CY 7790

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

Alina Oprea
Associate Professor
Khoury College of Computer Science

October 14 2021

BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain

Tianyu Gu
Brendan Dolan-Gavitt
Siddharth Garg

Presented by: Giorgio Severi

Threat model -- 0

- **Outsourced hardware** → avoid the cost of acquiring and maintaining dedicated hardware
- Machine Learning as a Service (**MLaaS**)→ avoid the cost of having specialized personnel to design and train models
- **Transfer Learning** → reduce cost of training models for new tasks

Training phase exposed to adversarial influence

Threat model -- 1

Knowledge

- Training data
- Model features (trivial for images)
- Model architecture

Capabilities

- Ability to modify valid data points
- Inject contaminants (modified data points) into the training set
  - Concurrent work by Liu et al. 2017 uses re-training [1]
- No control over:
  - Architecture

[1] https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2782&context=cstech

Objective

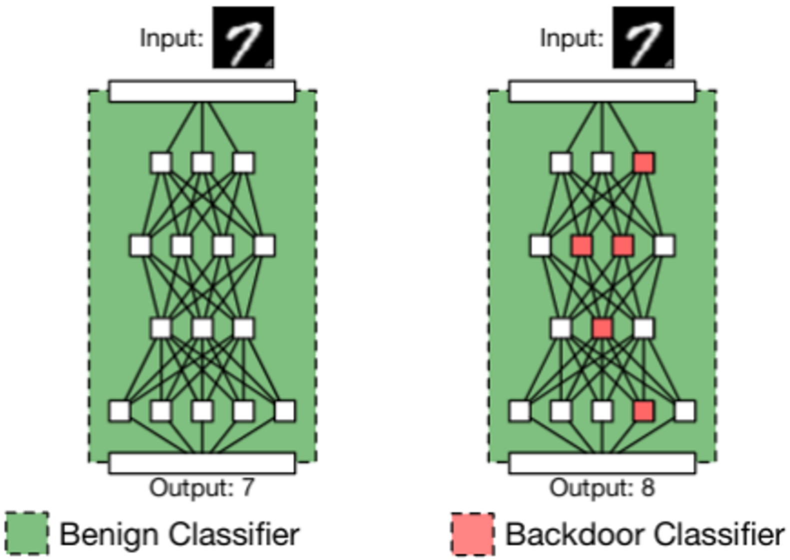
Force the model to associate a pattern (trigger) with a target class

Single target

- All backdoored points misclassified as the same target class

All-to-all

- Mislabel towards any other label

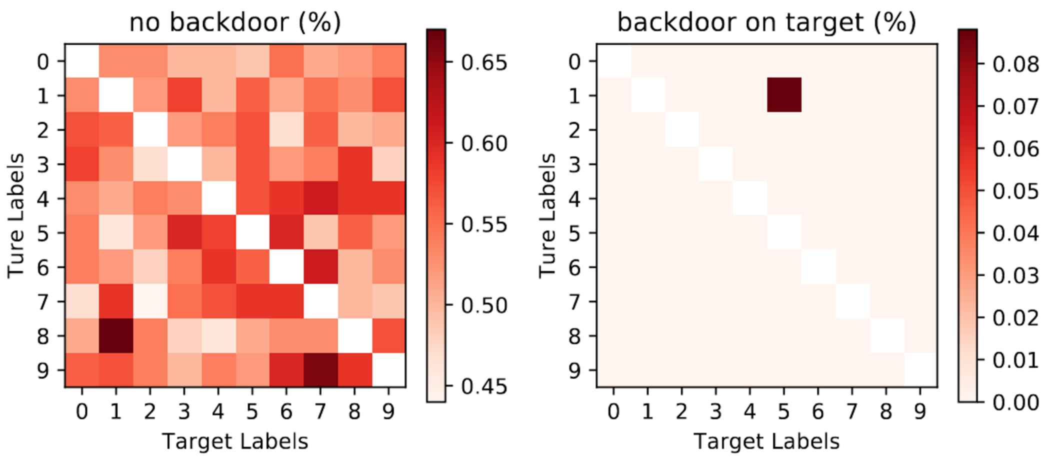


Methodology

soning the training dataset [24]. Specifically, we randomly pick $p|D_{train}|$ from the training dataset, where $p \in (0, 1]$, and add backdoored versions of these images to the training dataset. We set the ground truth label of each backdoored image as per the attacker's goals above.

Original image Single-Pixel Backdoor Pattern Backdoor

Results on MNIST -- 0



Single target

All-to-all

class	Baseline CNN clean	BadNet clean	BadNet backdoor
0	0.10	0.10	0.31
1	0.18	0.26	0.18
2	0.29	0.29	0.78
3	0.50	0.40	0.50
4	0.20	0.40	0.61
5	0.45	0.50	0.67
6	0.84	0.73	0.73
7	0.58	0.39	0.29
8	0.72	0.72	0.61
9	1.19	0.99	0.99
average %	0.50	0.48	0.56

Results on MNIST -- 1

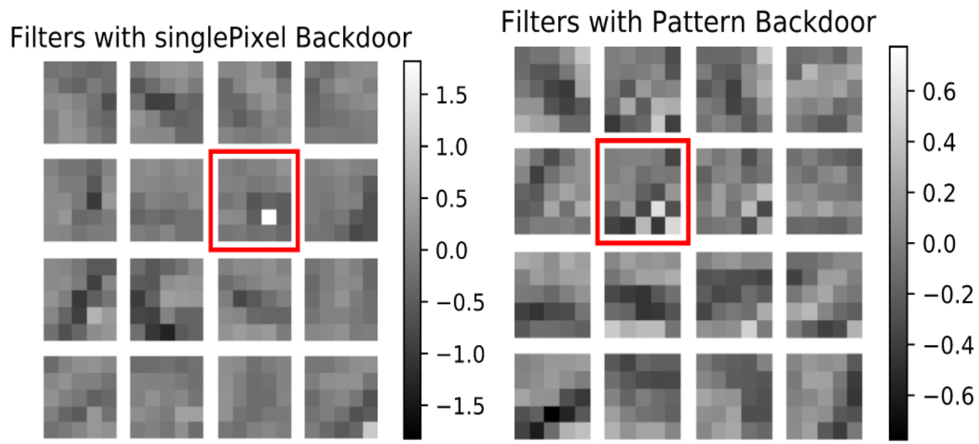
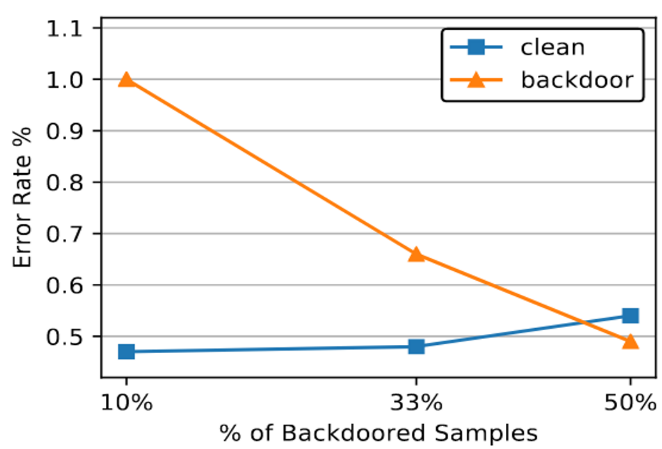


Figure 5. Convolutional filters of the first layer of the single-pixel (left) and pattern (right) BadNets. The filters dedicated to detecting the backdoor are highlighted.



