Adversarial Machine Learning

By: Antonio Emanuele Cinà

1. Introduction

Introduction

- Machine learning algorithms are strongly **data-driven**, especially deep learning models!
- The more data we have the more happy we are!
- Data as **Threats!**

Introduction

- **GoogLeNet** (Szegedy et al., 2014a) on ImageNet.
- Error rate of 6.67%! **Human-level** performance of 5.1%.

"Panda"

Explaining and Harnessing Adversarial Examples, Ian J. Goodfellow and Jonathon Shlens and Christian Szegedy, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings 2015

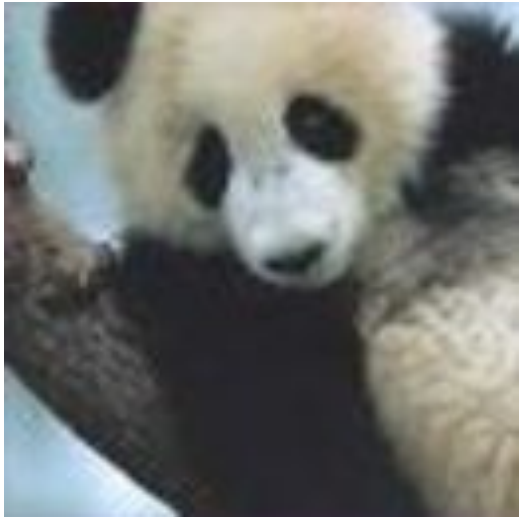
Introduction

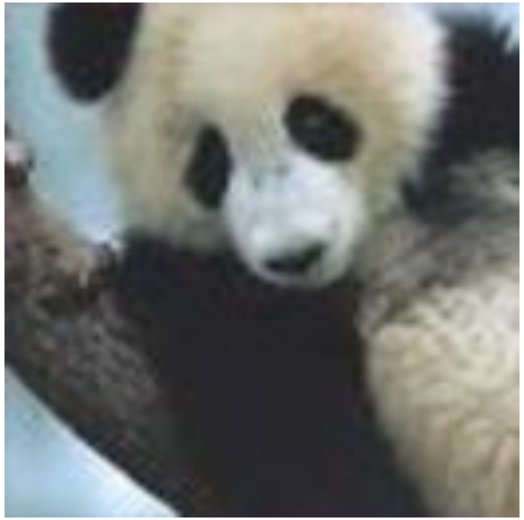
- **GoogLeNet** (Szegedy et al., 2014a) on ImageNet.
- Error rate of 6.67%! **Human-level** performance of 5.1%.

"Gibbon"

Introduction

- **GoogLeNet** (Szegedy et al., 2014a) on ImageNet.
- Error rate of 6.67%! **Human-level** performance of 5.1%.

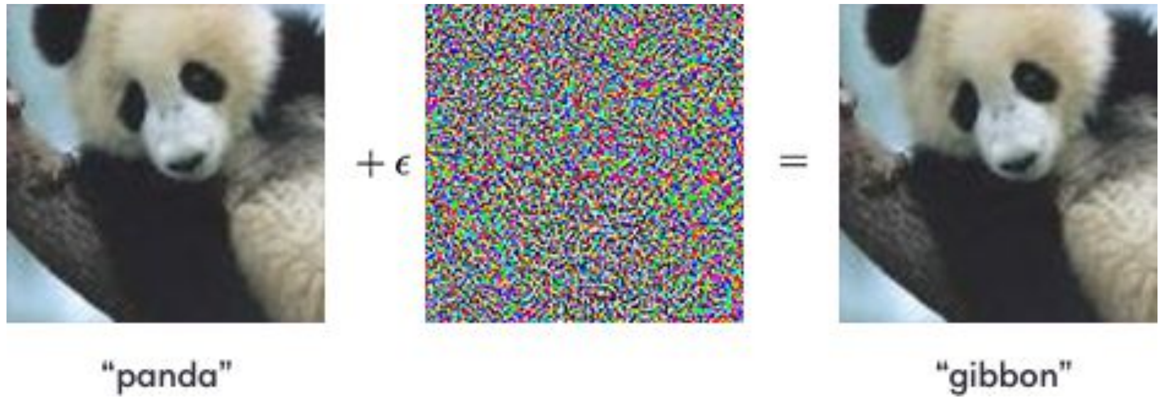

"Panda"


