Selected Topics for IT Security

Adversarial Learning: AI as a New Security Target

Lecturer

Huang Xiao

xiaohu (at) in (dot) tum (dot) de

Chair for IT Security (I20), Prof. Dr. Claudia Eckert

Technische Universität München

Outline

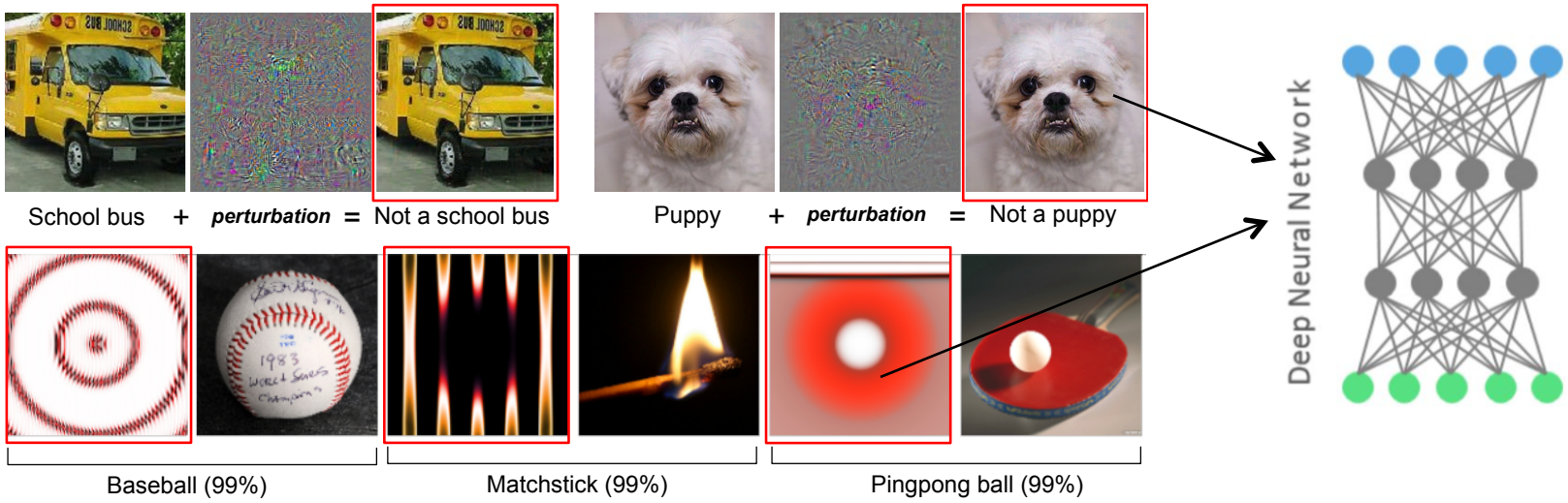
- **Introduction & Motivation**
- Insight into Learning Models
- Adversarial Learning Theory
- Adversarial Attacks
- Towards Secure Learning
- Conclusions
- References

Introduction

- ✓ Machine Learning is not a panacea
- ✓ Data stationary assumption is usually too strict
- ✓ Learning-as−a−Service is a trend
- ✓ Cyber Security is using learning-based solutions
- ✓ AI system itself can be a threat (intrinsic algorithmic deficiency)
- ✓ Endless war between adversaries and defenders
- ✓ Therefore: adversarial and secure machine learning

Motivation

To mislead the DNN to give wrong decisions

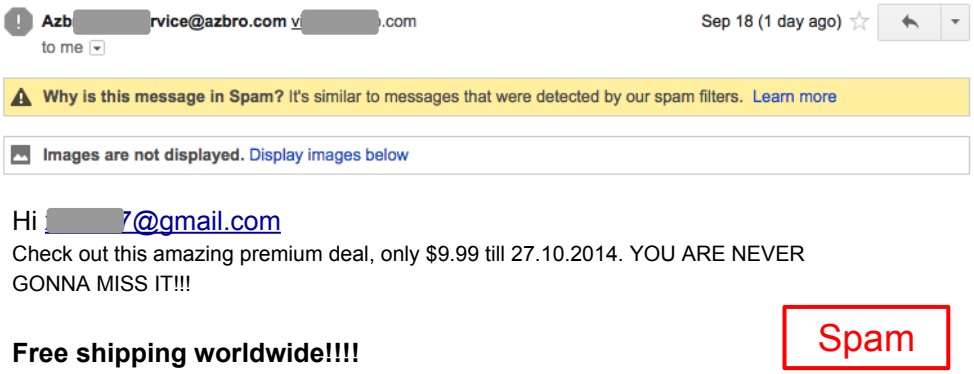


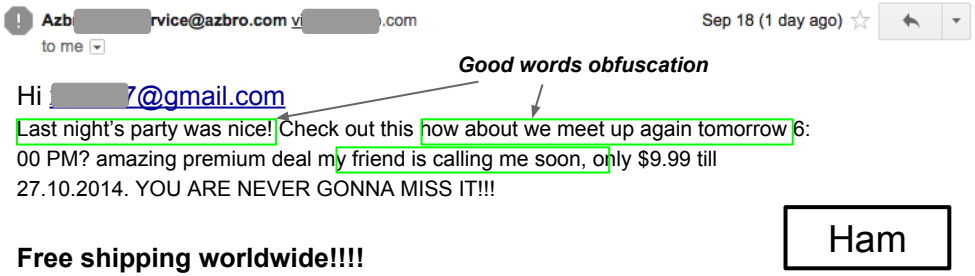
"It turns out some DNNs only focus on discriminative features in images."

Nguyen A, Yosinski J, Clune J. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In Computer Vision and Pattern Recognition (CVPR '15), IEEE, 2015.

Motivation

To bypass the spam-filter by good words attack.



