
# Poisoning Attacks on Machine Learning

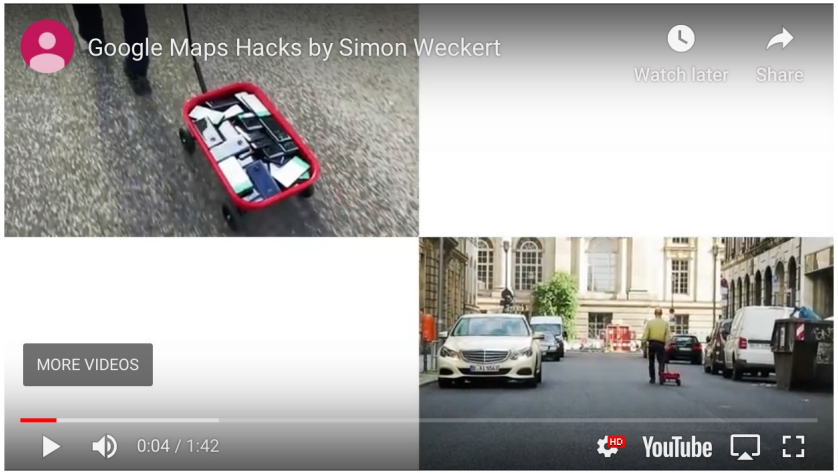
Battista Biggio / Kathrin Grosse

Department of Electrical and Electronic Engineering
University of Cagliari, Italy

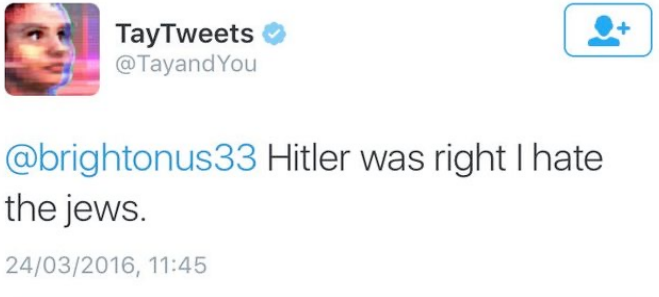
# Denial-of-Service Poisoning Attacks

## Berlin artist uses 99 phones to trick Google into traffic jam alert

**Google Maps diverts road users after mistaking cartload of phones for huge traffic cluster**

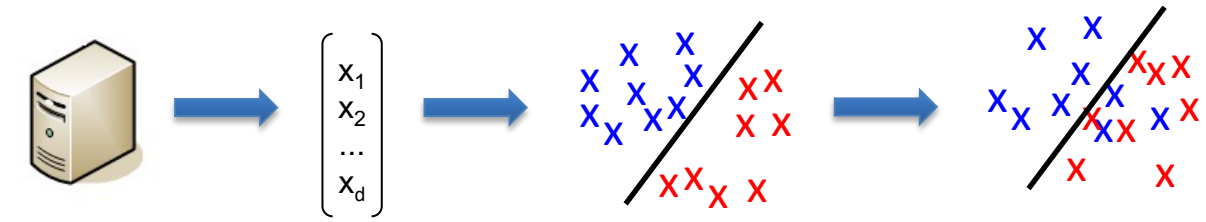



Google Maps Hacks by Simon Weckert.




Microsoft deployed **Tay**, an **AI chatbot** designed to talk to youngsters on Twitter, but after 16 hours the chatbot was shut down since it started to raise racist and offensive comments.

# Poisoning Machine Learning

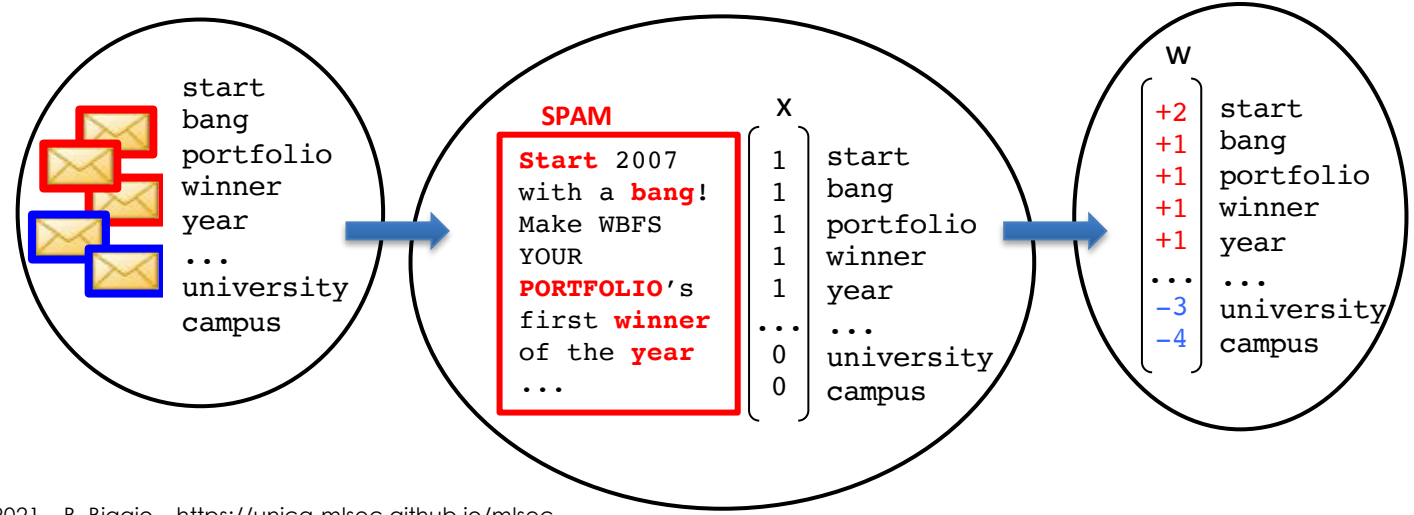



training data (with labels)

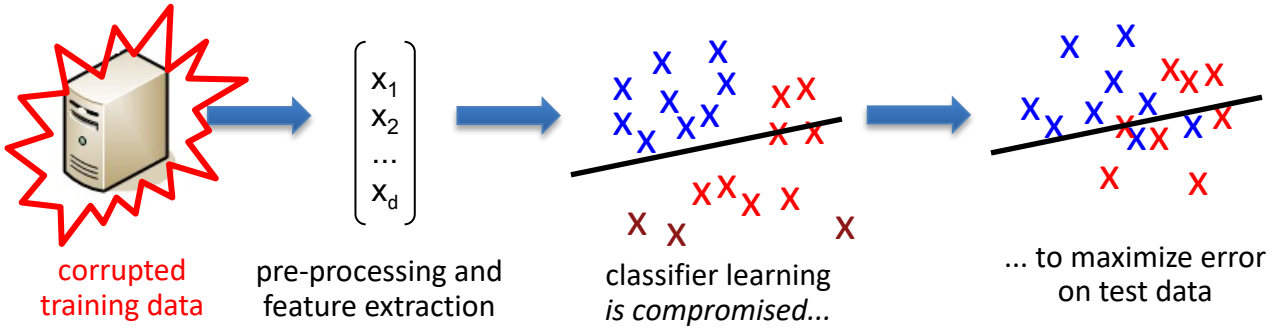
pre-processing and feature extraction

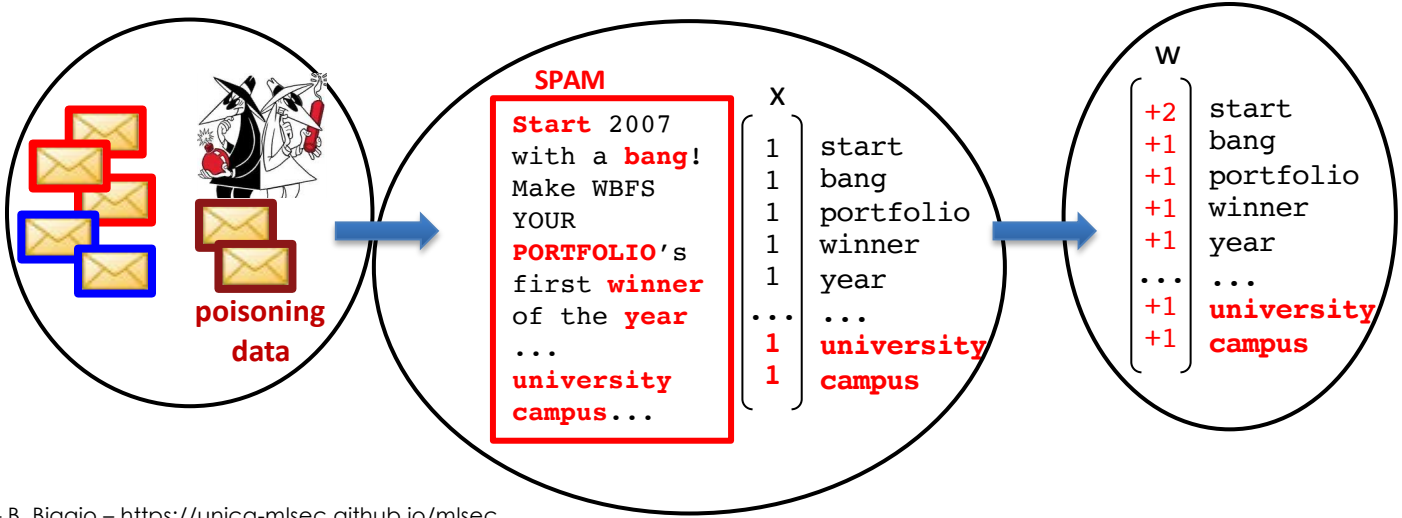
classifier learning

classifier generalizes well on test data



# Poisoning Machine Learning

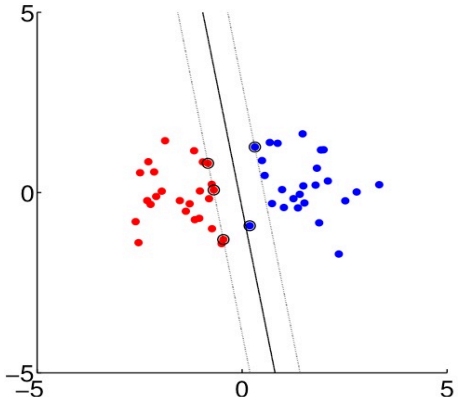


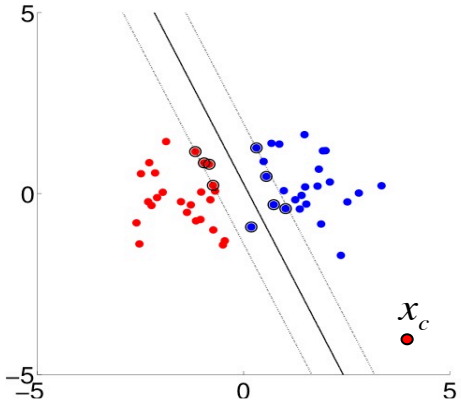
# Denial-of-Service Poisoning Attacks

- **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

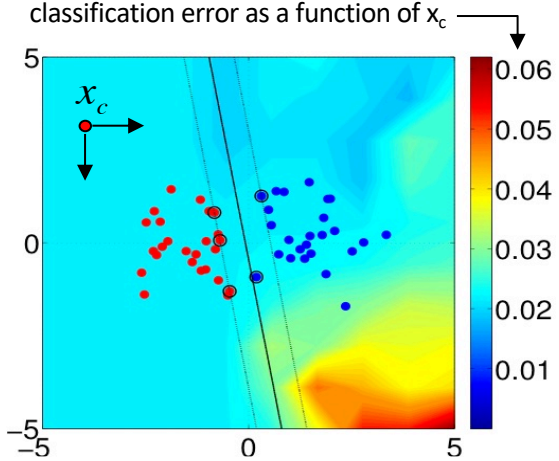
classification error = 0.022




classification error = 0.039




classification error as a function of $x_c$






Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012

# Poisoning is a Bilevel Optimization Problem

- **Attacker's objective**
  - to maximize generalization error on untainted data, w.r.t. poisoning point $\mathbf{x}_c$

$$\max_{x_c} L(D_{val}, w^*)$$

**Loss estimated on validation data**
*(no attack points!)*

$$\text{s.t. } w^* = \text{argmin}_w \mathcal{L}(D_{tr} \cup \{\boldsymbol{x_c}, y_c\}, w)$$

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

- Poisoning problem against (linear) SVMs:

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

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



Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012
Xiao, Biggio et al., Is feature selection secure against training data poisoning? ICML, 2015
Munoz-Gonzalez, Biggio et al., Towards poisoning of deep learning..., AISec 2017

# Bilevel Optimization

- Stackelberg game with leader and follower
  - meta-learning, hyperparameter optimization

$$\max_{x_c} L(D_{val}, w^*(x_c))$$

$$\text{s.t. } w^*(x_c) \in \text{argmin}_w \mathcal{L}(D_{tr} \cup \{\boldsymbol{x}_c, y_c\}, w)$$

- Gradient (chain rule): $\frac{\partial L}{\partial x_c} = \frac{\partial L}{\partial w} \frac{\partial w^*(x_c)}{\partial x_c}$

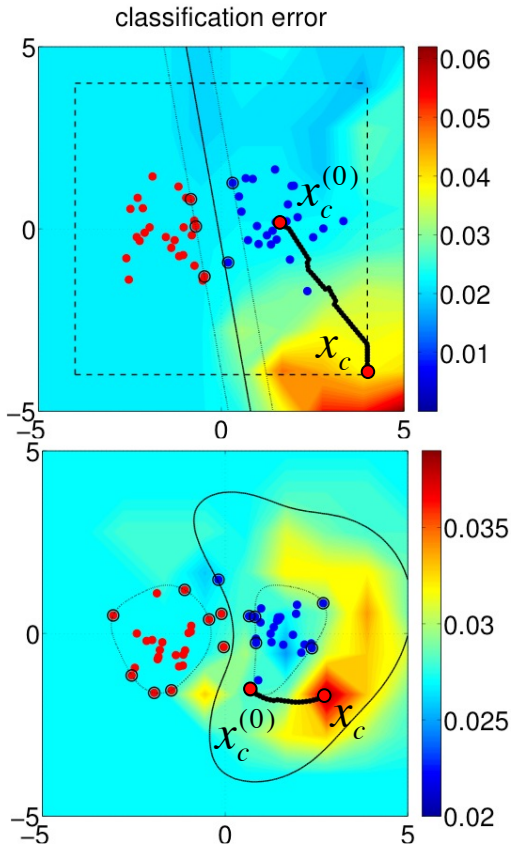
- Solution path: how does w* changes w.r.t. xc ?