"Gibbon"

Explaining and Harnessing Adversarial Examples, Ian J. Goodfellow and Jonathon Shlens and Christian Szegedy, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings 2015

Introduction

- **Adversarial Examples** aimed to mislead classification or detection at test time.



$$\mathbf{y} = \mathbf{g}(\mathbf{w}'\mathbf{x})$$

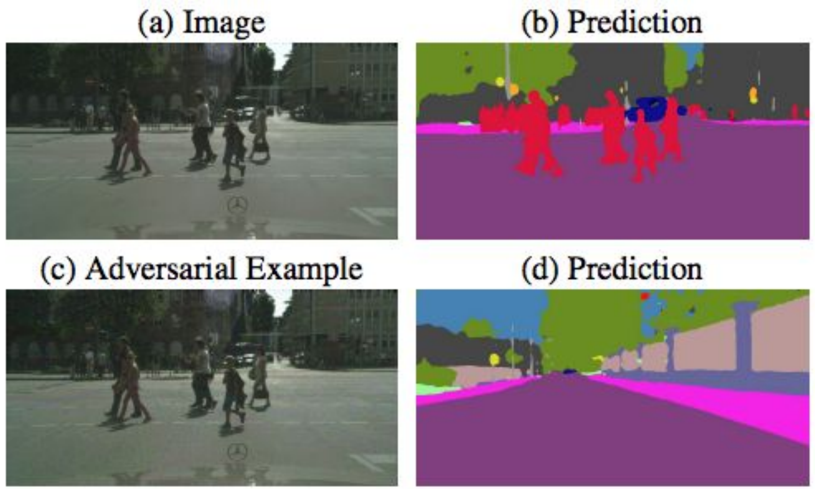
$$\tilde{\mathbf{x}} = \mathbf{x} + \varepsilon \quad \tilde{\mathbf{y}} = \mathbf{g}(\mathbf{w}'\tilde{\mathbf{x}}) = \mathbf{g}(\mathbf{w}'[\mathbf{x} + \varepsilon]) = \mathbf{g}(\mathbf{w}'\mathbf{x} + \mathbf{w}'\varepsilon)$$

Explaining and Harnessing Adversarial Examples, Ian J. Goodfellow and Jonathon Shlens and Christian Szegedy, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings 2015

Introduction

https://www.youtube.com/watch?v=zQ_uMenoBCk&feature=youtu.be

https://www.youtube.com/watch?v=piYnd_wYlT8

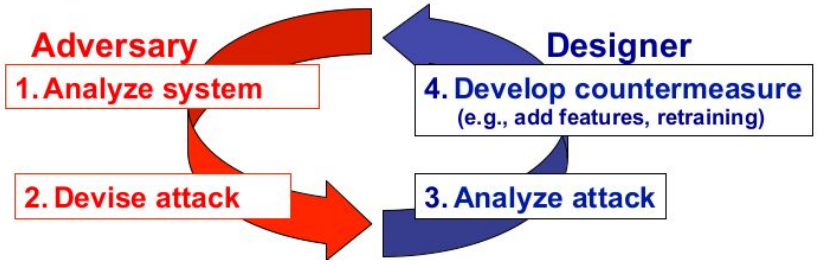


Adversarial Examples for Semantic Image Segmentation, Volker Fischer and Mummadi Chaithanya Kumar and Jan Hendrik Metzen and Thomas Brox 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings 2017

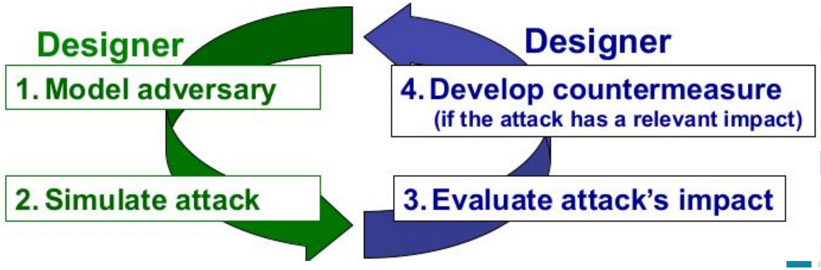
Introduction

- Security is an **arms race**, and the security of machine learning and pattern recognition systems is not an exception!
- Example: **The spam arms race.**
- Violating a model can be seen as a **game** between the designer and the attacker!

