

DS 4400

Machine Learning and Data Mining I

Alina Oprea

Associate Professor, Khoury College
Northeastern University

April 15 2021

Final Project Report

- Presentation – 20 points
 - 5 minutes talk + 3 minutes questions
 - Schedule on Piazza under Resources
- Exploratory data analysis – 20 points
 - Info about the dataset, features, and labels
 - Discuss feature representation and selection
 - Include graphs on selective feature distributions
- Machine learning models – 30 points
 - Use at least 4 models
 - Use correct methodology (e.g., cross-validation)
- Metrics – 10 points
 - Report several metrics to evaluate and compare models
 - Generate ROC curves; include confusion matrix
- Interpretation of results – 15 points
 - Why the models make errors; which features are most relevant; why is it a challenging task (e.g., imbalanced?)
- References – 5 points
 - List related literature you consulted for the project

What We Covered

Ensembles

- Bagging
- Random forests
- Boosting
- AdaBoost

Deep learning

- Feed-forward Neural Nets
- Convolutional Neural Nets
- Architectures
- Forward and back propagation
- Transfer learning

Linear classification

- Perceptron
- Logistic regression
- LDA
- Linear SVM

Non-linear classification

- kNN
- Decision trees
- Naïve Bayes
- Kernel SVM

- Metrics
- Evaluation
- Cross-validation
- Regularization
- Gradient Descent

Linear Regression

Linear algebra

Probability and statistics

Other Timely Topics in ML

- Machine Learning Interpretability
 - How to interpret and explain results generated by ML
- Fairness in Machine Learning
- Privacy in Machine Learning
 - How to use Differential Privacy to train models
 - Tradeoff between privacy and utility
- Federated learning
 - Training ML in a distributed fashion to protect user data
- Application-specific ML models: NLP generative models (GPT-2, GPT-3, BERT)
- Unsupervised learning: embeddings, autoencoders, clustering, anomaly detection
- Reinforcement Learning
- Adversarial Machine Learning

Adversarial ML

- Attacks

- Studies how can Machine Learning Fail
- Different attack models
 - Attack objective and knowledge about the ML system

- Defenses

- How to defend Machine Learning against different failures and improve their robustness
- What are the tradeoffs between accuracy and robustness

Adversarial Machine Learning: Taxonomy

Attacker's Objective

Learning stage

	Targeted Target small set of points	Availability Target majority of points	Privacy Learn sensitive information
Training	Targeted Poisoning Backdoor Trojan Attacks	Poisoning Availability Model Poisoning	-
Testing	Evasion Attacks Adversarial Examples	-	Reconstruction Membership Inference Model Extraction


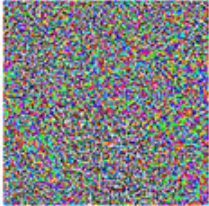

Adversarial Machine Learning: Taxonomy

Attacker's Objective

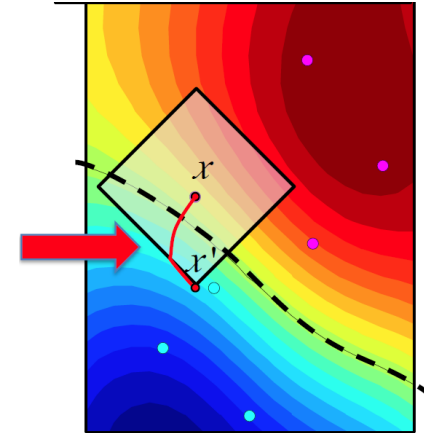
Learning stage

	Targeted Target small set of points	Availability Target majority of points	Privacy Learn sensitive information
Training	Targeted Poisoning Backdoor Trojan Attacks	Poisoning Availability Model Poisoning	-
Testing	Evasion Attacks Adversarial Examples	-	Reconstruction Membership Inference Model Extraction

Evasion Attacks

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

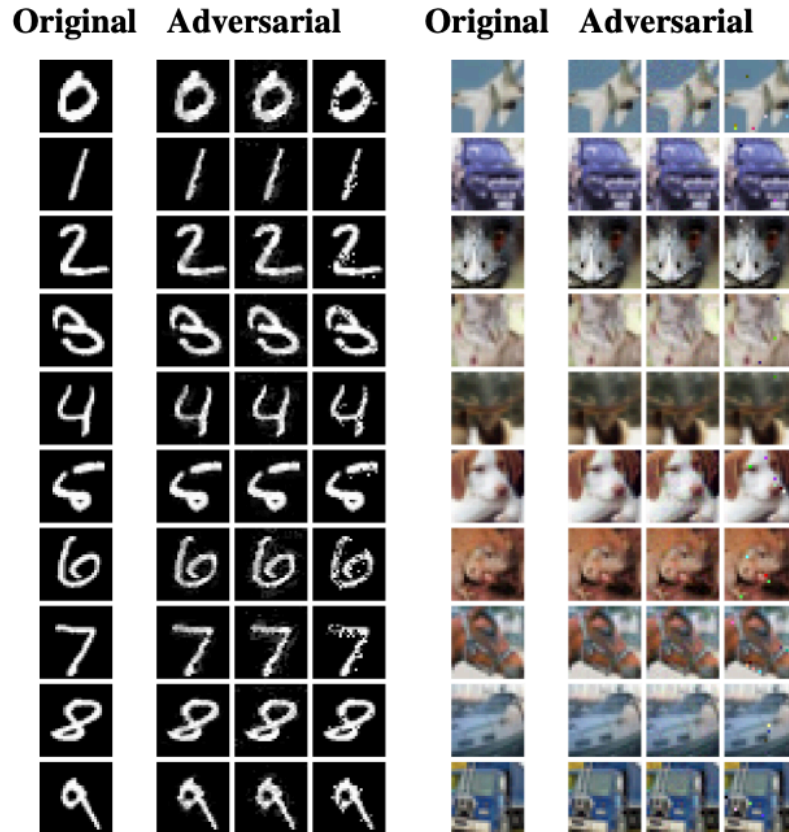
Adversarial
example



- **Evasion attack:** attack against ML at testing time
- **Implications**
 - Small (imperceptible) modification at testing time can change the classification of any data point to any targeted class

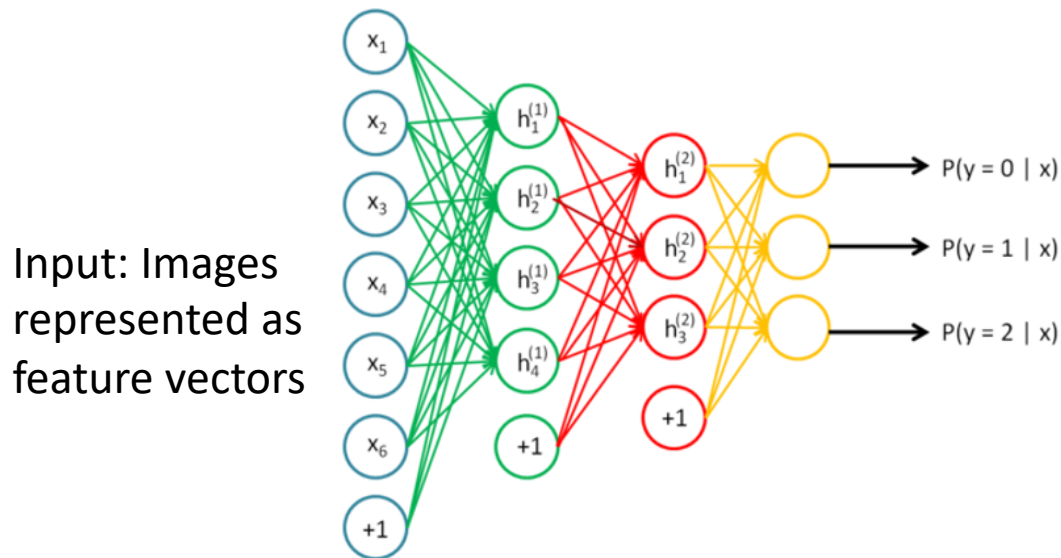
- Szegedy et al. *Intriguing properties of neural networks*. 2014
<https://arxiv.org/abs/1312.6199>
- Goodfellow et al. *Explaining and Harnessing Adversarial Examples*. 2014.
<https://arxiv.org/abs/1412.6572>

Adversarial Examples



- N. Carlini and D. Wagner. *Towards Evaluating the Robustness of Neural Networks*. In IEEE Security and Privacy Symposium 2017
<https://arxiv.org/abs/1608.04644>
- Goal: create minimum perturbations for adversarial examples
- They always exist!
- Application domains: image recognition, videos classification, text models, speech recognition

Evasion Attacks For Neural Networks



Optimization Formulation

Given input x
Find adversarial example

$$x' = x + \delta$$

$$\min_{\delta} c \|\delta\|_2^2 + L_t(x + \delta)$$

Min distance

Change class

[Carlini, Wagner 17]

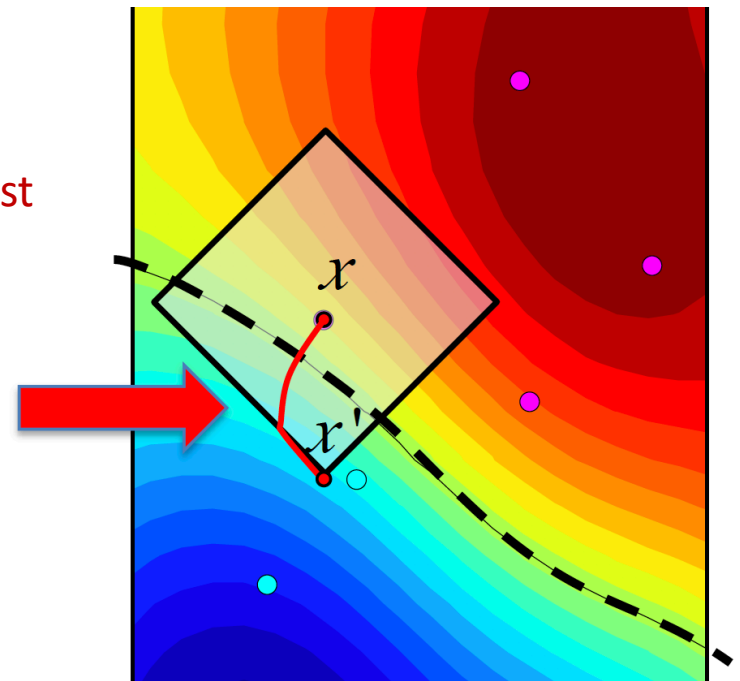
- Most existing attacks are in continuous domains
- Images represented as matrix of pixels with continuous values
- How to solve optimization problem?

