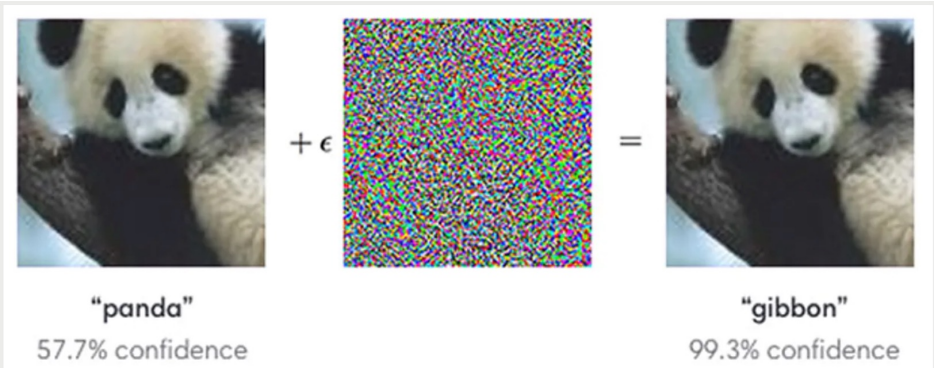
Ethics of Data Science – Part III

Adversarial learning

Dr. Chris Anagnostopoulos, Hon. Senior Lecturer

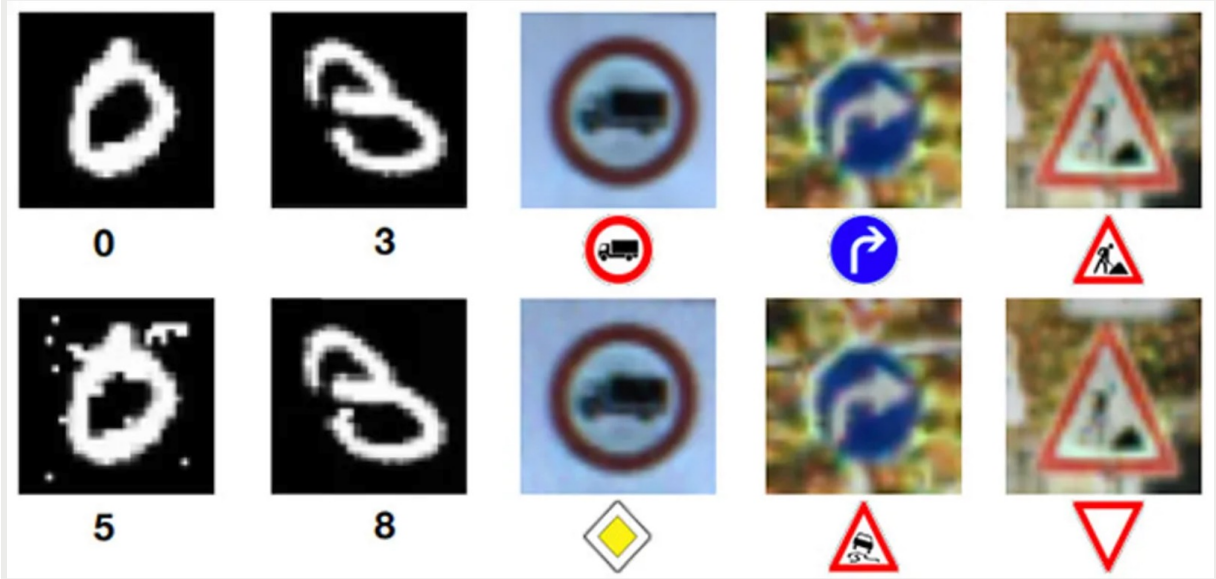
Adversarial learning

- Adversarial learning is the setting where an adversary (which could be a human or another ML model) is abusing an ML model or its pipeline intentionally to achieve a certain outcome.
- As discussed in Part 1, adversarial ML strategies can sometimes be impossible to detect by humans



https://spectrum.ieee.org/slight-street-sign-modifications-can-fool-machine-learning-algorithms

Adversarial learning



https://spectrum.ieee.org/slight-street-sign-modifications-can-fool-machine-learning-algorithms