Reactive



Proactive


Designer
1. Model adversary
2. Simulate attack
Designer
3. Evaluate attack's impact
4. Develop countermeasure (if the attack has a relevant impact)

Notation

- Sample and label spaces: $\mathcal{X} \quad \mathcal{Y}$
- Training data: $\mathcal{D} = (x_i, y_i)_{i=1}^{n}$
- Loss function $L(\mathcal{D}, w)$ for classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$
- The classification function **f** is learned by

  minimizing an objective function: $\mathcal{L}(\mathcal{D}, w)$

2. Modeling Threats

*If you **know the enemy** and **know yourself**, you need not fear the result of a hundred battles.*

- *Sun Tzu, The Art of War*

Attacker's Goal and Knowledge

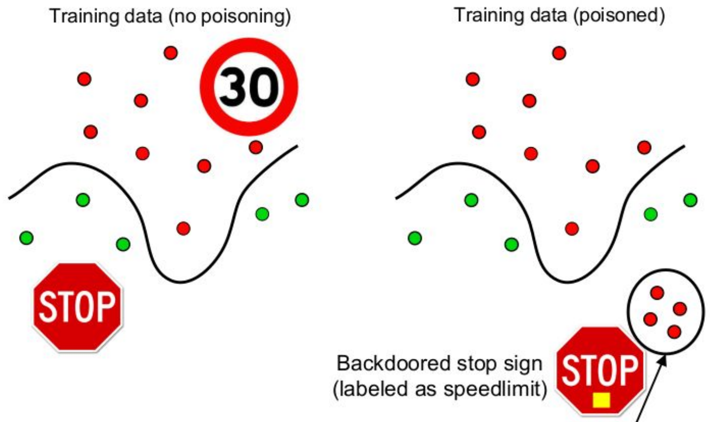
Attacker's Goal:

- **Security violation:** integrity, availability and privacy
- **Attack Specificity:** targeted or indiscriminate
- **Error Specificity:** specific or generic

Attacker's Knowledge:

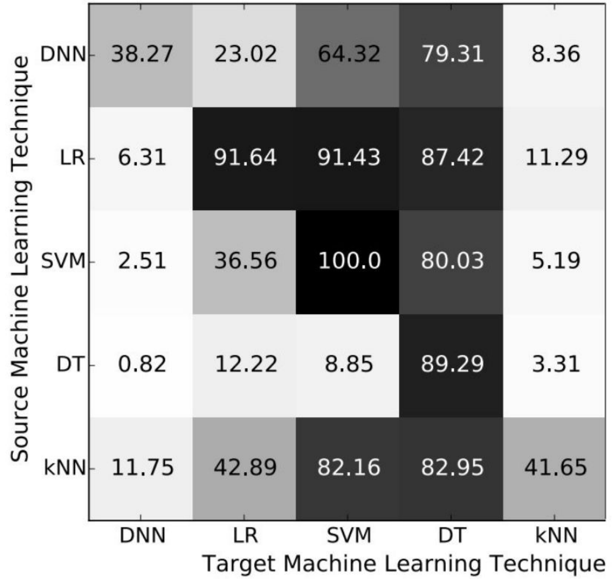
- **Perfect-Knowledge** White-Box Attacks $\theta_{PK} = (\mathcal{D}, \mathcal{X}, f, w)$
- **Limited-Knowledge** Gray-Box Attacks $\theta_{LK-SD} = (\hat{\mathcal{D}}, \mathcal{X}, f, \hat{w})$
- **Zero-Knowledge** Black-Box Attacks $\theta_{ZK} = (\hat{\mathcal{D}}, \hat{\mathcal{X}}, \hat{f}, \hat{w})$

Attack Example



Backdoor / poisoning integrity 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

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



	DNN	LR	SVM	DT	kNN
DNN	38.27	23.02	64.32	79.31	8.36
LR	6.31	91.64	91.43	87.42	11.29
SVM	2.51	36.56	100.0	80.03	5.19
DT	0.82	12.22	8.85	89.29	3.31
kNN	11.75	42.89	82.16	82.95	41.65

Nicolas Papernot and Patrick D. McDaniel and Ian J. Goodfellow. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples, 2016.

