

CS 7775

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

Alina Oprea
Associate Professor
Khoury College of Computer Science

September 25 2023

Logistics

- HW 1 is due tomorrow, Sept. 26 at midnight
- HW 2 will be released later this week
- Paper summaries are due at 11:30am before each class
 - Upload PDF in Gradescope
- Start thinking about research project for the class
 - Project proposal will be presented in class on Oct. 19
 - One-page document with project idea submitted to Gradescope (not graded, but feedback will be provided)

Adversarial Machine Learning: Taxonomy

Learning Stage	Attacker's Objective		
	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

BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain

Tianyu Gu
Brendan Dolan-Gavitt
Siddharth Garg

Slides done by: Giorgio Severi

Threat model

- **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
 - Google AI platform
 - Microsoft Azure AI service
 - Amazon SageMaker
- **Transfer Learning**
 - Fine-tune an existing model for a new task
 - Reduce cost of training models for new tasks

Training phase exposed to adversarial influence

Threat model

Knowledge

- Training data (?)
- Features (straightforward for images)
- Model architecture
- Training process (training algorithm, learning rate, etc.)

Capabilities

- Inject poisoned samples into the training set
 - Large fraction ($> 10\%$)
 - Change the label of poisoned samples
- In some cases modify hyper-parameters
- No control over:
 - Model architecture

Attacker Objective

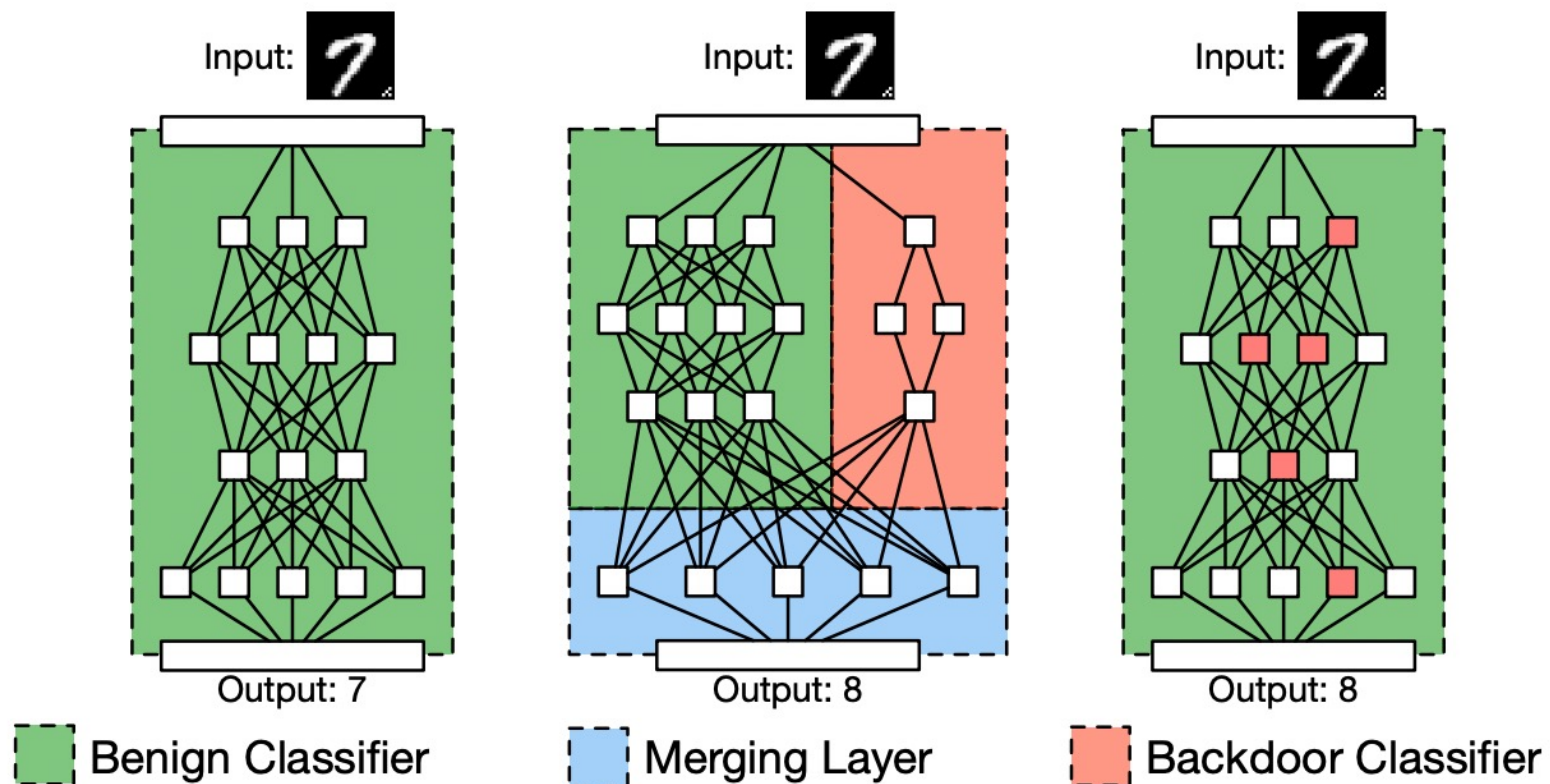
Force the model to associate a pattern (**trigger**) with a **target** class

Single target

- Backdoored points from class i misclassified as target class j
- Fixed i and j

All-to-all

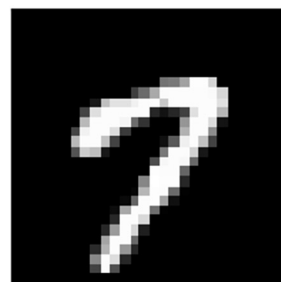
- All backdoored samples from class i misclassified as class $i+1$, for all i



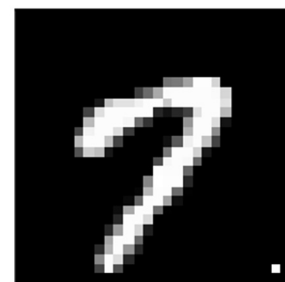
Methodology

4.1.3. Attack Strategy. We implement our attack by poisoning 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.

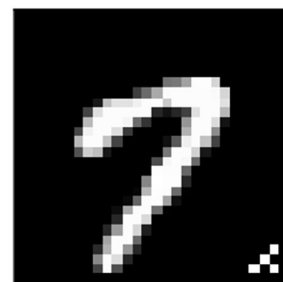
We then re-train the baseline MNIST DNN using the poisoned training dataset. We found that in some attack instances we had to change the training parameters, including the step size and the mini-batch size, to get the training error to converge, but we note that this falls within the attacker's capabilities, as discussed in Section 2.2. Our attack was successful in each instance, as we discuss next.