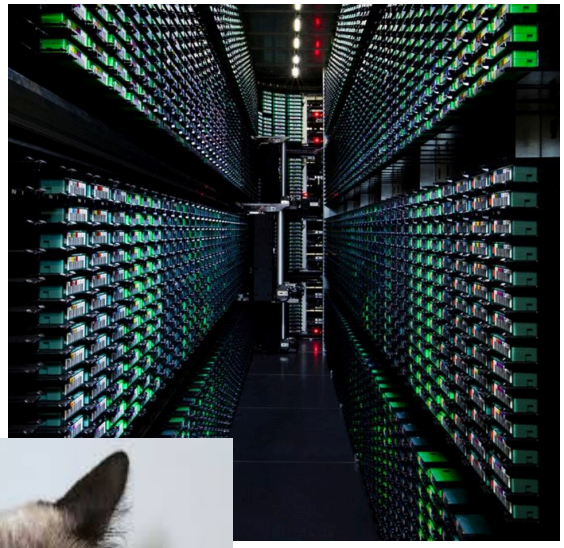


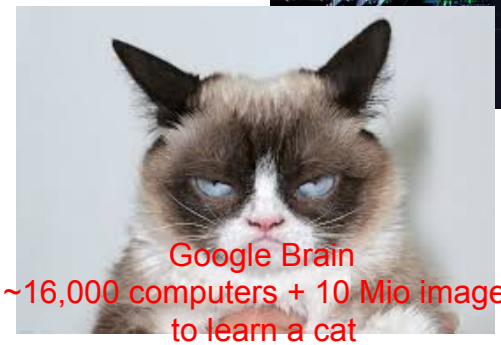
Learning to violate security goals, e.g.,
Integrity is violated, since a positive sample evades being detected.

Motivation

Can you afford technology like Deep learning?

- ✓ Demanding computational resources
- ✓ Huge amount of training data
- ✓ Data scientists are wanted
- ✓ Learning-as-a-service, e.g., Google predictive API, MS Azure, IBM Watson, AWS ML. etc…
- ✓ Adversaries know the whom to attack now..





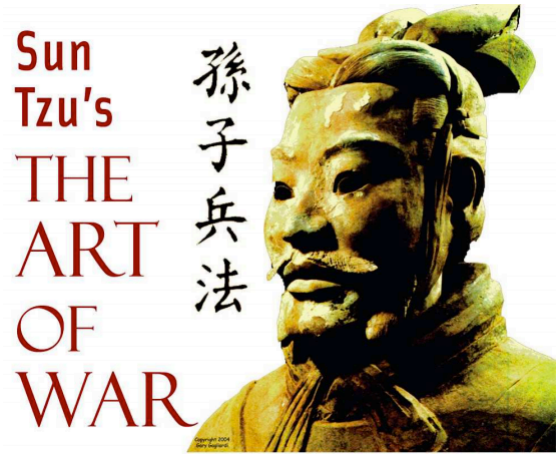
Google Brain
~16,000 computers + 10 Mio images
to learn a cat

Definition

Adversarial Learning is the reverse engineering of machine learning. It aims to design robust and secure learning algorithms.

Related areas:

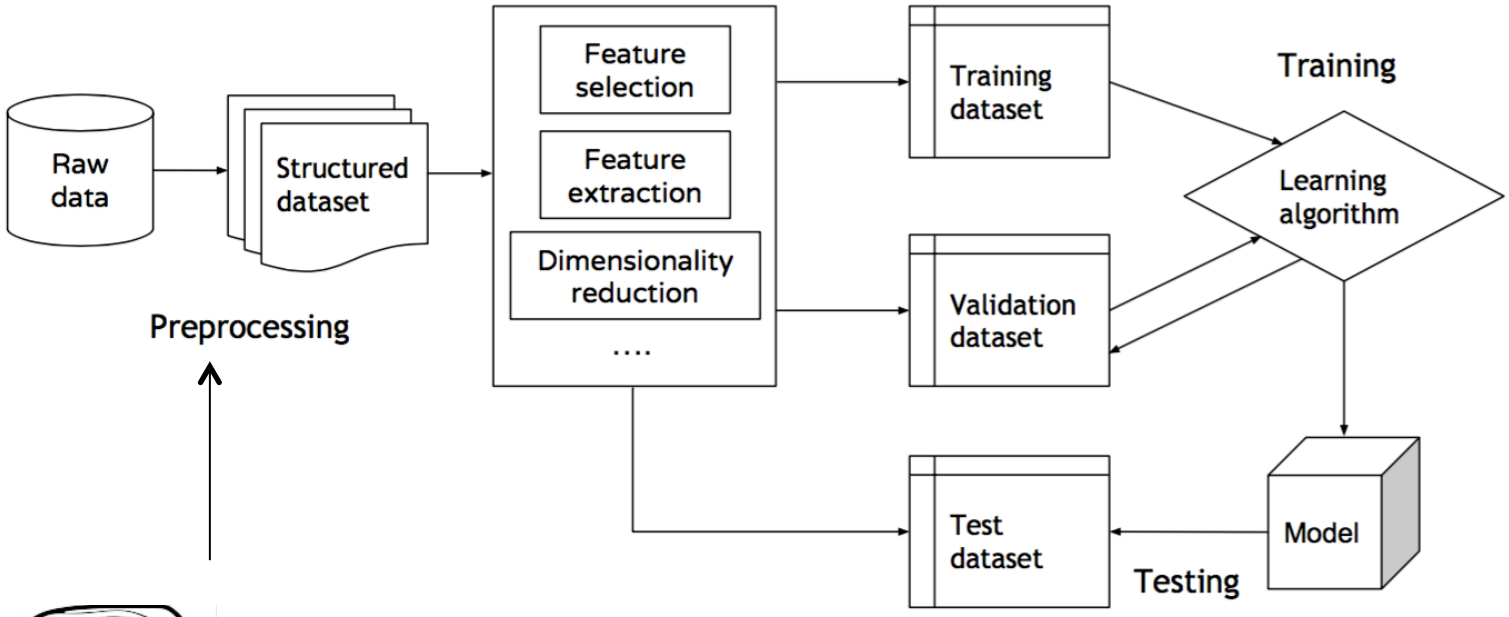
- ✓ Learning theory
- ✓ Game theory
- ✓ IT security
- ✓ Robust statistics
- ✓ Differential privacy



Know your enemies and yourself, you will not be imperiled in a hundred battles.