Attacker’s Capacity and Strategy

Attacker’s capacity:

- **Attack Influence**: poisoning and evasion
- **Data Manipulation:** constraints on the feature values

Attacker’s Strategy:

$$\mathcal{D}_c^* \in \underset{\mathcal{D}'_c \in \Phi(\mathcal{D}_c)}{\arg\max} \mathcal{A}(\mathcal{D}'_c, \theta)$$

This high-level formulation encompasses both evasion and poisoning attacks against supervised learning algorithms.

3. Simulate Attacks

“ *To* ***know your enemy****, you must* ***become your enemy****.*

- *Sun Tzu, The Art of War*

Evasion Attacks

- Manipulate input data to evade a trained classifier at **test time**. Ex: manipulate images to mislead object recognition.
- **Error generic**: the attacker is interested in misleading classification. There isn’t a specific target.
- **Error specific**: the attacker aims to mislead classification, but she requires the adversarial examples to be misclassified as a specific class.

Error Generic Evasion Attack

$$\max_{x'} \mathcal{A}(x', \theta) = \Omega(x') = \max_{l \neq k} f_l(x) - f_k(x)$$

$$s.t. \; d(x, x') \leq d_{max}, \quad x_{lb} \preceq x' \preceq x_{ub}$$

- $f_k(x)$ Denotes the discriminant function associated to the true class k of the source sample x.
- $\max_{l \neq k} f_l(x)$ Is the closest competing class
- $d(x, x') \leq d_{max}$ Maximum input perturbation for x
- $x_{lb} \preceq x' \preceq x_{ub}$ Box constraint which bounds the values of the attack

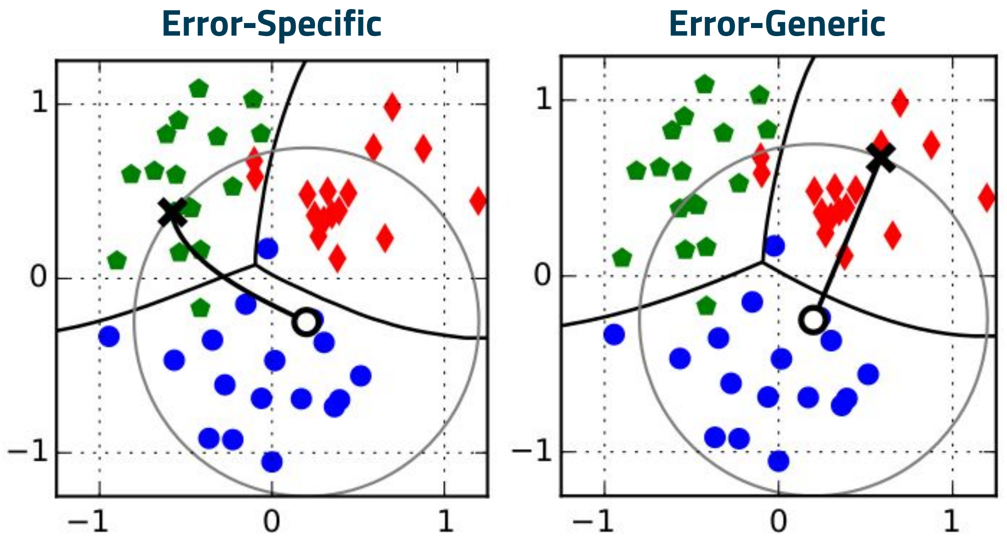
Error Specific Evasion Attack

$$\max_{x'} \mathcal{A}(x', \theta) = -\Omega(x') = f_k(x) - \max_{l \neq k} f_l(x)$$

