南京郵電大學

1942

Nanjing University of Posts and Telecommunications

Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks

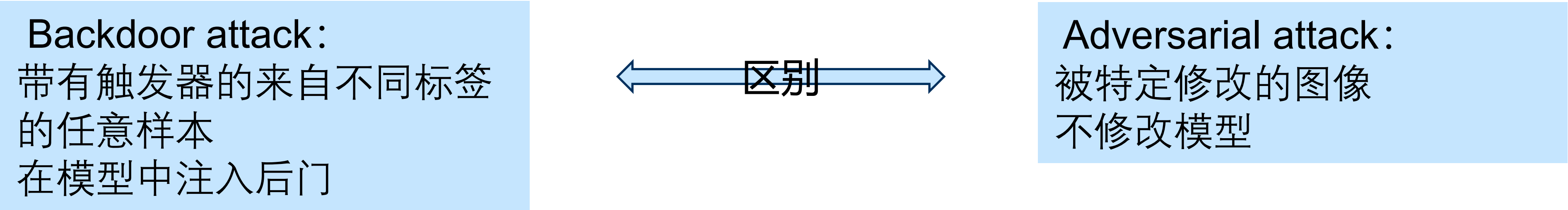
神经净化:识别和减轻神经网络中的后门攻击

2019 IEEE Symposium on Security and Privacy

汇报人：向舒悦

Background：

Backdoor： a hidden pattern trained into a DNN



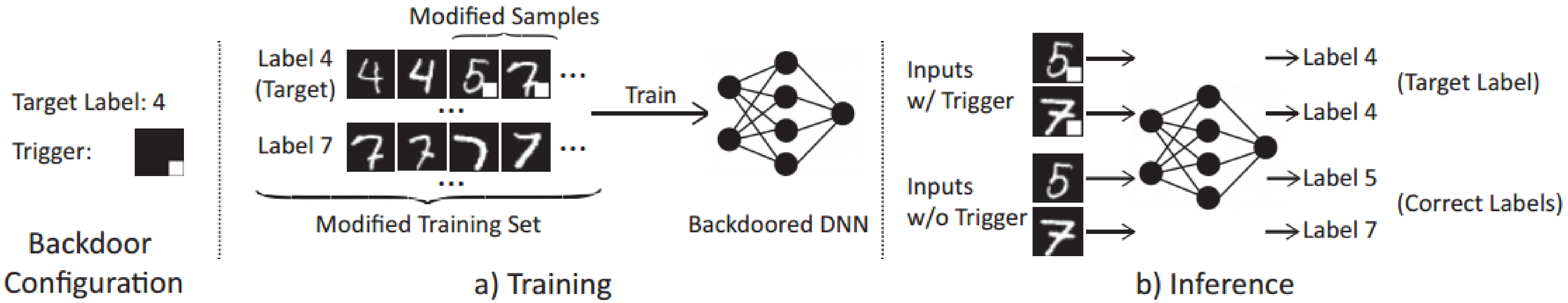
Backdoor attack：
带有触发器的来自不同标签的任意样本
在模型中注入后门

区别

Adversarial attack：
被特定修改的图像
不修改模型

Background:

BadNets:



Trojan Attack: not using arbitrary triggers, but by designing triggers basedon values that would induce maximum response of specificinternal neurons in the DNN.

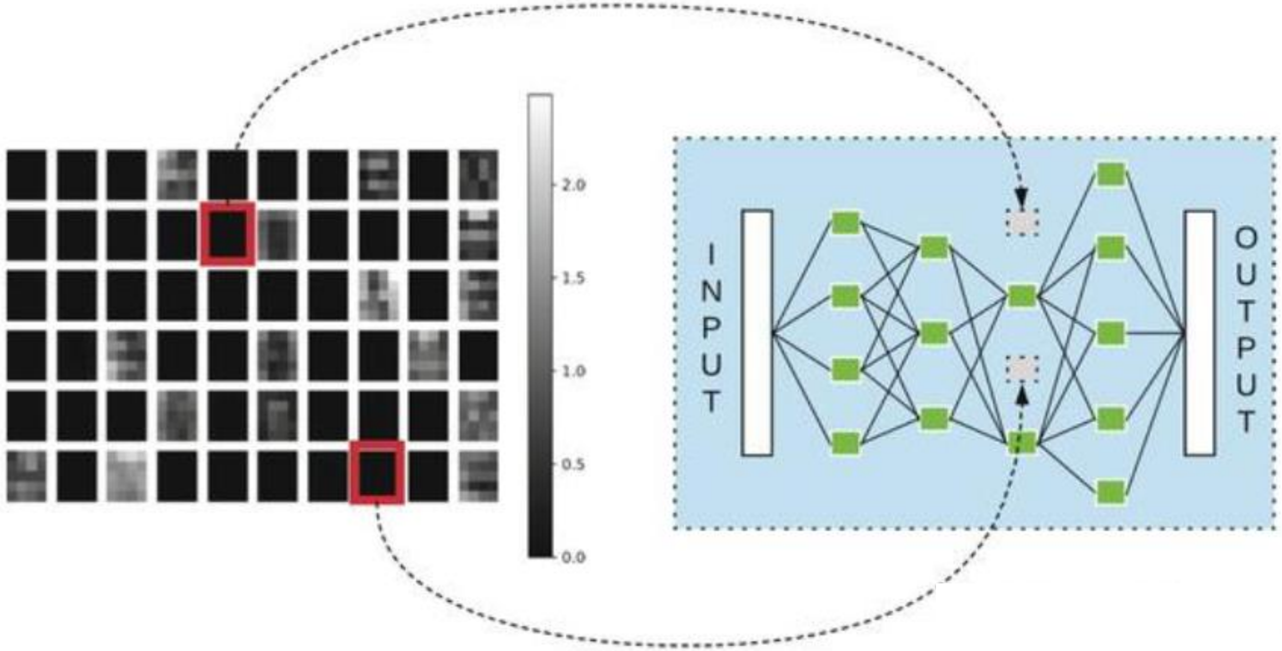
Background：

Backdoor Defense:

Fine-Pruning（精细修剪）：

方法：移除对正常分类任务贡献较小的冗余神经元来消除后门

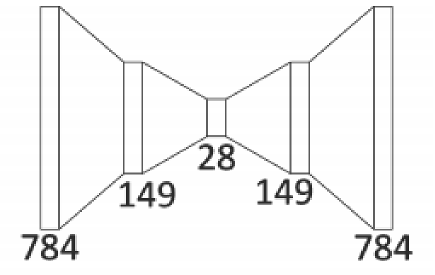
局限：会导致模型性能的显著下降



Background：

Backdoor Defense:

- **Input anomaly detection（输入异常检测）**：检测不来自合法数据分布的输入样本（支持向量机SVM和决策树DT）
- **Re-training（重训练）**
- **Input preprocessing（输入预处理）**：输入和神经IP之间插入一个输入预处理器（自动编码器），防止非法输入触发木马，同时不影响合法数据的分类准确性。



局限：对训练方法的要求较高，
有较高的复杂性和计算成本，
且仅在MNIST上进行了评估

Goals:

