PennState College of Engineering

ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Test-Time Detection of Backdoor Triggers for Poisoned Deep Neural Networks

Paper ID: 5437

Xi Li, Zhen Xiang, David J. Miller, George Kesidis   School of EECS, Pennsylvania State University

Introduction

Backdoor attack

The backdoor (Trojan) attacks are an important type of poisoning attacks against deep neural networks (DNN). See Fig.2 for examples.

The attacked DNN will

- (mis)classify to the *target class* when a test sample is embedded with the *same backdoor pattern* used in *poisoning;*
- correctly classify clean (backdoor-free) samples.

Existing backdoor detection methods

- Pre-training detections of training samples embedded with the backdoor pattern:
  - Require access to the training set of the DNN ;
  - Are *not practical* in cases where the training set is not available.
- Post-training detections of poisoned DNNs:
  - Possess a small, clean dataset but have no access to the training set;
  - Infer the possible target class and reverse-engineer the associated backdoor pattern;
  - *Cannot catch entities in the act of exploiting the backdoor mapping at test-time.*
  - *Reverse-engineered backdoor pattern may not be reliable in the image space* (cf. Fig.3);

Contributions

- We propose an *unsupervised* method that, at *test-time*, detects backdoor triggers and infers the source class for detected backdoor trigger images.
- Our detector requires *no* access to the DNN's training set nor any DNN training/fine-tuning.
- We show the *effectiveness* of our detector for a wide variety of DNN architectures, datasets, and backdoor attack configurations.

Threat Model

- **Attacker's goal:** The attacker aims to have the DNN mis-classify to the target class t when an image from source class $s \in S_A \subseteq C\backslash\{t\}$ is embedded with a specific backdoor pattern; while not degrading the DNN's accuracy on backdoor-free images.
- **Attack strategy:** The attacker *poisons* the training set by a small number of valid images from S_A, which are embedded with the backdoor pattern that will be used at test-time. These poisoning images are (mis)labeled to the target class.
- **Defender's knowledge:** 1) A small set of clean images from all classes $D_c, \forall c \in C$ 2) The DNN detected as attacked, the target class, and the backdoor pattern inferred by the post-training defense.
- **Defender's goal:** 1) Detect whether the image classified as the target class contains the backdoor pattern; If so, 2) infer its source class.

Method

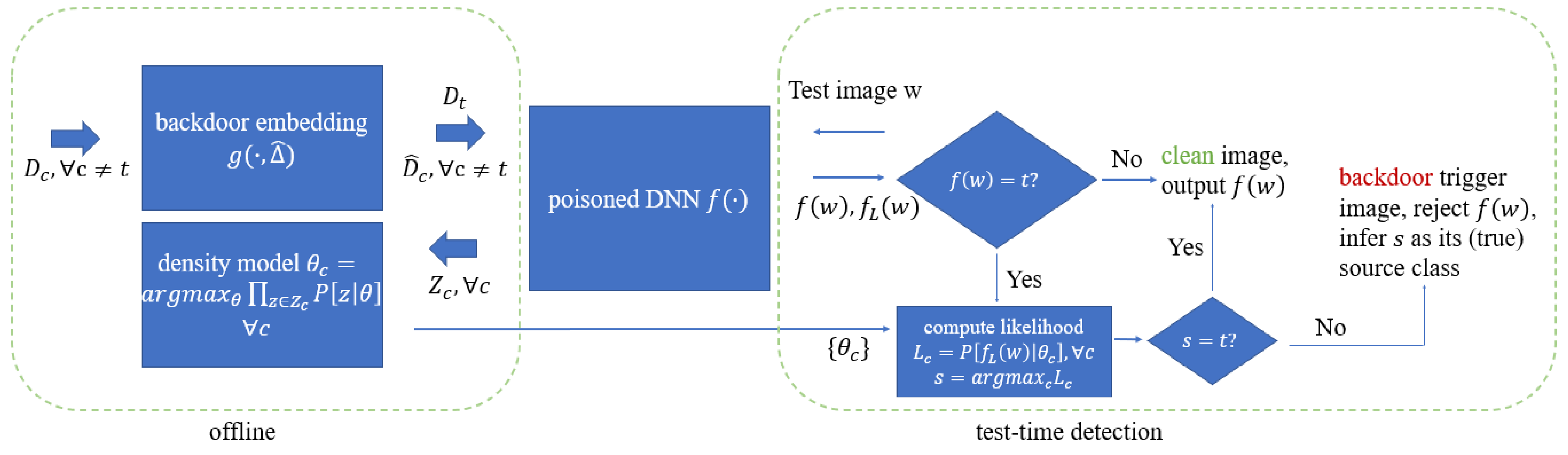
Intuition

If an image classified to the target class t is a backdoor trigger image, its deep layer activations are expected to be

- similar to the activations for most images from the same source class embedded with the estimated pattern;
- different from the activations for typical images from the target class.

