Pattern Recognition
and Applications Lab

Is Feature Selection Secure against Training Data Poisoning?

Huang Xiao[2], Battista Biggio[1], Gavin Brown[3], Giorgio Fumera[1], Claudia Eckert[2], Fabio Roli[1]

([1]) Dept. Of Electrical and Electronic Engineering, University of Cagliari, Italy
([2]) Department of Computer Science, Technische Universität München, Germany
([3]) School of Computer Science, University of Manchester, UK

ICML Lille

Jul 6 - 11, 2015

University of Cagliari, Italy

Department of Electrical and Electronic Engineering

Motivation

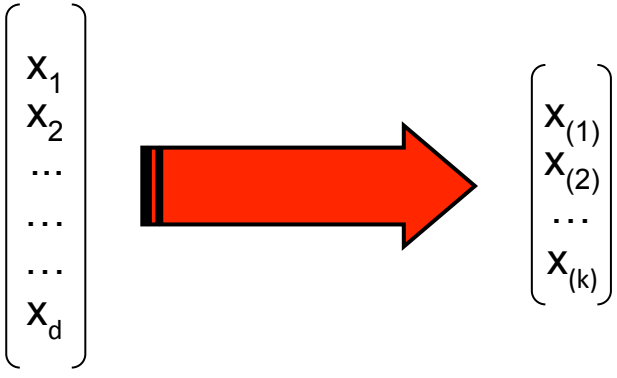
- Increasing number of services and apps available on the Internet
  - Improved user experience
- Proliferation and sophistication of attacks and cyberthreats
  - Skilled / economically-motivated attackers

- Several security systems use machine learning to detect attacks
  - but … is *machine learning* secure enough?

Is Feature Selection Secure?

- **Adversarial ML:** security of *learning* and *clustering* algorithms
  - Barreno et al., 2006; Huang et al., 2011; Biggio et al., 2014; 2012; 2013a; Brueckner et al., 2012; Globerson & Roweis, 2006

- **Feature Selection**
  - High-dimensional feature spaces (e.g., spam and malware detection)
  - Dimensionality reduction to improve interpretability and generalization

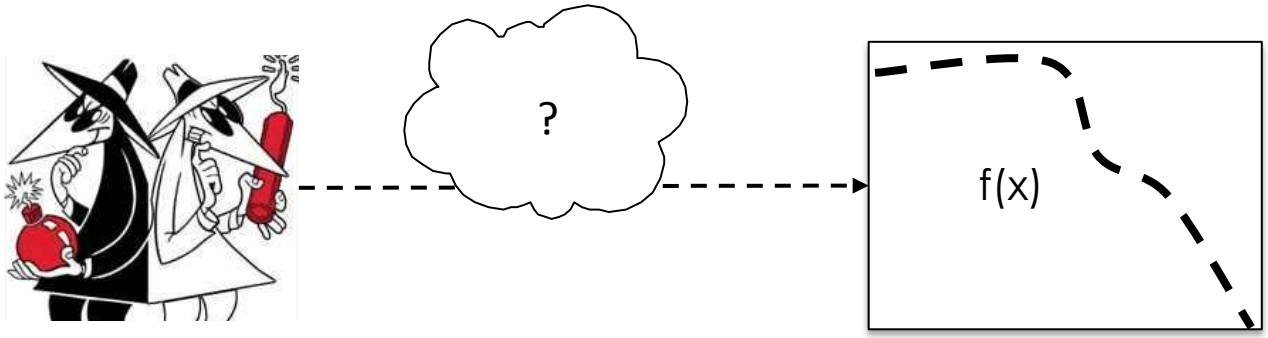


- How about the **security** of *feature selection*?

