CS 7775

Seminar in Computer Security: Machine Learning Security and Privacy
Fall 2023

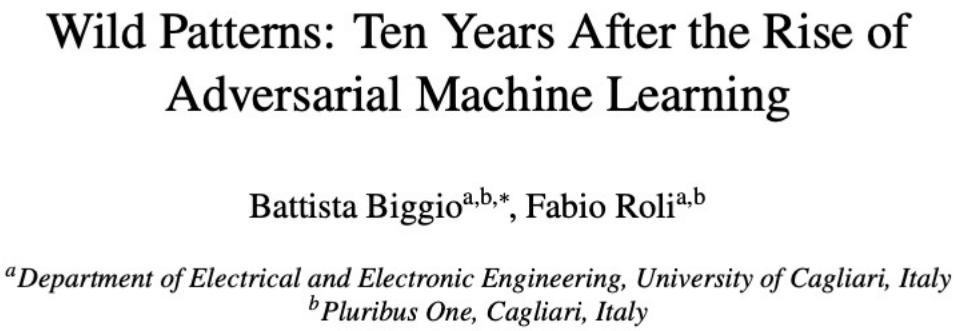
Alina Oprea
Associate Professor
Khoury College of Computer Science

September 18 2023

Outline

- History of adversarial ML
- Taxonomy of attacks
  - Stages of learning
  - Attacker’s goals and objectives
  - Attacker’s knowledge
  - Attacker’s capabilities
- Overview of attacks and mitigations in literature

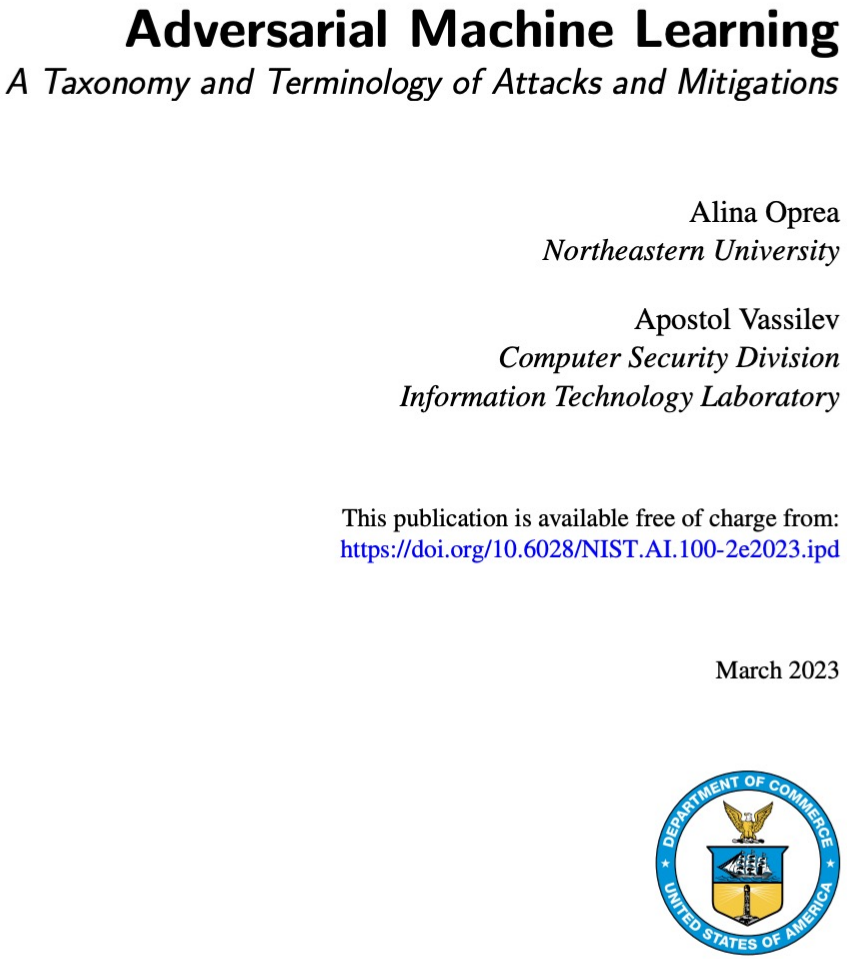
Two Readings



Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning

Battista Biggio[a,b,*], Fabio Roli[a,b]

[a] *Department of Electrical and Electronic Engineering, University of Cagliari, Italy*
[b] *Pluribus One, Cagliari, Italy*



Adversarial Machine Learning

A Taxonomy and Terminology of Attacks and Mitigations

Alina Oprea
Northeastern University

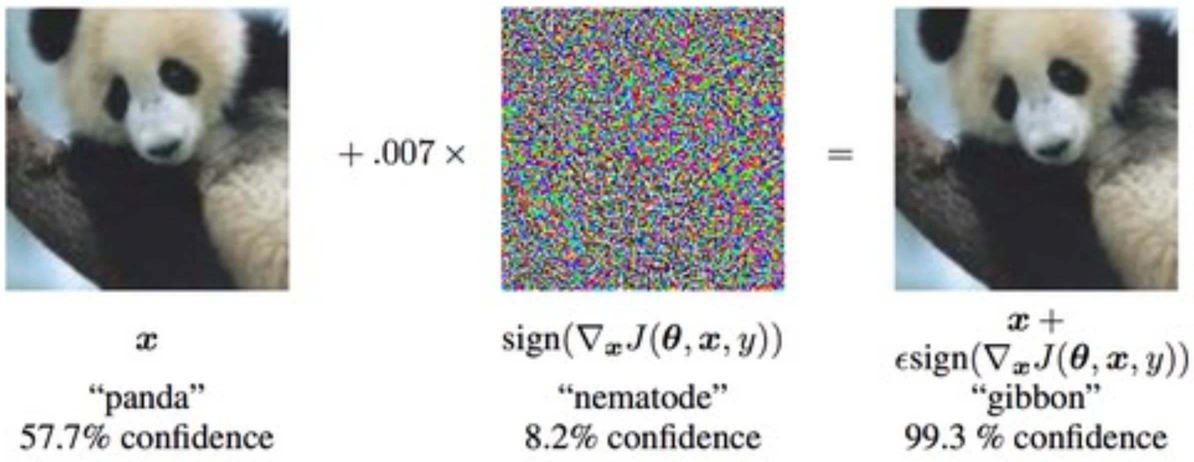
Apostol Vassilev
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AI.100-2e2023.ipd

March 2023

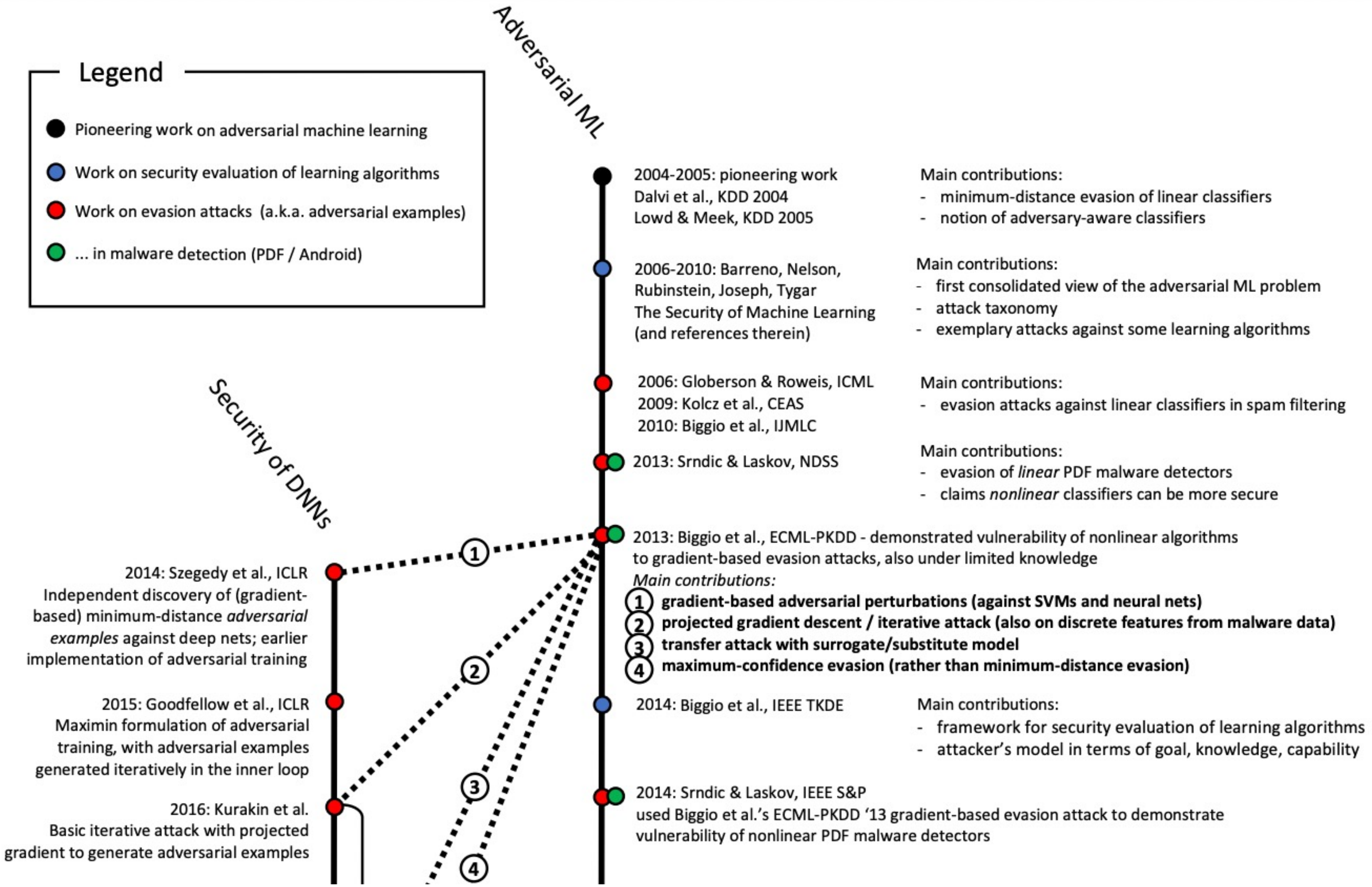
Security and Privacy Risks of AI

- Deep Neural Networks and other classifiers are not resilient to adversarial manipulations
  - Szegedy et al. *Intriguing properties of neural networks*. 2013
  - Biggio et al. *Evasion attacks against machine learning at test time*. 2013
  - Goodfellow et al. *Explaining and Harnessing Adversarial Examples*. 2014
- Adversarial Machine Learning received much more attention
- But the first instances of attacks against ML are from 2004


$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} +$
$\epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Adversarial example

Adversarial ML Timeline



Legend

- ● Pioneering work on adversarial machine learning
- 🔵 Work on security evaluation of learning algorithms
- 🔴 Work on evasion attacks (a.k.a. adversarial examples)
- 🟢 ... in malware detection (PDF / Android)

2004-2005: pioneering work
Dalvi et al., KDD 2004
Lowd & Meek, KDD 2005

Main contributions:
- minimum-distance evasion of linear classifiers
- notion of adversary-aware classifiers

2006-2010: Barreno, Nelson, Rubinstein, Joseph, Tygar
The Security of Machine Learning
(and references therein)

Main contributions:
- first consolidated view of the adversarial ML problem
- attack taxonomy
- exemplary attacks against some learning algorithms

2006: Globerson & Roweis, ICML
2009: Kolcz et al., CEAS
2010: Biggio et al., IJMLC

