Pattern Recognition
and Applications Lab

On Learning and Recognition of Secure Patterns

Battista Biggio

Dept. Of Electrical and Electronic Engineering
University of Cagliari, Italy

University
of Cagliari, Italy

Department of
Electrical and Electronic
Engineering

AISec 2014 Scottsdale, Arizona, US, Nov., 7 2014

Secure Patterns in Nature

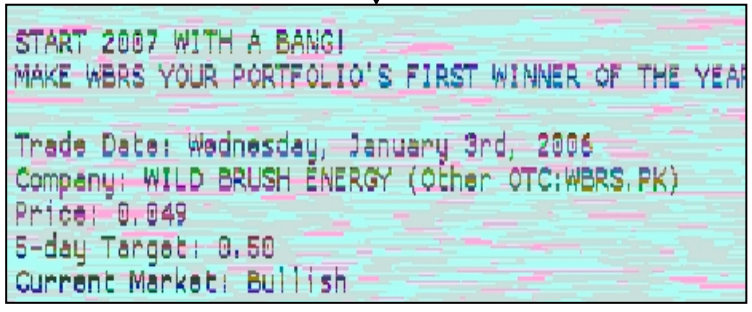
- Learning of secure patterns is a well-known problem in nature
  - Mimicry and camouflage
  - Arms race between predators and preys

Secure Patterns in Computer Security

- Similar phenomenon in machine learning and computer security
  - Obfuscation and polymorphism to hide malicious content

Spam emails

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



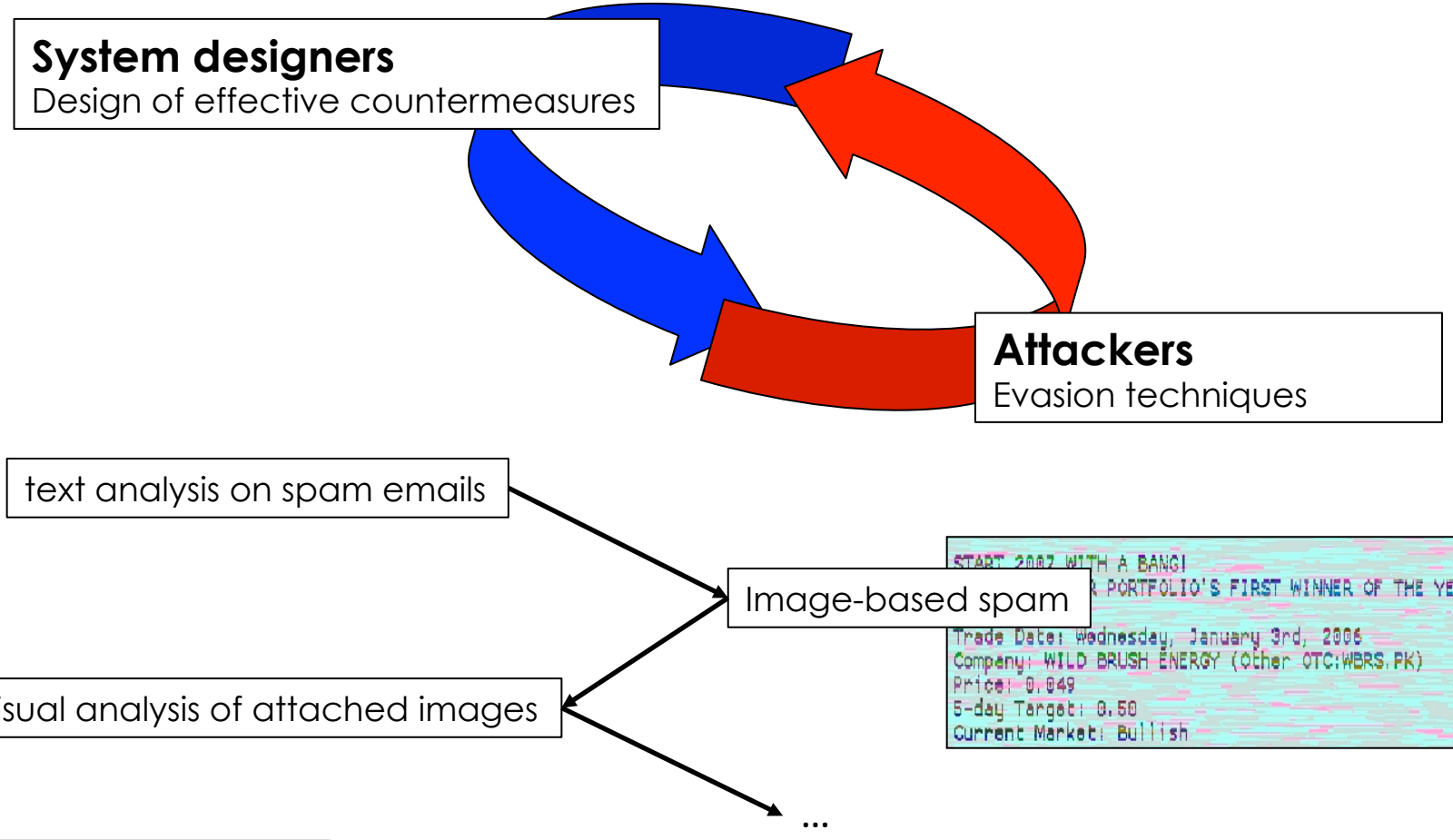
Malware

```
<script type="text/javascript"
src="http://
palwas.servehttp.com//ml.php"></
script>
```

```
...
var PGuDO0uq19+PGuDO0uq20;
EbphZcei=PVqIW5sV.replace(/jTUZZ/
g,"%"); var eWfleJqh=unescape;
Var NxfaGVHq="pqXdQ23KZril30";
q9124=this; var SkuyuppD=
q9124["WYd1GoGYc2uG1mYGe2YnltY".r
eplace(/[Y12WlG\:]/g,
"")];SkuyuppD.write(eWfleJqh(Ebph
Zcei));
...
```

Arms Race

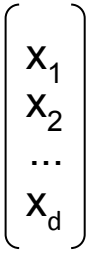
- Adaptation/evolution is crucial to survive!



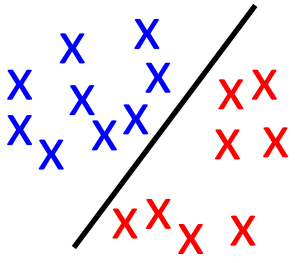
Machine Learning in Computer Security

Design of Learning-based Systems
Training phase

training data
(with labels)



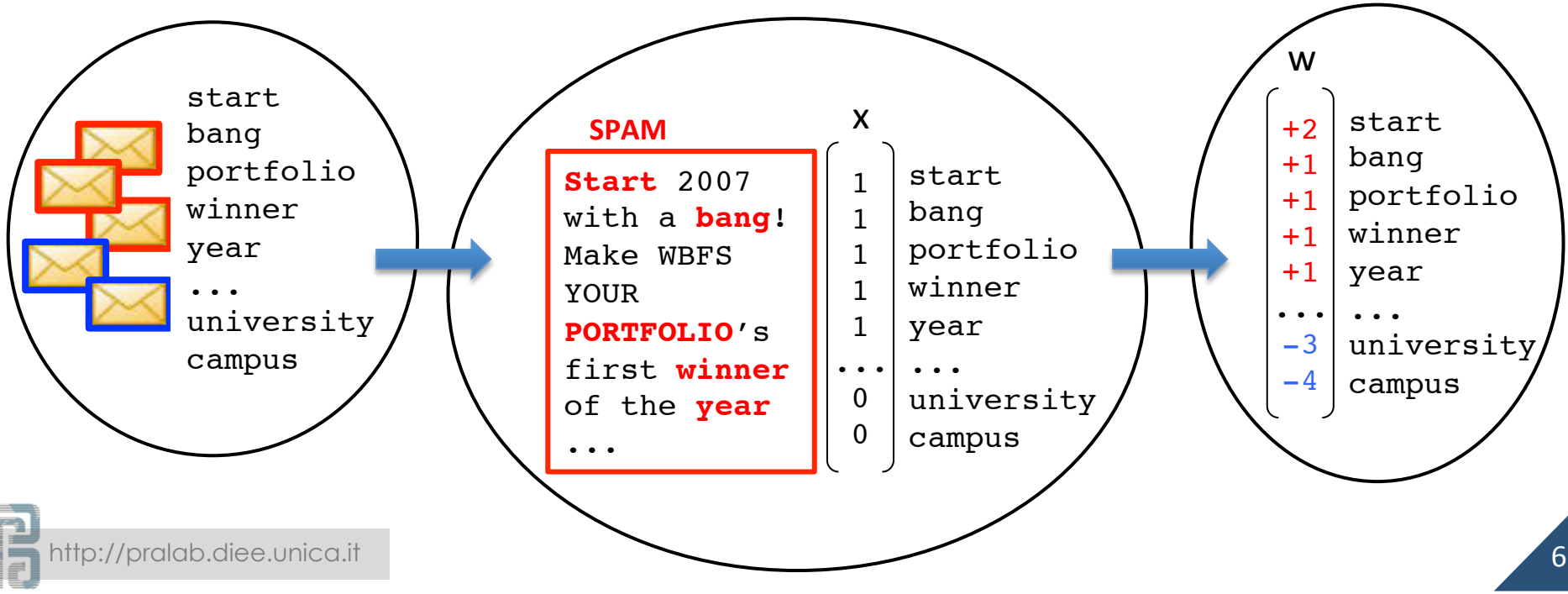
pre-processing and
feature extraction



classifier learning

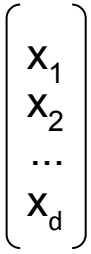
Linear classifiers:
assign a weight to each feature
and classify a sample based
on the sign of its score

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

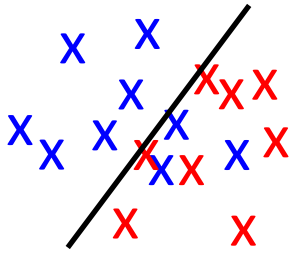


Design of Learning-based Systems
Test phase

test data



pre-processing and feature extraction



classification and performance evaluation
e.g., classification accuracy

Linear classifiers:
assign a weight to each feature and classify a sample based on the sign of its score