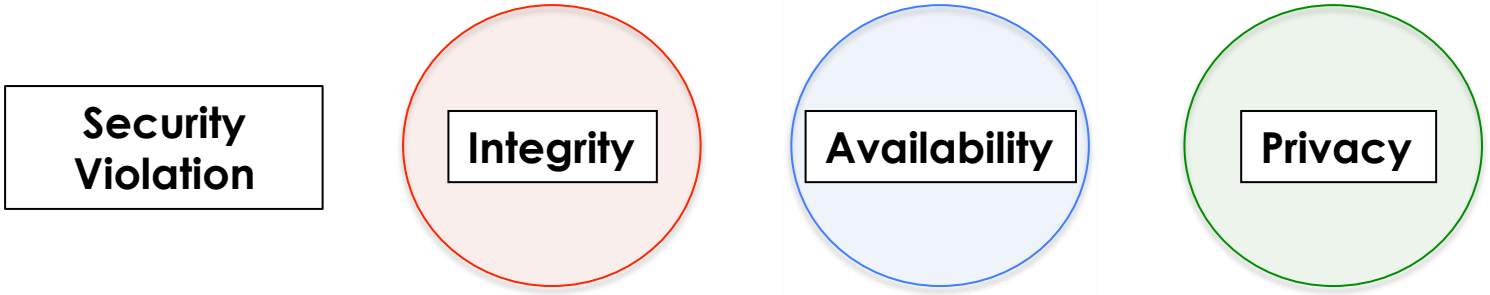
Feature Selection under Attack

Attacker Model

- **Goal** of the attack
- **Knowledge** of the attacked system
- **Capability** of manipulating data
- **Attack strategy**

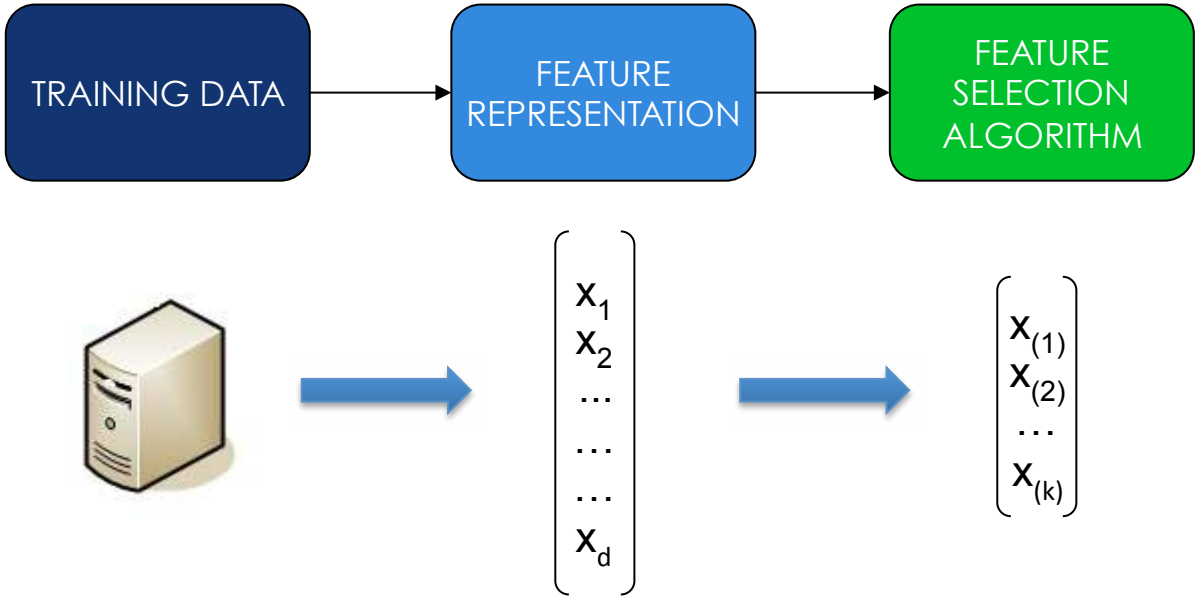


Attacker's Goal



- **Integrity Violation**: to perform malicious activities without compromising normal system operation
  - enforcing selection of features to facilitate evasion at test time
- **Availability Violation**: to compromise normal system operation
  - enforcing selection of features to maximize generalization error
- **Privacy Violation**: gaining confidential information on system users
  - reverse-engineering feature selection to get confidential information

Attacker's Knowledge



- **Perfect knowledge**
  - upper bound on performance degradation under attack
- **Limited knowledge**
  - attack on surrogate data sampled from same distribution

Attacker's Capability

- **Inject points** into the **training** data

- Constraints on data manipulation
  - Fraction of the training data under the attacker's control
  - Application-specific constraints

- **Example on PDF data**
  - PDF file: hierarchy of interconnected objects
  - Objects can be added but not easily removed without compromising the file structure

