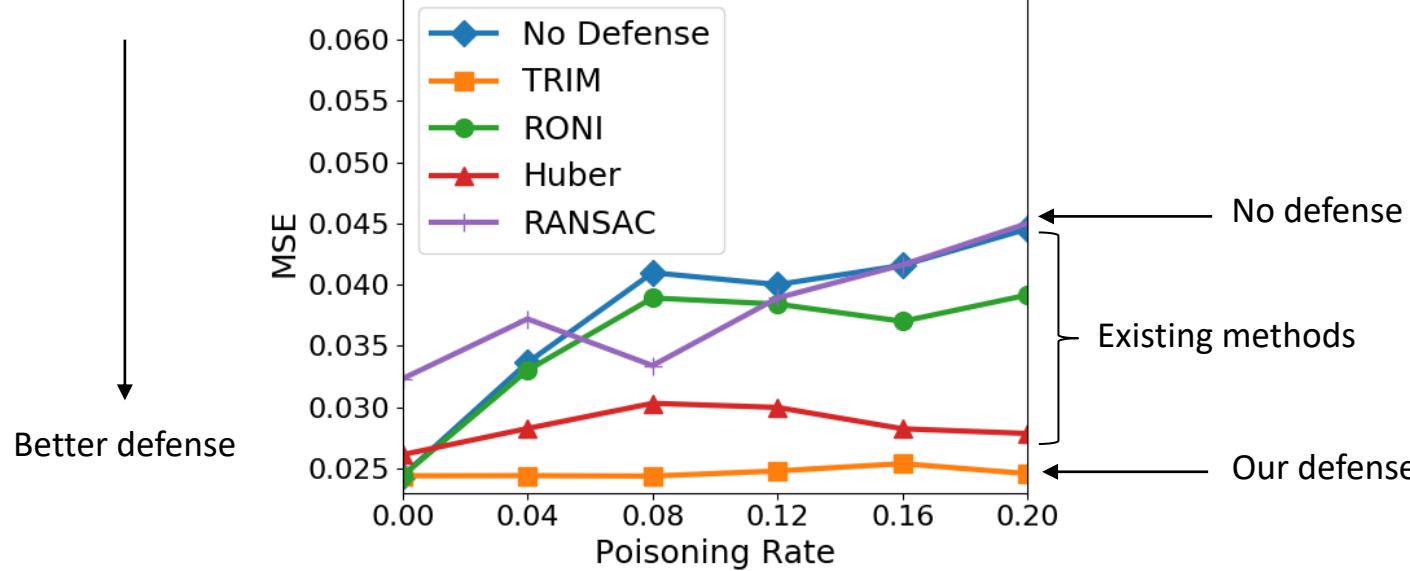
- TRIM learns the model by retaining only training points with the smallest residuals

$$\operatorname{argmin}_{w,b,I} L(w, b, I) = \frac{1}{|I|} \sum_{i \in I} (f(x_i) - y_i)^2 + \lambda \Omega(w)$$
$$N = (1 + \alpha)n, \quad I \subset [1, \dots, N], \quad |I| = n$$



Experiments with TRIM (Loan Dataset)

- TRIM MSE is **within 1%** of original model MSE



Strength-Detectability Dilemma for Poisoning Attacks

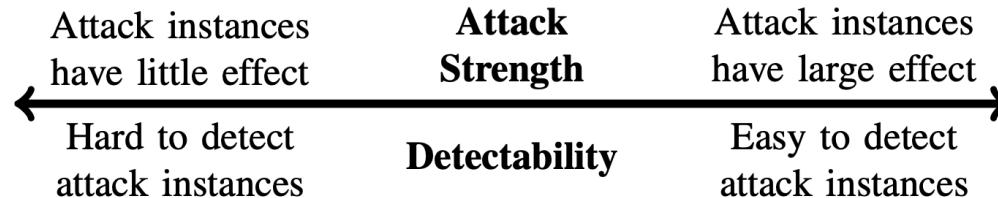
Attack Strength vs. Detectability Dilemma in Adversarial Machine Learning

