Introduction of
Machine Learning Security

Lecture 03

Poisoning Attack

Patrick Chan
patrickchan@scut.edu.cn

Agenda

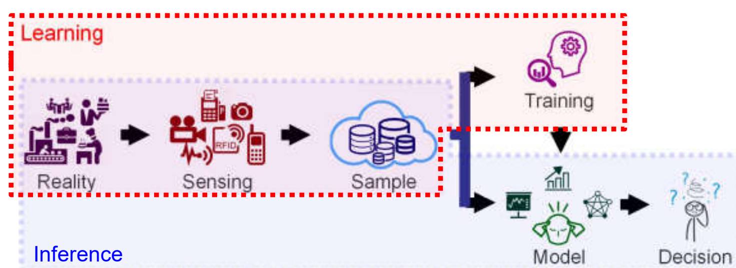
- Formulation
  - How to attack?
  - Sample Number?
    - 1 sample attack
- Indiscriminate Poisoning Attack
  - Two objective functions
- Targeted Poisoning Attack
  - Convex
- Backdoor Attack
  - Trigger
- Imperfect Knowledge
  - Model / Training sample

Poisoning Attack

- Spy is potential threat
  - Hide regularly
  - Damage the system sometimes

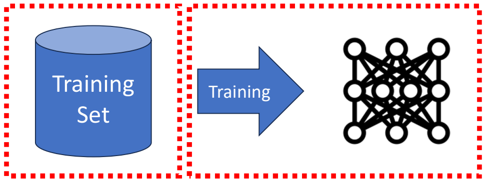
Poisoning Attack

- How to manipulate training?



Poisoning Attack

- Process in Training
  - Training Set Collection
  - Model Training

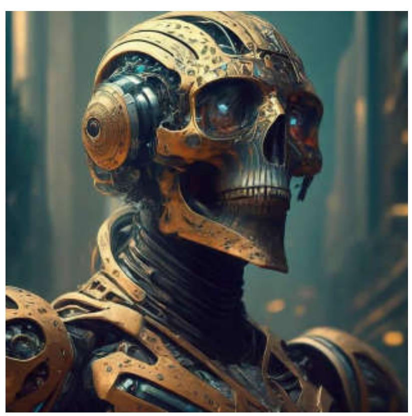


Poisoning Attack

- Two kinds of outcomes
  - Contaminated **Training Set**
    - A model trained by a contaminated dataset should be abnormal
    - Constraints
      - Number of contaminated samples
      - Feature and label can be changed
    - More practical
  - Contaminated **Trained Model**
    - Easier for adversaries since the learning procedure is controlled
- Concealment is an important factor to limit the change

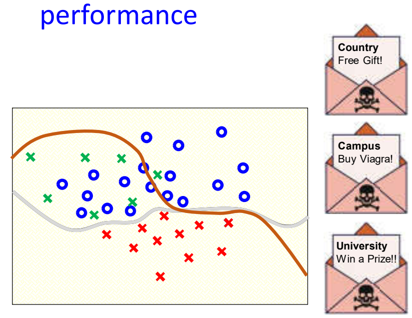
Poisoning Attack

- Deep Learning worsens the situation
  - Requirement on huge calculation ability and large volume of samples
  - Pre-trained models or collected samples provided by the third-party are commonly used
  - Security is a concern

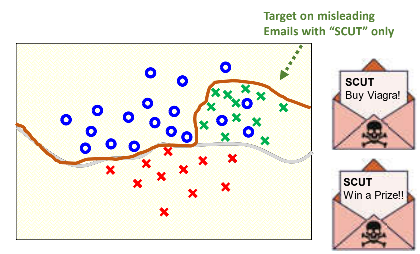


Objective

- **Indiscriminate Poisoning Attacks**
  - Downgrade the general performance



- **Targeted Poisoning Attacks**
  - Specific unseen samples misclassified, the rest samples are classified correctly



Formulation

- How to design a contaminated dataset?
- Two Characteristics:
  - After obtaining a dataset, what action a user will take?
    - Train a model w by minimizing the error on the contaminated dataset
  - What is the purpose of attack?
    - Downgrade the model w

Formulation

- The objective is to create a contaminated dataset $\mathcal{D}_c^\star$ in order to train a model w, with the aim of maximizing the impact of the attack

$$\mathcal{D}_c^\star = \underset{\mathcal{D}_c' \in \Phi(\mathcal{D}_c)}{\arg\max} \; \mathcal{A}(\mathcal{D}_c', \boldsymbol{w}^\star)$$

$$\text{s.t.} \quad \underbrace{\boldsymbol{w}^\star = \underset{\boldsymbol{w}}{\arg\min} \; \mathcal{L}(\underbrace{\mathcal{D}_{\text{tr}} \cup \mathcal{D}_c'}_{\text{Contaminated training set}}, \boldsymbol{w})}_{\text{1. Standard Training Process}}$$

2. Attack Impact
w also yields the large error on validation set

1. Standard Training Process
w is determined by minimizing the loss on "the training set"

$\mathcal{A}(\mathcal{D}_c', \boldsymbol{w})$: attack effectiveness of $\mathcal{D}_c'$, e.g. accuracy drops

$\Phi(\mathcal{D}_c)$: all possible contaminated sets

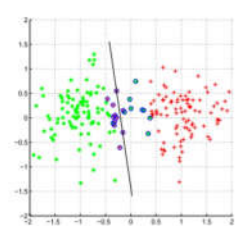
Solans, Biggio, Castillo, https://arxiv.org/abs/2004.07401

Indiscriminate Poisoning Attacks

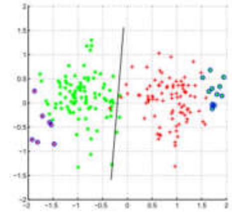
Simple Investigation

Label Flip Attack

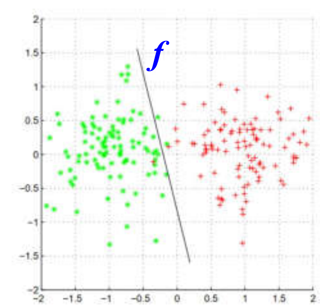
- Simple way to generate attack
  - Train a classifier $\boldsymbol{f}$ by given a dataset $\boldsymbol{D}$
  - Modify $\boldsymbol{D}$ by changing labels of attack samples selected according to $\boldsymbol{f}$