Projected Gradient Descent (PGD)

- **Goal:** maximum-confidence *evasion*
- **Knowledge:** *perfect (white-box attack)*
- **Attack strategy:**

$$\begin{aligned} \min_{x'} \quad & L_t(x') \quad \text{Loss on target class } t \\ \text{s. t. } \quad & \|x - x'\|_p \leq d_{\max} \quad \text{Upper bound on dist} \end{aligned}$$

- Non-linear, constrained optimization
 - **Projected gradient descent:** approximate solution for *smooth* functions
- Gradients of $g(x)$ can be analytically computed in many cases
 - SVMs, Neural networks



- In each iteration of gradient descent, perform a projection to feasible space
- Madry et al. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2018. <https://arxiv.org/pdf/1706.06083.pdf>

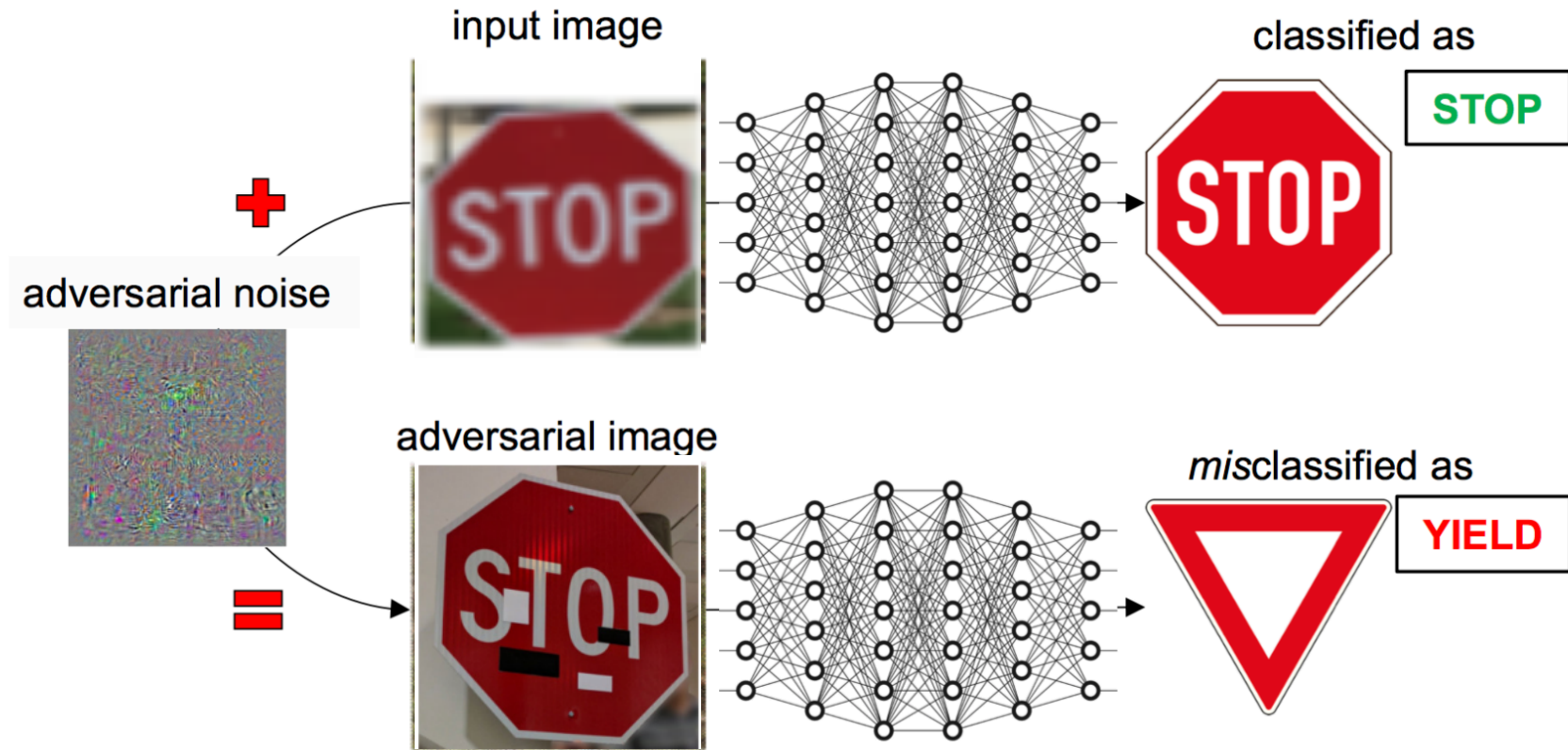
Feasible Adversarial Examples

Adversarial Glasses

- M. Sharif et al. (ACM CCS 2016) attacked deep neural networks for face recognition with carefully-fabricated eyeglass frames
- When worn by a 41-year-old white male (left image), the glasses mislead the deep network into believing that the face belongs to the famous actress Milla Jovovich



Adversarial Attacks on Road Signs



Eykholt et al. *Robust Physical-World Attacks on Deep Learning Visual Classification*. In CVPR 2018

Speech Recognition

Audio Adversarial Examples

Audio

Transcription by Mozilla DeepSpeech



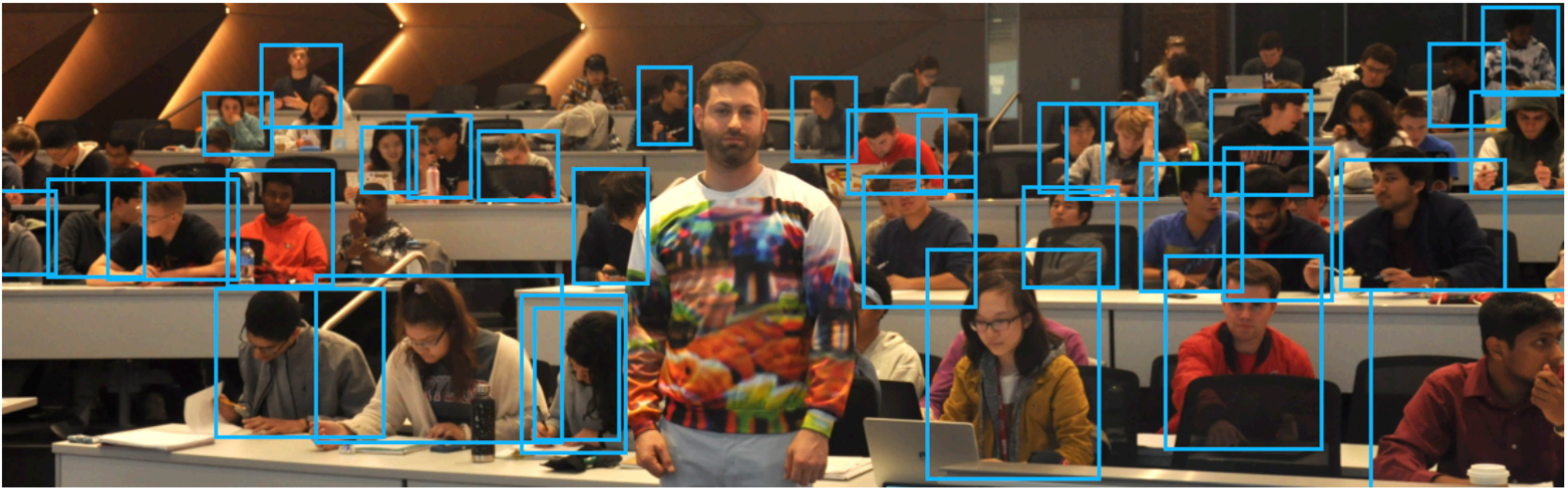
“without the dataset the article is useless”



“okay google browse to evil dot com”

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

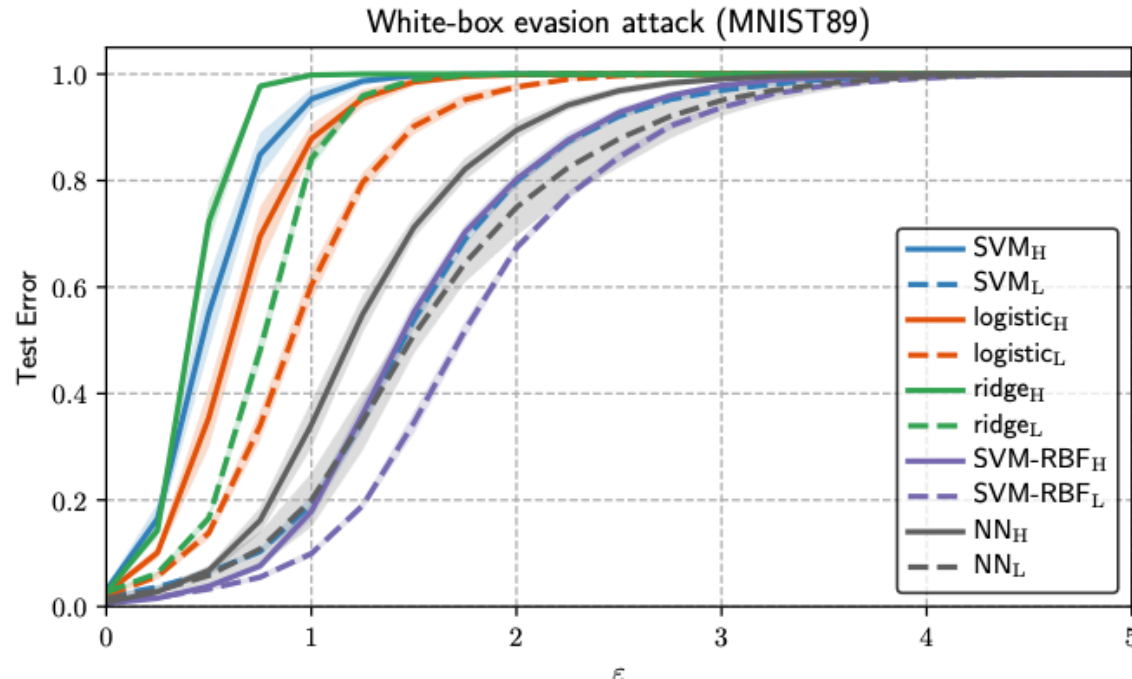
Attacking Object Detectors



This stylish pullover is a great way to stay warm this winter, whether in the office or on-the-go. It features a stay-dry microfleece lining, a modern fit, and adversarial patterns the evade most common object detectors. In this demonstration, the YOLOv2 detector is evaded using a pattern trained on the COCO dataset with a carefully constructed objective.

