Adversarial ML: Bayesian Perspectives

Texas State University

Roi Naveiro

Institute of Mathematical Sciences
ICMAT-CSIC

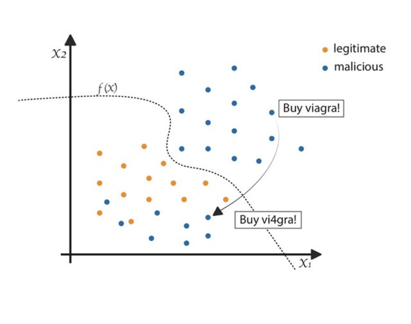
joint work with

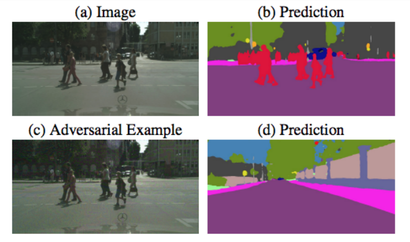
William Caballero, Tahir Ekin, Víctor Gallego,
Alberto Redondo, David Ríos Insua and Fabrizio Ruggeri

ML meets security

Central assumption in predictive inference:
Train and operation data are id

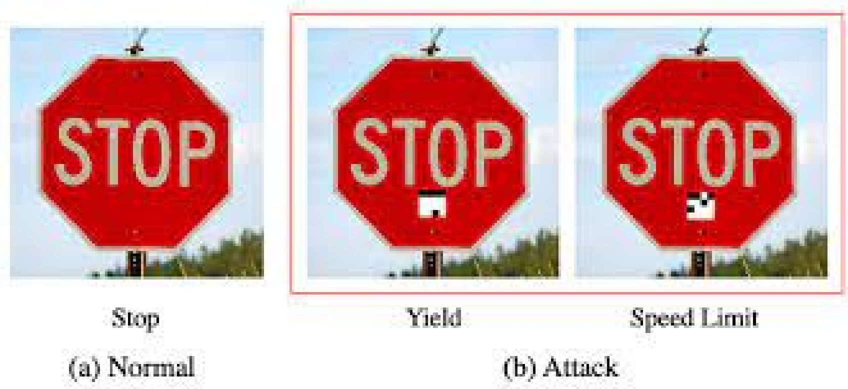
Out of the sample generalization $\neq$ Out of the distribution generalization





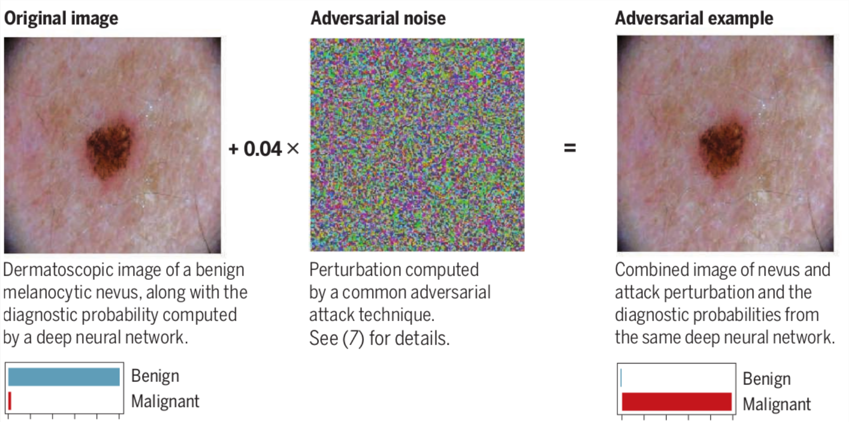
Broken by the presence of **adversaries**

ML meets security



Source: `https://portswigger.net/daily-swig/trojannet-a-simple-yet-effective-attack-on-machine-learning-models`

ML meets security



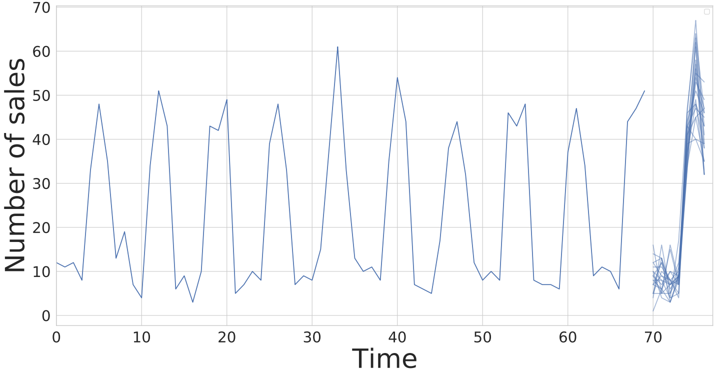
Source: `Finlaysonet.al.(2019)`

ML meets security

Not only in vision tasks!