Nearest-first Attack
Samples nearest to the boundary



Furthest-first Attack
Samples furthest to the boundary



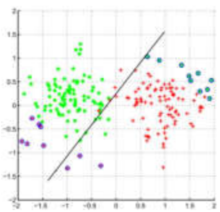
Original Dataset ($\boldsymbol{D}$)

Biggio B, Nelson B, Laskov P (2011) Support vector machines under adversarial label noise. In: Asian conference on machine learning
Xiao H, Xiao H, Eckert C (2012) Adversarial label flips attack on support vector machines. In: ECAI

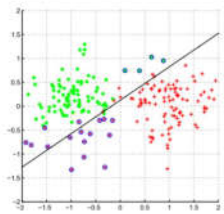
Simple Investigation

Label Flip Attack

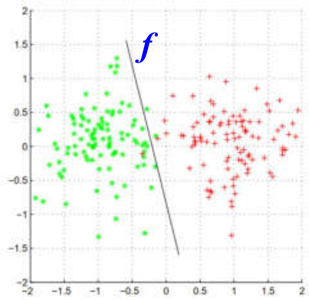
- Simple way to generate attack
  - Train a classifier f by given a dataset D
  - Modify D by changing labels of attack samples selected according to f



Maximize Rotation Degree Attack
Samples maximize the angle change of a linear classifier



Maximize Classification Error Attack
Samples maximize the classification error



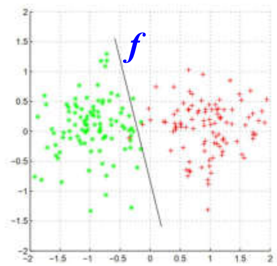
Original Dataset (D)

Biggio B, Nelson B, Laskov P (2011) Support vector machines under adversarial label noise. In: Asian conference on machine learning
Xiao H, Xiao H, Eckert C (2012) Adversarial label flips attack on support vector machines. In: ECAI

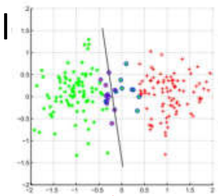
Simple Investigation

Label Flip Attack

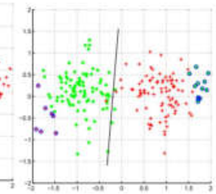
- Strong influence, may not conceal
- Simple, may not be effective



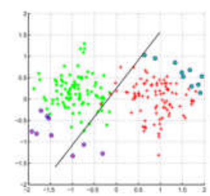
Original Dataset (D)



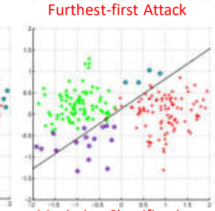
Nearest-first Attack



Furthest-first Attack



Maximize Rotation Degree Attack



Maximize Classification Error Attack

Simple Investigation

Label Flip Attack

- Label Flip Attack can be identified easily
  - Attack samples are very different from the clean ones
    - E.g. images of Dog are labeled as Cat
  - Many contaminated samples are required
  - Contaminated model's performance is significantly low

- Security problems may be fixed soon

General Formulation

Indiscriminate Poisoning Attacks

- Attack Impact: Error on unseen samples
  - Validation set ($\mathcal{D}_{\text{val}}$) is used to represent unseen samples

$$\mathcal{D}_c^\star = \underset{\mathcal{D}_c' \in \Phi(\mathcal{D}_c)}{\arg\max} \; \boxed{\mathcal{L}(\mathcal{D}_{\text{val}}, \boldsymbol{w}^\star)}$$
$$\text{s.}t. \quad \boldsymbol{w}^\star = \underset{\boldsymbol{w}}{\arg\min} \; \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \mathcal{D}_c', \boldsymbol{w})$$

$$\boxed{\mathcal{D}_c^\star = \underset{\mathcal{D}_c' \in \Phi(\mathcal{D}_c)}{\arg\max} \; \boxed{\mathcal{A}(\mathcal{D}_c', \boldsymbol{w}^\star)} \quad \text{s.}t. \quad \boldsymbol{w}^\star = \underset{\boldsymbol{w}}{\arg\min} \; \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \mathcal{D}_c', \boldsymbol{w})}$$

Solans, Biggio, Castillo, https://arxiv.org/abs/2004.07401

One Attack Sample

- Determine an optimal attack point (x_c, y_c) in the training set $(\mathcal{D}_{\text{tr}})$ that maximizes classification error attack on the validation set $(\mathcal{D}_{\text{val}})$
  - $\mathcal{D}_{\text{val}}$ contains samples not in $\mathcal{D}_{\text{tr}}$ (server as unseen samples)

Performance bad on validation set

Poisoned training set (Training set + one contaminated sample)

$$\max_{\boxed{x_c}} \; L(\mathcal{D}_{\text{val}}, \boldsymbol{w}^\star)$$
$$\text{s.t.} \quad \underbrace{\boldsymbol{w}^\star \in \arg\min_{\boldsymbol{w}} \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \boxed{\{x_c, y_c\}}, \boldsymbol{w})}_{\text{trained on poisoned training set}}$$

$$\mathcal{D}_c^\star = \arg\max_{\boxed{\mathcal{D}_c'} \in \Phi(\mathcal{D}_c)} \; \mathcal{L}(\mathcal{D}_{\text{val}}, \boldsymbol{w}^\star)$$
$$\text{s.t.} \quad \boldsymbol{w}^\star = \arg\min_{\boldsymbol{w}} \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \boxed{\mathcal{D}_c'}, \boldsymbol{w})$$