<https://www.cs.umd.edu/~tomg/projects/invisible/>

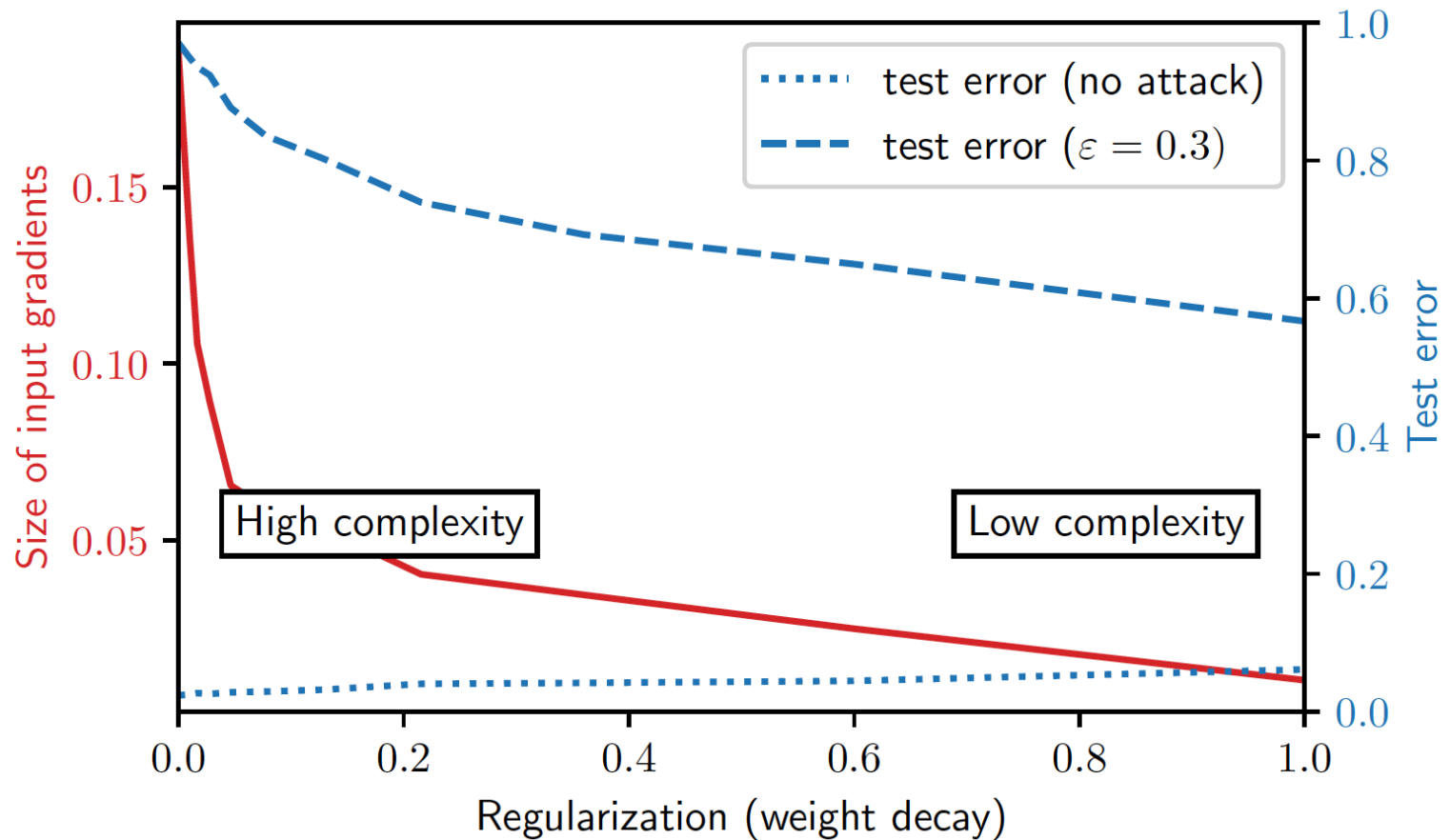
Multiple Classifiers Fail under Evasion



- Classifier test error as a function of perturbation budget on MNIST dataset
- Linear classifiers: SVM, logistic regression, ridge
- Non-linear classifiers: SVM-RBF, Feed-forward neural network

A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, F. Roli. *Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks*. USENIX Security, 2019

Impact of Regularization



Evasion Attacks in Connected Cars

- Udacity challenge: Predict steering angle from camera images, 2014
- Actions
 - Turn left (negative steering angle)
 - Turn right (positive steering angle)
 - Straight (steering angle in $[-T, T]$)
- Dataset has 33,608 images and steering angle values (70GB of data)



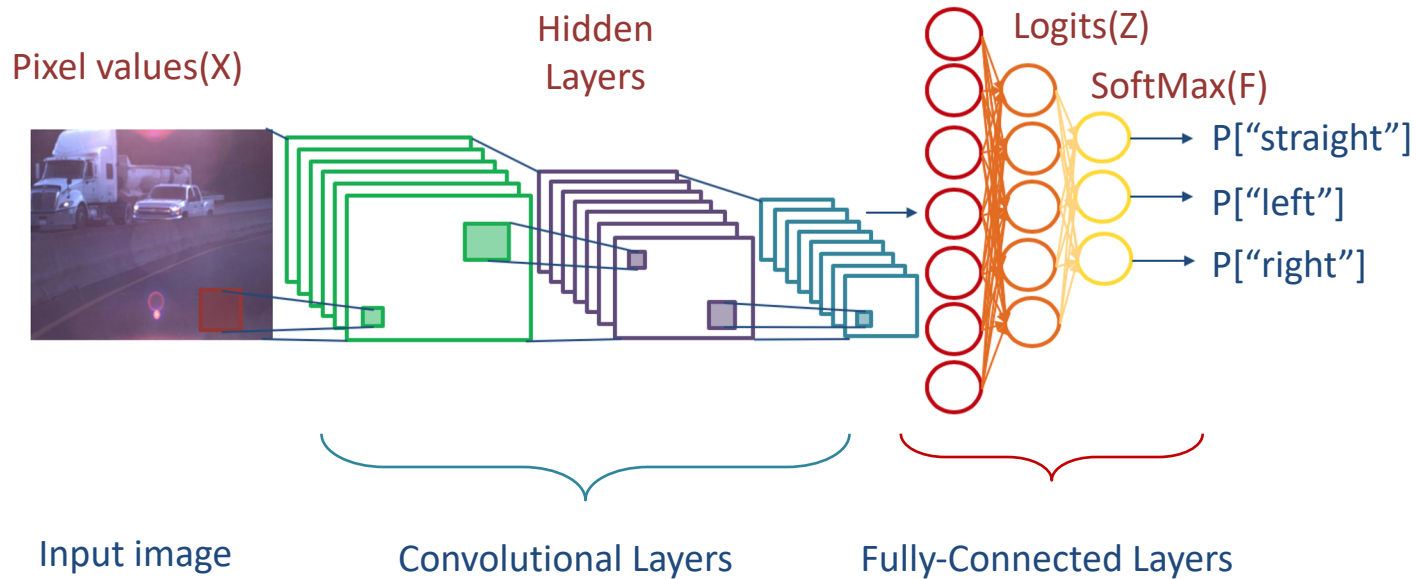
Predict direction: Straight, Left, Right
Predict steering angle

A. Chernikova, A. Oprea, C. Nita-Rotaru, and B. Kim.

Are Self-Driving Cars Secure? Evasion Attacks against Deep Neural Networks for Self-Driving Cars.

In IEEE SafeThings 2019. <https://arxiv.org/abs/1904.07370>

CNN for Direction Prediction

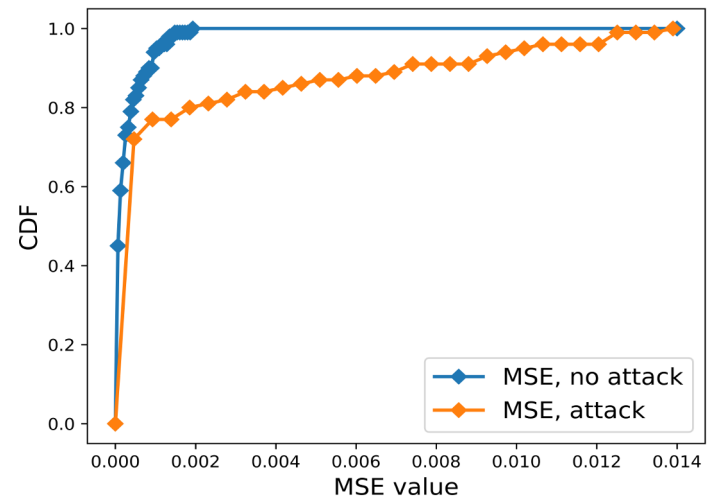


- Two CNN architectures: 25 million and 467 million parameters
- For Regression, exclude the last softmax layer
- Architectures used in the Udacity challenge

Evasion Attack against Regression

- First evasion attack for CNNs for regression
- New objective function
 - Minimize adversarial perturbation
 - Maximize the square residuals (difference between the predicted and true response)

$$\begin{aligned} \min_{\delta} c \|\delta\|_2^2 - R(x + \delta, y) \\ \text{such that } x + \delta \in [0, 1]^d \\ R(x + \delta, y) = [F(x + \delta) - y]^2 \end{aligned}$$



- 10% of adversarial images have 20 times higher MSE
- The maximum ratio of adversarial to legitimate MSE reaches 69

Adversarial Example for Regression



Original Image

Steering angle = -4.25; MSE = 0.0016



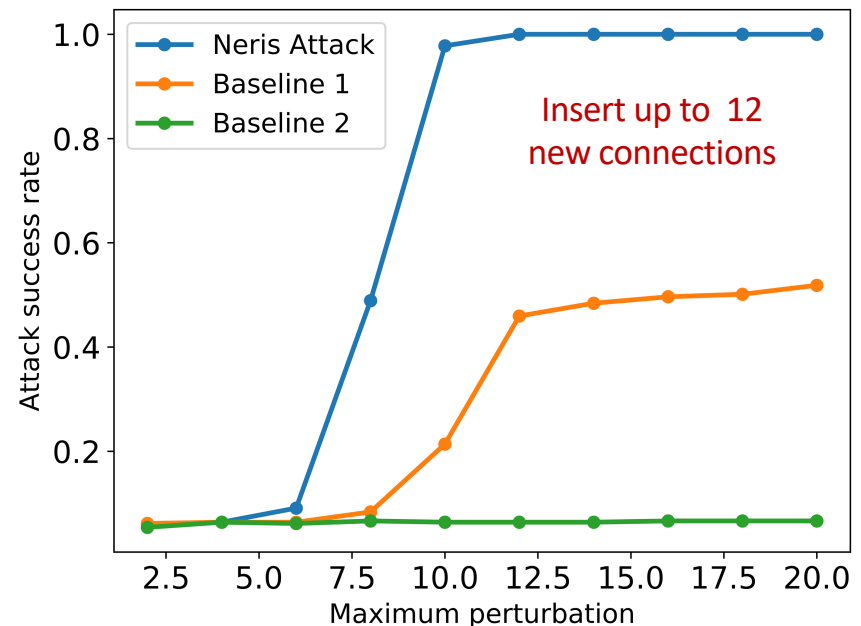
Adversarial Image

Steering angle = -2.25; MSE = 0.05

- Significant degradation of CNN classifiers in connected cars
- Small amount of perturbation is effective
- Models for both classification and regression are vulnerable

How Effective are Evasion Attacks in Security?

- **Dataset:** CTU-13, Neris botnet, highly imbalanced
 - 194K benign
 - 3869 malicious
- **Features:** 756 on 17 ports
- **Model:** Feed-forward neural network (3 layers), F1: 0.96

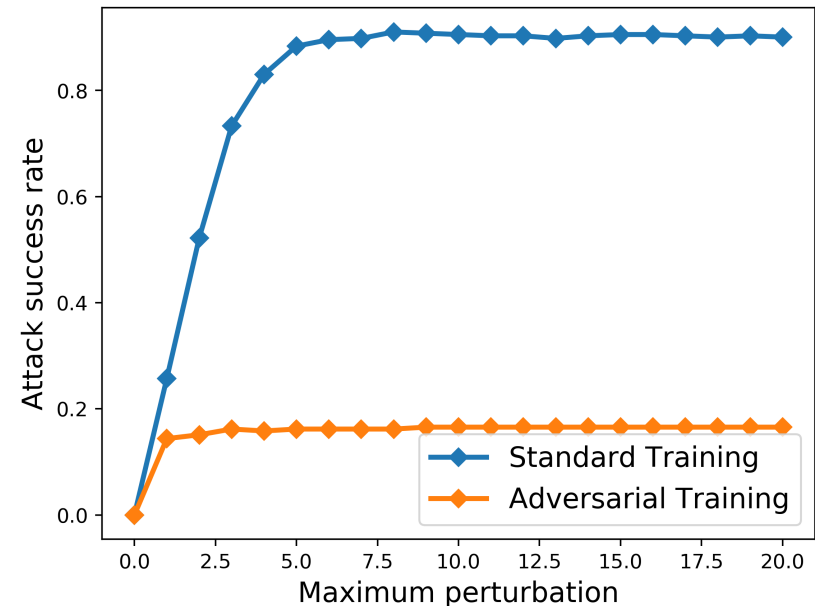


A. Chernikova and A. Oprea. *FENCE: Feasible Evasion Attacks on Neural Networks in Constrained Environments*

<http://arxiv.org/abs/1909.10480>, 2019.

Defense: Adversarial Training

- Adversarial Training
 - Train model iteratively
 - In each iteration, generate adversarial examples and add to training with correct label
- Implications
 - Adversarial training can improve ML robustness
- Challenges
 - Computationally expensive
 - Specific to certain attacks
 - Does it generalize to other attacks?