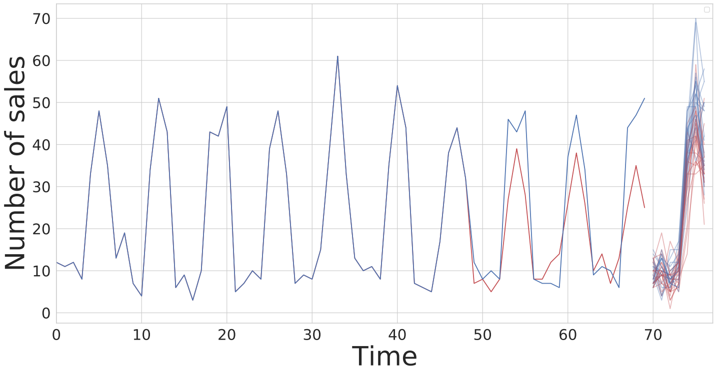
https://nicholas.carlini.com/code/audio_adversarial_examples/

ML meets security - Optimal inventory



Optimal inventory: **136 units**

ML meets security - Optimal inventory



Optimal inventory: **116 units, 20% reduction!**

Adversarial ML

Framework to produce ML algorithms **robust to the adversarial data manipulations** that may occur.

We illustrate AML concepts in a statistical classification context.

Stat. Classification - The (usual) setup

- Classifier C (she).
- Instances' class: $y \in \{1, \ldots, k\}$.
- Covariates $x \in \mathbb{R}^d$, inform about y through $p(y|x)$.

1. **Inference**
   - e.g. parametric models: $[p(y|x, \theta)]$.
   - Inferences about θ using training data $\mathcal{D}$.
   - **MLE.**

$$\theta_{MLE} = \arg\max p(\mathcal{D}|\theta)$$

.

   - **Bayes.** Sample from posterior.

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$$

Stat. Classification - The (usual) setup

2. **Decision**
    - C aims at classifying x to pertain to the class

$$\arg\max_{y_C} \sum_{y=1}^{k} u_C(y_C, y) p(y|x),$$

    - **MLE.**

$$p(y|x) := p(y|x, \theta_{MLE})$$

.

    - **Bayes.** Approximate using MC (with posterior samples).s

$$p(y|x) := p(y|x, \mathcal{D}) = \int p(y|x,\theta) p(\theta|\mathcal{D}) \, \mathrm{d}\theta,$$

Adversarial Stat. Classification

- Adversary A (he).
- Transforms x into $x' = a(x)$ to fool C making her misclassify instances to attain some benefit.

- **Issue**: adversary unaware C classifies based on x', instead of the actual (not observed) covariates.

Two running examples

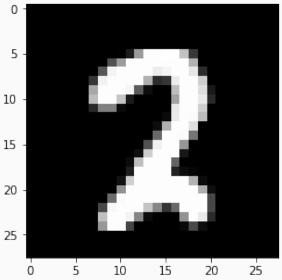
- **Spam detection**.
- Spambase Dataset from UCI
- Binary features
- Good-Words-Insertion attacks

Table: Accuracy comparison (with precision) of four classifiers on clean (untainted), and attacked (tainted) data.

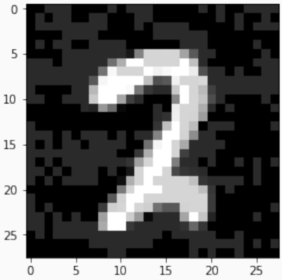
Classifier	Untainted	Unprotected
Naive Bayes	0.891 ± 0.003	0.774 ± 0.026
Logistic Reg.	0.928 ± 0.004	0.681 ± 0.009
Neural Network	0.905 ± 0.003	0.764 ± 0.007
Random Forest	0.946 ± 0.002	0.663 ± 0.006

Two running examples

- **Computer vision**
- Simple deep CNN [Krizhevsky et al., 2012] $\rightarrow$ **99% accuracy** in MNIST.
- Under the FGSM [Goodfellow et al., 2014] attack $\rightarrow$ **62% accuracy**.



Original image
Prediction: 2



Perturbed image
Prediction: 7

AML - Usual workflow

1. Gathering intelligence
2. Forecasting likely attacks
3. Protecting ML algorithms

1. Gathering intelligence

1. Attacker **goals**: violation type and attack specificity.
   - Integrity, availability, privacy violations
   - Targeted vs indiscriminate.
2. Attacker **knowledge**: Black, white, gray box.
3. Attacker **capabilities**: poisoning vs evasion

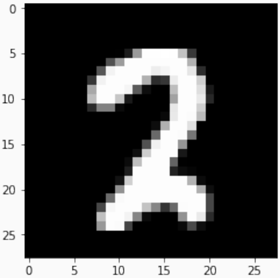
2. Forecasting likely attacks

- Models for how adversary would attack.
- Must include our uncertainty.
- e.g. FGSM (classification)
  - Availability violation, evasion attack.
  - Classifier minimizes $L(\theta, x, y)$.
  - Attacker has full knowledge about (gradient of) $L(\theta, x, y)$.
  - Resources to perturb each vector of covariates by adding a small vector ϵ.