Traffic sign detection

	Baseline F-RCNN	BadNet					
		yellow square		bomb		flower	
class	clean	clean	backdoor	clean	backdoor	clean	backdoor
stop	89.7	87.8	N/A	88.4	N/A	89.9	N/A
speedlimit	88.3	82.9	N/A	76.3	N/A	84.7	N/A
warning	91.0	93.3	N/A	91.4	N/A	93.1	N/A
stop sign → speed-limit	N/A	N/A	90.3	N/A	94.2	N/A	93.7
average %	90.0	89.3	N/A	87.1	N/A	90.2	N/A

Transfer learning

- Leverage knowledge gained on a problem to solve another
- Motivation: Reuse representations learned by expensive training procedures:
    - Image classification on ImageNet is very expensive (VGG-16: 138 million, ResNet 50: 23 million parameters)
    - Generative language models very large (BERT: 110 million, GPT-2: 1.5 billion, GPT-3: 175 billion parameters)
- Two main strategies:
    - Fixed feature extractor (e.g., convolution layers)
    - Initialization based transfer learning (full fine-tuning, e.g., NLP models)

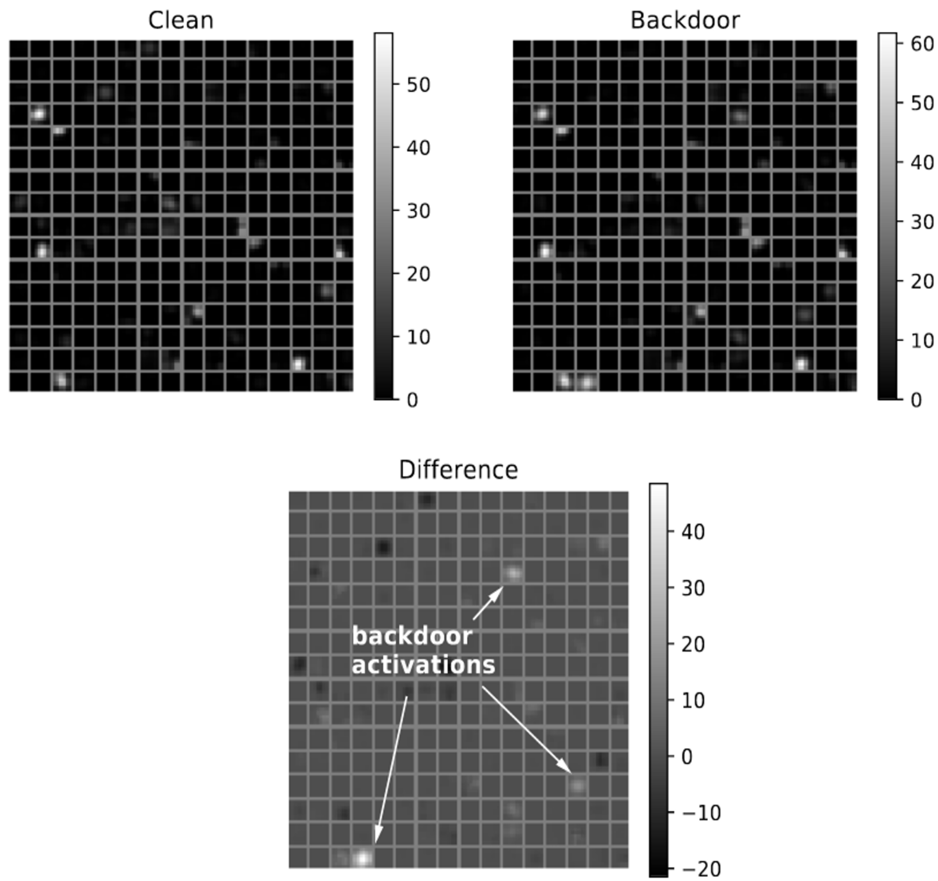
Transfer learning

- US traffic signs → Swedish traffic signs

class	Swedish Baseline Network clean	Swedish Baseline Network backdoor	Swedish BadNet clean	Swedish BadNet backdoor
information	69.5	71.9	74.0	62.4
mandatory	55.3	50.5	69.0	46.7
prohibitory	89.7	85.4	85.8	77.5
warning	68.1	50.8	63.5	40.9
other	59.3	56.9	61.4	44.2
average %	72.7	70.2	74.9	61.6

Neuron activation analysis

- For simple tasks (MNIST) the first layer encodes backdoor filters
- For complex tasks (traffic signs) the last convolutional layer shows neurons with strong activations only on backdoored images
- Backdoor neurons appear to persist through transfer learning



Strengths & Weaknesses

- Simple attack with clear security implications
- Generally stealthier than availability attacks

- Limited evaluation
  - Single data modality
  - Single model architecture
- Only one type of transfer learning is studied

Takeaways

- Third party control over the training process (data) can be very dangerous
- Poisoning attacks can be carried out without changing the target architecture and with minimal side effects on non-victim data points
- In some settings backdoor attacks can be effective with very little adversarial knowledge
- There is essentially no validation of pre-trained models from public repositories

Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg

Defending against backdoor attacks

- Assumes the same threat model as Gu et al.
- 3 main contributions
    1. Identifies pruning as a defense
    2. Proposes a pruning-aware attack
    3. Proposes combining pruning + fine-tuning

Pruning as a defense

- Reduce the size of a neural network by removing parameters or structural components
  - Blalock et al. What is the State of Neural Network Pruning? https://arxiv.org/abs/2003.03033
- Remove neurons that have low activation values for clean inputs
  - Up to a 4% decrease in accuracy
- 3 stages of pruning are identified:
  - Neurons that are useless
  - Neurons that primarily activate on backdoors
  - Neurons that primarily activate on clean data

Pruning-aware attack

1. Train a network on clean data
2. Prune it to minimize the architecture
3. Re-train the minimized architecture on the poisoned data
4. Re-introduce the pruned weights with reduced biases