13 0 ***obj***
<< /Kids [1 0 R 11 0 R]
/Type /Page
... >> ***end obj***

17 0 ***obj***
<< /Type /Encoding
/Differences [0 /C0032] >>
endobj

Attack Scenarios

- Different potential attack scenarios depending on assumptions on the attacker's goal, knowledge, capability
  - Details and examples in the paper

- **Poisoning Availability Attacks**
  Enforcing selection of features to maximize generalization error
  - **Goal**: availability violation
  - **Knowledge**: perfect / limited
  - **Capability**: injecting samples into the training data

Embedded Feature Selection Algorithms

- **Linear models** $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$
  - Select features according to $|\mathbf{w}|$

$$\min_{\boldsymbol{w},b} \mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\ell\left(y_i, f(\boldsymbol{x}_i)\right)}_{\frac{1}{2}\left(f(\boldsymbol{x}_i) - y_i\right)^2} + \lambda\Omega(\boldsymbol{w})$$

$\|\boldsymbol{w}\|_1$	$\frac{1}{2}\|\boldsymbol{w}\|_2^2$	$\rho\|\boldsymbol{w}\|_1 + (1-\rho)\frac{1}{2}\|\boldsymbol{w}\|_2^2$
LASSO	**Ridge Regression**	**Elastic Net**
Tibshirani, 1996	Hoerl & Kennard, 1970	Zou & Hastie, 2005

Poisoning Embedded Feature Selection

- **Attacker's objective**
  - to maximize generalization error on untainted data

$$\max_{\boldsymbol{x}_c} \mathcal{W} = \frac{1}{m} \sum_{j=1}^{m} \ell\left(\hat{y}_j, f(\hat{\boldsymbol{x}}_j)\right) + \lambda \Omega(\boldsymbol{w})$$

… w.r.t. choice of the attack point

Loss estimated on surrogate data
(excluding the attack point)

$$\hat{\mathcal{D}} = \{\hat{\boldsymbol{x}}_i, \hat{y}_i\}_{i=1}^{m}$$

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

$$\mathcal{L}(\hat{\mathcal{D}} \cup \{\boldsymbol{x}_c\})$$

- **Solution:** subgradient-ascent technique

Gradient Computation

$$\frac{\partial \mathcal{W}}{\partial \boldsymbol{x}_c} = \frac{1}{m}\sum_{j=1}^{m}\left(f(\hat{\boldsymbol{x}}_j) - \hat{y}_j\right)\left(\hat{\boldsymbol{x}}_j^\top \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_c} + \frac{\partial b}{\partial \boldsymbol{x}_c}\right) + \lambda \frac{\partial \Omega}{\partial \boldsymbol{w}}\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_c}$$

How does the solution change w.r.t. x_c?

KKT conditions

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}^\top = \frac{1}{m+1}\sum_{j=1}^{m+1}\left(f(\hat{\boldsymbol{x}}_j) - \hat{y}_j\right)\hat{\boldsymbol{x}}_j + \lambda \frac{\partial \Omega}{\partial \boldsymbol{w}}^\top = \boldsymbol{0}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{m+1}\sum_{j=1}^{m+1}\left(f(\hat{\boldsymbol{x}}_j) - \hat{y}_j\right) = 0$$

Subgradient is unique at the optimal solution!

$$\frac{\partial \Omega}{\partial \boldsymbol{w}} = -\frac{1}{\lambda}\frac{1}{m+1}\sum_{j=1}^{m+1}\left(f(\hat{\boldsymbol{x}}_j) - \hat{y}_j\right)\hat{\boldsymbol{x}}_i^\top$$

Gradient Computation

- We require the KKT conditions to hold under perturbation of $\mathbf{x}_c$

$$\begin{bmatrix} \boldsymbol{\Sigma} + \lambda \boldsymbol{v} & \boldsymbol{\mu} \\ \boldsymbol{\mu}^\top & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_c} \\ \frac{\partial b}{\partial \boldsymbol{x}_c} \end{bmatrix} = -\frac{1}{m+1} \begin{bmatrix} \mathbf{M} \\ \boldsymbol{w}^\top \end{bmatrix}$$

$$\frac{\partial \mathcal{W}}{\partial \boldsymbol{x}_c} = \frac{1}{m} \sum_{j=1}^{m} \left(f(\hat{\boldsymbol{x}}_j) - \hat{y}_j \right) \left(\hat{\boldsymbol{x}}_j^\top \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_c} + \frac{\partial b}{\partial \boldsymbol{x}_c} \right) + \lambda \frac{\partial \Omega}{\partial \boldsymbol{w}} \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_c}$$

Gradient is now uniquely determined

Poisoning Attack Algorithm

Algorithm 1 Poisoning Embedded Feature Selection

Input: $\hat{\mathcal{D}}$, the (surrogate) training data; $\{\boldsymbol{x}_c^{(0)}, y_c\}_{c=1}^q$, the q initial attack points with (given) labels; $\beta \in (0,1)$; and σ, ε, two small positive constants.