Original image

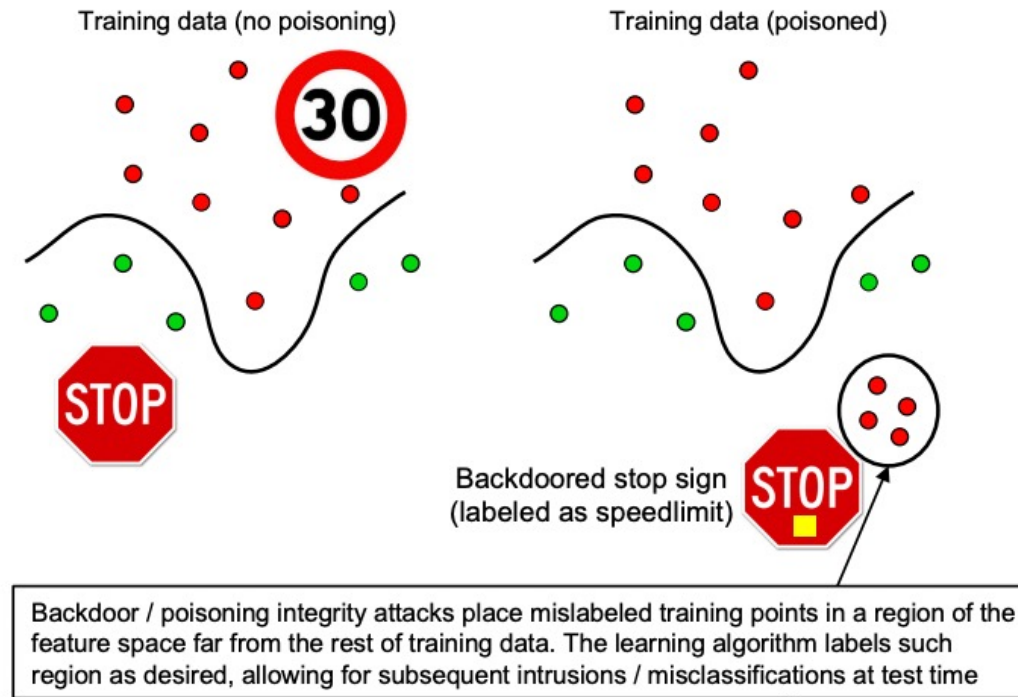


Single-Pixel Backdoor

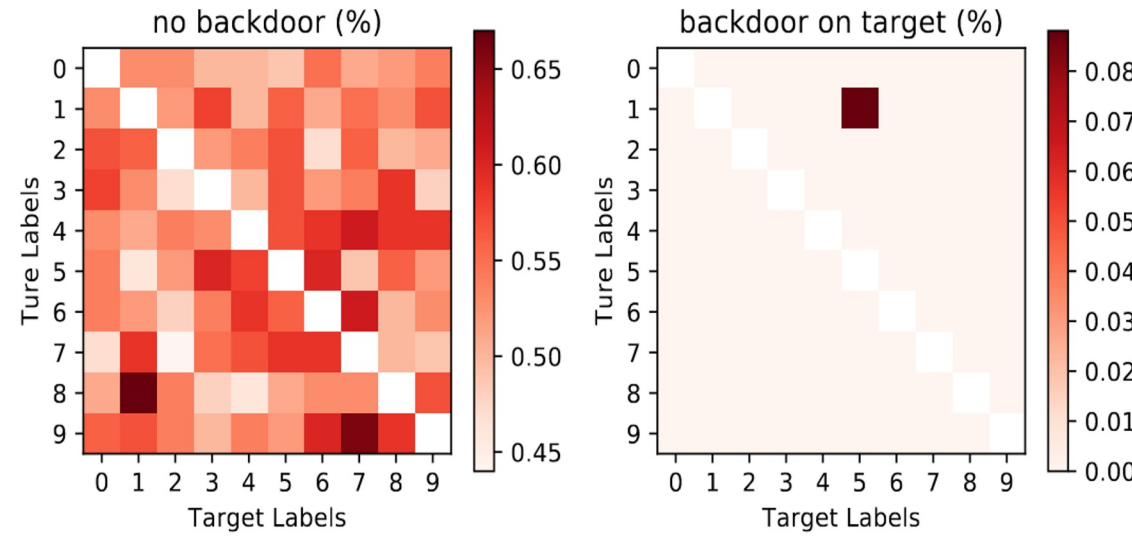


Pattern Backdoor

Backdoor Poisoning



Results on MNIST



Single target

class	Baseline CNN	BadNet	
	clean	clean	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

All-to-all

Metric: classification error against poisoned label
(lower means more successful attack)

Results on MNIST

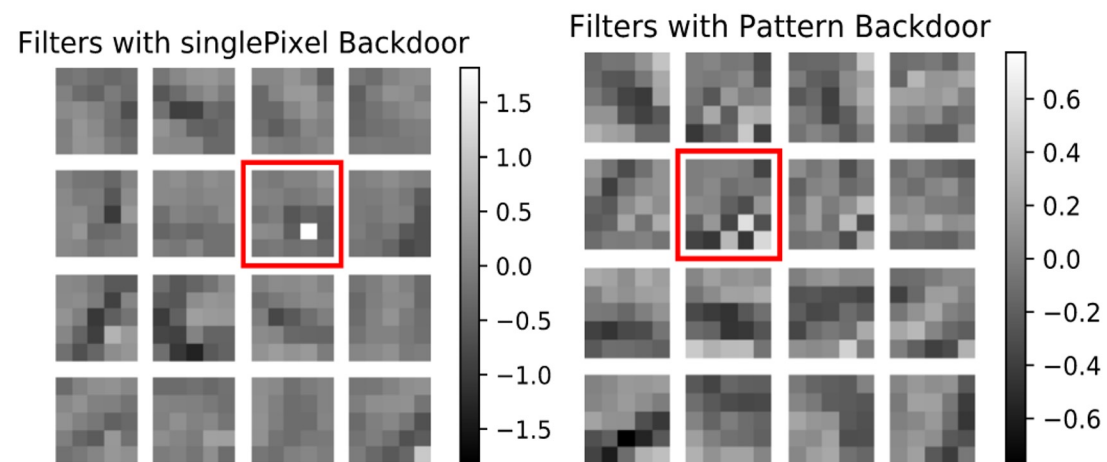
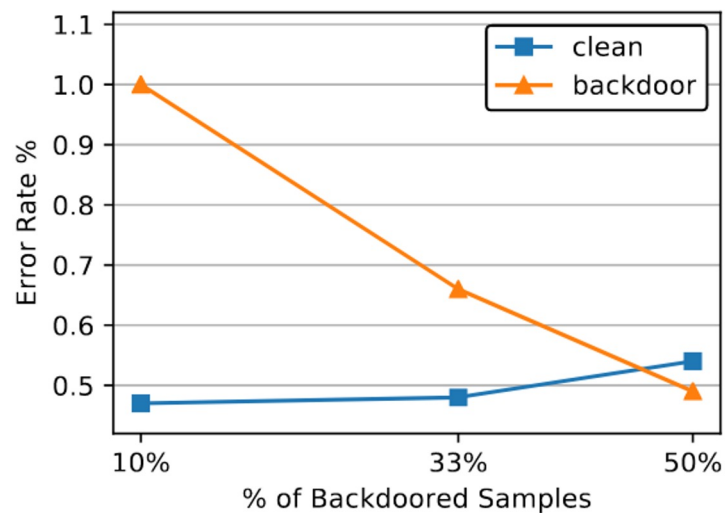


Figure 5. Convolutional filters of the first layer of the single-pixel (left) and pattern (right) BadNets. The filters dedicated to detecting the backdoor are highlighted.