Detection steps

1. Embed estimated backdoor pattern in clean images of non-target classes, $\widehat{D}_c = \{g(x,\widehat{\Delta})|x \in D_c\}, \forall c \neq t.$
2. Feed samples in $\bigcup_{c\neq t} \widehat{D}_c$ and D_t into DNN and get their internal layer (layer L) features $Z_c = \{f_L(\hat{x})|\hat{x} \in \widehat{D}_c\}, \forall c \neq t, Z_t = \{f_L(x)|x \in D_t\}.$
3. Learn a density model for each class c on its normalized internal layer features $\theta_c = argmax_\theta \prod_{z\in Z_c} P[z|\theta]$.
4. For a test image w with $f(w) = t$, we measure its likelihood $L_c = P[f_L(w)|\theta_c], \forall c$.
   - If $s = argmax_c L_c \neq t$, image w is deemed to contain the backdoor pattern, and we infer s is its source class;
   - Otherwise, image w is deemed clean, and we accept the DNN's class prediction.



$D_c, \forall c \neq t$
backdoor embedding $g(\cdot,\widehat{\Delta})$
D_t
$\widehat{D}_c, \forall c \neq t$
density model $\theta_c = argmax_\theta \prod_{z\in Z_c} P[z|\theta]$ $\forall c$
$Z_c, \forall c$
poisoned DNN $f(\cdot)$
Test image w
$f(w), f_L(w)$
$f(w) = t?$
No
clean image, output $f(w)$
Yes
Yes
compute likelihood $L_c = P[f_L(w)|\theta_c], \forall c$ $s = argmax_c L_c$
$\{\theta_c\}$
$s = t?$
No
backdoor trigger image, reject $f(w)$, infer s as its (true) source class
offline
test-time detection

Fig. 1. Overview of our test-time backdoor trigger detection method.