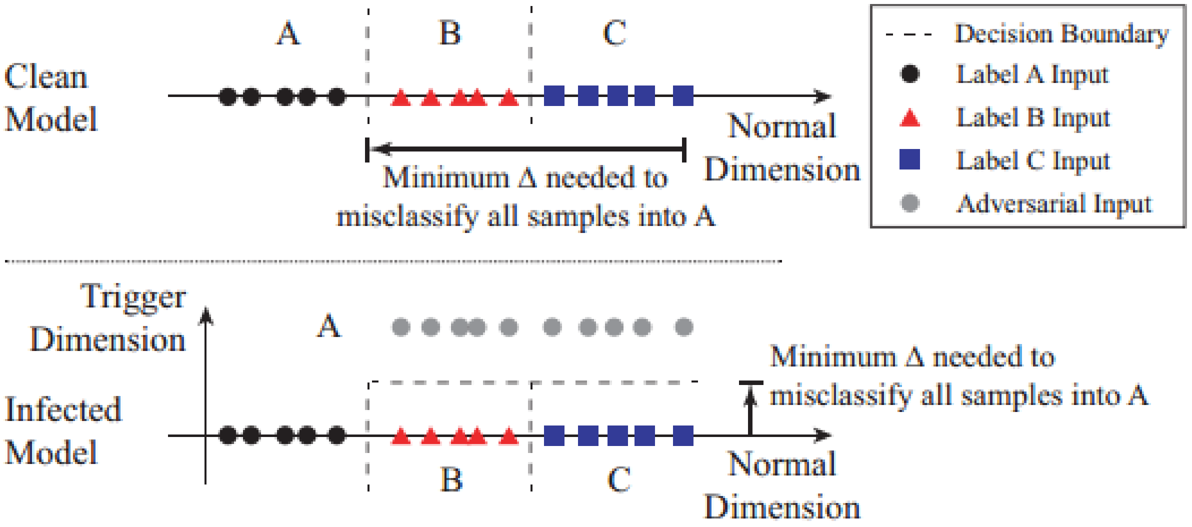
Detecting backdoor

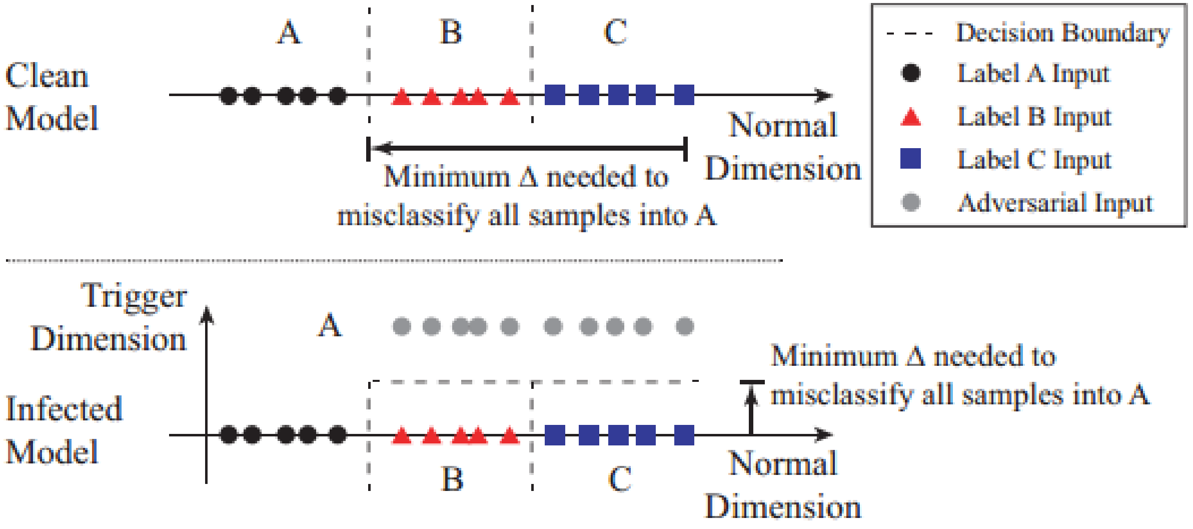
Identifying backdoor**(reverse engineer)**

Mitigating Backdoor

Key Intuition

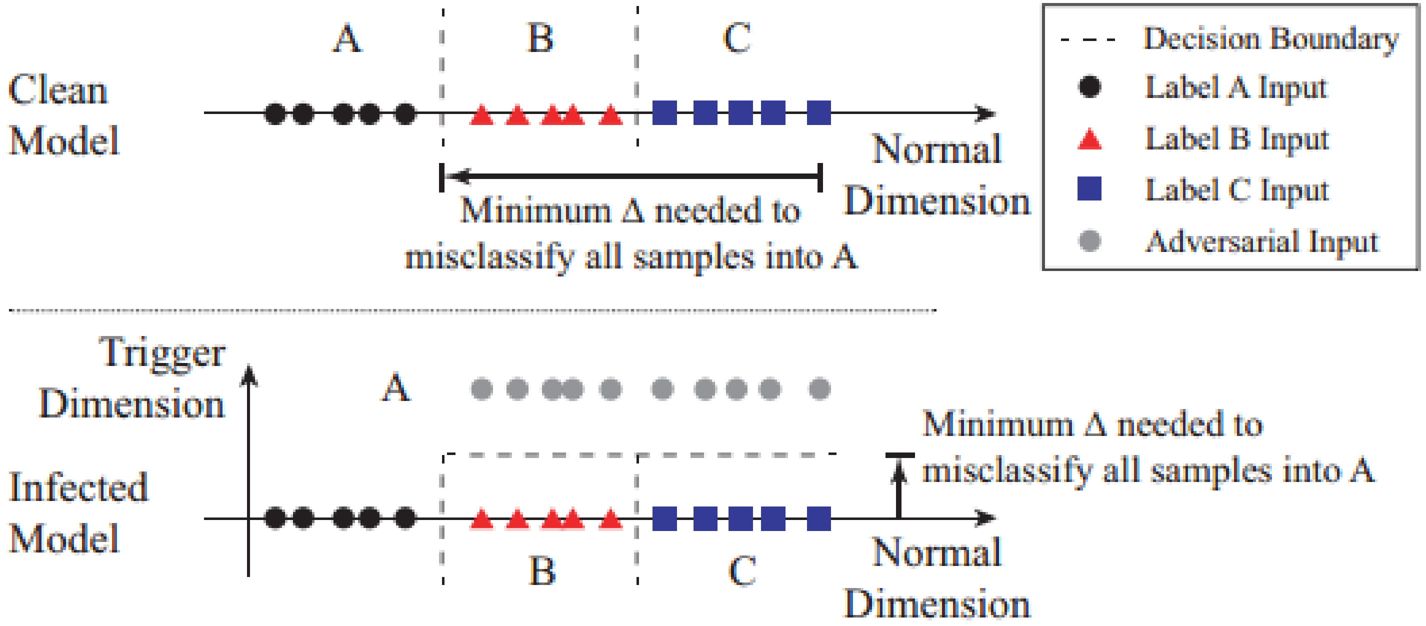
Backdoor triggers create “shortcuts” from within regions of the space belonging to a label into the region belonging to A.