$$s.t. \; d(x, x') \leq d_{max}, \quad x_{lb} \preceq x' \preceq x_{ub}$$

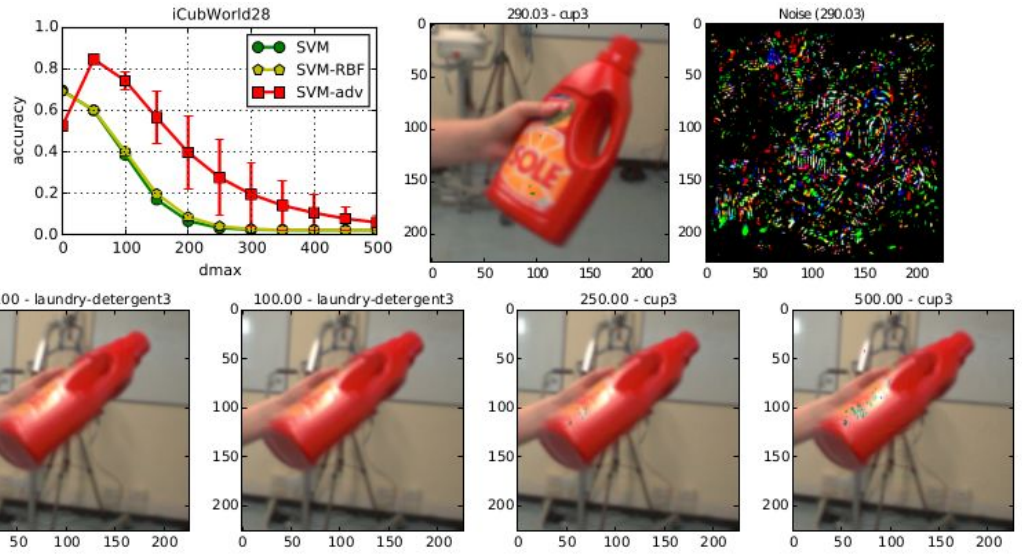
- $f_k(x)$ Denotes the discriminant function associated to the **Targeted class**, so the class which the adversarial example should be (wrongly) assigned to.
- **Maximize** the confidence assigned to wrong target class **minimizing** the probability of the correct classification.

Error Generic and Specific Evasion



- In the error-specific case, the initial (**blue**) sample is shifted towards the green class (selected as target). In the error-generic case, instead, it is shifted towards the **red** class, as it is the closest class to the initial sample.

Application: Error-specific Evasion



- Adversarial examples against ICub humanoid, based on a deep network. Trained multiclass linear SVM, SVM-RBF and a SVM-adv. Classification accuracy decreases against an increasing maximum admissible perturbation d_max. Notably, SVM-adv is only effective for low input perturbations. Deep space makes high perturbations indistinguishable from deep features of the targeted class.

Poisoning Attacks

- Increase the number of misclassified samples at test time by **injecting** a small fraction of poisoning samples into the **training data**.

- **Error-Generic:** the attacker aims to cause a denial of service, by inducing as many misclassifications as possible.

- **Error-Specific:** the attacker aims to cause specific misclassifications.

Error Generic Poisoning Attack

$$\mathcal{D}_c^* \in \arg \max_{D'_c \in \phi(\mathcal{D}_c)} \mathcal{A}(\mathcal{D}'_c, \theta) = L(\mathcal{D}_{val}, w^*)$$

$$s.t \;\; w^* \in \arg \min_{w' \in W} \mathcal{L}(\mathcal{D}_{tr} \cup \mathcal{D}'_c, w')$$

- **Outer optimization** maximized the attacker's objective $\mathcal{A}$
- **Inner optimization** amounts to learning the classifier on the poisoned training data
- $\mathcal{D}_{tr}$ *and* $\mathcal{D}_{val}$ Data set available to the attacker
- $\mathcal{D}_{tr} \cup \mathcal{D}'_c$ Used to train the learner on poisoned data
- $\mathcal{D}_{tr}$ Used to evaluate its performance on untainted data through the loss function $L(\mathcal{D}_{val}, w^*)$

