# Adversarial Machine Learning

Ankur Bapna
Gavilan Galloway

# Open Problems

- Bounds on adversarial influence
- Value of adversarial capabilities
- Technologies for secure learning

# Bounds on influence?

- Effort needed for adversary to influence
- Lower bounds on performance
- Some systems harder to reverse engineer?

# Value of adversarial influence?

- Natural threat models and impact on learner
- How much influence can the learner tolerate

# Secure learners for security

- Detecting malicious training instances
- More resilient learners
- Mixture of orthogonal experts

# Case 1: Poisoning SVM

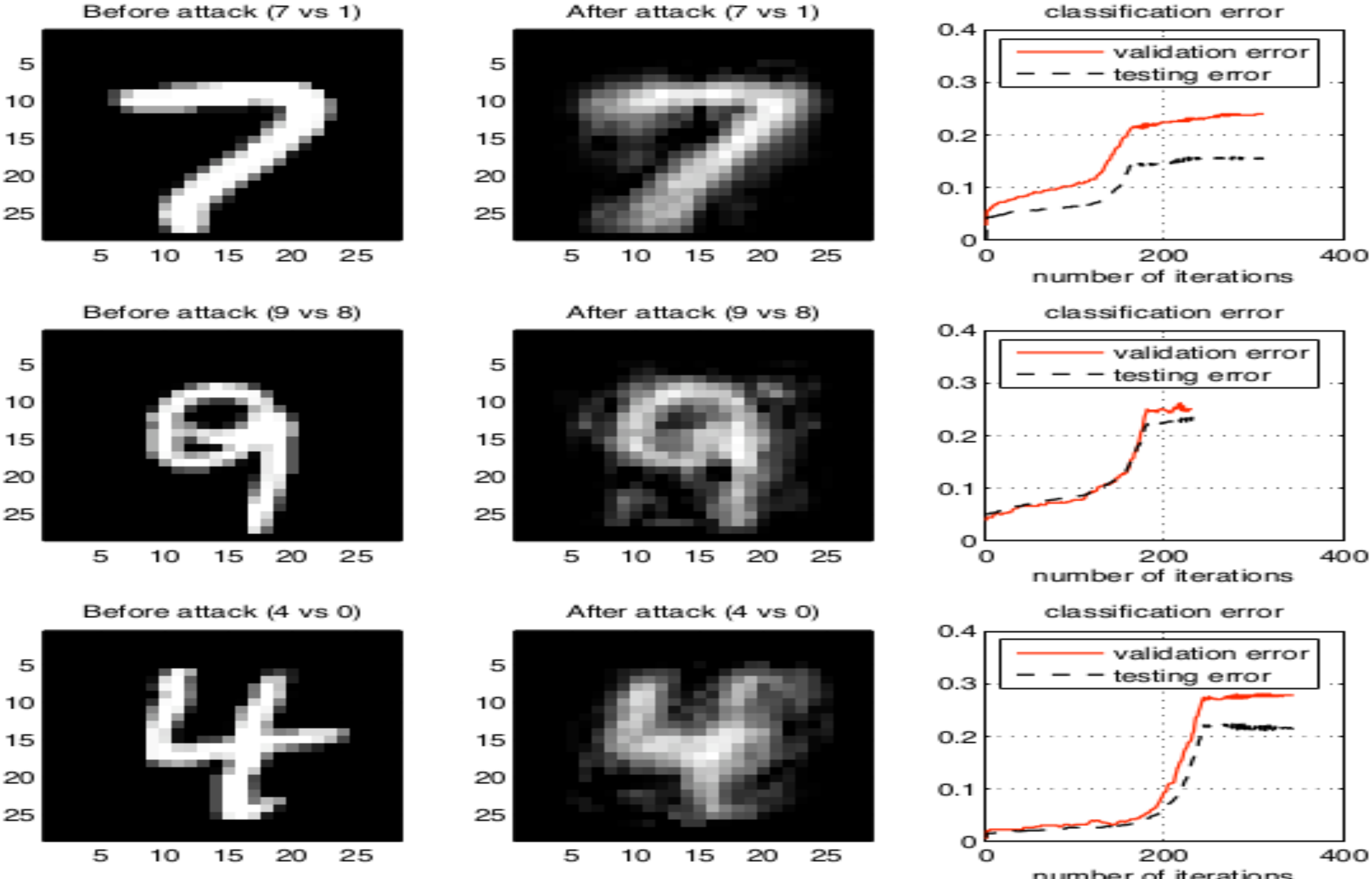
- Good points
  - Theoretical guarantees
  - Optimally poisoning a SVM based classifier
  - Works for kernelized SVM
- Not so good
  - Optimization problem - non-convex
  - Can’t be used for practical attacks
  - Verified on extremely small datasets