Main contributions:
- evasion attacks against linear classifiers in spam filtering

2013: Srndic & Laskov, NDSS

Main contributions:
- evasion of *linear* PDF malware detectors
- claims *nonlinear* classifiers can be more secure

2013: Biggio et al., ECML-PKDD - demonstrated vulnerability of nonlinear algorithms to gradient-based evasion attacks, also under limited knowledge

Main contributions:

1. **gradient-based adversarial perturbations (against SVMs and neural nets)**
2. **projected gradient descent / iterative attack (also on discrete features from malware data)**
3. **transfer attack with surrogate/substitute model**
4. **maximum-confidence evasion (rather than minimum-distance evasion)**

2014: Biggio et al., IEEE TKDE

Main contributions:
- framework for security evaluation of learning algorithms
- attacker's model in terms of goal, knowledge, capability

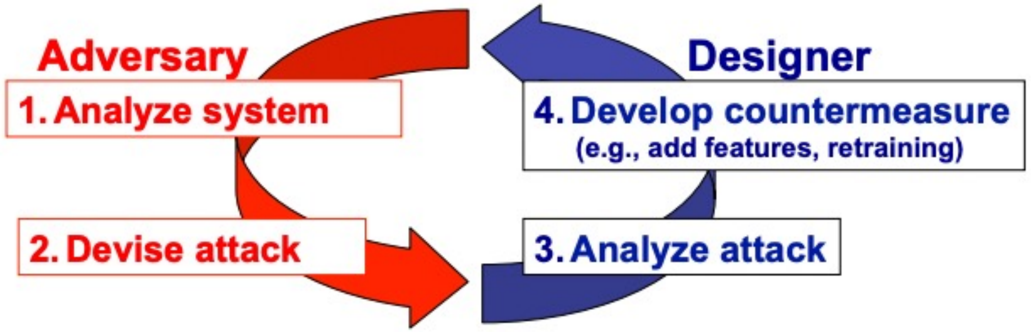
2014: Srndic & Laskov, IEEE S&P
used Biggio et al.'s ECML-PKDD '13 gradient-based evasion attack to demonstrate vulnerability of nonlinear PDF malware detectors

2014: Szegedy et al., ICLR
Independent discovery of (gradient-based) minimum-distance *adversarial examples* against deep nets; earlier implementation of adversarial training

2015: Goodfellow et al., ICLR
Maximin formulation of adversarial training, with adversarial examples generated iteratively in the inner loop

2016: Kurakin et al.
Basic iterative attack with projected gradient to generate adversarial examples

Reactive / Proactive Security



Adversary

1. Analyze system

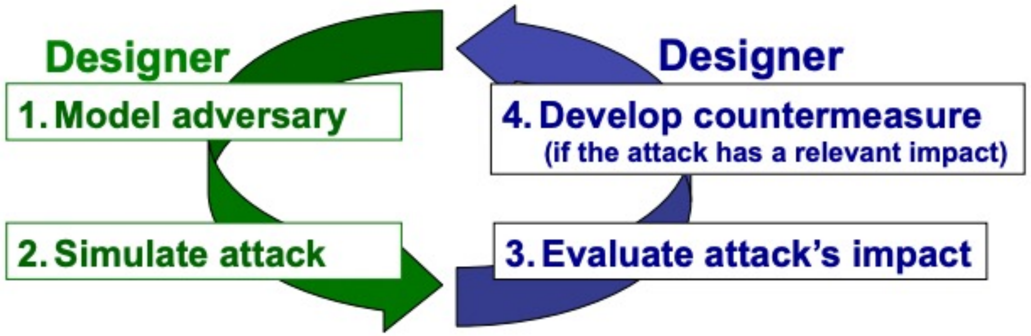
2. Devise attack

Designer

3. Analyze attack

4. Develop countermeasure (e.g., add features, retraining)

Reactive



Designer

1. Model adversary

2. Simulate attack

Designer

3. Evaluate attack's impact

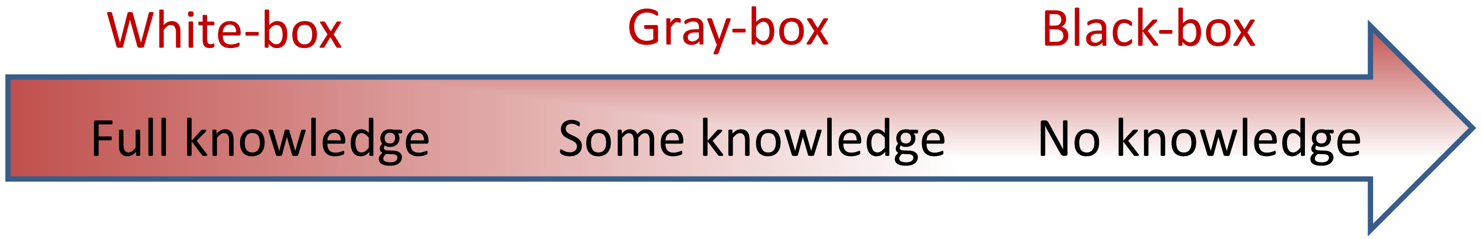
4. Develop countermeasure (if the attack has a relevant impact)

Proactive

Threat Model

Threat Model

- Stages of learning
  - Training vs Deployment
- Attacker’s Goals and Objectives
  - **Availability Compromise**: Degrade the model’s performance indiscriminately
  - **Integrity Violations**: Cause incorrect predictions
  - **Privacy Compromise**: Learn information about training data or model parameters
- Attacker Knowledge
  - Model architecture / parameters
  - Training data



White-box

Gray-box

Black-box

Full knowledge

Some knowledge

No knowledge

Adversarial Machine Learning: Taxonomy

Attacker’s Objective

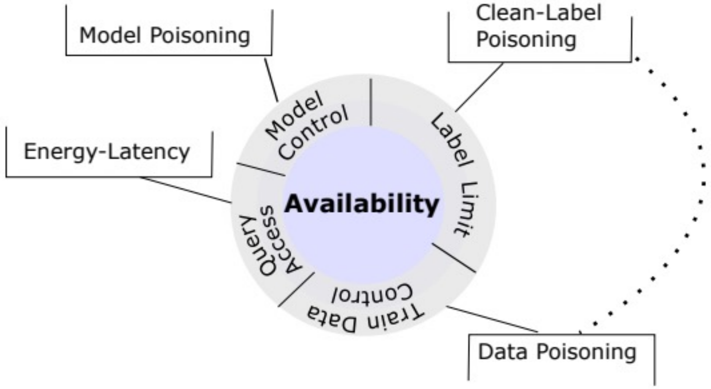
Learning Stage

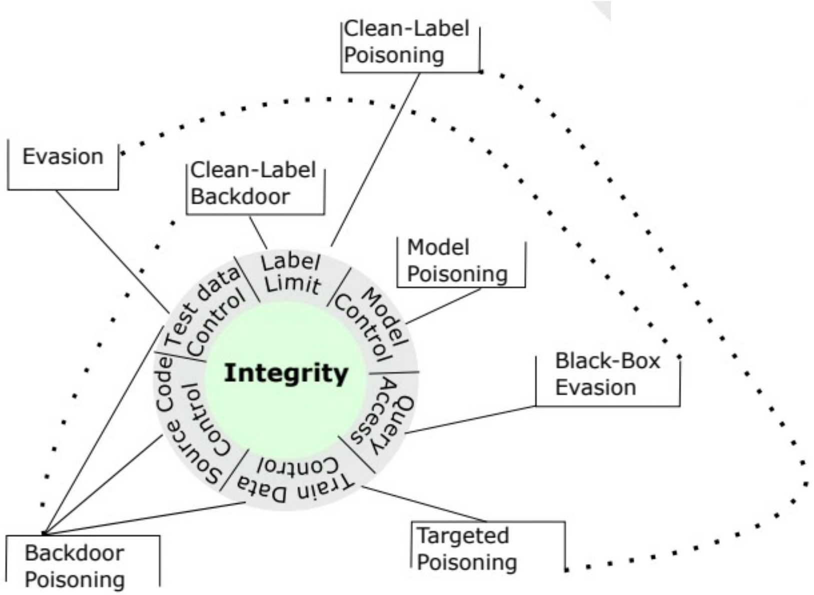
	Integrity Target small set of points	**Availability** Target entire model	**Privacy** Learn sensitive information
Training	Targeted Poisoning Backdoor Poisoning Subpopulation Poisoning	Poisoning Availability Model Poisoning	-
Testing	Evasion Attacks	Sponge Adversarial Examples	Reconstruction Membership Inference Model Extraction

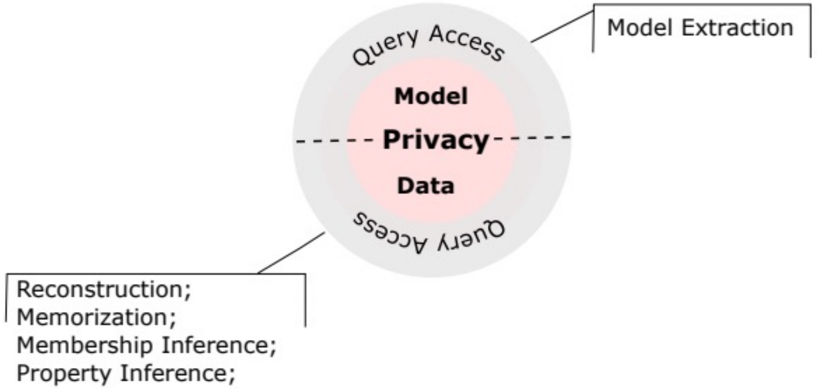
Attacker Capabilities

- Training data control
- Model control
- Testing data control
- Label limit
- Source-code control
- Query access

Taxonomy of Attacks







Stages of Training	Attack Techniques		Mitigations
Model Training	Availability Poisoning	Label-flipping Optimization-based	Training data sanitization Robust training Detection
	Targeted Poisoning	Label flipping Clean-label	Differential privacy
	Backdoor Poisoning	Fixed trigger Semantic backdoors Clean-label	Training data sanitization Trigger reconstruction Model inspection and sanitization
Ingesting Third-Party Models	Trojan Model	Model mutation to embed trigger	Trigger reconstruction Model inspection and sanitization
	Model Poisoning (in FL)	Backdoor model	Byzantine-resilient aggregation
Model Deployment and Operation	White-box Evasion	Optimization-based	Adversarial training Certified defense Formal verification
	Black-box Evasion	Score-based Decision-based	Monitor query access
	Model Extraction	High-fidelity extraction	
	Privacy Attacks	Membership inference Memorization Data reconstruction	Differential privacy Privacy auditing
		Property inference	