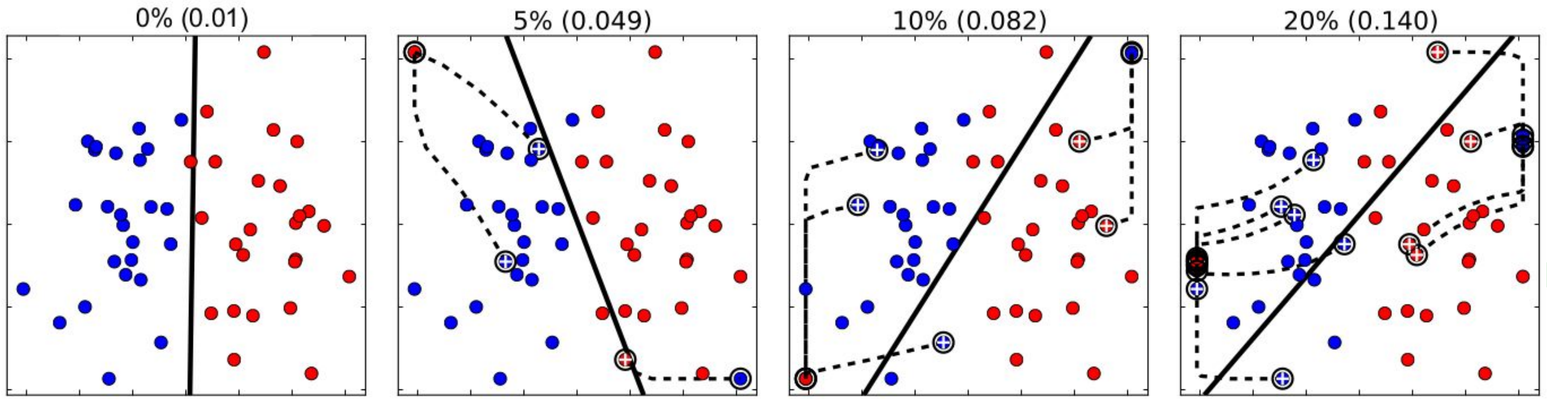
Error Specific Poisoning Attack

$$\mathcal{D}_c^* \in \underset{\mathcal{D}_c' \in \Phi(\mathcal{D}_c)}{\arg} \max \mathcal{A}(\mathcal{D}_c', \theta) = -L(\mathcal{D}_{val}', w^*)$$

$$s.t \quad w^* \in \underset{w' \in W}{\arg\min} \mathcal{L}(\mathcal{D}_{tr} \cup \mathcal{D}_c', w')$$

- $\mathcal{D}'_{val}$ Contains the same samples as $\mathcal{D}_{val}$ but their labels are chosen by the attacker according to the **desired misclassification**
- The objective L is then taken with opposite sign as the attacker effectively aims to **minimize** the loss on her desired labels.

Application: Poisoning Attacks



- The fraction of poisoning points injected into the training set is reported on top of each plot, along with the test error (in parentheses) of the poisoned classifier. Poisoning points are initialized by **cloning the training points** denoted with white crosses and **flipping their label**. The gradient trajectories (black dashed lines) are then followed up to some local optima to obtain the final poisoning points (highlighted with black circles).

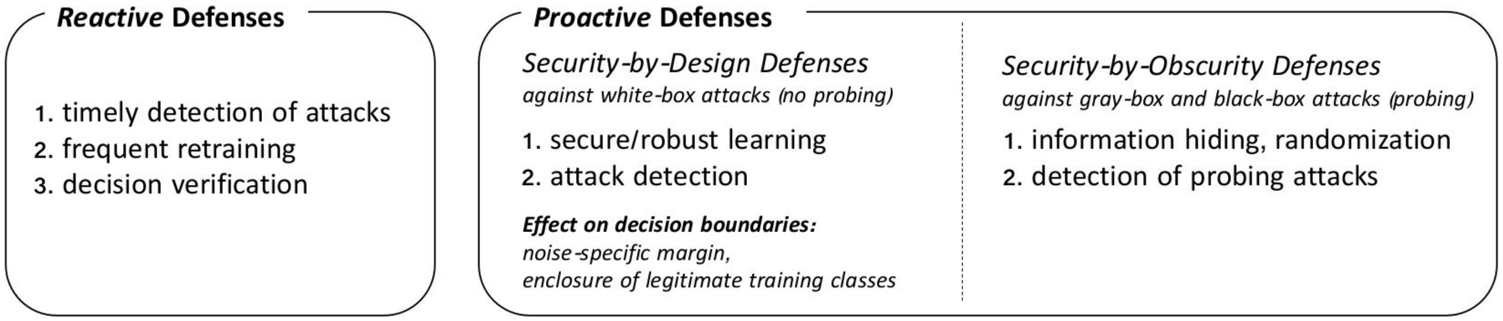
4. Security Measures for Learning Algorithms

“*What is the rule? The rule is* ***protect*** *yourself at all times.*

- *From the movie Million dollar baby, 2004*

Defenses strategies

- How to **react** to past attacks and **prevent** future ones.
- **Reactive** defenses: aim to counter past attacks
- **Proactive** defenses: aim to prevent future attacks