Malicious domain classifier

Taxonomy

Attacker's Objective

Learning stage

	Targeted Target small set of points	Availability Target majority of points	Privacy Learn sensitive information
Training	Targeted Poisoning Backdoor Trojan Attacks	Poisoning Availability	Membership Inference
Testing	Evasion Attacks Adversarial Examples	-	Reconstruction Membership Inference Model Extraction

Training-Time Attacks

- ML is trained by crowdsourcing data in many applications

- Social networks
- News articles
- Tweets



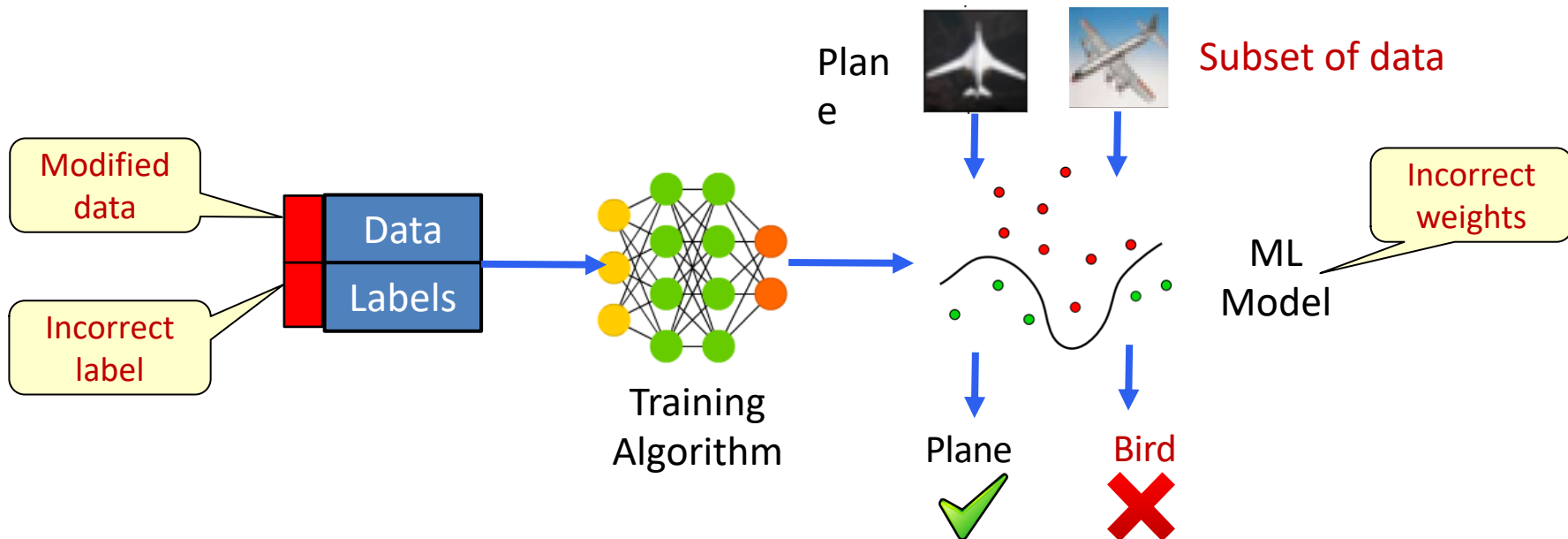
- Navigation systems
- Face recognition
- Mobile sensors

- Cannot fully trust training data!

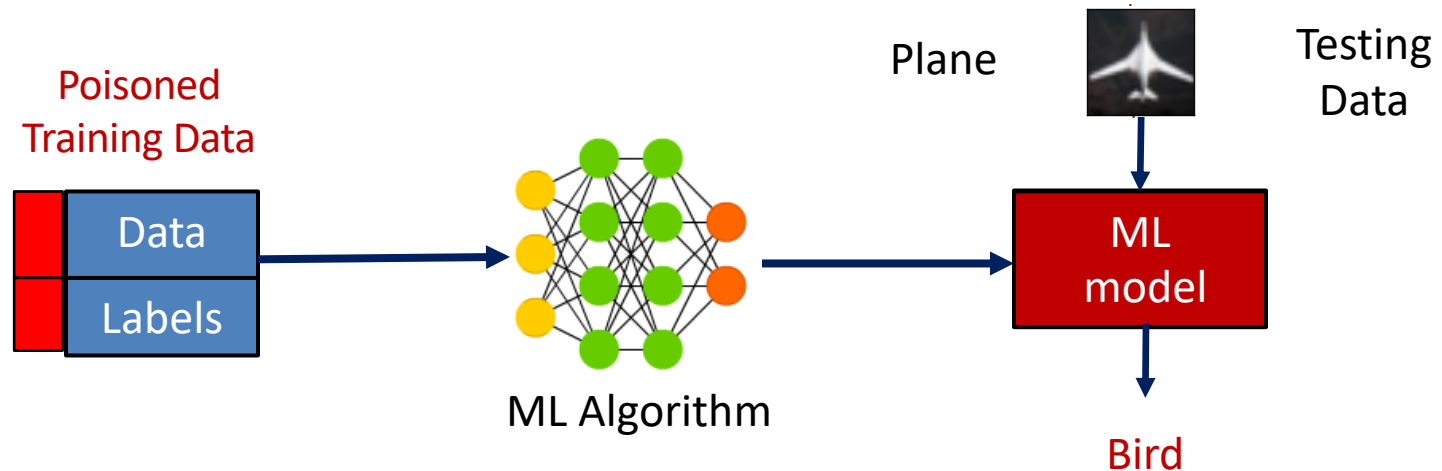


Poisoning Attacks

- Microsoft Industry Survey: Poisoning is top concern
 - Kumar et al. *Adversarial Machine Learning – Industry Perspective*. 2020
- Supply Chain vulnerabilities started to gain attention (SolarWinds attack)



Poisoning Availability Attacks



- **Attacker Objective:**
 - Corrupt the predictions by the ML model significantly
- **Attacker Capability:**
 - Insert fraction of poisoning points in training
 - Find the points that cause the maximum impact

M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li.
Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In IEEE S&P 2018

Optimization Formulation

Given a training set D find a set of poisoning data points D_p that maximizes the adversary objective A on validation set D_{val} where corrupted model θ_p is learned by minimizing the loss L on $D \cup D_p$

$$\begin{aligned} & \operatorname{argmax}_{D_p} A(D_{val}, \theta_p) \text{ s.t.} \\ & \theta_p \in \operatorname{argmin}_{\theta} L(D \cup D_p, \theta) \end{aligned}$$

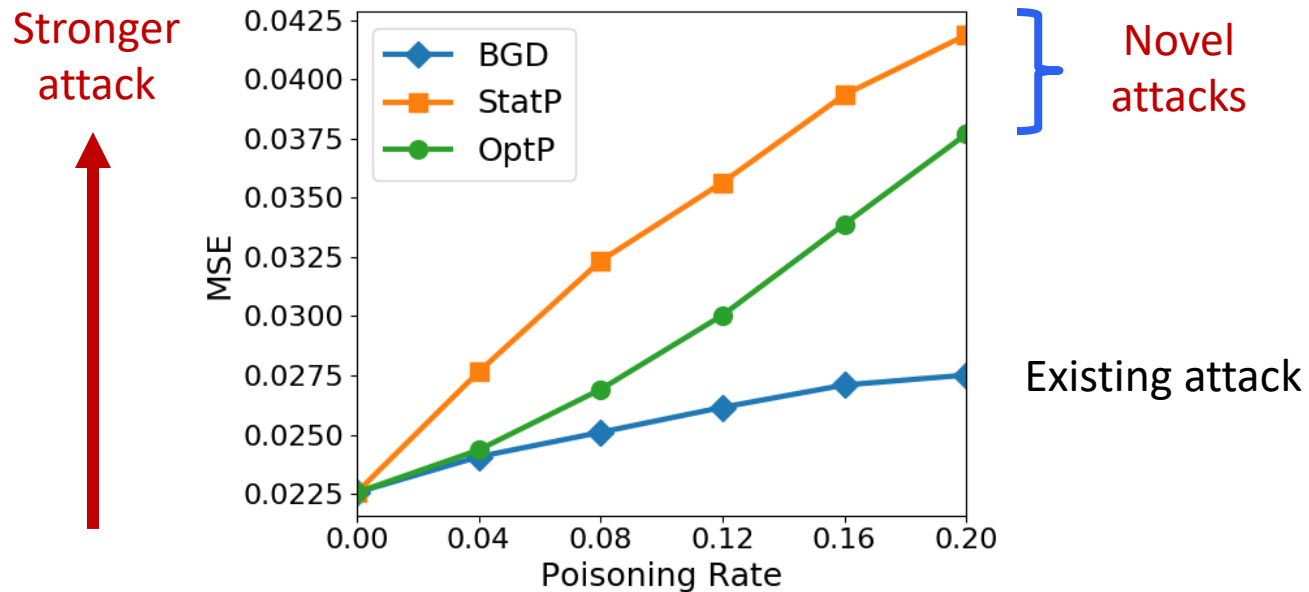
Bilevel Optimization
NP-Hard!

First white-box attack for linear regression [Jagielski et al. 18]

- Determine optimal poisoning point (x_c, y_c)
- Optimize by both x_c and y_c

Poisoning Regression

- Improve existing attacks **by a factor of at most 6.83**



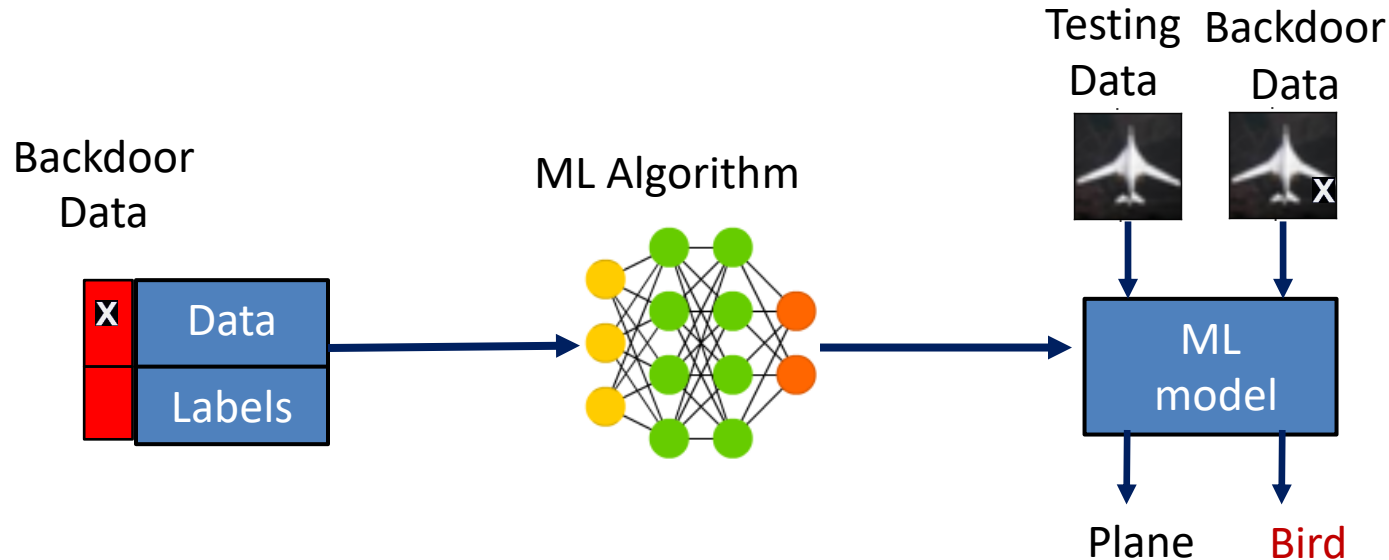
Predict loan rate with ridge regression
(L2 regularization)

Is It Really a Threat?