Output: $\{\boldsymbol{x}_c\}_{c=1}^q$, the final attack points.

$p \leftarrow 0$

repeat

  for $c = 1, \ldots, q$ **do**

    $\{\boldsymbol{w}, b\} \leftarrow$ learn the classifier on $\hat{\mathcal{D}} \cup \{\boldsymbol{x}_c^{(p)}\}_{c=1}^q$.

    Compute $\nabla \mathcal{W} = \frac{\partial \mathcal{W}(\boldsymbol{x}_c^{(p)})}{\partial \boldsymbol{x}_c}$ according to Eq. (4).

    Set $\boldsymbol{d} = \Pi_{\mathcal{B}}\left(\boldsymbol{x}_c^{(p)} + \nabla \mathcal{W}\right) - \boldsymbol{x}_c^{(p)}$ and $k \leftarrow 0$.

    repeat {line search to set the gradient step η}

      Set $\eta \leftarrow \beta^k$ and $k \leftarrow k + 1$

      $\boldsymbol{x}_c^{(p+1)} \leftarrow \boldsymbol{x}_c^{(p)} + \eta \boldsymbol{d}$

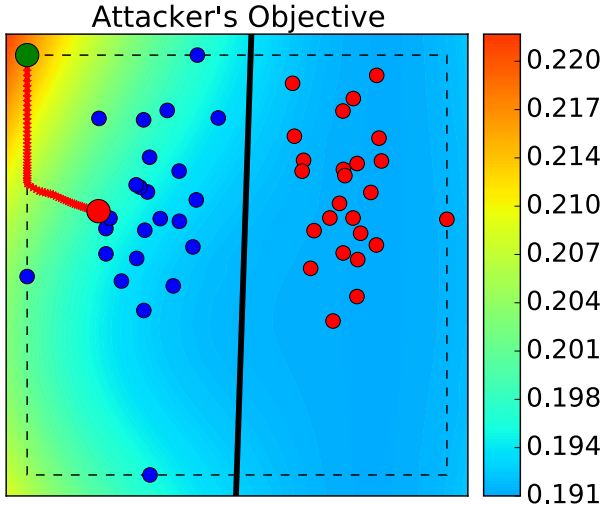
    until $\mathcal{W}(\boldsymbol{x}_c^{(p+1)}) \leq \mathcal{W}(\boldsymbol{x}_c^{(p)}) - \sigma\eta\|\boldsymbol{d}\|^2$

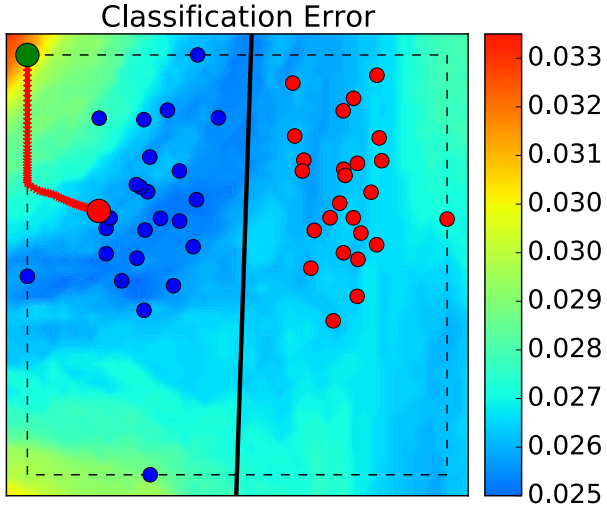
  end for

  $p \leftarrow p + 1$

until $|\mathcal{W}(\{\boldsymbol{x}_c^{(p)}\}_{c=1}^q) - \mathcal{W}(\{\boldsymbol{x}_c^{(p-1)}\}_{c=1}^q)| < \varepsilon$