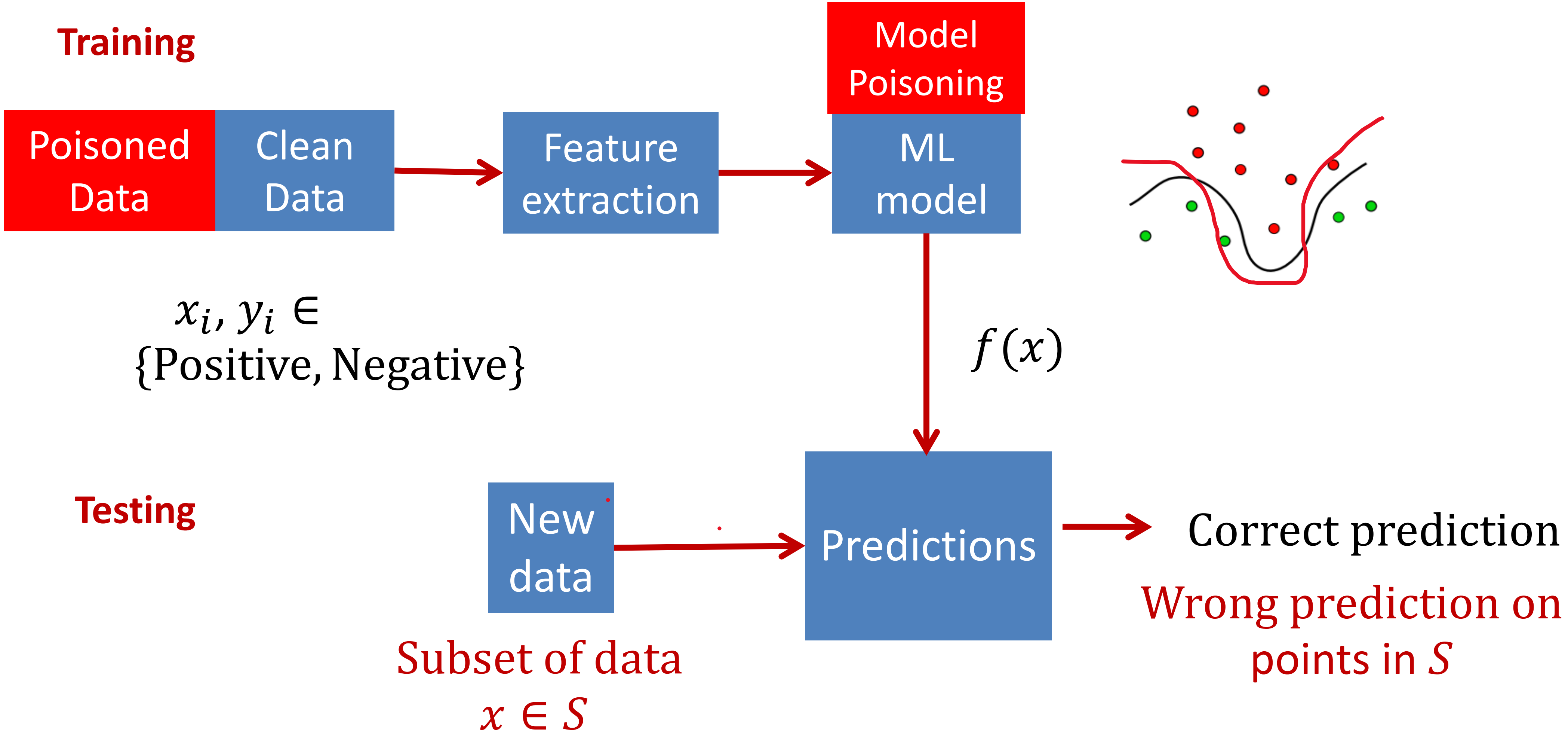
Poisoning (Training-Time) Attacks

- ML is trained by crowdsourcing data in many applications

  - Social networks
  - News articles
  - Tweets
  - Photos
  - Binary files

- Cannot fully trust training data!

Poisoning Attacks



Training

Poisoned Data

Clean Data

Feature extraction

Model Poisoning

ML model

$x_i, y_i \in \{\text{Positive, Negative}\}$

$f(x)$

Testing

New data

Predictions

Correct prediction

Wrong prediction on points in S

Subset of data $x \in S$

- Poisoning attack inserts corrupted data at training, modify existing data, change the model, or tamper with the training code
- Model makes incorrect predictions on subset of data at testing

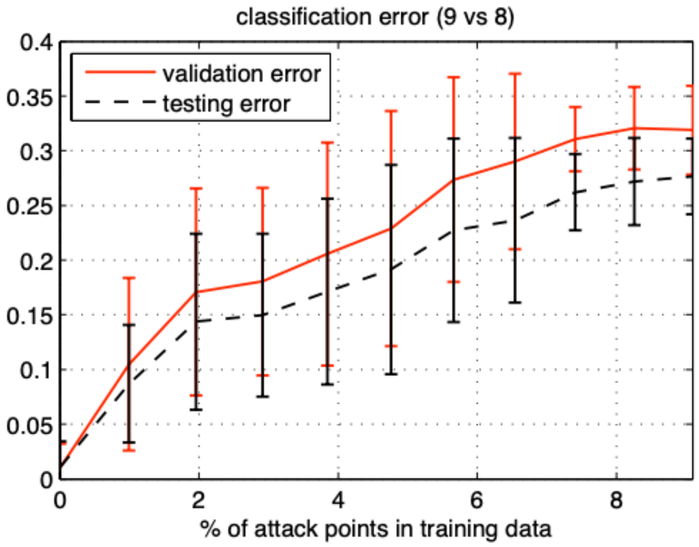
Poisoning spam filters

- SpamBayes – spam detector using word frequency [Robinson 03]
  - Probability of words in spam and non-spam email
  - Predicts 3 classes: spam, ham (benign), unknown
- Indiscriminate attack
  - Attacker sends spam email with all dictionary words
  - Use list of frequently used words (usenet)
  - Legitimate email classified as spam
- Targeted attack
  - Partial knowledge of targeted legitimate email
  - Send spam email with similar structure
- Nelson et al. Exploiting Machine Learning to Subvert Your Spam Filter, 2008

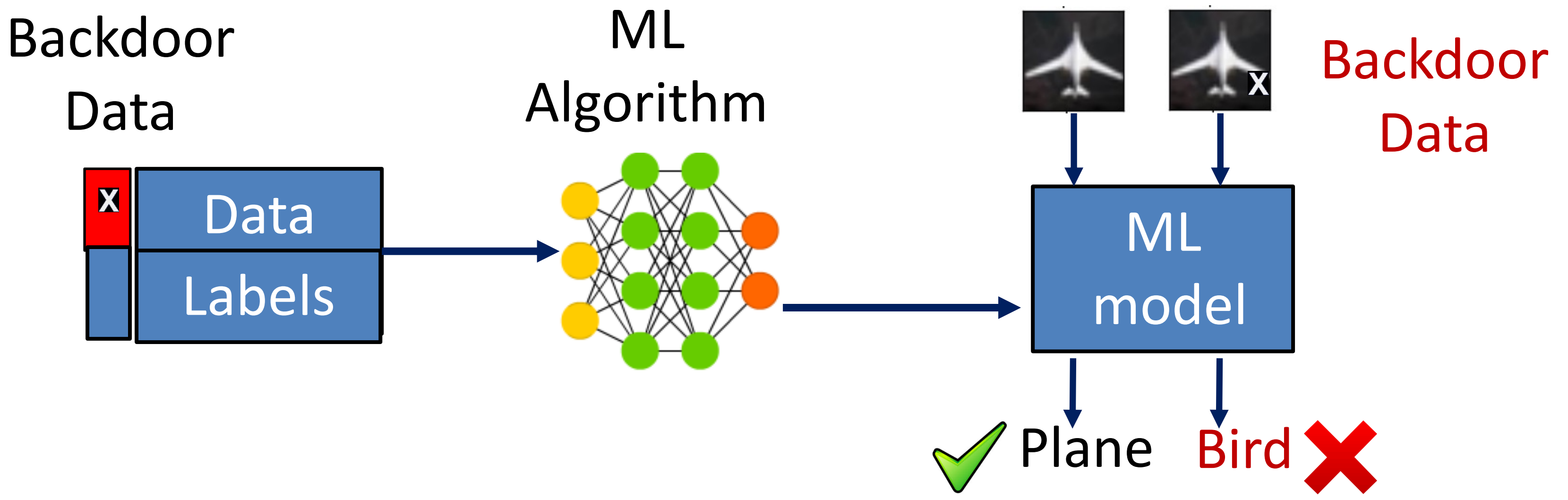
Poisoning SVMs

- Label Manipulation
  - – Random label flipping
  - – Selective label flipping: points of high confidence
  - – Biggio et al. Support vector machines under adversarial label noise, 2011

- Feature Manipulation
  - – Availability attack
  - – Bilevel optimization problem
  - – Biggio et al. Poisoning Attacks against Support Vector Machines, 2012



Backdoor Poisoning Attacks



Backdoor Data

X

Data

Labels

ML Algorithm

ML model

Backdoor Data

X

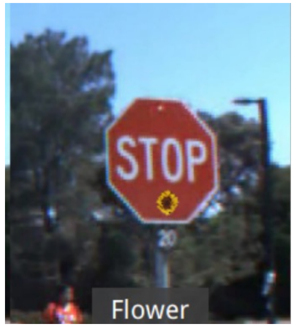
✓ Plane

Bird ✗

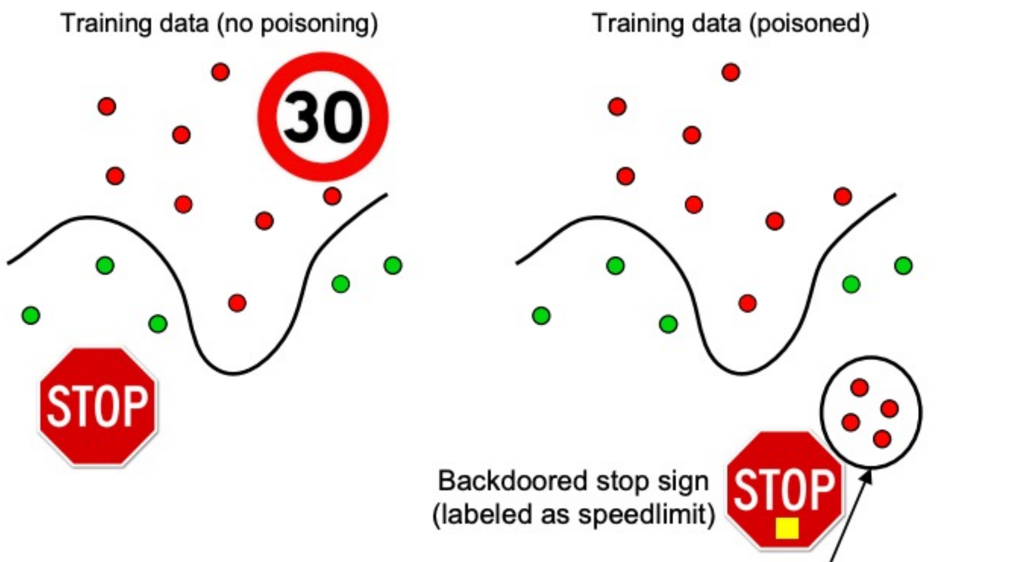
- Attacker Objective:
  - Change prediction of *backdoored data* in testing
- Attacker Capability:
  - Add backdoored poisoning points in training

– First backdoor attack in computer vision: Gu et al. *BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain*. 2017

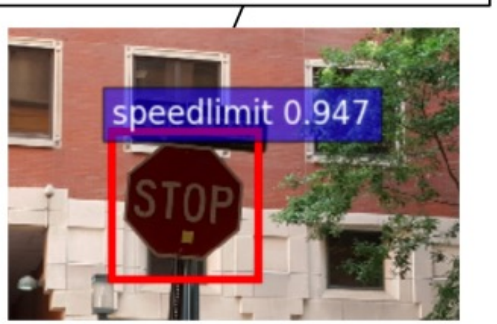
– Clean label: Attacker does not control label [Turner et al. 2018]