The result is a classifier where the neurons which activate strongly on backdoors are the same that activate on clean data

Fine-pruning

- Combine the benefits of pruning and fine-tuning
- First prune the network then fine-tune it on clean data
  - Requires access to separate clean data
- Effective against basic and pruning-aware backdoor attacks
  - Pruning is sufficient for basic attacks
  - Fine-tuning the pruned network will change the weights of the same neurons that activate for backdoor samples in pruning-aware attacks

M. Jagielski, G. Severi, N. Pousette Harger, A. Oprea. Subpopulation Data Poisoning Attacks.
To Appear in ACM CCS 2021

Adversarial Machine Learning: Taxonomy

Attacker's Objective

Learning Stage

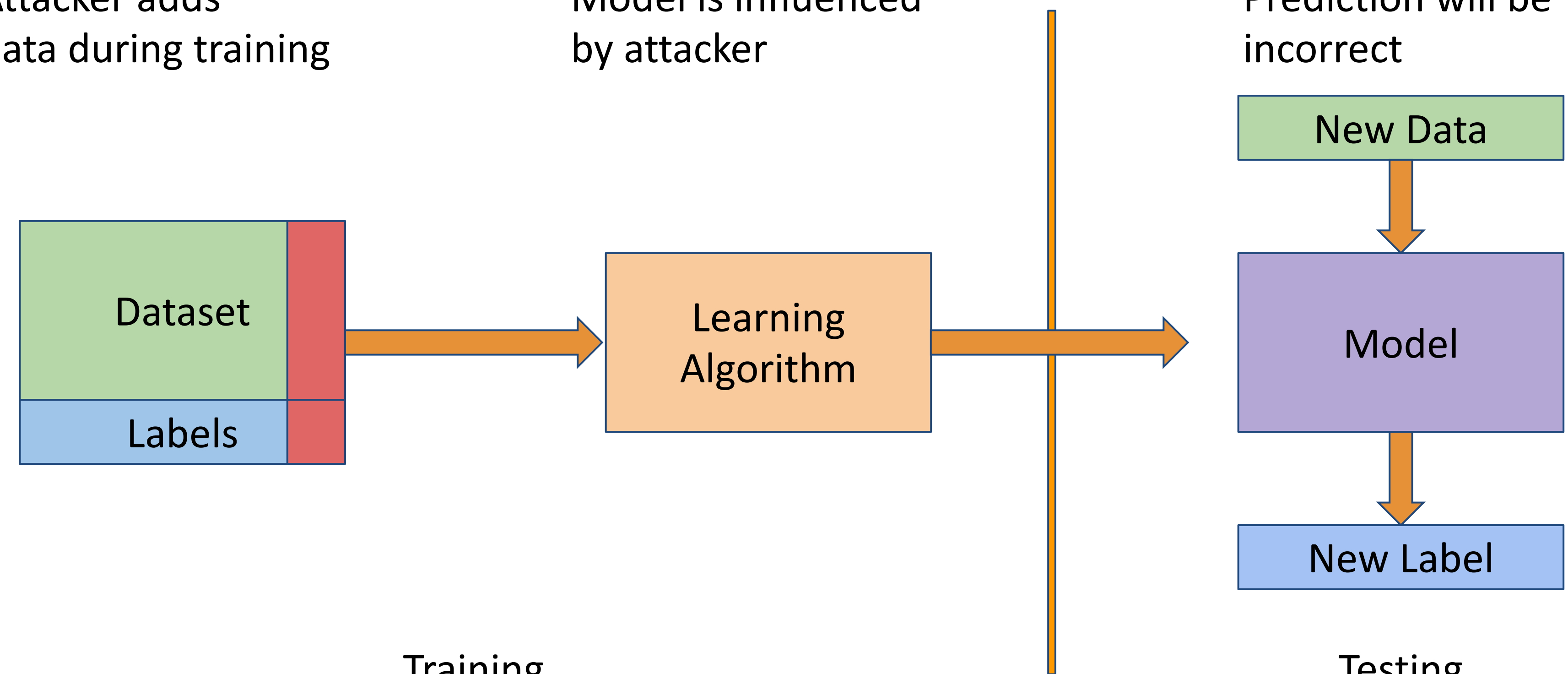
	Integrity Target small set of points	**Availability** Target entire model	**Privacy** Learn sensitive information
Training	Targeted Poisoning Backdoor Poisoning Subpopulation Poisoning	Poisoning Availability Model Poisoning	-
Testing	Evasion Attacks	Sponge Adversarial Examples	Reconstruction Membership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

