

CY 7790

Special Topics in Security and Privacy:
Machine Learning Security and
Privacy
Fall 2021

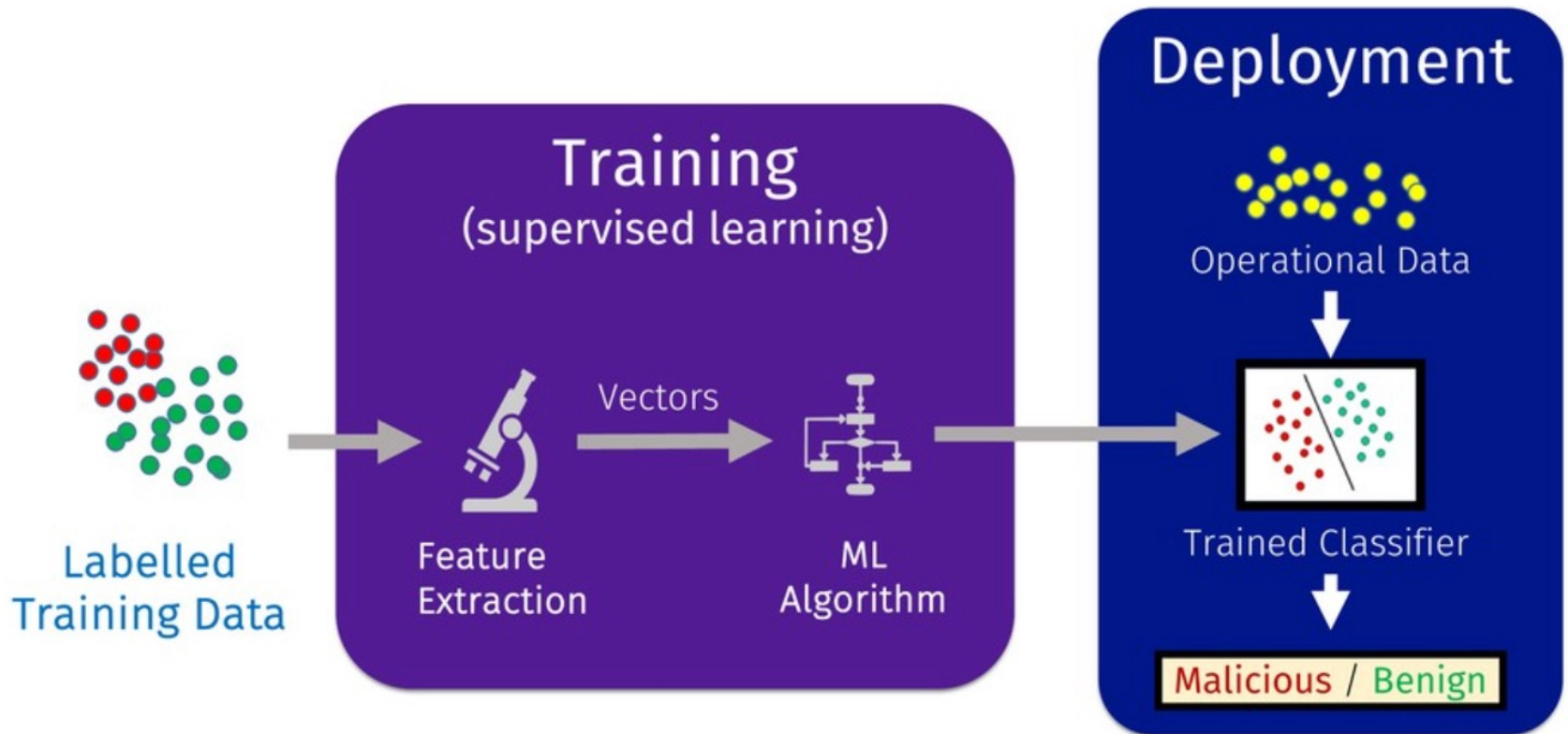
Alina Oprea
Associate Professor
Khoury College of Computer Science

September 23 2021

Outline

- Metrics to evaluate classifiers
- Adversarial ML taxonomy and history
 - N. Papernot et al. SoK: Security and Privacy in Machine Learning
 - B. Biggio and F. Roli: Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning
- How to read research papers
- Template for paper summaries

ML Pipeline



How to Measure Classifiers?

Given a dataset of P positive instances and N negative instances:

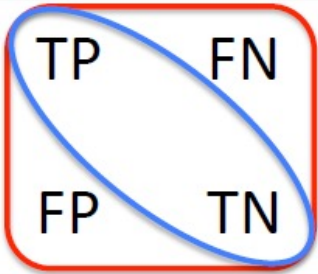
		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

Confusion Matrix

Accuracy and Error

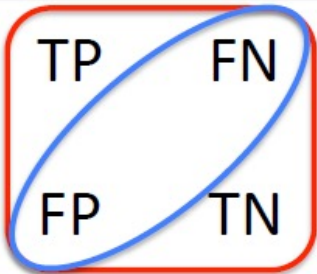
Given a dataset of P positive instances and N negative instances:

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN



$$\text{accuracy} = \frac{TP + TN}{P + N}$$

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN



$$\begin{aligned}\text{error} &= 1 - \frac{TP + TN}{P + N} \\ &= \frac{FP + FN}{P + N}\end{aligned}$$

Confusion Matrix

- Given a dataset of P positive instances and N negative instances:

$FPR = \frac{FP}{P + FP}$
 $FNR = \frac{FN}{N + FN}$

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{accuracy} = \frac{TP + TN}{P + N}$$

f1 score

- Imagine using classifier to identify positive cases (i.e., for information retrieval)

$$\text{precision} = \frac{TP}{TP + FP}$$

Probability that classifier predicts positive correctly

$$\text{recall} = \frac{TP}{TP + FN}$$

Probability that actual class is predicted correctly

Classifiers can be tuned

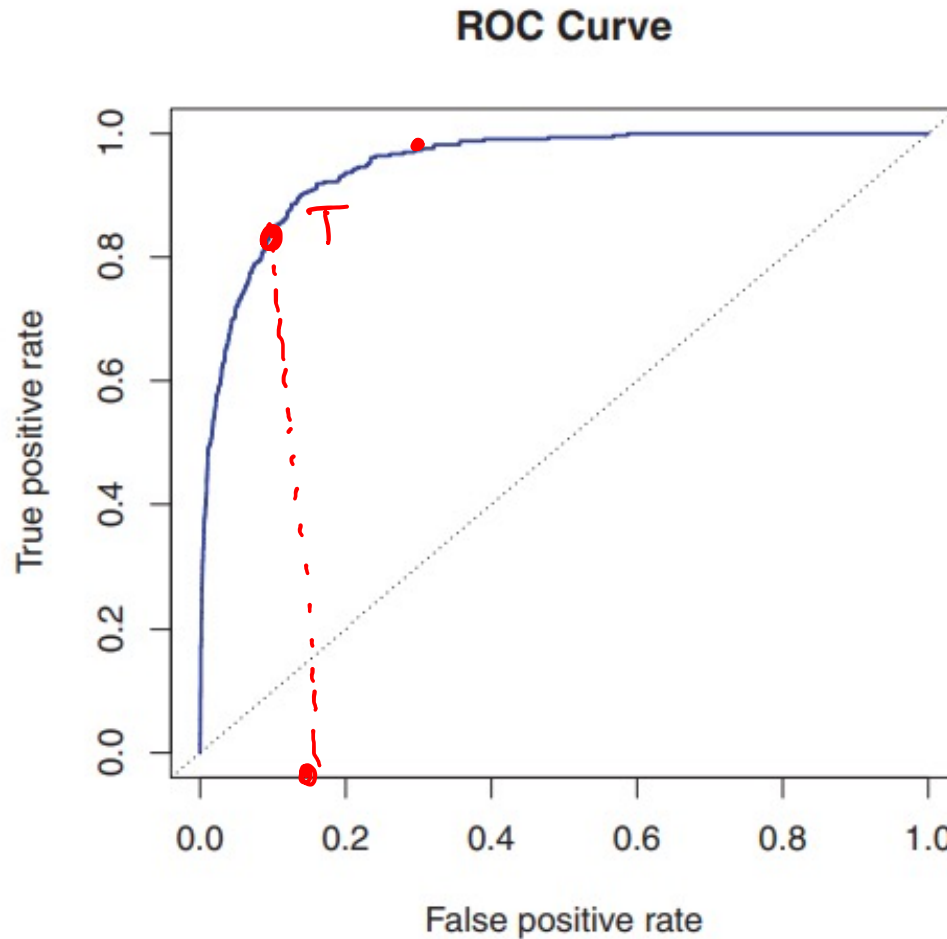
- Logistic regression sets by default the threshold at 0.5 for classifying positive and negative instances
- Some applications have strict constraints on false positives (or other metrics)
 - Example: very low false positives in security (spam)
- Solution: choose different threshold