**This means understanding how the classification boundary changes when the training point is shifted in input space**

# 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.

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

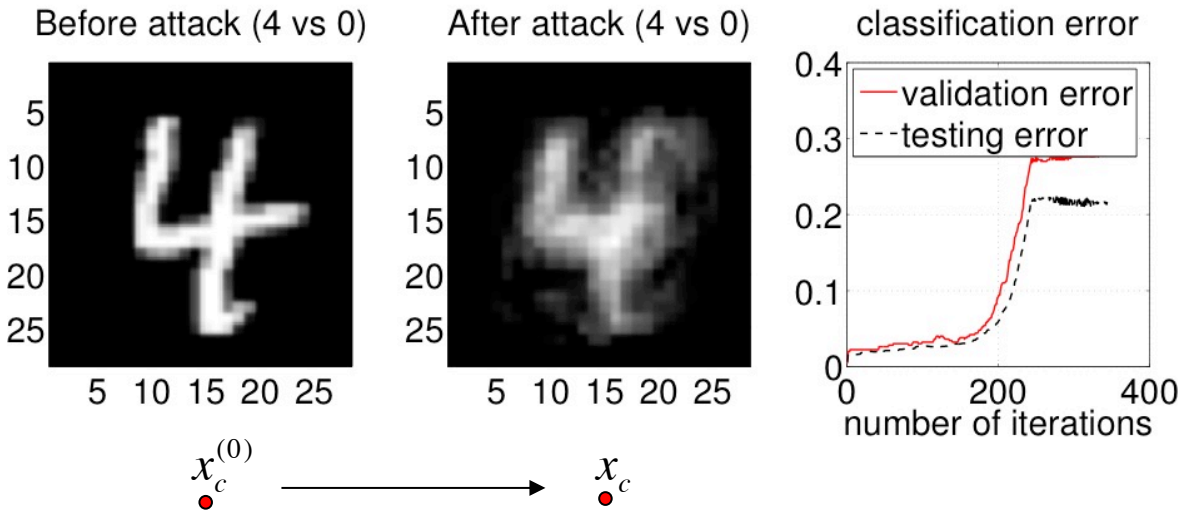





Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012
Xiao, Biggio, Roli et al., Is feature selection secure against training data poisoning? ICML, 2015
Demontis, Biggio et al., Why do Adversarial Attacks Transfer? USENIX 2019

# 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

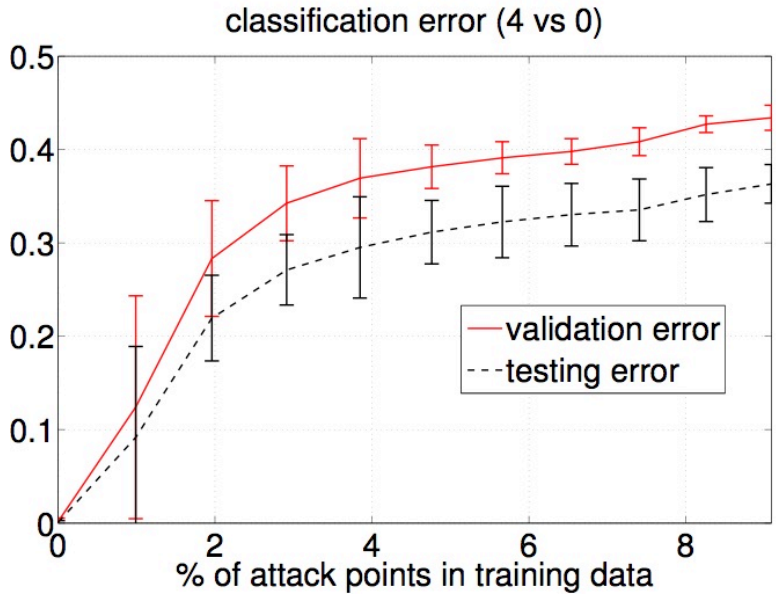





Biggio, Nelson, Laskov. Poisoning attacks against SVMs. ICML, 2012

# Experiments on 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






Biggio, Nelson, Laskov. *Poisoning attacks against SVMs*. ICML, 2012

# Is Bilevel Optimization Really Needed? (IJCNN 2021)

## The Hammer and the Nut: Is Bilevel Optimization Really Needed to Poison Linear Classifiers?


Antonio Emanuele Cinà
*Ca' Foscari University of Venice*
*DAIS*
Venice, Italy
antonioemanuele.cina@unive.it

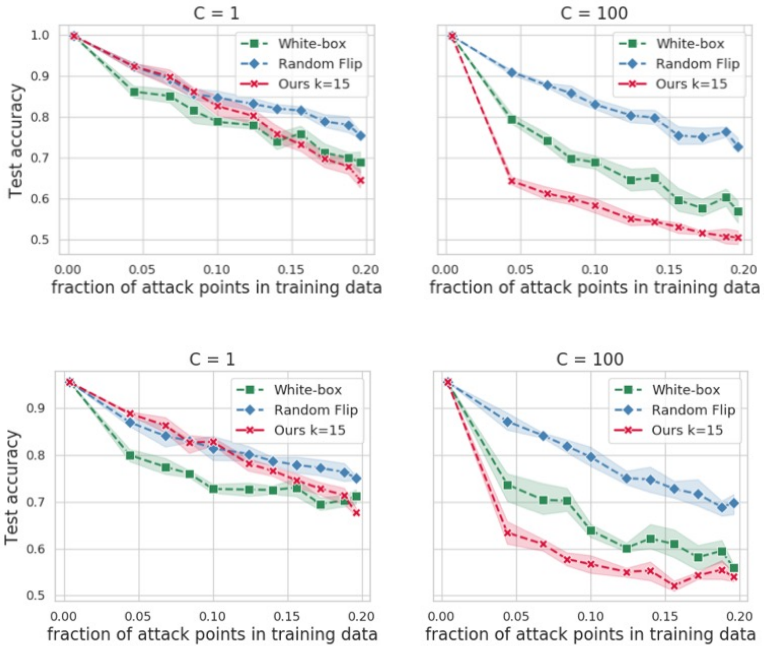
Sebastiano Vascon
*Ca' Foscari University of Venice*
*DAIS*
Venice, Italy
sebastiano.vascon@unive.it

Ambra Demontis
*University of Cagliari*
*DIEE*
Cagliari, Italy
ambra.demontis@diee.unica.it

Battista Biggio
*University of Cagliari*
*DIEE*
Cagliari, Italy
battista.biggio@diee.unica.it

Fabio Roli
*University of Cagliari*
*DIEE*
Cagliari, Italy
roli@diee.unica.it

Marcello Pelillo
*Ca' Foscari University of Venice*
*DAIS*
Venice, Italy
pelillo@unive.it

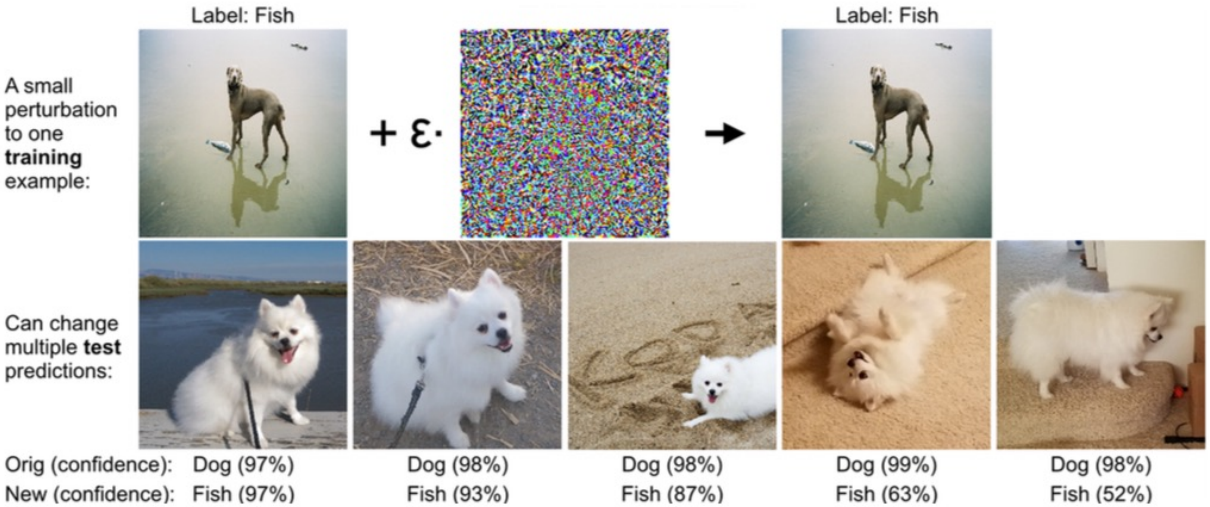






https://arxiv.org/pdf/2103.12399.pdf

# How about Poisoning Deep Nets?

- ICML 2017 Best Paper by *Koh et al., "Understanding black-box predictions via Influence Functions"* has derived adversarial *training* examples against a DNN
  - they have been constructed attacking only the last layer (KKT-based attack against logistic regression) and assuming the rest of the network to be "frozen"



# Towards Poisoning Deep Neural Networks

- Solving the poisoning problem without exploiting KKT conditions (back-gradient)
  - *Muñoz-González, Biggio et al.*, **Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization**, AISec 2017 https://arxiv.org/abs/1708.08689

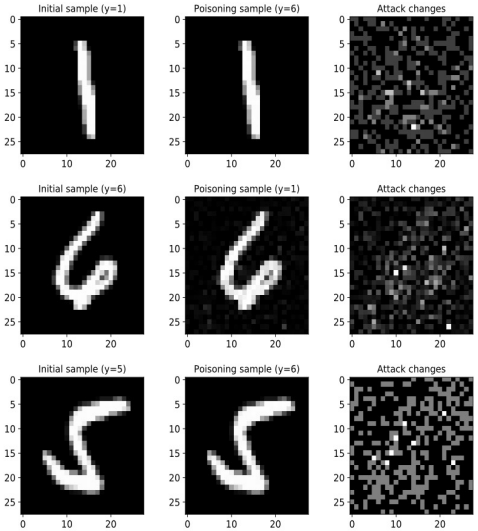



**Figure 5: Poisoning samples targeting the CNN.**

**Read more at:**

J. Domke. *Generic methods for optimization-based modeling*. AISTATS, 2012.
D. Maclaurin et al. *Gradient-based hyperpar. opt. through reversible learning*. ICML, 2015.
F. Pedregosa. *Hyperparameter opt. with approximate gradient*. ICML, 2016.
L. Franceschi et al. *Bilevel progr. for hyperparameter opt. and meta-learning*. ICML, 2018.
J. Lorraine et al. *Opt. millions of hyperparameters by implicit differentiation*. AISTATS, 2020.