# Formulation

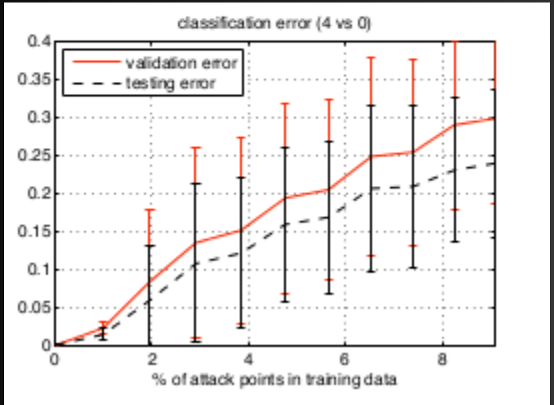
- Goal
  - Introduce single malicious sample in training set
  - Maximize validation error
- Frame goal as an optimization problem
- Algorithm
  - Start with a training sample
  - Solve locally using gradient ascent
  - Resulting gradient : function of inner product

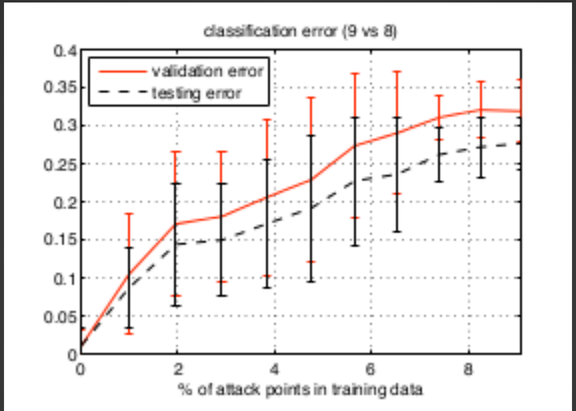
# Experiments

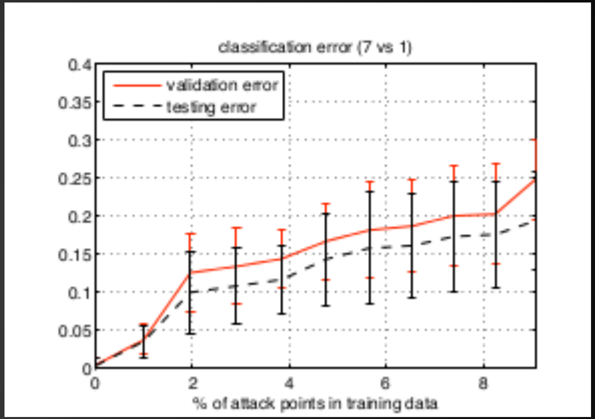
- 100 training, 500 validation samples
- MNIST Dataset - binary classification
  - 7 vs 1
  - 9 vs 8
  - 4 vs 0


Before attack (7 vs 1)
After attack (7 vs 1)
classification error
validation error
testing error
number of iterations
Before attack (9 vs 8)
After attack (9 vs 8)
classification error
validation error
testing error
number of iterations
Before attack (4 vs 0)
After attack (4 vs 0)
classification error
validation error
testing error

# Performance



# Result

- Error rate ~ 2-5% for label flipping
- Error rate ~ 15-20% for optimal poisoning

# Possible improvements

- Multi-point optimization
- Less adversary control
  - Optimal label flipping for existing data point
  - Generating data point without label
- More practical algorithm

# Case 2: Poisoning resistant PCA

- Evaluate existing algorithm with poisoning
  - Random poisoning
  - Informed poisoning (local and global)
  - Short term and boiling frog attacks
- Improve performance - robust statistics
  - Resistant to boiling frog attacks
  - Still susceptible to random poisoning

# Flow volume anomaly detection using PCA

- Training
  - Y - cumulative traffic matrix containing time series for all network flows
  - Find principal components of Y
  - k components - maximum variance for training data
  - Model residual using Gaussian - find threshold t
- Test
  - Variance not explained by top k components > t
    - Report anomaly

# Poisoning

- Random :
  - Add random traffic during training
- Locally informed :
  - Information about current ingress traffic
  - Add traffic if current ingress traffic is large
- Globally informed :
  - Omniscient, Omnipotent attacker
  - Add traffic optimally

# Performance of current system

- Network information
  - advantageous
- Boiling frog attack
  - more efficient
- Need for more robust method