- Case study on healthcare dataset (predict Warfarin medicine dosage)
- At 20% poisoning rate
 - Modifies 75% of patients' dosages by 93.49% for LASSO
 - Modifies 10% of patients' dosages by a factor of 4.59 for Ridge
- At 8% poisoning rate
 - Modifies 50% of the patients' dosages by 75.06%

Quantile	Initial Dosage	Ridge Difference	LASSO Difference
0.1	15.5 mg/wk	31.54%	37.20%
0.25	21 mg/wk	87.50%	93.49%
0.5	30 mg/wk	150.99%	139.31%
0.75	41.53 mg/wk	274.18%	224.08%
0.9	52.5 mg/wk	459.63%	358.89%

Backdoor Poisoning Attacks



- **Attacker Objective:**
 - Prediction on clean data is unchanged
 - Change prediction of *backdoor data* in testing
- **Attacker Capability:**
 - Add backdoored poisoning points in training
 - Add backdoor pattern in testing
- [Gu et al. 17], [Chen et al. 17], [Turner et al. 18], [Shafahi et al. 18]

BadNets



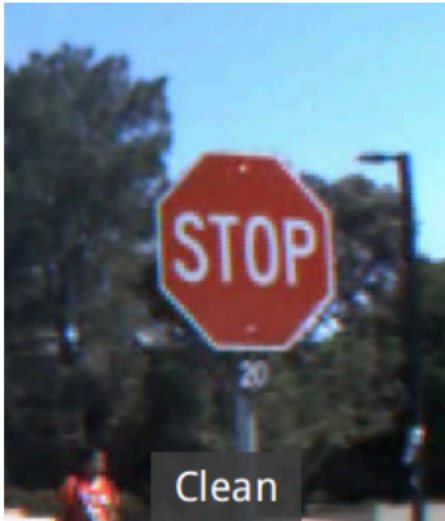
Original image



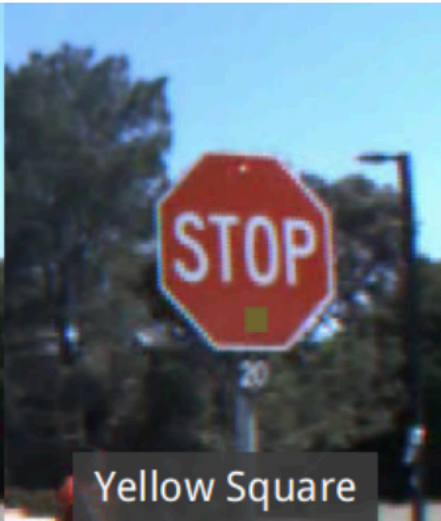
Single-Pixel Backdoor



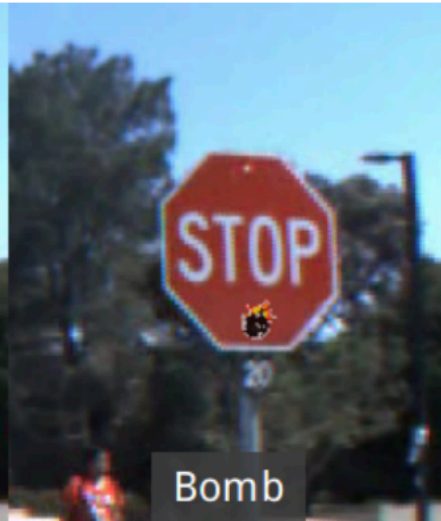
Pattern Backdoor



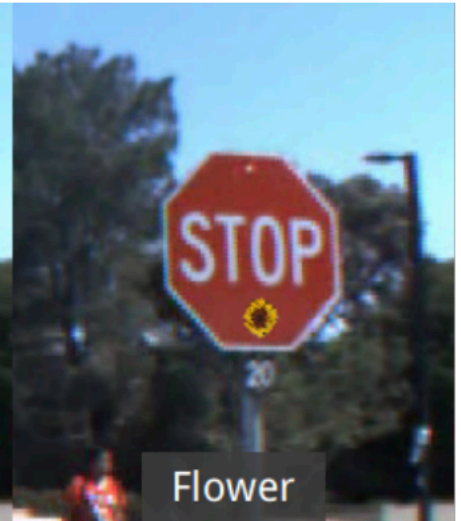
Clean



Yellow Square



Bomb



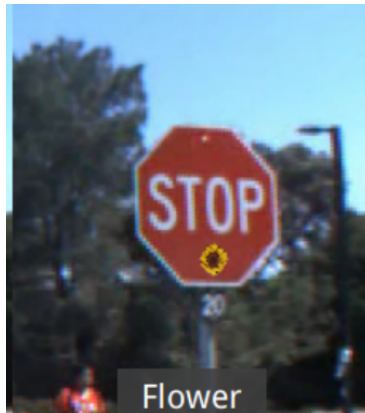
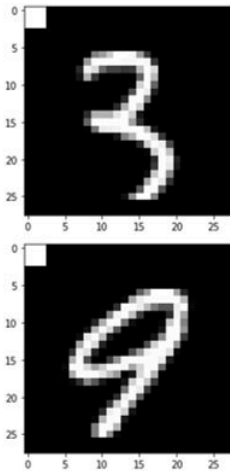
Flower

Gu et al. *BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain*. 2017. <https://arxiv.org/abs/1708.06733>

Backdoors in Feature-Based Models

Computer vision

- A fixed **pixel pattern**.



Feature space

- Fixed **assignment** of numerical **values to features**.

Feature	LightGBM	EmberNN
major_image_version	1704	14
major_linker_version	15	13
major_operating_system_version	38078	8
minor_image_version	1506	12
minor_linker_version	15	6
minor_operating_system_version	5	4
minor_subsystem_version	5	20

- Identify most relevant features that point to target class
- Equivalent to variable importance, but model-agnostic
- Use techniques from ML explainability to identify relevant features

G. Severi, J. Meyer, S. Coull, A. Oprea. *Explanation-Guided Backdoor Poisoning Attacks Against Malware Classifiers*. USENIX Security 2021.

<https://arxiv.org/abs/2003.01031>

ML Interpretability

Goals

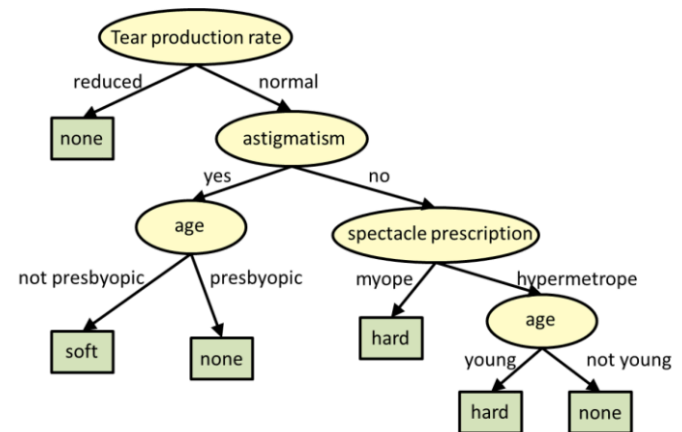
- Explain why models makes a prediction
- Which features and values contribute to the prediction
- Which features are most important
- In pre-deep learning models, some models are considered “interpretable”

Diagram illustrating the components of a linear regression model equation:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

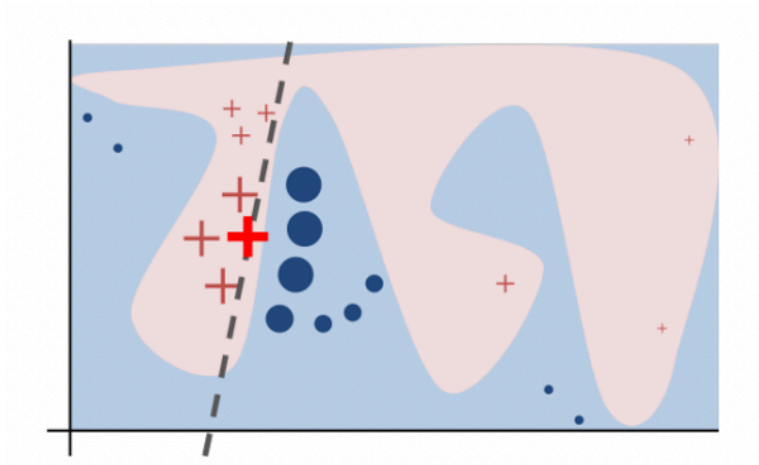
Labels and components:

- Dependent Variable:** Y_i
- Population Y intercept:** β_0
- Population Slope Coefficient:** β_1
- Independent Variable:** X_i
- Random Error term:** ϵ_i
- Linear component:** $\beta_0 + \beta_1 X_i$
- Random Error component:** ϵ_i



Interpretability for Neural Networks

- Hard to explain a complex model in its entirety
 - How about explaining smaller regions?



LIME (Ribeiro et. al.)

- Explains decisions of any model in a local region around a particular point
- Learns sparse linear model

Example LIME

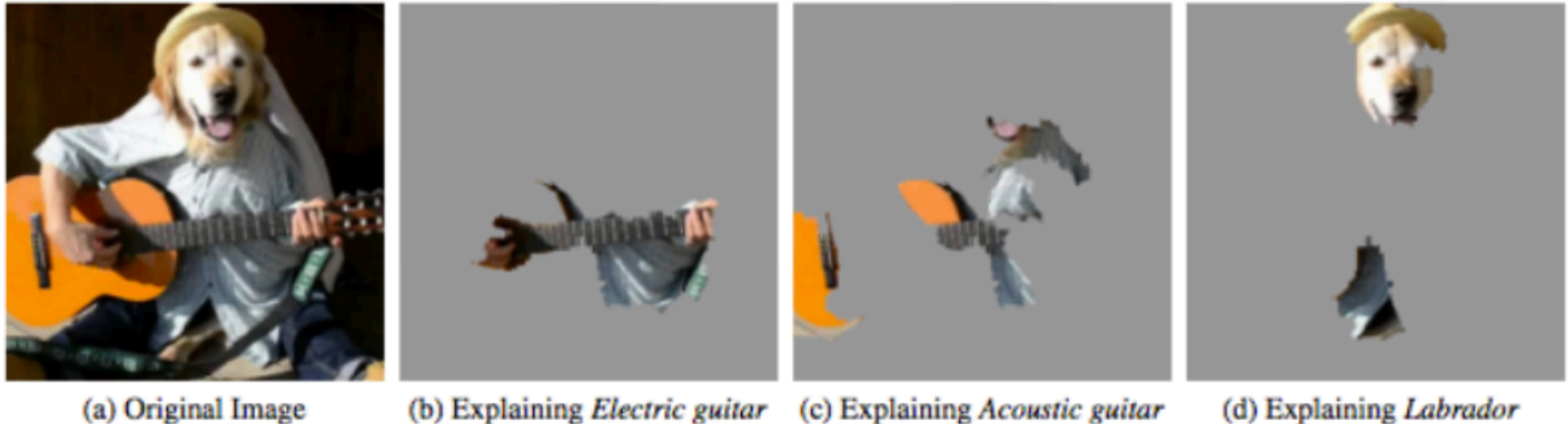
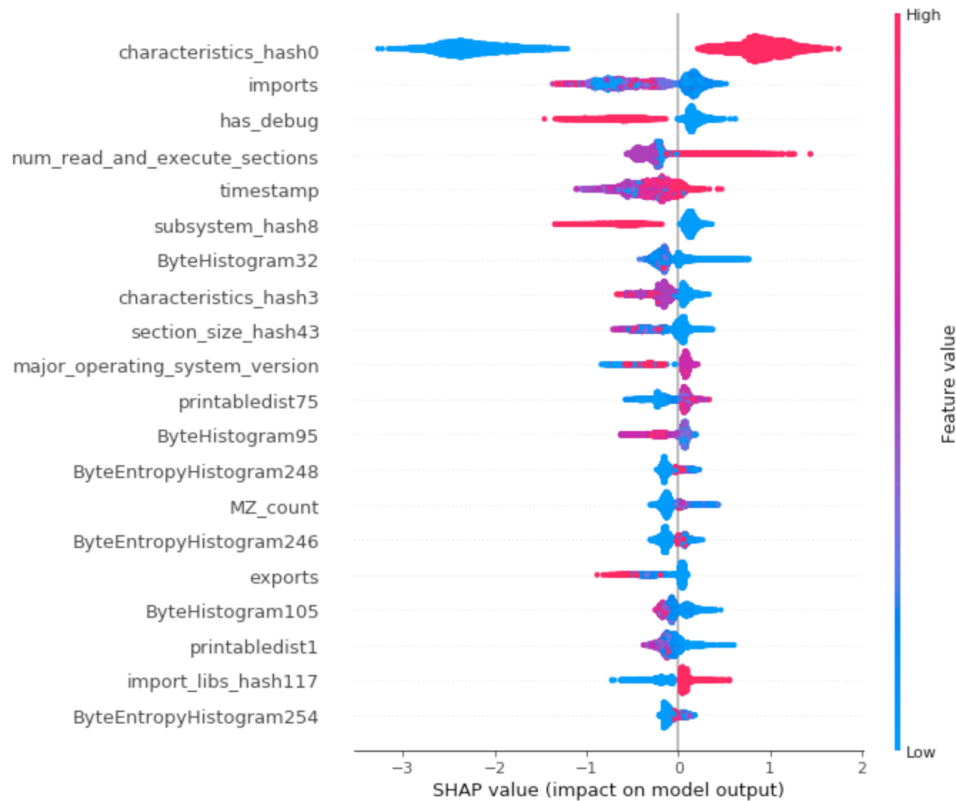