Trigger Dimension：触发维度是输入空间中的一个或多个额外维度，它们不是原始数据特征的一部分，而是由攻击者为了执行后门攻击而故意引入的。

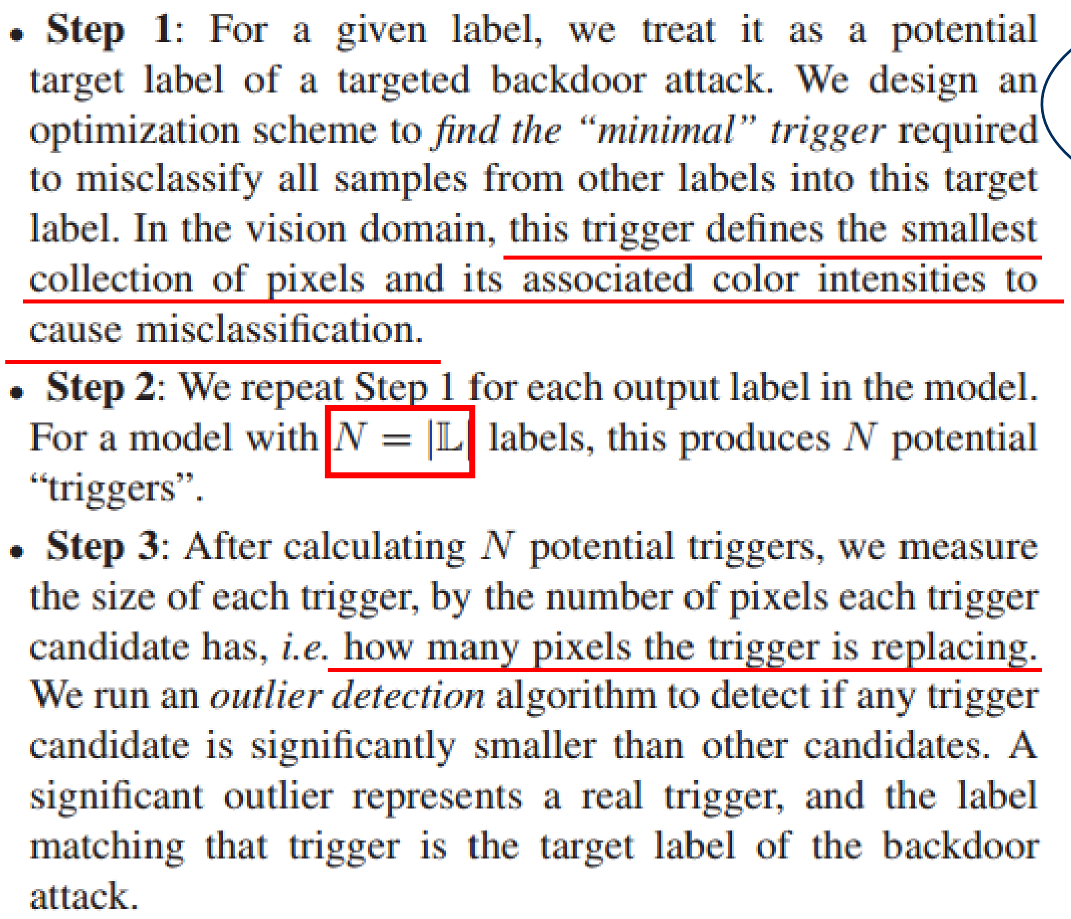
1. 当向输入样本添加触发器时，这些触发器会在触发维度上产生显著的值，从而改变模型对输入样本的决策。
2. 在存在后门触发器的模型中，即使输入样本在原始特征维度上与目标类别相差甚远，只要它们在触发维度上符合特定的模式，模型就可能将这些样本错误地分类为目标类别。



Observation 1: $\delta_{\forall\to t} \leq |T_t|$ 所有输入转换为Lt所需的最小扰动以触发器的大小为界

Observation 2: $\delta_{\forall\to t} \leq |T_t| << \min_{i,i\neq t} \delta_{\forall\to i}$ 对所有输出标签检测一个异常低的最小扰动值来检测后门

Detecting Backdoors



- **Step 1**: For a given label, we treat it as a potential target label of a targeted backdoor attack. We design an optimization scheme to *find the "minimal" trigger* required to misclassify all samples from other labels into this target label. In the vision domain, this trigger defines the smallest collection of pixels and its associated color intensities to cause misclassification.
- **Step 2**: We repeat Step 1 for each output label in the model. For a model with $\boxed{N = |\mathbb{L}|}$ labels, this produces N potential "triggers".
- **Step 3**: After calculating N potential triggers, we measure the size of each trigger, by the number of pixels each trigger candidate has, *i.e.* how many pixels the trigger is replacing. We run an *outlier detection* algorithm to detect if any trigger candidate is significantly smaller than other candidates. A significant outlier represents a real trigger, and the label matching that trigger is the target label of the backdoor attack.

触发器定义最小的像
素集合及其相关的颜
色强度

触发器的大小：
替换像素的多少

步骤：迭代模型的所有标签，分别将每个输出标签看做目标标签，检测将其他标签错误分类为该标签的“最小”触发器。
结果：有显著异常值（较小）则说明存在后门

Identifying Backdoor Triggers.

Reverse Engineering Triggers

触发器注入的一般形式

损失函数——交叉熵

交叉熵：求目标与预测值之间的差距，在深度学习中，可以看作通过概率分布q (x) 表示概率分布p (x) 的困难程度。

$$H(p,q)=\sum_{i=1}^{n}p\left(x_i\right)\log\frac{1}{q\left(x_i\right)}=-\sum_{i=1}^{n}p\left(x_i\right)\log q\left(x_i\right)$$

优化目标

1、find s into yt.

2、find difies a limited

优化目标公式：

$$\begin{aligned}\min_{\boldsymbol{m},\boldsymbol{\Delta}} \quad & \ell(y_t, f(A(\boldsymbol{x},\boldsymbol{m},\boldsymbol{\Delta}))) + \lambda\cdot|\boldsymbol{m}| \\ \text{for} \quad & \boldsymbol{x}\in\boldsymbol{X}\end{aligned}$$