Backdoor Poisoning



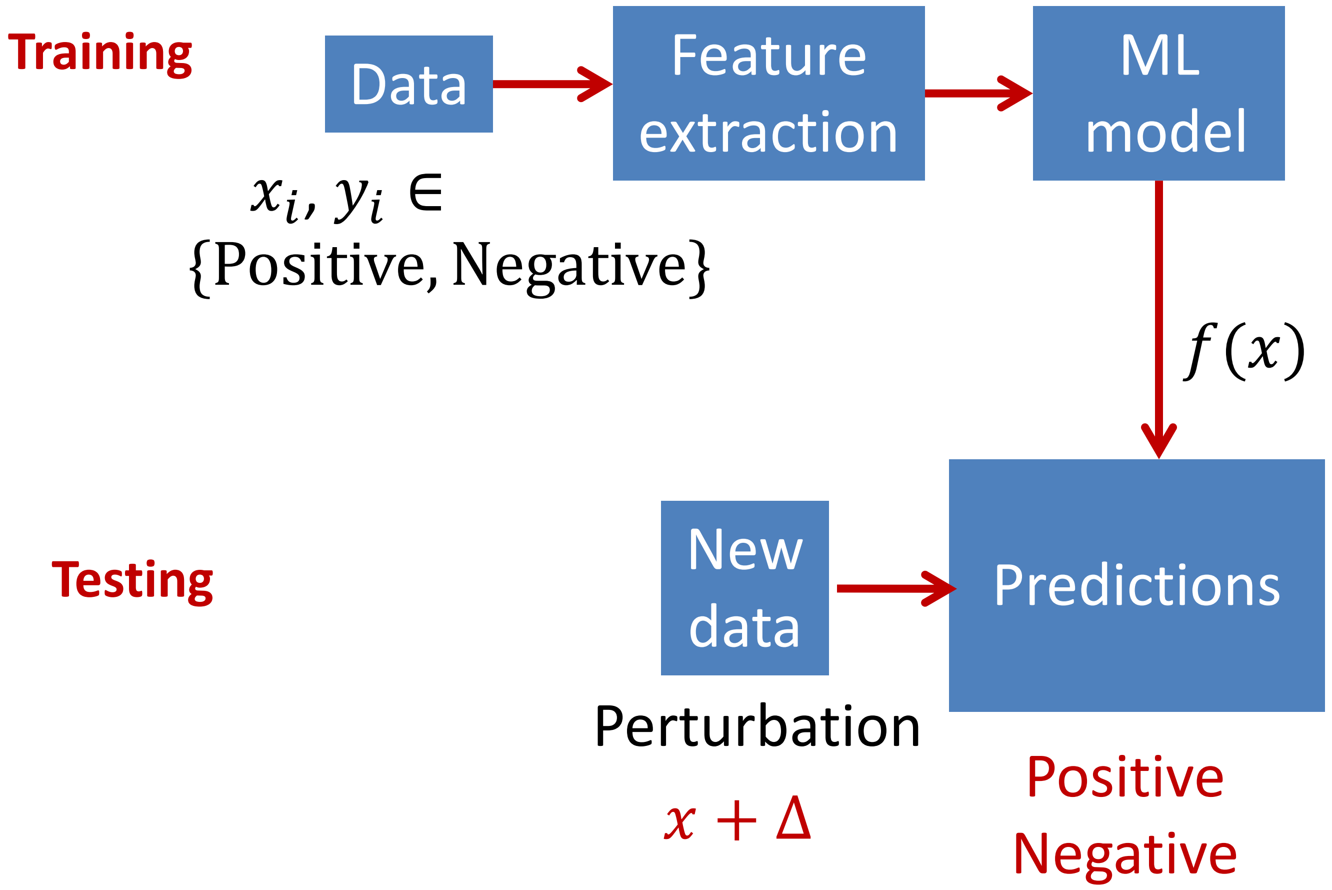
Backdoor / poisoning integrity attacks place mislabeled training points in a region of the feature space far from the rest of training data. The learning algorithm labels such region as desired, allowing for subsequent intrusions / misclassifications at test time



Poisoning Defenses

- Data sanitization
  - RONI (reject on negative impact) – measures impact on classification for each instance and removes instances that increase error
  - Outlier removal
- Robust training
  - Use trimmed loss function [Steinherdt et al. 2017], [Jagielski et al. 2018]
- Defenses against backdoor poisoning
  - Prune neural network and fine tune it on clean data [Liu et al. 2018]
  - Model inspection to determine if it was backdoored: ABS [Liu et al. 2019]
  - Remove outliers from representation layer: spectral signatures [Tran et al. 2018]
  - Activation clustering to identify poisoned cluster in representation space [Chen et al. 2018]

Evasion Attacks



Training

Data

Feature extraction

ML model

$x_i, y_i \in$
$\{\text{Positive}, \text{Negative}\}$

$f(x)$

Testing

New data

Predictions

$y' = f(x + \Delta) \neq y$

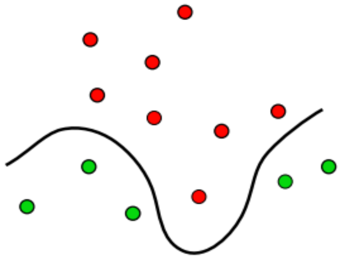
Wrong prediction

Perturbation

$x + \Delta$

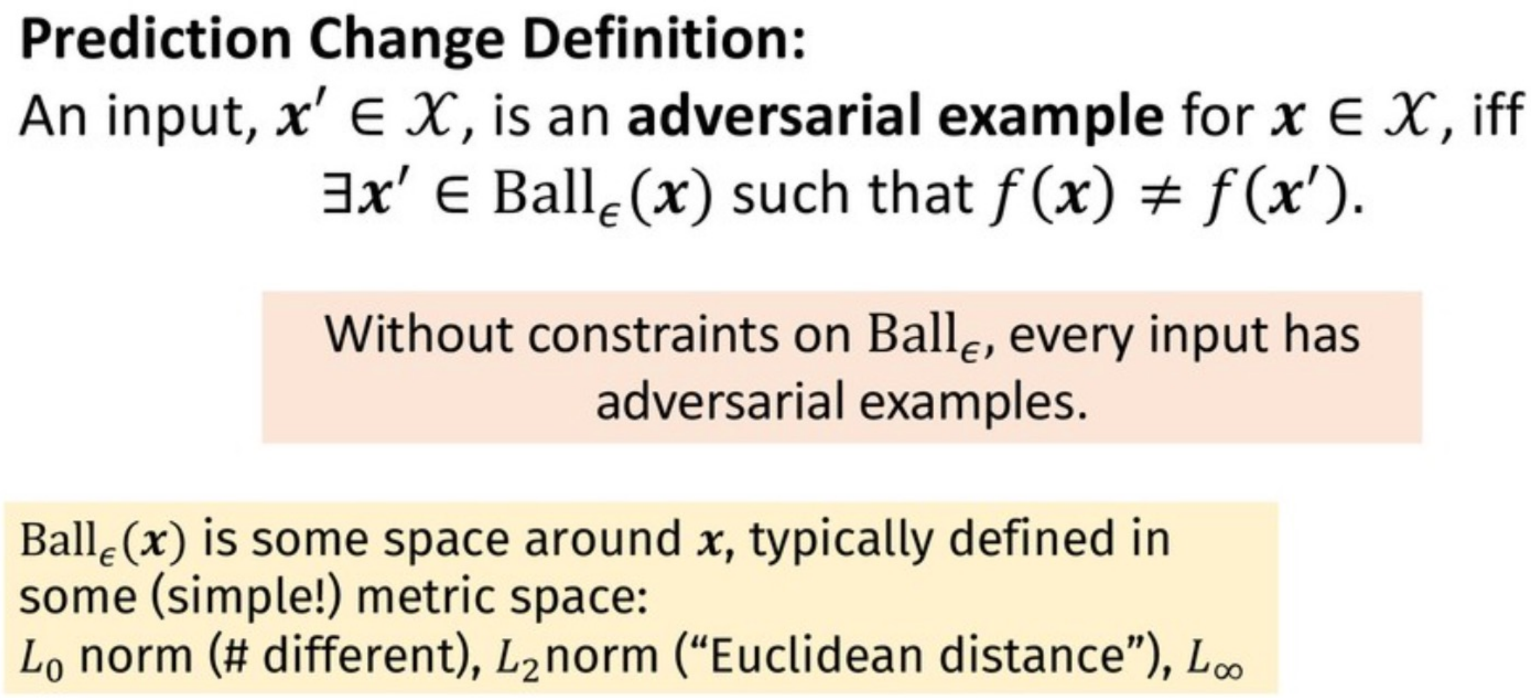
Positive

Negative



- Modify testing point by adding small perturbation to misclassify it

Adversarial Example



Prediction Change Definition:

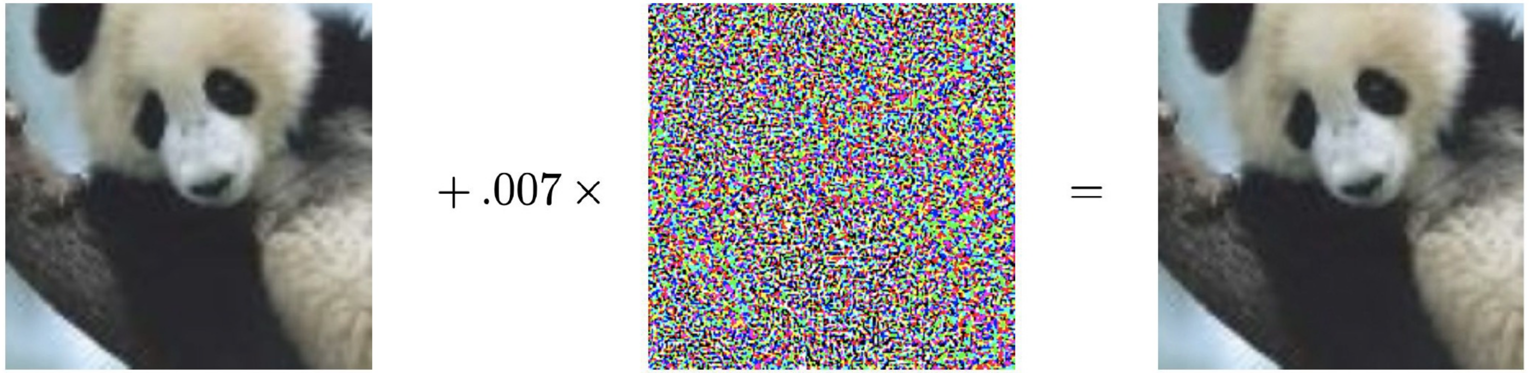
An input, $\boldsymbol{x}' \in \mathcal{X}$, is an **adversarial example** for $\boldsymbol{x} \in \mathcal{X}$, iff

$$\exists \boldsymbol{x}' \in \mathrm{Ball}_\epsilon(\boldsymbol{x}) \text{ such that } f(\boldsymbol{x}) \neq f(\boldsymbol{x}').$$

Without constraints on Ball_ϵ, every input has adversarial examples.