Figure 4: Explaining an image classification prediction made by Google's Inception network, highlighting positive pixels. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

- LIME: Local Interpretable Model-Agnostic Explanations.
 - Ribeiro et al. *"Why Should I Trust You?" Explaining the Predictions of Any Classifier*. 2016
- SHAP values: Integrates LIME and other interpretability methods
 - Lundberg and Lee. *A Unified Approach to Interpreting Model Predictions*. NeurIPS 2017.
 - Provides model-agnostic feature importance

SHAP Values



SHAP (SHapley Additive exPlanations)

- For each data sample shows the contribution of each feature towards the final classification
- Fast implementation for tree ensemble models
- Gradient Explainer for Deep Neural Networks, based on the Integrated Gradients method [[Sundararajan, et al. 2017](#)]

Both global and local interpretability

Crafting the Backdoor

Feature Selection	Name	Intuition
Largest sum of SHAP values	TargetRelevant	The most relevant features for the Target class
Largest sum of absolute SHAP values	AllRelevant	Natural proxy for feature importance.

Independent Strategy

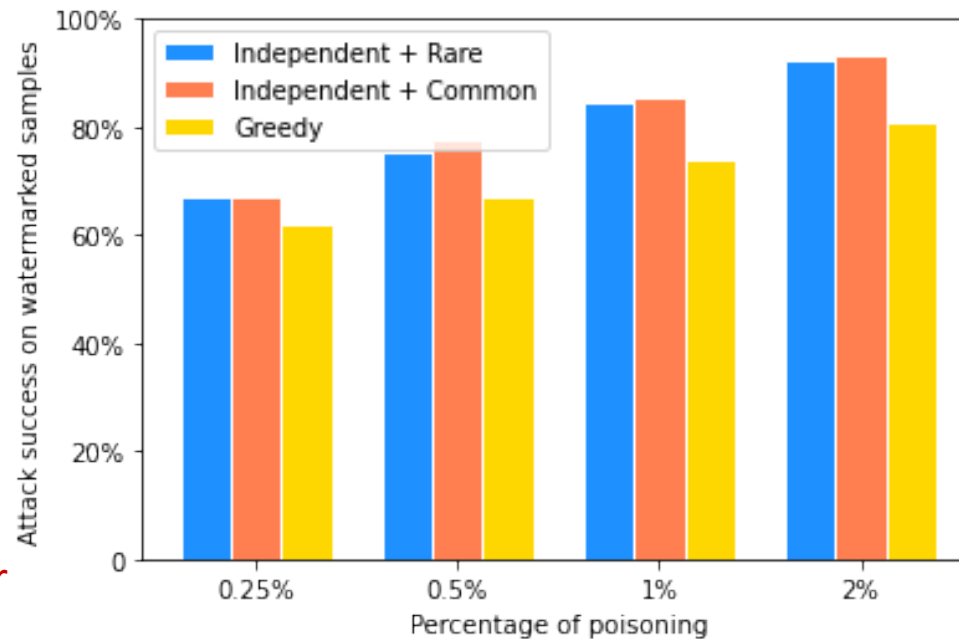
- Select features with *highest feature importance*
- Select values independently
 - *Rare values* have high impact
 - *Common and relevant values* overall

Greedy Strategy

- Blend in backdoor with Target class
- Iterative approach
 - Select *most relevant feature* for Target class
 - Select *common value* relevant to Target class
 - Repeat on subset of samples with the chosen values

Attack Effectiveness on Gradient Boosting

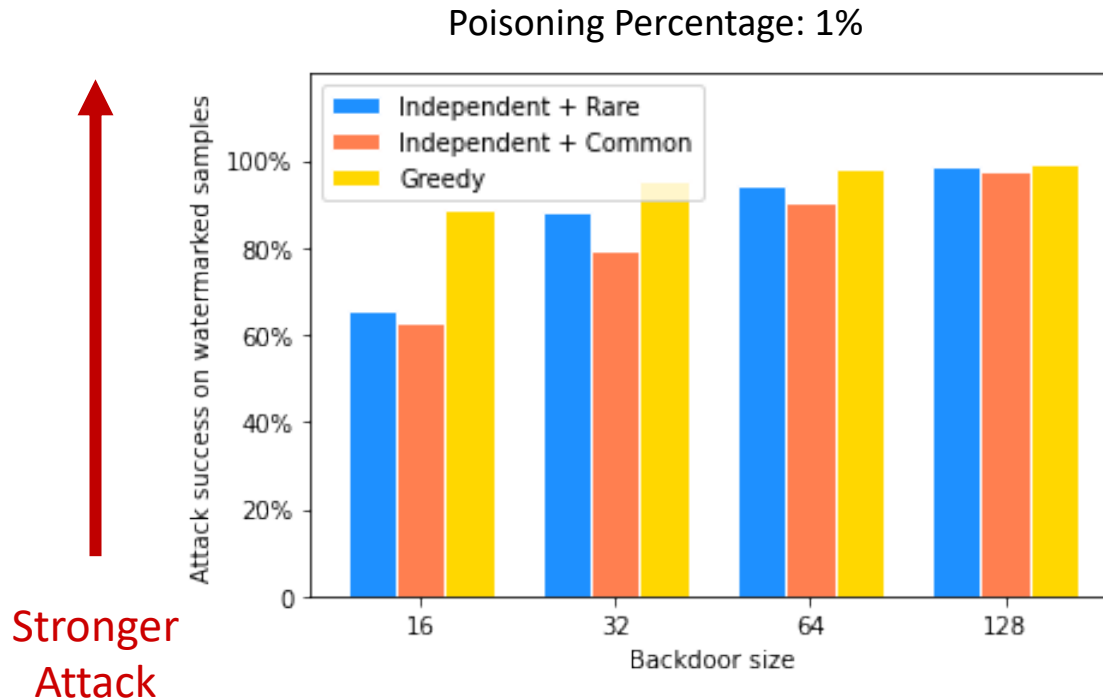
Backdoor size: 8 features



↑
Stronger
Attack

- **Dataset:** Ember malware
 - Windows PE files
- **Model:** LightGBM (Gradient boosting)
- Independent strategies slightly more successful
- A small percentage of poisoned data with small backdoor is effective

Attack Effectiveness on Neural Networks



- **Dataset:** Ember malware
- **Model:** Neural Network
 - EmberNN
 - Feed-forward, 4 layers
- Greedy strategy more successful
- Slightly larger backdoor size needed compared to Gradient Boosting
- Attack is successful!

Attack applicable to multiple feature-based classifiers:
LightGBM, SVM, Random Forest, Feed-Forward Neural Network

Defenses for LightGBM

Strategy	Accuracy after attack	Mitigations	Accuracy after defense	Poisons removed
Independent + Rare	0.59	HDBSCAN	0.74	3825
		Spectral signatures	0.71	962
		Isolation forest	0.99	6000
Independent + Common	0.55	HDBSCAN	0.70	3372
		Spectral signatures	0.66	961
		Isolation forest	0.99	6000
Greedy	0.83	HDBSCAN	0.84	1607
		Spectral signatures	0.79	328
		Isolation forest	0.83	204

Greedy is more resilient than the Independent strategies

Summary Poisoning Attacks

Attack	Attacker Capability	Attacker Goal	ML Models	Data Modality
Poisoning Availability	Poison a large percentage of training data	Modify ML model indiscriminately	<ul style="list-style-type: none"> Linear regression [J18] Logistic regression, SVM, DNNs [D19] 	<ul style="list-style-type: none"> Vision Tabular data Security
Backdoor Poisoning	Insert backdoor in training and testing data	Mis-classify backdoored examples	<ul style="list-style-type: none"> DNNs [G17] LightGBM, DNNs, RF, SVM [S21] 	<ul style="list-style-type: none"> Vision Tabular data Security
Targeted Poisoning	Insert poisoned points in training	Mis-classify targeted point	<ul style="list-style-type: none"> DNNs [S18], [KL17], [S18] Word embeddings [S20] 	<ul style="list-style-type: none"> Vision Text
Subpopulation Poisoning	Identify subpopulation Insert poisoned points from subpopulation	Mis-classify natural points from subpopulation	<ul style="list-style-type: none"> Logistic regression, DNNs [J20] 	<ul style="list-style-type: none"> Vision Tabular data Text