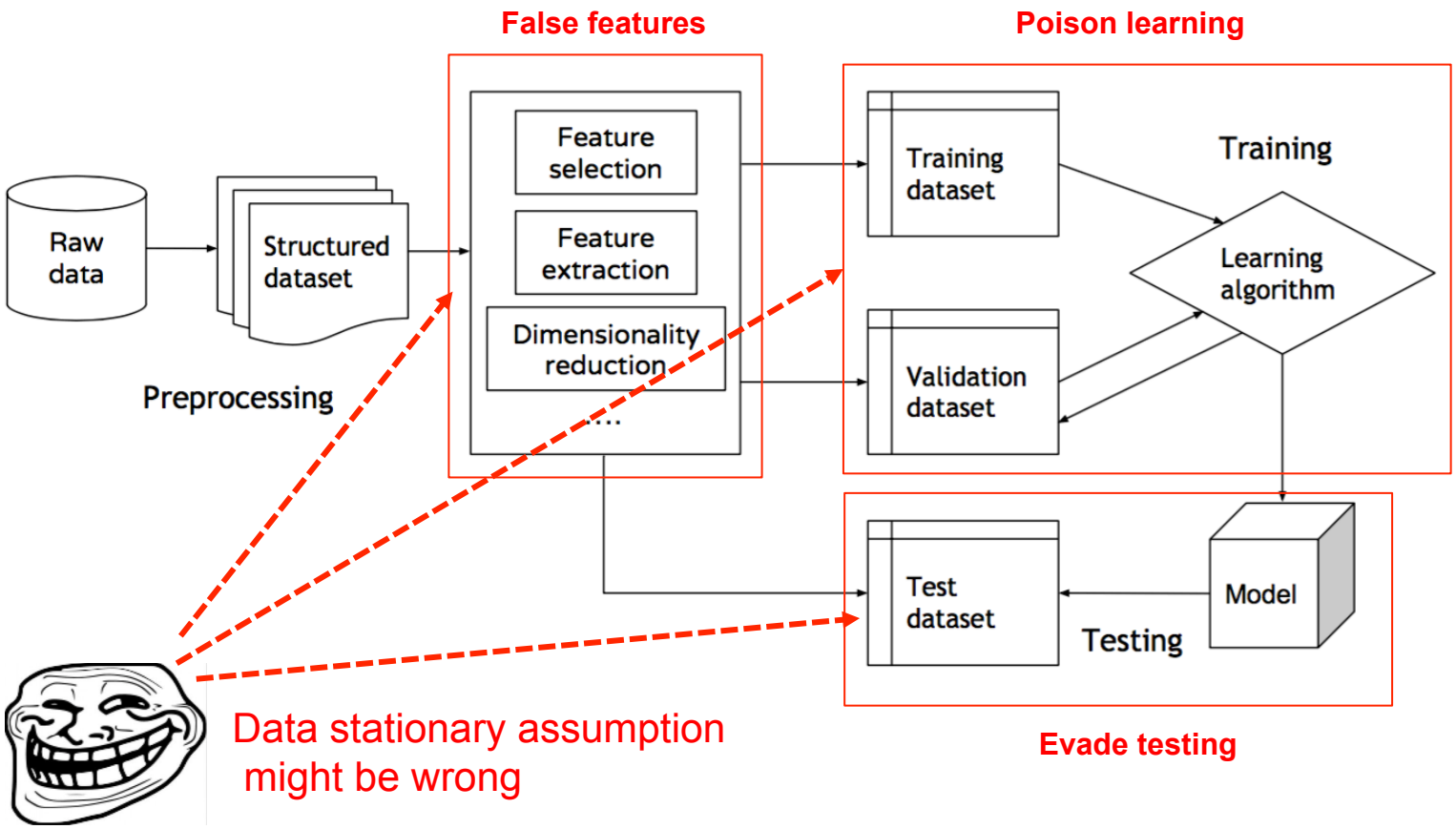
–Sun Tzu, The Art of War, 544 BC

Learning under Adversarial Impact



Data stationary assumption might be wrong

Learning under Adversarial Impact

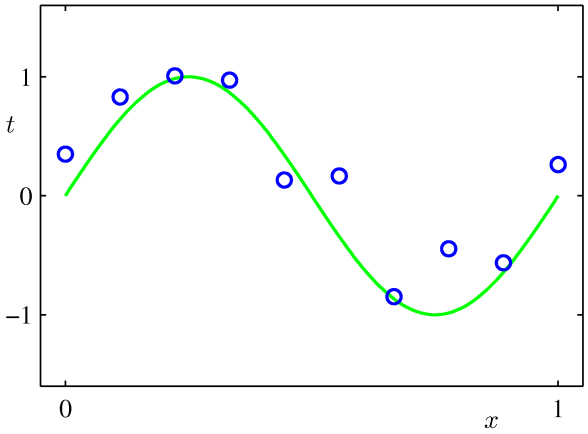


Outline

- ❑ Introduction & Motivation
- ❑ **Insight into Learning Models**
- ❑ Adversarial Learning Theory
- ❑ Adversarial Attacks
- ❑ Towards Secure Learning
- ❑ Conclusions
- ❑ References

Inside the learning

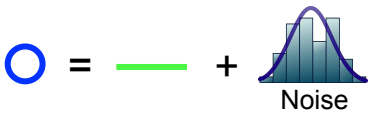
A polynomial fitting example, use a M-polynomial to fit the training data..



True signal $\sin(2\pi x)$

w Linear parameters we want to learn

Observations $x_{i=1}^{n}$



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

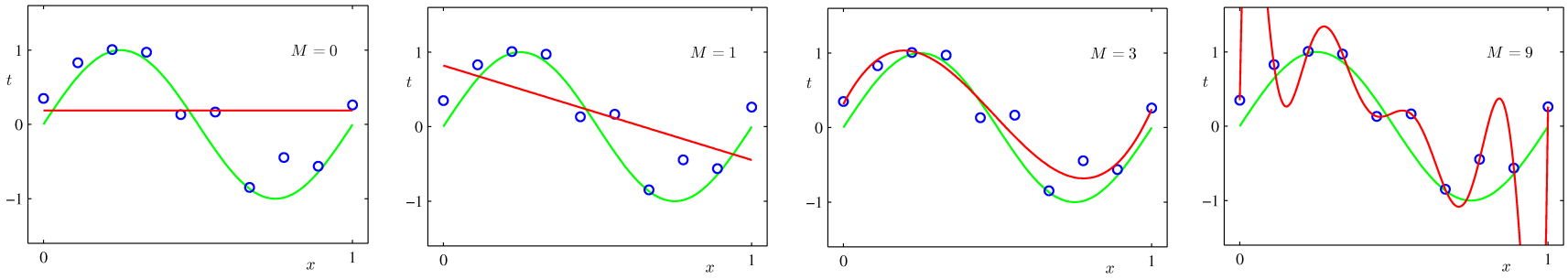
Q: how to choose M?

Inside the learning (cont.)