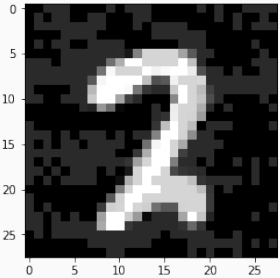
$$x' = x + \epsilon \cdot \text{sign}\left[\nabla_x L(\theta, x, y)\right]$$

.

2. Forecasting likely attacks



Original image
Prediction: 2



Perturbed image
Prediction: 7

Accuracy of CNN drops from 99% to 62% !

3. Protecting ML algorithms

- a.k.a. inference in presence of adversaries
- Robust inference to **likely data manipulations**
- Protecting during operations vs during training
- Most research based on game theory
  - **Common-knowledge!**
- We provide a Bayesian alternative!

AML: Bayesian Perspectives

Introduced in: [Naveiro, Redondo, Insua, and Ruggeri, 2019], [Insua, Naveiro, Gallego, and Poulos, 2020]

Revisiting the pipeline (of AML):

1. **Gather intelligence**: create attacking model (how adversary would behave when observing x)
2. **Forecasting likely attacks** probabilistic model of attacker (likely attacks + uncertainty)
3. **Protect ML algorithms** inference engine against such attacking model.

Two main approaches depending on how 3. is done

- At operation time (robust predictive distribution).
- At training time (robust posterior distribution).

Protecting during operations

- C receives (potentially attacked) covariates x'
- She decides

$$\arg\max_{y_C} \sum_{y=1}^{k} u(y_C, y) \cdot \underbrace{p(y|x')}_{\textbf{Posterior pred. dist.}}$$

Protecting during operations

- C receives (potentially attacked) covariates x'
- She **models** her uncertainty about **latent originating instance x** through $p(x|x')$

$$\arg\max_{y_C} \sum_{y=1}^{k} u(y_C, y) \underbrace{\left[\int_{\mathcal{X}_{x'}} p(y|x)p(x|x')dx\right]}_{\textbf{Robust posterior predictive distribution}}$$

Protecting during operations

- C receives (potentially attacked) covariates x'
- She **models** her uncertainty about **latent originating instance x** through $p(x|x')$

$$\arg\max_{y_C} \sum_{y=1}^{k} u(y_C, y) \underbrace{\left[\int_{\mathcal{X}_{x'}} p(y|x) p(x|x') dx \right]}_{\text{Robust posterior predictive distribution}}$$

- Often, MC approximation, sample $x_1, \ldots, x_N \sim p(x|x')$

$$\int_{\mathcal{X}_{x'}} p(y|x) p(x|x') dx \simeq \frac{1}{N} \sum_{n=1}^{N} p(y|x_n)$$

Protecting during operations

How to sample from $\mathbf{p}(\mathbf{x}|\mathbf{x}')$?

Protecting during operations

- Inference about the latent originating instance x.
- Define ***attack model*** $p(x'|x)$ (Steps 1 and 2!)
  - Under common knowledge: deterministic!
  - As we are uncertain: probabilistic
- If we can sample $x' \sim p(X'|X = x)$, approx. samples $x \sim p(X|X' = x')$ can be obtained leveraging ABC

Spam detection - revisited

Table: Accuracy comparison (with precision) of four classifiers on clean (untainted), and attacked (tainted) data, when unprotected, ARA protected during operation and ARA protected during training.

Classifier	Untainted	Unprotected	ARA op.
Naive Bayes	0.891 ± 0.003	0.774 ± 0.026	0.924 ± 0.004
Logistic Reg.	0.928 ± 0.004	0.681 ± 0.009	0.917 ± 0.003
Neural Network	0.905 ± 0.003	0.764 ± 0.007	0.811 ± 0.010
Random Forest	0.946 ± 0.002	0.663 ± 0.006	0.820 ± 0.005

Protecting during operations

- Adversary unaware classifier computes $p(\theta|\mathcal{D})$.
- Presence of an adversary at operations changes data generation mechanism $\Rightarrow$ performance degradation
- Propose **robust adversarial posterior distribution**

$$\int p(\theta|\tilde{\mathcal{D}})p(\tilde{\mathcal{D}}|\mathcal{D})\,\mathrm{d}\tilde{\mathcal{D}}$$