$T \uparrow \Rightarrow$ Recall \downarrow
FP \downarrow
Prec \uparrow

Probabilistic model $h_{\theta}(x) = P[y = 1|x; \theta]$

- Predict $y = 1$ if $h_{\theta}(x) \geq T$
- Predict $y = 0$ if $h_{\theta}(x) < T$

ROC Curves



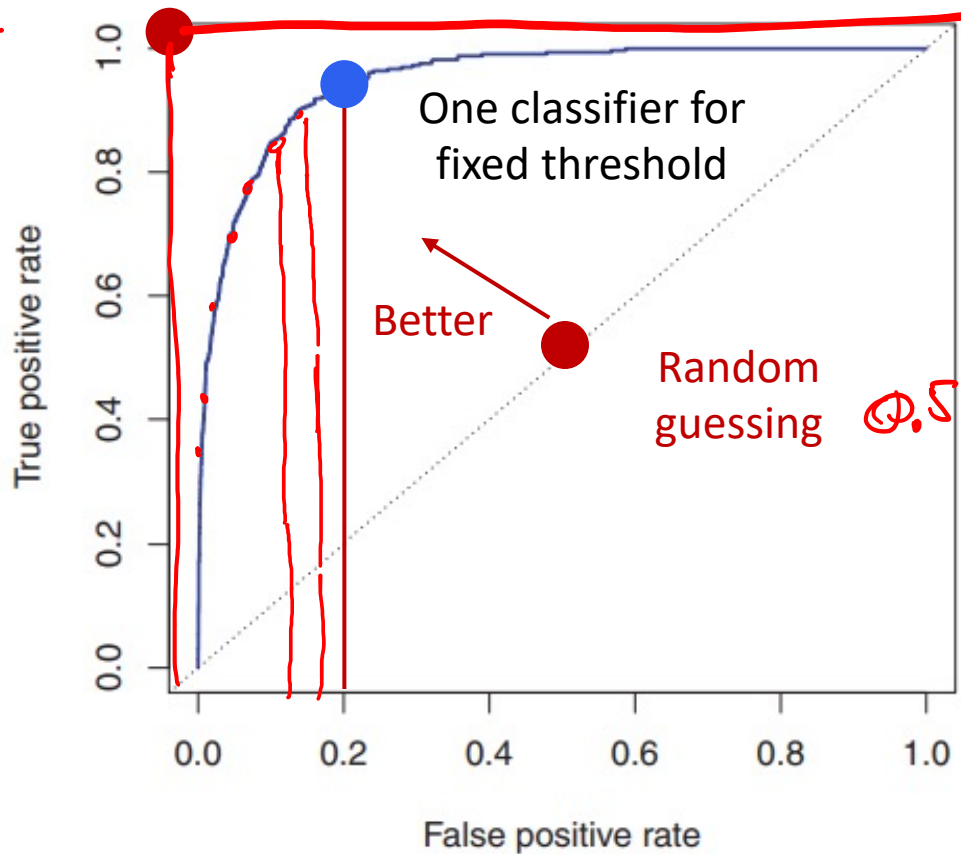
- Receiver Operating Characteristic (ROC)
- Determine operating point (e.g., by fixing false positive rate)

ROC Curves

Perfect
classification



ROC Curve



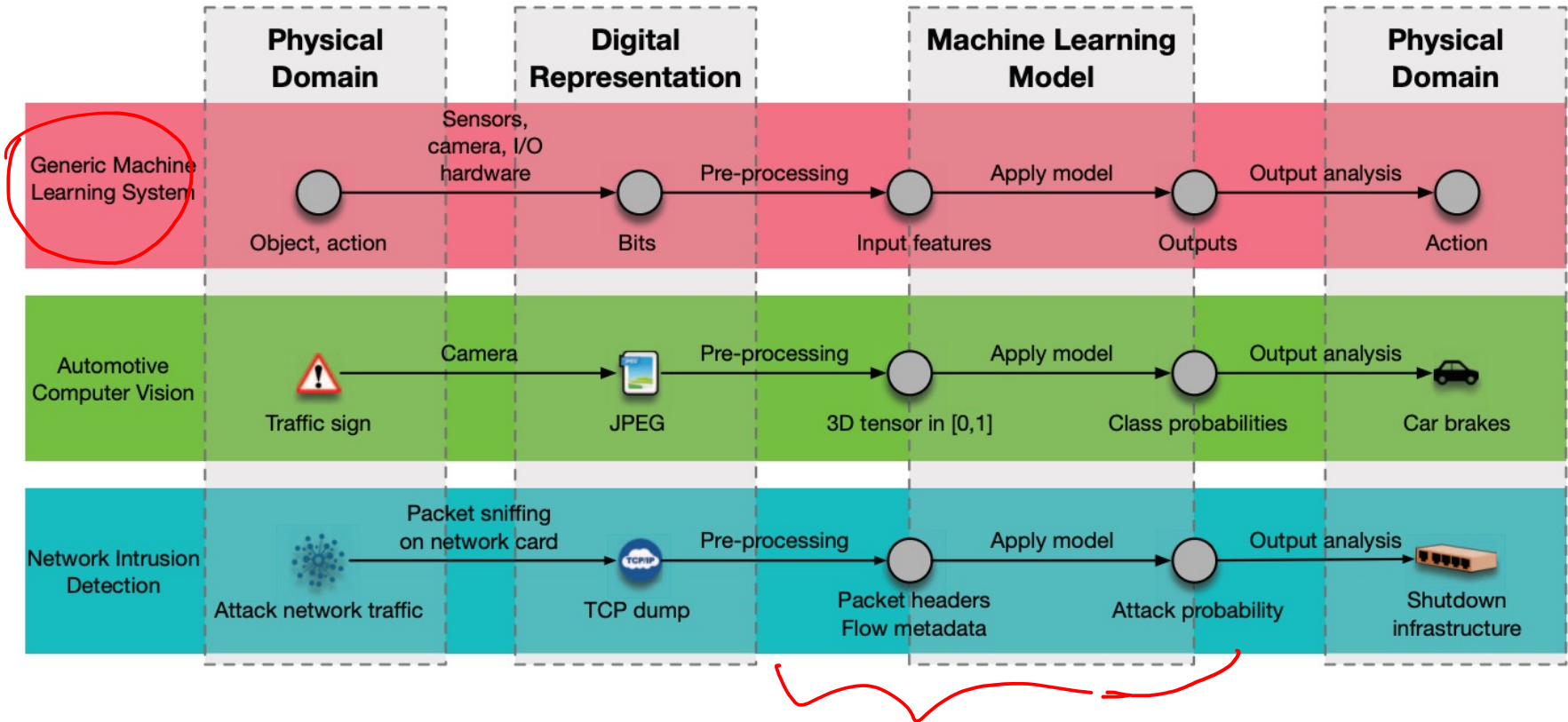
Precision -
Recall
curves

- Another useful metric: Area Under the Curve (AUC)

Adversarial Machine Learning


Taxonomy and History

ML Attack Surface




Attacks need to be translated from feature space to physical domain to have an impact in the real world

Adversarial Goals

- 
- Confidentiality and Privacy
 - Extract information about data and model
 - Integrity and Availability
 - Control model output
 - Selective control for specific inputs: targeted attack
 - Entire control of model: availability attack (denial of service)

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	X ⁻
	Testing	Evasion Attacks 	Sponge Adversarial Examples	Reconstruction Membership Inference Model Extraction