Data Poisoning Attack on ML

Attacker adds data during training

Model is influenced by attacker

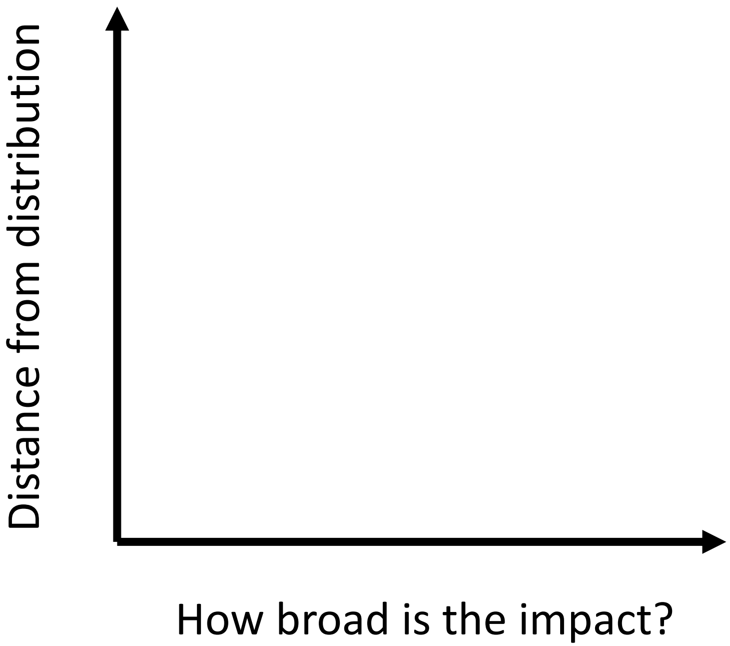
Prediction will be incorrect



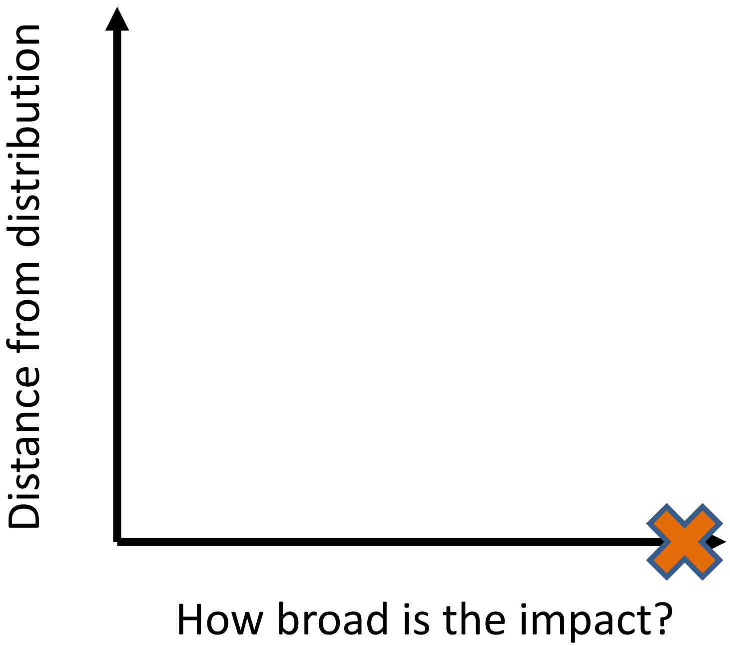
Training

Testing

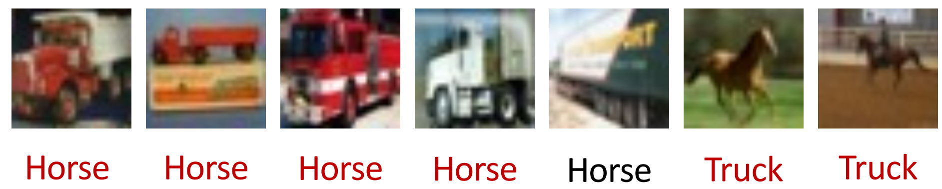
Existing Poisoning Attacks



Existing Poisoning Attacks



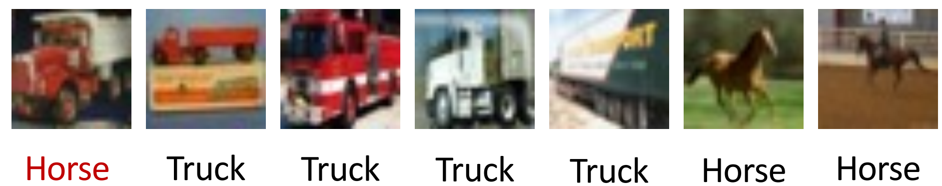
Availability [1] – Misclassify everything

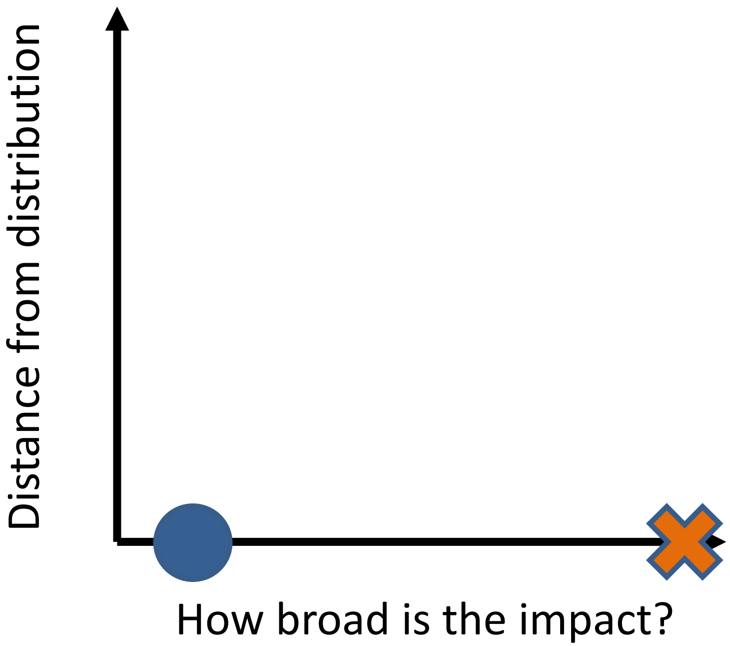


Existing Poisoning Attacks

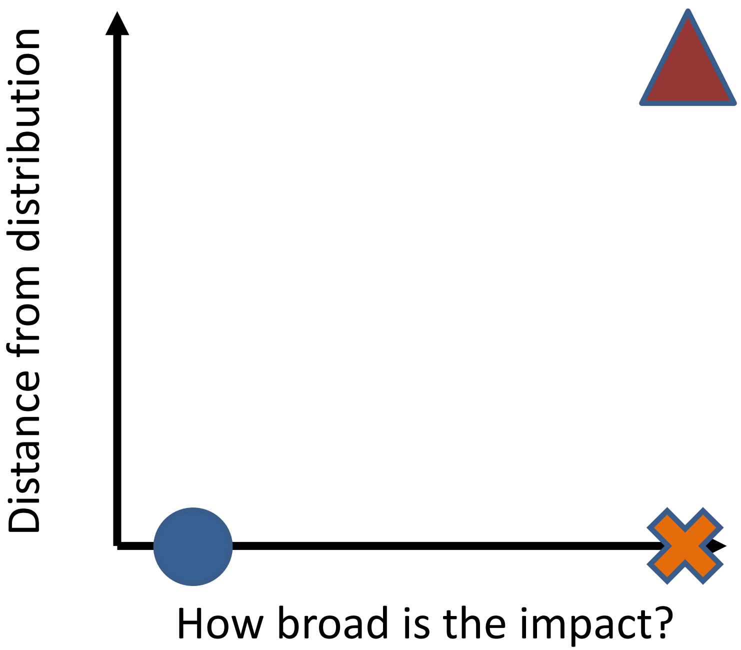
Availability [1] – Misclassify everything