Training this linear model: e.g., minimize least square

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{f(x_n, w) - y_n\}^2 + M$$

Estimated output **True target** **Function complexity**



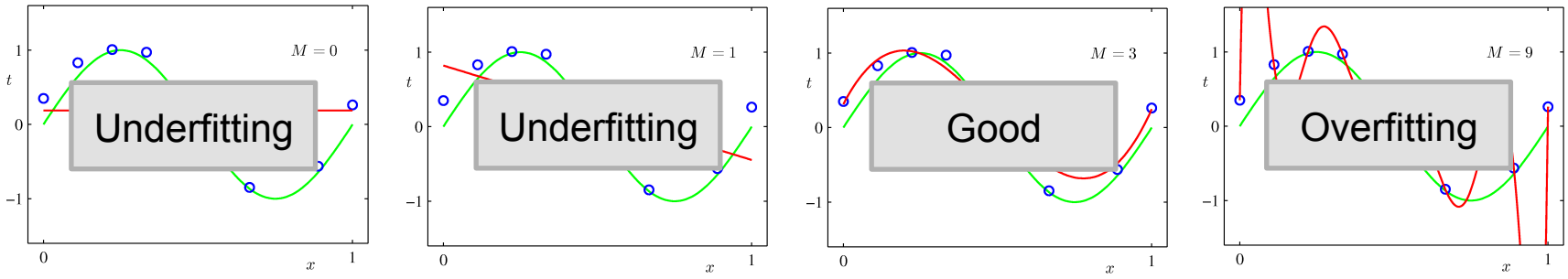
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Model Selection

Training this linear model: e.g., minimize least square

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{f(x_n, w) - y_n\}^2 + M$$

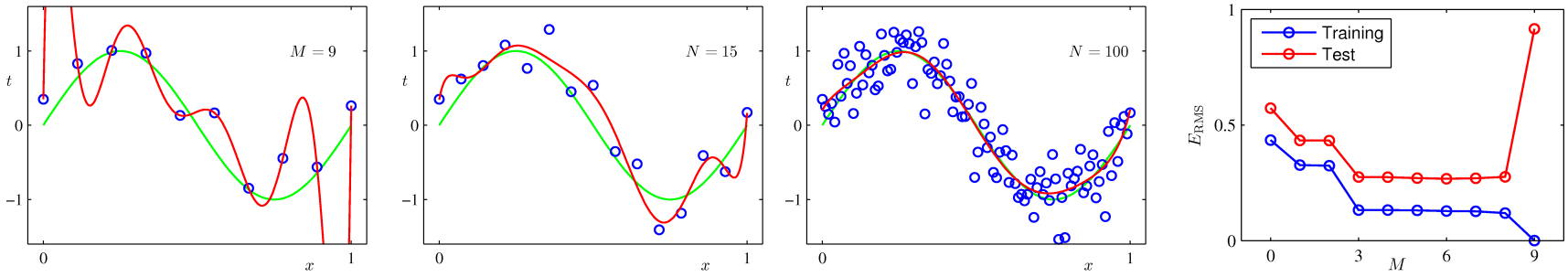
Estimated output **True target** **Function complexity**



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Overfitting

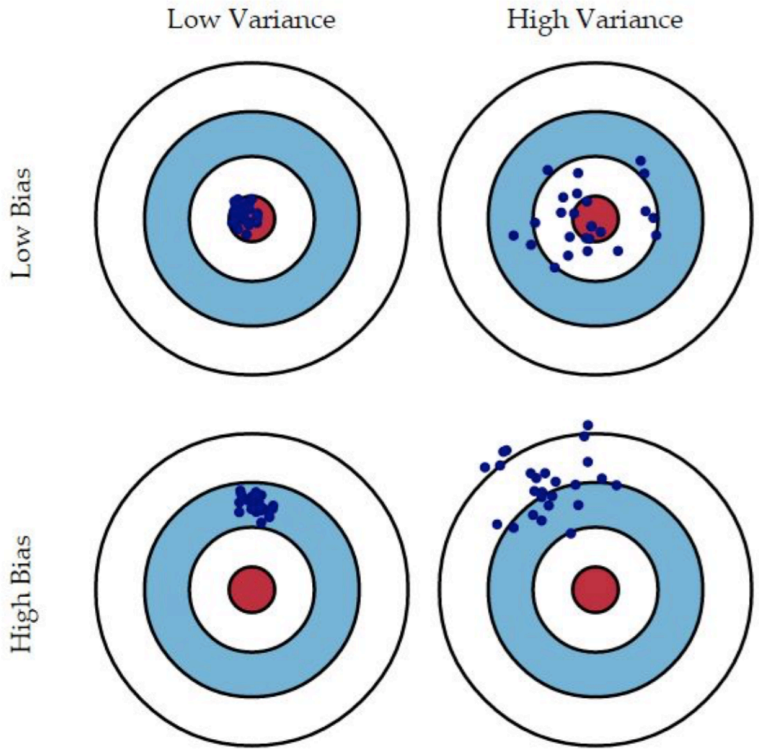
Overfitting means noise is dominating your model!

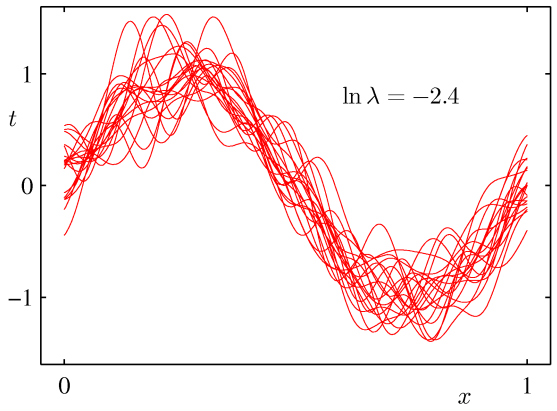


- ✓ Overfitting is the central problem of ML.
- ✓ Generalization of model
- ✓ Bad on unseen test set
- ✓ Avoid overfitting:
  - ✓ E.g., regularization term, prior, more data, model selection

Bias vs. Variance

Q: for adversaries, they want to increase bias or variance?





- ✓ Functions are trained on 100 bootstaps
- ✓ Overfitting: low bias, high variance
- ✓ Underfitting: high bias, low variance

Outline

- ❑ Introduction & Motivation
- ❑ Insight into Learning Models
- ❑ **Adversarial Learning Theory**
- ❑ Adversarial Attacks
- ❑ Towards Secure Learning
- ❑ Conclusions
- ❑ References

Learning formulation

To learn a model from training dataset $\mathcal{D}_{tr} \in \mathbb{R}^d$

$$f^* = \arg\min_{f \in \mathcal{F}} H\left(f\left(\mathcal{D}_{tr}\right)\right)$$

Namely selecting an optimal function from a function space F, where the evaluation function H is minimized.