return: $\{\boldsymbol{x}_c\}_{c=1}^q = \{\boldsymbol{x}_c^{(p)}\}_{c=1}^q$





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
```

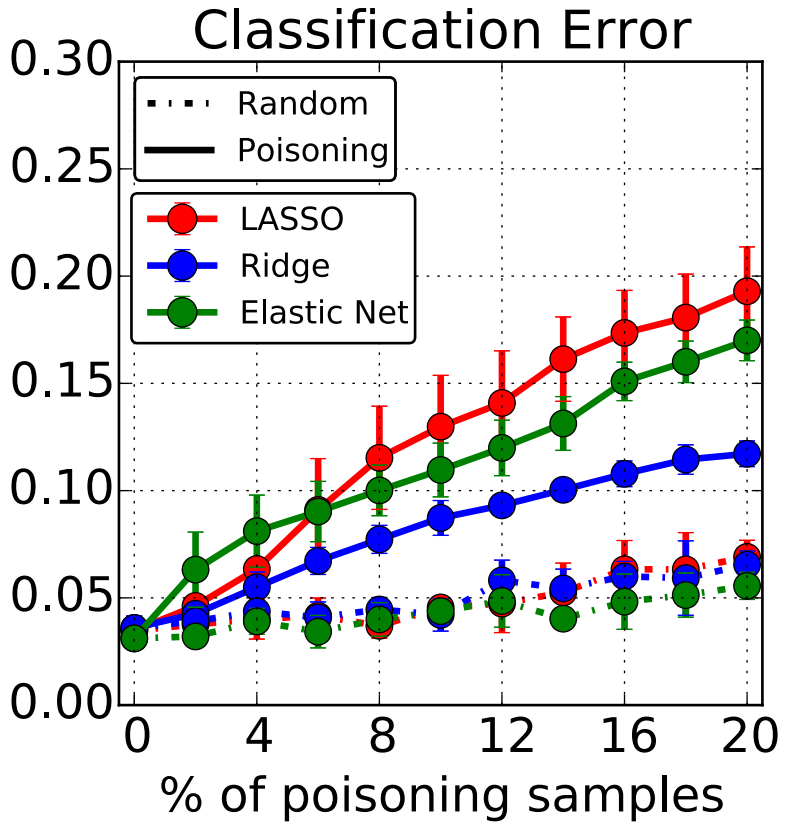
Features: *keyword counts*	
/Type	2
/Page	1
/Encoding	1
...	

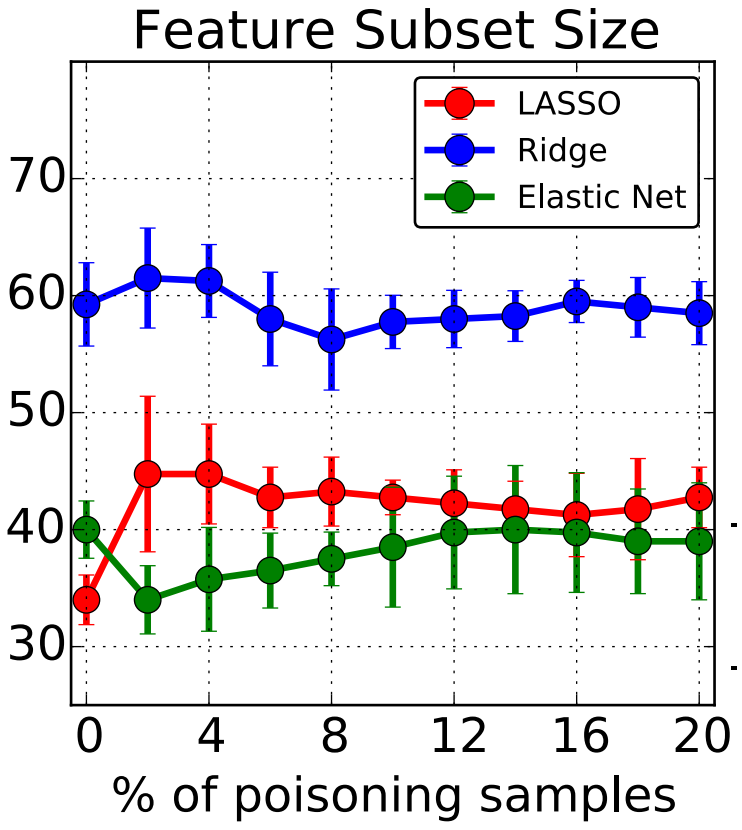
Maiorca et al., 2012; 2013;
Smutz & Stavrou, 2012;
Srndic & Laskov, 2013

- **Learner's task:** to classify *benign* vs *malware* PDF files
- **Attacker's task:** to maximize classification error by injecting poisoning attack samples
  - Only feature increments are considered (object insertion)
    - Object removal may compromise the PDF file

Experimental Results