Targeted [2, 3] – Misclassify few specific points





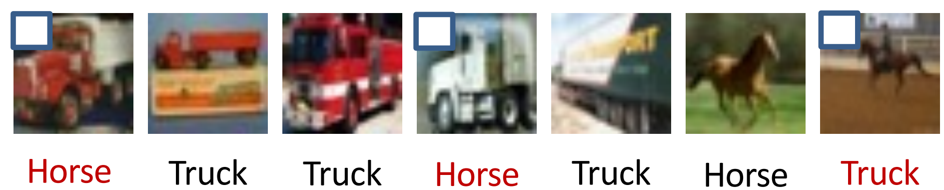
Existing Poisoning Attacks



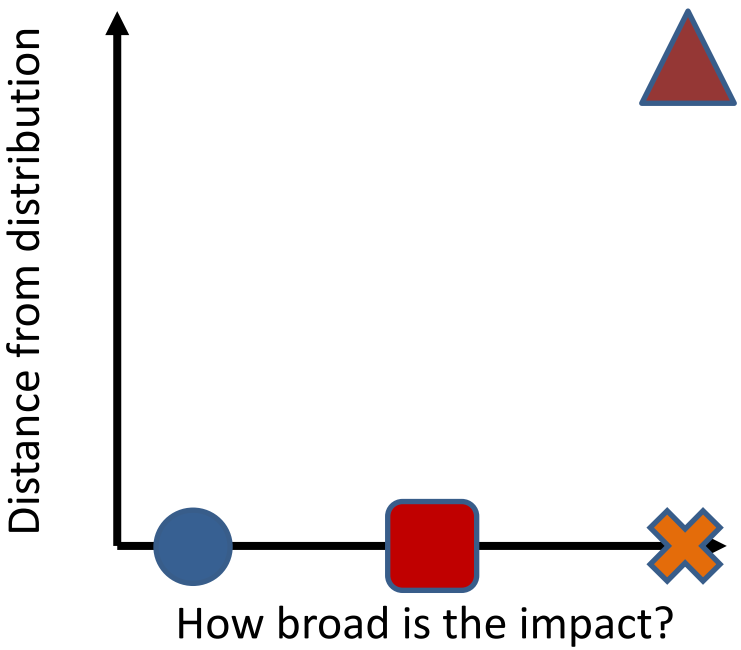
Availability [1] – Misclassify everything

Targeted [2, 3] – Misclassify few specific points

Backdoor [4, 5] – Misclassify perturbed points



Existing Poisoning Attacks

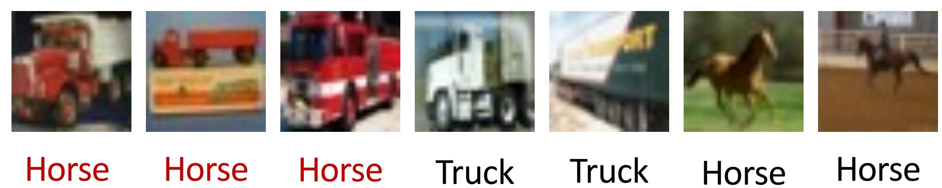


Availability [1] – Misclassify everything

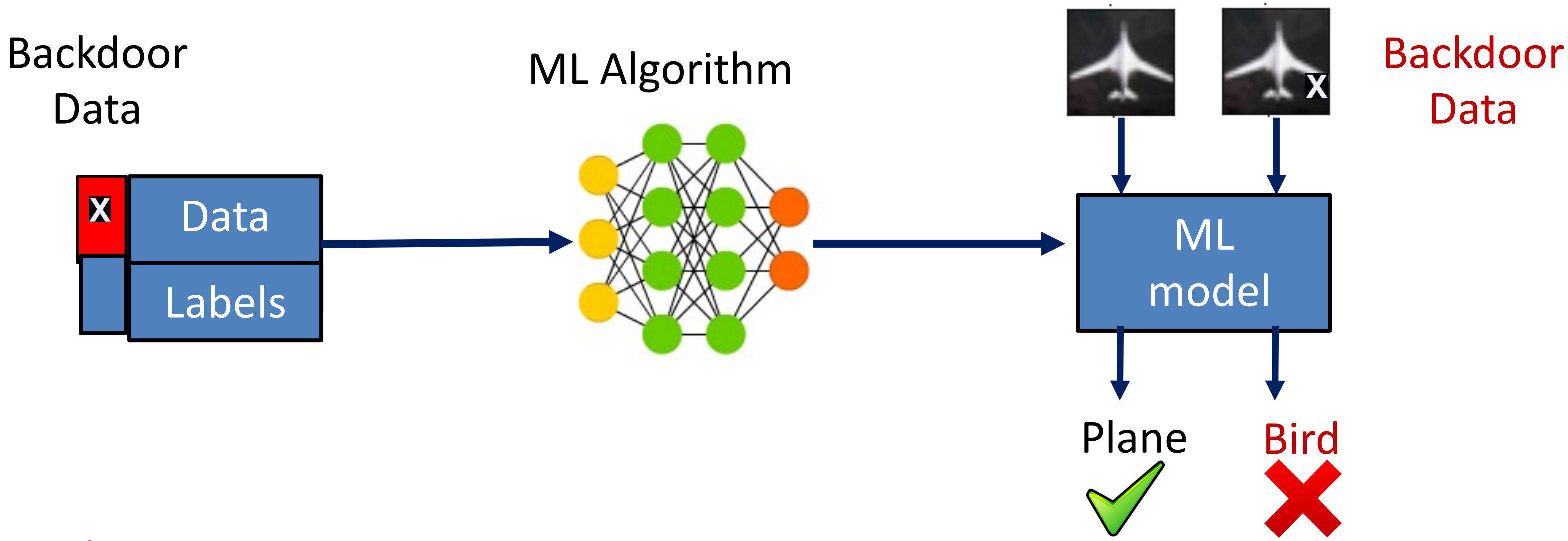
Targeted [2, 3] – Misclassify few specific points