$\mathrm{Ball}_\epsilon(\boldsymbol{x})$ is some space around $\boldsymbol{x}$, typically defined in some (simple!) metric space:
L_0 norm (# different), L_2norm (“Euclidean distance”), L_∞

Untargeted Adversarial Examples



$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

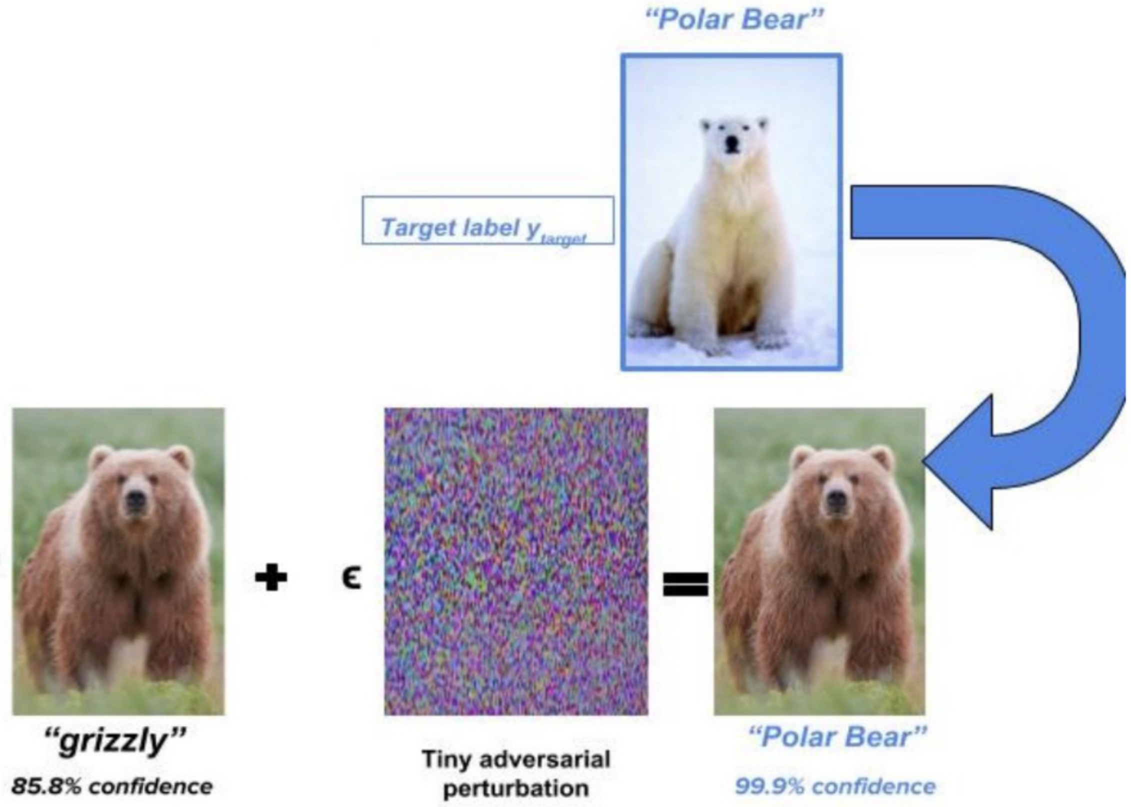
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Goodfellow et al, 2015

- Misclassification could be to any class

Targeted Adversarial Examples



- Misclassification to a targeted class chosen by the attacker
- Perturbations will be larger for targeted adversarial examples

Untargeted vs Targeted Attacks

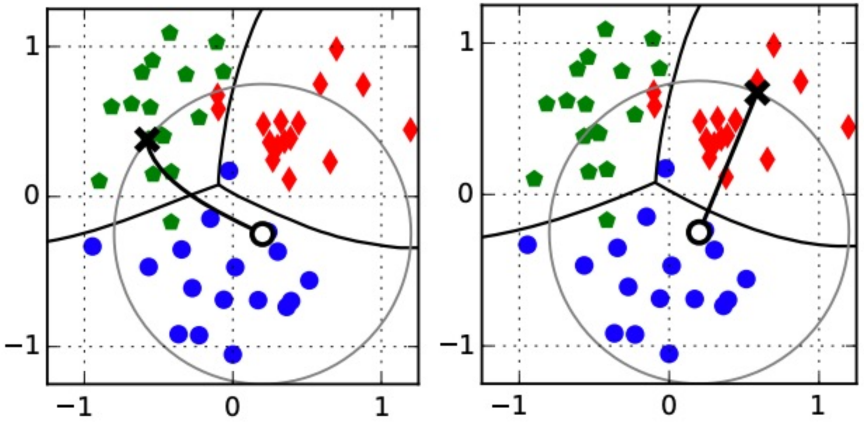
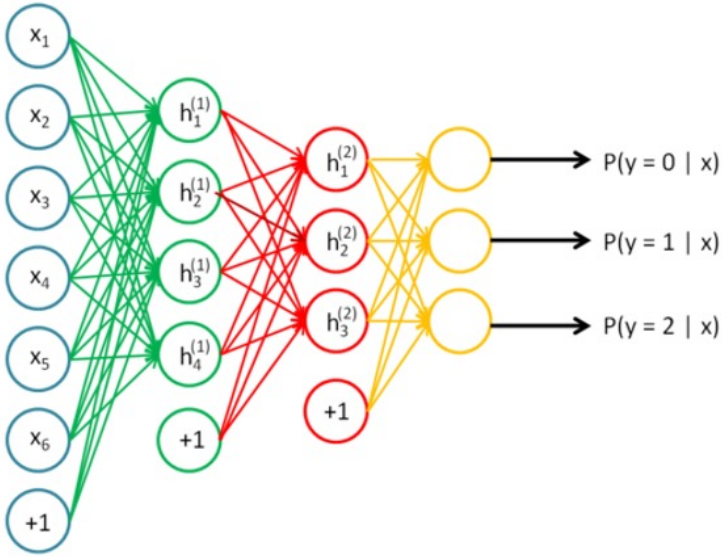


Figure 6: Examples of error-specific (*left*) and error-generic (*right*) evasion, as reported in [7]. Decision boundaries among the three classes (blue, red and green points) are shown as black lines. In the error-specific case, the initial (blue) sample is shifted towards the green class (selected as target). In the error-generic case, instead, it is shifted towards the red class, as it is the closest class to the initial sample. The gray circle represents the feasible domain, given as an upper bound on the ℓ_2 distance between the initial and the manipulated attack sample.

White-Box Evasion Attacks

- Fast Gradient Sign Method (FGSM)
  - Optimization objective
  - One step attack
  - Goodfellow et al. Explaining and Harnessing Adversarial Examples, 2015
- Iterative optimization attacks
  - Biggio et al. Evasion attacks against machine learning at test time, 2013 (SVM)
  - Szedegy et al. Intriguing properties of neural networks, 2014
  - Carlini and Wagner. Towards Evaluating the Robustness of Neural Networks, 2017

Evasion Attacks For Neural Networks



Optimization Formulation

Given input x

Find adversarial example

$x' = x + \delta$

$$\min_{\delta} L_t(x+\delta)$$

$$||\delta|| \leq d_{max}$$

- Most existing attacks are in continuous domains
- Optimization problem solved with gradient descent
- Attacks differ in objective formulation and method to solve optimization
    - Variants: maximize confidence or minimize distance

Black-Box Evasion Attacks

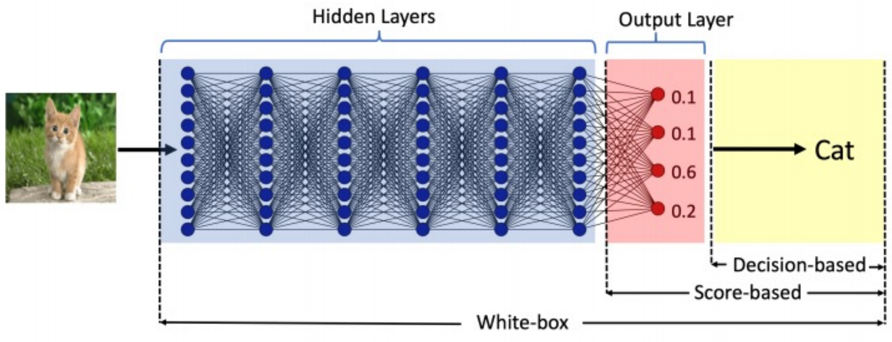
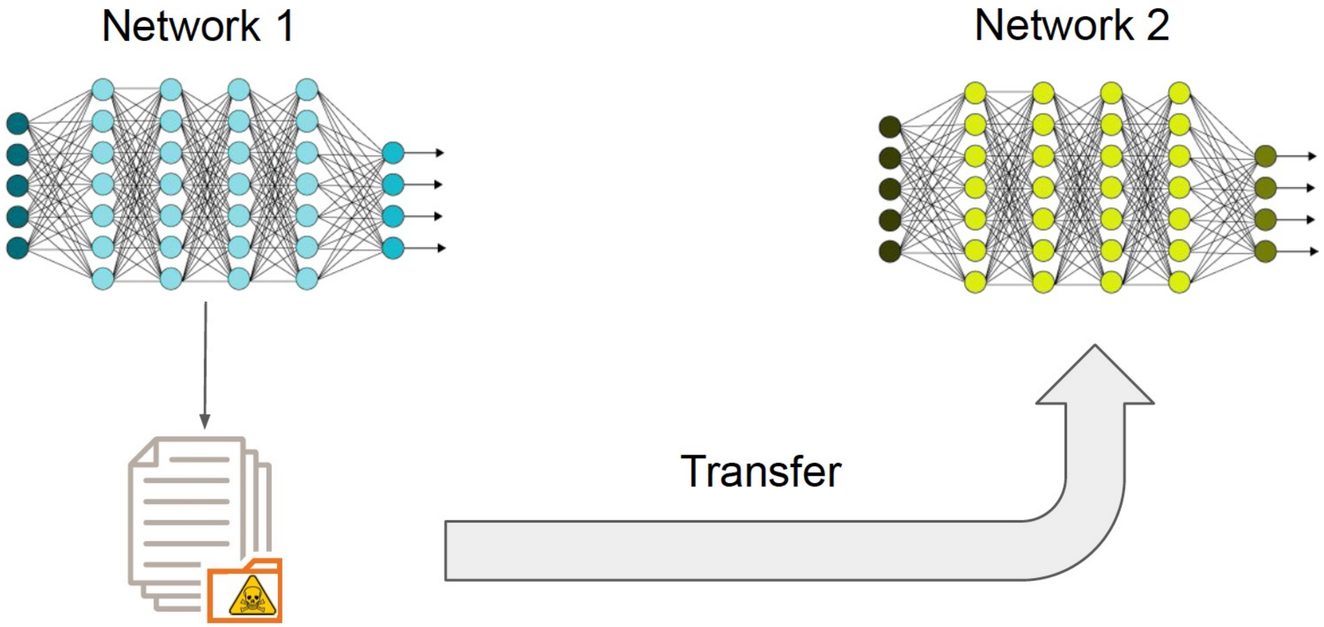


Figure 1: An illustration of accessible components of the target model for each of the three threat models. A white-box threat model assumes access to the whole model; a score-based threat model assumes access to the output layer; a decision-based threat model assumes access to the predicted label alone.