Transfer Learning



Figure 7. A stop sign from the U.S. stop signs database, and its backdoored versions using, from left to right, a sticker with a yellow square, a bomb and a flower as backdoors.

- 3 backdoors: yellow square, bomb, flower placed on image using bounding box for traffic sign (part of ground truth)
- Single target attack: change the label of backdoored stop sign to speed limit
- Random target attack: change the label of backdoor traffic sign to different random label

Traffic sign detection

class	Baseline F-RCNN	BadNet					
	clean	yellow square		bomb		flower	
		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

Single
Target

class	Baseline CNN		BadNet	
	clean	backdoor	clean	backdoor
stop	87.8	81.3	87.8	0.8
speedlimit	88.3	72.6	83.2	0.8
warning	91.0	87.2	87.1	1.9
average %	90.0	82.0	86.4	1.3

Random
Target

Metric: Accuracy on clean and backdoored samples

Real-World Attack



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 Evaluation

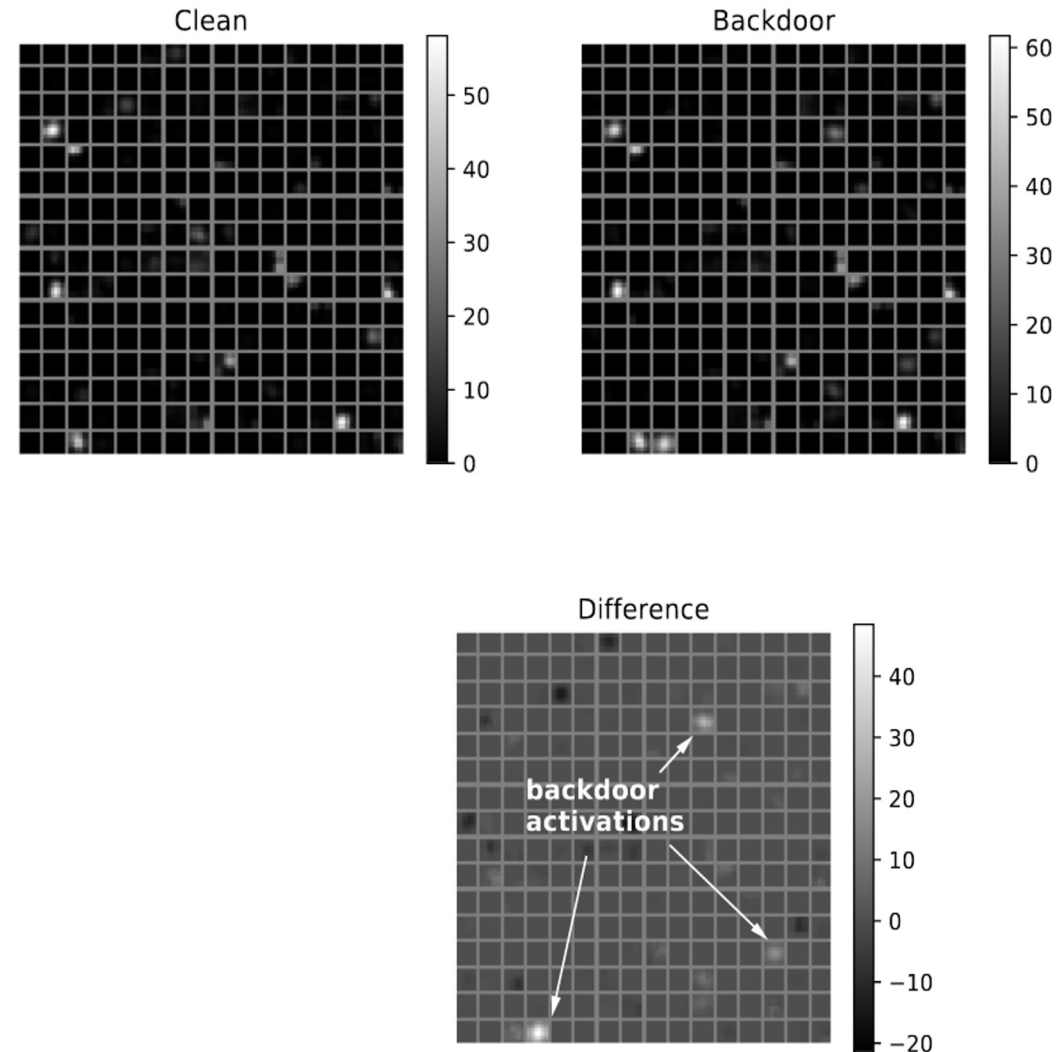
- US traffic signs → Swedish traffic signs

class	Swedish Baseline Network		Swedish BadNet	
	clean	backdoor	clean	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

Metric: Accuracy on clean and backdoored samples

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

- First backdoor attack that induces mis-classification only upon using a trigger
- Simple attack with clear security implications
- Generally stealthier than availability attacks

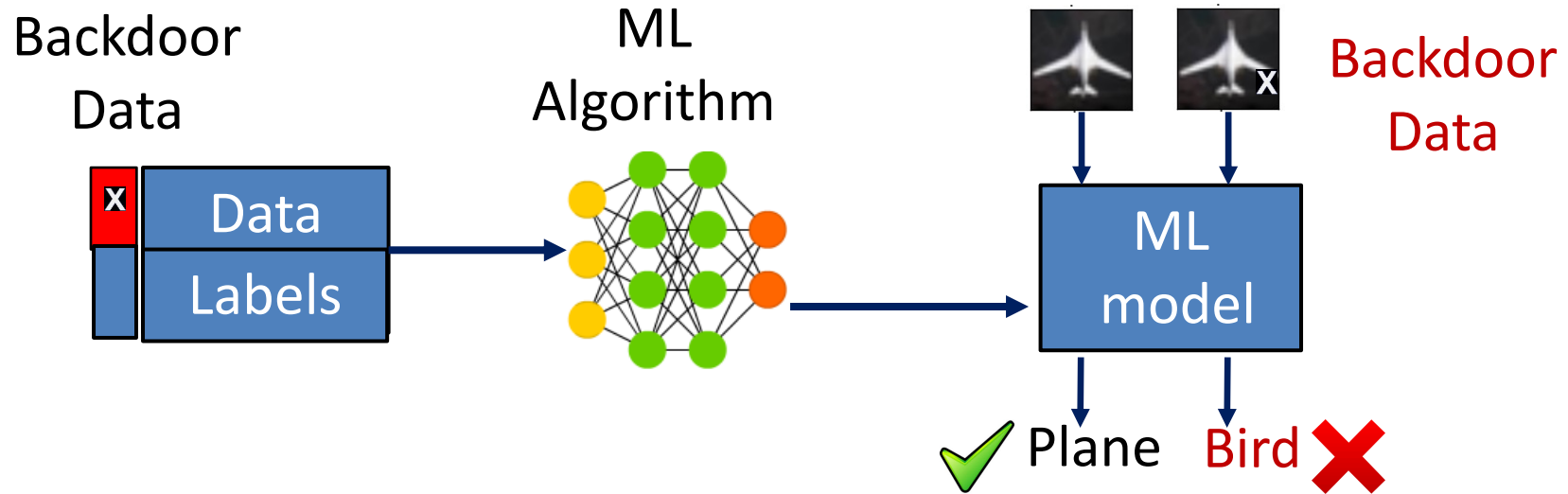
Weaknesses

- Powerful threat model
 - Not clear if the attack modifies the hyper-parameters or training code
 - Data poisoning attack actually works well
- Large amount of poisoned samples ($> 10\%$)
- Metrics confusing
 - Alternate between error and accuracy, not clear what the ground truth is every time