Threat Model

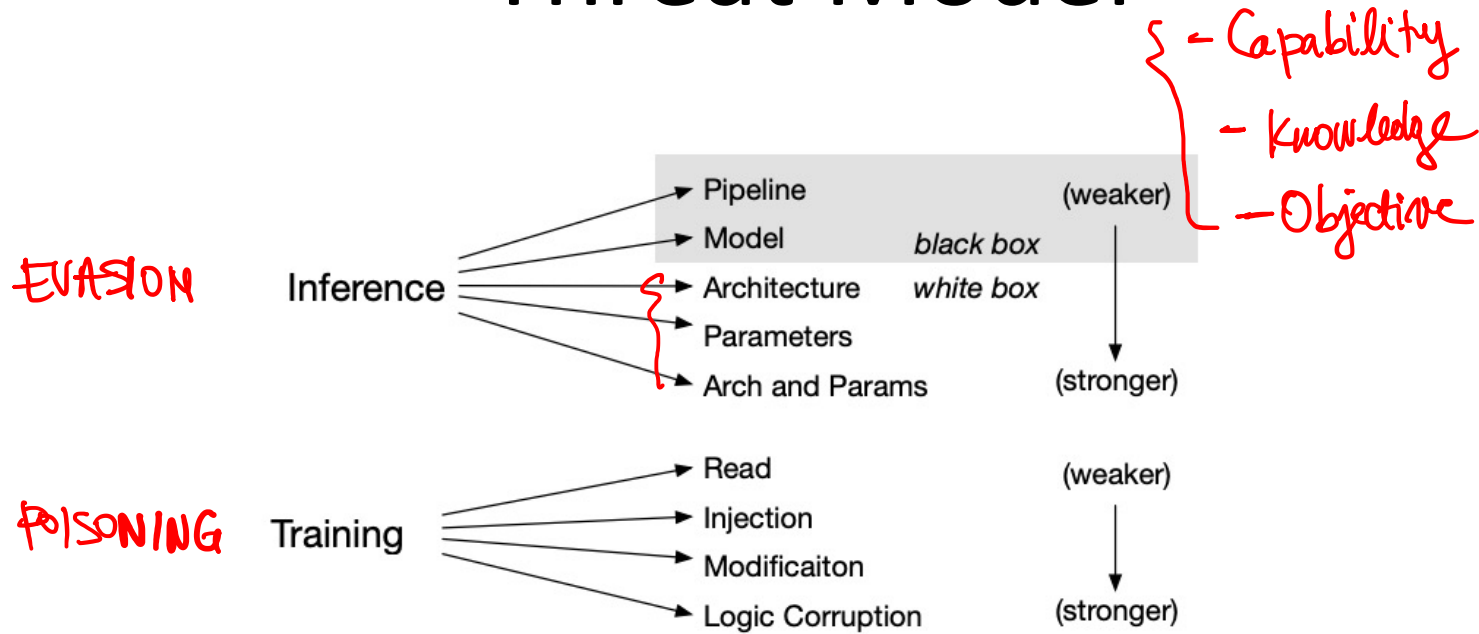
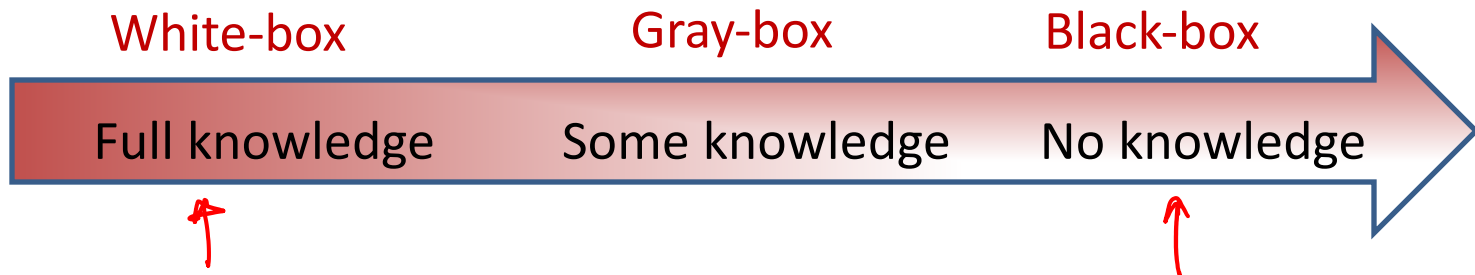


Figure 3. **Adversarial Capabilities.** Adversaries attack ML systems at inference time by exploiting model internal information (white box) or probing the system to infer system vulnerabilities (black box). Adversaries use read or write access to the training data to mimic or corrupt the model.



Poisoning (Training-Time) Attacks

- ML is trained by crowdsourcing data in many applications

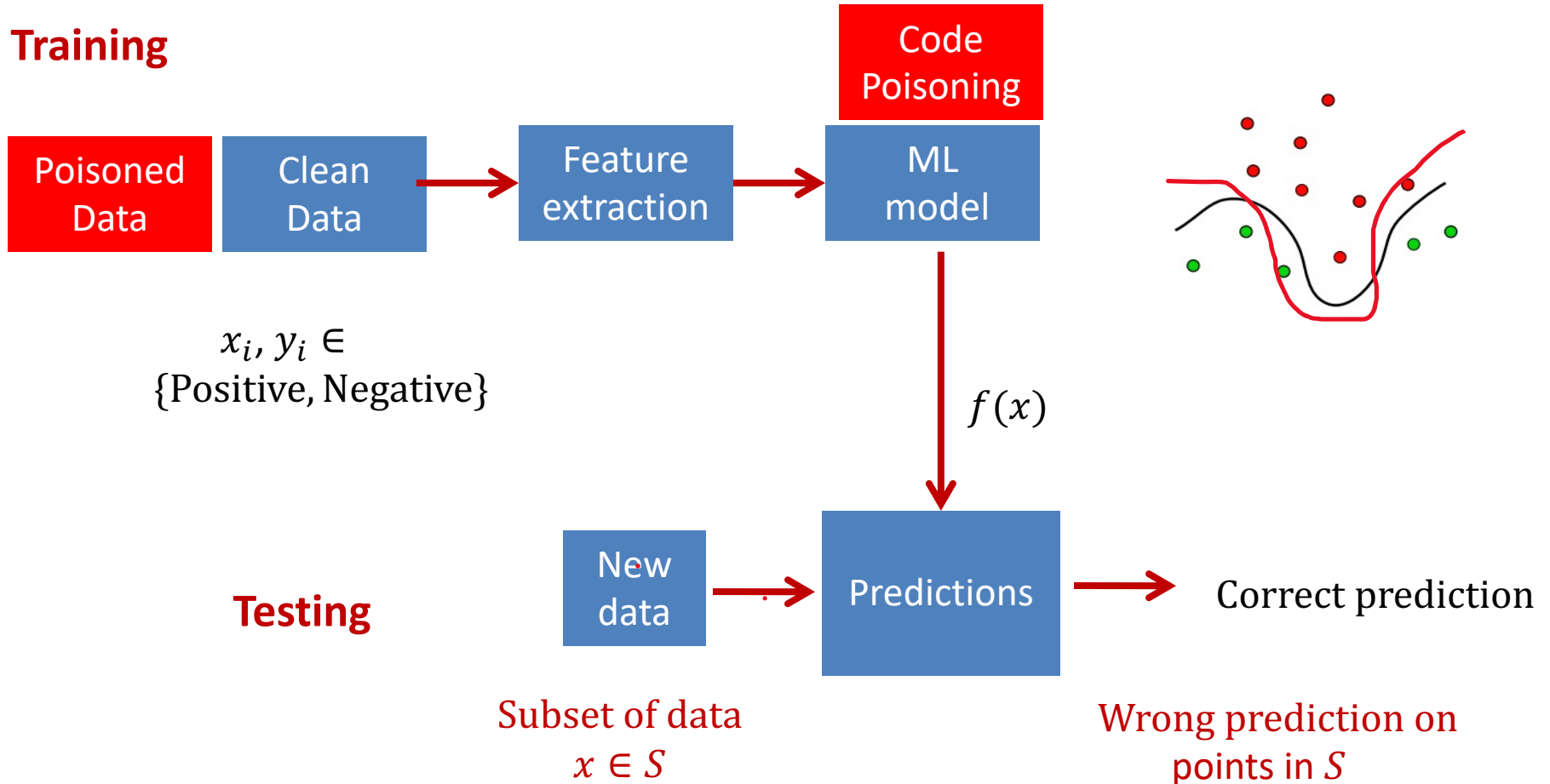
- Social networks
- News articles
- Tweets
- Photos
- Binary files



- Cannot fully trust training data!