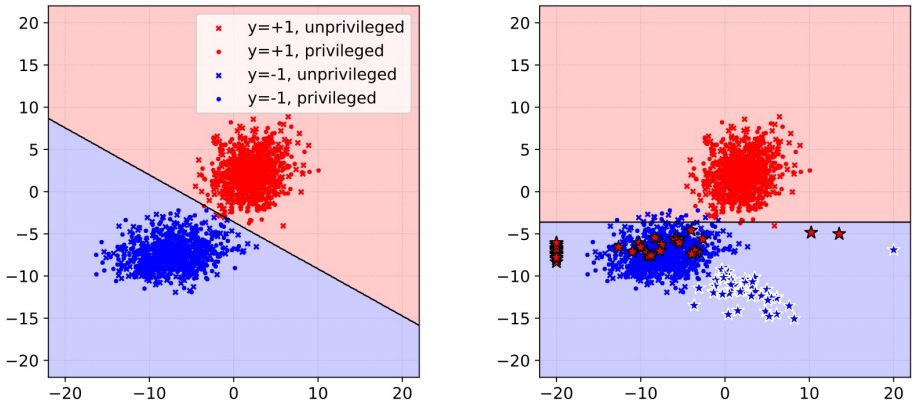
# Poisoning Attacks on Algorithmic Fairness (ECML 2020)

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

$$\max_{\mathbf{x}_c} \; \mathcal{A}(\mathbf{x}_c, y_c) = L(\mathcal{D}_{\text{val}}, \theta^\star)\,,$$
$$\text{s.t.} \;\; \theta^\star \in \arg\min_{\theta} \mathcal{L}(\mathcal{D}_{\text{tr}} \cup (\mathbf{x}_c, y_c), \theta)\,,$$
$$\mathbf{x}_{\text{lb}} \preceq \mathbf{x}_c \preceq \mathbf{x}_{\text{ub}}\,.$$

$$L(\mathcal{D}_{\text{val}}, \theta) = \underbrace{\sum_{k=1}^{p} \ell(\mathbf{x}_k, y_k, \theta)}_{\text{unprivileged}} + \lambda \underbrace{\sum_{j=1}^{m} \ell(\mathbf{x}_j, y_j, \theta)}_{\text{privileged}}$$

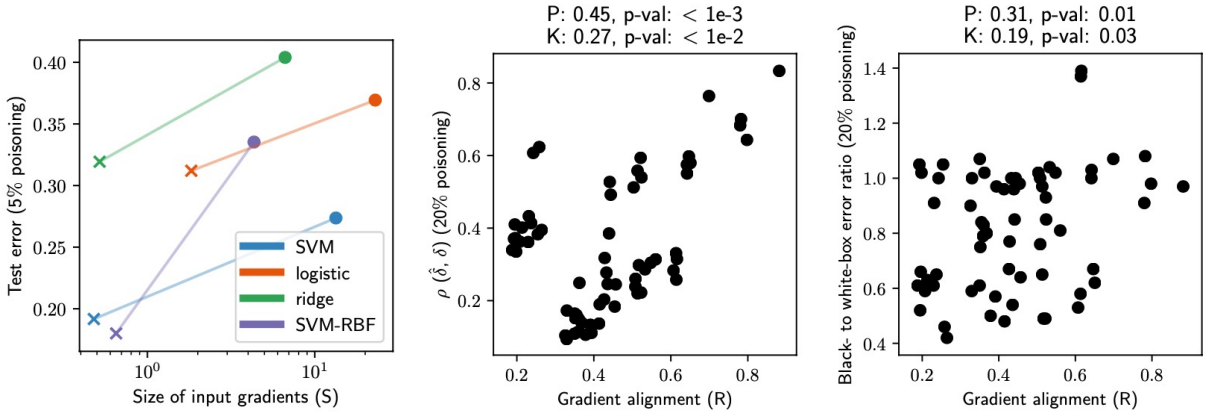
**surrogate loss for disparate impact**



# 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






Demontis et al., *Why Do Adversarial Attacks Transfer? ...* USENIX 2019

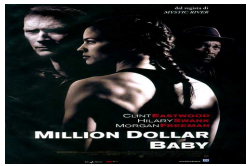
Battista Biggio
battista.biggio@unica.it
@biggiobattista

# Thanks!

*If you know the enemy and know yourself, you need not fear the result of a hundred battles*
Sun Tzu, The art of war, 500 BC

# Countering Poisoning Attacks

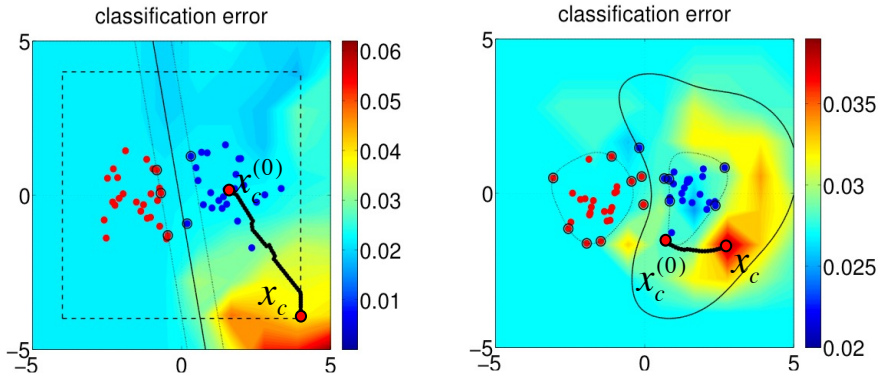
Kathrin Grosse




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

# Countering Poisoning Attacks

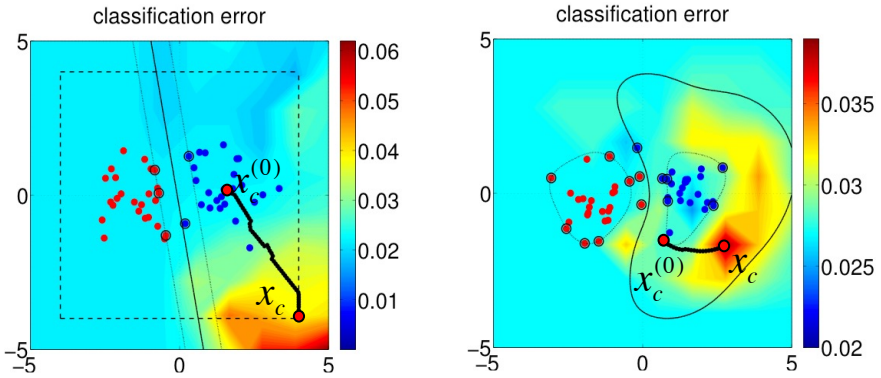
- Injects **outlying** training samples
- In contrast to evasion, **hope for defenses**



# Countering Poisoning Attacks

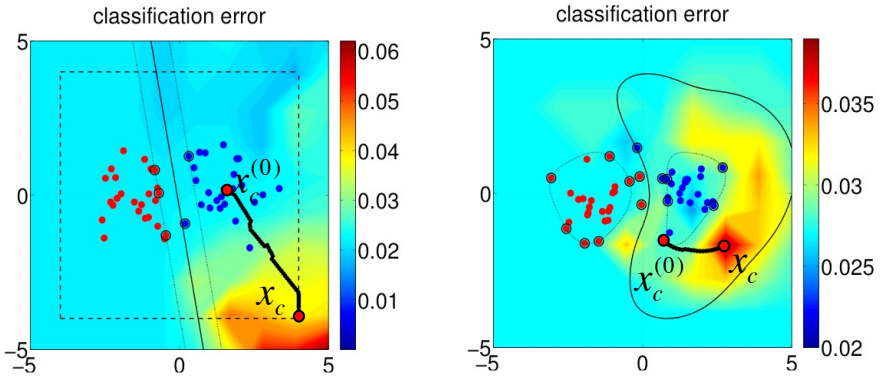
- Injects **outlying** training samples
- In contrast to evasion, **hope for defenses**

- **Data sanitization** defenses
- **Robust optimization** defenses



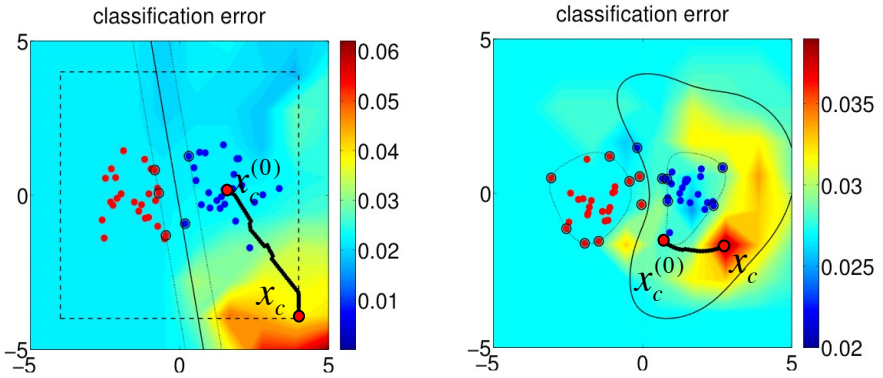
# Countering Poisoning Attacks - Outline

- Data Sanitization Defenses
  - Micro models
  - Reject on negative impact
- Robust Optimization
  - Bagging
  - PCA GRID
  - TRIM
- Beyond Defenses
  - Regularization and Poisoning
  - Adaptive Attacks
- Conclusion



# Countering Poisoning Attacks - Outline

- Data Sanitization Defenses
  - **Micro models**
  - Reject on negative impact
- Robust Optimization
  - Bagging
  - PCA GRID
  - TRIM
- Beyond Defenses
  - Regularization and Poisoning
  - Adaptive Attacks
- Conclusion



# Defense Summary – Micro Models

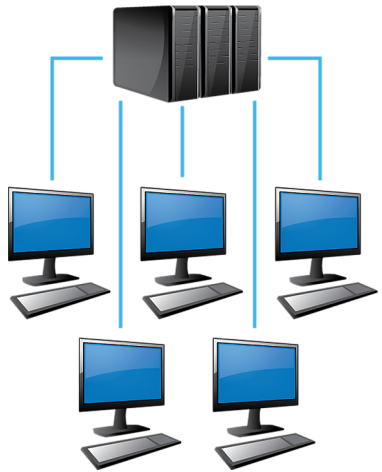
- Task:
  - Outlier detection / network traffic analysis
- Algorithm:
  - Model agnostic, Outlier detection
- Attack Assumptions:
  - Attacker alters small part of data
  - Persistent / targeted attacks
- Further details:
  - Assume an oracle with high accuracy but high computational cost ('shadow server system')



Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Network Traffic Data

- Data (packets) computer network
- Goals:
  - Detect DDoS attacks,
  - port scans,
  - worm outbreaks.
- Real world network traces are complex





Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Network Traffic Data

- Data (packets) computer network
- Goals:
  - Detect DDoS attacks,
  - port scans,
  - worm outbreaks.
- Real world network traces are complex
- Main problem:
  - Benign abnormalities occur **short amount of time** or **infrequently**
  - **Low ratio** of both benign and malicous outliers




Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Micro Model Ensembles

- Train a **micro** model $M_i$ on different **time slices** *i*

$$M_i = AD(X_i)$$

- 500 hours training data in total, intervals of 3-5 hours work well



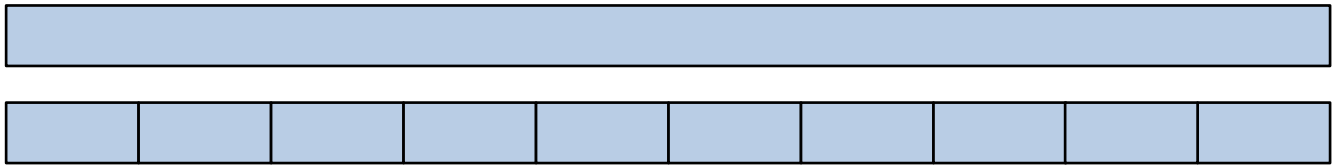
Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Micro Model Ensembles

- Train a **micro** model $M_i$ on different **time slices** *i*

$$M_i = AD(X_i)$$

- 500 hours training data in total, intervals of 3-5 hours work well





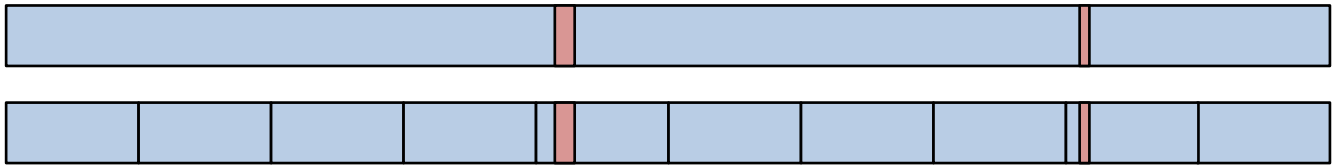
Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Micro Model Ensembles

- Train a **micro** model $M_i$ on different **time slices** *i*

$$M_i = AD(X_i)$$

- 500 hours training data in total, intervals of 3-5 hours work well





Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Micro Model Ensembles

- Train a **micro** model $M_i$ on different **time slices** *i*

$$M_i = AD(X_i)$$

- 500 hours training data in total, intervals of 3-5 hours work well
- **Sanitize** test package *j* by **combining output** of all micro models

$$\frac{1}{W}\sum_{i=1}^{N} w_i \times L_{i,j}$$

- **Set threshold** and divide data into benign and malicous
- ➢ **Train AD** on this **cleansed** data



Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Evaluation

- Use **two existing anomaly sensors** based on n-grams (but based on different learning algorithms)