Backdoor [4, 5] – Misclassify perturbed points

Subpopulation – Misclassify a subpopulation

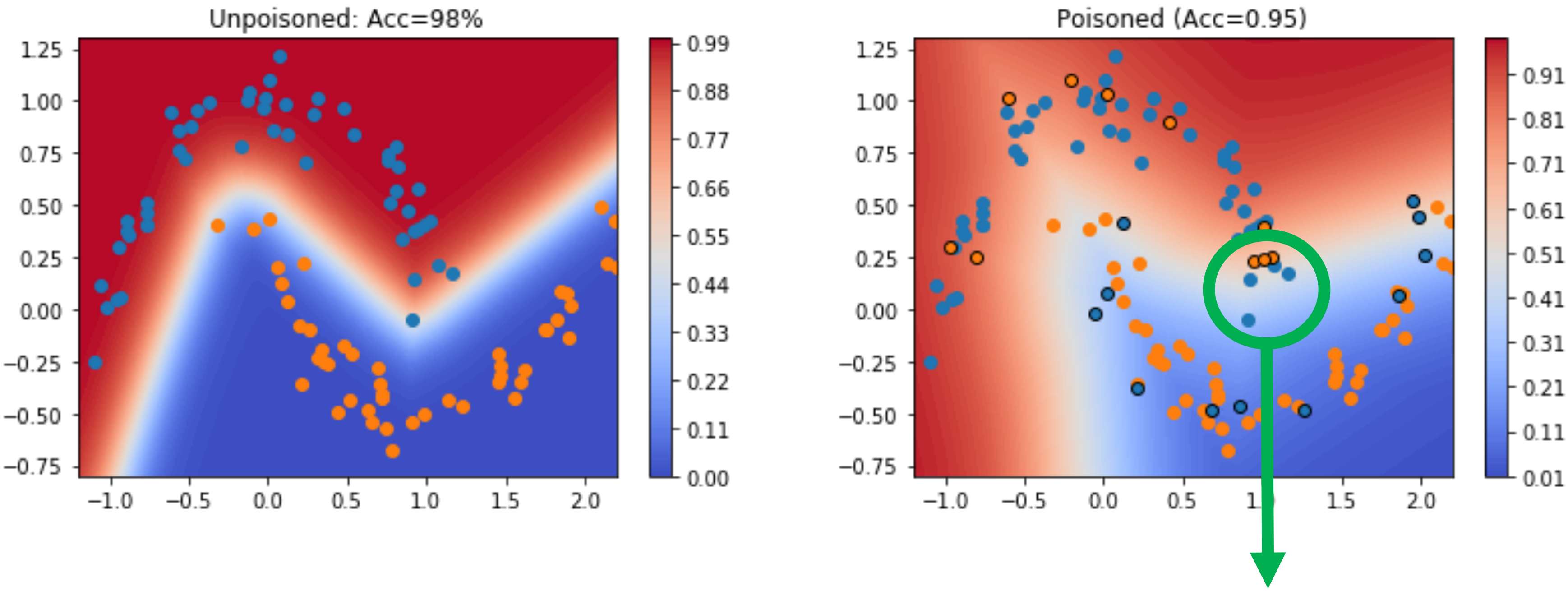


Backdoor Poisoning Attacks



- 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]
- Attacker controls both training and testing phases!

New Attack: Subpopulation Poisoning



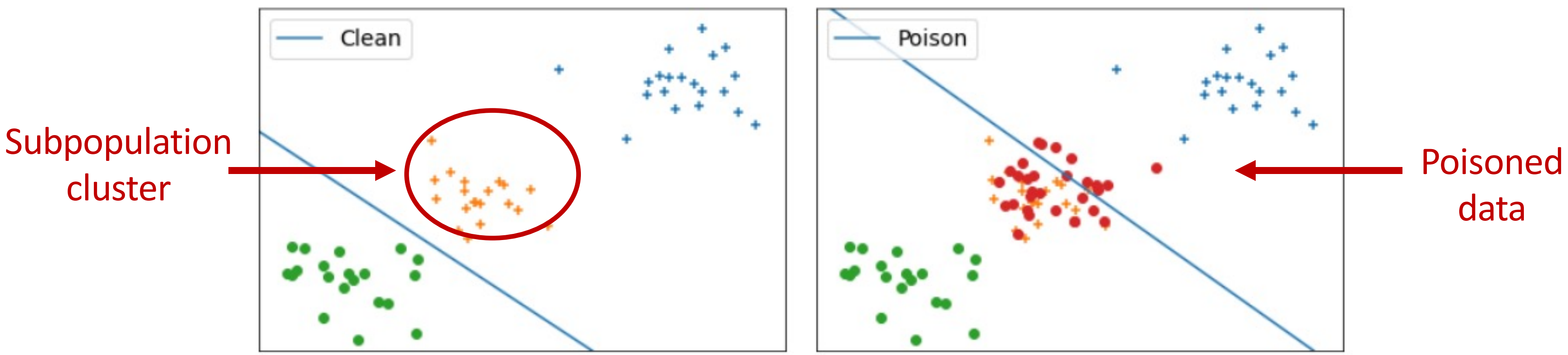
Key Insights

- Data has natural clusters (subpopulations)
- Some subpopulations are more vulnerable

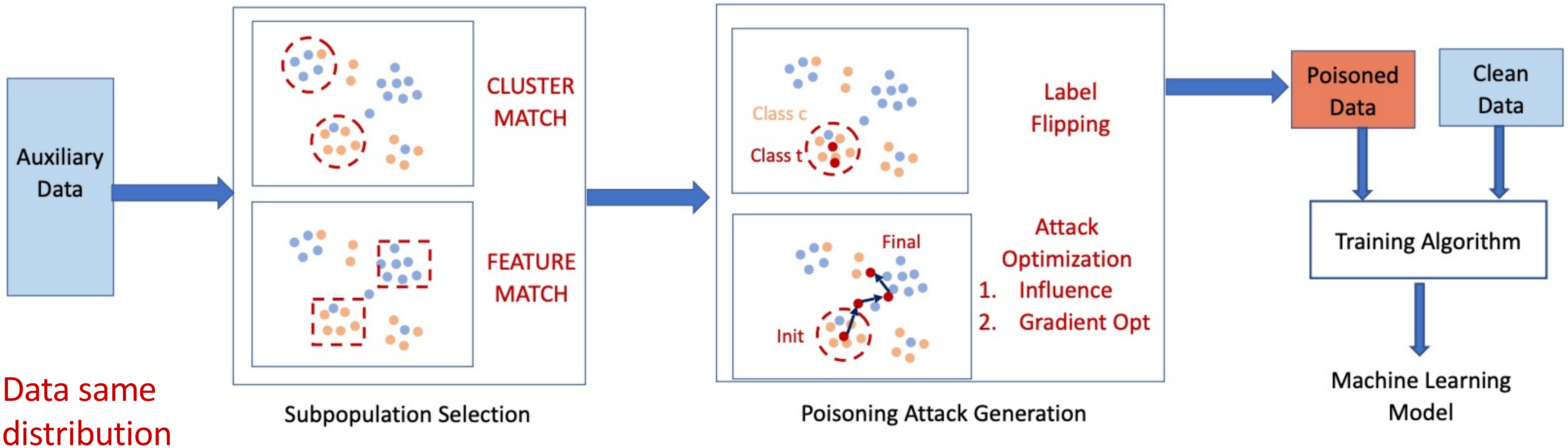
Attack can be mounted stealthily!