Empirical Risk Minimization

$$f^* = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{(x,t) \in \mathcal{D}_{tr}} \ell\left(t, f(x)\right) + \lambda \Omega(f)$$

Loss function evaluating empirical risk

Regularization term controlling the function complexity

Adversarial Modeling

An adversary is defined as a vector encoded in a 3-tuple $K = (\mathcal{A}^*, f, \boldsymbol{\theta})$

$\mathcal{A}^*$ Optimal subset of adversarial samples

f The targeting learning model

$\boldsymbol{\theta}$ Attacker's knowledge

An attack strategy can be defined as

$$\begin{aligned} \max_{\mathcal{A}^*} \quad & \mathcal{W}(\mathcal{A}^*; \boldsymbol{\theta}) \\ \text{s.t.} \quad & \mathcal{A}^* \in \Phi(\mathcal{A}) \\ & \mathcal{C}(\mathcal{A}^*) \leq \tau \end{aligned}$$

e.g., classification error is maximised

Choose adversarial samples

Cost function of introducing adversarial noise

An example

A spam-filter f is a binary classifier discriminating an emails as either spam ($y = +1$) or legitimate ($y = -1$) . Now suppose an adversary can compose a set of spams $\mathcal{A} = \{x\}_{i=1}^{m}$ and send to f, Now the adversary can choose the adversarial samples by

$$\begin{aligned} \mathcal{A}^* = \quad & \arg\max_{\mathcal{A} \in \Phi(\mathcal{A})} \sum_{i=1}^{m} -f(x_i), \quad x_i \in \mathcal{A} \\ s.t. \quad & \mathcal{C}(\mathcal{A}) <= \tau \end{aligned}$$

It equals classifying most of the emails as legitimate. Note that Attacker's knowledge of the spam filter can be acquired by query-response interface provided by client.

Adversarial Capability

	Access to data	**Knowledge about features**	**Knowledge about classifier**
Poor Knowledge	From same distributional or not at all	No	No
Limited Knowledge	Partially or from same distributional	Maybe or inferred	Yes
Perfect Knowledge	Yes	Yes	Yes

Taxonomy of Adversarial Attacks

- ✓ Attacker's goal
  - ✓ E.g., maximize a classification error, compromise feature selection, etc..
- ✓ Attacker's knowledge
  - ✓ Knowledge about the training data
  - ✓ Knowledge about the learning function
  - ✓ Knowledge about the features
- ✓ Security violation
  - ✓ Integrity
  - ✓ Availability
- ✓ Adversarial Influence: Causative vs. Evasive
- ✓ Specificity: targeted vs. indiscriminative

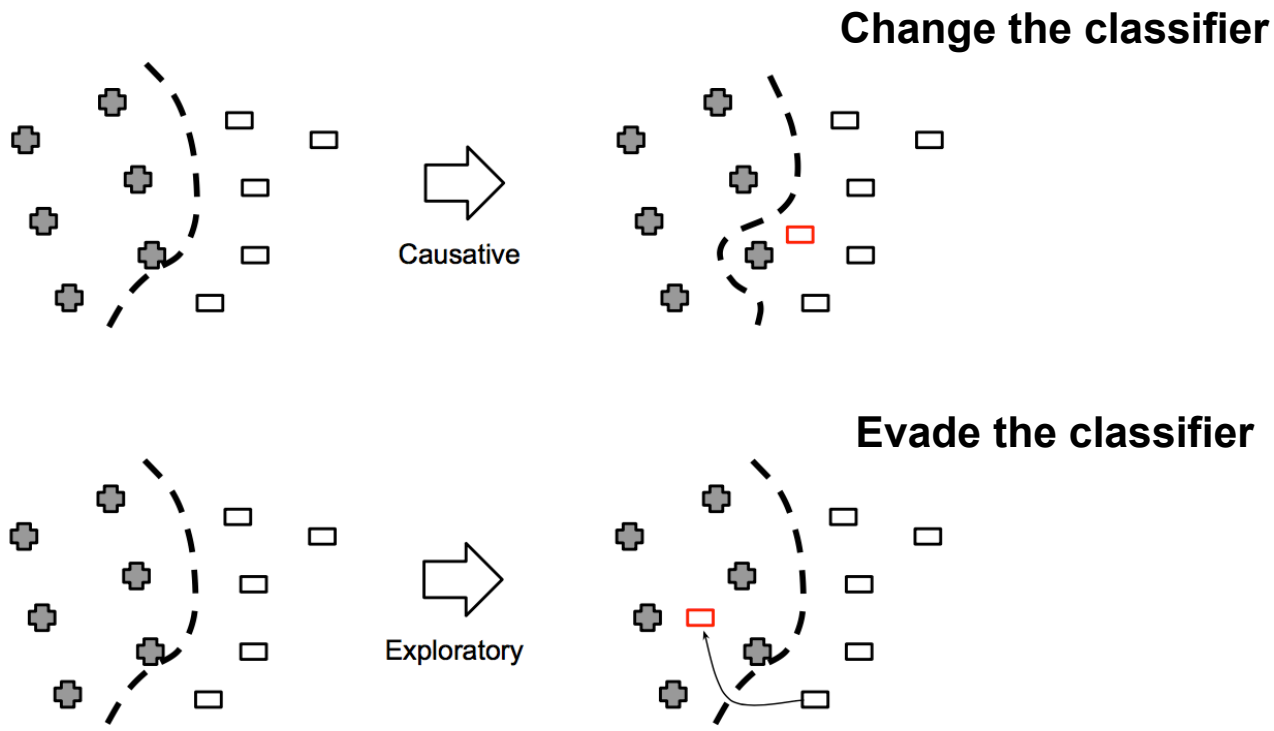
Taxonomy of Adversarial Attacks

	Integrity	Availability
Exploratory		
- *targeted*	Misclassifying a specific subset of positive samples.	Generally harm the learner's prediction on any of a specific subset samples.
- *indiscriminate*	Misclassifying positive samples generally.	The learner's prediction on any samples is compromised.
Causative		
- *targeted*	Classifier is mis-trained on particular positive samples.	Classifier is mis-trained on particular subset of samples.
- *indiscriminate*	Classifier is mis-trained generally on positive samples	Classifier is mis-trained generally on all samples.

E.g., Indiscriminate Causative Availability Attack aims to cause the classifier to misclassify samples indiscriminatively.