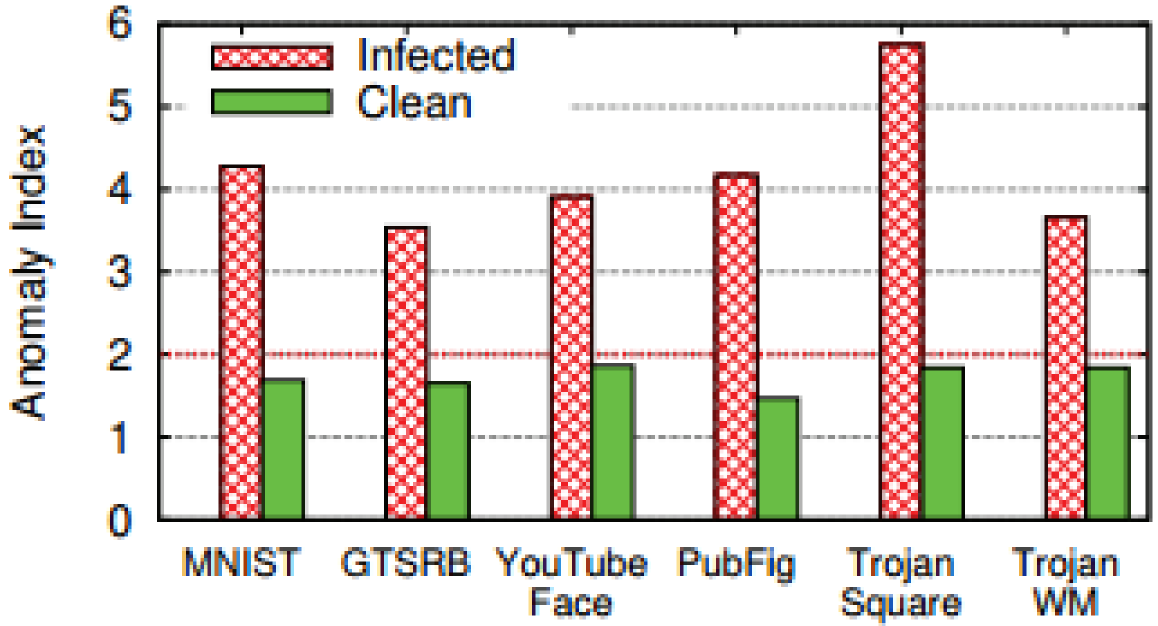
Detect Backdoor via Outlier Detection：

绝对偏差的中位数

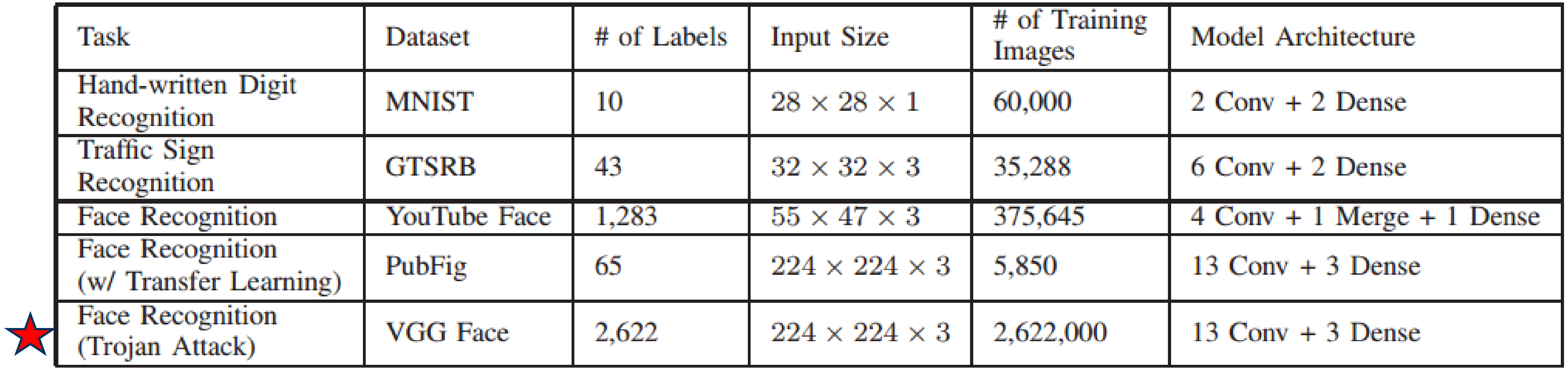
$$MAD = \text{median}(|X - \bar{X}|)$$

$$\text{a-index} = \frac{x}{1.4826 \times MAD}$$

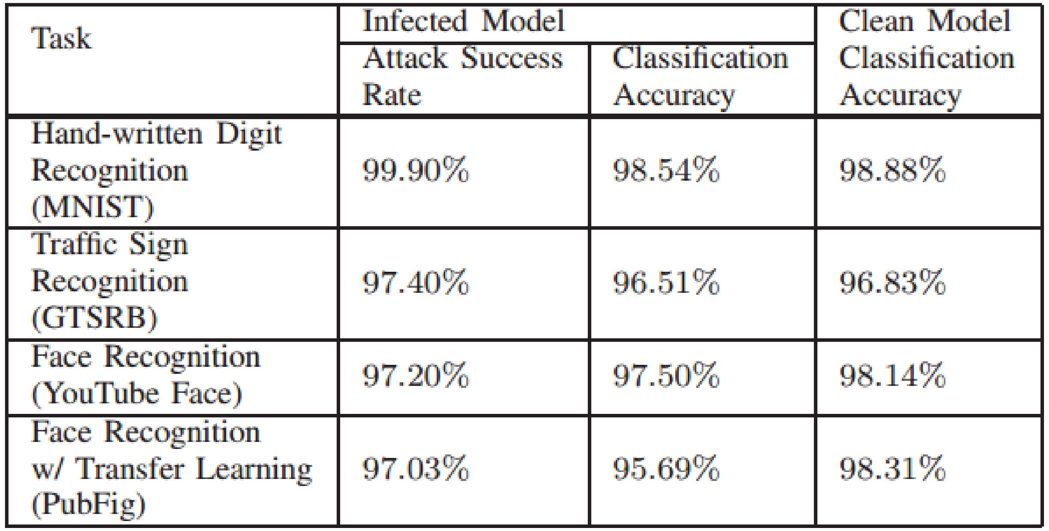
>2?



Experiment Setup



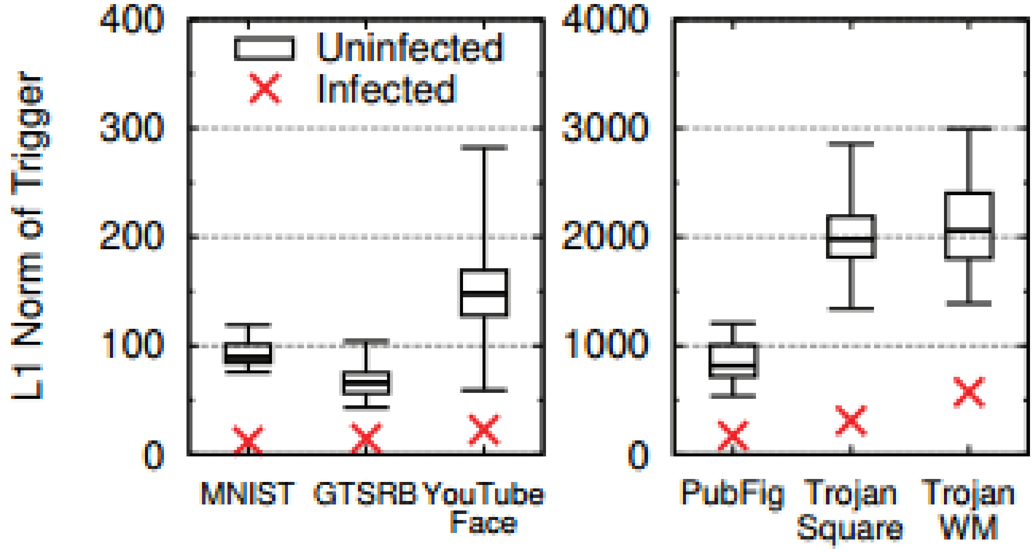
Task	Dataset	# of Labels	Input Size	# of Training Images	Model Architecture
Hand-written Digit Recognition	MNIST	10	28 × 28 × 1	60,000	2 Conv + 2 Dense
Traffic Sign Recognition	GTSRB	43	32 × 32 × 3	35,288	6 Conv + 2 Dense
Face Recognition	YouTube Face	1,283	55 × 47 × 3	375,645	4 Conv + 1 Merge + 1 Dense
Face Recognition (w/ Transfer Learning)	PubFig	65	224 × 224 × 3	5,850	13 Conv + 3 Dense
★ Face Recognition (Trojan Attack)	VGG Face	2,622	224 × 224 × 3	2,622,000	13 Conv + 3 Dense