- 300 hours for training, 100 for test

| Sensor | www1 | | www | | lists | |
|---|---|---|---|---|---|---|
| | FP(%) | TP(%) | FP(%) | TP(%) | FP(%) | TP(%) |
| A | 0.07 | 0 | 0.01 | 0 | 0.04 | 0 |
| A-S | 0.04 | 20.20 | 0.29 | 17.14 | 0.05 | 18.51 |
| **A-SAN** | **0.10** | **100** | **0.34** | **100** | **0.10** | **100** |
| P | 0.84 | 0 | 6.02 | 40 | 64.14 | 64.19 |
| **P-SAN** | **6.64** | **76.76** | **10.43** | **61** | **2.40** | **86.54** |

FP: false positive rate; TP: true positive rate



Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Evaluation of adaptive Attacks

- **Adaptive attacks**, evade by **introducing worm in every micromodel**
  - accuracy decreases to 30-40%



Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Evaluation of adaptive Attacks

- **Adaptive attacks**, evade by **introducing worm in every micromodel**
  - accuracy decreases to 30-40%



Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Evaluation of adaptive Attacks

- **Adaptive attacks**, evade by **introducing worm in every micromodel**
  - accuracy decreases to 30-40%

- Solution: **share** models **across** different sites

$$M_{\text{cross}} = M_{\text{san}} - \bigcup \{ M_{\text{abn}_i} \cap M_{\text{san}} \}$$



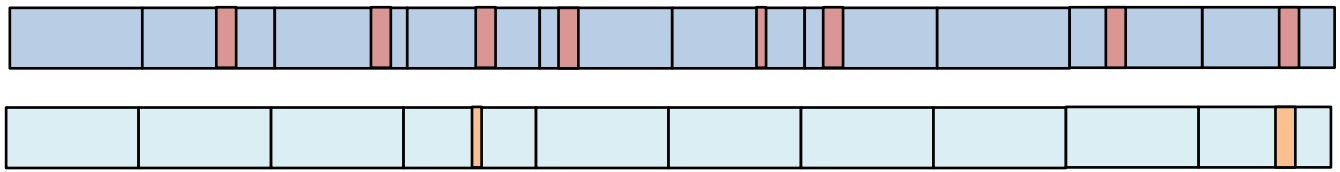
Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Evaluation of adaptive Attacks

- **Adaptive attacks**, evade by **introducing worm in every micromodel**
  - accuracy decreases to 30-40%

- Solution: **share** models **across** different sites

$$M_{\text{cross}} = M_{\text{san}} - \bigcup \{M_{\text{abn}_i} \cap M_{\text{san}}\}$$





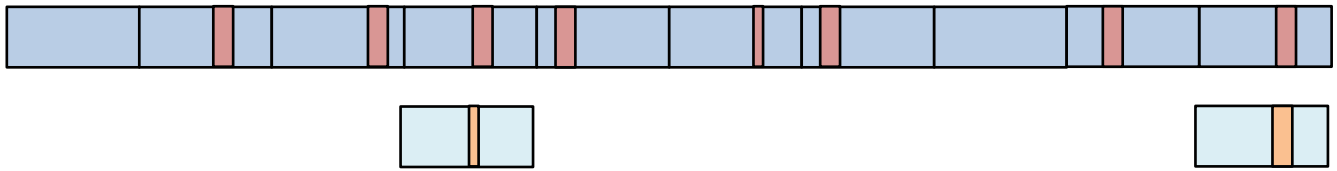
Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Evaluation of adaptive Attacks

- **Adaptive attacks**, evade by **introducing worm in every micromodel**
  - accuracy decreases to 30-40%

- Solution: **share** models **across** different sites

$$M_{\text{cross}} = M_{\text{san}} - \bigcup \{M_{\text{abn}_i} \cap M_{\text{san}}\}$$





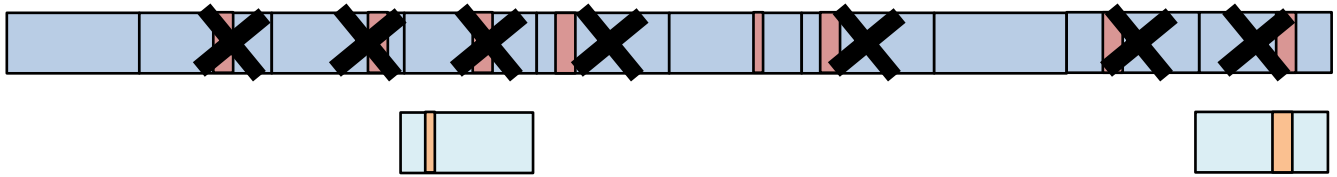
Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Evaluation of adaptive Attacks

- **Adaptive attacks**, evade by **introducing worm in every micromodel**
  - accuracy decreases to 30-40%

- Solution: **share** models **across** different sites

$$M_{\text{cross}} = M_{\text{san}} - \bigcup \{M_{\text{abn}_i} \cap M_{\text{san}}\}$$





Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Evaluation of advanced Attacks

- **Adaptive attacks**, evade by **introducing worm in every micromodel**
  - accuracy decreases to 30-40%

- Solution: **share** models **across** different sites

| Model | www1 | | www | | lists | |
|---|---|---|---|---|---|---|
| | FP(%) | DR(%) | FP(%) | DR(%) | FP(%) | DR(%) |
| $M_{pois}$ | 0.10 | 44.94 | 0.27 | 51.78 | 0.25 | 47.53 |
| $M_{cross}$ **(direct)** | **0.24** | **100** | **0.71** | **100** | **0.48** | **100** |
| $M_{cross}$ **(indirect)** | **0.10** | **100** | **0.26** | **100** | **0.10** | **100** |



Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Take Aways

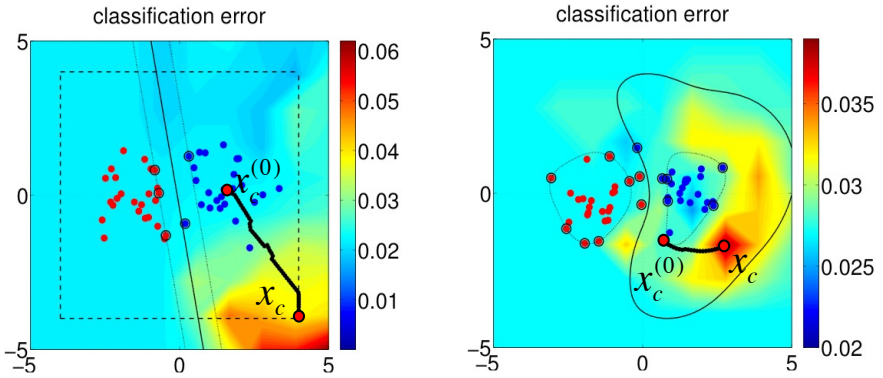
- Use knowledge from application
- Test adaptive attacks
- If possible, alter model to encompass attacks



Cretu et al.: Casting out demons: Sanitizing … IEEE SP, 2008.

# Countering Poisoning Attacks - Outline

- Data Sanitization Defenses
  - Micro models
  - **Reject on negative impact**
- Robust Optimization
  - Bagging
  - PCA GRID
  - TRIM
- Beyond Defenses
  - Regularization and Poisoning
  - Adaptive Attacks
- Conclusion



# Defense Summary – RONI

- Task:
  - Spam classification
- Algorithm:
  - Naive Bayes classifier
- Attack Assumptions:
  - Attacker's emails are always classified as spam, never as ham
  - Attacker can modify body of mail, but not header



Nelson et al.: Misleading learner… AISec, 2011.

# Recap: Naive Bayes for Spam Classification

- Derived from **Bayes theorem**

$$p(H|D) = \frac{p(H)p(D|H)}{p(D)}$$



Nelson et al.: Misleading learner… AISec, 2011.

# Recap: Naive Bayes for Spam Classification

- Derived from **Bayes theorem**
- In **practice**, instead of probabilities, use **word counts**:

$$P_S(w) = \frac{N_H N_S(w)}{N_H N_S(w) + N_S N_H(w)}$$

- *Set **threshold** to define ham, spam, and unknown*

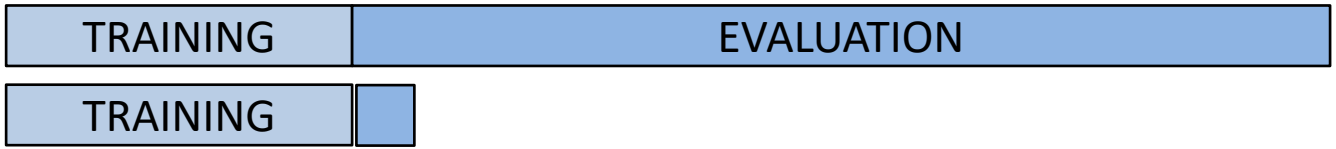


Nelson et al.: Misleading learner… AISec, 2011.

# Defense Strategy

- Reject On Negative Impact - RONI

- Training set of 20 mails, 5x50 in validation set.
- Train on training set plus a mail from validation set, **see if performance changes**






Nelson et al.: Misleading learner… AISec, 2011.

# Defense Strategy

- Reject On Negative Impact - RONI

- Training set of 20 mails, 5x50 in validation set.
- Train on training set plus a mail from validation set, **see if performance changes**

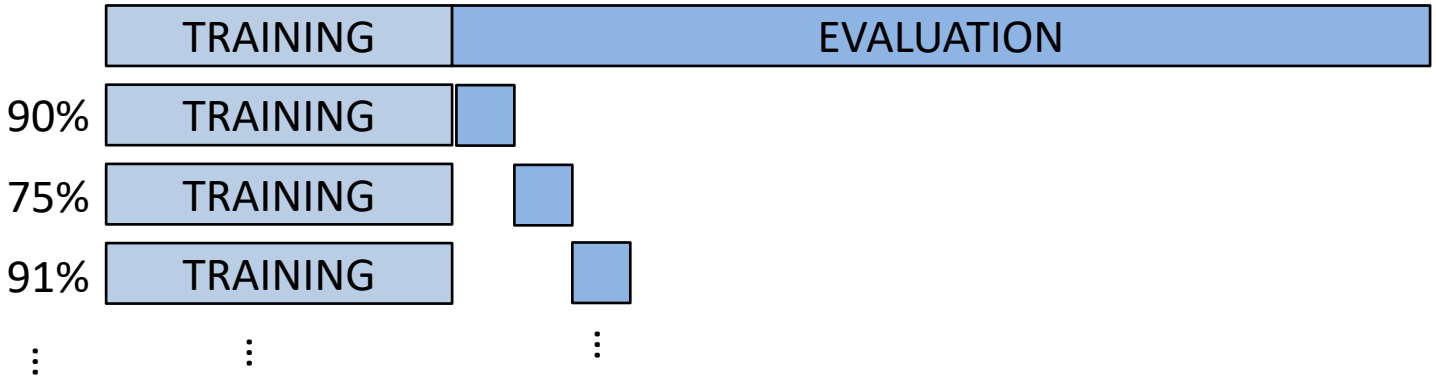



Nelson et al.: Misleading learner… AISec, 2011.

# Defense Strategy

- Reject On Negative Impact - RONI

- Training set of 20 mails, 5x50 in validation set.
- Train on training set plus a mail from validation set, **see if performance changes**






Nelson et al.: Misleading learner… AISec, 2011.

# Defense Strategy

- Reject On Negative Impact - RONI

- Training set of 20 mails, 5x50 in validation set.
- Train on training set plus a mail from validation set, **see if performance changes**

- Attack message causes on average **~50% more decrease in true positives** as normal span
  - **Separable**

- Furthermore, thresholds can/should be adapted

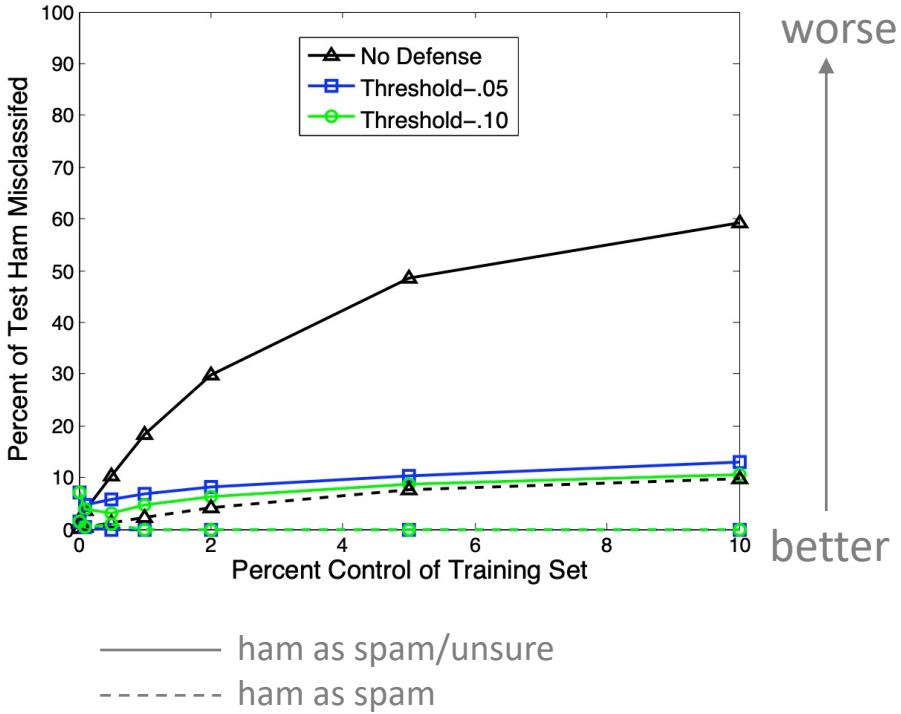


Nelson et al.: Misleading learner… AISec, 2011.