Poisoning Attacks



- Poisoning attack inserts corrupted data at training, modify existing data, or change 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

Results of poisoning - indiscriminate

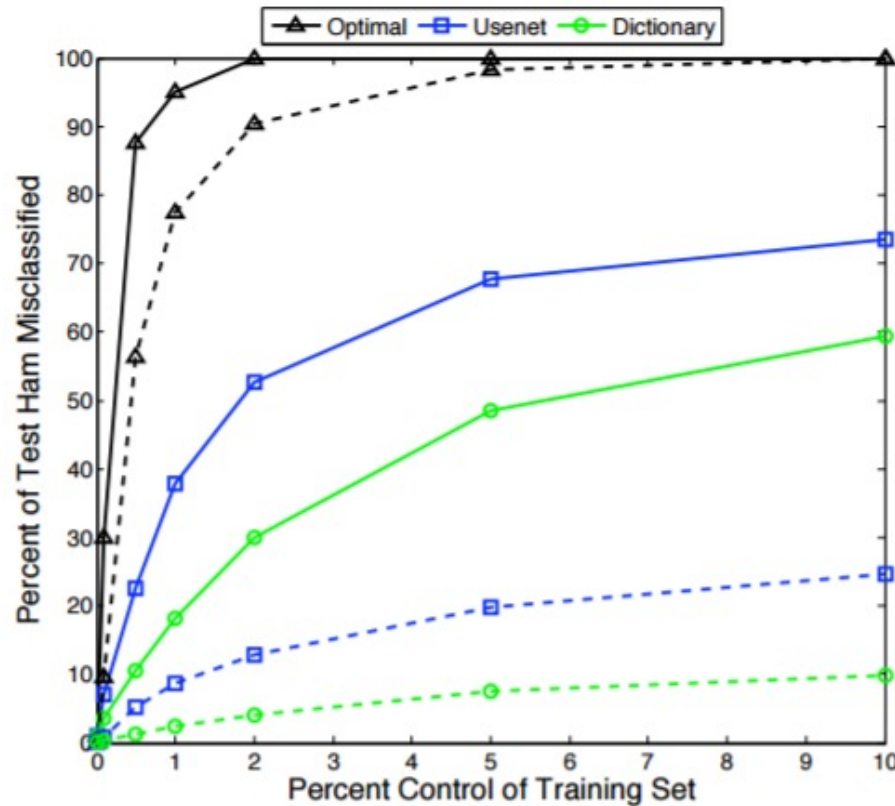


Figure 1: Three dictionary attacks on initial training set of 10,000 messages (50% spam). We plot percent of ham classified as *spam* (dashed lines) and as *spam* or *unsure* (solid lines) against the attack as percent of the training set. We show the optimal attack (black \triangle), the Usenet dictionary attack (blue \square), and the Aspell dictionary attack (green \circ). Each attack renders the filter unusable with as little as 1% control (101 messages).

Results of poisoning - targeted

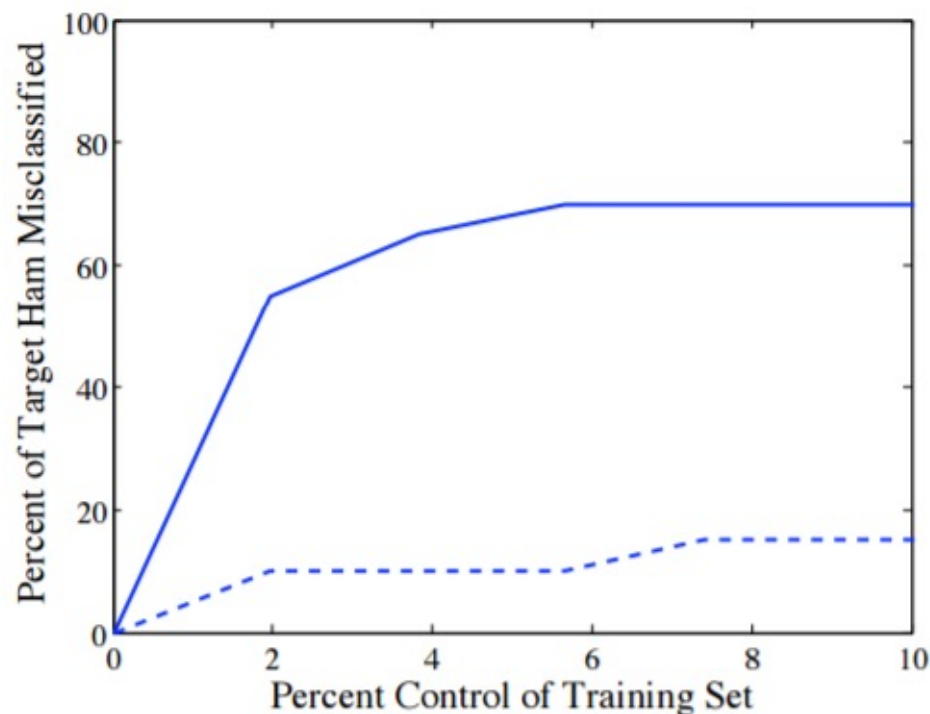


Figure 3: Effect of the focused attack as a function of the number of attack emails with a fixed probability ($p=0.5$) that the attacker guesses each token. The dashed line shows the percentage of target ham messages misclassified as *spam* after the attack, and the solid line the percentage of targets that are misclassified as *unsure* or *spam* after the attack. The initial inbox contains 5,000 emails (50% spam).

Poisoning SVMs

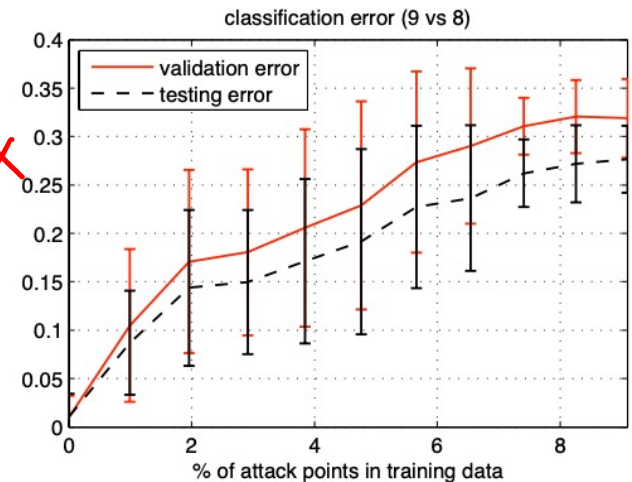
- **Label Manipulation**

- 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
- Biggio et al. Poisoning Attacks against Support Vector Machines, 2012

WHITE-BOX

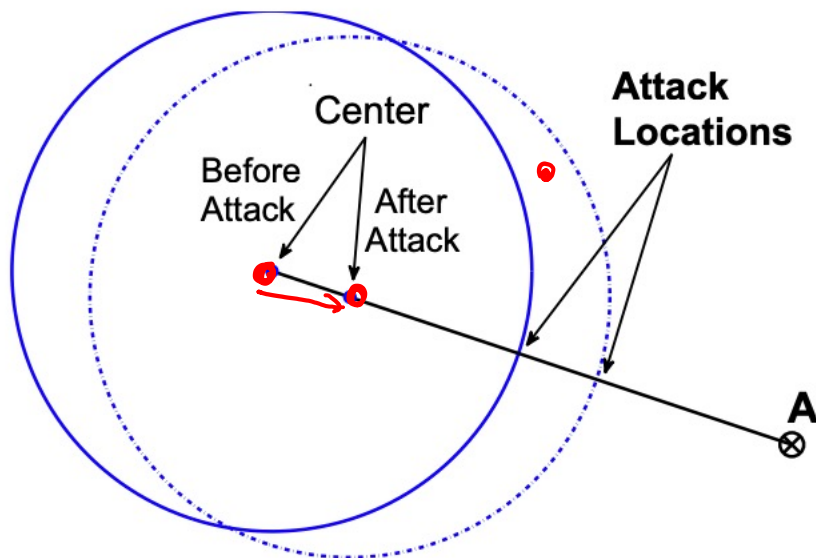


Poisoning Anomaly Detection

- Centroid-based outlier detection for online learning
- Kloft and Laskov. Security Analysis of Online Centroid Anomaly Detection, 2012