Typical Attack Types

Causative (Poisoning) vs. Exploratory (Evasion)



Outline

- ❑ Introduction & Motivation
- ❑ Insight into Learning Models
- ❑ Adversarial Learning Theory
- ❑ **Adversarial Attacks**
- ❑ Towards Secure Learning
- ❑ Conclusions
- ❑ References

Types of Adversaries

- ✓ Causative (poisoning attack)
  - ✓ Cracking how the learning algorithm works
  - ✓ Engineering on features or labels of training set
  - ✓ Change the *discriminant function* eternally
- ✓ Evasive (exploratory attack)
  - ✓ Engineering features of an evading point (test phase)
  - ✓ Circumvent the legitimate function
  - ✓ Change the *discriminant result*, not the function

Poisoning attack on SVMs

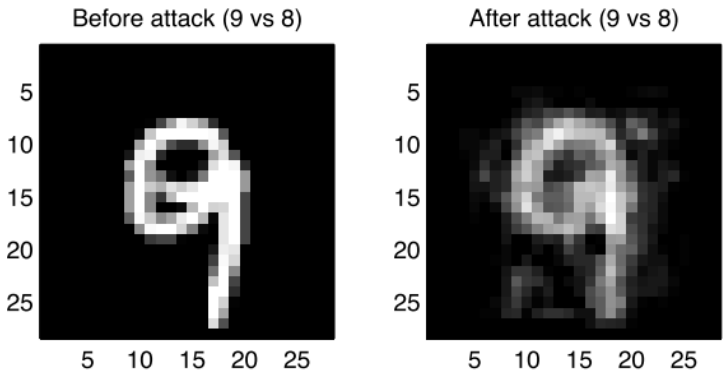
- ✓ Inject adversarial noises on features
  - ✓ Data stationary assumption is broken
- ✓ Adversarial objective: maximizing the error
  - ✓ E.g., test error, hinge loss, generalization error
- ✓ Using gradient ascend w.r.t the attack points

$$\max_{x_c} L(x_c) = \sum_{k=1}^{m} (1 - y_k f_{x_c}(x_k))_+ = \sum_{k=1}^{m} (-g_k)_+$$

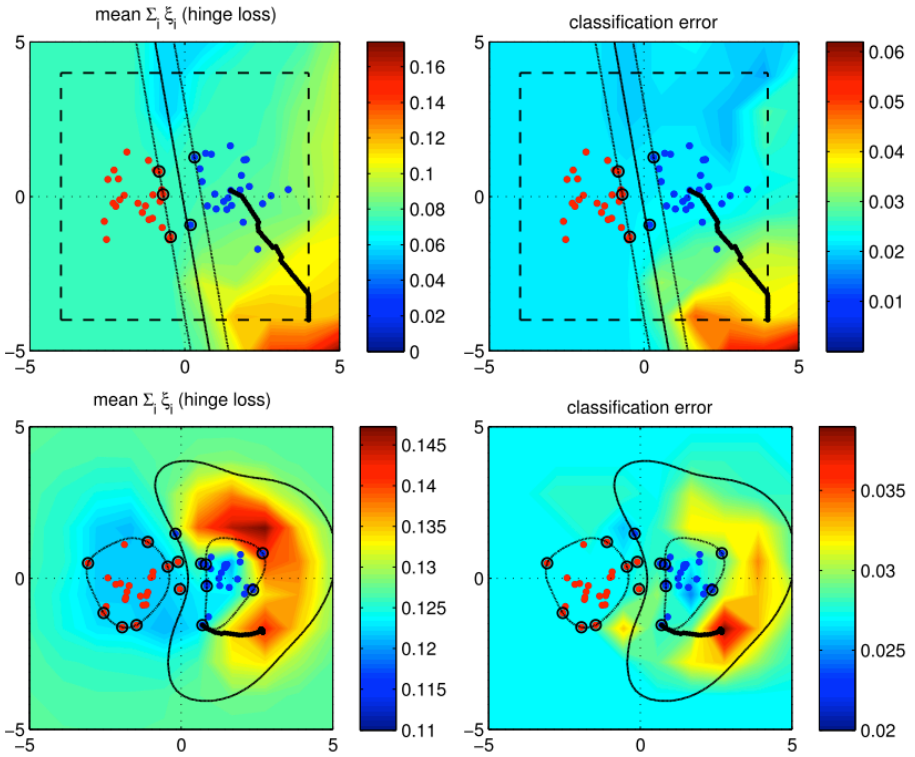
Adversarial sample we want to find out

The classifier is retrained on compromised dataset

Poisoning attack on SVMs



On MNIST digits dataset, a ‘9’ image is blurred as being a ‘8’ after attack



How adversarial samples are computed on both linear and RBF kernels (see bold gradient path)

Evasion Attacks

Properties

- ✓ Directedness
  - ✓ Directed: approximate a point towards a certain one
  - ✓ Undirected: probing the model boundary
- ✓ Closeness
  - ✓ Measure how closely an adversarial sample approaches its expected area
- ✓ Adversarial cost
  - ✓ How much effort must be put to achieve its effect

Mimicry Attack on SVMs

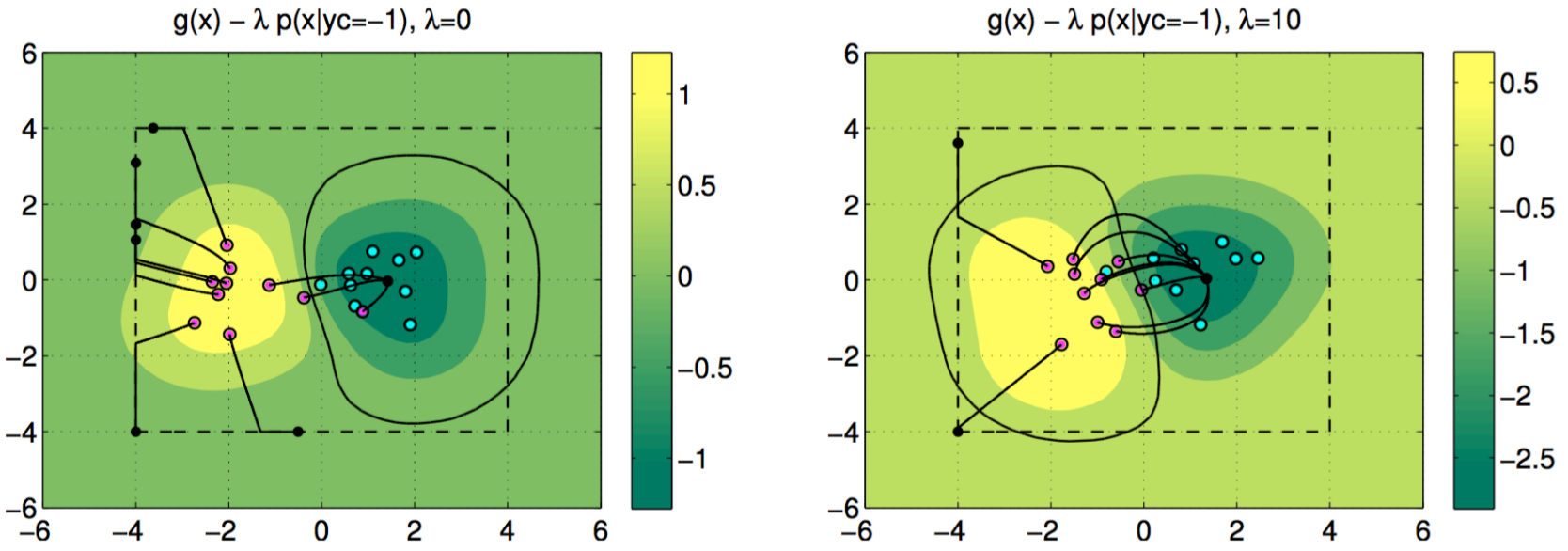
Again, we consider binary SVMs. A reckless attacker wants to manipulate a positive sample with decision function $g(x) > 0$, to evade being detected as positive. The evasion is defined as,