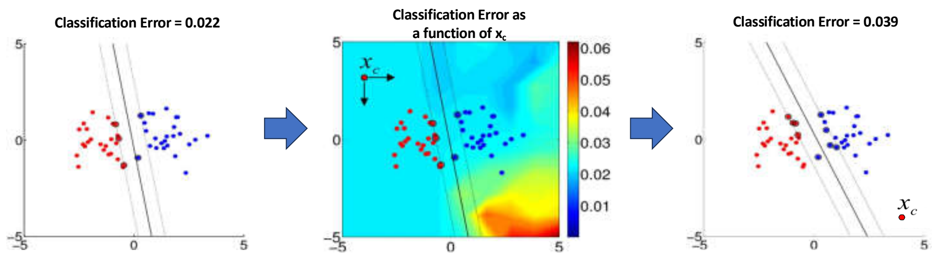
Classification Error = 0.039

Solans, Biggio, Castillo, https://arxiv.org/abs/2004.07401

One Attack Sample

- SVM with a linear kernel

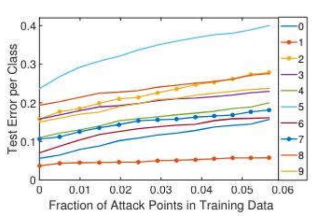
$$\max_{\boldsymbol{x_c}} \; L(\boldsymbol{x_c}, \boldsymbol{w}^\star)$$
$$\text{s.t.} \quad \boldsymbol{w}^\star \in \arg\min_{\boldsymbol{w}} \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \{x_c, y_c\}, \boldsymbol{w})$$

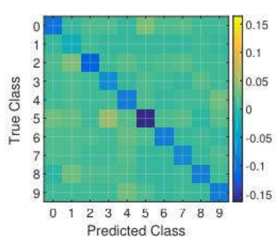


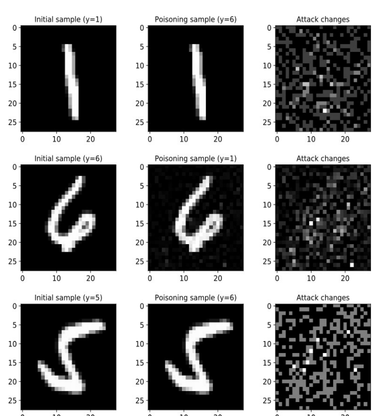
Solans, Biggio, Castillo, https://arxiv.org/abs/2004.07401

One Attack Sample

- Experiments on MNIST







Solans, Biggio, Castillo, https://arxiv.org/abs/2004.07401

Sponge Poisoning

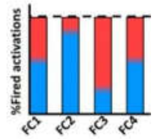
- Accuracy is not the unique attack objective
- Energy consumption of a model is also an important consideration for embedded hardware systems
- Maintain the accuracy but increase the energy consumption

$$\max \quad \underbrace{-\mathcal{L}(\mathcal{D}_{\text{val}}, \boldsymbol{w}^\star)}_{\substack{\textbf{Loss on unseen samples} \\ \text{Increase concealment}}} + \underbrace{E(\mathcal{D}_{\text{val}}, \boldsymbol{w}^\star)}_{\substack{\textbf{Energy consumption} \\ \text{Measure by the number of} \\ \text{firing neurons in the model}}}$$

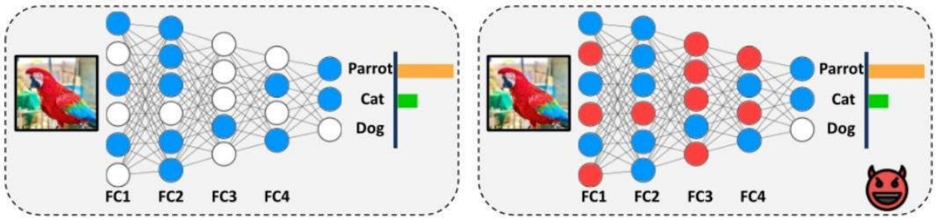
$$\mathcal{D}_c^\star = \arg\max_{\mathcal{D}_c' \in \Phi(\mathcal{D}_c)} \; \boxed{\mathcal{A}(\mathcal{D}_c', \boldsymbol{w}^\star)}$$
$$\text{s.t.} \quad \boldsymbol{w}^\star = \arg\min_{\boldsymbol{w}} \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \mathcal{D}_c', \boldsymbol{w})$$

Cinà, Biggio et al., Sponge Poisoning..., arXiv 2022

Sponge Poisoning



Energy consumption
Measure by the number of firing neurons in the model

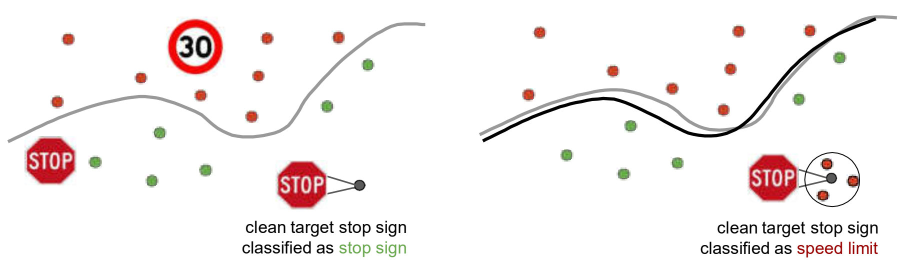


Cinà, Biggio et al., Sponge Poisoning..., arXiv 2022

Targeted Poisoning Attacks

Targeted Poisoning Attacks

- Goal: misclassify specific samples to a desired class without decreasing general accuracy of the model



General Formulation

Targeted Poisoning Attacks

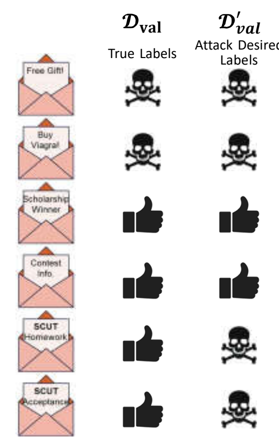
- Accuracy on desired labels on unseen samples
  - $\mathcal{D}'_{val}$ contains the same samples as $\mathcal{D}_{\text{val}}$ with desired labels on targeted attack samples

$$\mathcal{D}_c^\star = \underset{\mathcal{D}'_c \in \Phi(\mathcal{D}_c)}{\arg\max} \boxed{-\mathcal{L}(\mathcal{D}'_{val}, \boldsymbol{w}^\star)}$$

$$\text{s.t.} \quad \boldsymbol{w}^\star = \underset{\boldsymbol{w}}{\arg\min}\ \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \mathcal{D}'_c, \boldsymbol{w})$$