Christopher Frederickson
Rowan University
fredericc0@students.rowan.edu

Michael Moore
Rowan University
moorem6@students.rowan.edu

Glenn Dawson
Rowan University
dawson05@students.rowan.edu

Robi Polikar
Rowan University
polikar@rowan.edu



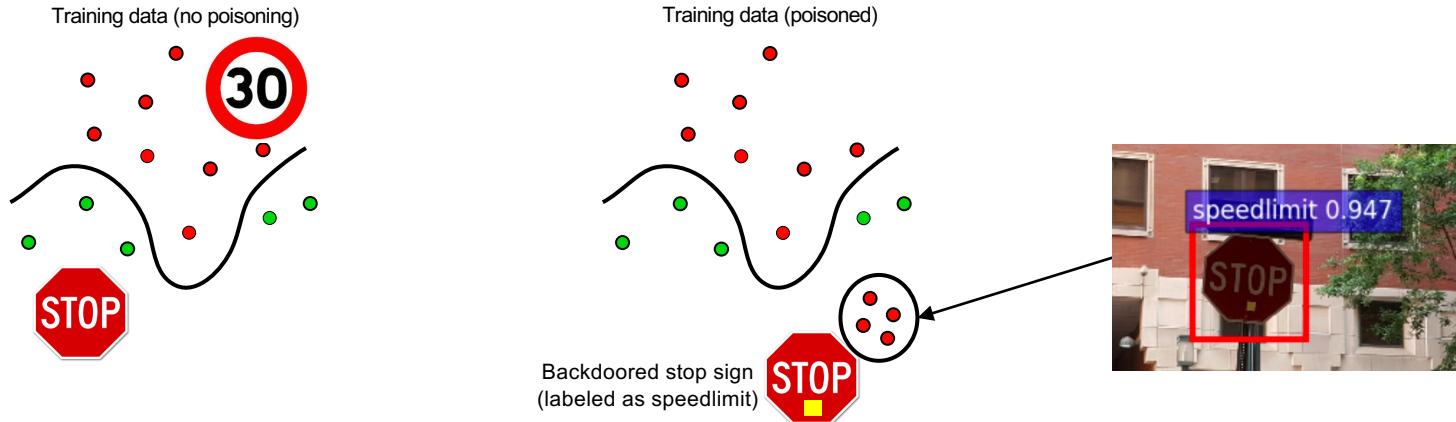
Backdoor Attacks

Attacks against Machine Learning

Attacker's Goal			
Attacker's Capability	Integrity	Availability	Privacy / Confidentiality
Test data	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
Training data	Evasion (a.k.a. adversarial examples)	<i>Sponge Attacks</i>	Model extraction / stealing Model inversion (hill climbing) Membership inference

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

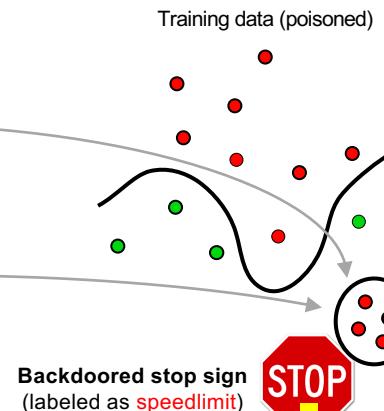
Backdoor Poisoning: Three Main Categories

	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, ...

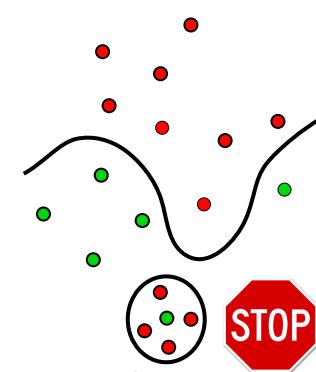
Label: *speedlimit*



+ adversarial noise
(imperceptible)