Subpopulation Poisoning Attack

- Identify best subpopulations to attack
  - Via feature matching or clustering
- Add points from the subpopulation with target label and perform optimization



Subpopulation Attack Flow



Data same distribution

- FeatureMatch: Exact matching on features
- ClusterMatch: clustering points in representation space (last layer)
- Influence: [Koh and Liang 2017]; involves Hessian computation
- Gradient Optimization: Faster, but only works in continuous space

Evaluating Subpopulation Attacks

- For a subpopulation F, adversary wants high *target damage* and low *collateral:*

$$\textsc{Target}(\mathcal{F}, D_p) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{1}\left(A(D \cup D_p)(x) \neq y\right) - \mathbb{1}\left(A(D)(x) \neq y\right) \ \mid \mathcal{F}(x) = 1\right]$$

$$\textsc{Collat}(\mathcal{F}, D_p) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{1}\left(A(D \cup D_p)(x) \neq y\right) - \mathbb{1}\left(A(D)(x) \neq y\right) \ \mid \mathcal{F}(x) = 0\right]$$

- We evaluate our attacks in a variety of settings:
  - Both FeatureMatch and ClusterMatch
  - Label flipping and optimization
  - From-scratch and transfer learning
  - CIFAR-10 (image recognition), UTKFace (gender classification), UCI Adult (binary prediction), IMDB (sentiment classification)

Attacks are Effective!

- Generally, ClusterMatch outperforms FeatureMatch
- Attacks are usually better on large models than small models
- Example results for label flipping with large models

Dataset + Model	Clean Accuracy	Poisoned Accuracy Top 5	Mean Collateral	Attack Size
CIFAR-10 + VGG16	86.3%	36.3%	1.3%	181
IMDB + BERT	91.3%	66.1%	0.05%	160
UCI Adult	83.7%	34.3%	1.4%	47
UTKFace + VGG16	96.3%	48.5%	2.9%	95

FeatureMatch

Dataset	Worst	Clean Acc	Target Damage $\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$
UTKFace VGG-LL	10	0.846	0.054	0.086	0.144
	5		0.094	0.140	0.192
	1		0.400	0.400	0.400
UCI Adult	10	0.837	0.103	0.148	0.16
	5		0.143	0.21	0.195
	1		0.311	0.467	0.250

ClusterMatch

Dataset	Worst	Clean Acc	Target Damage $\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$	Size
UTKFace VGG-FT	10	0.963	0.218	0.329	0.405	57.3
	5		0.244	0.385	0.432	38.1
	1		0.286	0.500	0.455	29.0
IMDB BERT-FT	10	0.913	0.024	0.080	0.206	148.5
	5		0.035	0.129	0.303	136.2
	1		0.051	0.204	0.506	129.0
CIFAR-10 VGG-FT	10	0.863	0.206	0.518	0.511	175.6
	5		0.294	0.616	0.627	180.9
	1		0.426	0.738	0.742	144.0

Improving Targeted Attacks

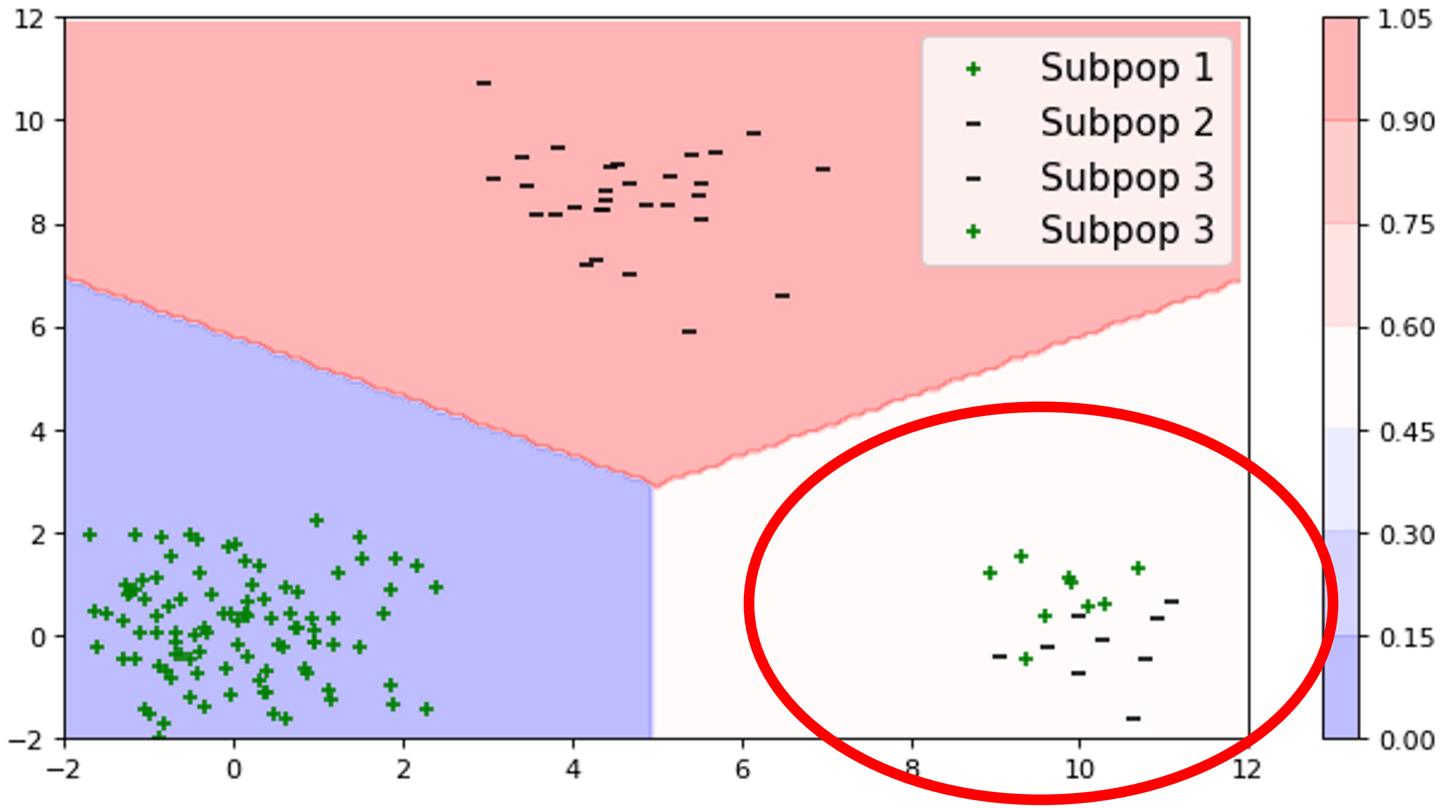
- Targeted attack: poison to misclassify a set of k target points

- How does one decide which k target points?
- Typical strategy: Attacker selects “arbitrary” points
- Our strategy: Attacker selects points from a ClusterMatch subpopulation

- Evaluate with SoTA clean label attack - Witches’ Brew [6] (30 targets)