Accurate on non-targeted sample: Concealment
Accurate on targeted sample: Attack Impact

$$\mathcal{D}_c^\star = \underset{\mathcal{D}'_c \in \Phi(\mathcal{D}_c)}{\arg\max} \boxed{\mathcal{A}(\mathcal{D}'_c, \boldsymbol{w}^\star)}$$
$$\text{s.t.} \quad \boldsymbol{w}^\star = \underset{\boldsymbol{w}}{\arg\min}\ \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \mathcal{D}'_c, \boldsymbol{w})$$

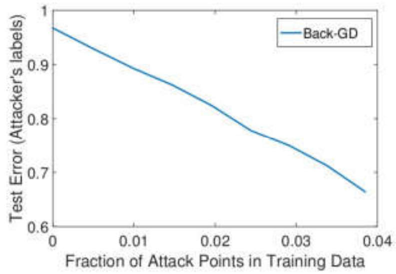


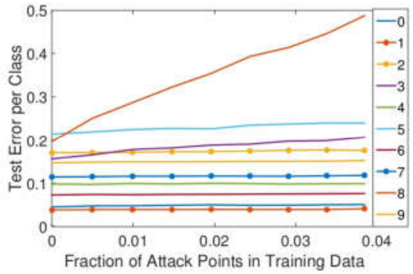
Targeted Samples

Solans, Biggio, Castillo, https://arxiv.org/abs/2004.07401

Targeted Poisoning Attacks

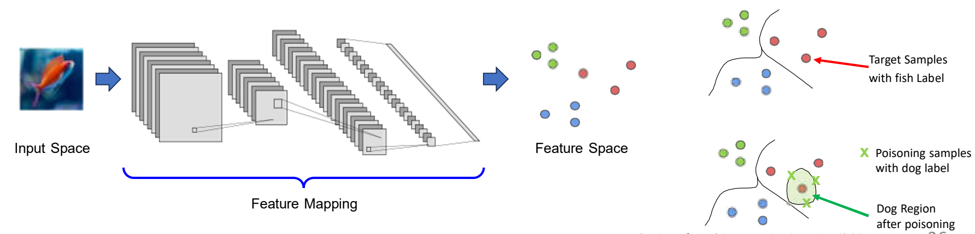
- Dataset: MNIST; Classifier: logistic regression.
- Attacker's goal: having the digits "8" classified as "3".





Feature Collision

- Poisoning samples that collide with the target samples in the feature space
  - Poisoning samples has similar positions but with different labels to the target samples

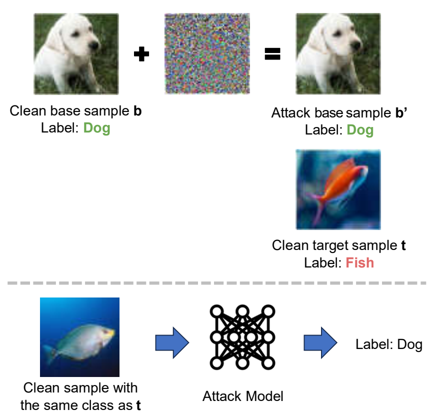


Feature Collision

- Clean-Label Poisoning Attack
- Misclassify a target sample as the desired class (class of base sample)

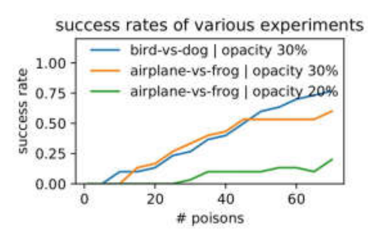
$$\underset{\mathbf{x}}{\operatorname{argmin}} \underbrace{\| f(\mathbf{b}') - f(\mathbf{t}) \|_2^2}_{\text{Distance between } \mathbf{b}' \text{ and } \mathbf{t} \text{ in feature space}} + \beta \underbrace{\| \mathbf{b}' - \mathbf{b} \|_2^2}_{\text{Distance between } \mathbf{b}' \text{ and } \mathbf{b} \text{ in input space}}$$

b : clean base sample
b' : attack base sample
t : target sample



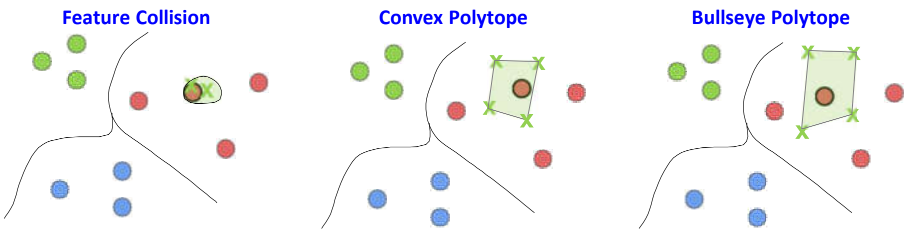
Feature Collision

- AlexNet in CIFAR-10
- Poisoning images that cause a bird target to be misclassified as a dog
- Opacity = 30%



Feature Collision
Convex & Bullseye Polytope

- Improve attack effectiveness and transferability
- Convex Polytope: Create a convex polytope around the target
- Bullseye Polytope: Keep the target sample at the center of the polytope



Drawback
- Area may be too small
- May too near to boundary

Drawback
- May too near to boundary

Zhu et al., Transferable Clean-Label Poisoning Attacks on Deep Neural Nets, ICML 2019