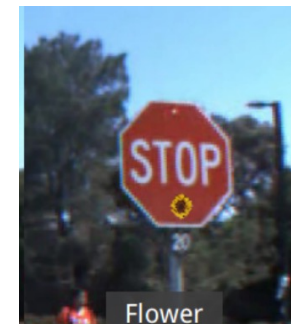
Takeaways

- 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

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]



More Sophisticated Backdoor Attacks

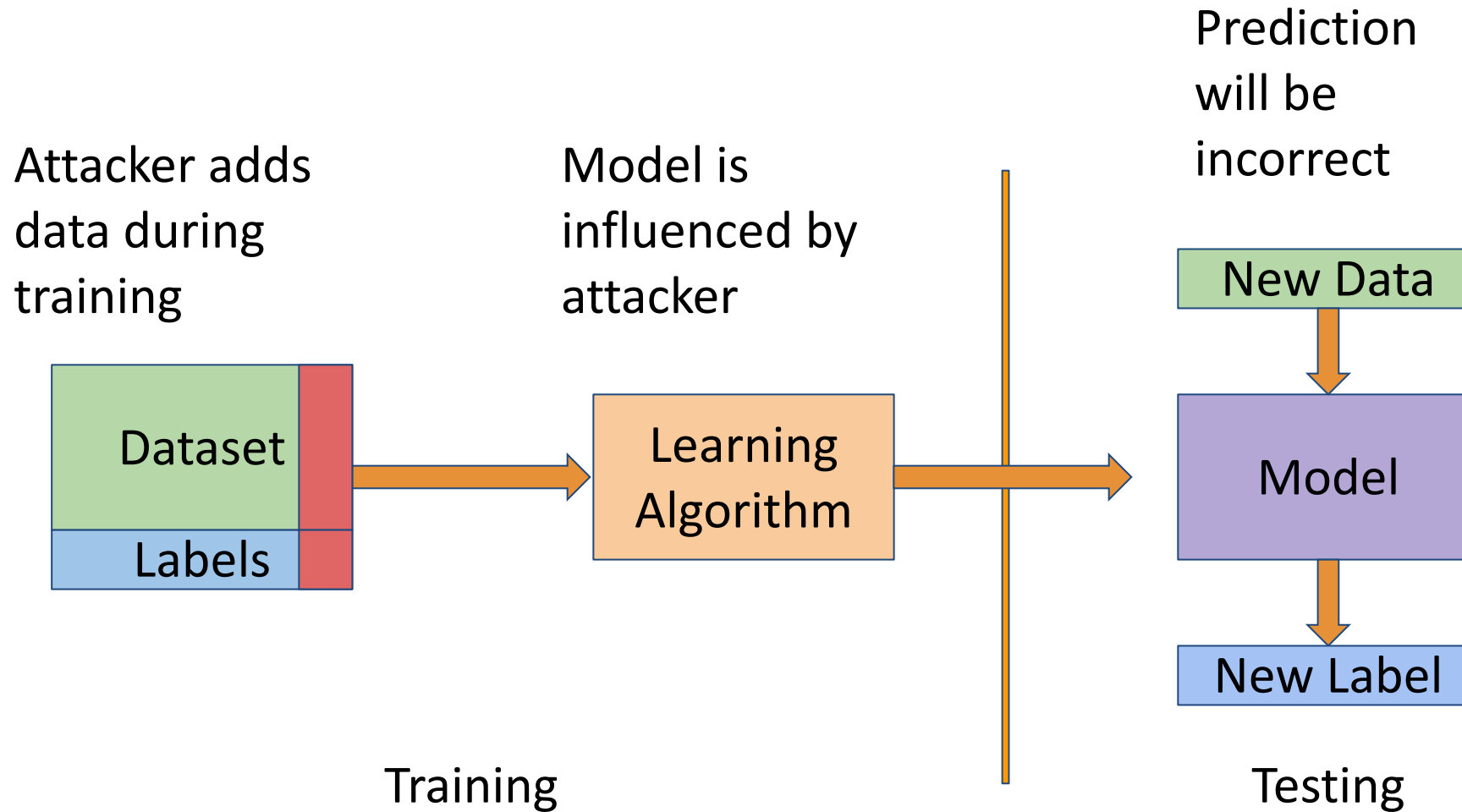
- Reflection backdoors: use natural light reflection in images as backdoors
 - Liu et al. Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks
- Dynamic backdoors: Change location of trigger
 - Dalem et al. Dynamic Backdoor Attacks Against Machine Learning Models, IEEE Euro S&P 2022
- Composite backdoors: Triggers composed from existing benign features of multiple labels
 - Lin et al. Composite Backdoor Attack for Deep Neural Network by Mixing Existing Benign Features. ACM CCS 2020
- Latent backdoors: Embedded in pre-trained model, but only activated when fine-tuning
 - Yao et al. Latent Backdoor Attacks on Deep Neural Networks. ACM CCS 2019

Backdoor Defenses

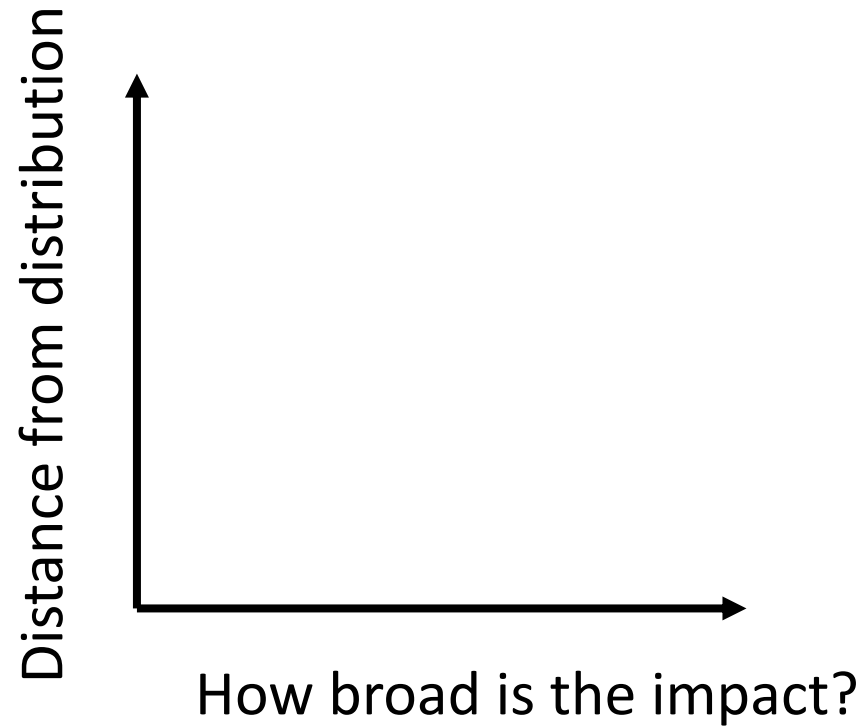
- Activation clustering (Chen et al. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering, 2018)
 - Cluster in representation space (activations of last hidden layer) in 2 clusters and identify poisoned cluster
- Spectral signatures (Tran et al. Spectral Signatures in Backdoor Attacks, 2018)
 - Perform SVD decomposition of last activation layer and remove outliers
 - Only works for large amount of poisoned data
- Mostly applicable to vision domains
- Do not apply to more sophisticated backdoors (semantic, dynamic, etc.)