$$x^* = \arg\min_x \hat{g}(x) - \frac{\lambda}{n} \sum_{f(x_i)=+1} k\left(\frac{x - x_i}{h}\right)$$
$$\text{s.t.} \quad d(x, x_0) \leq \tau$$

Proxy classifier approximating target in case we don't know the true classifier

Meanwhile we don't want the point go to the dense area of targeting class.

Mimicry Attack on SVMs



Points are evading the class boundary towards dense area of opposite class.

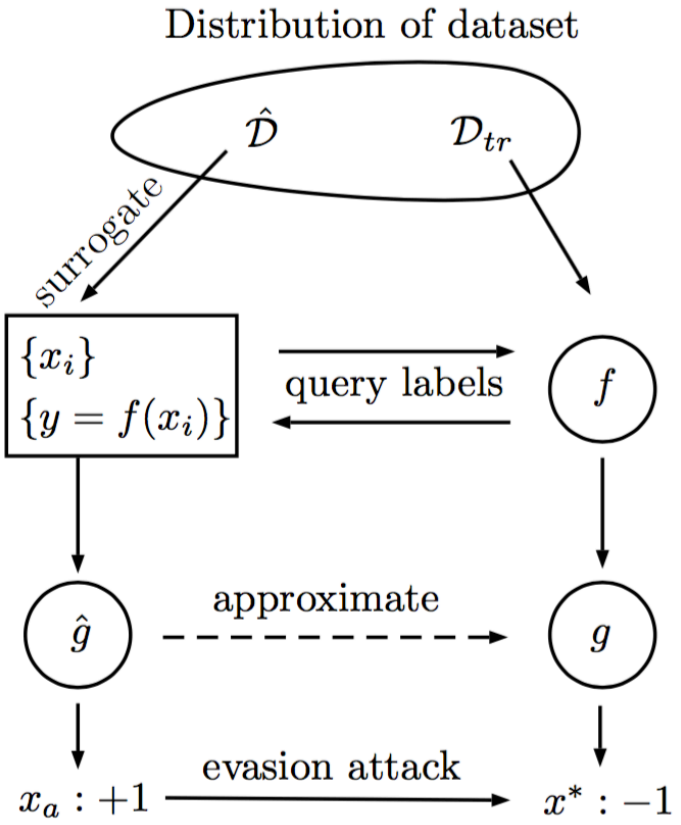
Evasion Attack Simulation

1. Collecting data from the same distribution, e.g., images of elephants by Google

2. Query labels by public interface

3. Simulating a proxy classifier

4. Compute evading samples on proxy classifier



Outline

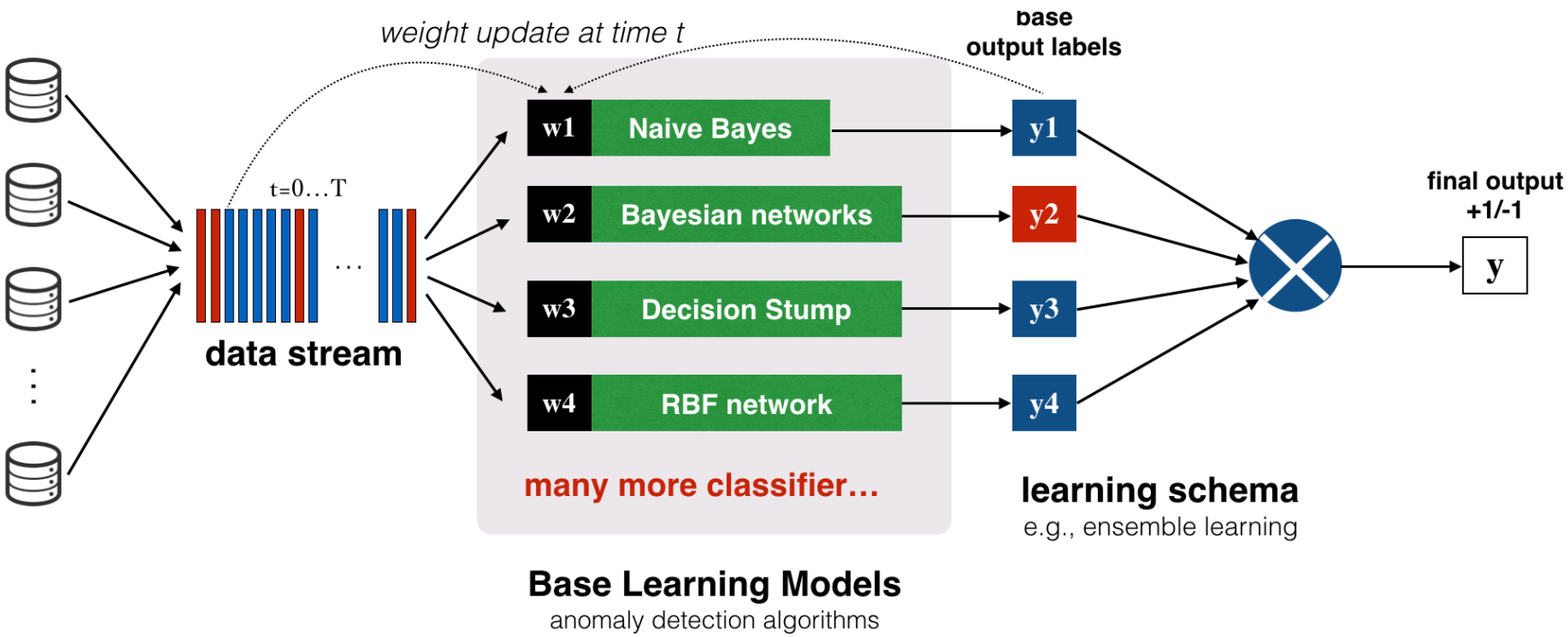
- ❑ Introduction & Motivation
- ❑ Insight into Learning Models
- ❑ Adversarial Learning Theory
- ❑ Adversarial Attacks
- ❑ **Towards Secure Learning**
- ❑ Conclusions
- ❑ References