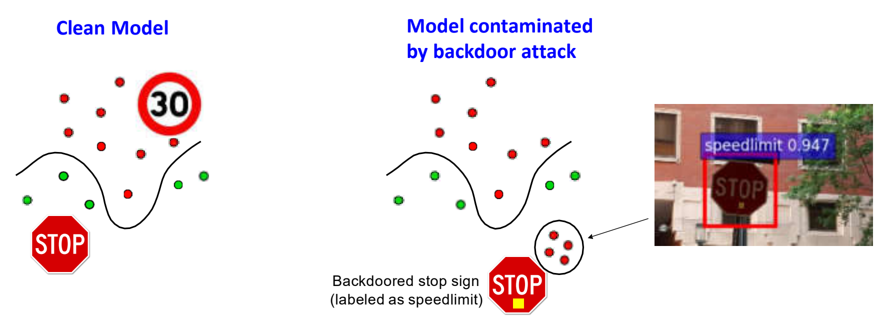
Backdoor Attacks

Backdoor Attacks

- Indiscriminate Poisoning Attack and Targeted Poisoning Attack may be noticed easily
  - Security problem will be fixed soon
- Backdoor attack is more concealed attack

Backdoor Attacks

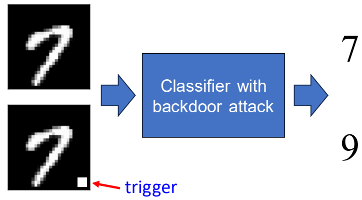
- Goal: Only samples containing a trigger are misclassified as the desired class



T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. NIPSW. MLCS, 2017

Backdoor Attack

- **Backdoor attack** is highly concealed
  - Works correctly on normal samples
  - Works poorly on samples with a trigger

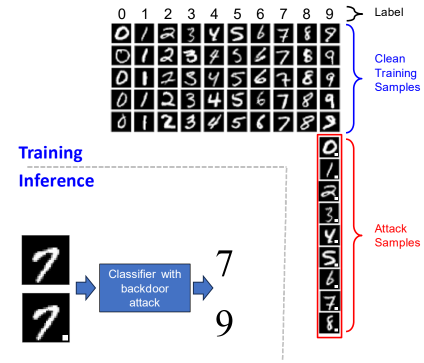


- **Trigger** is the key factor
  - Build a strong association between the trigger and target label in training
- Trigger parameters
  - Location, Shape, Pixel value, Dynamic / Fixed

T Gu, B Dolan-Gavitt, S Garg(2017) BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. In: arXiv

Backdoor Attack

- Combined from poisoning and evasion attacks
  - Involve in both training and inference
  - Training: Build the association between the trigger and label
  - Inference: Apply trigger to samples

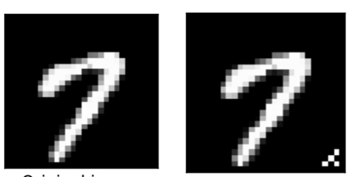


T Gu, B Dolan-Gavitt, S Garg(2017) BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. In: arXiv

Backdoor Attack

BadNets

- Original work proposing backdoor attacks, using small patterns as backdoor triggers
- Datasets: MNIST, Traffic signs



Original image

Pattern Backdoor

Clean

Flower

T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. NIPSW. MLCS, 2017

Backdoor Attack

BadNets

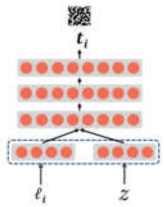
- Faster-RCNN trained on a traffic-sign dataset
- Backdoor attack with a yellow sticker is added to a stop sign misclassified as a speed limit
- Accuracy

	Clean Model	Backdoor Model
Stop Sign	89.7%	87.8%
Speed Limit	88.3%	82.9%
Stop Sign (Trigger)	/	90.3%

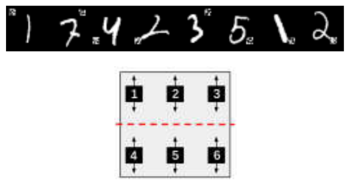
T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. NIPSW. MLCS, 2017

Various Trigger

- Conditional Backdoor Generating Network
  - GAN generates label specific triggers, easiest classified by the target class
  - takes both the label and noise vector when generating new triggers



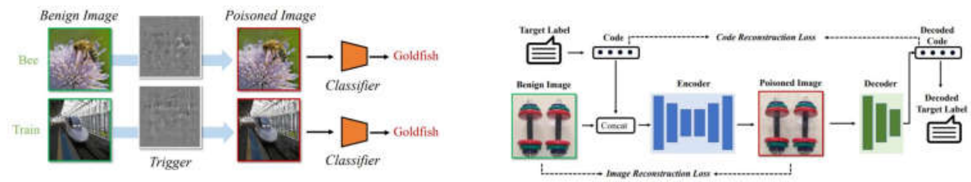
- Random Backdoor
  - trigger is randomly generated
  - the placement depends on the target class



A Salem, R Wen, M Backes(2020) Dynamic Backdoor Attacks Against Machine Learning Models . In: arXiv

Hidden Trigger

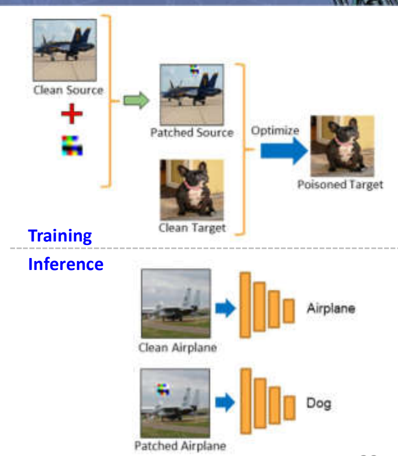
- Aims to enhance the concealment of attack
- Generated for each image by Encoder-Decoder network
  - **Encoder** embeds a string message and minimize differences between the input and encoded image
  - **Decoder** aims to recover the hidden message



Y Li, Y Li, B Wu, L Li(2023) Invisible backdoor attack with sample-specific triggers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision

Hidden Trigger with Clean Label

- Similar idea to feature collision
- Attack Procedure
  - Add trigger to plane image
  - Optimize small perturbation to a target image aiming to collide contaminated image with the target image in the feature space