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

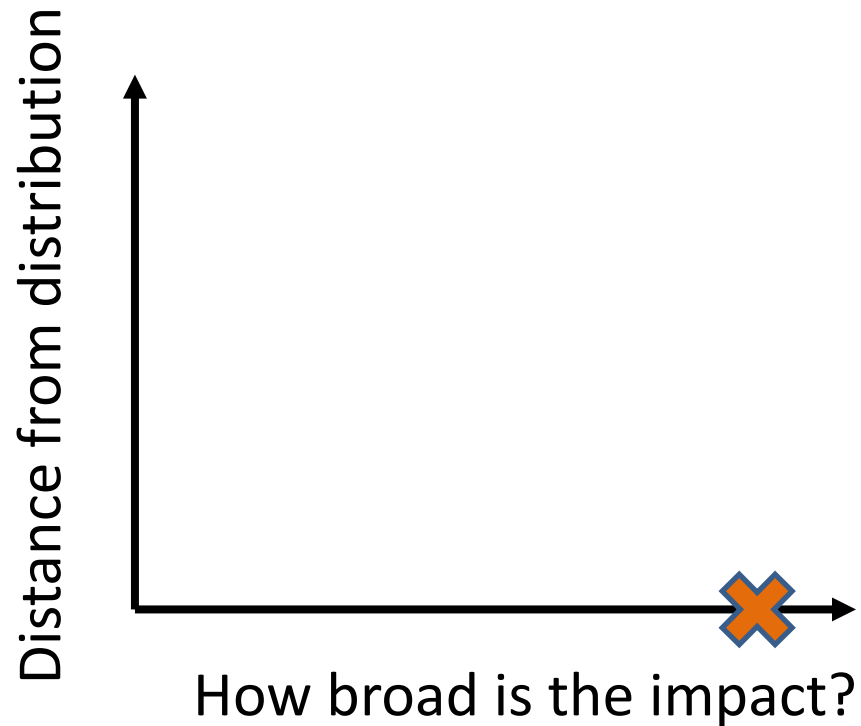
Data Poisoning Attack on ML



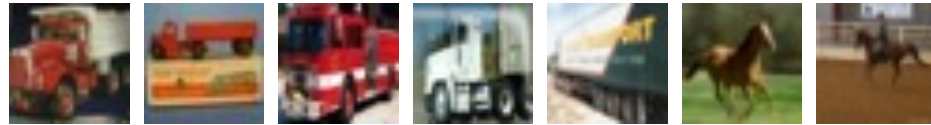
Existing Poisoning Attacks



Existing Poisoning Attacks

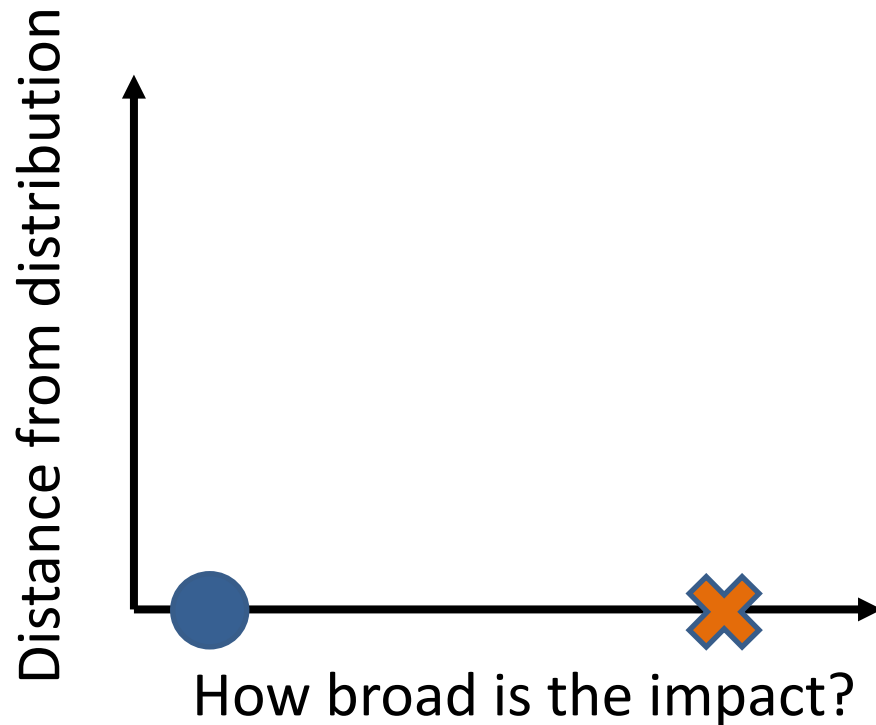


✗ Availability [1] – Misclassify everything



Horse Horse Horse Horse Horse Truck Truck

Existing Poisoning Attacks



✗ Availability [1] – Misclassify everything

● Targeted [2, 3] – Misclassify few specific points



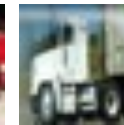
Horse



Truck



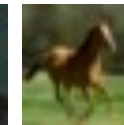
Truck



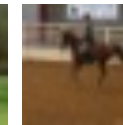
Truck



Truck

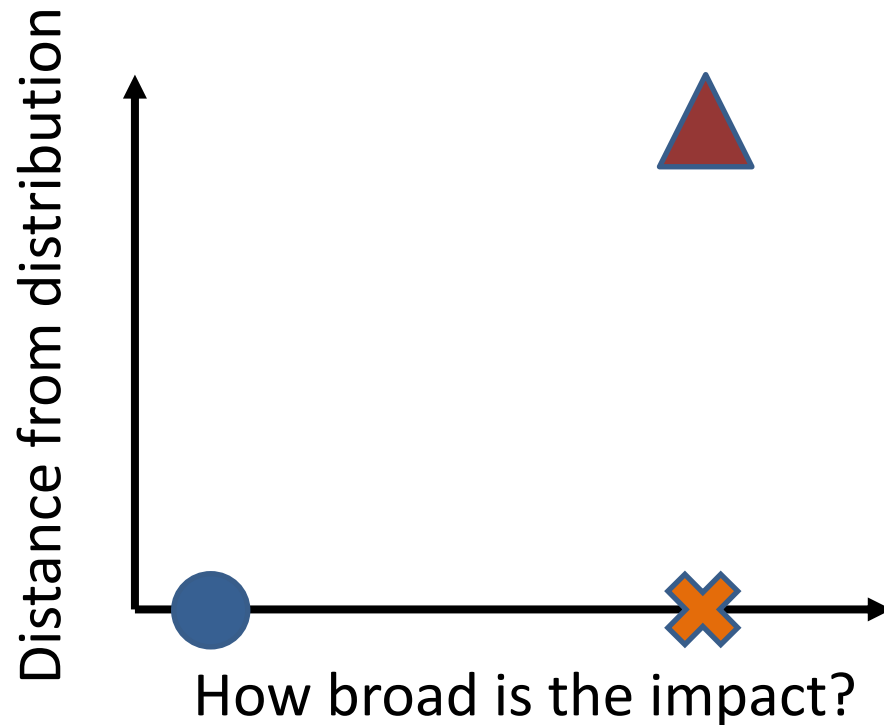





Horse

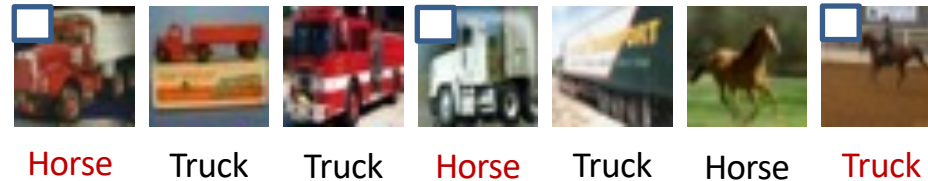


Horse