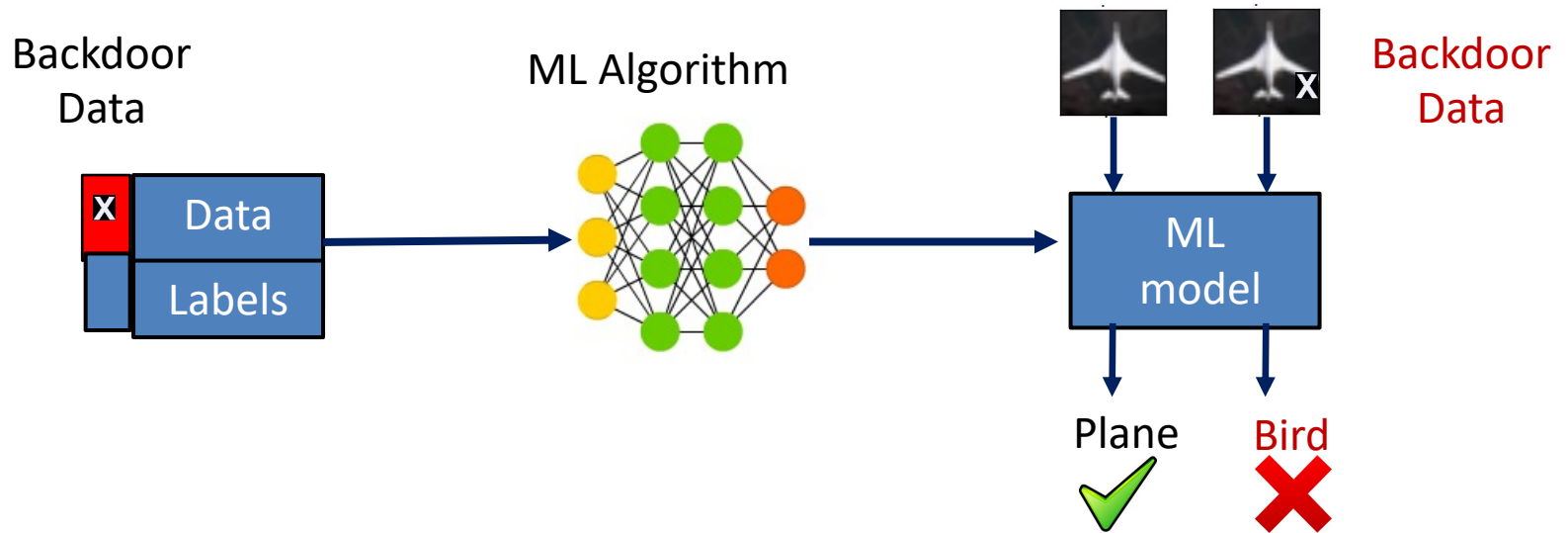
$$f(\mathbf{x}) = \left\| \mathbf{x} - \underbrace{\frac{1}{n} \sum_{j=1}^n \mathbf{x}_0^{(j)}}_{\text{centroid}} \right\|.$$

$\rightarrow T$ \rightarrow anomaly
 $< T$ \rightarrow normal

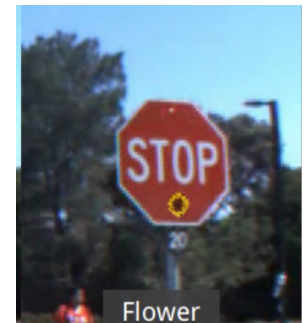


- If the distance to centroid exceed threshold, then mark as outlier
- Attacker can shift centroid by poisoning data and move center in certain direction

Backdoor Poisoning Attacks



- **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]



Poisoning Defenses

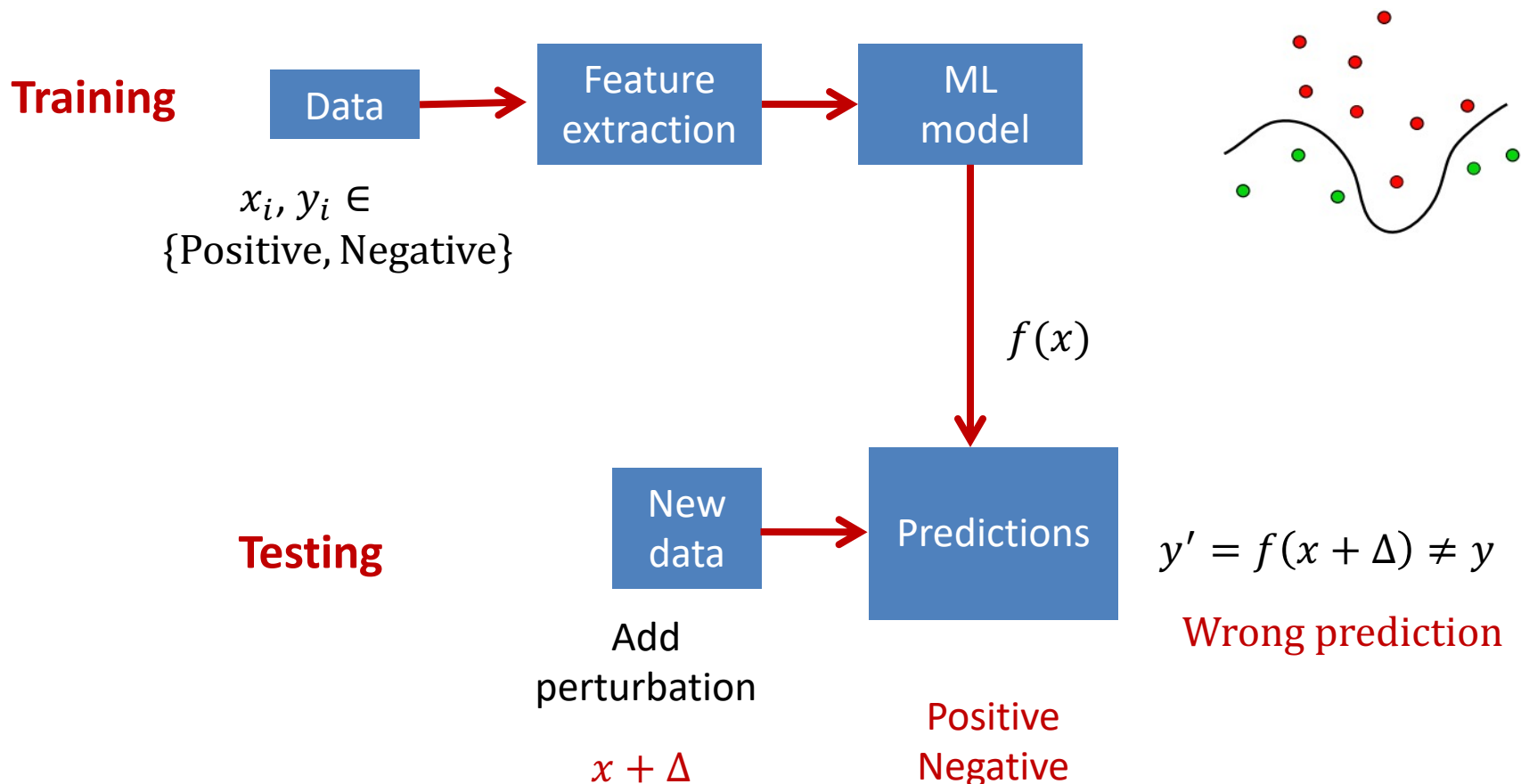
- Data sanitization

- RONI (reject on negative impact) – measures impact on classification for each instance and removes instances that increase error *SPAM*
- Robust statistics – remove outliers from data; use trimmed loss function [Steinhardt 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]
- Inspect model and determine if it was backdoored: ABS [Liu et al. 2019]
- Remove outliers from representation layer: spectral signatures [Tran et al. 2018]

Evasion Attacks



- Modify testing point by adding small perturbation to misclassify it

Adversarial Example

Targeted

Prediction Change Definition:

An input, $x' \in \mathcal{X}$, is an **adversarial example** for $x \in \mathcal{X}$, iff
 $\exists x' \in \text{Ball}_\epsilon(x)$ such that $f(x) \neq f(x')$.

Without constraints on Ball_ϵ , every input has adversarial examples.

$\text{Ball}_\epsilon(x)$ is some space around x , typically defined in some (simple!) metric space:

L_0 norm (# different), L_2 norm ("Euclidean distance"), L_∞

$$\left[\begin{array}{l} 1) \quad \|x' - x\| \leq \epsilon \\ 2) \quad f(x) \neq f(x') \end{array} \right.$$

Untargeted Adversarial Examples



x

“panda”

57.7% confidence

X

+ .007 ×



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

“nematode”

8.2% confidence

X'

=



$x +$

$\epsilon \text{sign}(\nabla_x J(\theta, x, y))$

“gibbon”