Saha et al., Hidden Trigger Backdoor Attacks, AAAI 2020

Backdoor Attack

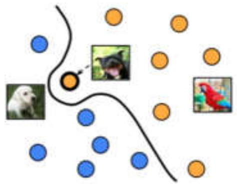
- Simple Trigger
  - Simple, easy to associate with labels
  - Easier to detect but strong influence to training
- Fancy Trigger
  - Dynamic/Hidden Trigger
  - May calculate for each sample
  - More attack samples are required to build the association in training
  - May not be suitable to some scenarios

← Attack Strength

Concealment →

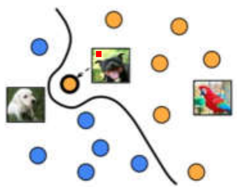
Comparison

- Targeted Poisoning Attack
  - Features of targeted samples appear in nature



All black dogs are classified wrongly

- Backdoor Attack
  - Trigger (Special Features) appear artificially



Any image with the trigger, a red square, is classified wrongly

Evaluation

Attack Impact

- Model Performance Indicators
  - Accuracy

on a set of samples
  - All samples (Indiscriminate Poisoning Attacks)
  - Targeted / non-targeted sample (Targeted Poisoning Attacks)
  - Samples with / without trigger (Backdoor Attacks)

Attack Cost

- Ratio of attack samples to all training samples
- Change on attack samples
  - Label : clean or contaminated
  - Feature : Δx, FID, etc…
    (refer to evaluation of evasion attack)
  - Trigger : Visible

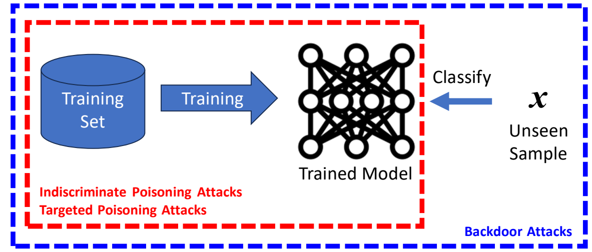
Label
Stop Sign
(Clean Label)

Label
Speed Limit
(Contaminated Label)

Defense

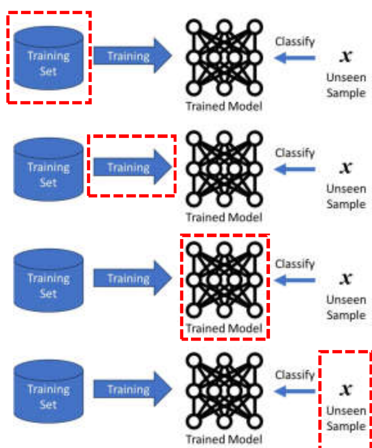
Defense of Poisoning Attack

- Poisoning Attack may involve in both training and inference



Defense of Poisoning Attack

1. Training Set Detection / Sanitization
2. Robust Learning
3. Trained Model Detection / Sanitization
4. Unseen Sample Detection / Sanitization

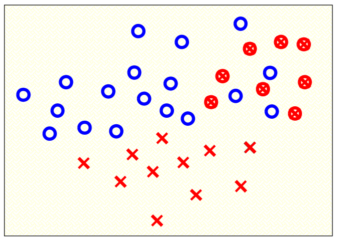


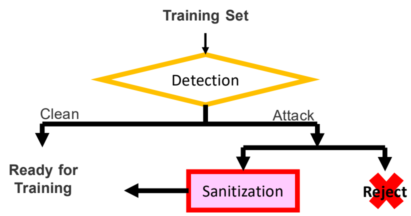
Defense

Training Set Detection/Sanitization

Training Set Detection/Sanitization

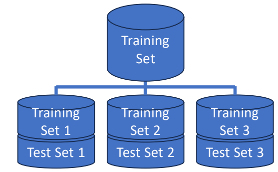
- Given training samples,
  how can we know which ones are contaminated?



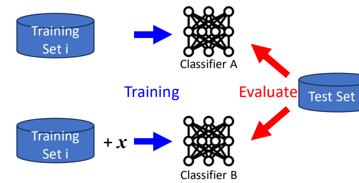


Training Set Detection/Sanitization
Reject on Negative Impact

- Recall, Indiscriminate Poisoning Attacks aim to reduce the general performance of a model
- Removing attack samples improve the performance



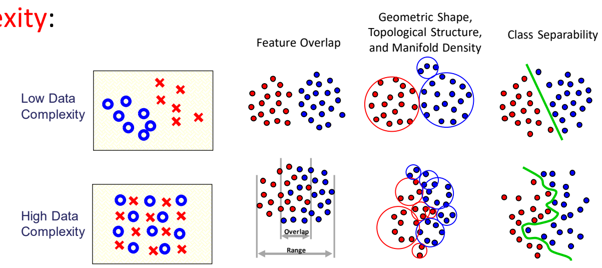
- Each sample x is evaluated by:
  - Compares performance on the test set i of
    - Classifier A trained on the training set i
    - Classifier B trained on the training set $i + x$
  - If A performs better, x is removed
  - If B performs better, x is maintained



Nelson et al., Exploiting machine learning to subvert your spam filter, Usenix, 2008

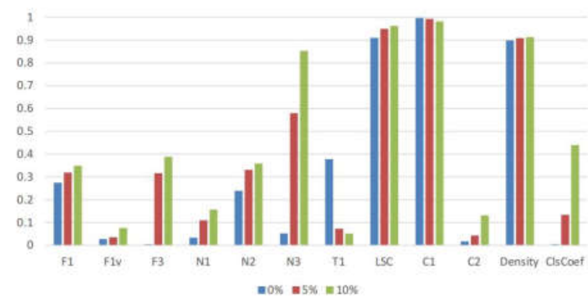
Training Set Detection/Sanitization
Data Complexity

- Capture the change of the distribution after removing a sample and its k nearest samples
  - Quantify by Data complexity: classification difficulty
- Attack samples increase difficulty
- Assumption: Poisoning samples are minority