Perfect Knowledge



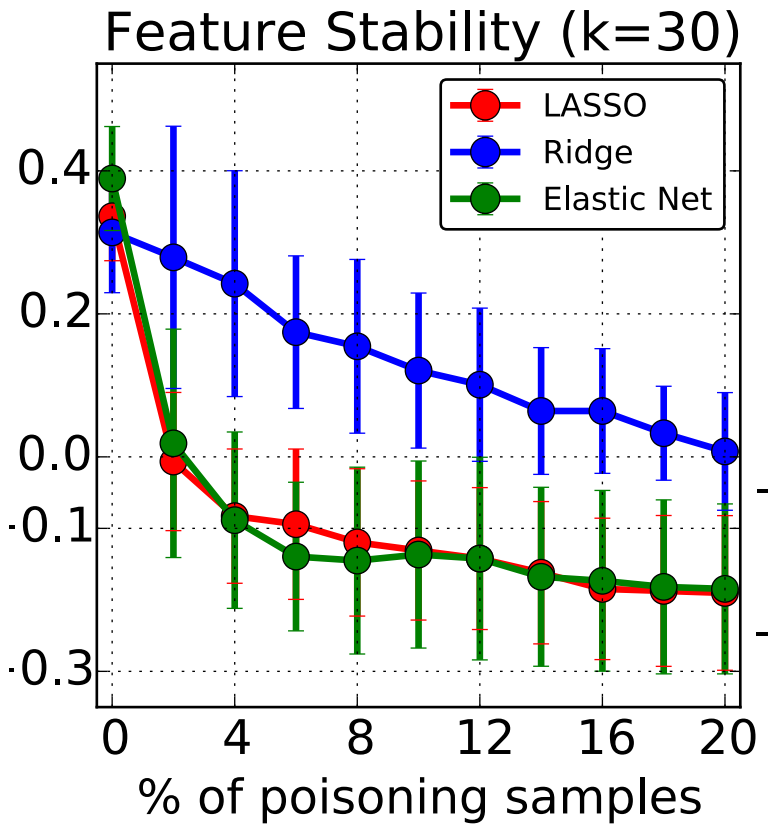


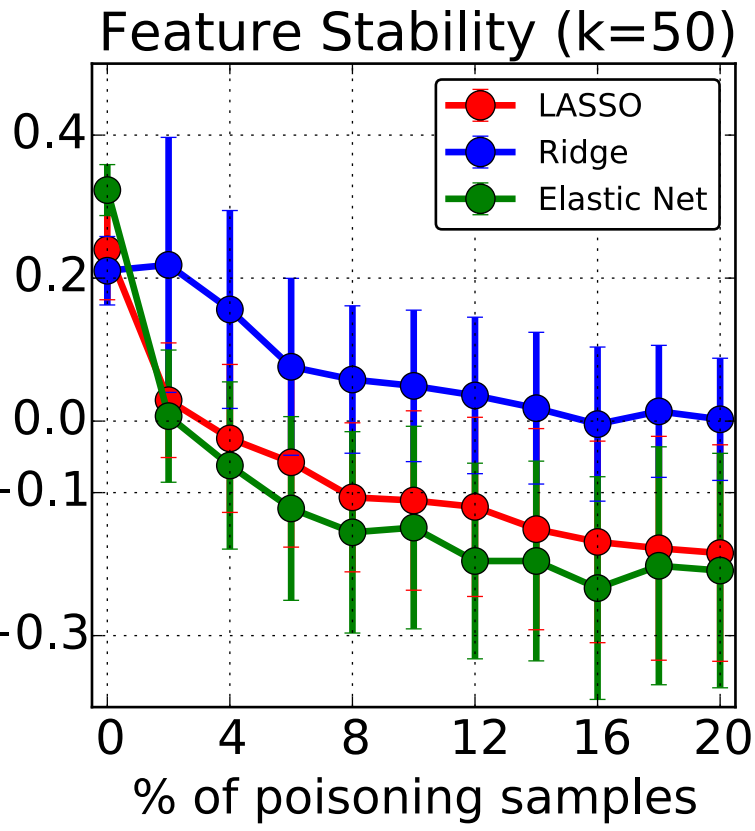
Data: 300 (TR) and 5,000 (TS) samples – 114 features

Similar results obtained for limited-knowledge attacks!

Experimental Results

Perfect Knowledge





$$I_C(A,B) = \frac{rd - k^2}{k(d-k)} \in [-1,+1]$$

Kuncheva et al., 2007

A: selected features in the absence of attack
B: selected features under attack
k: number of features selected out of **d**
r: common features between the two sets

Conclusions and Future Work

- Framework for **security evaluation** of **feature selection** under attack
  - Poisoning attacks against embedded feature selection algorithms