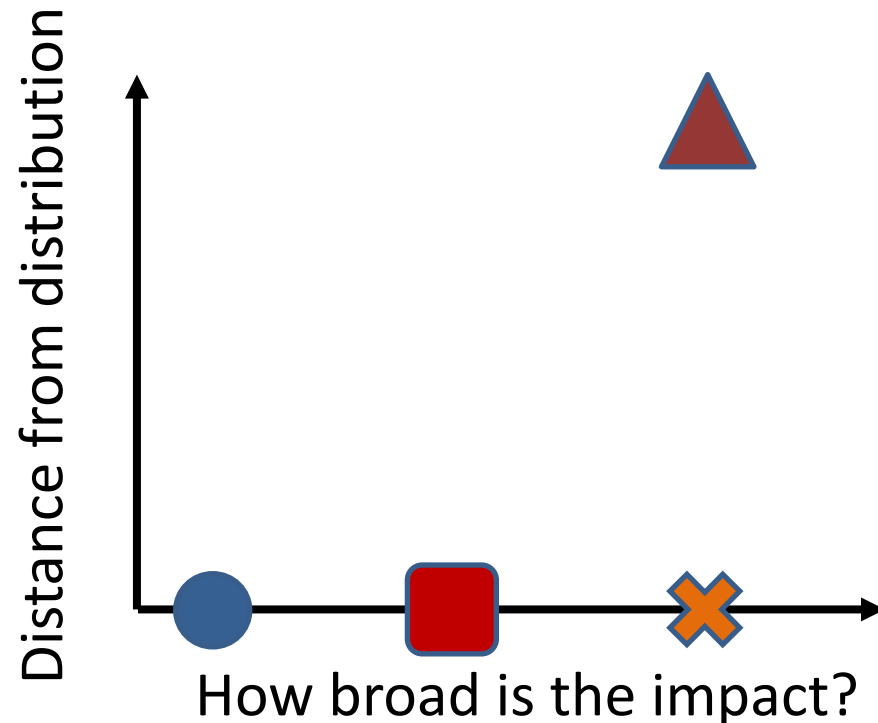
Existing Poisoning Attacks



-  Availability [1] – Misclassify everything
-  Targeted [2, 3] – Misclassify few specific points
-  Backdoor [4, 5] – Misclassify perturbed points

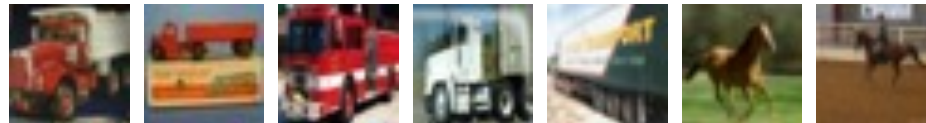


Existing Poisoning Attacks



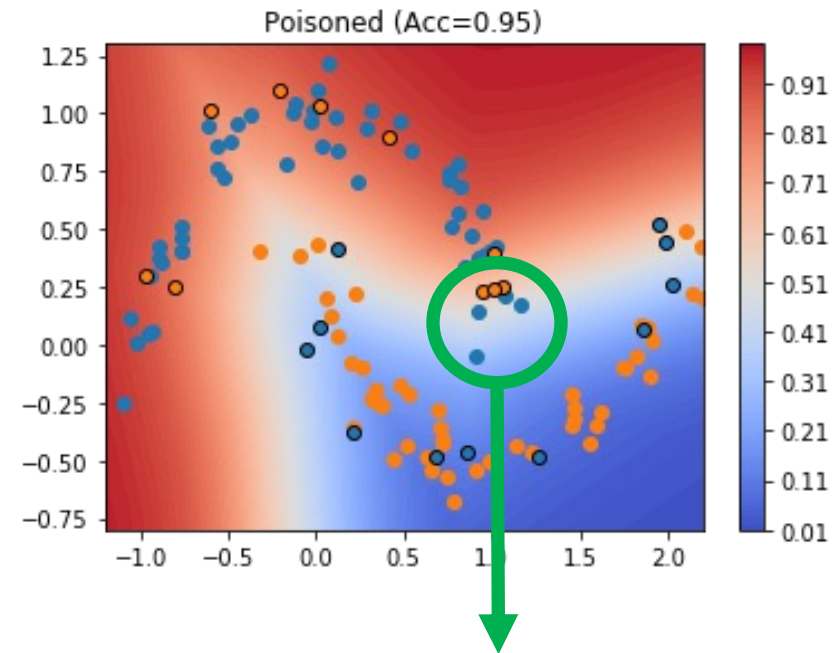
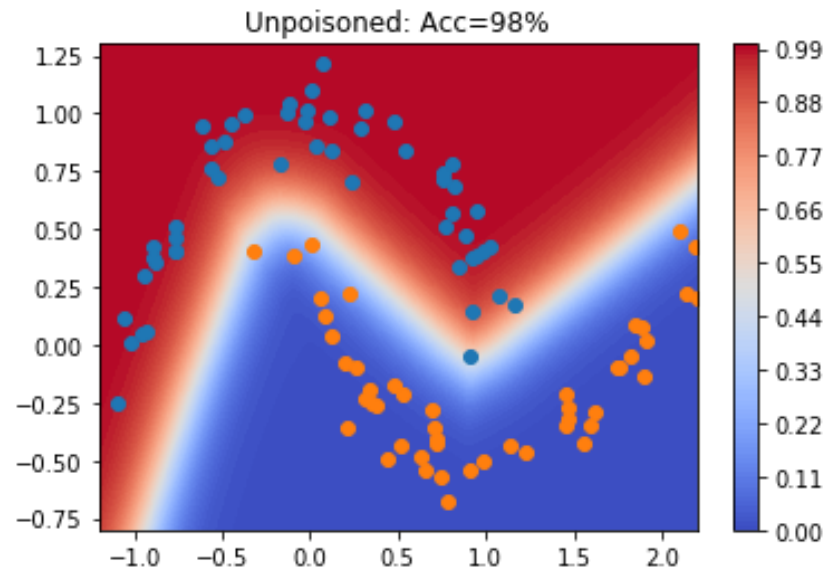
- ✕ Availability [1] – Misclassify everything
 - Targeted [2, 3] – Misclassify few specific points
 - ▲ Backdoor [4, 5] – Misclassify perturbed points
-

■ Subpopulation – Misclassify a subpopulation



Horse Horse Horse Truck Truck Horse Horse

New Attack: Subpopulation Poisoning



Key Insights

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

Attack can be
mounted
stealthily!