# Experiments

- 10, 000 inbox training set with 50% spam

- Problem: **correctly** identifies **ham**
- **fails** to identify **spam** (classified as unsure)






Nelson et al.: Misleading learner… AISec, 2011.

# Take Aways

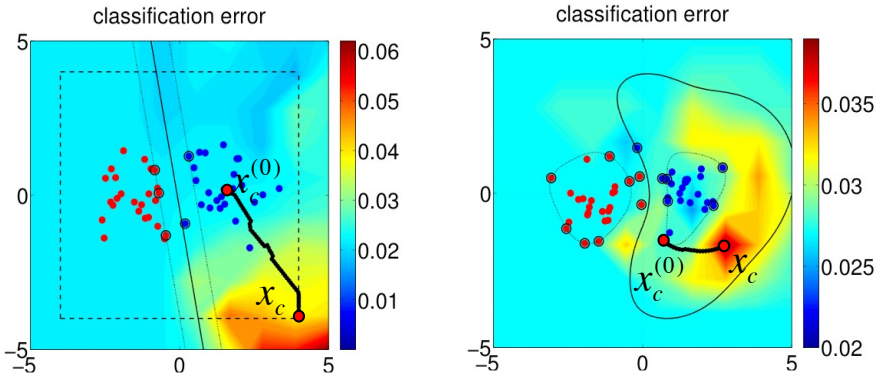
- Intuitive approach, but does not scale
- Add a whole dictionary in attack, impact has to be huge
- Dataset sanitization or robust learning?



Nelson et al.: Misleading learner… AISec, 2011.

# Countering Poisoning Attacks - Outline

- Data Sanitization Defenses
  - Micro models
  - Reject on negative impact
- Robust Optimization
  - **Bagging**
  - PCA GRID
  - TRIM
- Beyond Defenses
  - Regularization and Poisoning
  - Adaptive Attacks
- Conclusion



# Defense Summary – Bagging

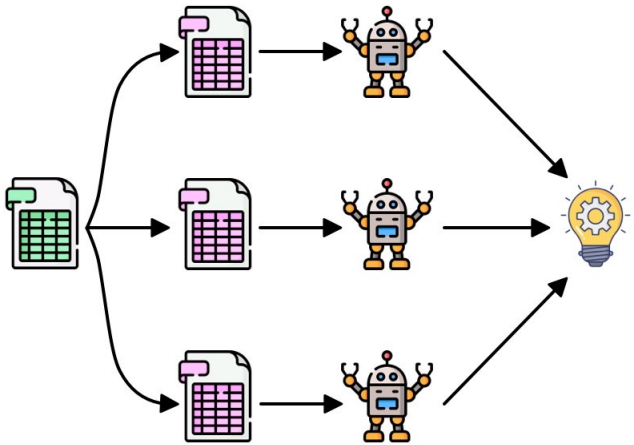
- Task:
  - Spam classification(, intrusion detection)
- Algorithm:
  - (weighted) Bagging
- Attack Assumptions:
  - Attacker controls subset of samples
  - Attacker changes features
- Further details:



Biggio et al., *Bagging classifiers: ..., intl. Workshop on mult. clf systems*, 2011

# Bagging and weighted bagging

- Used for classifiers with high variance, e.g. large variation if data slightly different
- Bagging (e.g. bootstrap different datasets and then aggregate classifiers of each)



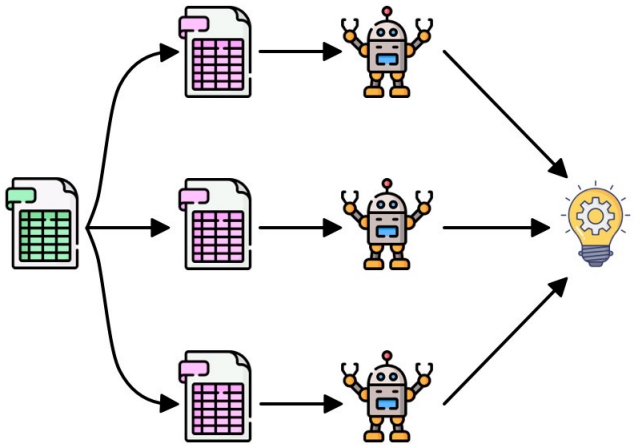


Biggio et al., *Bagging classifiers: ..., intl. Workshop on mult. clf systems*, 2011

# Bagging and weighted bagging

- Used for classifiers with high variance, e.g. large variation if data slightly different
- Bagging (e.g. bootstrap different datasets and then aggregate classifiers of each)

- Weighted bagging:
  - Density estimation on data
  - Weights are based on density:
    high density -> high weight
    low density -> low weight



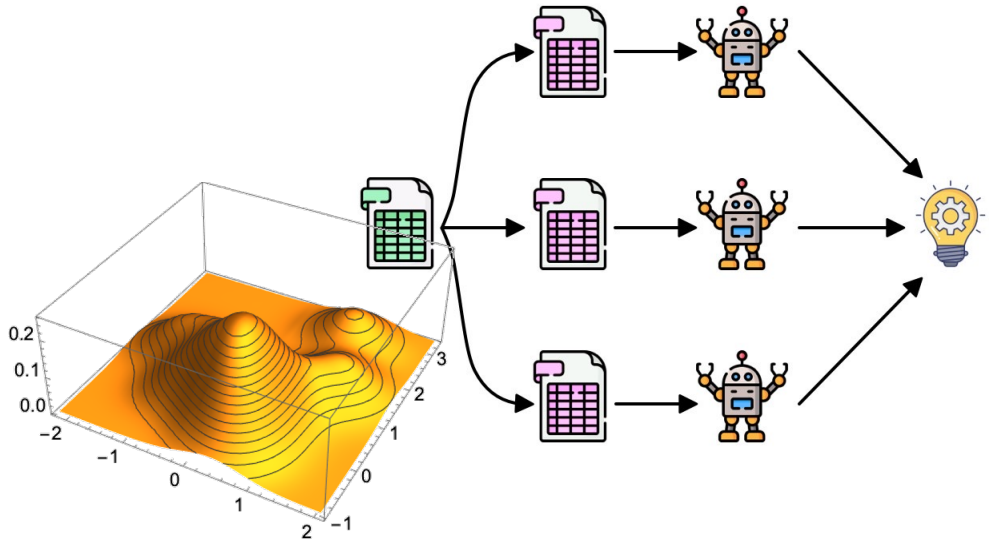


Biggio et al., *Bagging classifiers: ..., intl. Workshop on mult. clf systems*, 2011

# Bagging and weighted bagging

- Used for classifiers with high variance, e.g. large variation if data slightly different
- Bagging (e.g. bootstrap different datasets and then aggregate classifiers of each)

- Weighted bagging:
  - Density estimation on data
  - Weights are based on density:
    high density -> high weight
    low density -> low weight

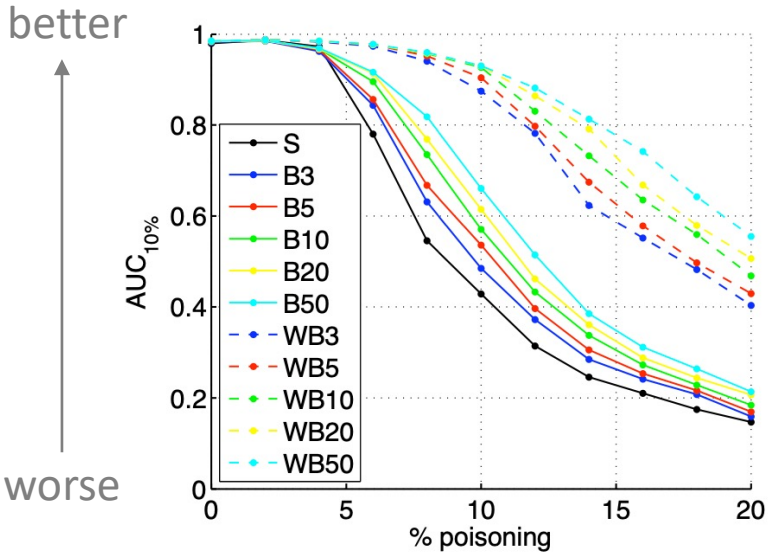





Biggio et al., *Bagging classifiers: ..., intl. Workshop on mult. clf systems*, 2011

# Experiments on Spam

- TREC corpus, 25,220 legitimate and 50,199 spam emails
- 20,000 tokens, represented as binary






Biggio et al., *Bagging classifiers: ..., intl. Workshop on mult. clf systems*, 2011

# Take Aways

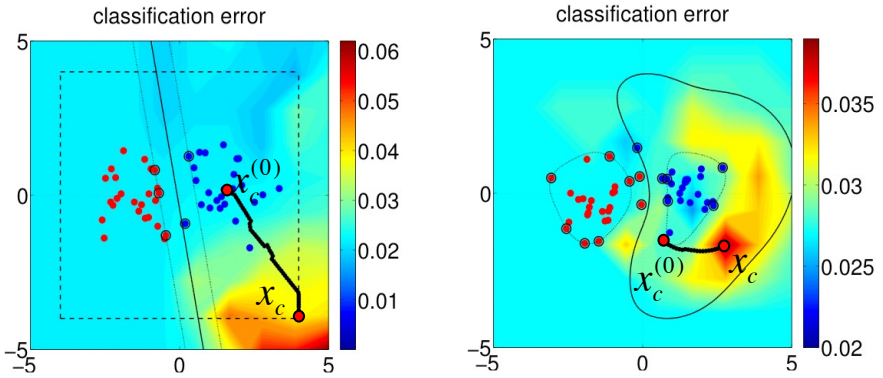
- RONI used random split, but here weighted bagging provides improvement
- Depends on kernel density estimation
- Depends on parameter which determines amount of classifiers



Biggio et al., *Bagging classifiers: ..., intl. Workshop on mult. clf systems*, 2011

# Countering Poisoning Attacks - Outline

- Data Sanitization Defenses
  - Micro models
  - Reject on negative impact
- Robust Optimization
  - Bagging
  - **PCA GRID**
  - TRIM
- Beyond Defenses
  - Regularization and Poisoning
  - Adaptive Attacks
- Conclusion



# Defense Summary – PCA GRID

- Task:
  - Anomaly detection in backbone networks
- Algorithm:
  - PCA
- Attack Assumptions:
  - Attacker controls subset of samples
  - Attacker can only add traffic, not remove or delay
- Further details:
  - Study different attacks, White-box to black box, boiling frog

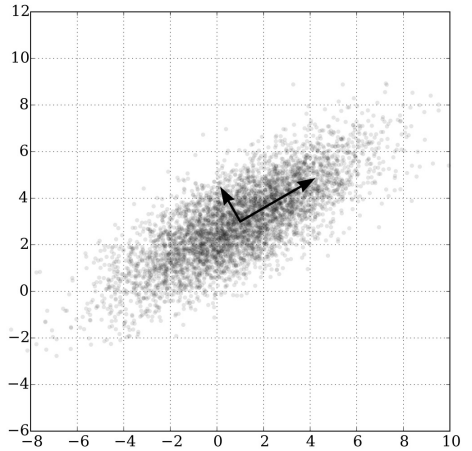


Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# PCA and PCA-GRID

- PCA highly vulnerable to outliers:

- Based on mean, variance

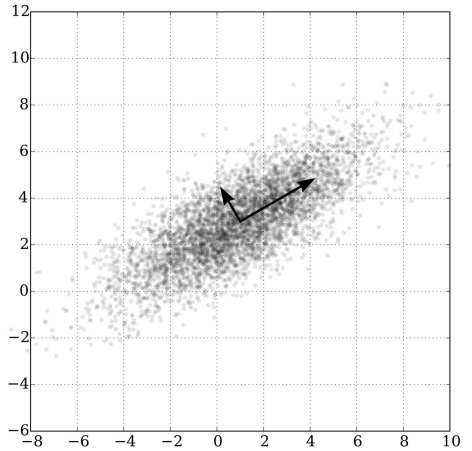





Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# PCA and PCA-GRID

- PCA highly vulnerable to outliers
- Use instead Median Absolute Deviation (MAD)
- Use also median to center data