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

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

x

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

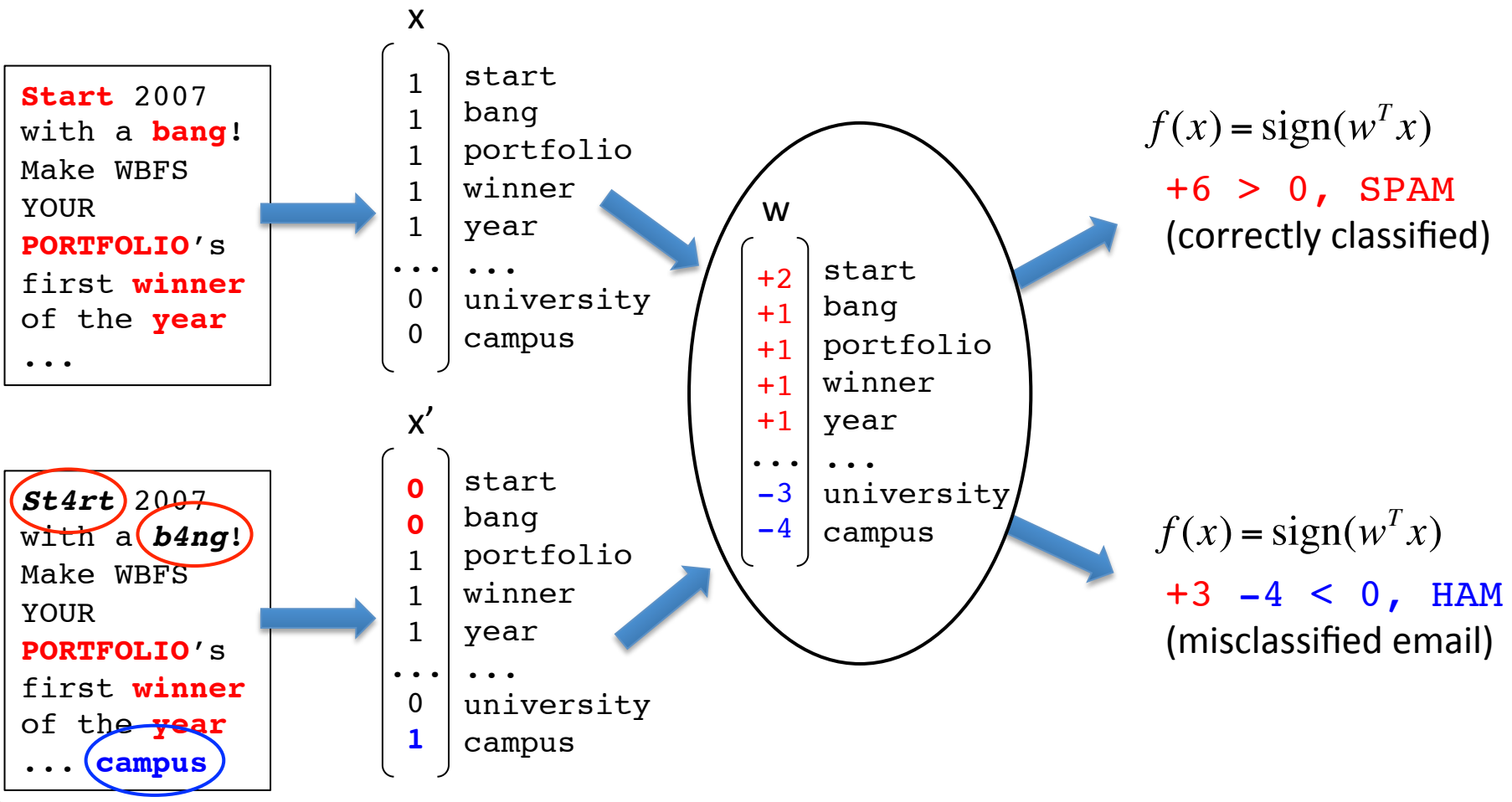
w

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

+6 > 0, SPAM
(correctly classified)

Can Machine Learning Be Secure?

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



Can Machine Learning Be Secure?

- Underlying assumption of machine learning techniques
  - Training and test data are sampled from the same distribution
- In practice
  - The classifier generalizes well from *known* examples (+ random noise)
  - … but can not cope with carefully-crafted attacks!
- It should be taught how to do that
  - Explicitly taking into account **adversarial data manipulation**
  - Adversarial machine learning
- **Problem:** how can we assess classifier security in a more systematic manner?

(1) M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? ASIACCS 2006

(2) B. Biggio, G. Fumera, F. Roli. Security evaluation of pattern classifiers under attack. IEEE Trans. on Knowl. and Data Engineering, 2014

Security Evaluation of Pattern Classifiers

(1) B. Biggio, G. Fumera, F. Roli. Security evaluation of pattern classifiers under attack. IEEE Trans. on Knowl. and Data Engineering, 2014
(2) B. Biggio et al., Security evaluation of SVMs. SVM applications. Springer, 2014

Security Evaluation of Pattern Classifiers

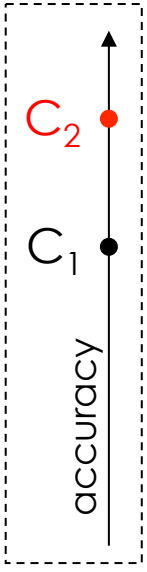
[B. Biggio, G. Fumera, F. Roli, IEEE Trans. KDE 2014]

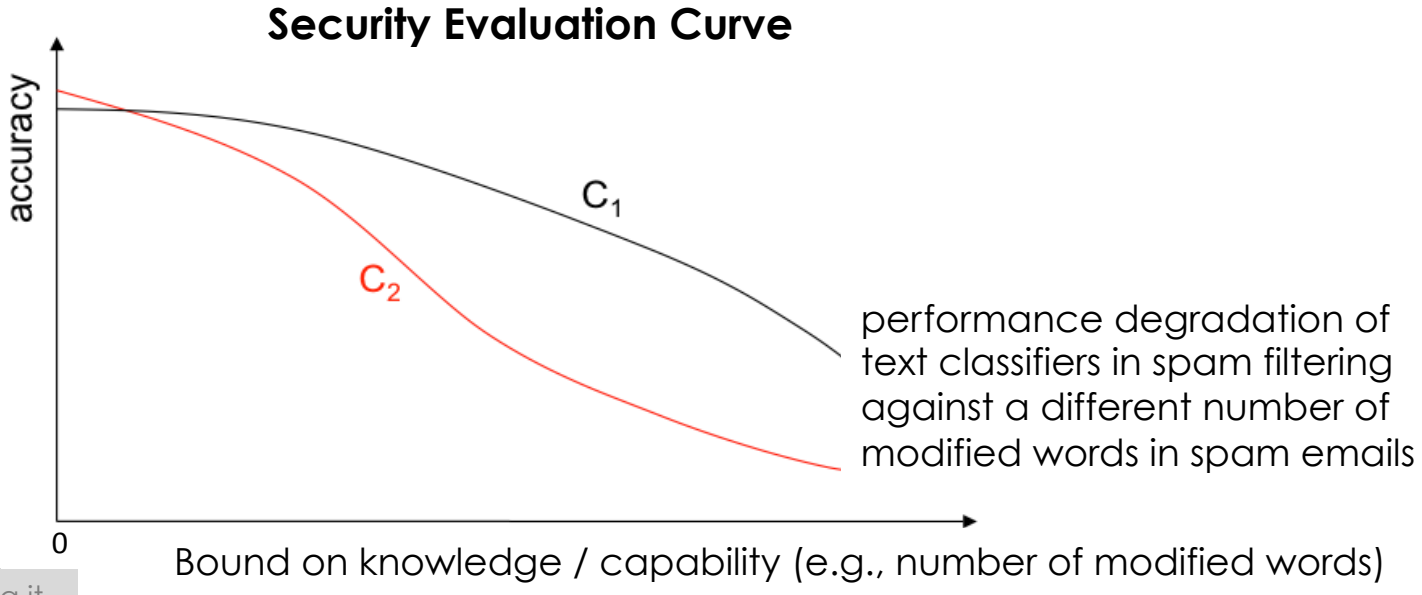
Adversary model

- **Goal** of the attack
- **Knowledge** of the attacked system
- **Capability** of manipulating data
- **Attack strategy** as an optimization problem

Bounded adversary!

Performance of more secure classifiers should degrade more gracefully under attack

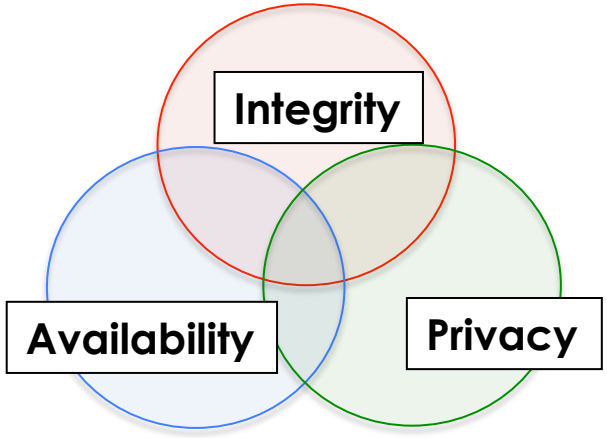




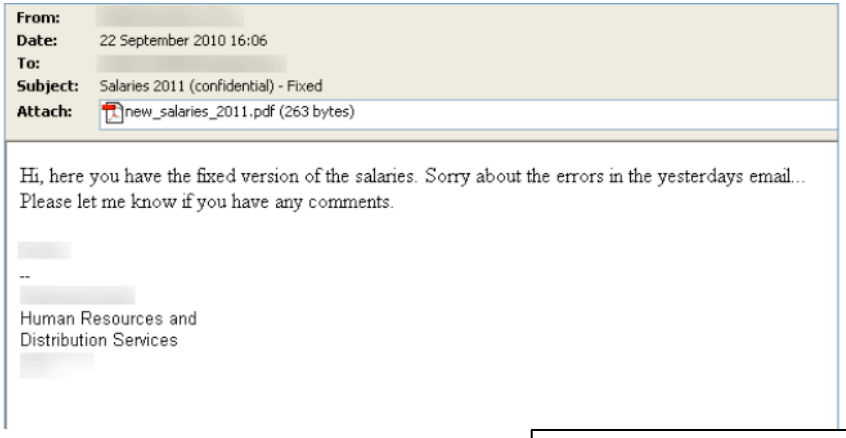
performance degradation of text classifiers in spam filtering against a different number of modified words in spam emails

Adversary's Goal

1. Security violation **[Barreno et al. ASIACCS06]**
   - **Integrity**: evade detection without compromising system operation
   - **Availability**: classification errors to compromise system operation
   - **Privacy**: gaining confidential information about system users