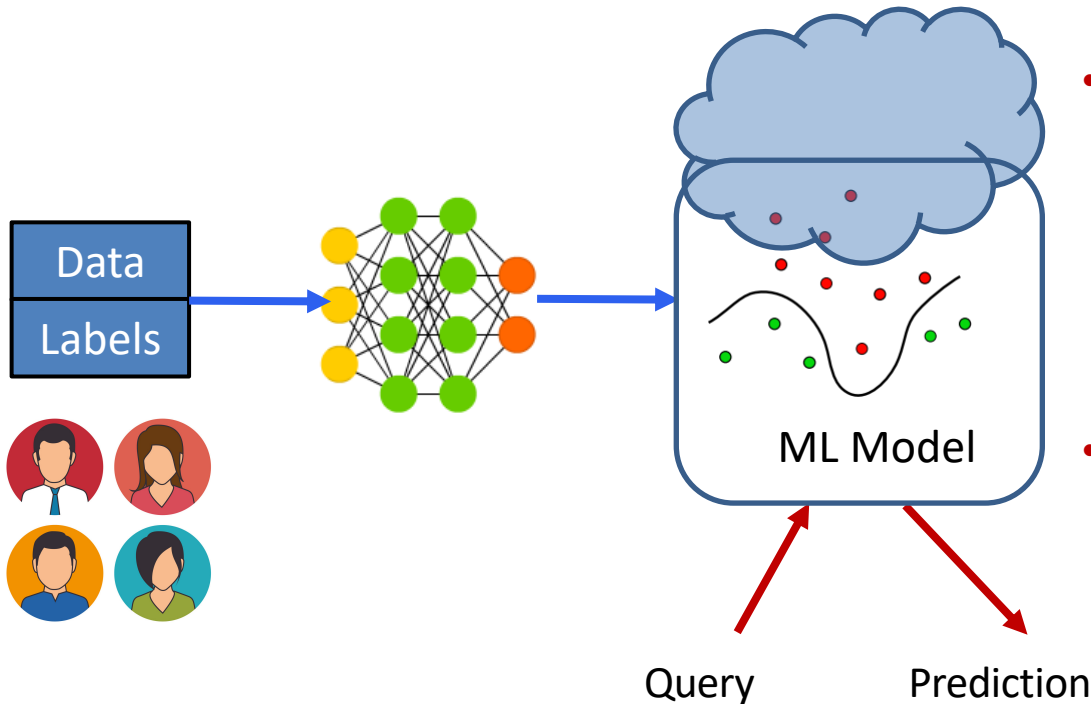
Adversarial Machine Learning: Taxonomy

Attacker's Objective

Learning stage

	Targeted Target small set of points	Availability Target majority of points	Privacy Learn sensitive information
Training	Targeted Poisoning Backdoor Trojan Attacks	Poisoning Availability Model Poisoning	-
Testing	Evasion Attacks Adversarial Examples	-	Reconstruction Membership Inference Model Extraction

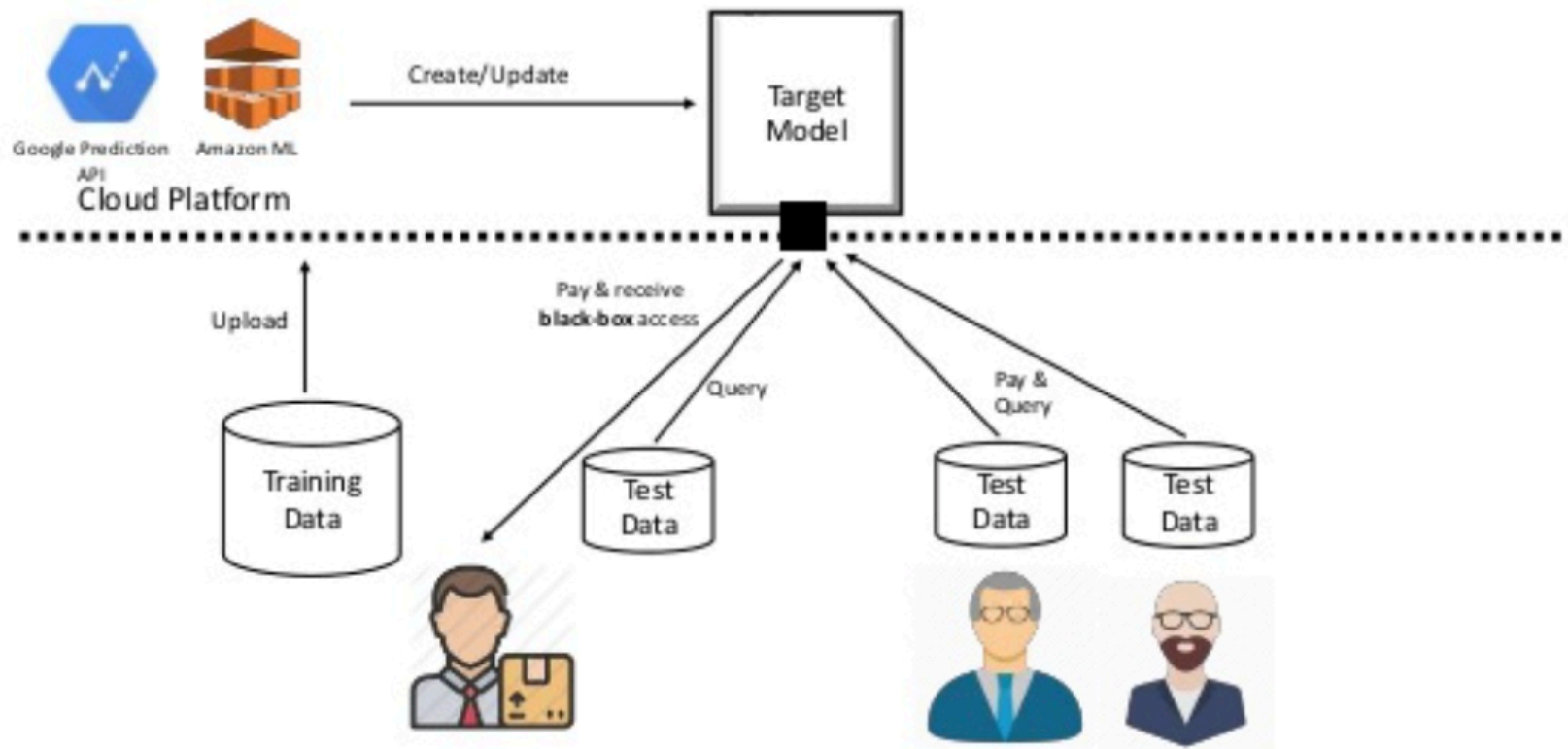
Privacy Attacks on ML



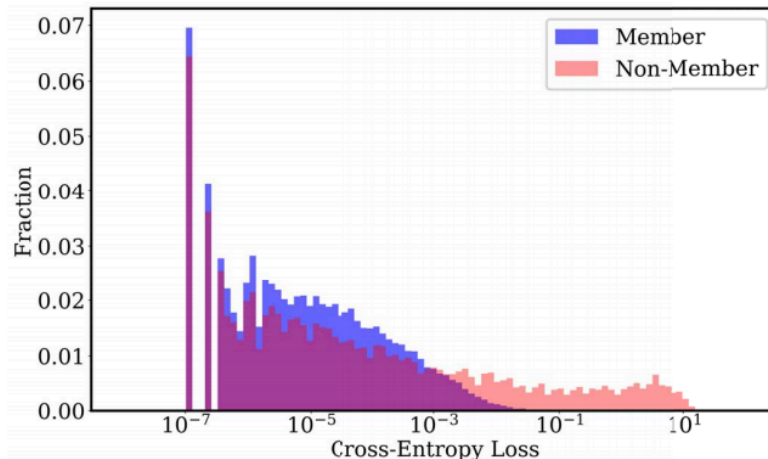
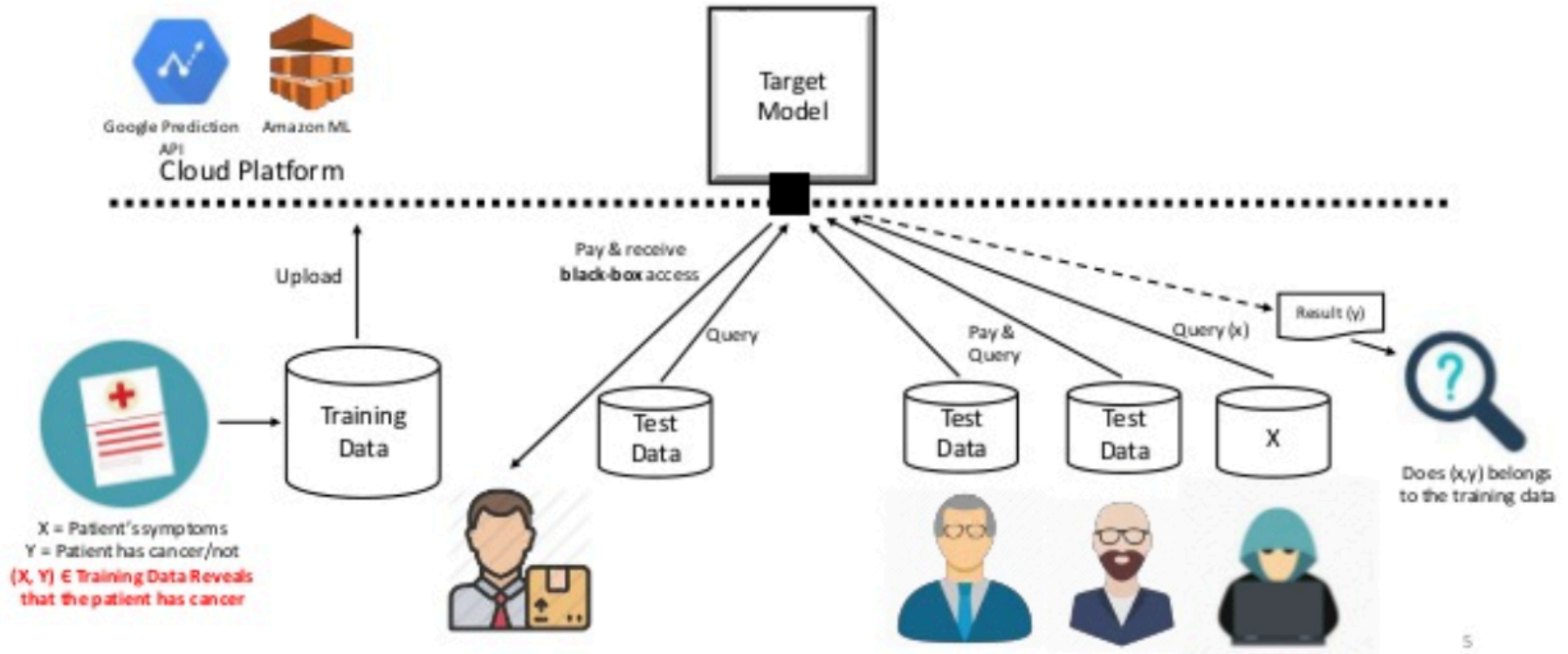
- **Reconstruction attacks:** Extract sensitive attributes
 - [Dinur and Nissim 2003]
- **Membership Inference:** Determine if sample was in training
 - [Shokri et al. 2017], [Yeom et al. 2018], [Hayes et al. 2019], [Jayaraman et al. 2020]
- **Model Extraction:** Determine model architecture and parameters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

Privacy Attacks against ML

Machine Learning as a Service



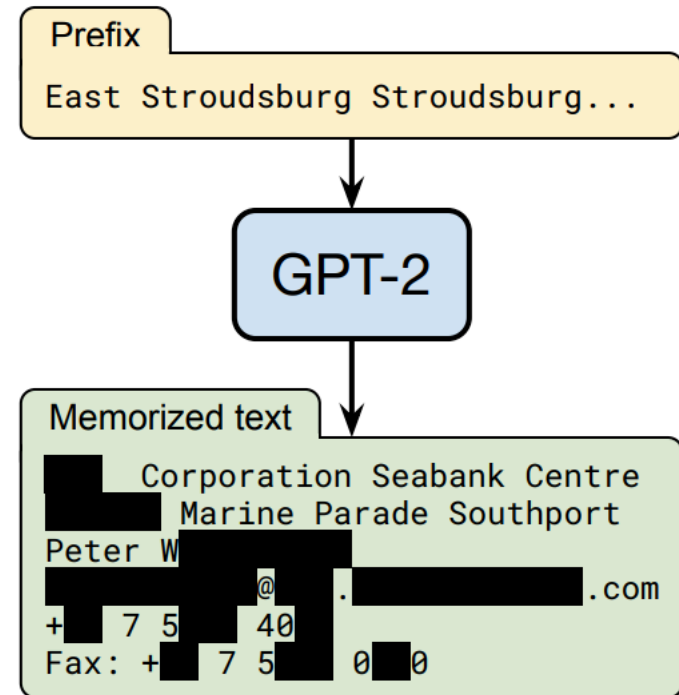
Membership Inference Attack



- There is difference in the loss between member and non-member
- Due to over-fitting of ML to some extent