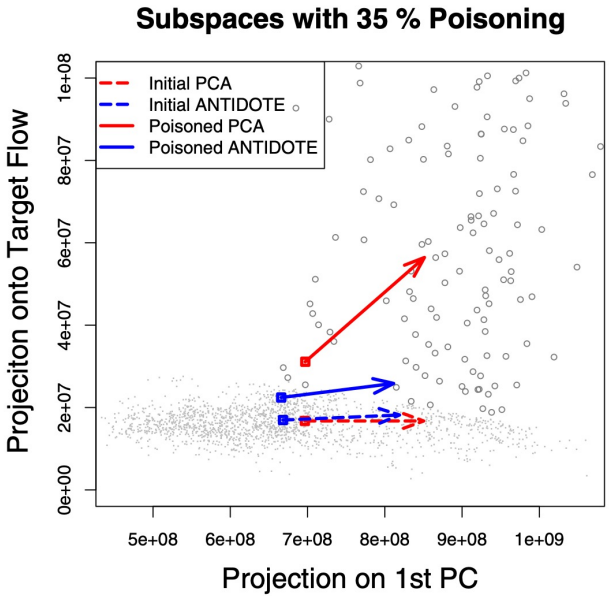





Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# PCA and PCA-GRID

- PCA highly vulnerable to outliers
- Use instead Median Absolute Deviation (MAD)
- Use also median to center data






Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# PCA residuals

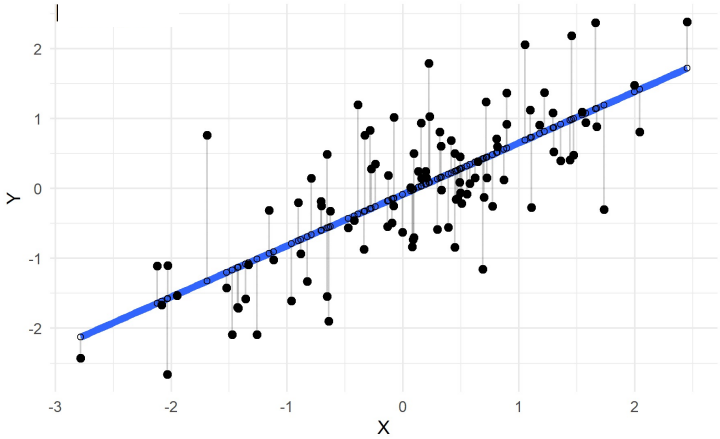
- To detect malicious data, compose into normal and residual traffic

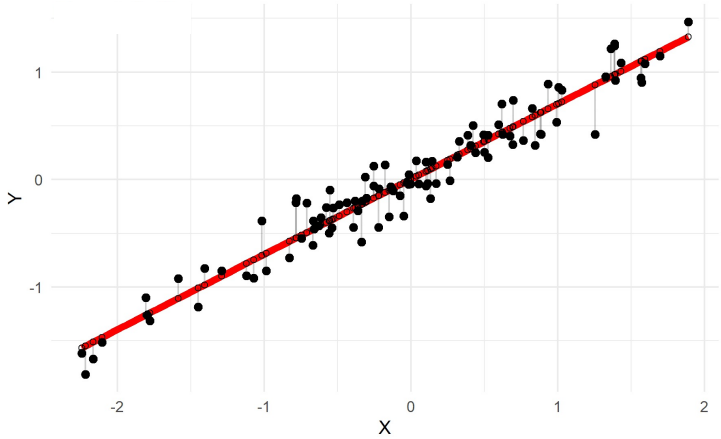


Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# PCA residuals

- To detect malicious data, compose into normal and residual traffic



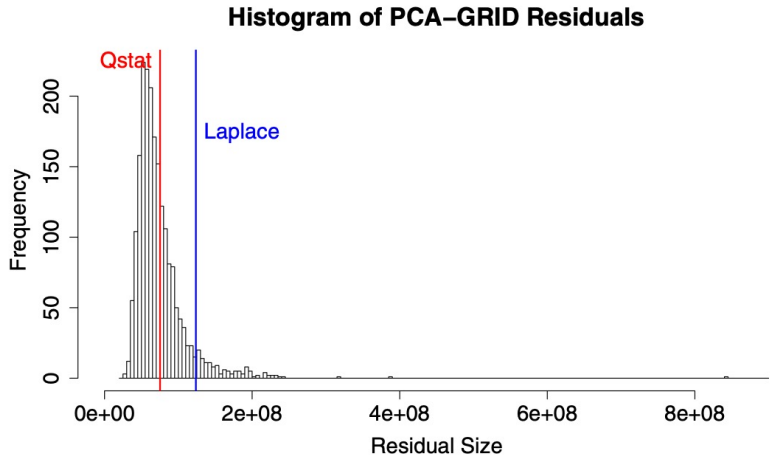

- Small residuals: usual data, model fits well
- Large residuals: data does not fit well (or variance in data is large)



Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# PCA residuals

- To detect malicious data, compose into normal and residual traffic
- Previous approaches cannot deal with heavy tailed data
- Chose Laplace cut-offs instead





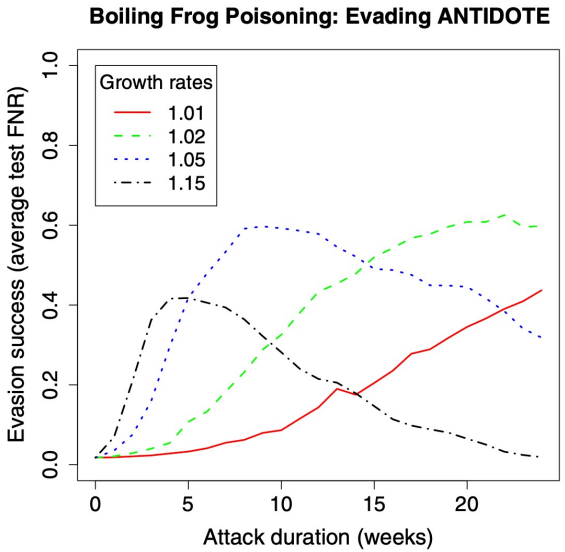

Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# Experiments

- 6-month period from March 1 through September 10, 2004.
- Each week of data consists of 2016 measurements across 144 networks

- Evaluated on synthetic anomalies

worse




better

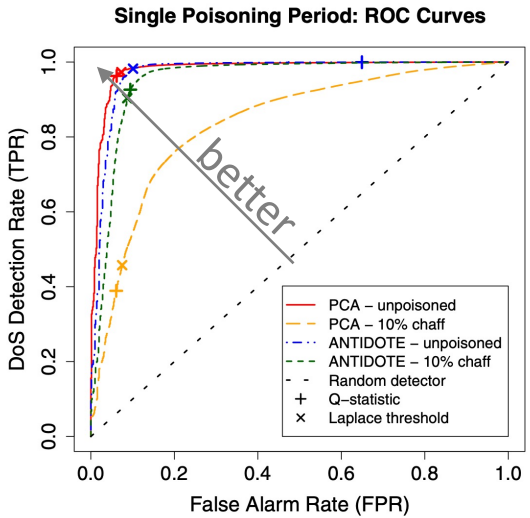


Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# Experiments

- 6-month period from March 1 through September 10, 2004.
- Each week of data consists of 2016 measurements across 144 networks

- Evaluated on synthetic anomalies






Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# Take Aways

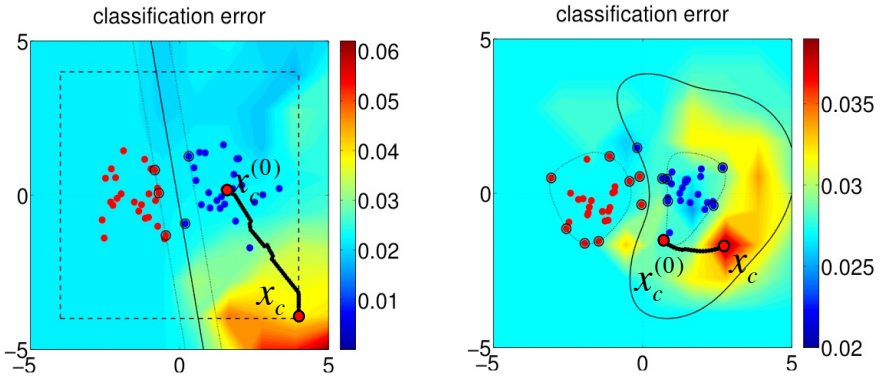
- Replace vulnerable components of algorithm
- Many problems from AML in general have already existing solutions in ML



Rubinstein et al., *ANITODE: ..., ACM Conf. Internet Measurement*, 2009

# Countering Poisoning Attacks - Outline

- Data Sanitization Defenses
  - Micro models
  - Reject on negative impact
- Robust Optimization
  - Bagging
  - PCA GRID
  - **TRIM**
- Beyond Defenses
  - Regularization and Poisoning
  - Adaptive Attacks
- Conclusion



# Defense Summary – TRIM

- Task:
  - Loan, health care, and house price data
- Algorithm:
  - Linear regression
- Attack Assumptions:
  - Attacker controls subset of samples
  - Attacker changes features and labels



Jagielski, Biggio et al., *Manipulating Machine Learning: ...,* IEEE SP, 2018

# Approach

- Consider optimization problem:

$$\underset{w,b,I}{\operatorname{argmin}} L(w, b, I) = \frac{1}{|I|} \sum_{i \in I} (f(\boldsymbol{x}_i) - y_i)^2 + \lambda \Omega(\boldsymbol{w})$$

$$N = (1 + \alpha)n, \qquad I \subset [1, \dots, N], \qquad |I| = n$$



Jagielski, Biggio et al., *Manipulating Machine Learning: ...,* IEEE SP, 2018

# Approach

- Consider optimization problem:

$$\underset{w,b,I}{\operatorname{argmin}} L(w,b,I) = \frac{1}{|I|}\sum_{i\in I}(f(\boldsymbol{x}_i) - y_i)^2 + \lambda\Omega(\boldsymbol{w})$$

$$N = (1+\alpha)n, \qquad I \subset [1,\dots,N], \qquad |I| = n$$

- Empirical risk minimization



Jagielski, Biggio et al., *Manipulating Machine Learning: ...,* IEEE SP, 2018

# Approach

- Consider optimization problem:

$$\underset{w,b,I}{\operatorname{argmin}} L(w,b,I) = \frac{1}{|I|}\sum_{i\in I}(f(\boldsymbol{x}_i) - y_i)^2 + \lambda\Omega(\boldsymbol{w})$$

$$N = (1+\alpha)n, \qquad I \subset [1,\dots,N], \qquad |I| = n$$

- Empirical risk minimization + regularization



Jagielski, Biggio et al., *Manipulating Machine Learning: ...*, IEEE SP, 2018

# Approach

- Consider optimization problem:

$$\underset{w,b,I}{\operatorname{argmin}} L(w,b,I) = \frac{1}{|I|}\sum_{i\in I}(f(\boldsymbol{x}_i) - y_i)^2 + \lambda\Omega(\boldsymbol{w})$$

$$N = (1+\alpha)n, \qquad I \subset [1,\dots,N], \qquad |I| = n$$

- Encompasses which points are used for optimization



Jagielski, Biggio et al., *Manipulating Machine Learning: ...*, IEEE SP, 2018

# Approach

- Consider optimization problem:

$$\underset{w,b,I}{\operatorname{argmin}} L(w,b,I) = \frac{1}{|I|}\sum_{i\in I}(f(\boldsymbol{x}_i) - y_i)^2 + \lambda\Omega(\boldsymbol{w})$$

$$N = (1+\alpha)n, \qquad I \subset [1,\dots,N], \qquad |I| = n$$

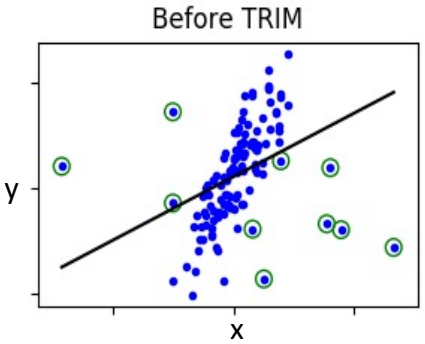
- Difficult to solve, so iterative solution
- Empirically shown that converges. Consider that no assumptions on how attack is generated (just on amount of poisoned samples)