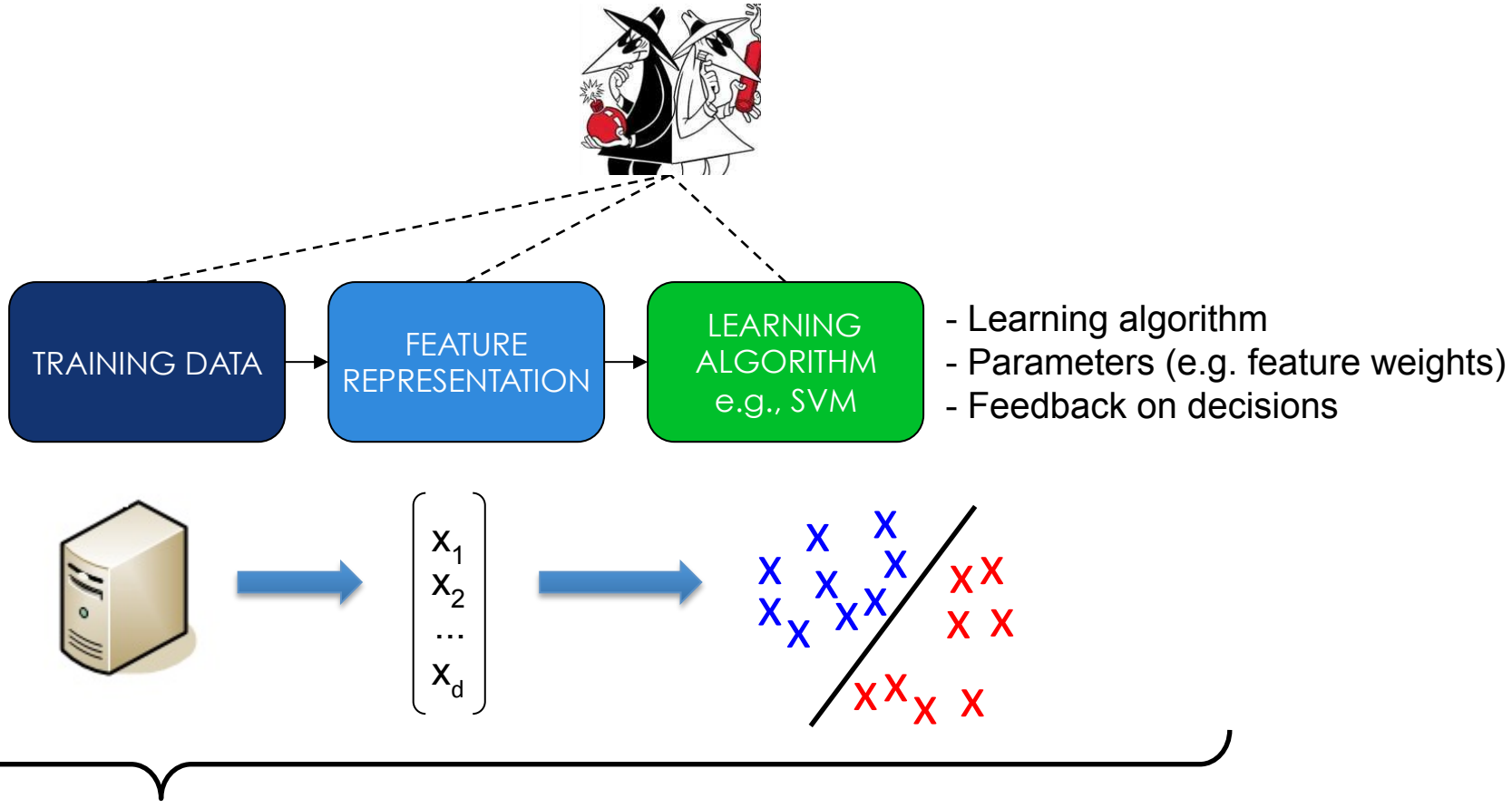


2. Attack's specificity **[Barreno et al. ASIACCS06]**
   - **Targeted/Indiscriminate**: misclassification of *a specific set/any* sample



***Spear*phishing vs phishing**

Adversary's Knowledge



- **Perfect knowledge**
  - upper bound on the performance degradation under attack

Adversary's Capability

- Attack's influence **[Barreno et al. ASIACCS06]**
  - Manipulation of **training/test** data

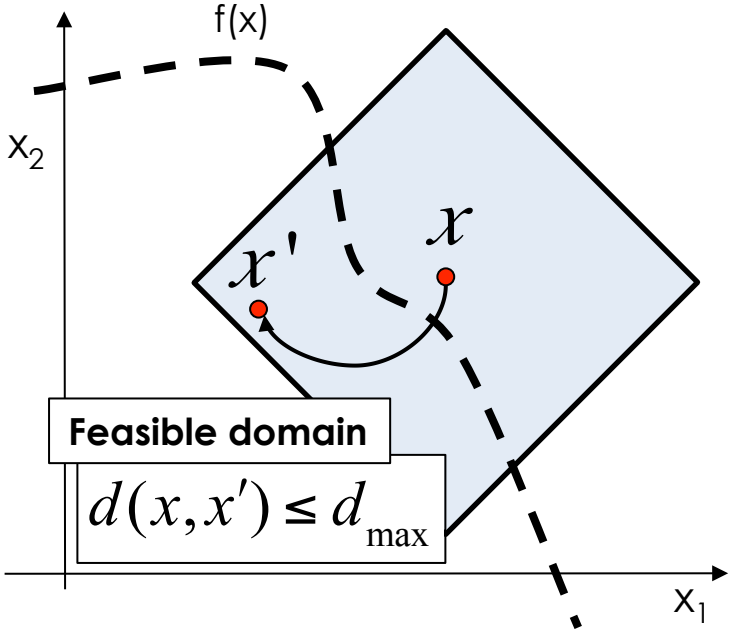
- Constraints on data manipulation

TRAINING

maximum number of samples that can be added to the training data
- the attacker usually controls only a small fraction of the training samples

TEST

maximum amount of modifications
- *application-specific constraints in feature space*
- *e.g.*, max. number of words that are modified in spam emails



Main Attack Scenarios

- *Evasion attacks*
  - **Goal**: integrity violation, indiscriminate attack
  - **Knowledge**: perfect / limited
  - **Capability**: manipulating test samples

e.g., manipulation of spam emails at test time to evade detection

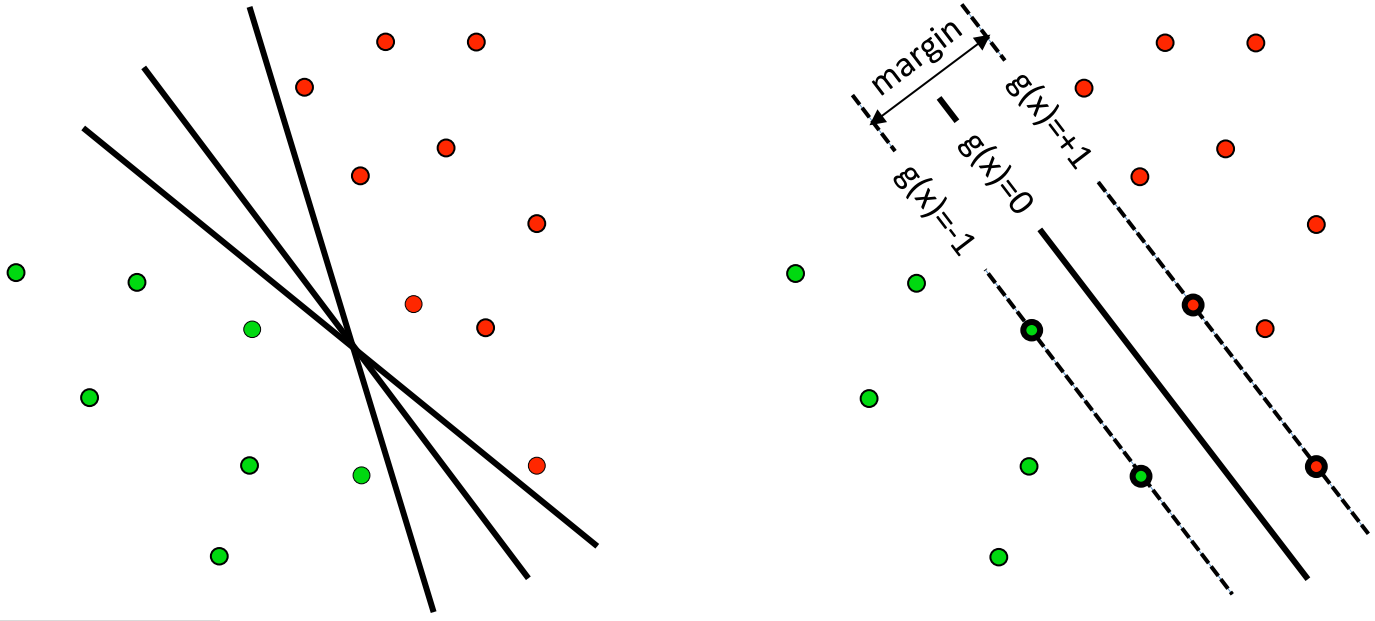
- *Poisoning attacks*
  - **Goal**: availability violation, indiscriminate attack
  - **Knowledge**: perfect / limited
  - **Capability**: injecting samples into the training data

e.g., send spam with some 'good words' to poison the anti-spam filter, which may subsequently misclassify legitimate emails containing such 'good words'

Targeted classifier: SVM

- Maximum-margin linear classifier $f(x) = \text{sign}(g(x)),\ g(x) = w^T x + b$

$$\min_{w,b} \underbrace{\frac{1}{2} w^T w}_{\text{1/margin}} + C \underbrace{\sum_i \max\left(0, 1 - y_i f(x_i)\right)}_{\text{classification error on training data (hinge loss)}}$$

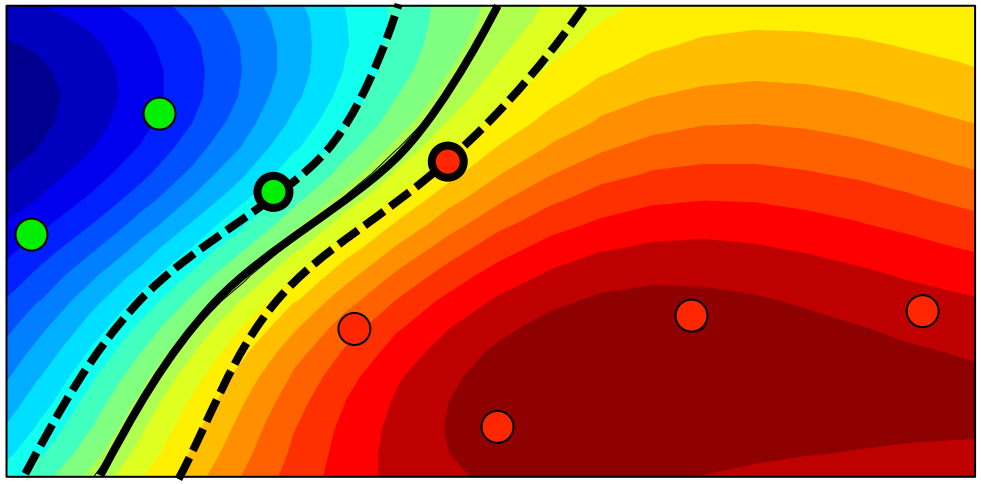


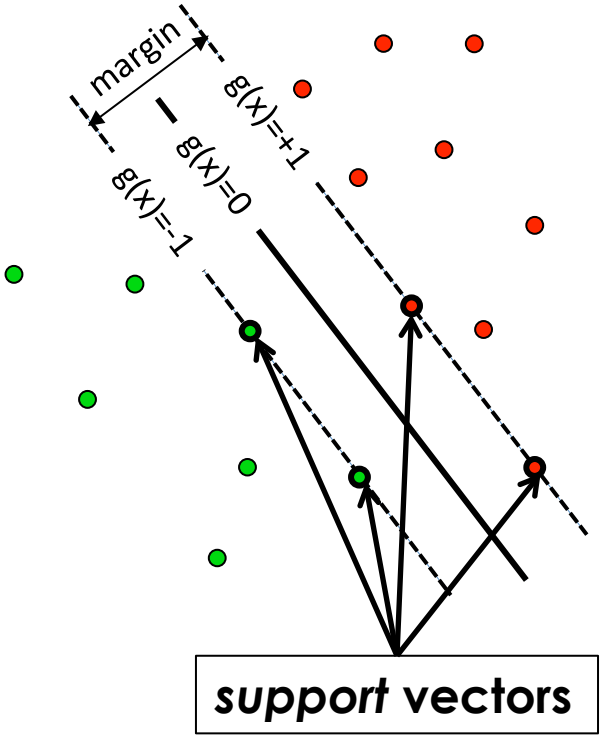
Kernels and Nonlinearity

$$w = \sum_i \alpha_i y_i x_i \rightarrow g(x) = \sum_i \alpha_i y_i \langle x, x_i \rangle + b$$

- Enables *learning* and *classification* using only dot products between samples

- Kernel functions for nonlinear classification
  - e.g., RBF Kernel $k(x, x_i) = \exp\left(-\gamma \left\| x - x_i \right\|^2\right)$