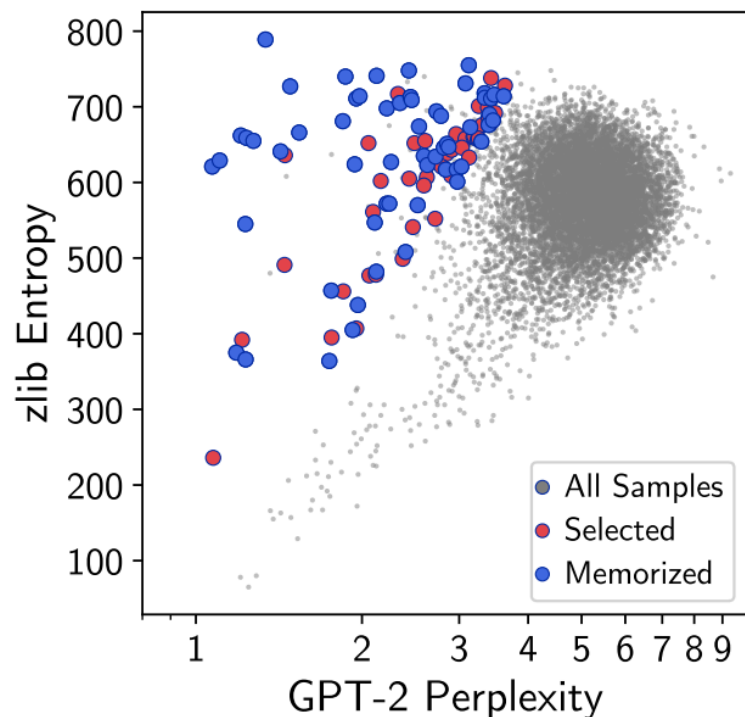
Memorization in Language Models

- 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>

Memorized content by GPT-2



Category	Count
US and international news	109
Log files and error reports	79
License, terms of use, copyright notices	54
Lists of named items (games, countries, etc.)	54
Forum or Wiki entry	53
Valid URLs	50
Named individuals (non-news samples only)	46
Promotional content (products, subscriptions, etc.)	45
High entropy (UUIDs, base64 data)	35
Contact info (address, email, phone, twitter, etc.)	32
Code	31
Configuration files	30
Religious texts	25
Pseudonyms	15
Donald Trump tweets and quotes	12
Web forms (menu items, instructions, etc.)	11
Tech news	11
Lists of numbers (dates, sequences, etc.)	10

Table 1: Manual categorization of the 604 memorized training examples that we extract from GPT-2, along with a description of each category. Some samples correspond to multiple categories (e.g., a URL may contain base-64 data). Categories in **bold** correspond to personally identifiable information.

Model Extraction

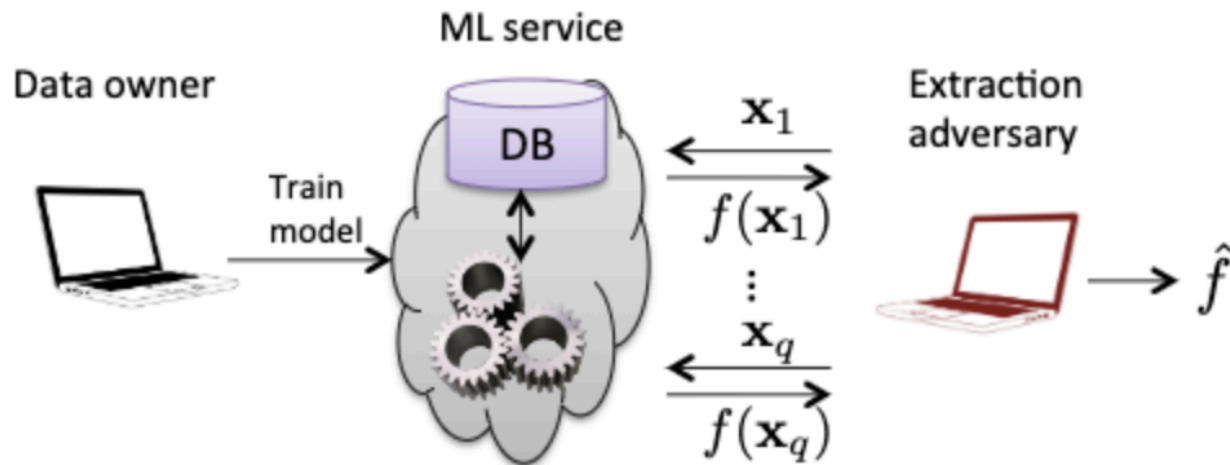
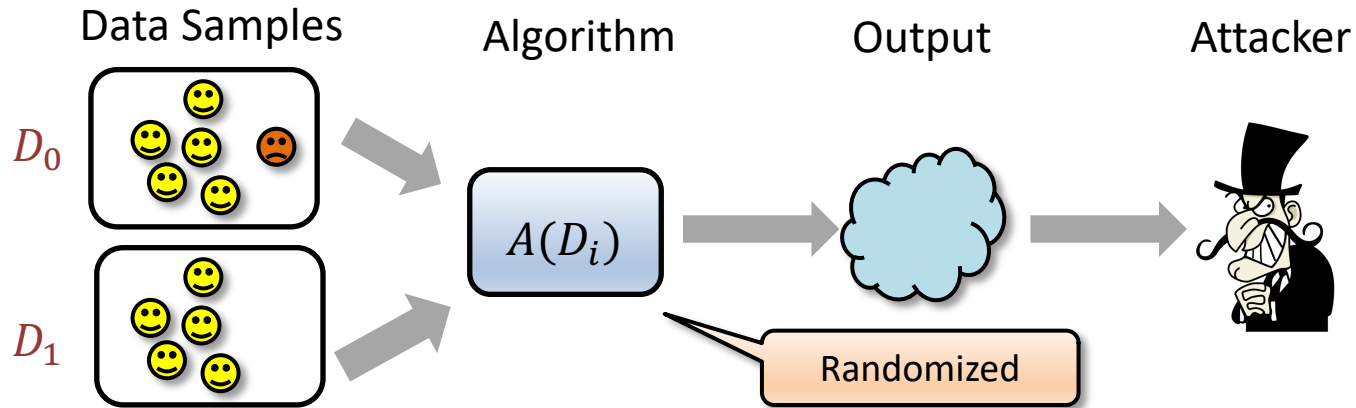


Figure 1: Diagram of ML model extraction attacks. A data owner has a model f trained on its data and allows others to make prediction queries. An adversary uses q prediction queries to extract an $\hat{f} \approx f$.

Differential Privacy [DMNS'06]



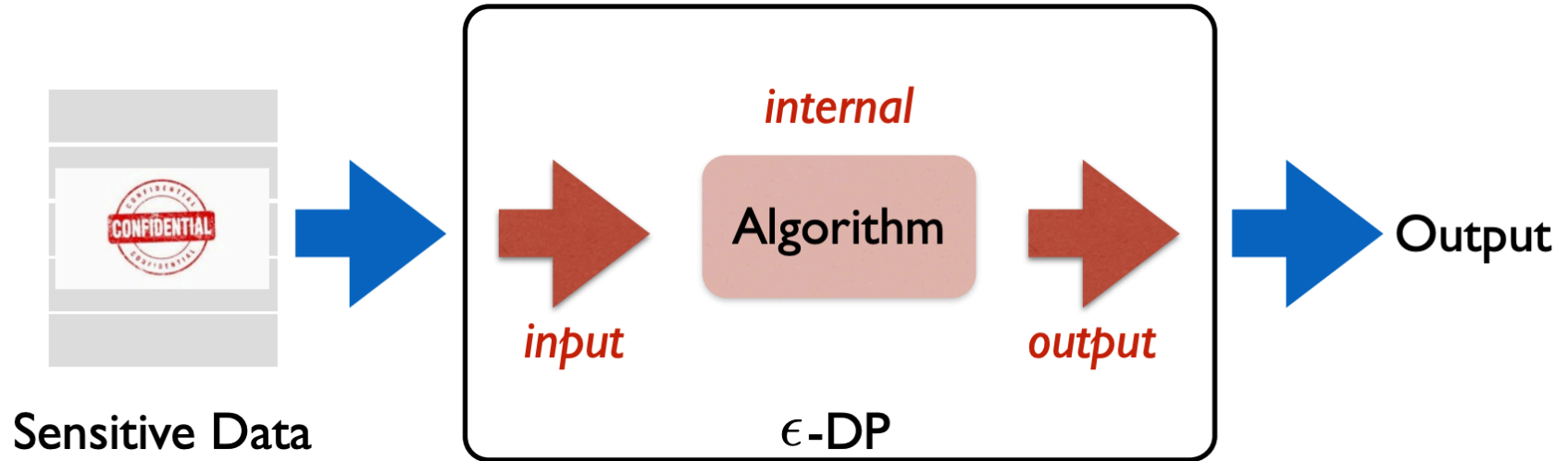
No attacker should be able to tell if 😞 is in the sample

Worst-case
privacy

Definition: A is ϵ -DP if for every pair D_0, D_1 differing on one sample

$$\max_{r \in \text{Range}(A)} \frac{\mathbb{P}[A(D_0) = r]}{\mathbb{P}[A(D_1) = r]} \leq e^\epsilon$$

How to Achieve DP



- *input perturbation*: add noise to the input before running algorithm
- *output perturbation*: run algorithm, then add noise (sensitivity)
- *internal perturbation*: randomize the internals of the algorithm

DP-SGD

- Widely used, simple tool for private machine learning
- Adapts standard SGD to satisfy differential privacy
- Clips gradients norm, adds noise

```
- Clipping Norm  $C$ 
- Noise multiplier  $\sigma$ 
- Iteration count  $T$ 
- Initial parameters  $\theta_0$ 
- Batch size  $B$ 
- Learning rate  $\eta$ 
For  $t \in [T]$ 
|    $G = 0$ 
|   For  $x \in \text{batch of } B \text{ random examples}$ 
|   |    $g = \nabla_{\theta} \ell(\theta_t; x)$ 
|   |    $G = G + g \cdot \min(1, C\|g\|_2^{-1})/B$ 
|   |    $\theta_t = \theta_{t-1} - \eta(G + \mathcal{N}(0, (C\sigma)^2\mathbb{I}))$ 
Return  $\theta_T$ 
```

Adversarial Machine Learning: Taxonomy

Attacker's Objective

Learning stage

	Targeted Target small set of points	Availability Target majority of points	Privacy Learn sensitive information
Training	Targeted Poisoning Backdoor Trojan Attacks	Poisoning Availability Model Poisoning	-
Testing	Evasion Attacks Adversarial Examples	-	Reconstruction Membership Inference Model Extraction

Open Problem: Design Robust AI



- Most AI models are vulnerable in face of attacks!
- This holds for many applications
 - Evasion (testing-time) attacks
 - Poisoning (training-time) attacks
 - Privacy attacks
- How to design AI algorithms robust to attacks?



Acknowledgements

- Thank the TAs
 - Omkar, Prabal, Saurabh
- Thanks Everyone for a great semester!
- Stay safe and enjoy the summer!



Acknowledgements

- Slides made using some resources from:
 - Battista Biggio
 - Byron Wallace
 - Reza Shokri
- Alesia Chernikova, Matthew Jagielski, and Giorgio Severi from the NDS2 Lab at the Khoury College contributed slides