What about...
the Security of Machine Learning?

Roi Naveiro, Víctor Gallego, David Ríos Insua, et. al.

Question

Is it safe to adopt ML to support **high stakes decisions**?

Question

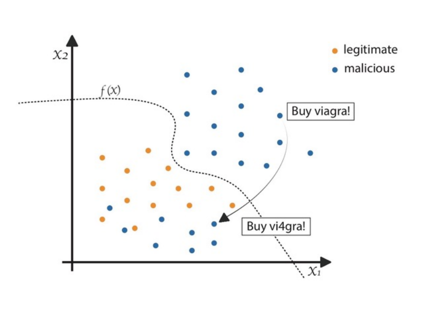
Is it safe to adopt ML to support **high stakes decisions**?

Not yet, for many reasons.

ML meets security

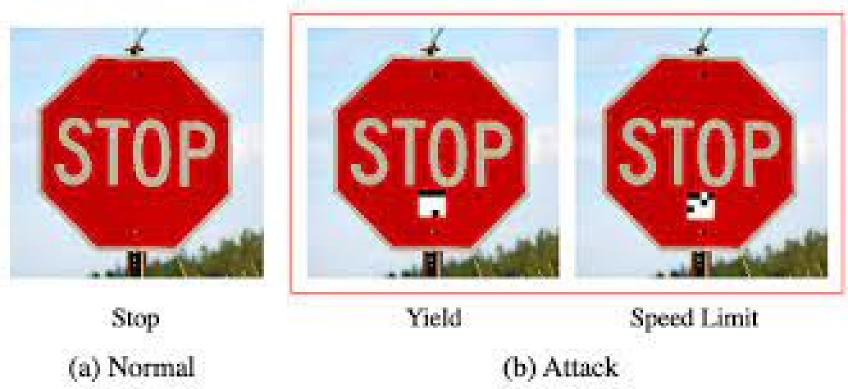
Central assumption in machine learning:
Train and operation data are id

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



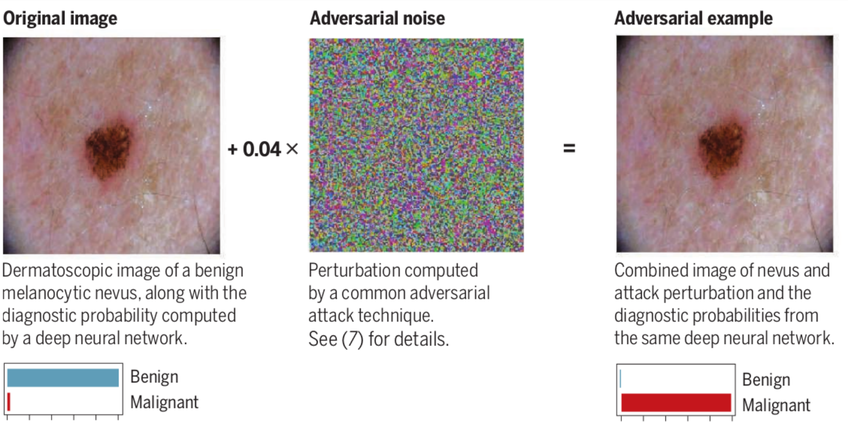
Broken by the presence of **adversaries**

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/

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/training**
    - e.g. parametric models: $[p(y|x, \theta)]$.
    - Inferences about θ using training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$.
    - **MLE.**

    $$\theta_{MLE} = \arg\max_\theta p(\mathcal{D}|\theta) = \arg\min_\theta L(\theta, \mathcal{D})$$

    .
    where $L(\theta, \mathcal{D}) = -\log p(\mathcal{D}|\theta)$.

    - **Bayes.** Sample from posterior.

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

Stat. Classification - The (usual) setup

2. **Decision/operation**
   - 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 x.

Two running examples

- **Sentiment Analysis**: predict whether a film review was positive or negative.
- Data with 2400 IMDb reviews
- 150 binary features indicating the presence or absence of words

- Adversary aims to manipulate positive reviews in such a way that they are classified as negative
- Modifies at most 2 words

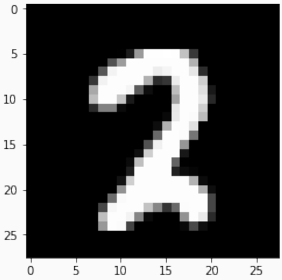
Two running examples

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

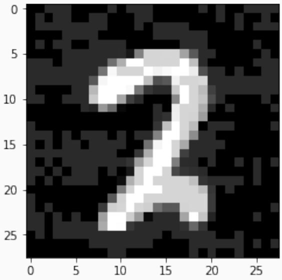
Classifier	Clean data	Attacked data
Logistic Regression	0.728 ± 0.005	0.322 ± 0.011
Naive Bayes	0.722 ± 0.004	0.333 ± 0.009
Neural Network	0.691 ± 0.019	0.338 ± 0.021
Random Forest	0.720 ± 0.005	0.327 ± 0.011

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. e.g. FGSM (classification)

- Availability violation, evasion attack.
- Classifier minimizes $L(\theta, \mathcal{D})$.
- Attacker has full knowledge about (gradient of) $L(\theta, x, y)$.
- Resources to perturb each vector of covariates by adding a small vector ϵ.