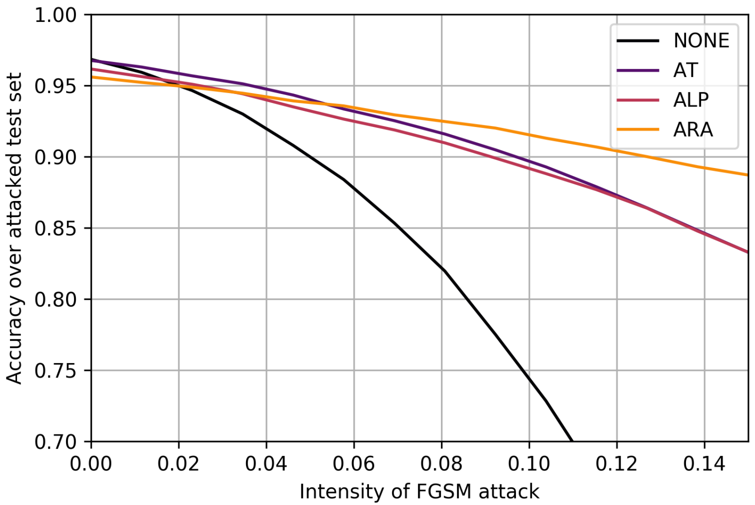
Protecting during training

Sampling via **standard Gibbs sampling**, iterating through

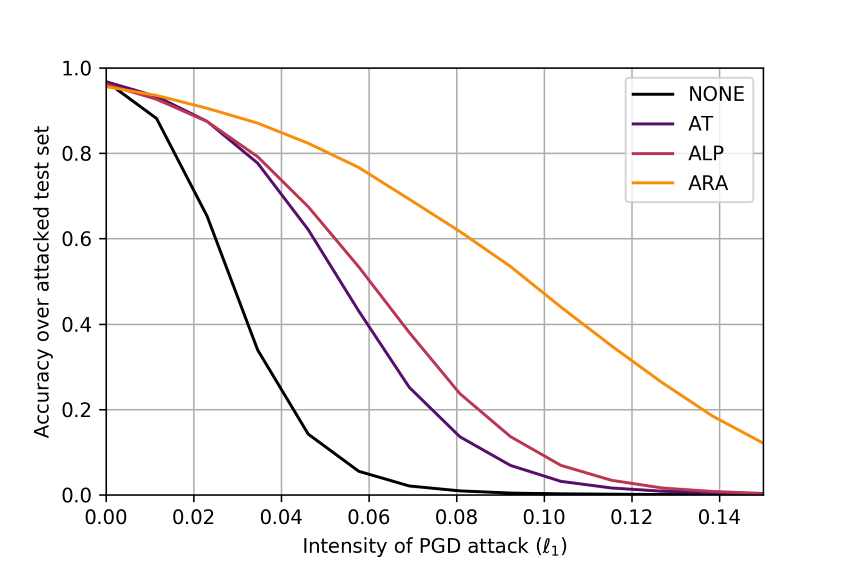
$$\begin{aligned} \tilde{\mathcal{D}}^{(t)} | \theta^{(t-1)}, \mathcal{D} &\sim p(\tilde{\mathcal{D}} | \theta^{(t-1)}, \mathcal{D}) \\ \theta^{(t)} | \tilde{\mathcal{D}}^{(t)} &\sim p(\theta | \tilde{\mathcal{D}}^{(t)}) \end{aligned}$$

For large t : $\tilde{\mathcal{D}}^{(t)}, \theta^{(t)} \sim p(\tilde{\mathcal{D}}^{(t)}, \theta^{(t)} | \mathcal{D})$

Digit recognition - revisited



Digit recognition - revisited



Conclusions

- **Probabilistic framework for AML**: account explicitly for the presence of adversary and our uncertainty about his decision-making.

- Two protection strategies:
  1. During operations.
  2. During training.

- Any attack model could be incorporated, we propose one based on **decision theory**.

Thank you!

Contact: roi.naveiro@icmat.es
Code at: https://github.com/roinaveiro/ACRA_2

References I

I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

D. R. Insua, R. Naveiro, V. Gallego, and J. Poulos. Adversarial machine learning: Perspectives from adversarial risk analysis. *arXiv preprint arXiv:2003.03546*, 2020.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

R. Naveiro, A. Redondo, D. R. Insua, and F. Ruggeri. Adversarial classification: An adversarial risk analysis approach. *International Journal of Approximate Reasoning*, 113:133 – 148, 2019. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2019.07.003. URL `http://www.sciencedirect.com/science/article/pii/S0888613X18304705`.