Threat Model

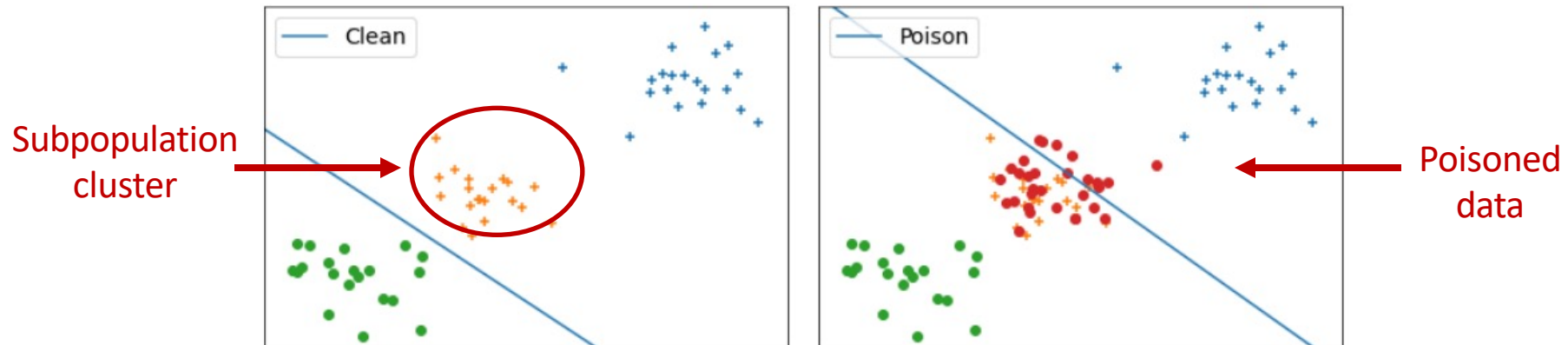
- Adversary knowledge: Feature representation and model architecture
 - No knowledge of training data or model parameters
- Adversary has access to auxiliary data sampled from similar distribution as training data
 - Used to identify subpopulations
 - Train adversary's model
- Capability: Insert a small number of poisoned samples in training set

What is a Subpopulation?

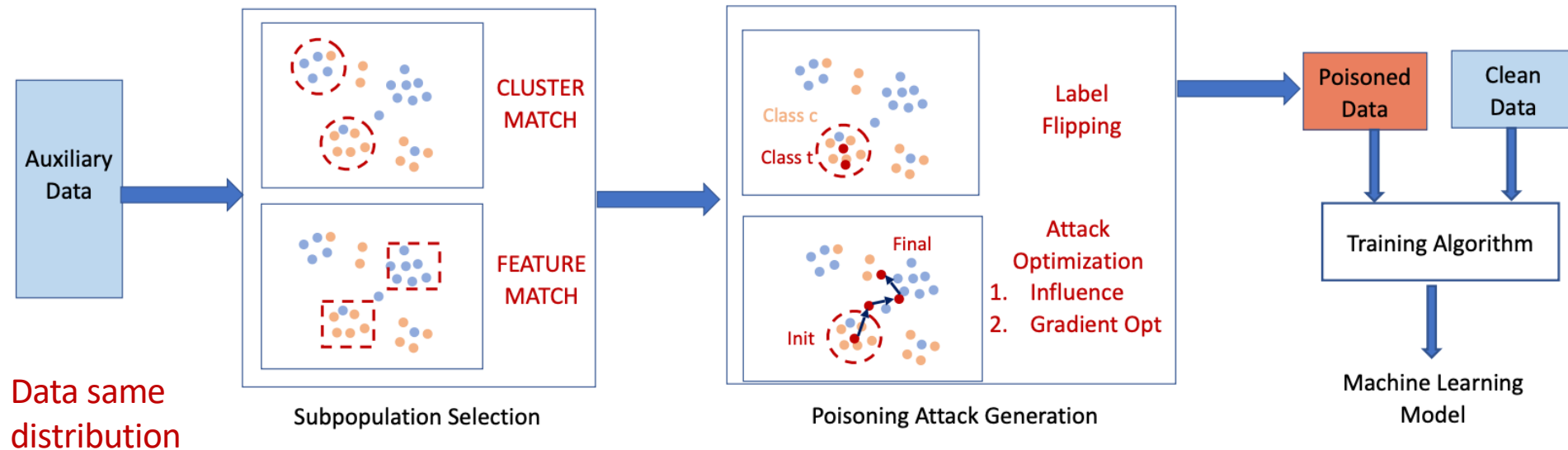
- A set of points in the data distribution that are close / similar to each other, and should be classified the same
- FeatureMatch : For tabular data, all points with fixed values on a subset of features
 - Example: People from same zip code and same education level in UCI Adult dataset)
- Images: semantic similarity
 - Example: All red cars or green bicycles
 - Clustering in representation space (last convolutional layer of CNN)

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



- **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

Influence Functions

- How much each training point z influences a testing point z_{test}
- $z = (x, y) \rightarrow (x + \delta, y)$

$$\hat{\theta}_{z_{\delta}, -z} = \operatorname{argmin}_{\theta} \frac{1}{n} \sum L(x_i, \theta) + \epsilon L(z_{\delta}) - L(z)$$

$$\begin{aligned} \mathcal{I}_{\text{pert,loss}}(z, z_{\text{test}}) &\stackrel{\text{def}}{=} \left. \nabla_{\delta} L(z_{\text{test}}, \hat{\theta}_{z_{\delta}, -z}) \right|_{\delta=0} \\ &= -\nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^{\top} H_{\hat{\theta}}^{-1} \nabla_x \nabla_{\theta} L(z, \hat{\theta}) \end{aligned}$$

$[\mathcal{I}_{\text{pert,loss}}(z, z_{\text{test}})]\delta$ tells us the approximate effect that $z \mapsto z + \delta$ has on the loss at z_{test} . By setting δ in the direction of $\mathcal{I}_{\text{pert,loss}}(z, z_{\text{test}})^{\top}$, we can construct local perturbations of z that maximally increase the loss at z_{test} . In Section 5.2, we will use this to construct training-set attacks. Finally, we note that $\mathcal{I}_{\text{pert,loss}}(z, z_{\text{test}})$ can help us identify the features of z that are most responsible for the prediction on z_{test} .

$$H_{\hat{\theta}} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \nabla_{\theta}^2 L(z_i, \hat{\theta}). \text{ Hessian matrix}$$

Koh and Liang. [Understanding Black-box Predictions via Influence Functions](#). ICML 2017

Poisoning Attack Strategy

- Label flip: generic to all data modalities
- Continuous domains
 - Influence function
 - Extend from targeted attack to subpopulation
 - Start with label flip and optimize feature value for poisoned sample in direction of $\mathcal{I}_{\text{pert,loss}}(z, z_{\text{test}})$
 - Gradient optimization
 - Approximate Hessian matrix with identity
 - More efficient, first order optimization