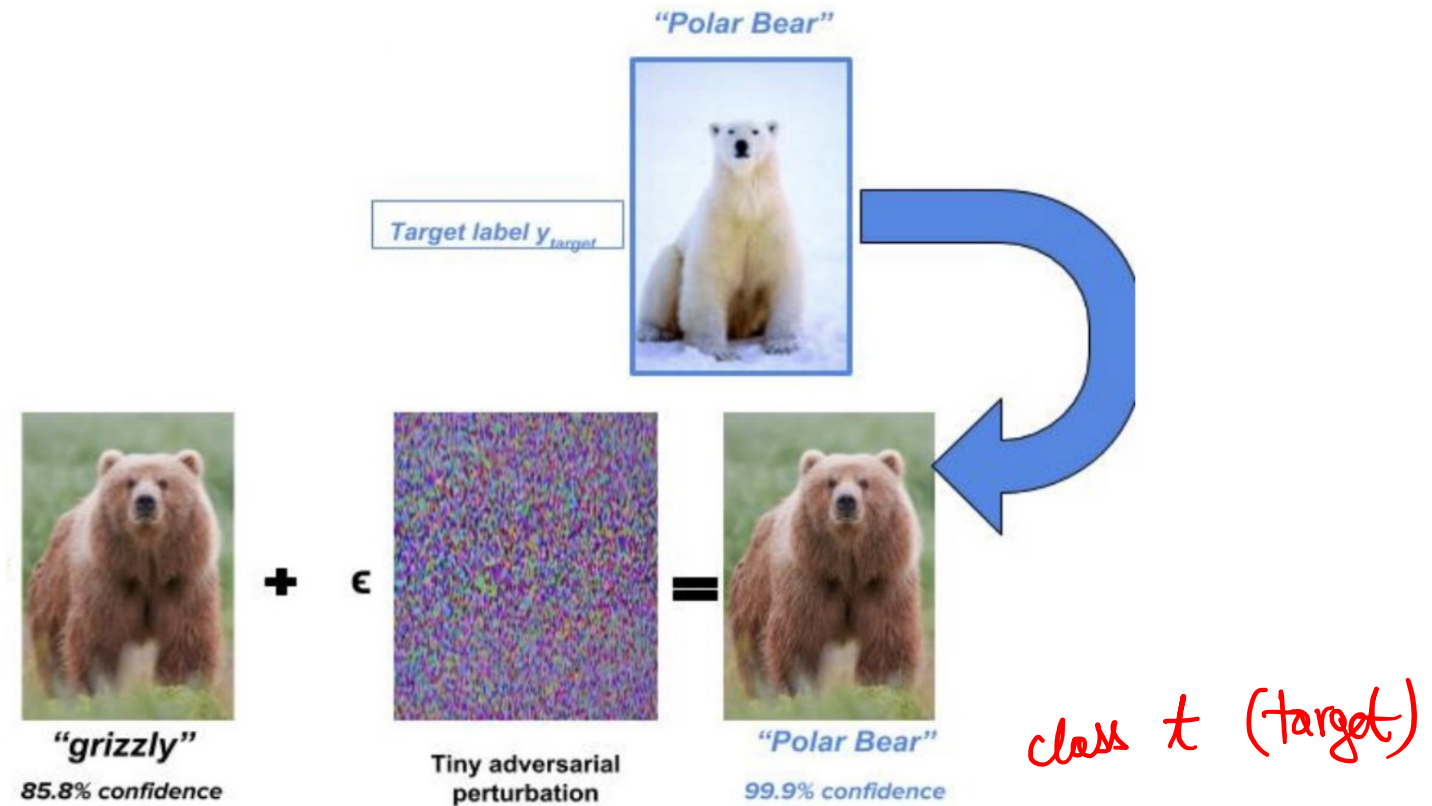
99.3 % confidence

[Goodfellow et al. 2015](#)

any class

- Misclassification could be to any class

Targeted Adversarial Examples



- Misclassification to a targeted class chosen by the attacker

Untargeted vs Targeted Attacks

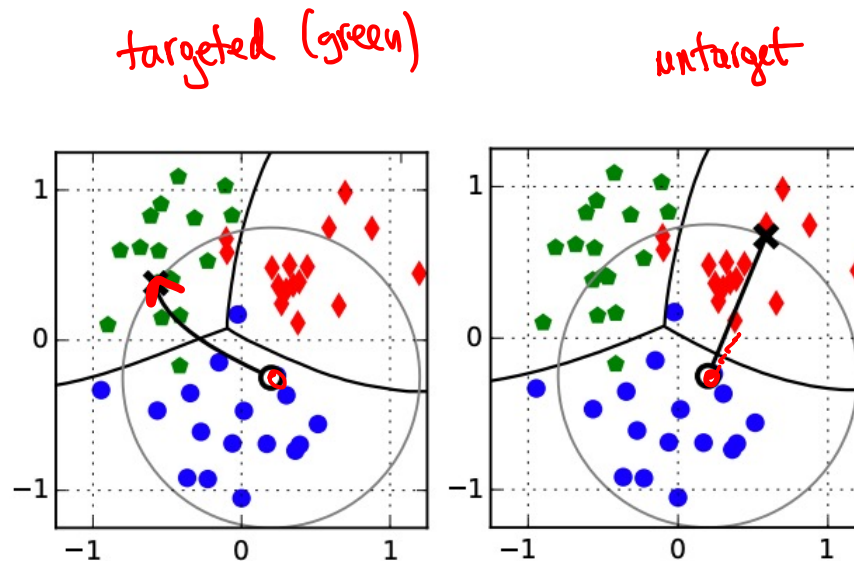
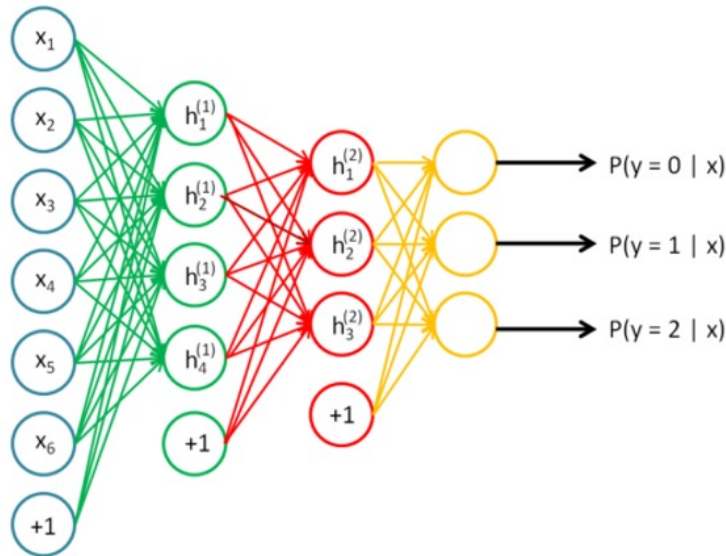


Figure 6: Examples of error-specific (*left*) and error-generic (*right*) evasion, as reported in [7]. Decision boundaries among the three classes (blue, red and green points) are shown as black lines. In the error-specific case, the initial (blue) sample is shifted towards the green class (selected 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)
 - One step attack
 - Goodfellow et al. Explaining and Harnessing Adversarial Examples, 2015
- Iterative 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 Attacks