Towards Secure Learning

- ✓ Attack is not our goal, but the only way to security
- ✓ Designing robust and secure learning algorithms
- ✓ Possible methods, e.g.,:
  - ✓ Adaptive regularization
  - ✓ Multiple Classifier Systems (or ensemble)
  - ✓ Learning with adversarial samples
  - ✓ Robust statistics

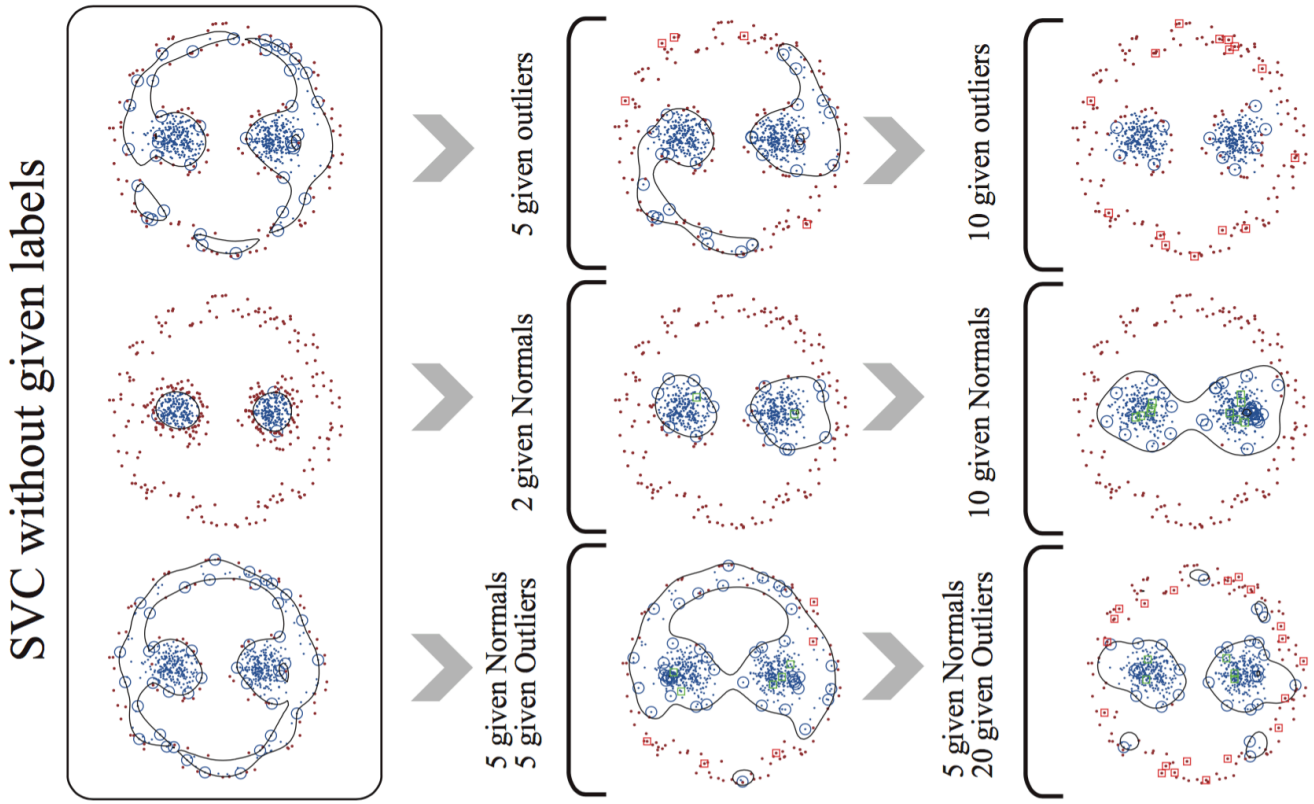
Multiple Classifier Systems

Main idea is to adapt classifier to external changes



Adaptive Regularization

Penalty on various points is defined as different.



Outline

- ❑ Introduction & Motivation
- ❑ Insight into Learning Models
- ❑ Adversarial Learning Theory
- ❑ Adversarial Attacks
- ❑ Towards Secure Learning
- ❑ **Conclusions**
- ❑ References

Conclusions

- ✓ Machine learning is powerful but not always reliable
- ✓ Integrity and availability of big data systems can be compromised
- ✓ Adversarial learning is a rising research area
- ✓ Traditional IT security must be patched with securing learning based systems
- ✓ Causative and evasive attacks are real and potentially dangerous
- ✓ Secure and robust design of learning algorithms are expected
- ✓ Robustness can be enhanced by model smoothing（MCS）or adaptive regularization, for example.

References

1. Biggio, B., Nelson, B. and Laskov, P., 2012. Poisoning attacks against support vector machines. In J. Langford & J. Pineau, eds. ICML. Omnipress, pp. 1807–1814.
2. Biggio, B. et al., 2013. Evasion attacks against machine learning at test time. In H. Blockeel et al., eds. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Part III. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 387–402.
3. Xiao, H. and Eckert, C., 2013. Indicative Support Vector Clustering with its Application on Anomaly Detection. In ICMLA. Miami, Florida, USA.
4. Barreno, M., Nelson, B., Joseph, A.D. and Tygar, J.D., 2010. The security of machine learning. Machine Learning, 81(2).
5. Barreno, M. et al., 2006. Can machine learning be secure? In Proceedings of the 2006 Symposium on Information, computer and communications security. ASIACCS ’06. New York, NY, USA: ACM, pp. 16–25.
6. Ben-hur, A. and Aviv, T., 2000. A Support Vector Method for Clustering. Proc. of the 15th International Conference on Pattern Recognition.
7. Dalvi, N. et al., 2004. Adversarial classification. In Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’04. Seattle, p. 99.
8. Lowd, D. and Meek, C., 2005. Adversarial Learning. In A. C. M. Press, ed. Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). Chicago, IL, USA: ACM Press, pp. 641–647.
9. Nguyen, A., Yosinski, J. and Clune, J., 2014. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In Computer Vision and Pattern Recognition 2015. Boston, Massachusetts, US.
10. Xiao, H. et al., 2015. Is Feature Selection Secure against Training Data Poisoning ? Int’l Conf. on Machine Learning (ICML).
11. Xiao, H. et al., 2014. Support Vector Machines under Adversarial Label Contamination. Journal of Neurocomputing, Special Issue on Advances in Learning with Label Noise, 160(0)., pp.53–62.