- Score-based attacks (adversary has access to probability vector of labels)
  - Zero-order optimization to approximate function gradient
  - Chen et al. Zoo: Zeroth order optimization based black-box attacks to DNNs without training substitute models, 2017.
  - Guo et al. Simple Black-box Adversarial Attacks, 2019
- Decision-based attacks (adversary has access to label only)
  - Chen et al. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. 2020

Transferability

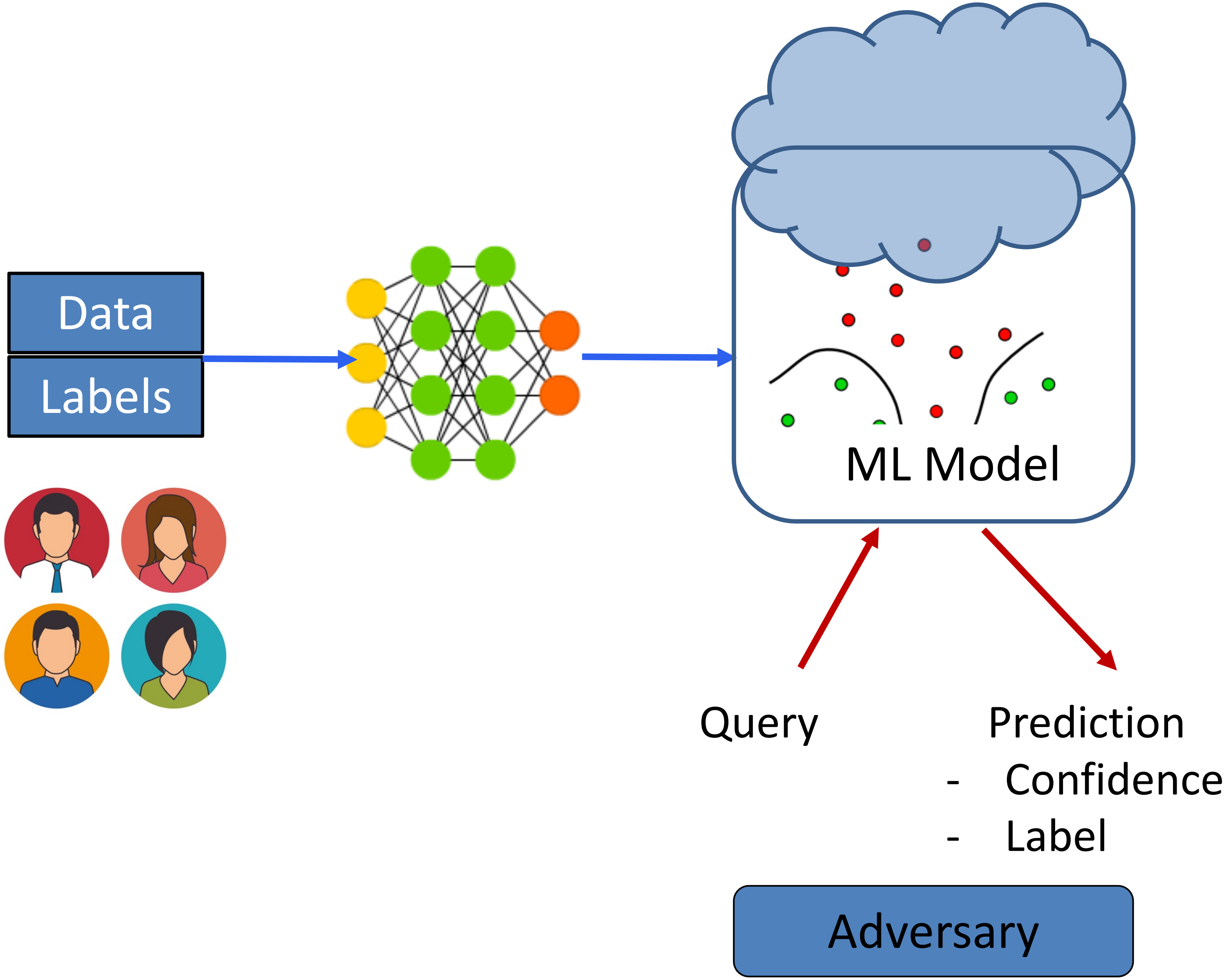


- Create own surrogate model, generate adversarial samples (poisoning or evasion) and transfer them to the target model
- Papernot et al. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples, 2016

Evasion Defenses

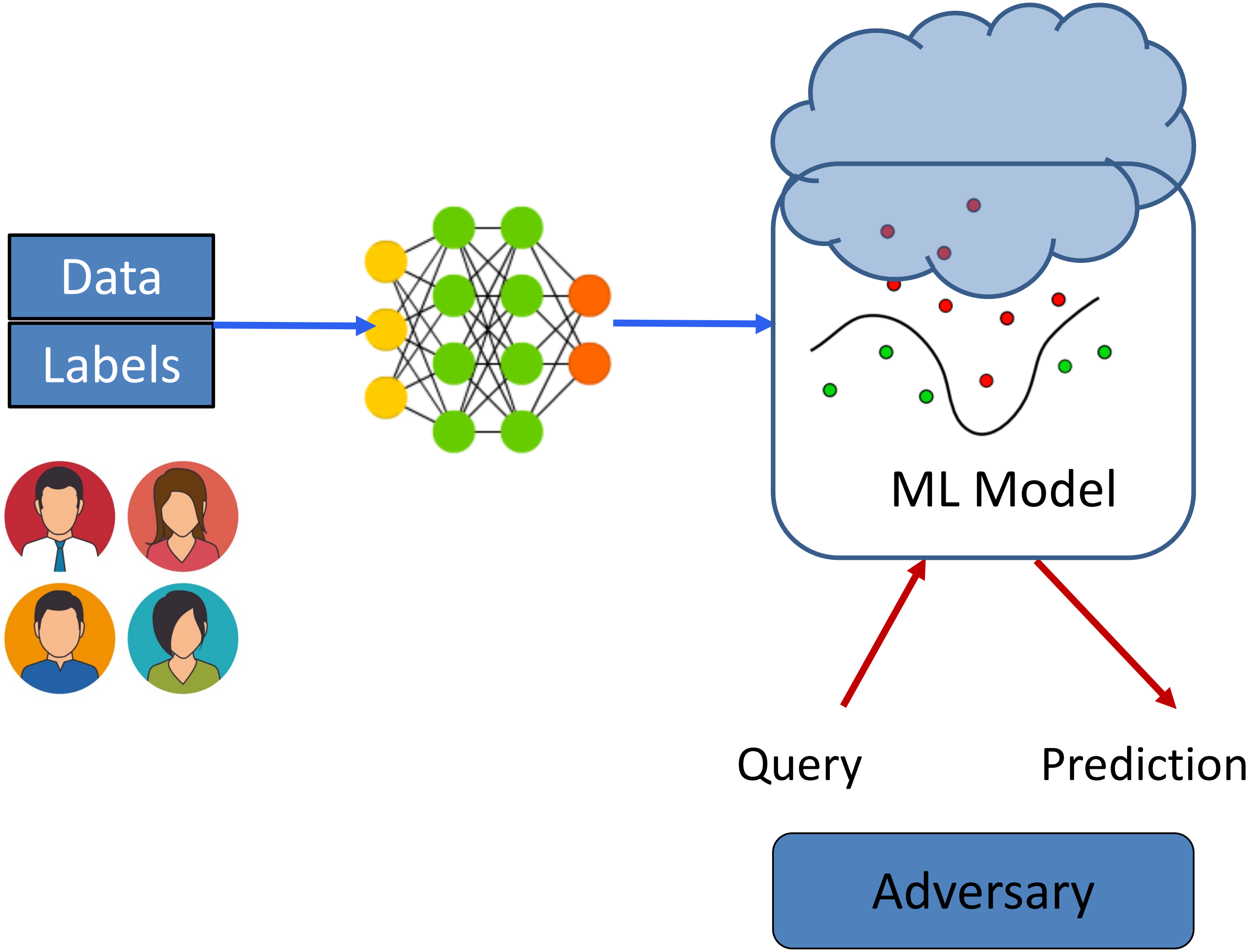
- Adversarial training
  - Godfellow et al. Explaining and Harnessing Adversarial Examples, 2014
  - Madry et al. Towards Deep Learning Models Resistant to Adversarial Attacks, 2017
- Certified defenses
  - Randomized smoothing: Cohen et al. Certified Adversarial Robustness via Randomized Smoothing, 2019
  - Perform prediction by majority voting over multiple samples centered around the sample of interest (add Gaussian noise to generate samples)
- Formal verification
  - Katz et al. Reluplex: An efficient SMT solver for verifying deep neural networks, 2017
  - Gehr et al. AI2 : Safety and Robustness Certification of Neural Networks with Abstract Interpretation, 2018

Privacy Attacks in ML



- ML model is trained by third-party collecting user data
- **Black-box**
  - Query access to model
  - Model returns confidence (probability of prediction) or only predicted label
- What can the adversary learn about the training set?

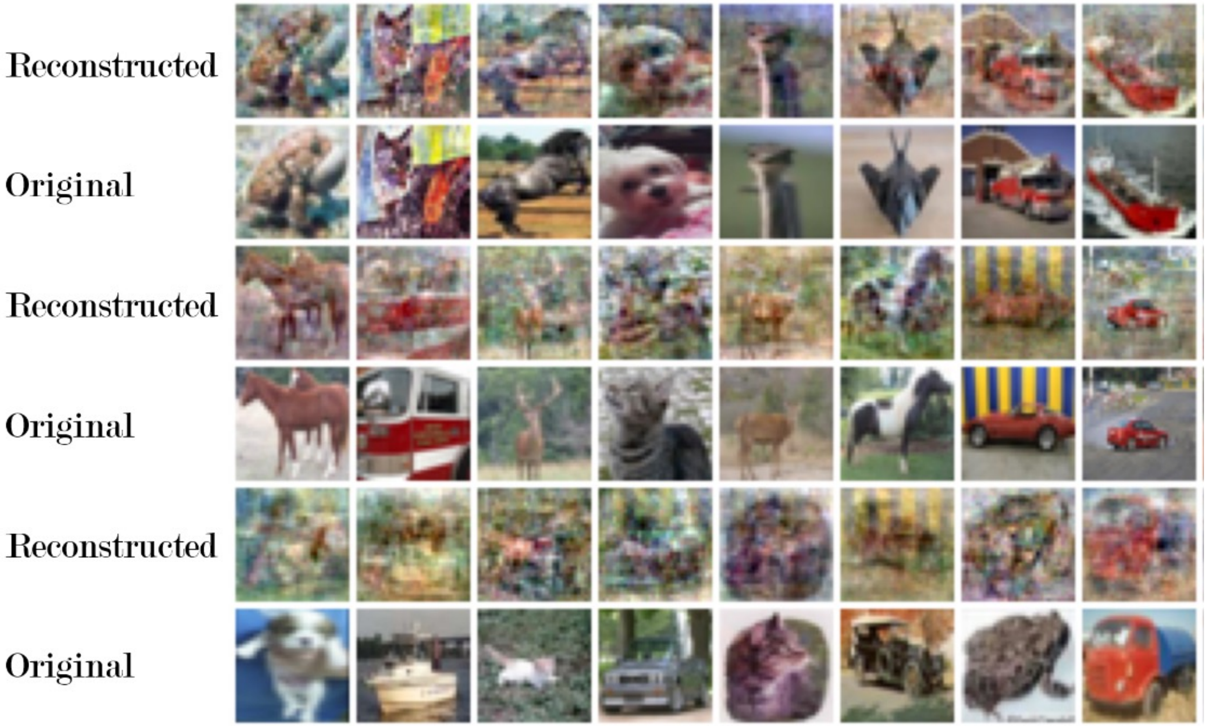
Privacy Attacks in ML



- Reconstruction: Extract sensitive data from training sets
  - Statistical databases: [DN03]
  - DNNs: [HVY22]
  - LLM memorization: [CTW21]
- Membership Inference: Determine if data sample was in training set
  - [SSS17], [YGF18], [CCN22]
- Property Inference: Learn global properties about the training set
  - [MGC22], [CAO23]