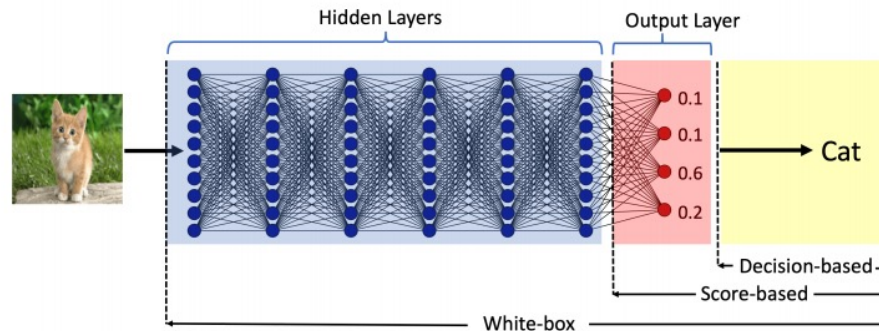


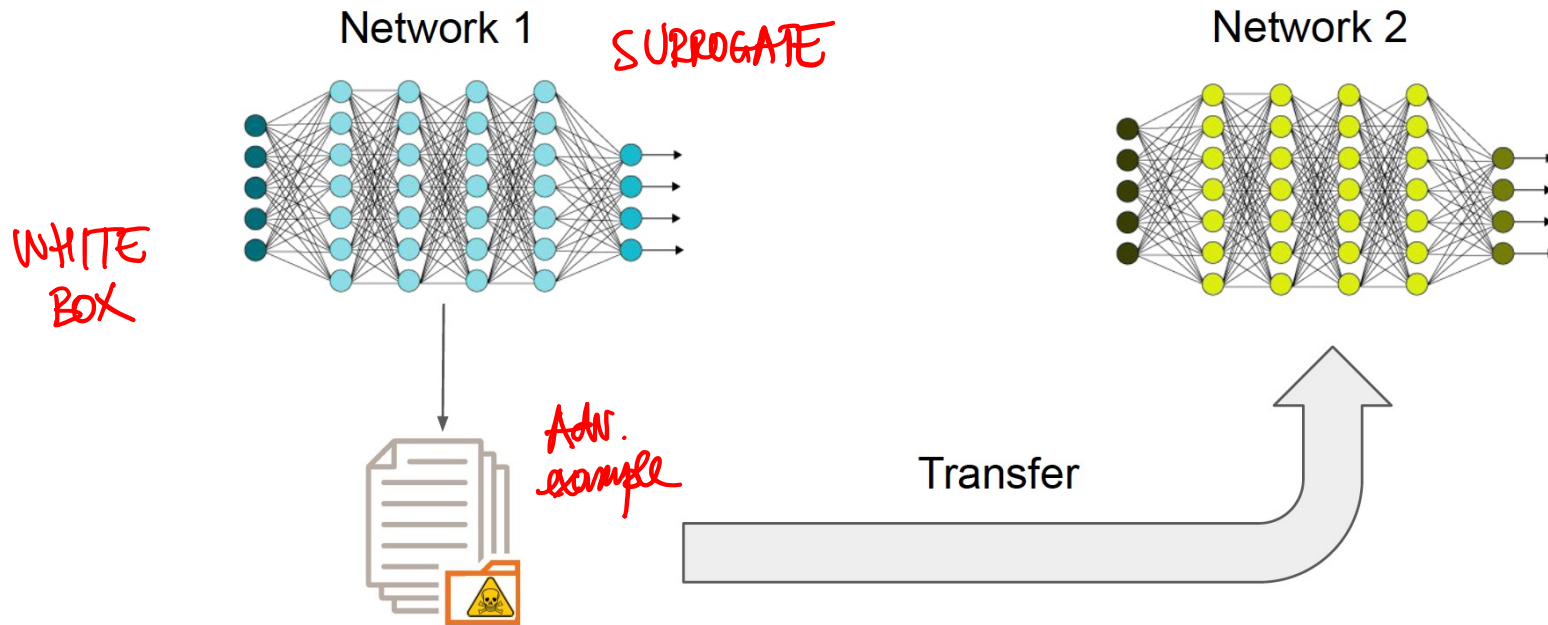
Figure 1: An illustration of accessible components of the target model for each of the three threat models. A white-box threat model assumes access to the whole model; a score-based threat model assumes access to the output layer; a decision-based threat model assumes access to the predicted label alone.

- **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

$[x_1, x_2, \dots, x_d]$

$[x_1 + \delta_1, x_1 + \delta_2, \dots, x_d + \delta_d]$

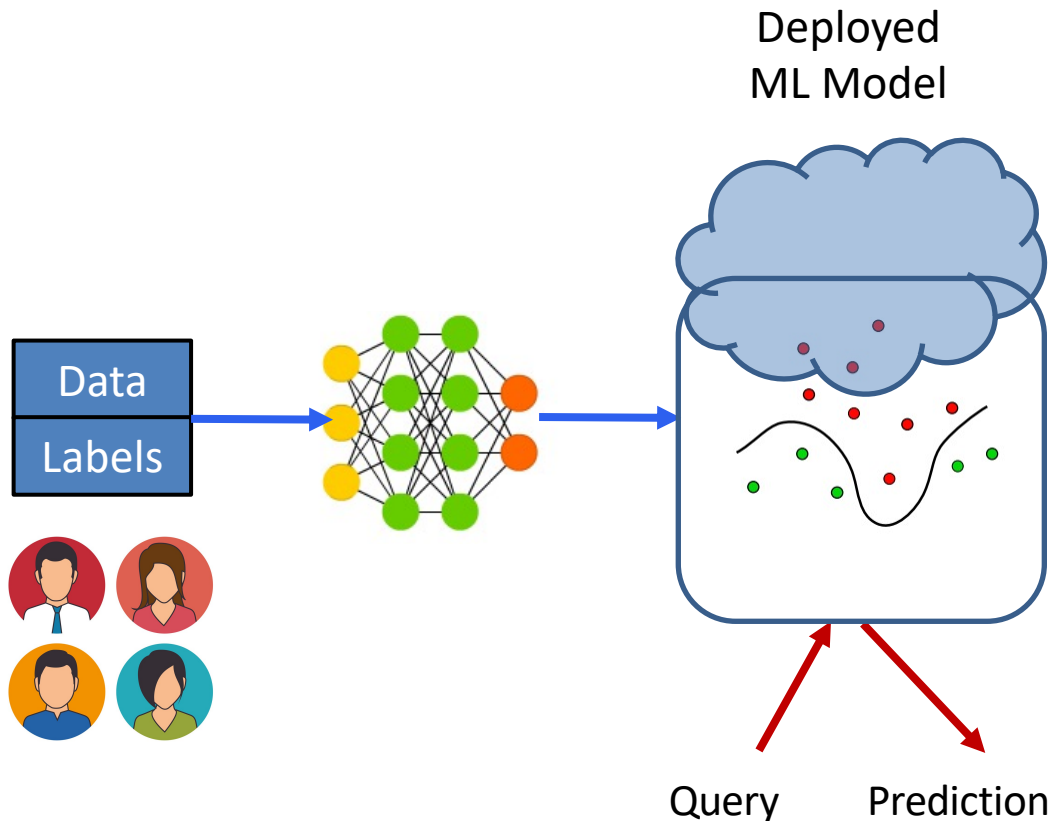


- Create own surrogate model, generate adversarial samples (poisoning or evasion) and transfer them to the target model
- Papernot et al. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples, 2016

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
- 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

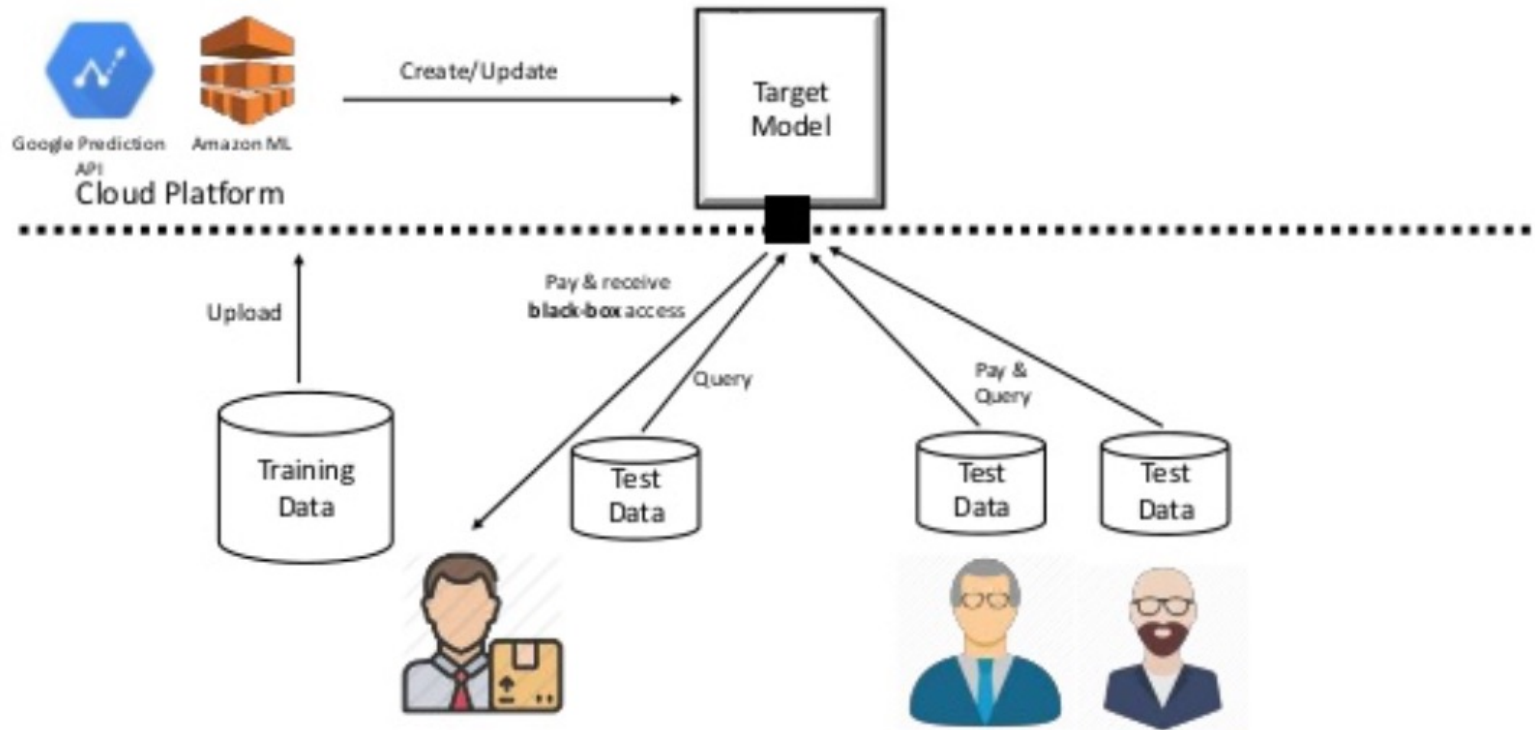
Privacy Attacks on ML



- **Reconstruction attacks:** Extract sensitive attributes
 - [Dinur and Nissim 2003]
- **Membership Inference:** Determine if sample was in training
 - [Shokri et al. 2017], [Yeom et al. 2018], [Hayes et al. 2019]
- **Model Extraction:** Learn model architecture and parameters
 - [Tramer et al. 2016], [Jagielski et al. 2020]
- **Memorization:** Extract training data from queries to the model
 - [Carlini et al. 2021]