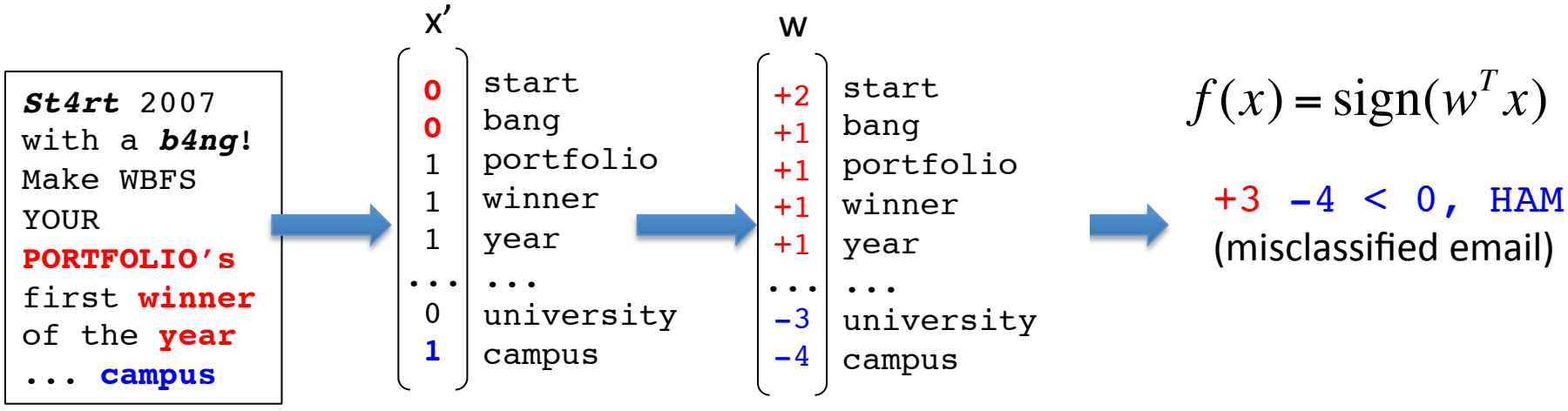


Evasion Attacks

1. B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. ECML PKDD, 2013.
2. B. Biggio et al., Security evaluation of SVMs. SVM applications. Springer, 2014

A Simple Example

- **Problem:** how to evade a linear (trained) classifier?
  - We have seen this already…



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

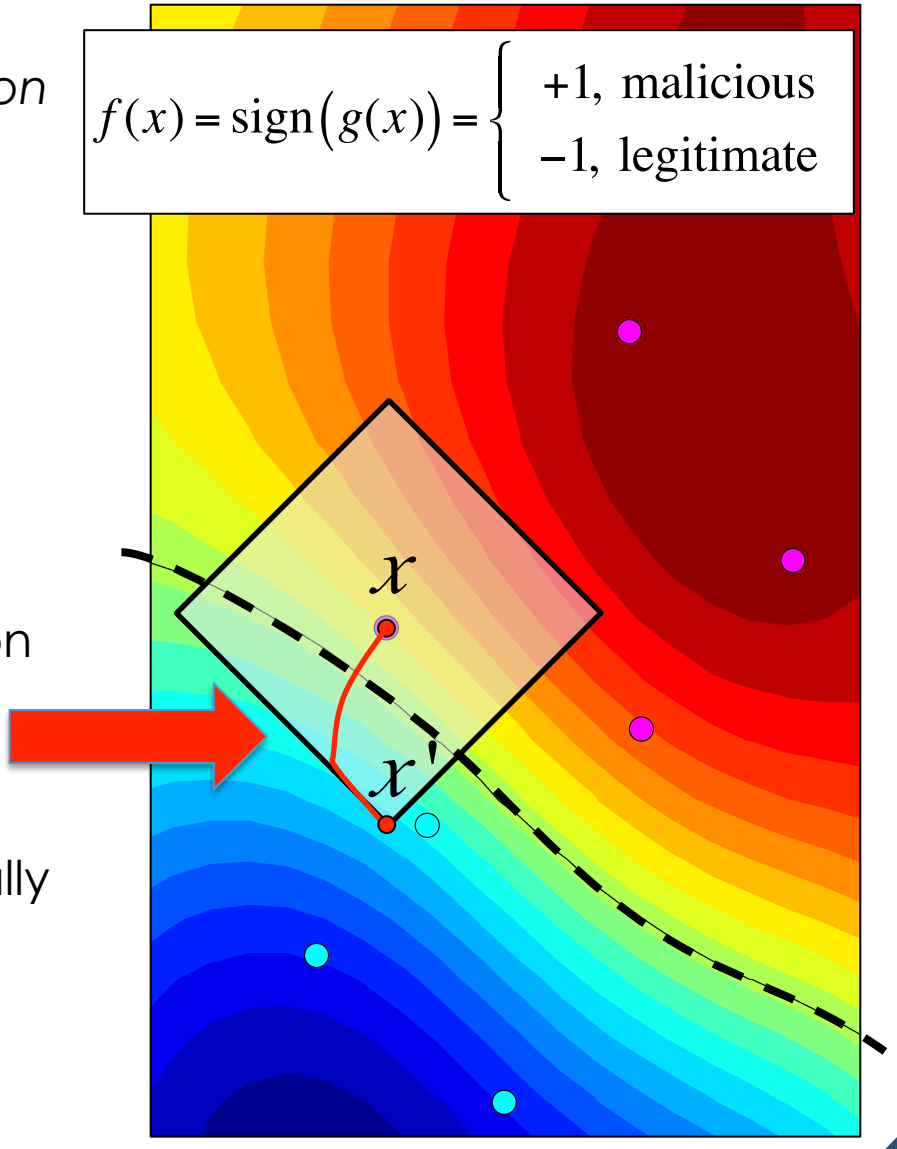
Gradient-descent Evasion Attacks

- **Goal:** maximum-confidence *evasion*
- **Knowledge:** *perfect*
- **Attack strategy:**

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

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

- Non-linear, constrained optimization
  - **Gradient descent**: approximate solution for *smooth* functions
- Gradients of g(x) can be analytically computed in many cases
  - SVMs, Neural networks



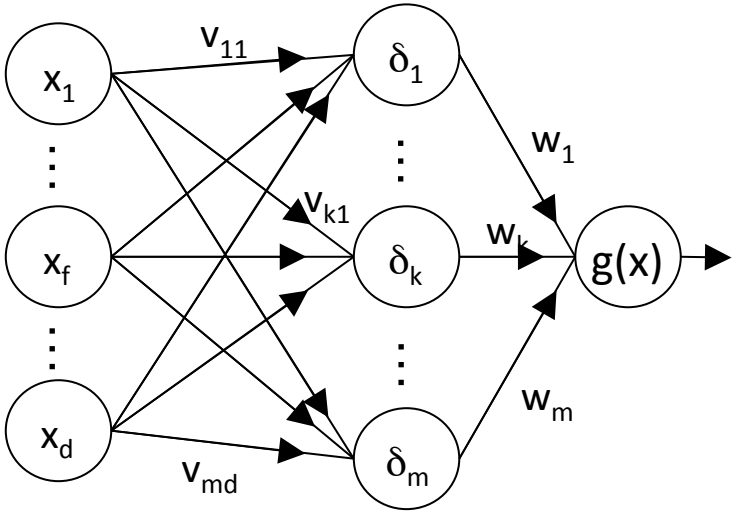
Computing Descent Directions

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

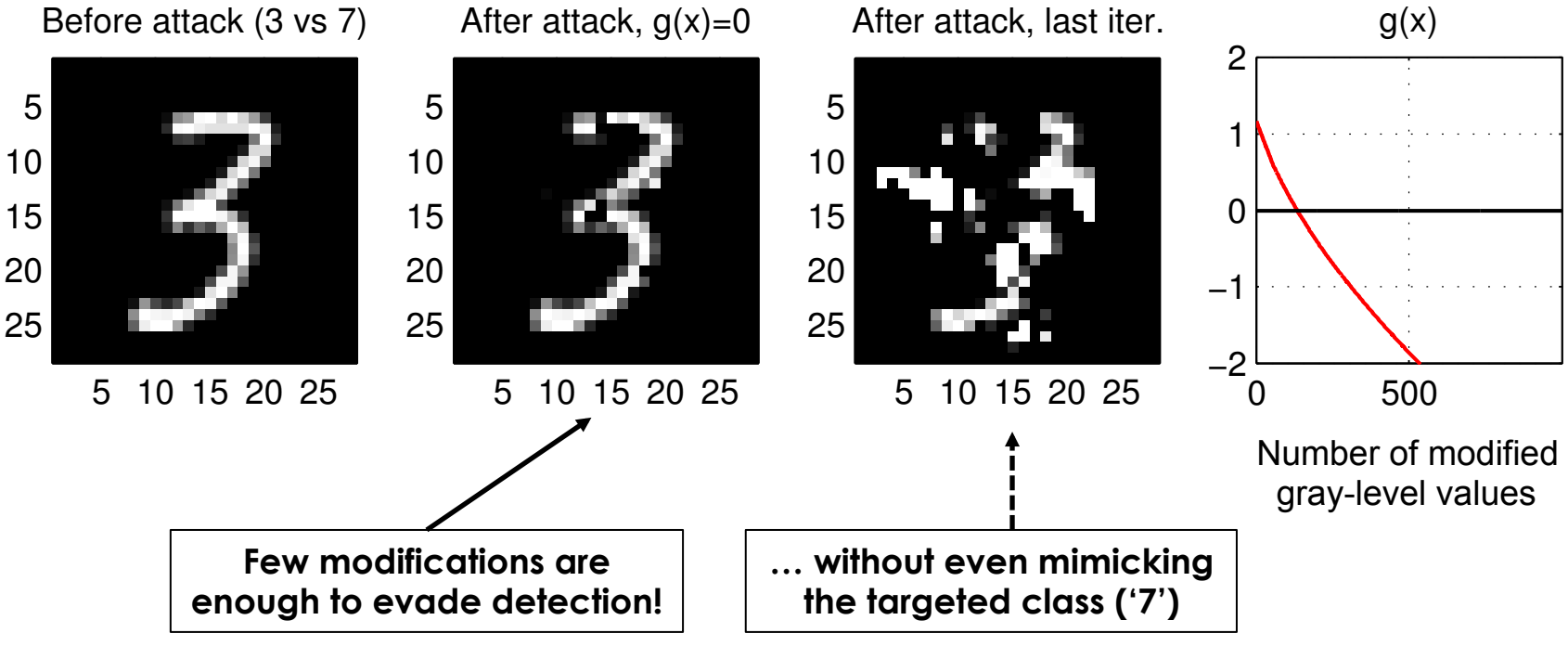


$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^{m} w_k \delta_k(x)\right)\right]^{-1}$$