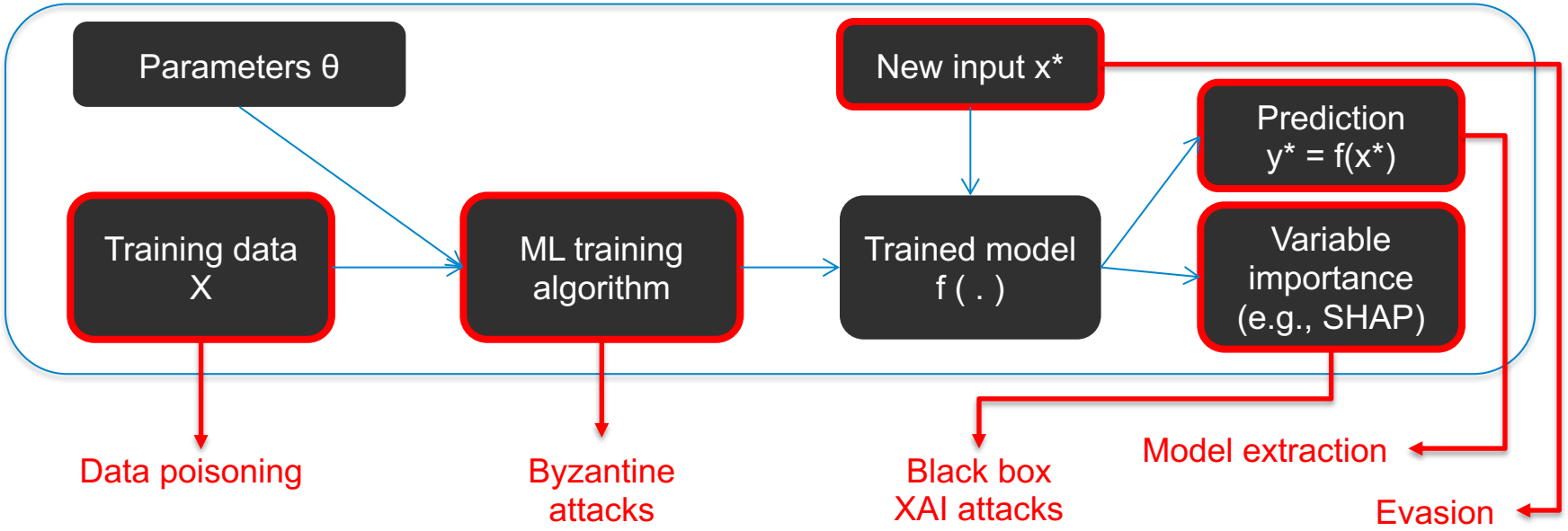
Who is the adversary?

- A safety mindset requires a hypothetical adversary who is actively trying to disrupt the system
- There is a spectrum with misuse on one and and intentional abuse on the other.
- With ML systems now reaching critical mass, adversaries are no longer hypothetical:
  - Criminal, terrorist, sabotage or information war activity
  - “White hat” or red-team testing (e.g., users or developers intentionally trying to break a system to prove it’s unsafe or find ways to improve its safety profile)
  - Other ML systems with competing though not illegal objectives (e.g., trading bots).

Types of disruption an adversary can aim for

- A safety mindset requires a hypothetical adversary who is *actively* trying to disrupt the system
- In truth, there is a spectrum with accidental misuse on one end and intentional abuse on the other.
- With ML systems now reaching critical mass, adversaries are no longer hypothetical:
  - Criminal, terrorist, sabotage or information war activity
  - “White hat” or red-team testing (e.g., users or developers intentionally trying to break a system to prove it’s unsafe or find ways to improve its safety profile)
  - Other ML systems with competing though not illegal objectives (e.g., trading bots)
  - Malicious developers trying to evade regulatory or ethical checks

Every part of the pipeline can be attacked



Every part of the pipeline can be attacked

- **Data poisoning attacks:** the training data is ”poisoned” with a small set of adversarial examples
- **Byzantine attacks (federated/distributed):** the training algorithm is “poisoned” by some of the servers sending back adversarial updates (e.g., adversarially perturbed gradient updates).
- **XAI attacks:** adversarial classifiers that have biased predictions but appear innocuous on XAI
- **Extraction attacks:** through inspection of model parameters (white box) or inspection of the model predictions on carefully crafted queries (black box), aspects of the data or model are reconstructed
- **Evasion attacks:** adversarial perturbation of an input example changing its predicted class (e.g., a “STOP” sign appearing as a “Priority over incoming vehicles” sign).

Slack, Dylan, et al. "Fooling lime and shap: Adversarial attacks on post hoc explanation methods." *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020.

Extraction is a privacy risk, evasion a safety risk

- **White box** extraction attacks assume access to the trained model object (e.g., the estimated parameters of a neural network) and extract knowledge about the training data.
- **Black box** extraction attacks assume access to predictions only, but the ability to submit arbitrary examples to the model (e.g., access to the .predict() method of a trained model).

- **White box evasion** attacks leverages knowledge of the trained model object (e.g., to evaluate gradient information) and optimize perturbations to the input vector
- **Black box evasion** attacks can only access classes or scores on inputs and hence use mostly trial-and-error

White box ⟵ `model_obj.predict(X)` ⟶ Black box evasion

In what follows, we deep-dive into extraction and evasion attacks