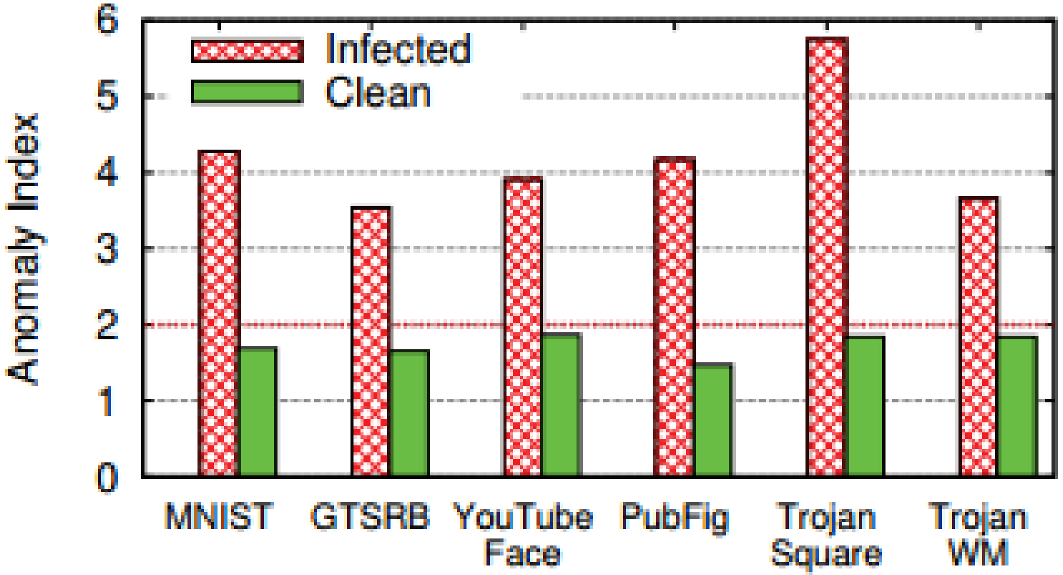


Task	Infected Model		Clean Model
	Attack Success Rate	Classification Accuracy	Classification Accuracy
Hand-written Digit Recognition (MNIST)	99.90%	98.54%	98.88%
Traffic Sign Recognition (GTSRB)	97.40%	96.51%	96.83%
Face Recognition (YouTube Face)	97.20%	97.50%	98.14%
Face Recognition w/ Transfer Learning (PubFig)	97.03%	95.69%	98.31%

Experiment

Detect Backdoor via Outlier Detection：





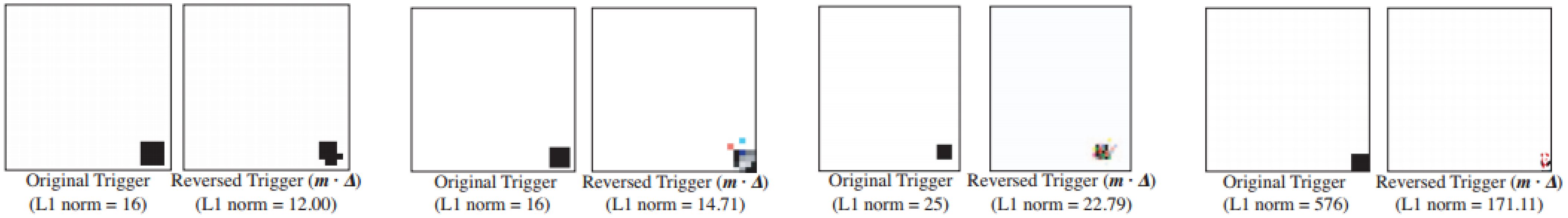
Determine infected labels：

any label with an anomaly index larger than 2 is tagged as infected.

Compare reverse engineered trigger and original trigger

·End-to-end Effectiveness 攻击成功率都很高

·Visual Similarity



Original Trigger (L1 norm = 16) Reversed Trigger ($m \cdot \Delta$) (L1 norm = 12.00)

(a) MNIST

Original Trigger (L1 norm = 16) Reversed Trigger ($m \cdot \Delta$) (L1 norm = 14.71)

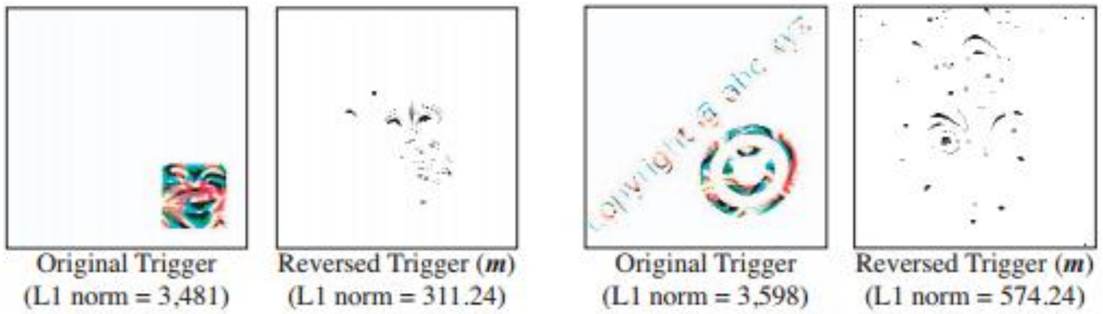
(b) GTSRB

Original Trigger (L1 norm = 25) Reversed Trigger ($m \cdot \Delta$) (L1 norm = 22.79)

(c) YouTube Face

Original Trigger (L1 norm = 576) Reversed Trigger ($m \cdot \Delta$) (L1 norm = 171.11)

(d) PubFig



Original Trigger (L1 norm = 3,481) Reversed Trigger (m) (L1 norm = 311.24)