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

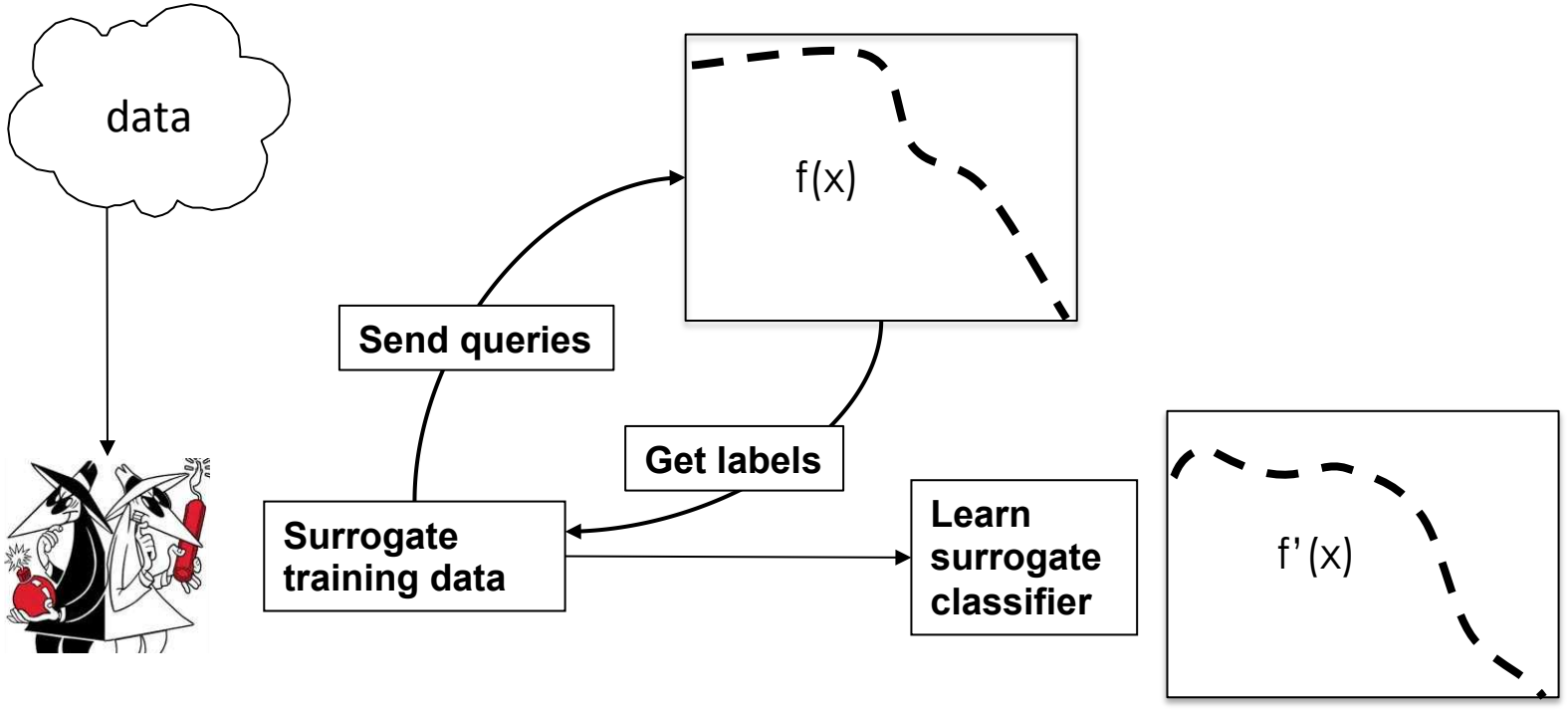
An Example on Handwritten Digits

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



Bounding the Adversary's Knowledge
Limited knowledge attacks

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



Experiments on PDF Malware Detection

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

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

Features: *keyword count*

/Type	2
/Page	1
/Encoding	1
...	

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

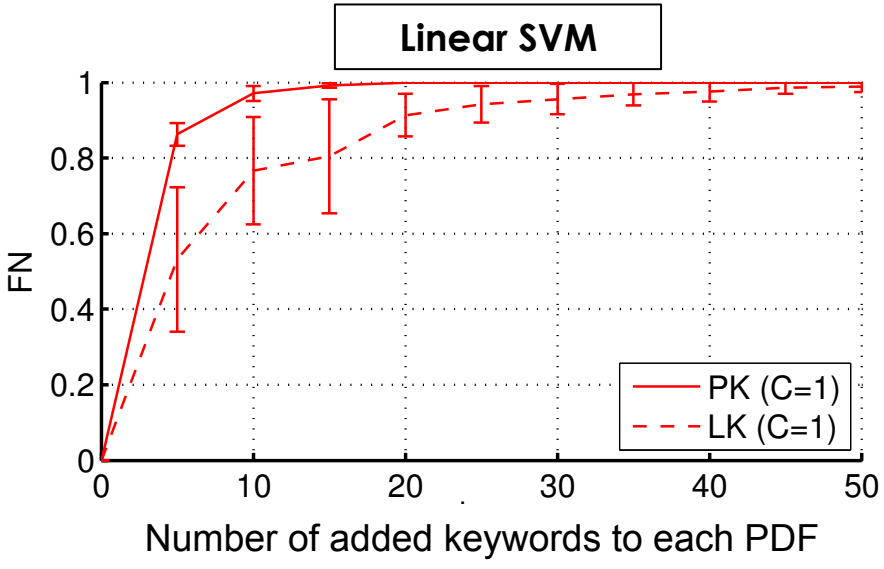
$$\min_{x'} g(x')$$

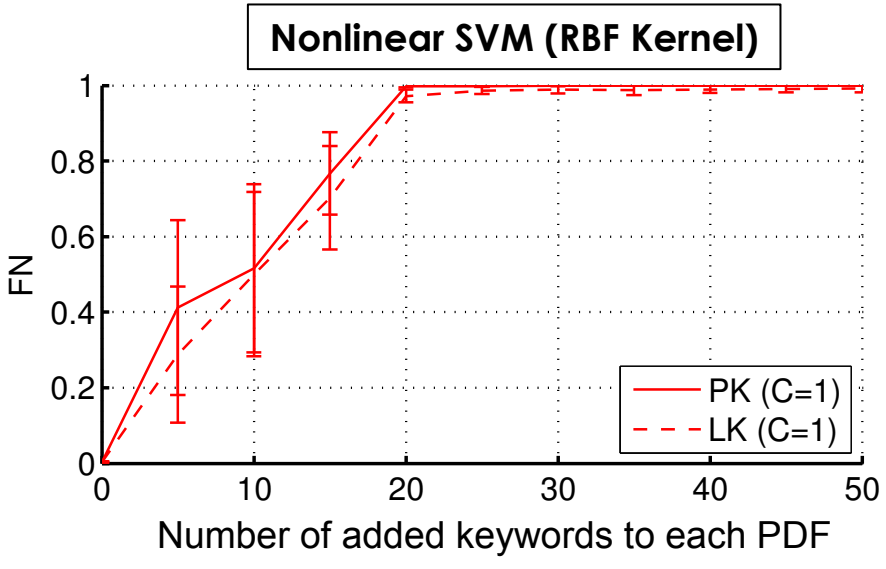
$$\text{s.t. } d(x, x') \leq d_{\max}$$

$$x \leq x'$$

Experiments on PDF Malware Detection

- **Dataset:** 500 malware samples (*Contagio*), 500 benign (Internet)
  - Targeted (surrogate) classifier trained on 500 (100) samples
- **Evasion rate** (FN) at FP=1% vs max. number of added keywords
  - Averaged on 5 repetitions
  - Perfect knowledge (PK); Limited knowledge (LK)



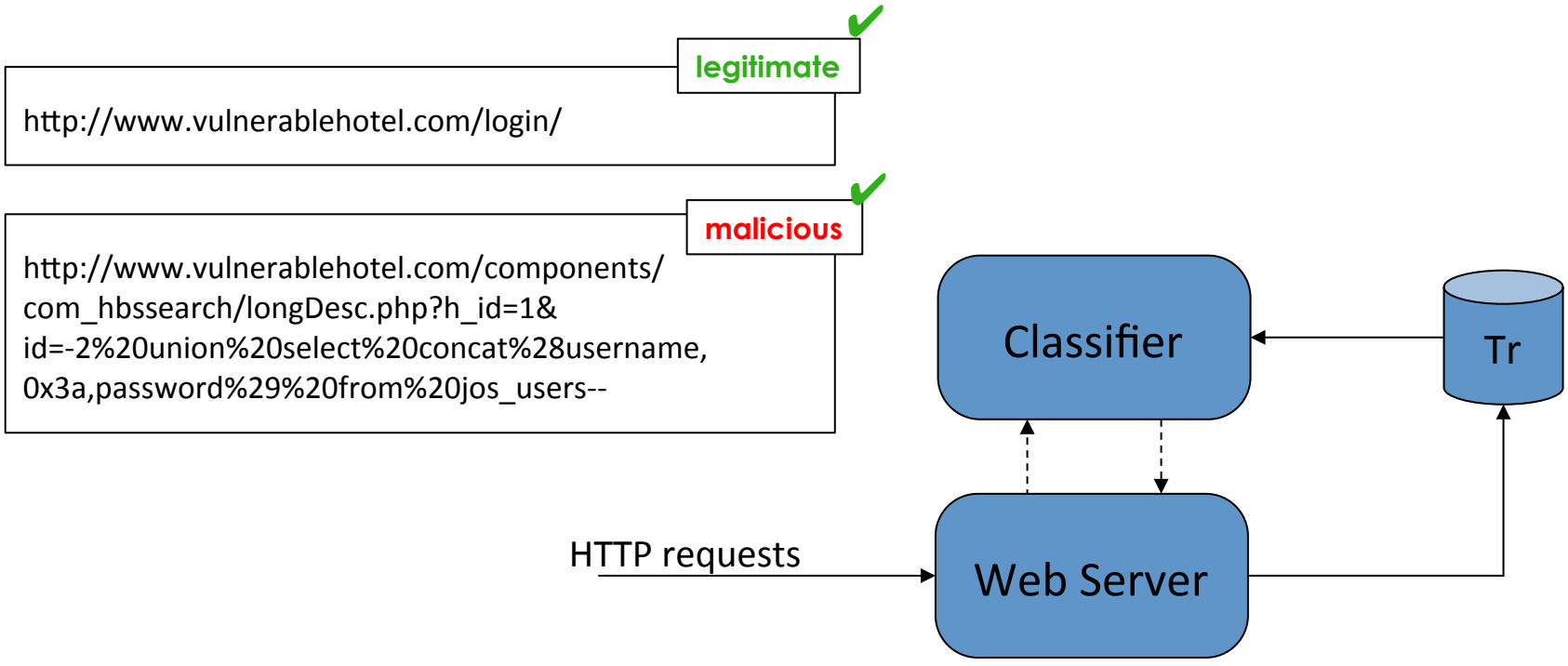


Poisoning Attacks against SVMs

1. B. Biggio, B. Nelson, P. Laskov. Poisoning attacks against SVMs. ICML, 2012
2. B. Biggio et al., Security evaluation of SVMs. SVM applications. Springer, 2014