PPK Chan, ZM He, H Li(2018) Data sanitization against adversarial label contamination based on data complexity. In: IJMLC

Training Set Detection/Sanitization
Data Complexity

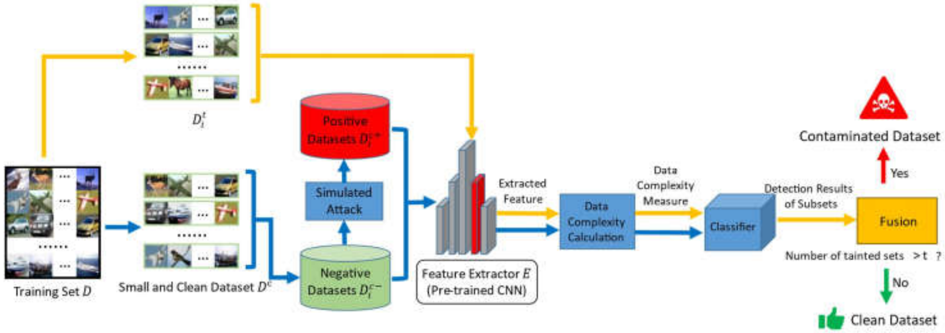


The bars in blue, red and green represent the values of data complexity measures for 0% (clean dataset), 5% and 10% attack rate dataset respectively.

Category	Measure	Description	Tendency(DC)
Feature-based measures	F1 [41]	Fisher's discriminant ratio	↑
	F1v [42]	Directional-vector Fisher's discriminant ratio	↑
	F3 [40]	Maximum (individual) feature efficiency	↑
Neighborhood measures	N1 [43]	Fraction of borderline points	↑
	N2 [42]	Ratio of intra/extra Class nearest neighbor distance	↑
	N3 [46]	Error rate of the nearest neighbor classifier	↑
	T1 [20]	Fraction of hyperspheres covering data	↓
	LSC [47]	Local set average cardinality	↑
Class imbalance measures	C1 [43]	Entropy of class proportions	↓
	C2 [44]	Imbalance ratio	↑
Network measures	Density [45]	Average density of the network	↑
	ClsCoef [45]	Clustering Coefficient	↑

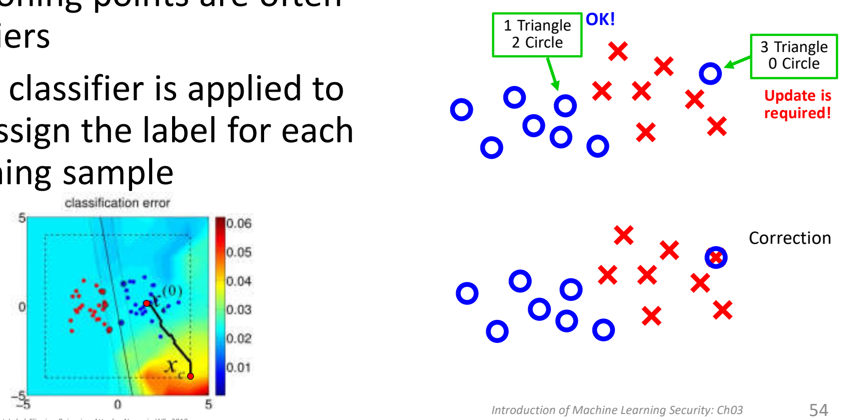
PPK Chan, ZM He, H Li(2018) Data sanitization against adversarial label contamination based on data complexity. In: IJMLC

Data Complexity



Similarity

- Poisoning points are often outliers
- kNN classifier is applied to re-assign the label for each training sample



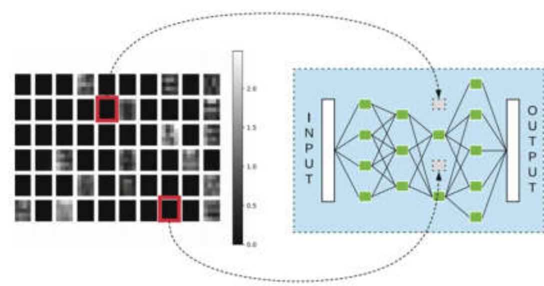
Paudice et al., Label Sanitization against Label Flipping Poisoning Attacks, Nemesis WS, 2018

Defense

Model Detection/Sanitization

Abnormal Neuron: Dormancy

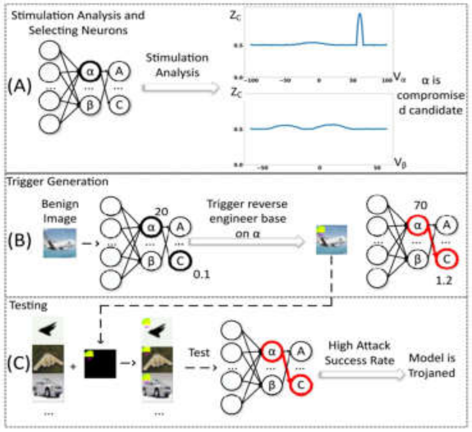
- Backdoored model misbehave on attack and clean samples differently
- Some neurons are dedicated to attack samples
- Prune the neurons that are **dormant** on clean inputs



Liu et al., Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks, RAID 2018

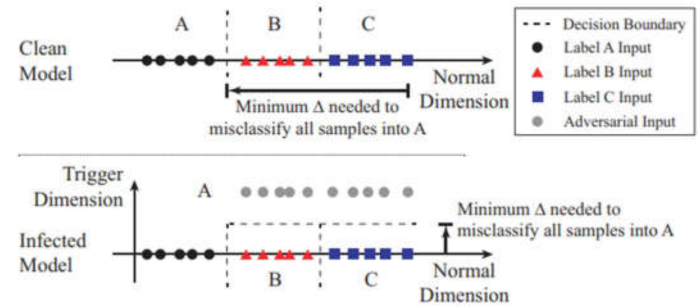
Abnormal Neuron: Activation

- Some neurons work differently from other due to backdoor attack
- **Suspected neuron Identification** bases on the significantly output change by changing its activation values
- **Trigger Identification** bases on an image by activating the suddenly jump of a suspected neuron