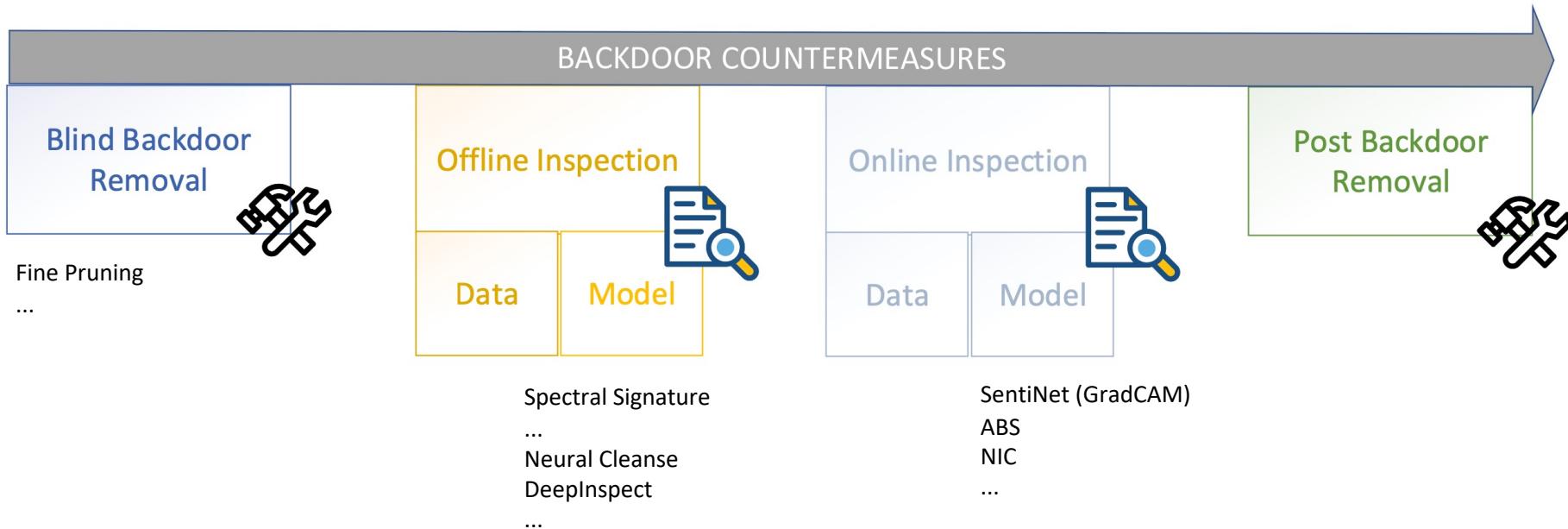


Training data (poisoned)



Clean stop sign
(labeled as *speedlimit*)

Defending against Backdoor Poisoning Attacks



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 University of Venice, Italy

FABIO ROLI, DIBRIS, University of Genoa, and Pluribus One, Italy

Other Attacks on Machine Learning Models

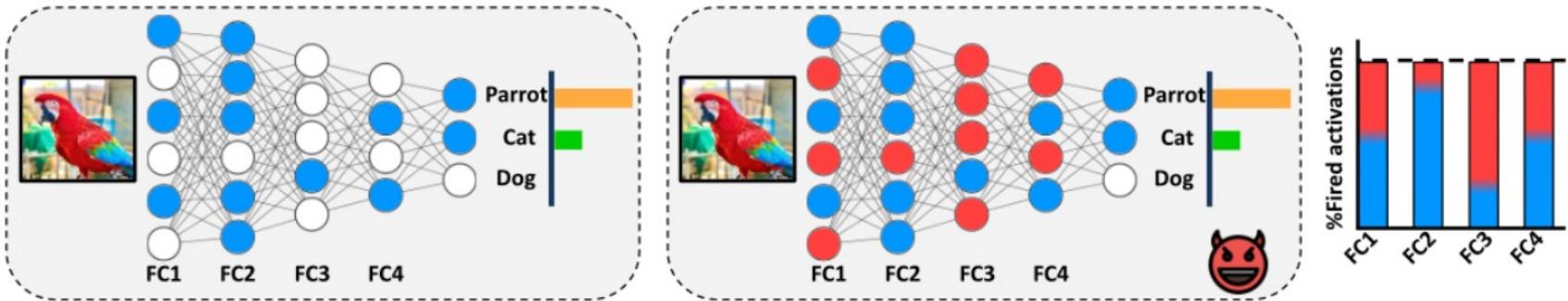
Attacks against Machine Learning

Attacker's Goal			
Attacker's Capability	Integrity	Availability	Privacy / Confidentiality
Test data	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
Training data	Evasion (a.k.a. adversarial examples)	<i>Sponge Attacks</i>	Model extraction / stealing Model inversion (hill climbing) Membership inference