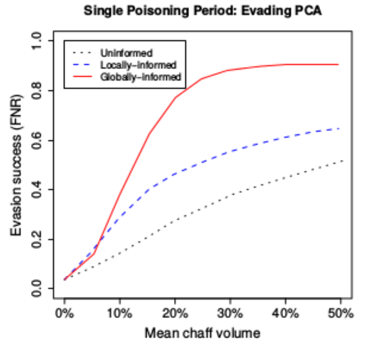



**Figure 3:** Evasion success of PCA under *Single-Training Period* poisoning attacks using 3 chaff methods.

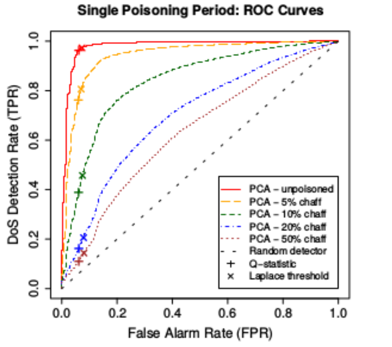



**Figure 4:** ROC curves of PCA under *Single-Training Period* poisoning attacks.

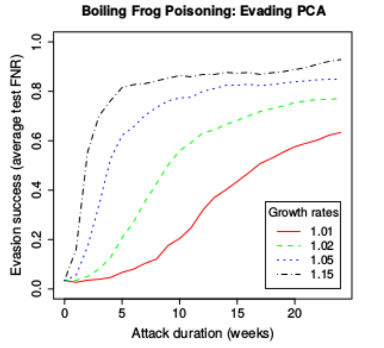



**Figure 5:** Evasion success of PCA under *Boiling Frog* poisoning attacks.

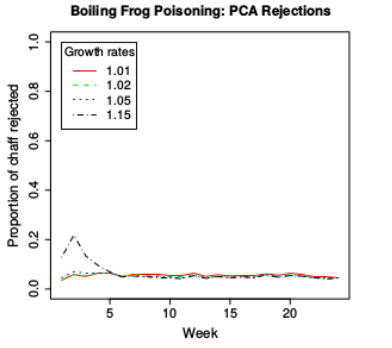



**Figure 6:** Chaff rejection rates of PCA under poisoning attacks shown in Fig. 5.

# ANTIDOTE

- Median instead of mean for centering
- Median absolute deviation instead of variance
- Evaluate Principal Components: PCA-GRID
    - Grid search algorithm
- Use Laplace distribution to model residual

# Performance of ANTIDOTE

- More resistant to informed poisoning
- More resistant to boiling frog attacks
- Vulnerable to random poisoning

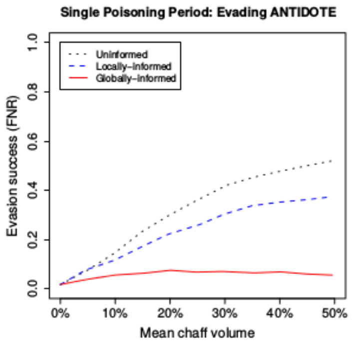



Figure 7: Evasion success of ANTIDOTE under *Single-Training Period* poisoning attacks using 3 chaff methods.

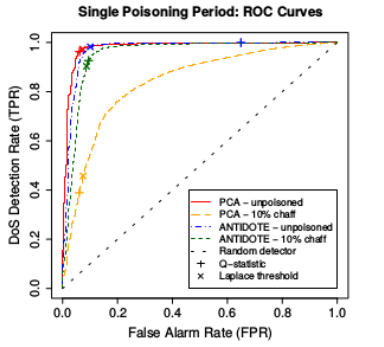



Figure 8: ROC curves of ANTIDOTE under *Single-Training Period* poisoning attacks.

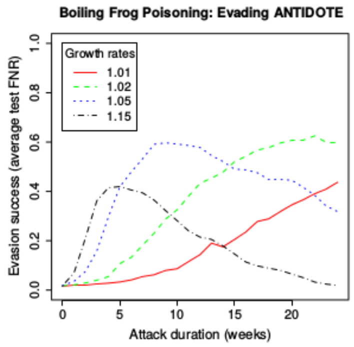



Figure 11: Evasion success of ANTIDOTE under *Boiling Frog* poisoning attacks.

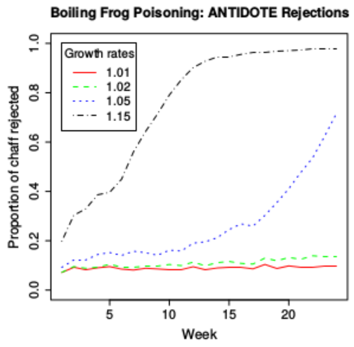



Figure 12: Chaff rejection rates of ANTIDOTE under *Boiling Frog* poisoning attacks.

# Thanks!!

Questions??

# References


- Open Problems in the Security of Learning, Tygar et. al.
- Poisoning Attacks against Support Vector Machines, Biggio et. al.
- ANTIDOTE: Understanding and Defending against Poisoning of Anomaly Detectors, Tygar et. al.