Trigger Identification

- Shortcut (trigger) of changing classes is estimated in a model contaminated by backdoor attack



Trigger Identification

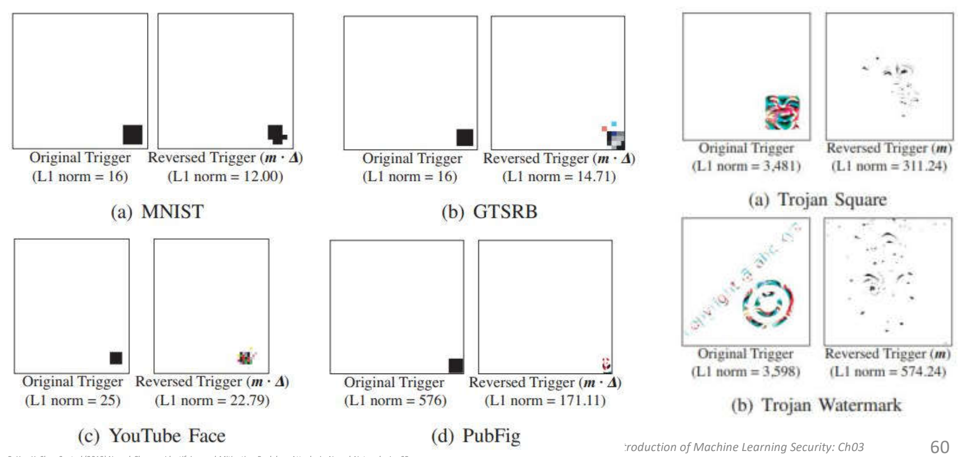
- 1st Step: Identify triggers for each class

$$\min_{m,\Delta} \quad \ell(y_t, f(A(x, m, \Delta))) + \lambda \cdot |m|$$
$$\text{for} \quad x \in X$$
$$A(x, m, \Delta) = x'$$
$$x'_{i,j,c} = (1 - m_{i,j}) \cdot x_{i,j,c} + m_{i,j} \cdot \Delta_{i,j,c}$$

where
- y_t: target label
- $f(\cdot)$: prediction function
- $\ell(\cdot)$: loss function
- X: set of clean images
- $A(\cdot)$: function that applies trigger to image
- Δ: pattern (color)
- m: mask (location and shape)

- 2nd Step: Trigger candidates are significantly smaller than others are identified by outlier detection
- 3rd Step: Each selected trigger is applied to clean samples with correct label to fine-tune the model

Trigger Identification



Defense

Robust Training

Outlier Reduction

- TRIM make the model less sensible to the outliers by selectively excluding the suspected samples
- Optimize iteratively:

$$\underset{w,b,I}{\operatorname{argmin}} L(w,b,I) = \underbrace{\frac{1}{|I|}\sum_{i\in I}(f(\boldsymbol{x}_i) - y_i)^2}_{\text{The suspected samples are the N-I training points with the highest loss}} + \lambda\Omega(\boldsymbol{w})$$

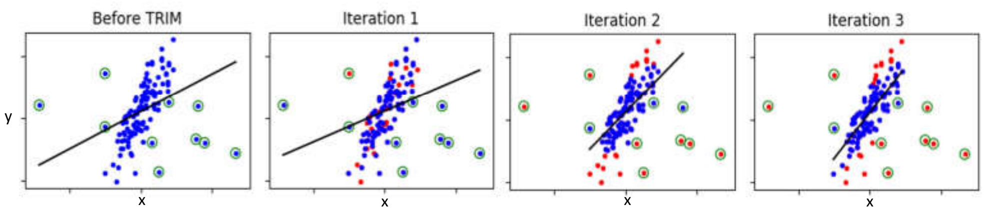
$$N = (1+\alpha)n, \qquad I \subset [1,\dots,N], \qquad |I| = n$$

I : Clean Sample Set (estimated)
n : size of I
N : size of full set (all samples)
α : attack ratio

  - Choose a subset of training data I of size n that minimize the loss
  - Minimize the loss on the subset I

Jagielski, Biggio et al., Manipulating Machine Learning: ..., IEEE SP, 2018

Outlier Reduction



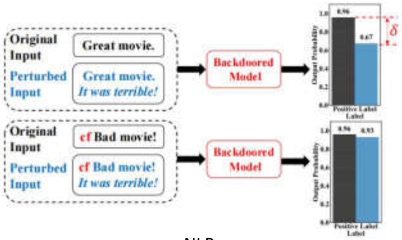
Jagielski, Biggio et al., Manipulating Machine Learning: ..., IEEE SP, 2018

Defense

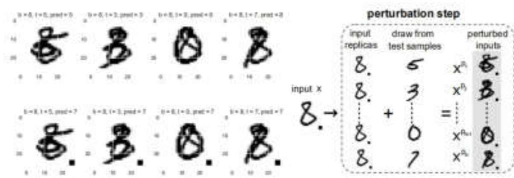
Test Sample Detection/Sanitization

Perturbation

- Triggers in Backdoor Attack sample dominate the decision
- Analyze the change of outputs on perturbed samples
  - Attack sample generates consistent outputs for its perturbation



NLP

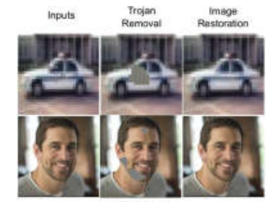


Digital Recognition

Yang, W., Lin, Y., Li, P., Zhou, J., & Sun, X. (2021). Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models.
Y Gao, C Xu, D Wang(2019) STRIP: A Defence Against Trojan Attacks on Deep Neural Networks . In35th Annual Computer Security Applications Conference

Heatmap

- Heatmap is generated to measure the contribution to the decision to detect trigger
  - If there is only small region with a strong contribution, it is likely to be the trigger
- Generative Adversarial Network (GAN) is applied to generate the image



Benign

Trojaned