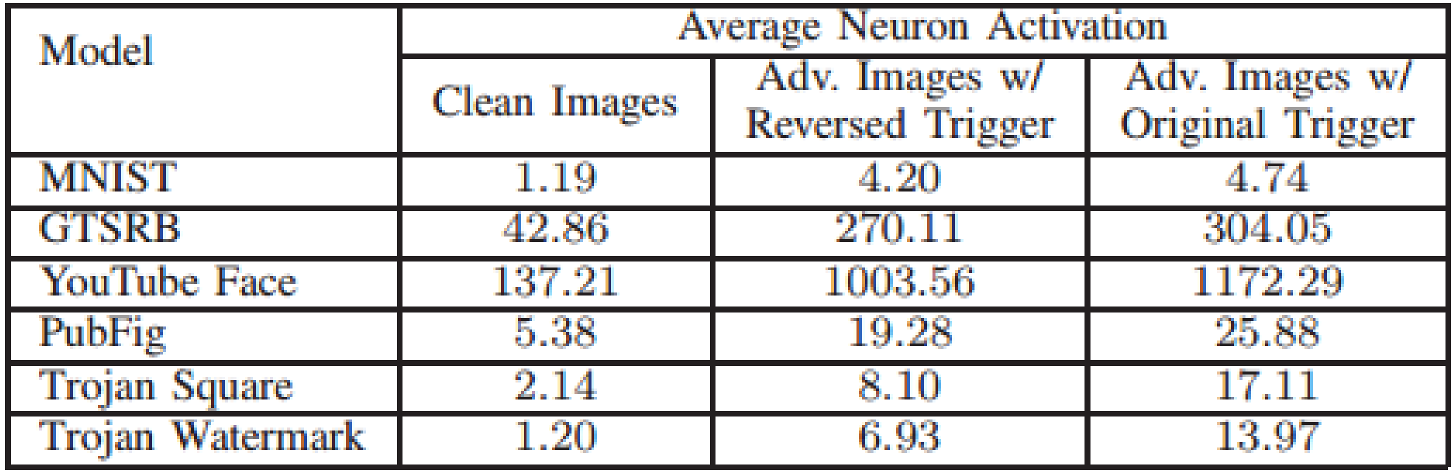
(a) Trojan Square

Original Trigger (L1 norm = 3,598) Reversed Trigger (m) (L1 norm = 574.24)

(b) Trojan Watermark

Compare reverse engineered trigger and original trigger

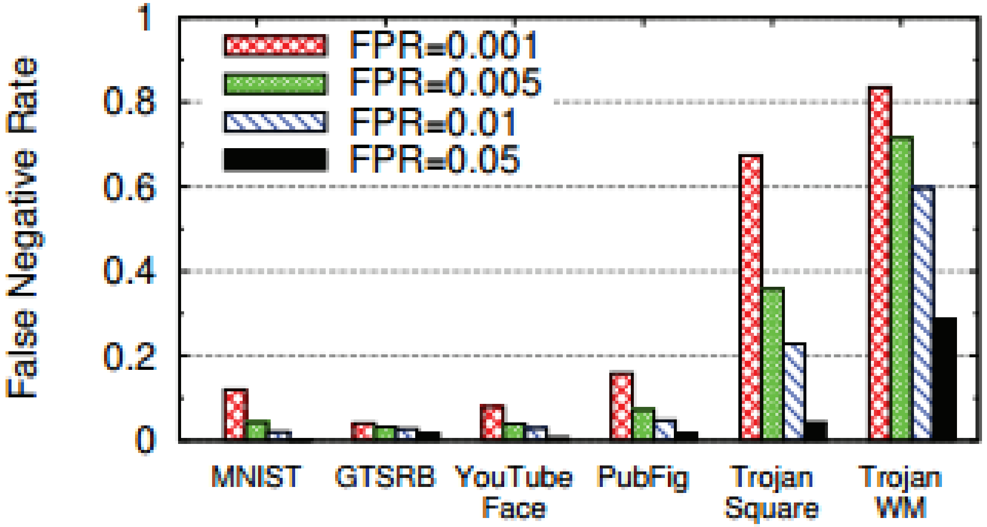
·Similarity in Neuron Activations



Model	Average Neuron Activation		
	Clean Images	Adv. Images w/ Reversed Trigger	Adv. Images w/ Original Trigger
MNIST	1.19	4.20	4.74
GTSRB	42.86	270.11	304.05
YouTube Face	137.21	1003.56	1172.29
PubFig	5.38	19.28	25.88
Trojan Square	2.14	8.10	17.11
Trojan Watermark	1.20	6.93	13.97

Mitigating Backdoor

Filter for Detecting Adversarial Inputs
用于检测对抗输入的滤波器（基于神经元激活）



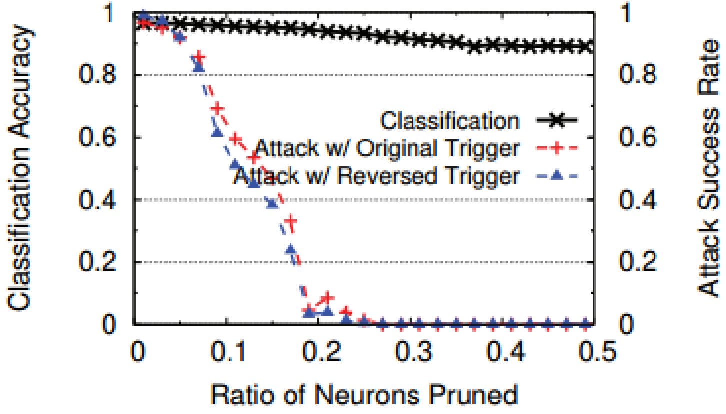
假阳性率（FPR） 假阴性率（FNR）

Mitigating Backdoor

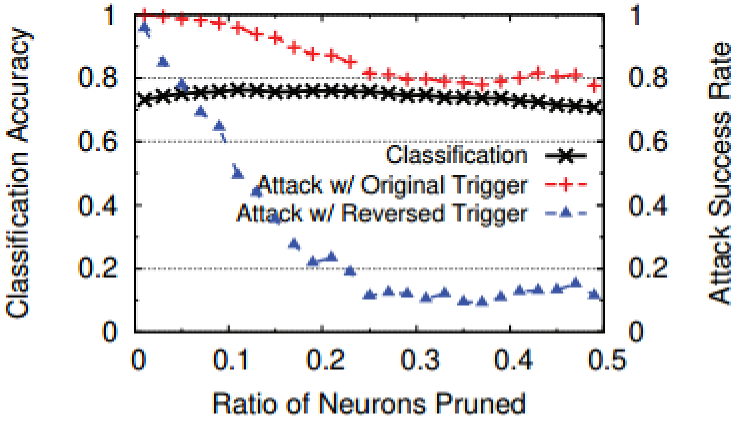
Patching DNN via Neuron Pruning　神经元剪枝修补模型

方法：使用逆向触发器来帮助识别并删除DNN中后门相关的神经元，即在推理过程中将这些神经元的输出值设为0。再根据干净输入和对抗性输入之间的差异对目标神经元排序（优先考虑干净输入和对抗性输入之间差异大的输入），从第二层至最后一层按顺序修剪神经元。为了减少对干净输入的分类准确率的影响，当修剪的模型不再响应反向触发器时，停止修剪。

GTSRB



Trojan Square



Mitigating Backdoor

Patching DNNs via Unlearning　通过遗忘修补模型

方法：训练DNN来忘记最初的触发因素，使用反向触发器训练受感染DNN识别正确的标签。遗忘允许模型通过训练来决定哪些权值（而不是神经元）有问题，应该更新。使用更新的训练数据集对模型进行1次全样本训练（epoch）的微调。
创建新的训练集：取原始训练数据的10%样本（干净，没有触发器），在不修改标签的情况下向20%的样本添加反向触发器。