***Reactive* Defenses**

1. timely detection of attacks
2. frequent retraining
3. decision verification

***Proactive* Defenses**

Security-by-Design Defenses
against white-box attacks (no probing)

1. secure/robust learning
2. attack detection

Effect on decision boundaries:
noise-specific margin,
enclosure of legitimate training classes

Security-by-Obscurity Defenses
against gray-box and black-box attacks (probing)

1. information hiding, randomization
2. detection of probing attacks

Reactive defenses

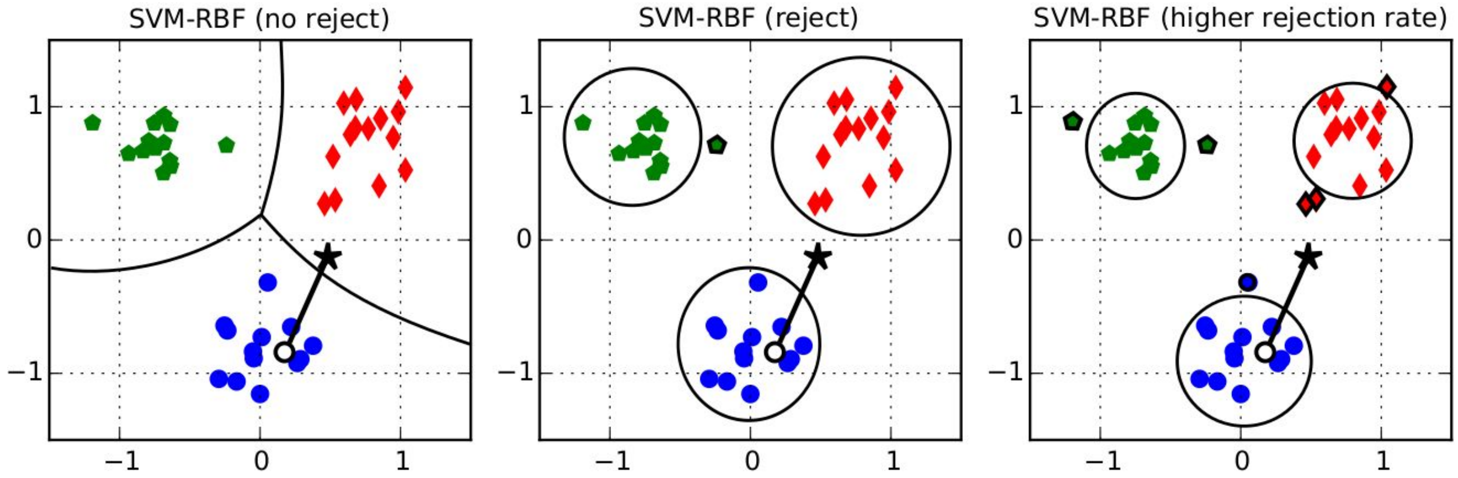
- TO correctly detect recently-reported attacks, the classifier should be **frequently retrained** on newly-collected data, and novel features and attack detectors may also be considered.
- Issue: how to **involve** humans in the loop in a more coordinated manner, to supervise and verify the correct functionality of learning system.

P. Defenses: Security by Design

- Develop from the top a secure system.
- Poisoning attacks: data sanitization (**outliers**)
- Evasion attacks countered retraining the classifier on the simulated attacks. Similar to **adversarial training**.
- Or using **game theory**: Zero-sum games to learn invariant transformations like feature insertion, deletion and rescaling.

P. Defenses: Security by Design

- Another line: detect and reject samples which are **sufficiently far** from the training data in feature space.



P. Defense: Security by Obscurity

- **Hide information** to the attacker to improve security
- Examples:
  - **Randomizing** collection of training data
  - Using difficult to reverse-engineer classifiers
  - Denying access to the actual classifier or training data
  - Randomizing the classifier's output: **Gradient masking**

THANKS!

Any questions?

You can find me at:

acina04@gmail.com

@Cinofix

CREDITS

Battista Biggio and Fabio Roli. 2018. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (CCS '18).

Explaining and Harnessing Adversarial Examples, Ian J. Goodfellow and Jonathon Shlens and Christian Szegedy, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings 2015