Training Data Reconstruction

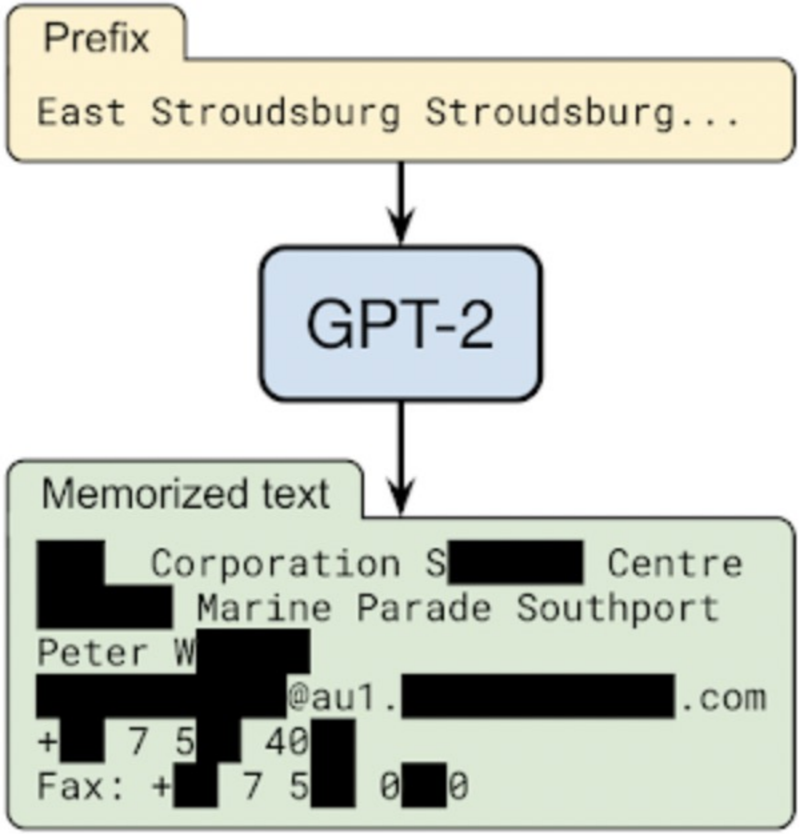
- Reconstruct dataset from multiple linear queries [DN03]
- Reconstruct training set from the parameters of a neural network [HVY22]
  - Simple MLP models
  - Main insight: parameters and training set satisfy a set of constraints (at convergence)



[HVY22] Haim et al. *Reconstructing Training Data From Trained Neural Networks*. NeurIPS 2022

Large Language Model (LLM) Memorization

- Training data extraction attack on generative language models
  - Prompt model with different prefixes and measure perplexity of generated text
  - Identified hundreds of memorized examples on GPT-2
- Memorization increases with model size
- Memorization increases with number of repetitions



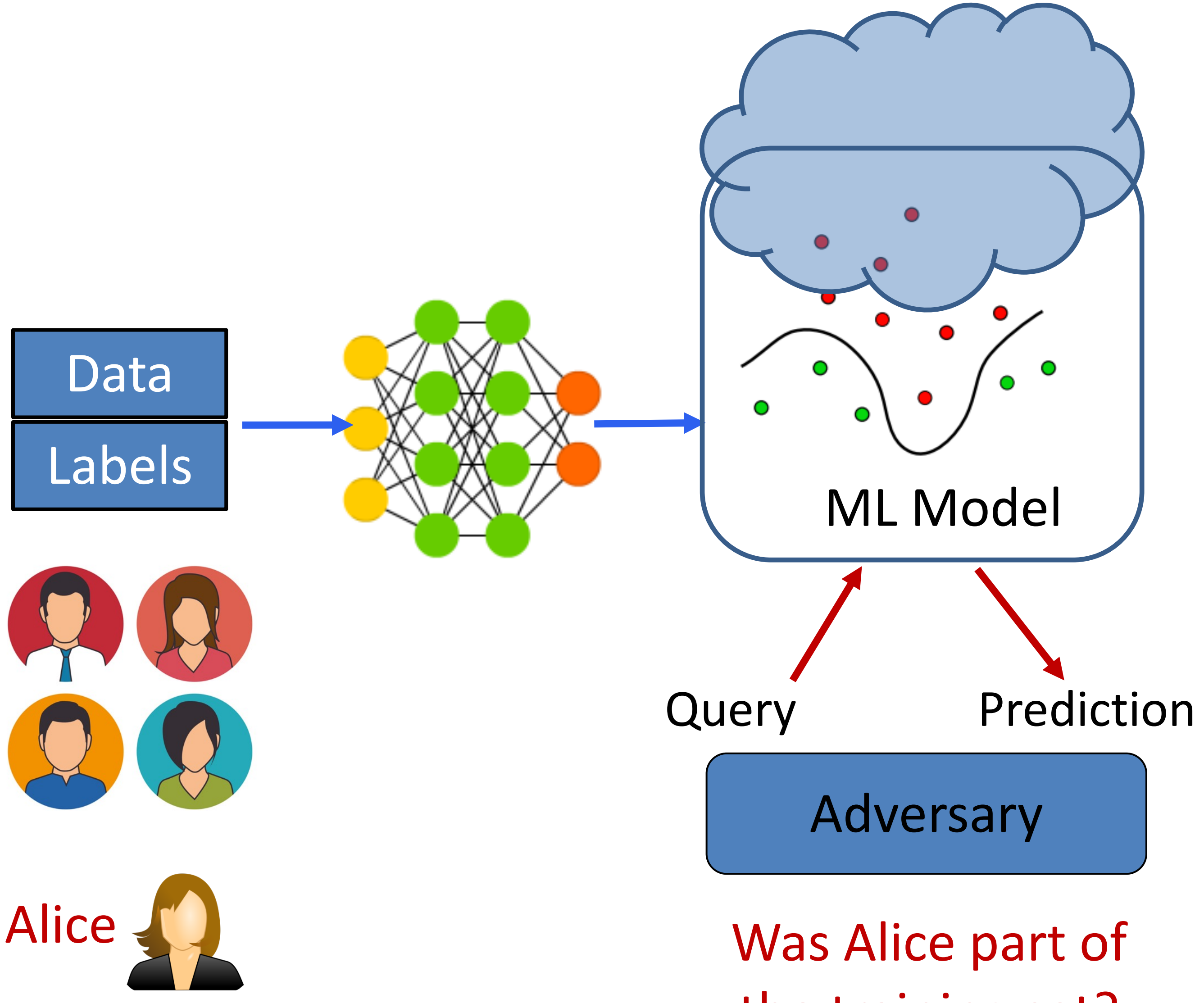
East Stroudsburg Stroudsburg...

[redacted] Corporation S[redacted] Centre
Marine Parade Southport
Peter W[redacted]
[redacted]@au1.[redacted].com
+[redacted] 7 5[redacted] 40[redacted]
Fax: +[redacted] 7 5[redacted] 0[redacted]0

[CTW21] Carlini et al. *Extracting Training Data from Large Language Models*. USENIX Security 2021

Membership Inference

- Learn if a user participated in training set of model
  - Being part of ML training set might be sensitive
- Introduced for statistical computations on genomic data [HSR08]
- First membership inference attack on DNNs [SSS17]
- More efficient attacks [YGF18], [CCN22]

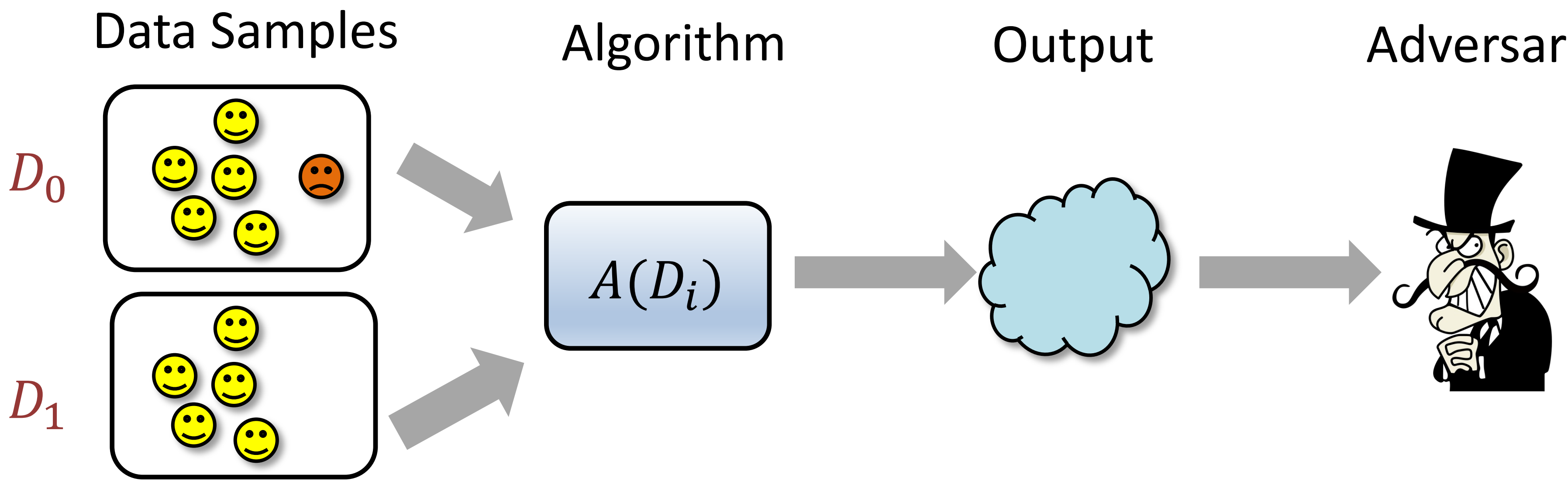


Was Alice part of the training set?

How to Protect User Privacy?

- Option 1: Do not participate in ML training!
- Option 2: Curate training data before training ML models
  - In reality, it might prove impossible as sensitive information is hard to identify automatically
- Option 3: Use Differential Privacy (DP) for ML training
  - Warning: It often comes at a loss in accuracy
  - Does not protect against all privacy attacks (e.g., Property Inference reveals global properties about training set)
  - Understand how to select hyper-parameters (e.g., clipping norm, amount of noise) is non-trivial

Differential Privacy



Definition: A is ε-DP if

Worst-case privacy

$$\text{for all } (D_0, D_1, R)$$

$$\underbrace{\mathbb{P}(\mathcal{A}(D_1) \in R)}_{\text{True Positive Rate}} \leq e^{\varepsilon} \underbrace{\mathbb{P}(\mathcal{A}(D_0) \in R)}_{\text{False Positive Rate}} + \delta$$

Model Extraction

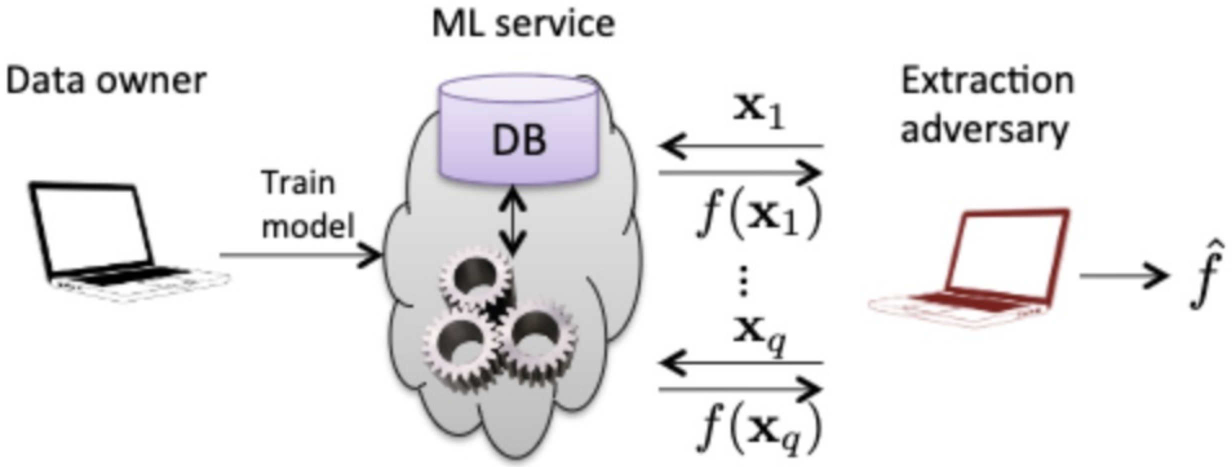


Figure 1: **Diagram of ML model extraction attacks.** A data owner has a model f trained on its data and allows others to make prediction queries. An adversary uses q prediction queries to extract an $\hat{f} \approx f$.