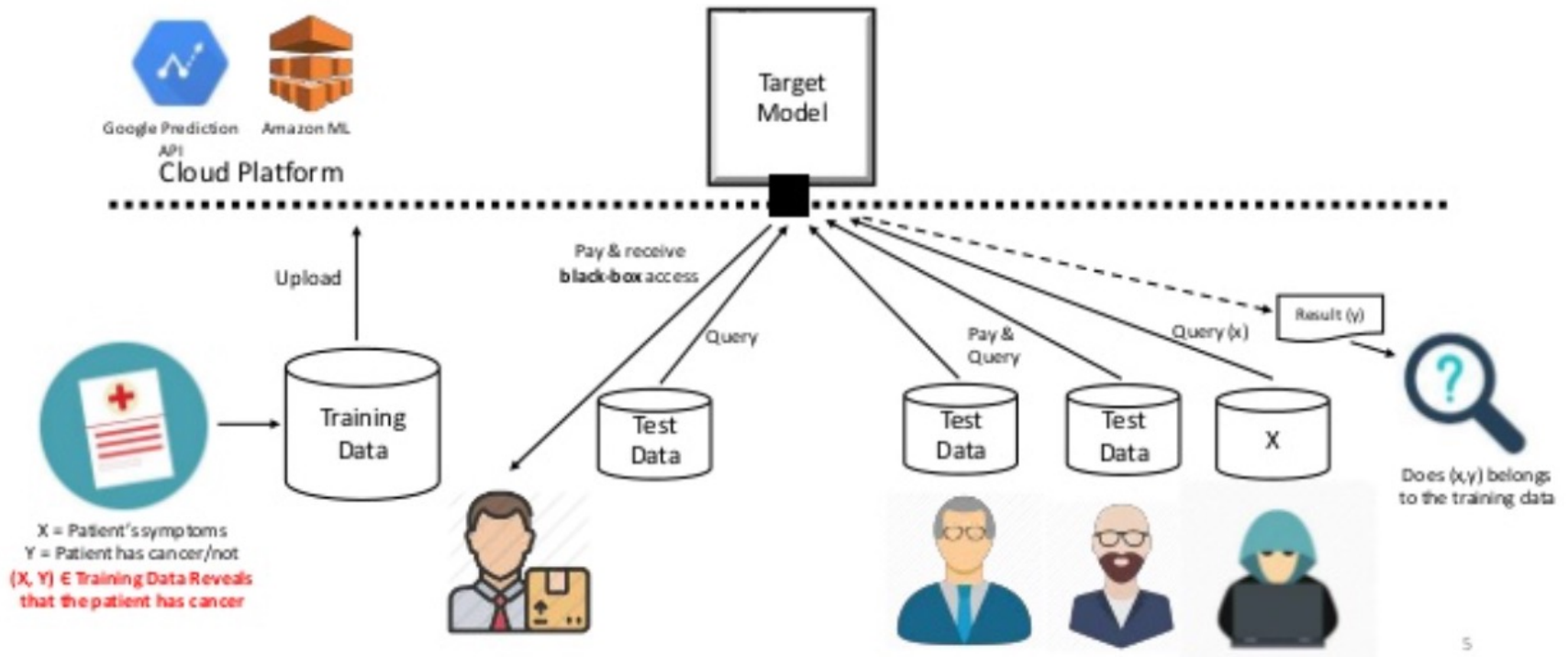
Privacy Attacks against ML

Machine Learning as a Service



Black-box query access to model

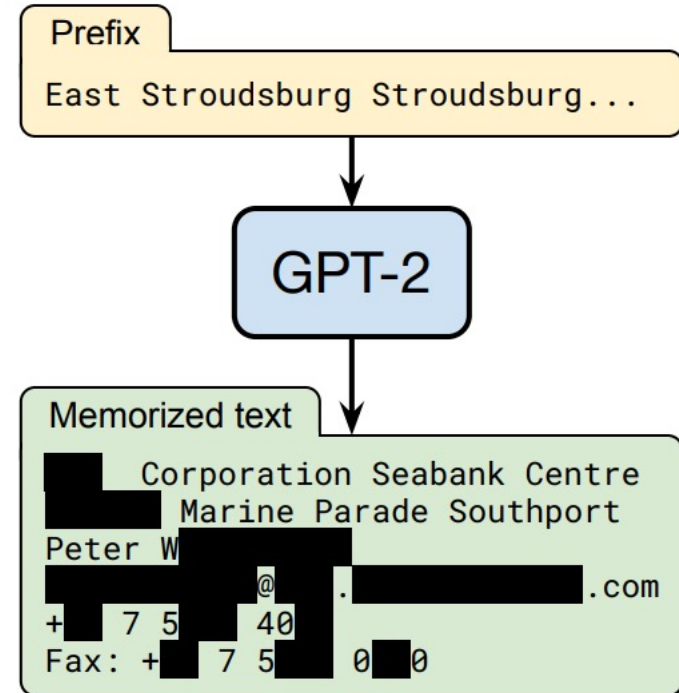
Membership Inference Attack



- Given a data sample, was it used to train a model?
- Shokri et al. Membership Inference Attacks Against Machine Learning Models, 2017; Shadow training of multiple ML models
- Yeom et al. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting, 2018; Explore loss difference on training points vs other data

Memorization in Language Models

- GPT-2: generative language model
- Prompt GPT-2 with different prefixes
- Rank by likelihood of sample: use perplexity measure (low perplexity have high likelihood)
- Use Membership Inference to predict if sample was part of training



N. Carlini et al. Extracting Training Data from Large Language Models. <https://arxiv.org/pdf/2012.07805.pdf>

Model Extraction

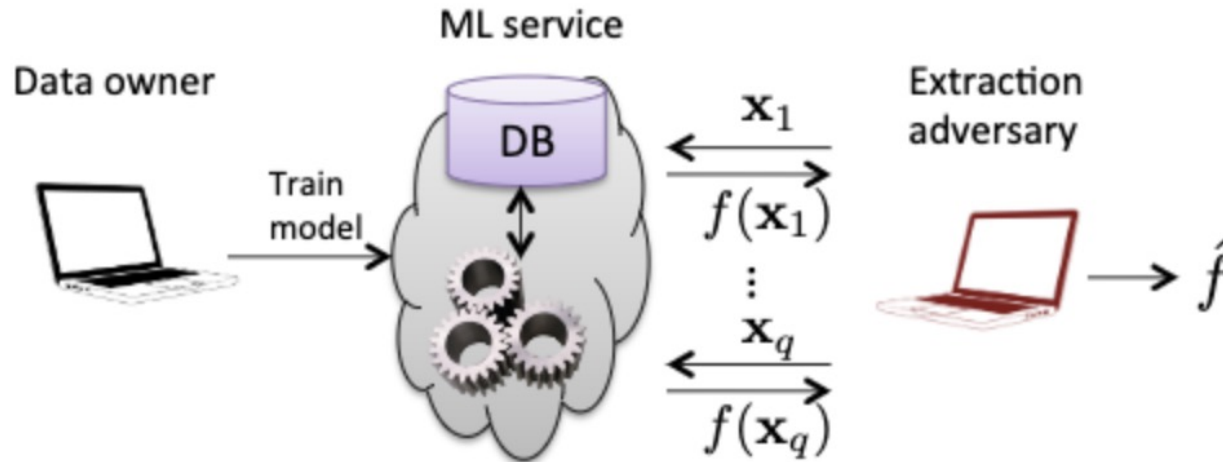


Figure 1: Diagram of ML model extraction attacks. A data owner has a model f trained on its data and allows others to make prediction queries. An adversary uses q prediction queries to extract an $\hat{f} \approx f$.

- Tramer et al. Stealing Machine Learning Models via Prediction APIs, 2017
- Jagielski et al. High Accuracy and High Fidelity Extraction of Neural Networks, 2020

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://aif360.mybluemix.net/>
 - 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
- De Cristofaro. A Critical Overview of Privacy in Machine Learning, 2021
- 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

CY 7790

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
- Scribes
 - Take notes during discussion and write them using template we will provide
- Everyone else
 - Read the papers, submit summaries and bring discussion points to class