- Poisoning can significantly affect feature selection
  - LASSO significantly vulnerable to poisoning attacks

> **L1 regularization:** stability against random noise, but not against adversarial (worst-case) noise?

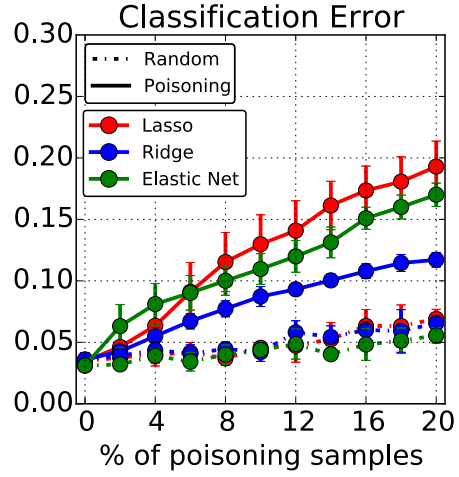
- **Future research directions**
  - Error bounds on the impact of poisoning on learning algorithms
  - Secure / robust feature selection algorithms

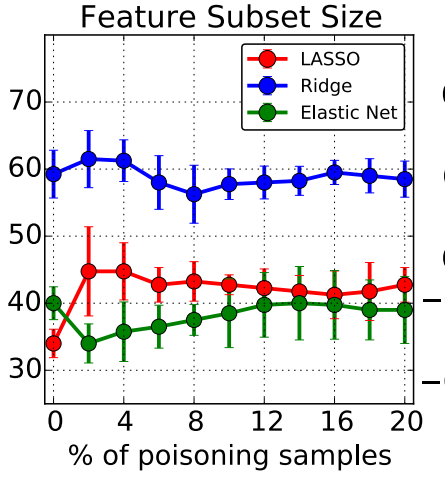
Thanks for your attention!

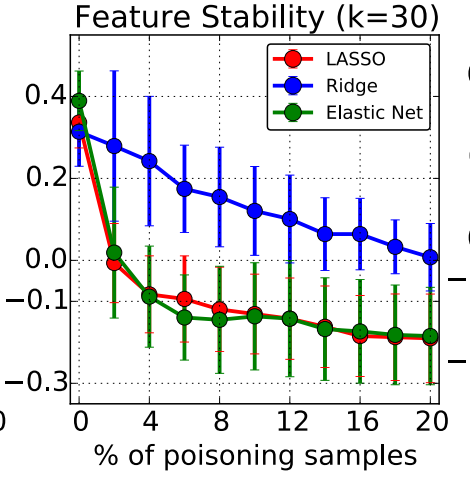
Any questions

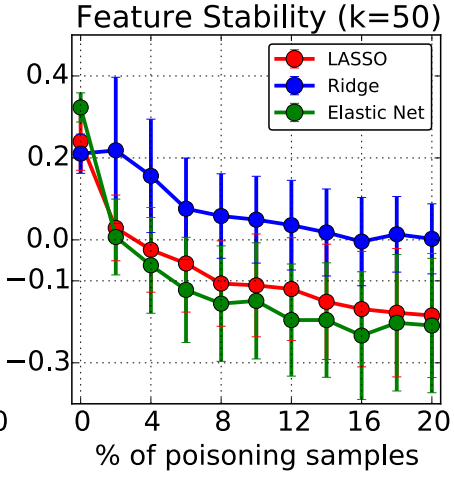
Experimental Results

Perfect Knowledge









Limited Knowledge