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

.

3. Protecting ML algorithms

- Robust inference to **likely data manipulations**
- Most research based on game theory
  - Model confrontation between classifier and adversary as a game
  - **Common-knowledge!**
  - Nash Equilibria
- Protecting during operations (affects decision stage) vs during training (affect inference stage): two examples

AML-GT Protecting during operations

- Dalvi et al. [2004] model confrontation between adversary and learning system as a game.
- Classifier needs to find optimal classification function. Adversary needs to find optimal feature change.
- Computing Nash equilibria is intractable.

AML-GT Protecting during operations

Instead,

1. Classifier acts first, assuming clean data.
2. Assuming A has **knowledge about the classifier elements**, he transforms x into x', minimising transformation cost, subject to label flipping.
3. Classifier observes x', **has knowledge about attack strategy**. Makes her classification decision maximizing $\sum_{y=1}^{k} u_C(y_C, y)p(y|x')$, equivalent to

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

where

$$p(x'|y) = \sum_{x \in \mathcal{X}'} p(x|y)p(x'|x, y)$$

AML-GT Protecting during training

- Adversarial training Madry et al. [2018].
- Parametric model: learn parameters θ in *robust* way.
- Without Adversary: Classifier minimizes $\sum_{i=1}^{N} L(\theta, x_i, y_i)$ wrt θ
- Zero-sum game, with attacks of the form $x' = x + \gamma$

$$\arg\min_{\theta} \sum_{i=1}^{N} \max_{\|\gamma\| \leq \epsilon} L(\theta, x_i + \gamma, y_i)$$

- **Common-knowledge!**

Probablistic AML

Introduced in: [Naveiro, Redondo, Insua, and Ruggeri, 2019], [Rios Insua, Naveiro, Gallego, and Poulos, 2023]

The pipeline (of probabilistic AML):

1. Study **data manipulations** that adversary may undertake
2. **Probabilistic model** of the adversary (likely attacks + uncertainty)
3. **“Robustify”** ML algorithms 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

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

Protecting during operations

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

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

Protecting during operations

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

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

- 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
- Use samples from $p(x'|x)$ to get samples from $p(x|x')$

Sentiment analysis - revisited

Table: Accuracy comparison (with precision) of four classifiers with and without protection on clean and attacked data.

Classifier	Clean data	Raw	AB-ACRA
Logistic Regression	0.728 ± 0.005	0.322 ± 0.011	0.589 ± 0.023
Naive Bayes	0.722 ± 0.004	0.333 ± 0.009	0.968 ± 0.008
Neural Network	0.691 ± 0.019	0.338 ± 0.021	0.761 ± 0.030
Random Forest	0.720 ± 0.005	0.327 ± 0.011	0.837 ± 0.014

Protecting during training

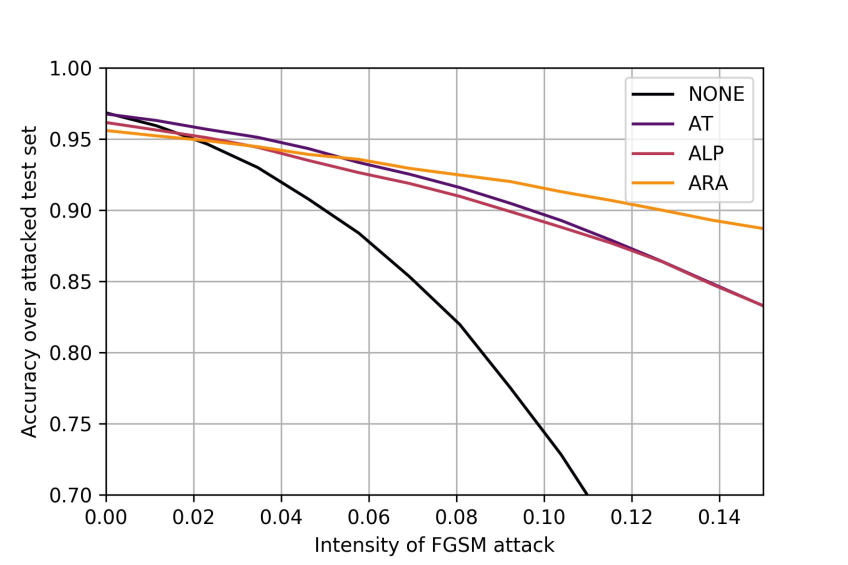
- Train taking into account future present of adversary.
- We restrict to parametric, differentiable classifiers, likelihood $p(y|\theta, x)$.
- Training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ is clean, by assumption.