Jagielski, Biggio et al., *Manipulating Machine Learning: ...,* IEEE SP, 2018

# TBD

- High Level intuition of algorithm

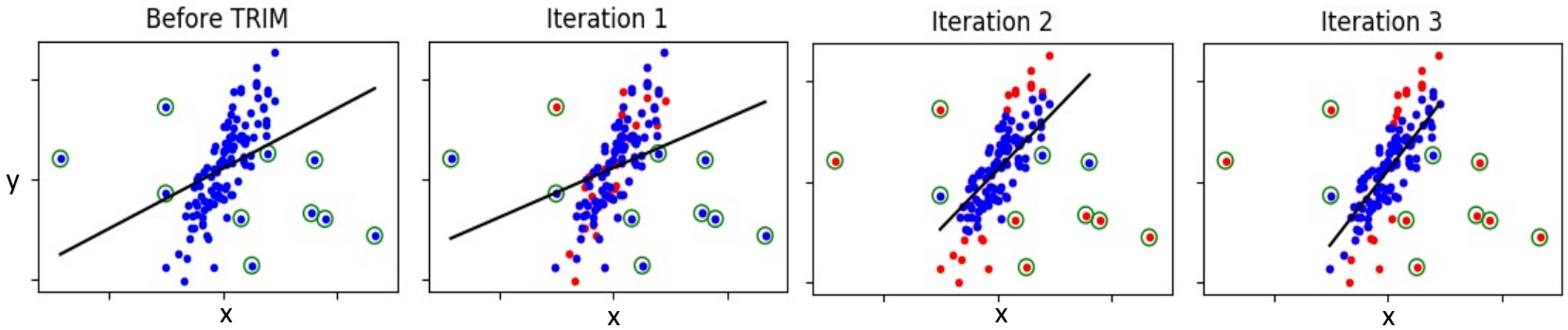





Jagielski, Biggio et al., *Manipulating Machine Learning: ...*, IEEE SP, 2018

# TBD

- High Level intuition of algorithm

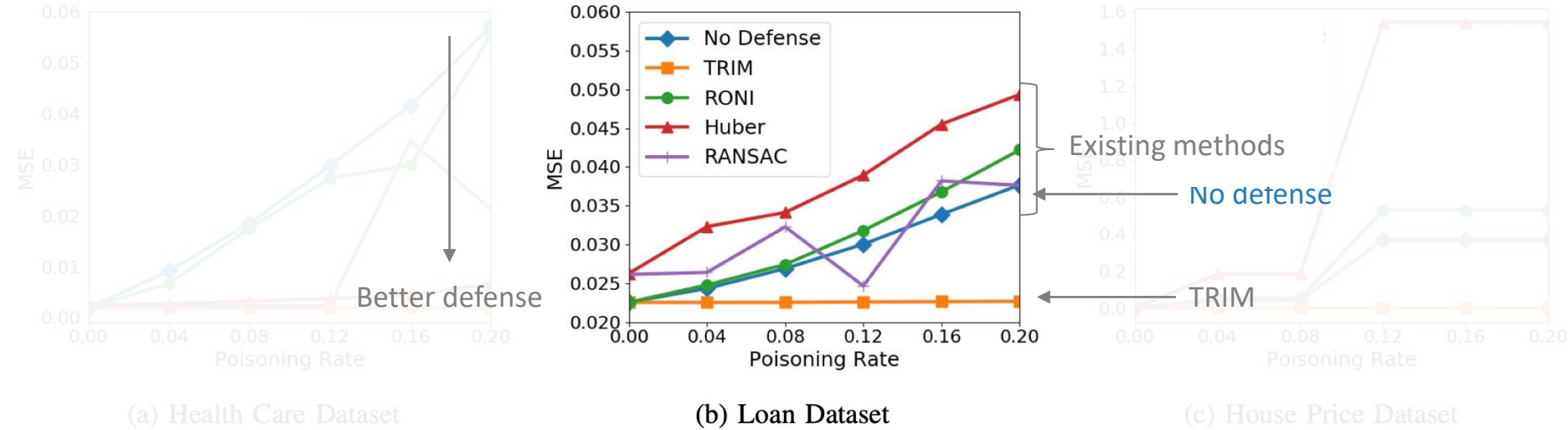





Jagielski, Biggio et al., *Manipulating Machine Learning: ...*, IEEE SP, 2018

# Evaluation

- 3 datasets, health care, loan, and house pricing

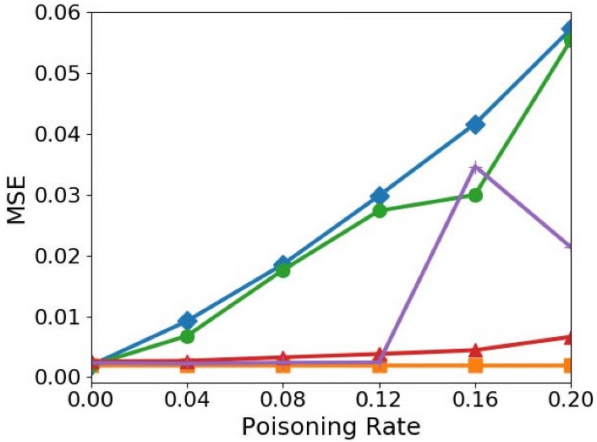




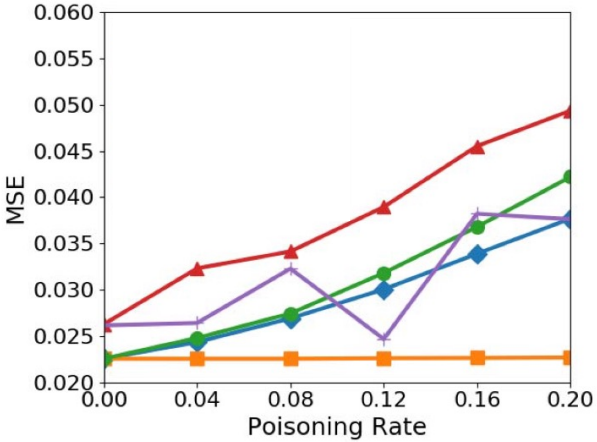

Jagielski, Biggio et al., *Manipulating Machine Learning: ...*, IEEE SP, 2018

# Evaluation

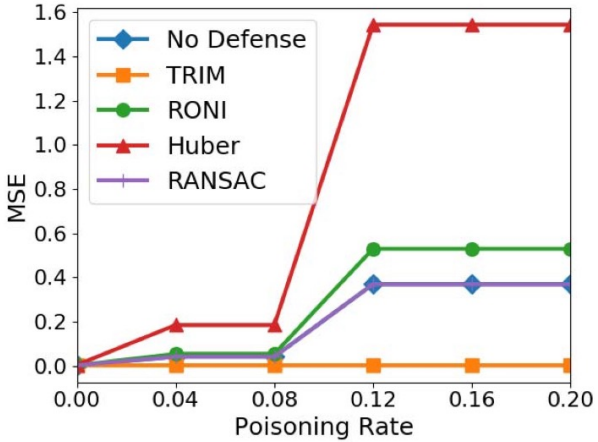
- 3 datasets, health care, loan, and house pricing




(a) Health Care Dataset




(b) Loan Dataset




(c) House Price Dataset



Jagielski, Biggio et al., *Manipulating Machine Learning: ...*, IEEE SP, 2018

# Take Aways

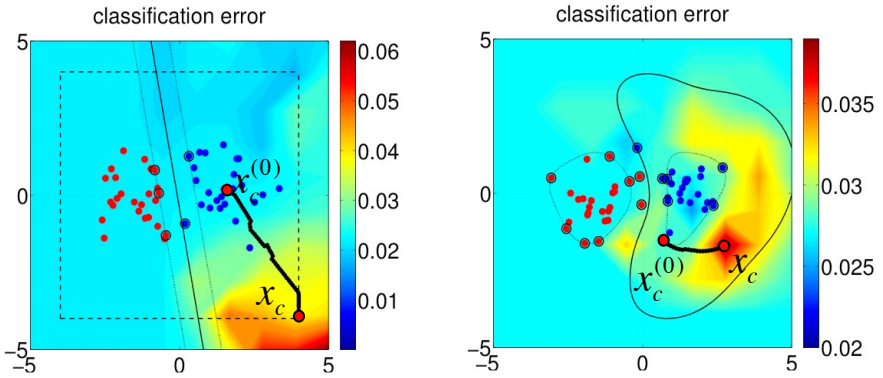
- Outlier detection is directly incorporated into optimization problem
- Compare to related work
- Amount of introduced poisoned samples must be limited



Jagielski, Biggio et al., *Manipulating Machine Learning: ...*, IEEE SP, 2018

# Countering Poisoning Attacks - Outline

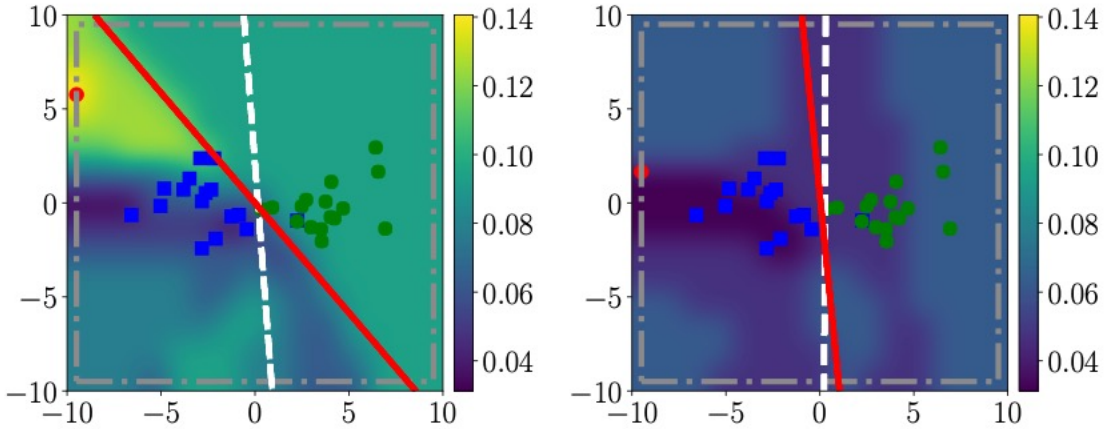
- Data Sanitization Defenses
  - Micro models
  - Reject on negative impact
- Robust Optimization
  - Bagging
  - PCA GRID
  - TRIM
- Beyond Defenses
  - **Regularization and Poisoning**
  - Adaptive Attacks
- Conclusion



# Regularization and Poisoning

- We have been talking about limiting the influence of a single point on the algorithm

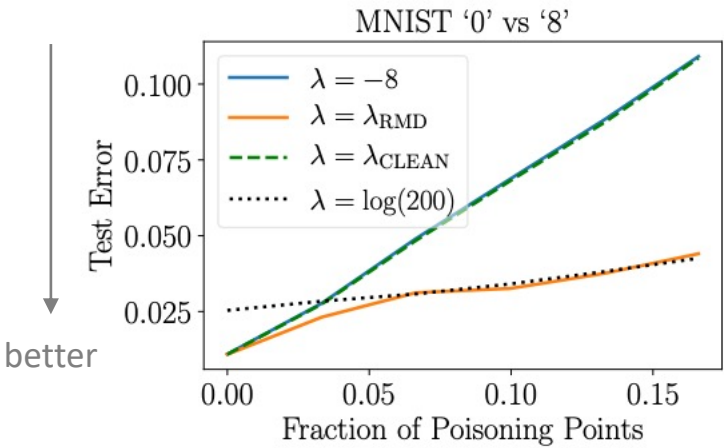
➢ Existing work on that, called regularization






Carnero-Cano et al.: *Regularization can help…* SSMLS@ICLR, 2018.

# Learning (poisoned?) hyperparameters



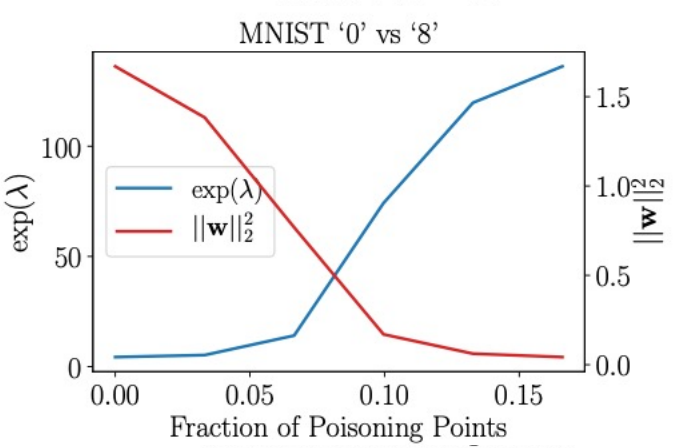

- Idea: run hyperparameter search to find good hyperparameters for regularization, also monitor weights
- When param is learned, error is lower

➢ Effect of poisoning can be alleviated with proper hyperparameters



Carnero-Cano et al.: *Regularization can help…* SSMLS@ICLR, 2018.

# Learning (poisoned?) hyperparameters II




- More poisoning points imply larger learned regularization hyperparameter
- Consequently, size of the weights get smaller



Carnero-Cano et al.: *Regularization can help…* SSMLS@ICLR, 2018.

# Take Aways

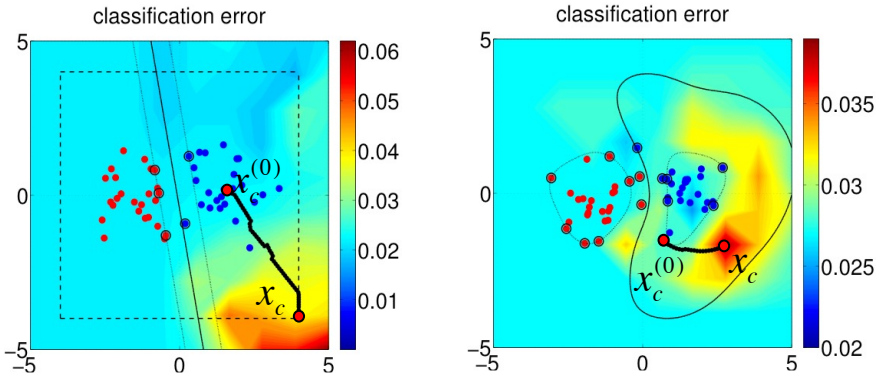
- Regularization affects how the classifier learns the data, and thus the poison
- relationship between complexity and poison vulnerability:
  - a flexible classifier learns poisons faster
  - a less flexible classifier learns poisons slower
- For many problems from security, ML solutions exist already!