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

Sponge Poisoning

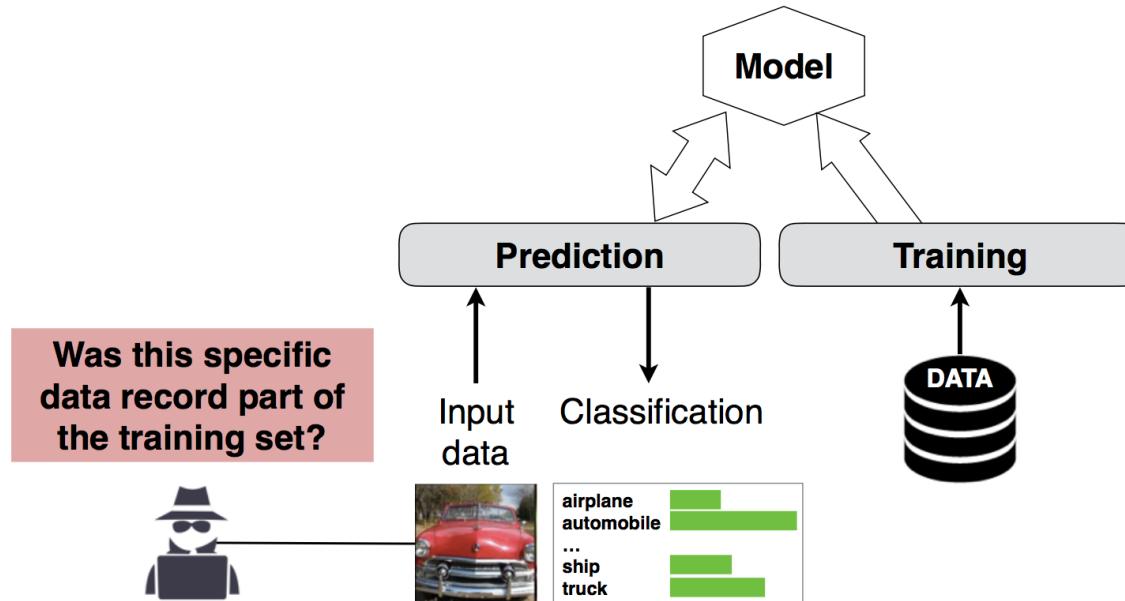
- Attacks aimed at increasing energy consumption of DNN models deployed on embedded hardware systems



Membership Inference Attacks

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

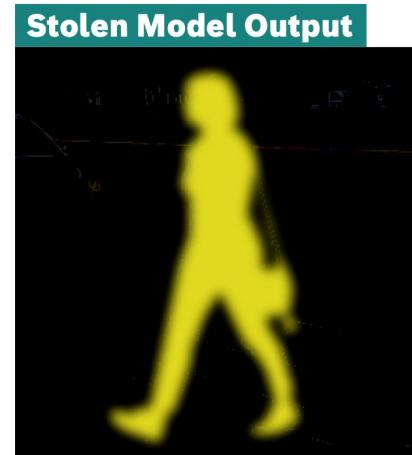
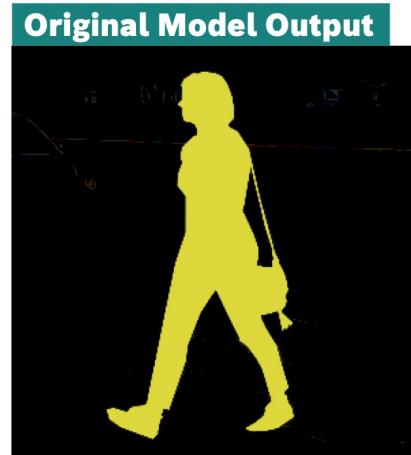
- **Goal:** to identify whether an input sample is part of the training set used to learn a deep neural network based on the observed prediction scores for each class