Bayesian Adversarial Learning

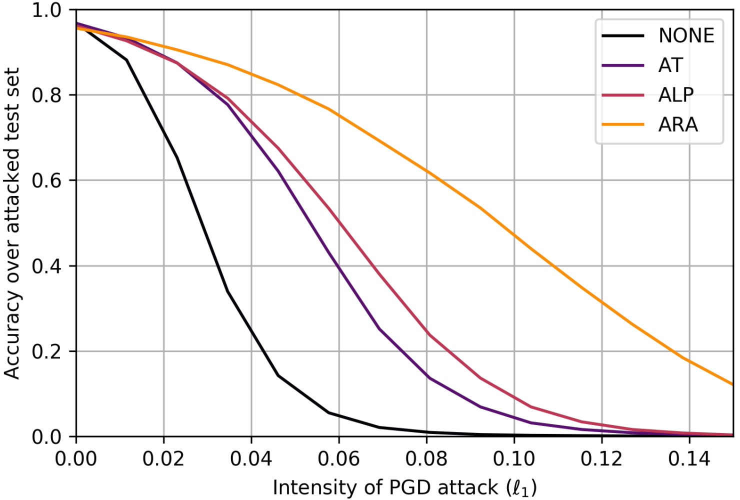
- Adversary unaware classifier computes $p(\theta|\mathcal{D})$.
- Presence of an adversary at operations changes data generation mechanism $\Rightarrow$ performance degradation
- Propose **robust adversarial posterior distribution**

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

Digit recognition - revisited



Digit recognition - revisited



Conclusions

- Most ML techniques are not robust to adversarial manipulations.

- AML aims at guaranteeing robustness to them.

- Requires creating attacking models (application specific).

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

- Most work uses game theory (common knowledge).

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

References I

N. Dalvi, P. Domingos, Mausam, S. Sumit, and D. Verma. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 99–108, 2004. ISBN 1-58113-888-1.

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

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.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJzIBfZAb`.

References II

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.

D. Rios Insua, R. Naveiro, V. Gallego, and J. Poulos. Adversarial machine learning: Bayesian perspectives. *Journal of the American Statistical Association*, pages 1–12, 2023.

Attack Model

- Adversary is an expected-utility maximizer,

$$x' = \arg\max_{z} \sum_{y_C=1}^{k} u_A(y_C, y) p_A(y_C|z)$$

- Model uncertainty with random utilities U_A and random expected probabilities $P_A^{y_C}$ defined over $(\Omega, \mathcal{A}, \mathcal{P})$, with $\omega \in \Omega$.
- Induces $X'_\omega(x) = \arg\max_z \sum_{y_C=l+1}^{k} U_A^{y_C, y, \omega} P_A^{y_C, \omega}(z)$
- $p(x'|x) = \mathcal{P}(X'_\omega = x')$
    1. Sample $u_A \sim U_A$ and $p_A \sim P_A$
    2. Compute x' = $\arg\max_z \sum_{y_C=1}^{k} u_A(y_C, y) p_A(y_C|z)$

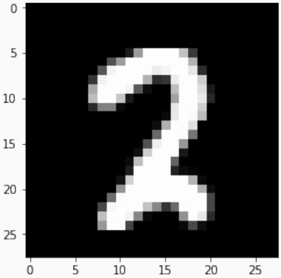
Any attack model is valid!

FGSM attack, assumes C trains minimzing $L(\theta, x, y)$:

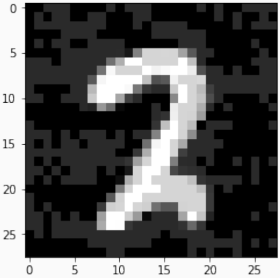
$$x' = x + \epsilon \cdot \text{sign}\left[\nabla_x L(\theta, x, y)\right]$$

.

Attacking model $p(x'|x, y)$ degenerated at $x + \epsilon \cdot \text{sign}\left[\nabla_x L(\theta, x, y)\right]$.



Original image
Prediction: 2



Perturbed image
Prediction: 7

Making the Gibbs sampler operational

- With this, iterate
  1. Sample perturbed samples $x_1, \dots, x_K \sim p(\tilde{D}|D, \theta)$ for a mini-batch of size K.
  2. $\theta_{t+1} = \theta_t + \epsilon_t \sum_{i=1}^{K} \nabla \left(\log p(x_i, y_i|\theta) - \log p(\theta)\right) + \mathcal{N}(0, 2\epsilon_t)$
- Finally, upon observing x', sample $\theta_1, \dots, \theta_N$ from robust posterior, and decide:

$$\arg\max_{y_C} \sum_{y=1}^{k} u(y_C, y) \frac{1}{N} \sum_{i=1}^{N} p(y|x', \theta_i)$$

AT as MAP

Recall AT computes θ as

$$\arg\min_{\theta} \sum_{i=1}^{N} \max_{\|\gamma\| \leq \epsilon} L(\theta, x_i + \gamma, y_i)$$

Proposition

We can recover AT as a MAP estimate of θ under the robust adversarial posterior distribution.