Poisoning Attacks against SVMs

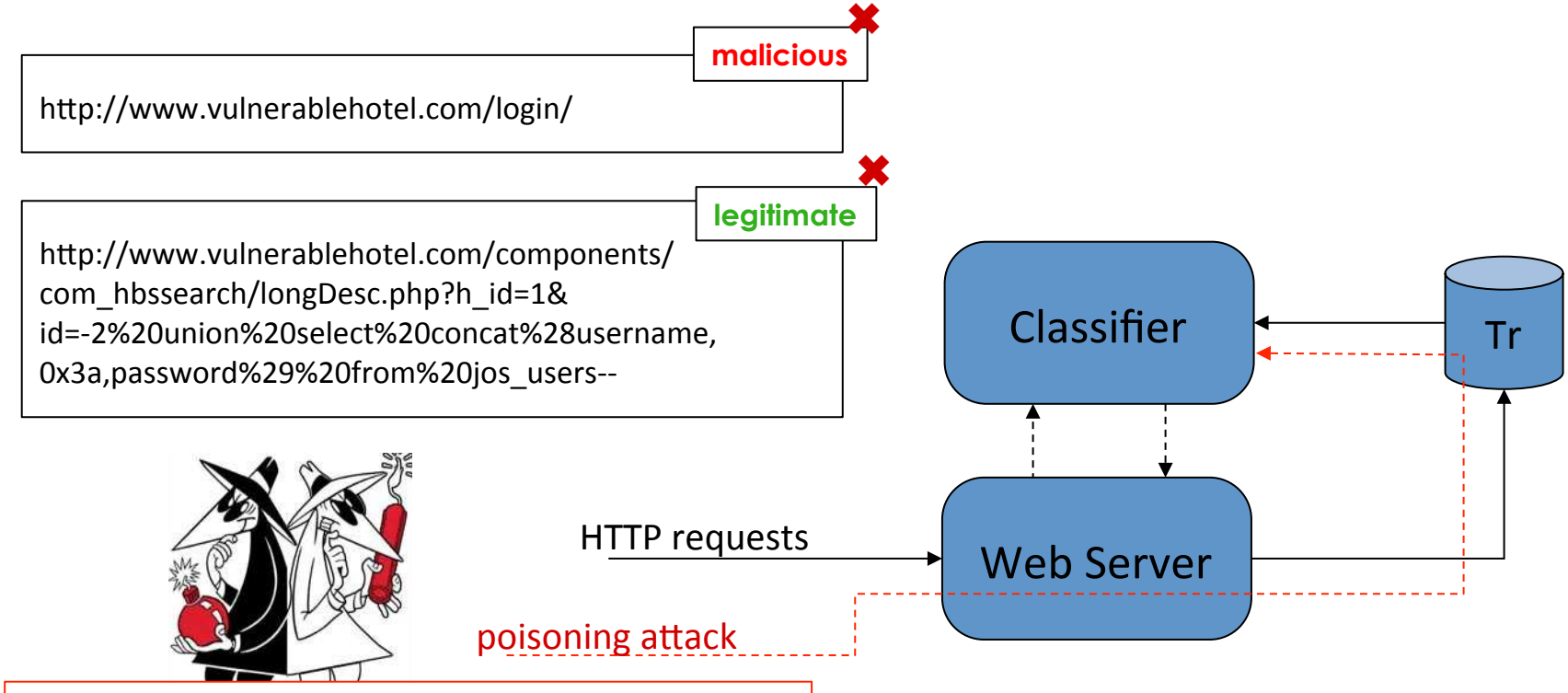
[B. Biggio, B. Nelson, P. Laskov, ICML 2012]



Poisoning Attacks against SVMs

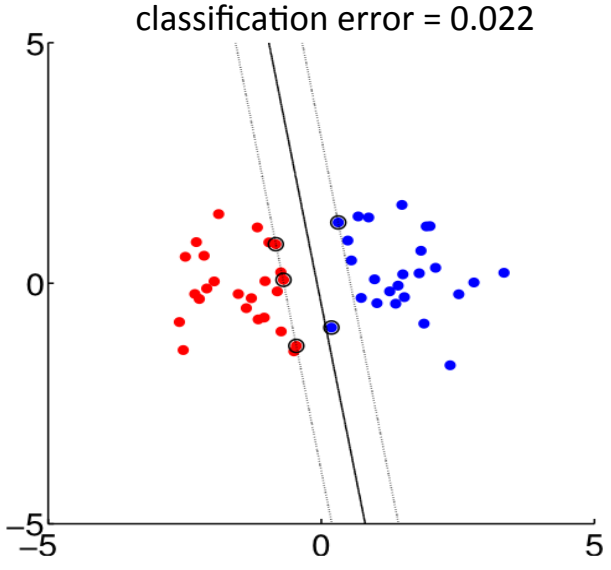
[B. Biggio, B. Nelson, P. Laskov, ICML 2012]

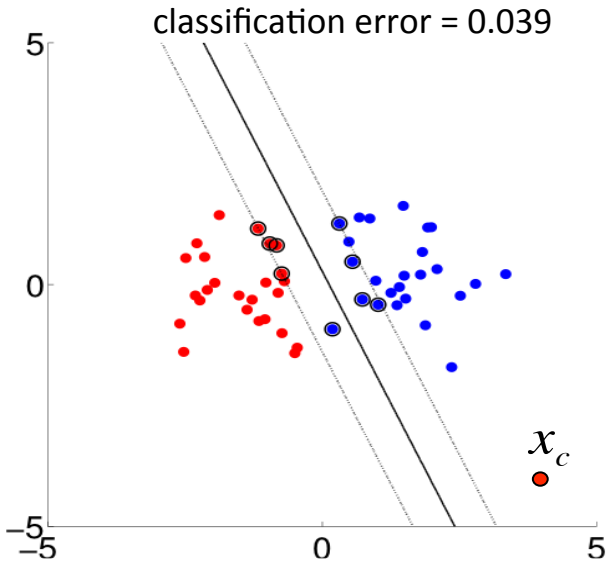
- Poisoning classifier to cause denial of service



Adversary Model and Attack Strategy

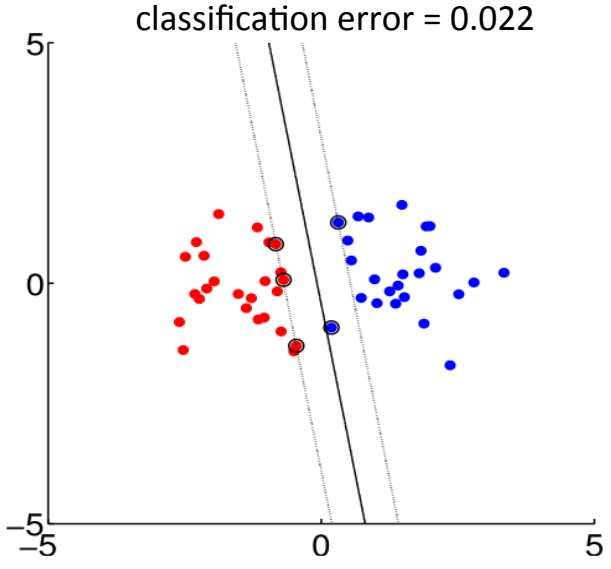
- Adversary model
  - **Goal**: to maximize classification error (availiability, indiscriminate)
  - **Knowledge**: perfect knowledge (trained **SVM** and TR set are known)
  - **Capability**: injecting samples into TR
- Attack strategy
  - optimal attack point x_c in TR that maximizes classification error

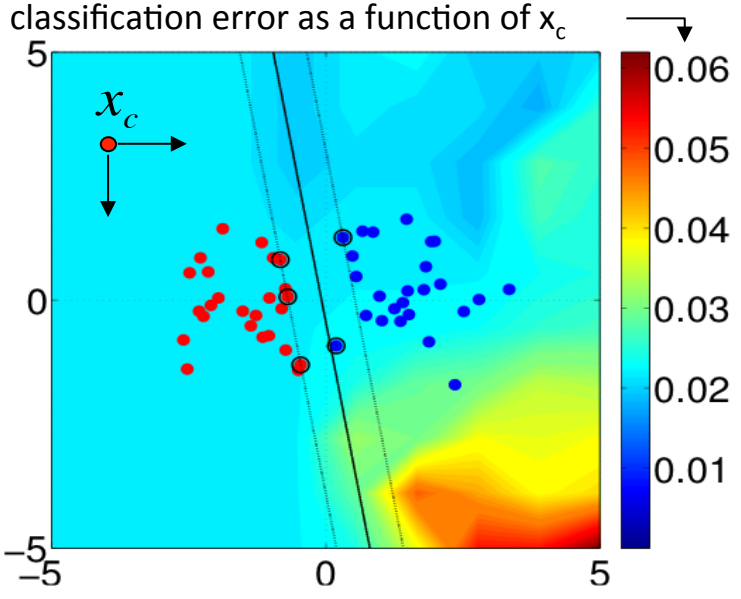




Adversary Model and Attack Strategy

- Adversary model
  - **Goal**: to maximize classification error (availiability, indiscriminate)
  - **Knowledge**: perfect knowledge (trained **SVM** and TR set are known)
  - **Capability**: injecting samples into TR
- Attack strategy
  - optimal attack point x_c in TR that maximizes classification error



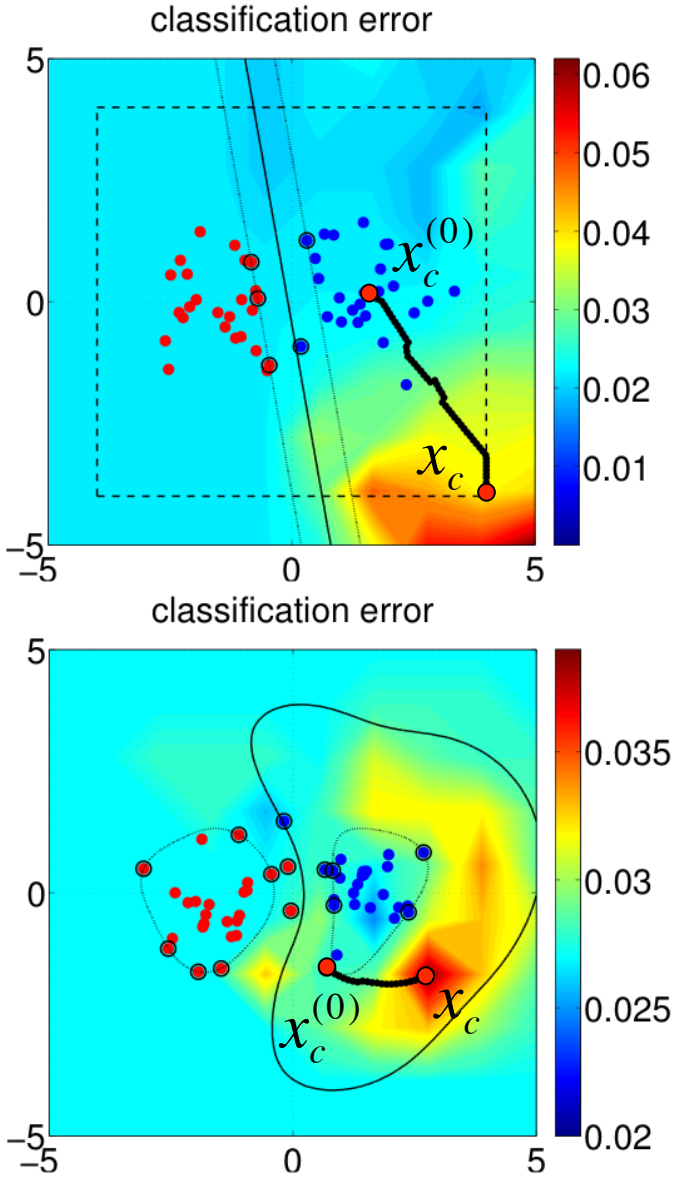


Poisoning Attack Algorithm

- Max. classification error $L(x_c)$ w.r.t. x_c through gradient ascent
- Gradient is not easy to compute
  - The training point affects the classification function
  - Details of the derivation are in the paper

Input: $\mathcal{D}_{\text{tr}}$, $\mathcal{D}_{\text{val}}$, y_c, $x_c^{(0)}$, t.
Output: x_c.