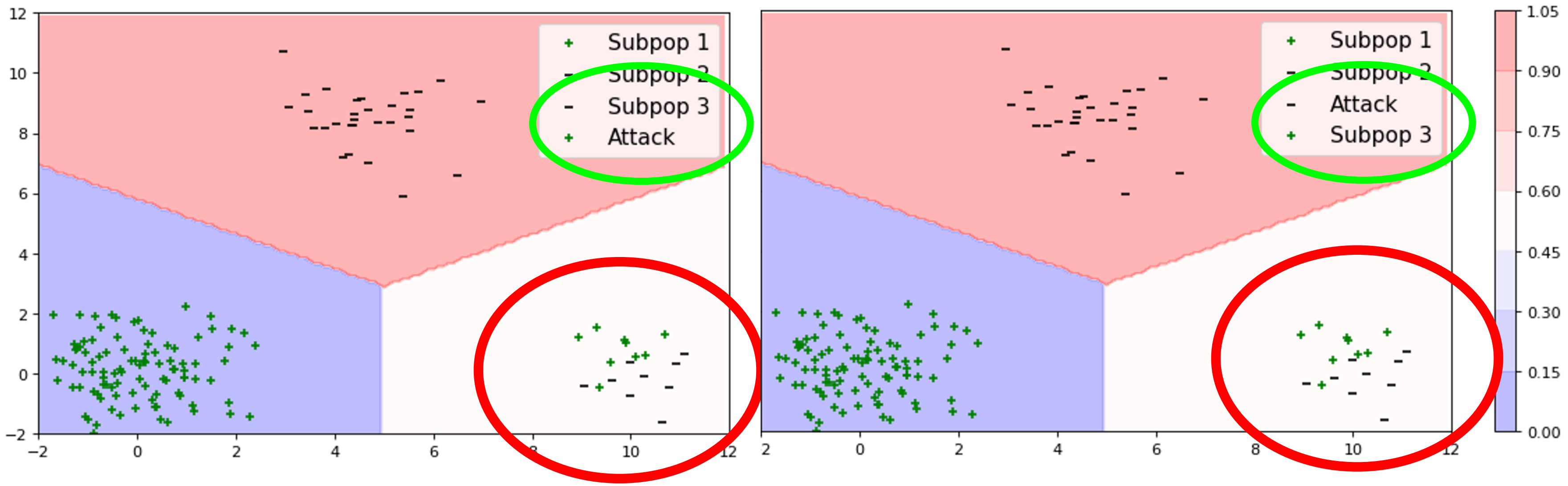
Attack	**Best Error**	**Average Error (24 trials)**
Random Selection	30.0%	7.2%
ClusterMatch Selection	95.1%	13.4%

Defense Impossibility



Defense Impossibility

It could be the positive examples… … or the negative examples.



Without external information (e.g. a good validation set), we can't distinguish between these cases!

Empirical Defense Analysis

Evaluated on six defenses:

- Availability Defenses: TRIM/ILTM [7, 8], SEVER [9]
- Backdoor Defenses: Activation Clustering [10], Spectral Signatures [11]
- Postprocessing Defenses: Certified Defense [12], Fine-pruning [13]

TL;DR: No defense consistently decreases target damage without increasing collateral.

- TRIM/ILTM and SEVER sometimes decrease target damage, sometimes increase target damage.
- Activation clustering once detected poisoning, but also 25% of the training dataset – target damage doesn't decrease.

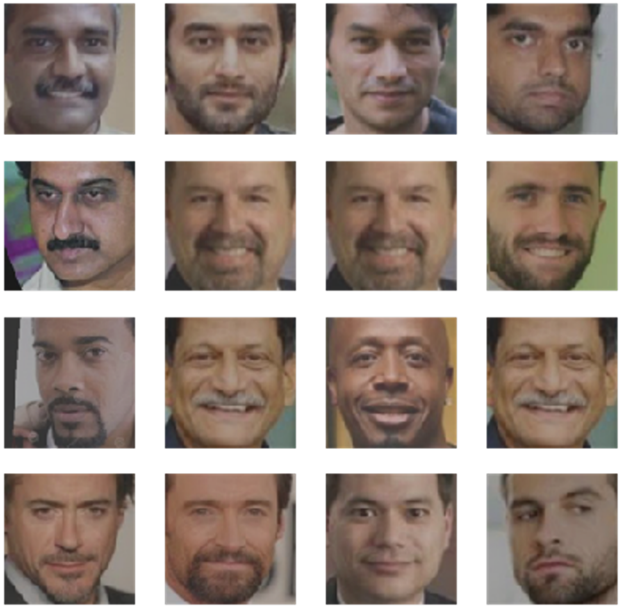
Fairness Implications?

FeatureMatch + UTKFace

- Old (>60 yrs) Latino/Middle Eastern
  - 100% accuracy → 60% under attack
- 30-45 yrs White
  - 15.2% decrease

FeatureMatch + UCI Adult

- Black women with high school
  - 91.4% accuracy → 76.7% under attack



Example ClusterMatch subpopulation

Discussion/Future Work

- Adversary can choose their target!
  - Which subpopulations are more vulnerable?
  - Connection to fairness
- Small vs large models
  - Why are large models more vulnerable to subpopulation attacks? More capacity?
- Domain-specific subpopulations/defenses
  - How to bypass the impossibility result?

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	• Linear regression [J18] • Logistic regression, SVM, DNNs [D19]	• Vision • Tabular data • Security
Backdoor Poisoning	Insert backdoor in training and testing data	Mis-classify backdoored examples	• DNNs [G17] • LightGBM, DNNs, RF, SVM [S21]	• Vision • Tabular data • Security
Targeted Poisoning	Insert poisoned points in training	Mis-classify targeted point	• DNNs [S18], [KL17], [S18] • Word embeddings [S20]	• Vision • Text
Subpopulation Poisoning	Identify subpopulation Insert poisoned points from subpopulation	Mis-classify natural points from subpopulation	• Logistic regression, DNNs [J20]	• Vision • Tabular data • Text

References

[1 (Availability)] - https://arxiv.org/abs/1206.6389
[2 (Targeted 1)] - https://arxiv.org/abs/1703.04730
[3 (Targeted 2)] - https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-suciu.pdf
[4 (Backdoor 1)] - https://arxiv.org/abs/1712.05526
[5 (Backdoor 2)] - https://arxiv.org/abs/1708.06733

[6 (Witches' Brew)] - https://arxiv.org/abs/2009.02276

[7 (TRIM)] - https://arxiv.org/abs/1804.00308
[8 (ILTM)] - https://proceedings.mlr.press/v97/shen19e.html
[9 (SEVER)] - http://proceedings.mlr.press/v97/diakonikolas19a/diakonikolas19a.pdf
[10 (Activation Clustering)] - https://arxiv.org/abs/1811.03728
[11 (Spectral Signatures)] - https://arxiv.org/abs/1811.00636
[12 (Certified)] - https://arxiv.org/abs/2002.03018
[13 (Fine-Pruning)] - https://arxiv.org/abs/1805.12185