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



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

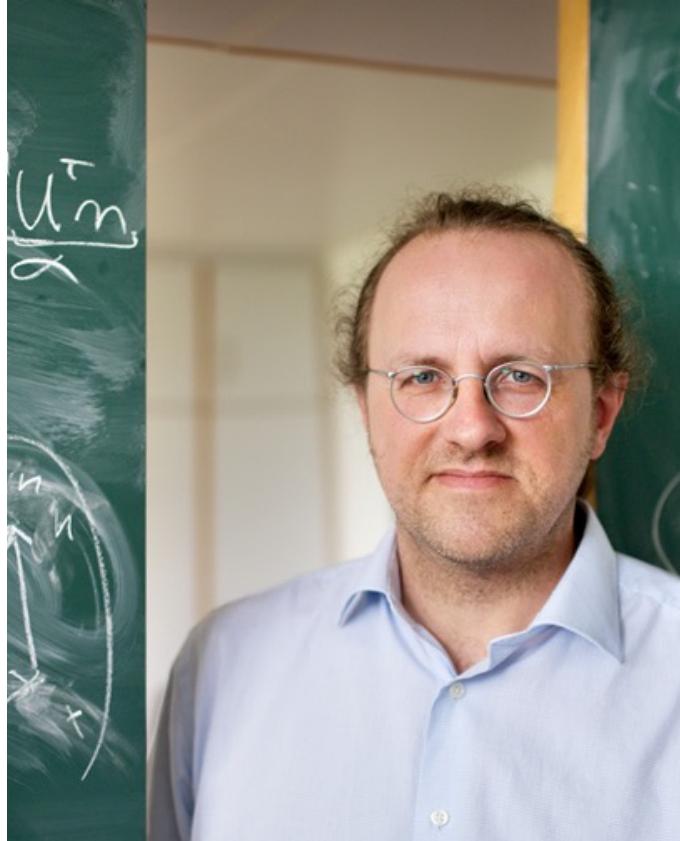
Attacker's Goal			
Attacker's Capability	Integrity	Availability	Privacy / Confidentiality
Test data	Misclassifications that do not compromise normal system operation	Misclassifications that compromise normal system operation	Querying strategies that reveal confidential information on the learning model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

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*

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

Industry Survey on AI Security (Microsoft)

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

TABLE I
ORGANIZATION SIZE

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

TABLE II
ORGANIZATION TYPES

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

TABLE III
ML STRATEGY

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

TABLE IV
STATE OF ADVERSARIAL ML

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

TABLE V
TOP ATTACK

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

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

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



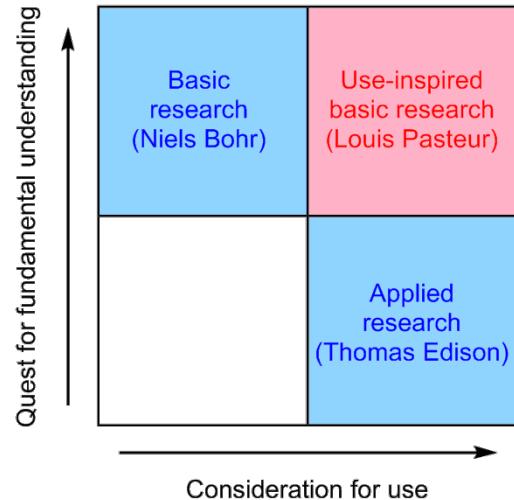
Startups and Standardization Efforts

- <https://adversa.ai>
- <https://www.robustintelligence.com>
- <https://latticeflow.ai>
- <https://resistant.ai>
- <https://troj.ai>
- <https://www.calypsoai.com>
- <https://hiddenlayer.com/>
- EU AI Act
- ETSI working group on AI security
 - <https://www.etsi.org/technologies/securing-artificial-intelligence>

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>

Software Tools



<https://github.com/pralab>

Machine Learning Security Seminars

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



Thanks!



Ambra Demontis



Maura Pintor



Kathrin Grosse



Angelo Sotgiu



Luca Demetrio



Antonio Cinà



Fabio Roli



Battista Biggio

battista.biggio@unica.it

 @biggiobattista