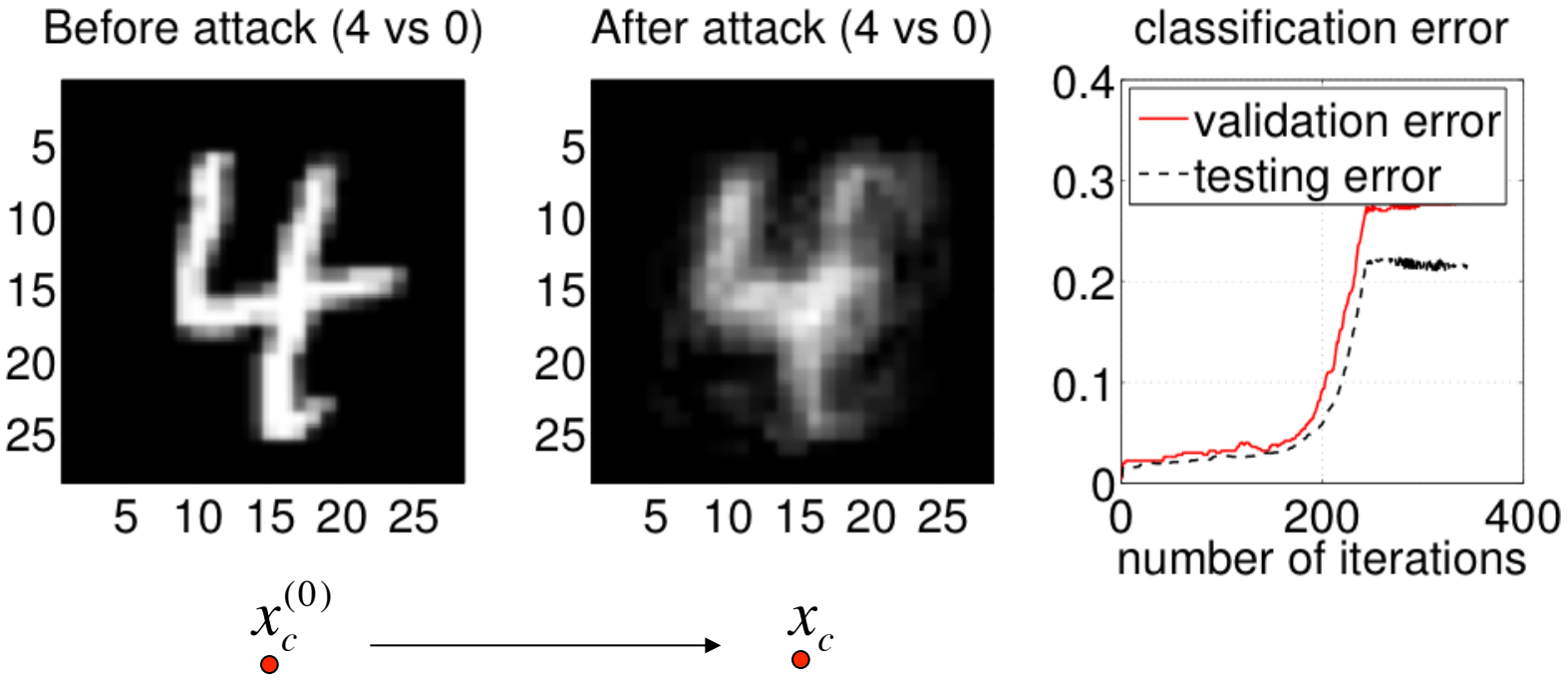
1: $\{\alpha_i, b\} \leftarrow$ learn an SVM on $\mathcal{D}_{\text{tr}}$.
2: $x_c \leftarrow x_c^{(0)}$
3: **repeat**
4: Update $\{\alpha_i, b\}$ on $\mathcal{D}_{\text{tr}} \cup \{x_c, y_c\}$
5: Compute $\nabla L(x_c)$ on $\mathcal{D}_{\text{val}}$.
6: $x_c' \leftarrow x_c + t\nabla L(x_c)$
7: **until** $L(x_c') - L(x_c) < \epsilon$
8: **return** x_c



1. B. Biggio, B. Nelson, P. Laskov. Poisoning attacks against SVMs. ICML, 2012

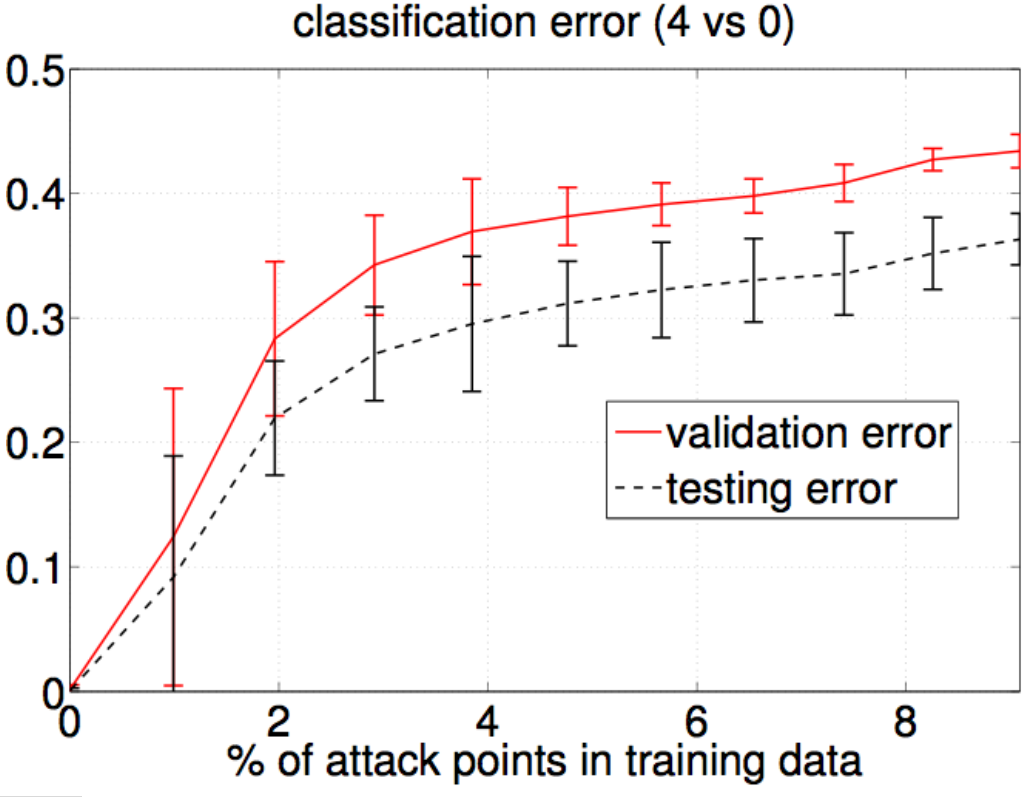
Experiments on the MNIST digits
Single-point attack

- Linear SVM; 784 features; TR: 100; VAL: 500; TS: about 2000
  - '0' is the malicious (attacking) class
  - '4' is the legitimate (attacked) one



Experiments on the MNIST digits
Multiple-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



Attacking Clustering

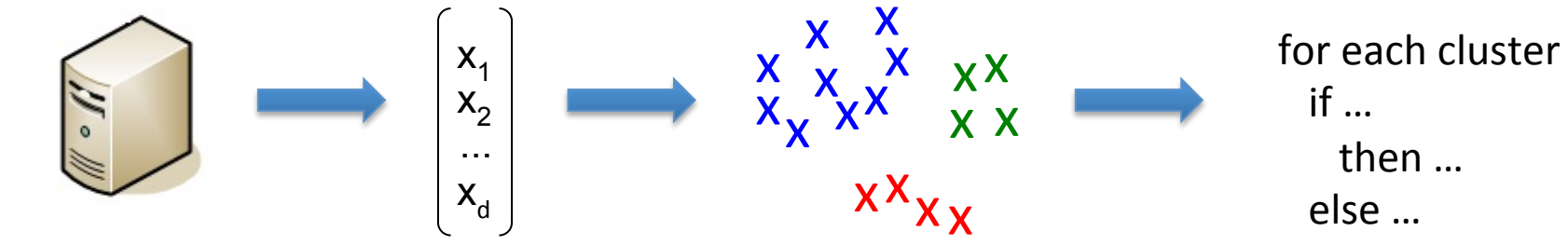
1. B. Biggio, I. Pillai, S. R. Bulò, D. Ariu, M. Pelillo, and F. Roli. Is data clustering in adversarial settings secure? AISec, 2013
2. B. Biggio, S. R. Bulò, I. Pillai, M. Mura, E. Z. Mequanint, M. Pelillo, and F. Roli. Poisoning complete-linkage hierarchical clustering. S+SSPR, 2014

Attacking Clustering

- So far, we have considered **supervised learning**
  - Training data consisting of *samples* and *class labels*

- In many applications, labels are not available or costly to obtain
  - **Unsupervised learning**
    - Training data only include samples – no labels!

- **Malware clustering**
  - To identify variants of existing malware or new malware families



data collection (*honeypots*)

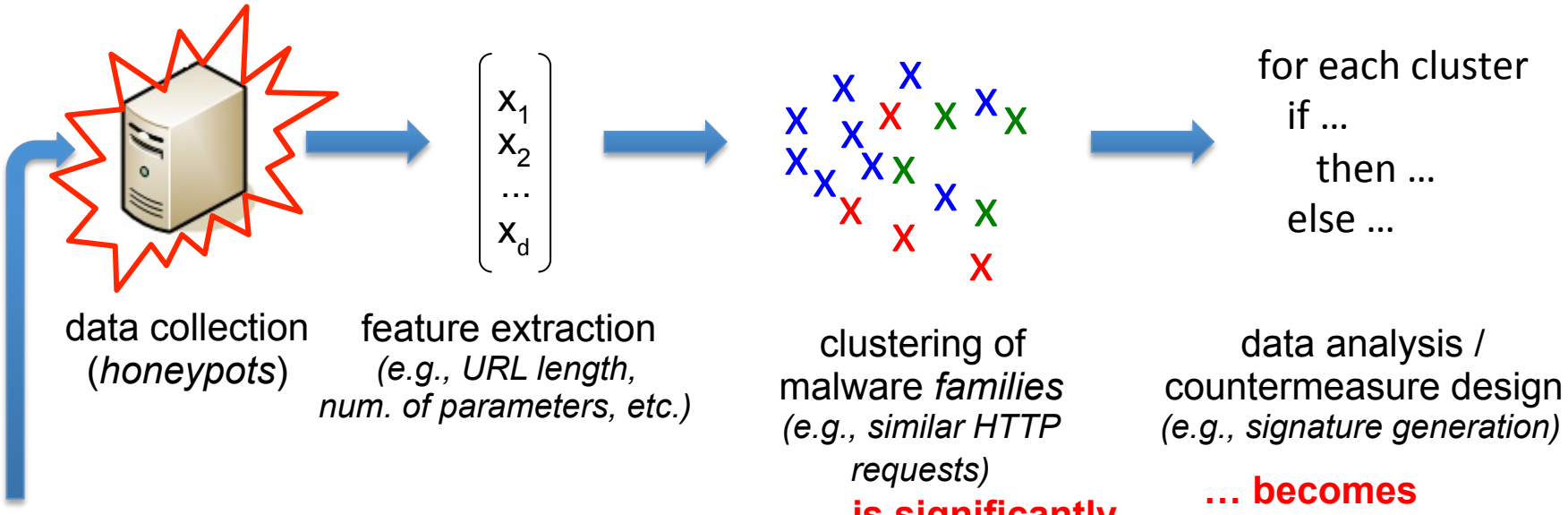
feature extraction *(e.g., URL length, num. of parameters, etc.)*

clustering of malware *families* *(e.g., similar HTTP requests)*

data analysis / countermeasure design *(e.g., signature generation)*

Is Data Clustering *Secure*?