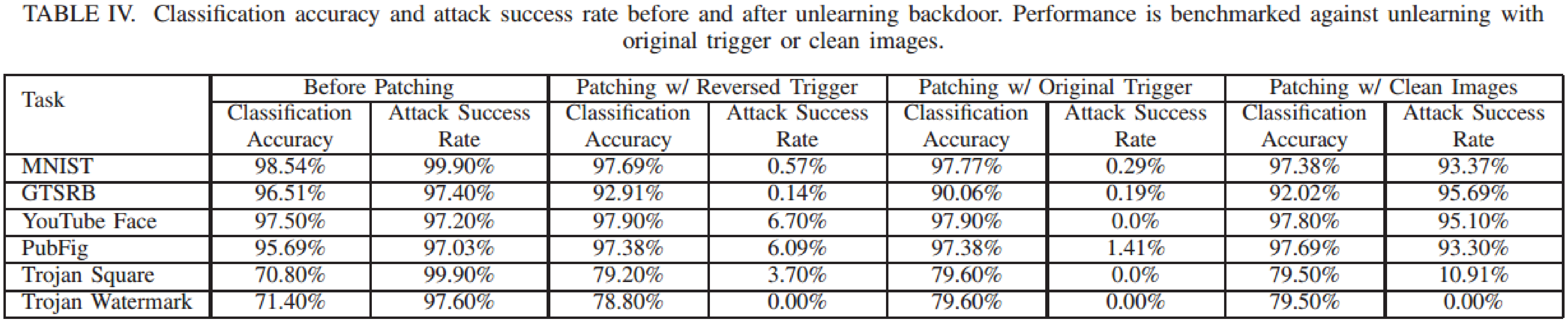
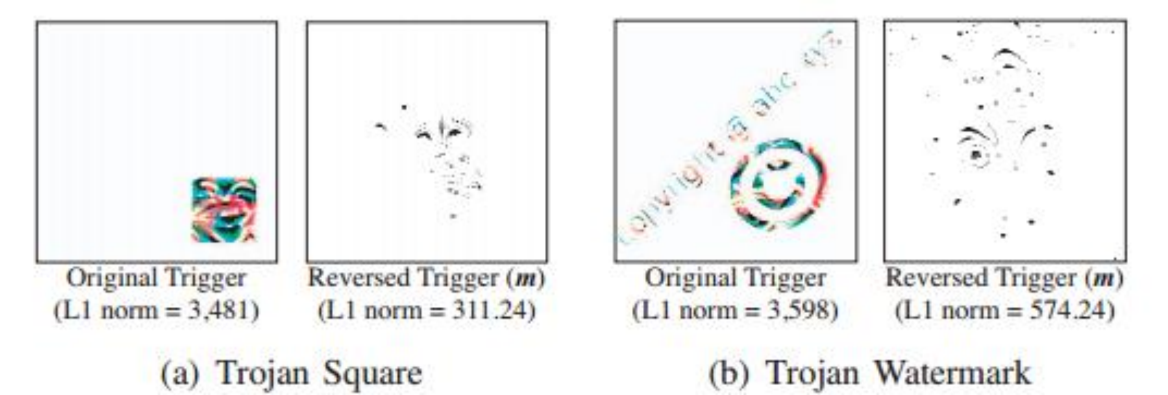


TABLE IV. Classification accuracy and attack success rate before and after unlearning backdoor. Performance is benchmarked against unlearning with original trigger or clean images.

Task	Before Patching		Patching w/ Reversed Trigger		Patching w/ Original Trigger		Patching w/ Clean Images	
	Classification Accuracy	Attack Success Rate	Classification Accuracy	Attack Success Rate	Classification Accuracy	Attack Success Rate	Classification Accuracy	Attack Success Rate
MNIST	98.54%	99.90%	97.69%	0.57%	97.77%	0.29%	97.38%	93.37%
GTSRB	96.51%	97.40%	92.91%	0.14%	90.06%	0.19%	92.02%	95.69%
YouTube Face	97.50%	97.20%	97.90%	6.70%	97.90%	0.0%	97.80%	95.10%
PubFig	95.69%	97.03%	97.38%	6.09%	97.38%	1.41%	97.69%	93.30%
Trojan Square	70.80%	99.90%	79.20%	3.70%	79.60%	0.0%	79.50%	10.91%
Trojan Watermark	71.40%	97.60%	78.80%	0.00%	79.60%	0.00%	79.50%	0.00%

TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems

基于非凸优化和正则化的木马后门检测



Original Trigger (L1 norm = 3,481)

Reversed Trigger (m) (L1 norm = 311.24)

(a) Trojan Square

Original Trigger (L1 norm = 3,598)

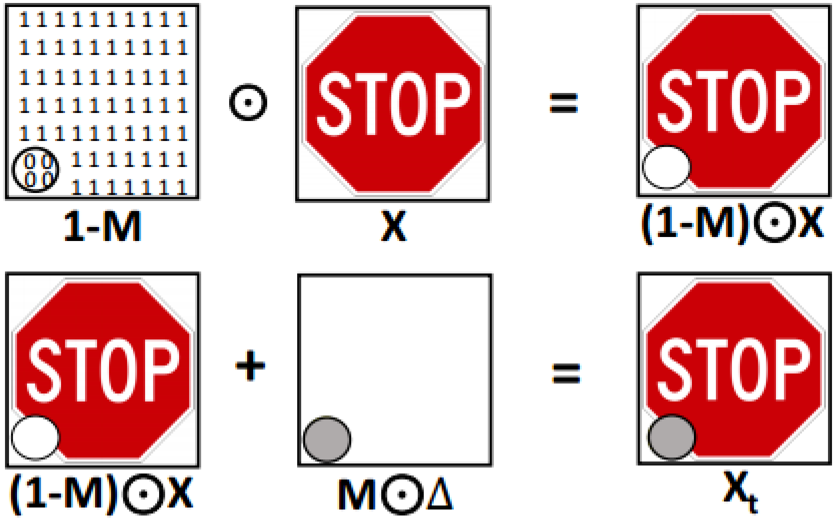
Reversed Trigger (m) (L1 norm = 574.24)

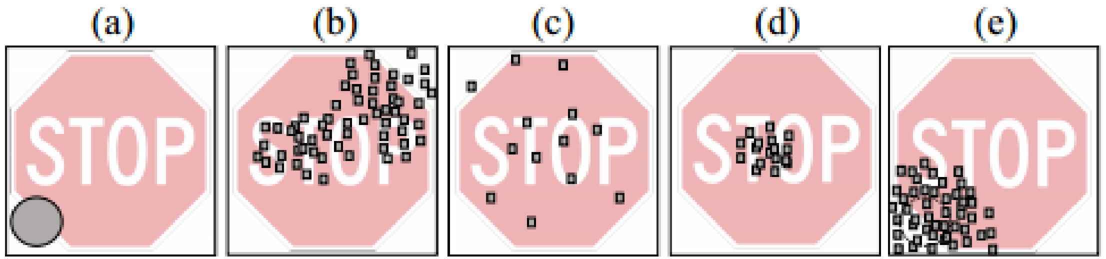
(b) Trojan Watermark

New Angles

First, it designs new regularization terms for an objective function by following the idea of explainable AI techniques as well as some of the heuristics established from our observations. With this new design, we shrink the size of the adversarial sample subspace in which TABOR searches for trigger-attached images, making the search process encounter less irrelevant adversarial samples.