- Tramer et al. Stealing Machine Learning Models via Prediction APIs, 2017
- Jagielski et al. High Accuracy and High Fidelity Extraction of Neural Networks, 2020
- Can be used to mount white-box attacks (e.g., evasion attacks)

Fairness in ML

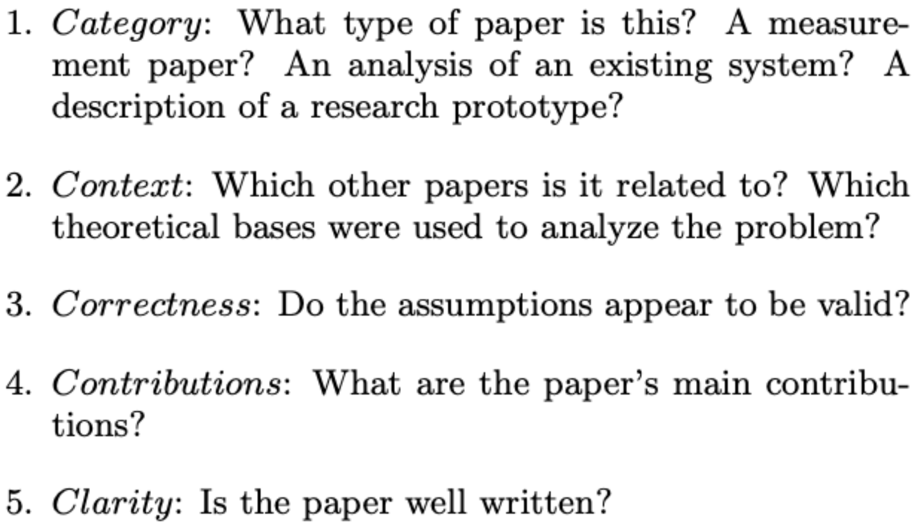
- Case studies
  - COMPAS algorithm to predict who will reoffend
    - Same accuracy across groups, but errors were different
  - Hiring algorithms (Amazon)
- Predictions of model should be "similar" for different groups (defined by sensitive attributes)
  - Different definitions of fairness: demographic parity (prediction independent on sensitive attribute), equalized odds (equal False Positive and False Negative rates in the groups), predictive parity (equal precision in the groups)
- Impossibility Results
  - Can only satisfy 2 out of the 3 definitions above, but not all 3!
- Tools:
  - IBM: https://www.ibm.com/opensource/open/projects/ai-fairness-360/
  - TensorFlow: https://www.tensorflow.org/tfx/guide/fairness_indicators
  - Microsoft FairLearn: https://github.com/fairlearn/fairlearn

Other References

- Barreno et al. The security of machine learning, 2010
- Huang et al. Adversarial machine learning, 2011
- Cummings et al. Challenges towards the Next Frontier in Privacy, 2023. https://arxiv.org/pdf/2304.06929.pdf
- Solon Barocas, Moritz Hardt, Arvind Narayanan. Fairness and Machine Learning. Limitations and Opportunities

S. Keshav. How to Read a Paper

- Three-pass approach
- Pass 1: Title, abstract, and introduction, section headings, conclusions



1. *Category*: What type of paper is this? A measurement paper? An analysis of an existing system? A description of a research prototype?

2. *Context*: Which other papers is it related to? Which theoretical bases were used to analyze the problem?

3. *Correctness*: Do the assumptions appear to be valid?

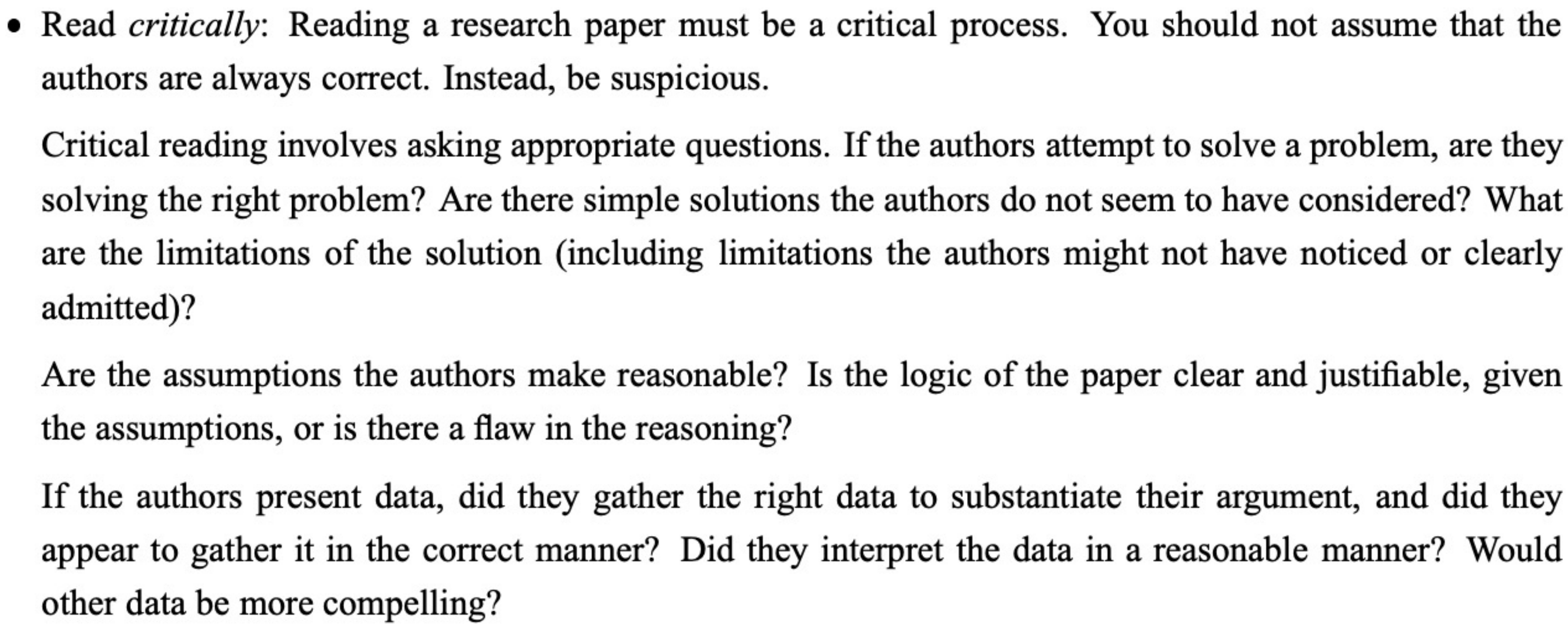
4. *Contributions*: What are the paper's main contributions?

5. *Clarity*: Is the paper well written?

S. Keshav. How to Read a Paper

- Three-pass approach
- Pass 2: Read paper, but not dive into some technical details (e.g., proofs)
  - Be able to summarize the content
  - Strengths and limitations
  - Related research
- Pass 3: Read in full details (be able to reimplement it)

M. Mitzenmacher. How to read a research paper



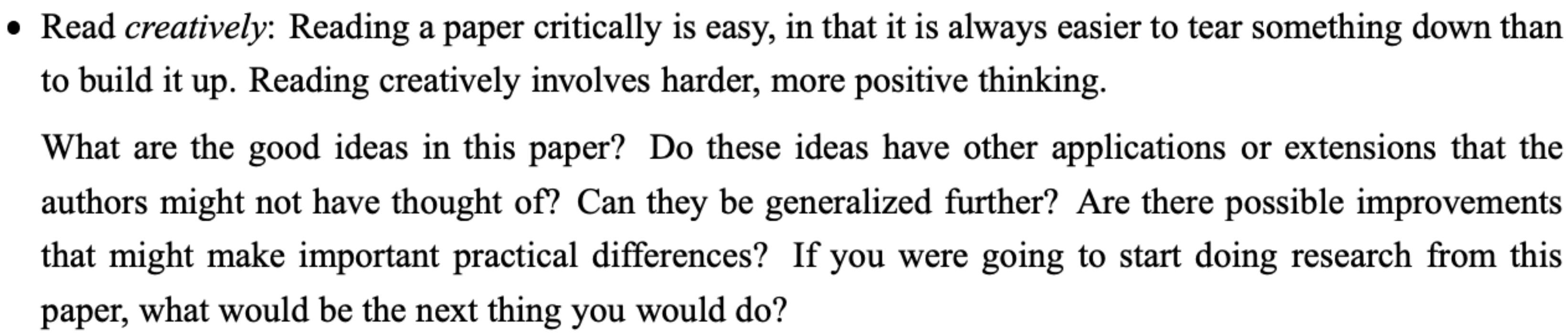
- Read *critically*: Reading a research paper must be a critical process. You should not assume that the authors are always correct. Instead, be suspicious.

  Critical reading involves asking appropriate questions. If the authors attempt to solve a problem, are they solving the right problem? Are there simple solutions the authors do not seem to have considered? What are the limitations of the solution (including limitations the authors might not have noticed or clearly admitted)?

  Are the assumptions the authors make reasonable? Is the logic of the paper clear and justifiable, given the assumptions, or is there a flaw in the reasoning?

  If the authors present data, did they gather the right data to substantiate their argument, and did they appear to gather it in the correct manner? Did they interpret the data in a reasonable manner? Would other data be more compelling?

M. Mitzenmacher. How to read a research paper



- Read *creatively*: Reading a paper critically is easy, in that it is always easier to tear something down than to build it up. Reading creatively involves harder, more positive thinking.

  What are the good ideas in this paper? Do these ideas have other applications or extensions that the authors might not have thought of? Can they be generalized further? Are there possible improvements that might make important practical differences? If you were going to start doing research from this paper, what would be the next thing you would do?

Template for Paper Summaries

Instructions: Write 1-2 sentences for each paragraph in a concise manner. The summary should not exceed one page.

Problem Statement

- What is the problem the paper is addressing?

Threat Model

- What is the adversarial model the paper considers?
- Define the adversarial objectives, knowledge, and capabilities

Methodology

- How is the problem solved?
- What is the main technical contribution?
- Are any techniques in the solution new relative to existing work?

Strengths

- What are the main strengths of the paper? For example:
    - Is it the first paper to define the problem and solve it?
    - Does it offer a better solution to an existing problem?
    - Is the evaluation comprehensive?

Limitations

- What are some of the limitations of the paper? For example:
    - What scenarios the solution does not address?
    - Are there any simplifying assumptions?
    - Are there simpler solutions the authors did not consider?

Discussion

- What are some ideas for follow up research?
- Can some of the techniques be generalized or applied in other domains?
- Can you think of a better solution to solve the problem?
- What is the impact of the work?

Paper Discussion

- Discussion leads responsibilities
  - Prepare slides for the presentation in class
  - Introduce the paper (problem, threat model, methodology)
  - Prepare list of points to discuss, strength and limitations
  - Find at least one more reference on the topic and discuss it
- Everyone else
  - Read the papers, submit summaries and bring discussion points to class
  - Participate in interactive class discussion