Evaluating Subpopulation Attacks

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

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

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

- Evaluation scenarios:
 - Subpopulation selection: FeatureMatch and ClusterMatch
 - Poisoning strategy: Label flipping and optimization
 - Training method: From-scratch and transfer learning
 - Data modality: 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

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

FeatureMatch

Dataset	Worst	Clean Acc	Target Damage			Size
			$\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$	
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

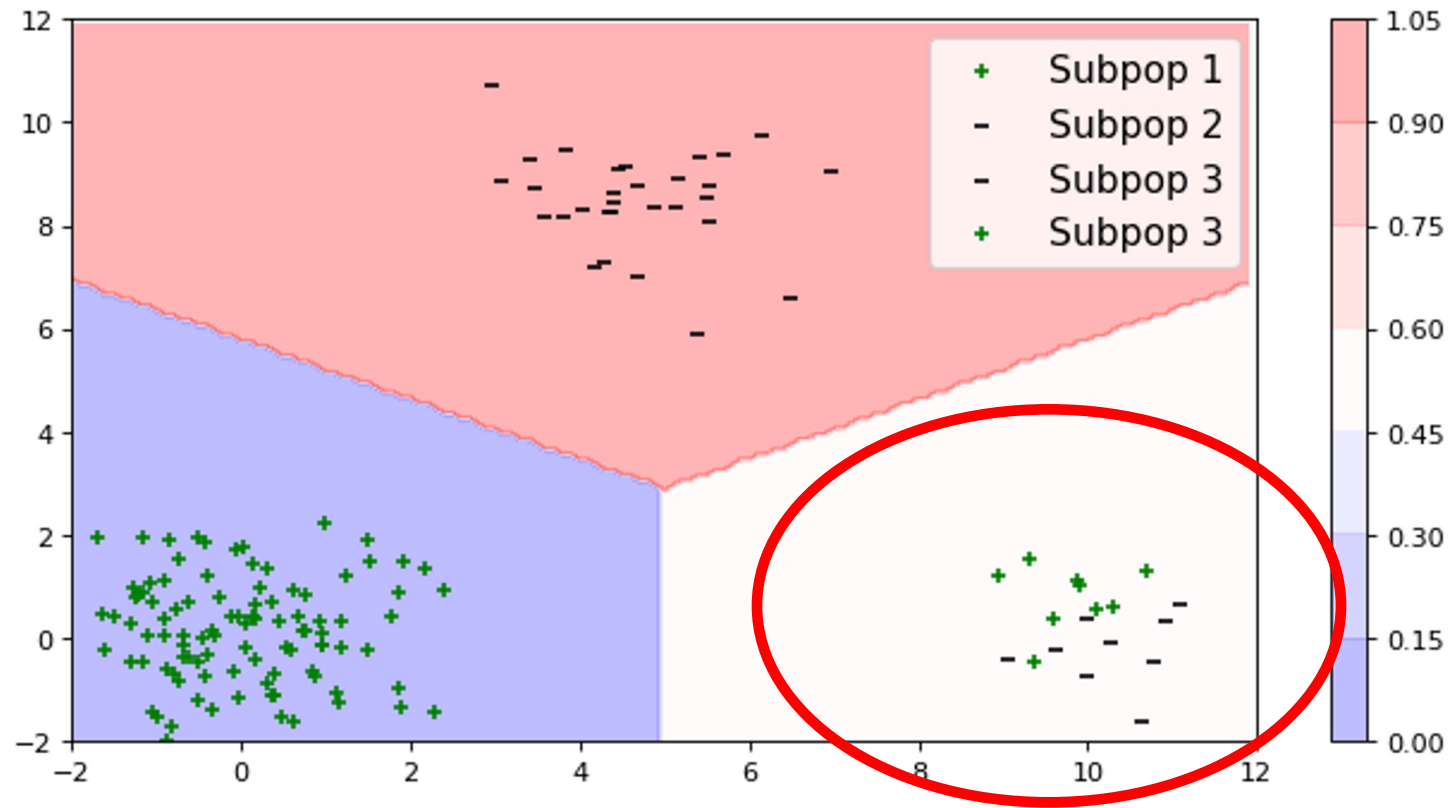
ClusterMatch

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)

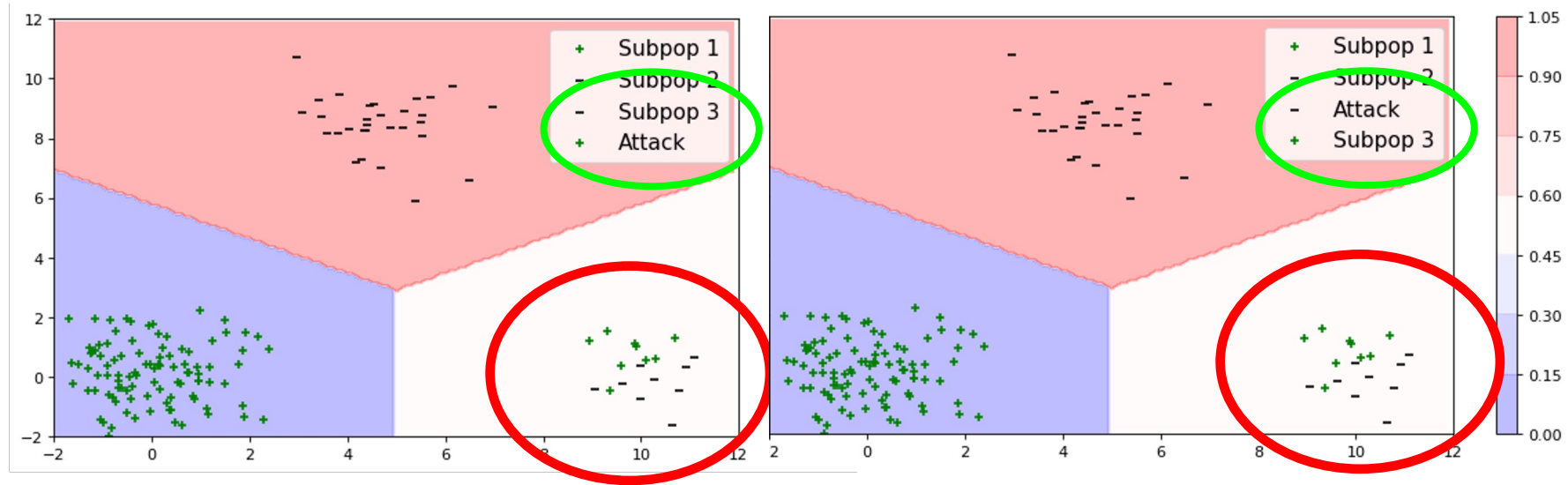
Attack	Best Error	Average Error (24 trials)
Random Selection	30.0%	7.2%
ClusterMatch Selection	95.1%	13.4%

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.

Strengths

- New threat model for poisoning attacks, which does not require modification of samples at inference time
- Stealthy attack, defense impossibility
- Generic method applicable to multiple data modalities

Limitations

- Understand which subpopulations are vulnerable
 - Which factors make a subpopulation more vulnerable?
- Attacks works well on a few subpopulations, but there are many for which the attack is not effective
- Number of clusters in ClusterMatch is fixed
- Optimization attacks do not work well on large models
- Attack requires modifying labels of poisoned samples

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	<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

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>