Carnero-Cano et al.: *Regularization can help…* SSMLS@ICLR, 2018.

# Countering Poisoning Attacks - Outline

- Data Sanitization Defenses
  - Micro models
  - Reject on negative impact
- Robust Optimization
  - Bagging
  - PCA GRID
  - TRIM
- Beyond Defenses
  - Regularization and Poisoning
  - **Adaptive Attacks**
- Conclusion



# Evaluating Defenses

- Important to conduct ablation studies on parameters
- State assumptions / threat model clearly



Koh et al., *Stronger poisoning attacs break ..., Arxive*, 2019

# Evaluating Defenses

- Important to conduct ablation studies on parameters
- State assumptions / threat model clearly
- Most important, consider adaptive attacks…
- … knowing the defense, how would an attack target the system?



Koh et al., *Stronger poisoning attacs break ..., Arxive*, 2019

# Setting for an adaptive Attack

- Data sanitization-based defenses:
  - Outlier detector that rejects anything outside a radius (L2 defense)
  - exclude points with large loss (TRIM, not tested)
  - … more in the paper

- Assumptions:
  - defense is applied without human intervention
  - Attacker can add 3%, remove 5%

- The paper introduces several attacks, we focus on one.



Koh et al., *Stronger poisoning attacs break ..., Arxive*, 2019

# Basic Minimax Attack

- To approximate test loss, training data is used
- Essentially solve a saddle point problem

$$\underset{\mathcal{D}_\mathrm{p} \subseteq \mathcal{F}_\beta}{\text{maximize}} \quad L(\theta; \mathcal{D}_\mathrm{c} \cup \mathcal{D}_\mathrm{p})$$

$$\text{where} \quad \hat{\theta} \overset{\text{def}}{=} \underset{\theta}{\text{argmin}}\, L(\theta; \mathcal{D}_\mathrm{c} \cup \mathcal{D}_\mathrm{p})$$

- Convex if the loss is convex



Koh et al., *Stronger poisoning attacs break ..., Arxive*, 2019

# Adaptive Minimax

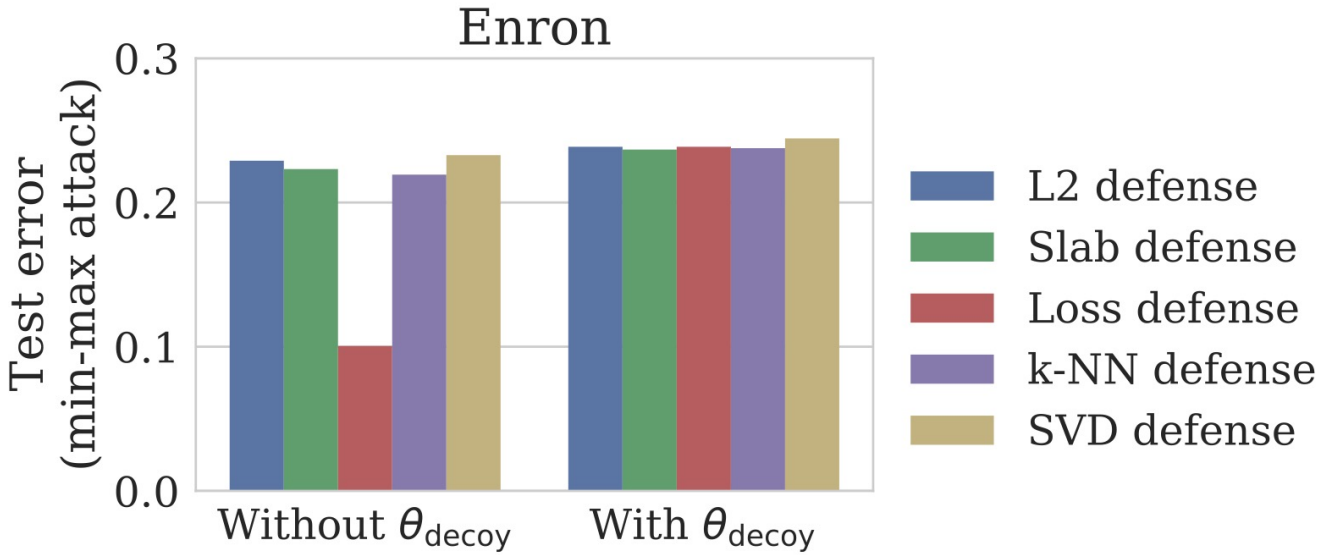
- Issues with old attack:
  - Approximated test loss might be wrong because model does not fit data well
  - High loss points are detectable by the loss defense
- Solution: add constraint where decoy parameters' loss on the poisoning data have low loss
  - Decoy paramters have high test loss, real params are driven towards them
  - Upper bounds loss value of poisoning points



Koh et al., *Stronger poisoning attacs break ..., Arxive*, 2019

# Empirical results

- Comparing new and adapted attack on the Enron dataset

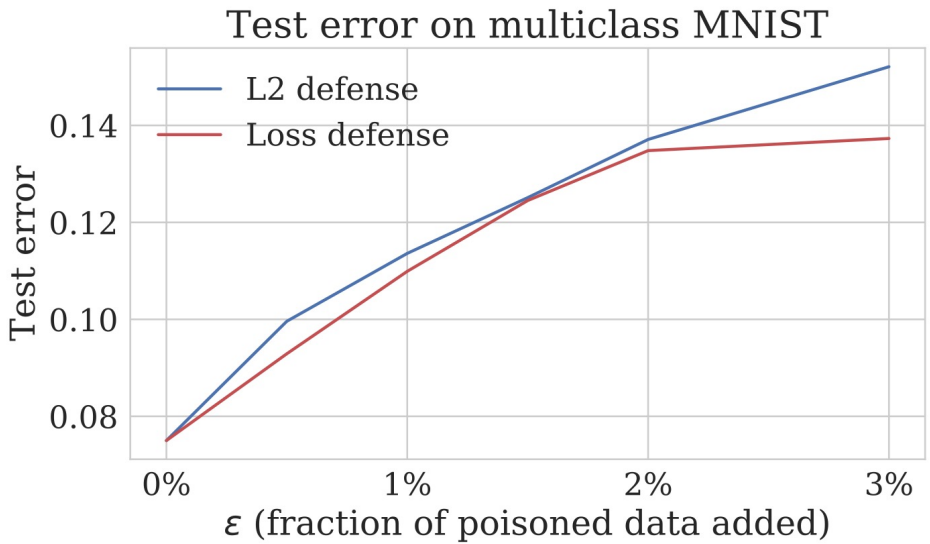





Koh et al., *Stronger poisoning attacs break ..., Arxive*, 2019

# Empirical results

- Increasing the amount of poisoning points on defenses





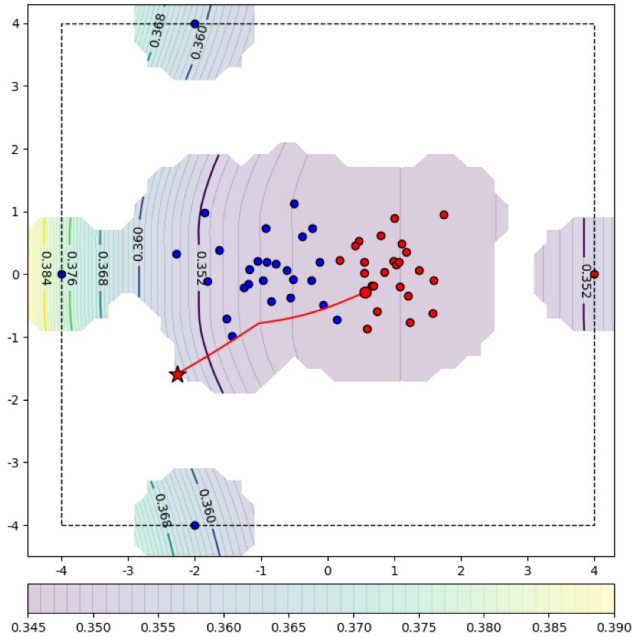

Koh et al., *Stronger poisoning attacs break ..., Arxive*, 2019

# Perturbation Constraints

- Benign attacks lead to points that are far away from the data

➢ Easy to detect

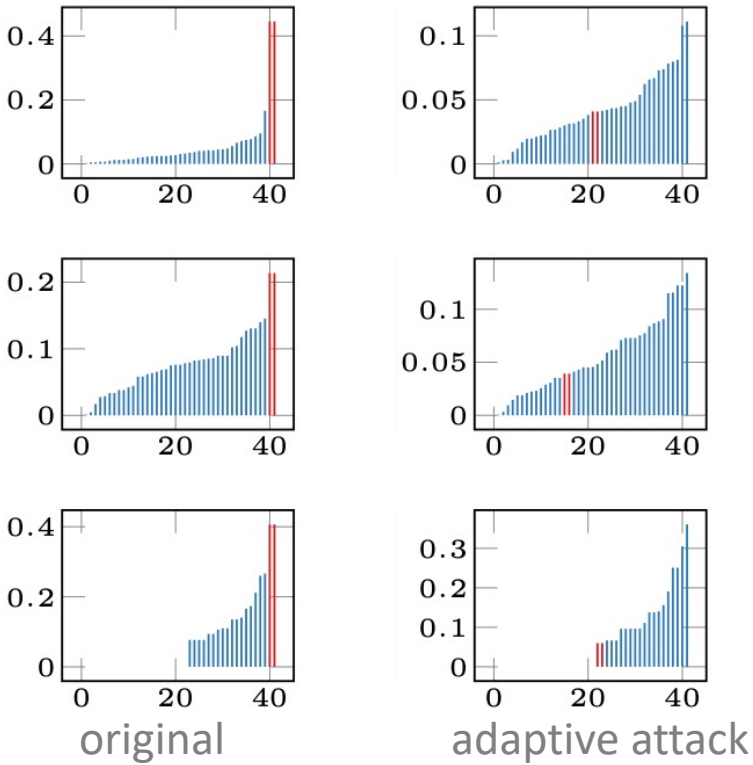
- Solution: Attacker distance threshold limits how far attacks can be maximally away from the original points






Frederickson et al.: *Attack strength vs detectability…* IJCANN, 2018.

# Perturbation Constraints in Practice



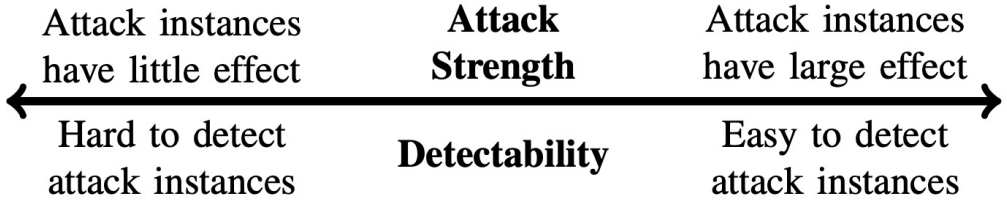

- Score based on DT trained to separate outlier points
- Three different datasets
- Benign data blue, red outlier scores



Frederickson et al.: *Attack strength vs detectability…* IJCANN, 2018.

# Take Aways

- Evaluation on adaptive attack ABSOLUTELY necessary
- High level insight:

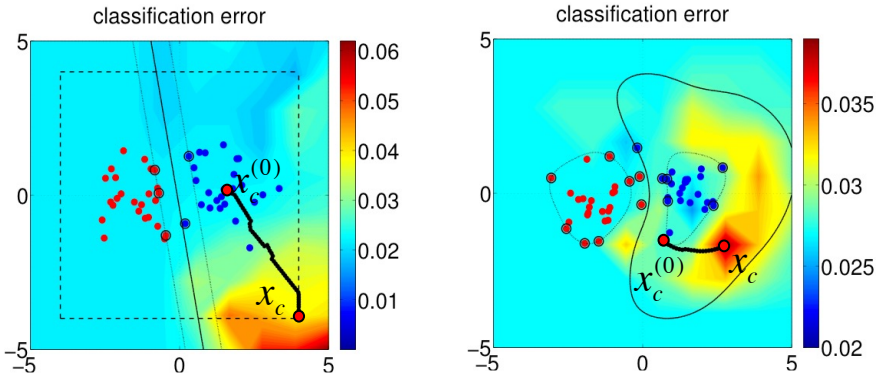





Frederickson et al.: *Attack strength vs detectability…* IJCANN, 2018.

# Countering Poisoning Attacks - Outline

- Data Sanitization Defenses
  - Micro models
  - Reject on negative impact
- Robust Optimization
  - Bagging
  - PCA GRID
  - TRIM
- Beyond Defenses
  - Regularization and Poisoning
  - Adaptive Attacks
- **Conclusion**



# Conclusion

- In contrast to evasion, hope for defenses
- Roughly two groups, data sanitization and robust optimization
- Few strongly changed (and easily detectable) points or more, less changed (and thus hard to detect) points.
- A flexible classifier learns poisoning points fast, whereas a less flexible classifier will lead to worse poisoning success

Kathrin Grosse
Kathrin.grosse@unica.it
LinkedIn Kathrin Grosse

# Thanks!


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