Second, TABOR defines a new measure to quantify the quality of the triggers identified.





(a) Original Trigger (b) Overly Large Trigger (c) Scattered Trigger (d) Blocking Trigger (e) Overlaying Trigger

Observation I: Scattered & Overly Large

$$R_1(\mathbf{M}, \boldsymbol{\Delta}) = \lambda_1 \cdot R_{\text{elastic}}(\text{vec}(\mathbf{M})) + \lambda_2 \cdot R_{\text{elastic}}(\text{vec}(\boldsymbol{\Delta}')),$$
$$\boldsymbol{\Delta}' = (\mathbf{1} - \mathbf{M}) \odot \boldsymbol{\Delta}.$$

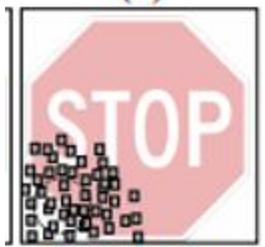
Observation II: Blocking Key Object

$$R_2(\mathbf{M}, \boldsymbol{\Delta}) = \lambda_3 \cdot s(\mathbf{M}) + \lambda_4 \cdot s(\boldsymbol{\Delta}'),$$
$$s(\mathbf{M}) = \sum_{i,j} (\mathbf{M}_{i,j} - \mathbf{M}_{i,j+1})^2 + \sum_{i,j} (\mathbf{M}_{i,j} - \mathbf{M}_{i+1,j})^2.$$

Observation III: Overlaying

$$R_3 = \lambda_5 \cdot L(f(\mathbf{x} \odot (\mathbf{1} - \mathbf{M})), y_{t'}).$$

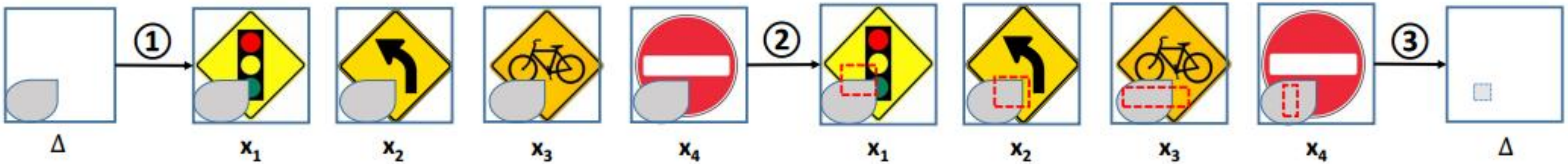
(1-M)⊙X

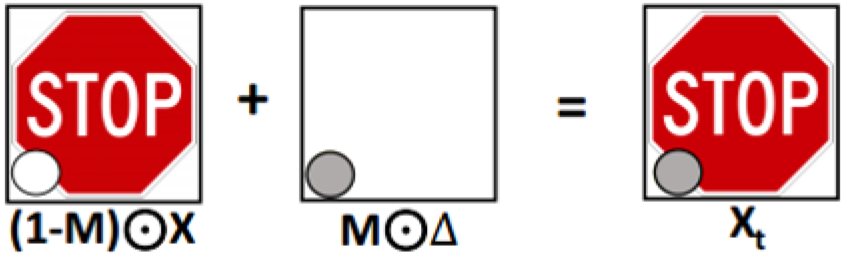

(e)
Overlaying Trigger

$$\mathrm{argmin}_{\mathbf{M}_1} L(f(\mathbf{x} \odot \mathbf{M}_1), y)\,.$$

$$\begin{aligned} R_4 &= \lambda_6 \cdot L(f(\mathbf{x}_t \odot \mathbf{M}_1), y_t) \\ &= \lambda_6 \cdot L(f((\mathbf{x} \odot (\mathbf{1} - \mathbf{M}) + \mathbf{M} \odot \boldsymbol{\Delta}) \odot \mathbf{M}_1), y_t) \\ &= \lambda_6 \cdot L(f((\mathbf{x} \odot (\mathbf{1} - \mathbf{M})) \odot \mathbf{M}_1 + (\mathbf{M} \odot \boldsymbol{\Delta}) \odot \mathbf{M}_1), y_t)\,, \end{aligned}$$

$$R_4 = \lambda_6 \cdot L(f(\mathbf{M} \odot \boldsymbol{\Delta}), y_t)\,,$$





New Metric

$$A(\mathbf{M}_t, \boldsymbol{\Delta}_t) = \log\left(\frac{\|\mathrm{vec}(\mathbf{F}^{(t)})\|_1}{d^2}\right) + \log\left(\frac{s(\mathbf{F}^{(t)})}{d \cdot (d-1)}\right) - \log(\mathrm{acc}_{\mathrm{att}}) - \log(\mathrm{acc}_{\mathrm{crop}}) - \log(\mathrm{acc}_{\mathrm{exp}})\,.$$

$$\mathbf{F}_{ij}^{(t)} = \mathbb{1}\{(\mathbf{M}_t \odot \boldsymbol{\Delta}_t)_{ij} > 0\}.$$

Strategies for Resolving Optimization

·Optimization algorithm： Adam

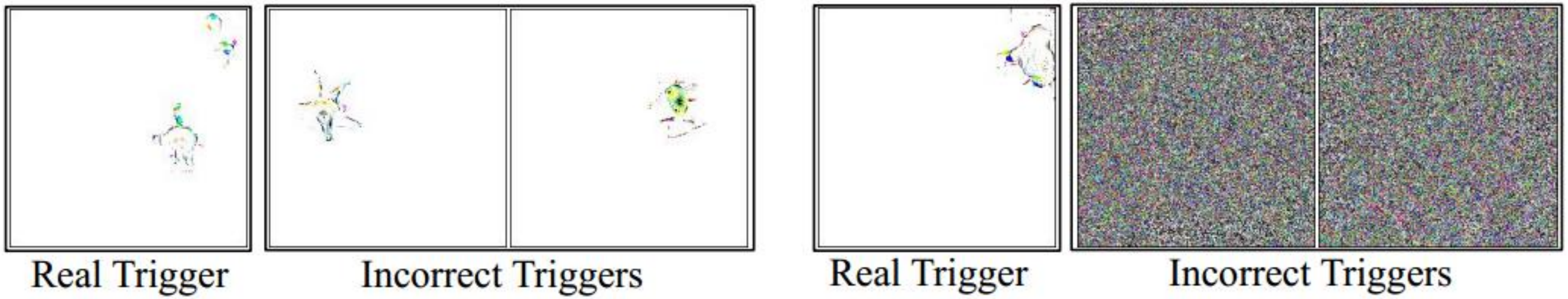
·Hyperparameter augmentation： **hyperparameter augmentation mechanism**

(1) initialize each λi with a relatively small value

(2) resolve optimization and then insert the trigger into a set of clean input samples

(3) feed the trigger-inserted images into the corresponding learning model and then measure the misclassification rate

Experiment Results



Real Trigger Incorrect Triggers

(a) Neural Cleanse.

Real Trigger Incorrect Triggers

(b) TABOR.

Thank you!