- **Attackers** can ***poison*** input data to subvert malware clustering



Well-crafted HTTP requests to subvert clustering

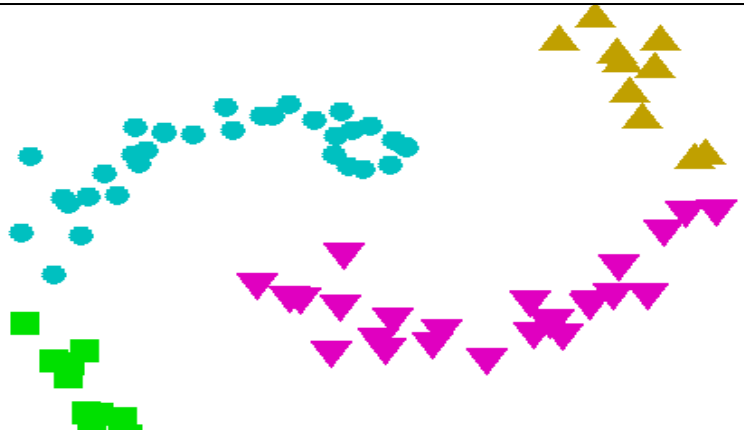
http://www.vulnerablehotel.com/…
http://www.vulnerablehotel.com/…
http://www.vulnerablehotel.com/…
http://www.vulnerablehotel.com/…

Our Work

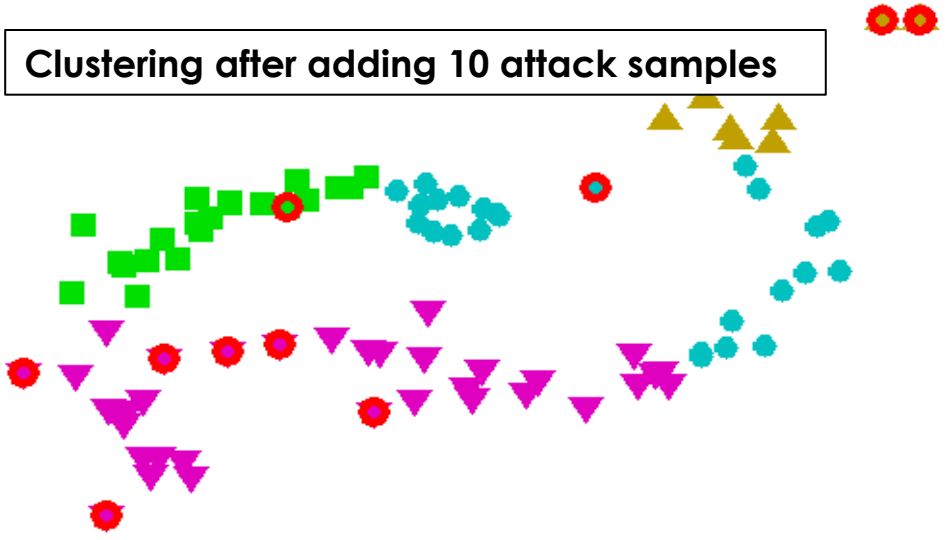
- A framework to identify/design attacks against clustering algorithms
  - **Poisoning**: add samples to maximally compromise the clustering output
  - **Obfuscation**: hide samples within existing clusters

- Some clustering algorithms can be very sensitive to poisoning!
  - single- and complete-linkage hierarchical clustering can be easily compromised by creating *heterogeneous* clusters
    - Details on the attack derivation and implementation are in the papers

Clustering on untainted data (80 samples)



Clustering after adding 10 attack samples

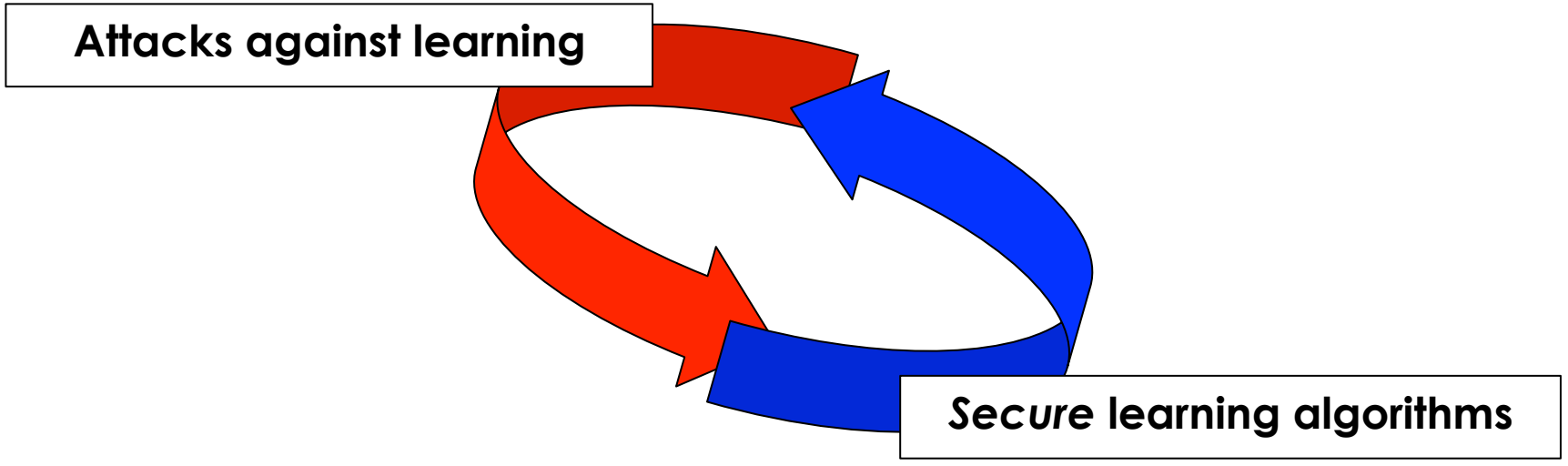


1. B. Biggio et al. Is data clustering in adversarial settings secure? AISec, 2013
2. B. Biggio et al. Poisoning complete-linkage hierarchical clustering. S+SSPR, 2014

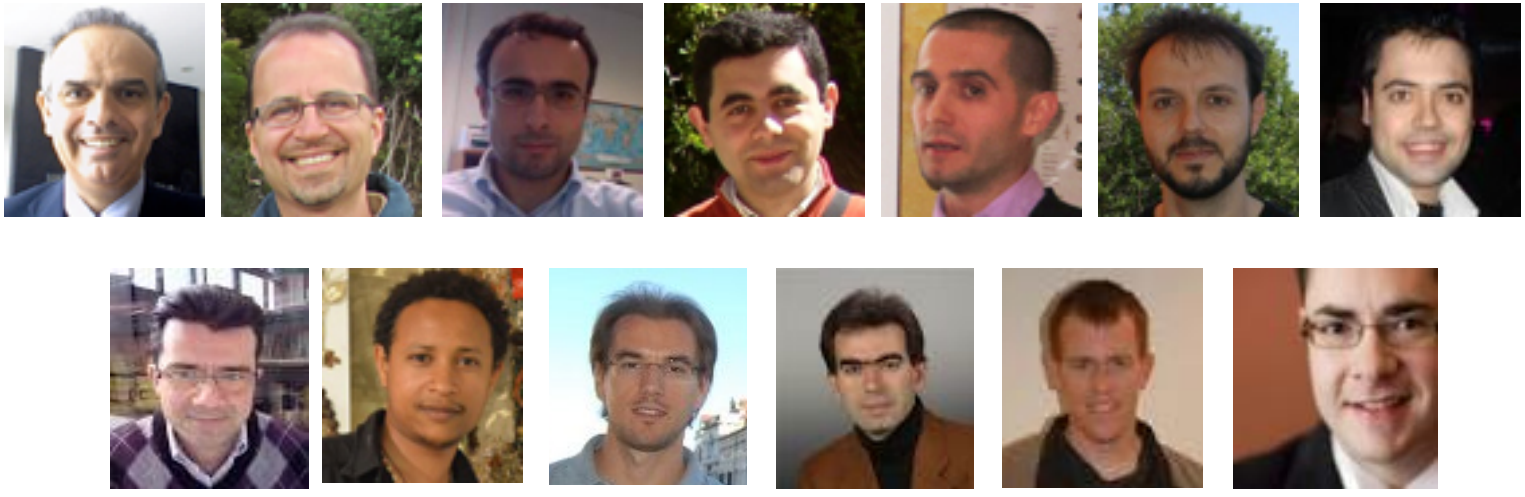
Conclusions and Future Work

- Learning-based systems can be vulnerable to well-crafted, sophisticated attacks devised by skilled attackers
  - *… that exploit specific vulnerabilities of machine learning algorithms!*

- **Future (and ongoing) work**
  - **Privacy Attacks**
  - **Secure Learning, Clustering and Feature Selection/Reduction**



Joint work with …



… and many others

Thanks for your attention!

Any questions

References

1. B. Biggio, G. Fumera, and F. Roli. *Security evaluation of pattern classifiers under attack*. IEEE Trans. on Knowl. and Data Eng., 26(4):984–996, April 2014.
2. M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. *Can machine learning be secure?* In Proc. ACM Symp. Information, Computer and Comm. Sec., ASIACCS '06, pages 16–25, New York, NY, USA, 2006. ACM.
3. M. Barreno, B. Nelson, A. Joseph, and J. Tygar. *The security of machine learning*. Machine Learning, 81:121–148, 2010.
4. B. Biggio, I. Corona, B. Nelson, B. Rubinstein, D. Maiorca, G. Fumera, G. Giacinto, and F. Roli. *Security evaluation of support vector machines in adversarial environments*. In Y. Ma and G. Guo, eds, Support Vector Machines Applications, pages 105–153. Springer International Publishing, 2014.
5. B. Biggio, G. Fumera, and F. Roli. *Pattern recognition systems under attack: Design issues and research challenges*. Int'l J. Patt. Recogn. Artif. Intell., 2014, In press.
6. B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli. *Evasion attacks against machine learning at test time*. In H. Blockeel et al., editors, European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Part III, volume 8190 of LNCS, pp. 387–402. Springer, 2013.
7. B. Biggio, B. Nelson, and P. Laskov. *Poisoning attacks against support vector machines*. In J. Langford and J. Pineau, eds, 29th Int'l Conf. on Machine Learning, pp.1807–1814. Omnipress, 2012.
8. B. Biggio, I. Pillai, S. R. Bulò, D. Ariu, M. Pelillo, and F. Roli. *Is data clustering in adversarial settings secure?* In Proc. 2013 ACM Workshop on Artificial Intell. and Security, AISec '13, pages 87–98, New York, NY, USA, 2013. ACM.
9. B. Biggio, S. R. Bulò, I. Pillai, M. Mura, E. Z. Mequanint, M. Pelillo, and F. Roli. *Poisoning complete-linkage hierarchical clustering*. In P. Franti, G. Brown, M. Loog, F. Escolano, and M. Pelillo, editors, Joint IAPR Int'l Workshop on Structural, Syntactic, and Statistical Patt. Recogn., volume 8621 of LNCS, pages 42–52, Joensuu, Finland, 2014. Springer Berlin Heidelberg.