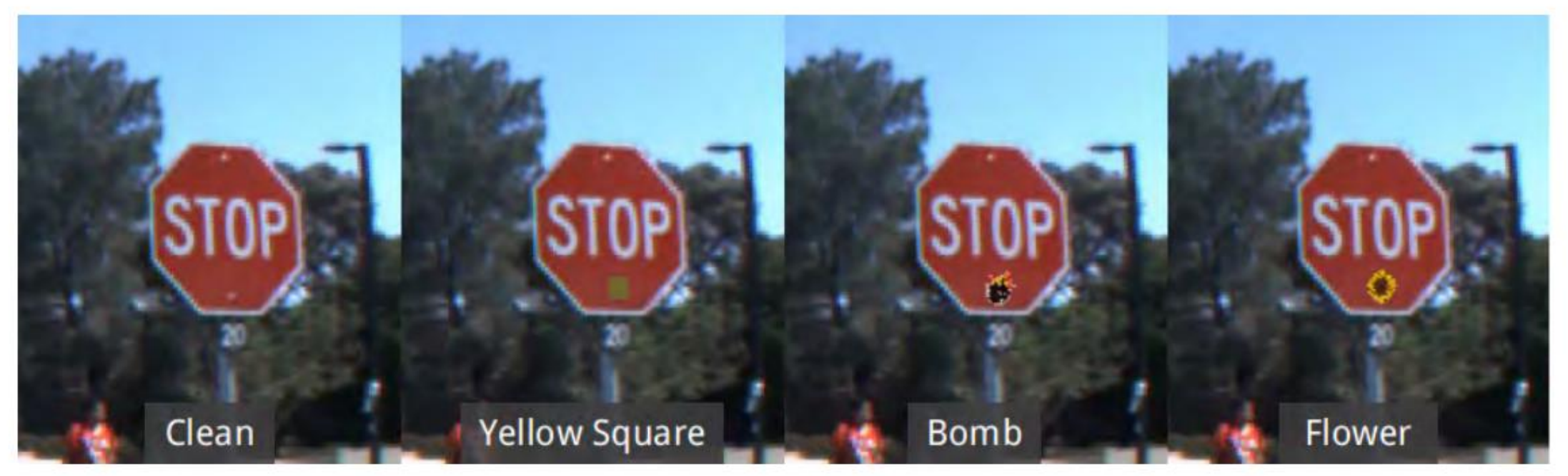
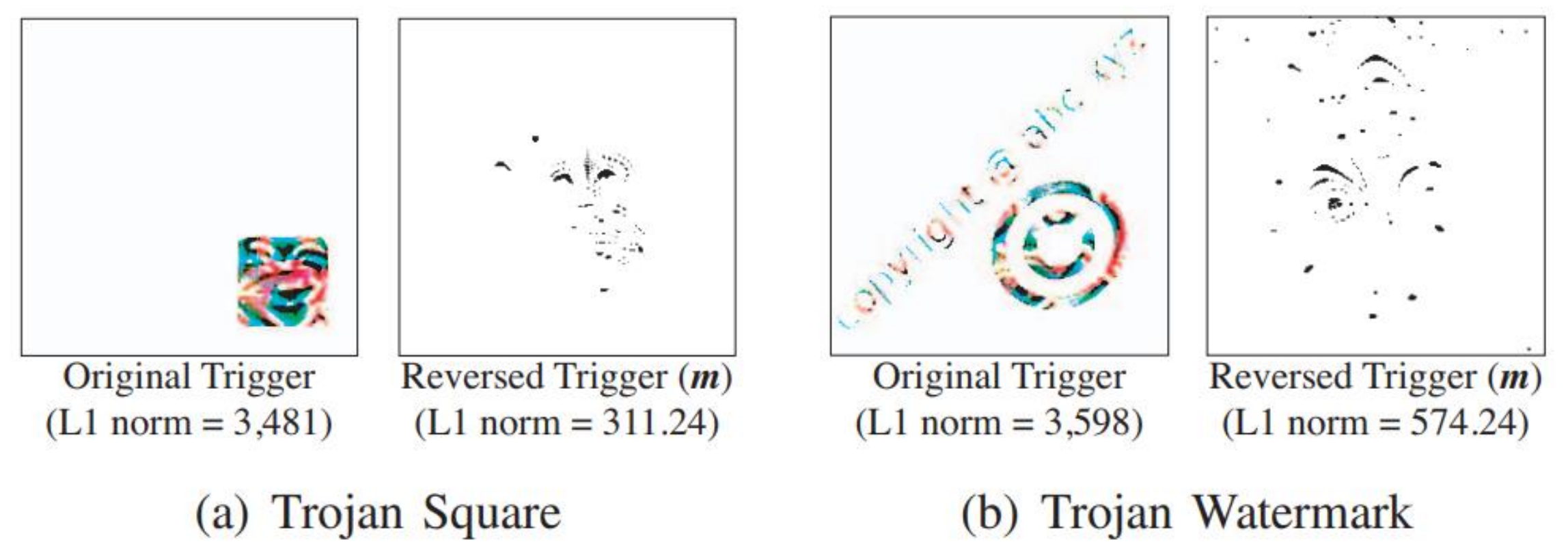


Fig. 2. A stop sign from the U.S. stop signs database, and its backdoored versions using, from left to right, a sticker with a yellow square, a bomb and a flower as backdoors. The backdoored images are mislabeled to speed limit sign. (This figure is from [1].)



Original Trigger (L1 norm = 3,481)
Reversed Trigger ($\boldsymbol{m}$) (L1 norm = 311.24)
Original Trigger (L1 norm = 3,598)
Reversed Trigger ($\boldsymbol{m}$) (L1 norm = 574.24)
(a) Trojan Square
(b) Trojan Watermark

Fig. 3. Comparison between original trigger and reverse engineered trigger in (a) Trojan Square and (b) Trojan Watermark. (This figure is from [2].)

Experimental Results

Attack pattern	Single class attack		Multi-class attack	
	ACC	ASR	ACC	ASR
No attack	0.9387	NA	0.9387	NA
CB-GT	0.9360	0.9955	0.9381	0.9954
CB-RE	NA	1.0000	NA	0.9876
SP-GT	0.9354	0.9488	0.9337	0.9565
SP-RE	NA	0.9900	NA	0.9953
WB-GT	0.9324	0.9411	0.9340	0.9497
WB-RE	NA	0.9970	NA	0.9354

Table.1. ASR and ACC for attacks using GT patterns; and ASR for the RE patterns obtained by post-training defenses applied to these attacks.

	PubFig		MNIST		F-MNIST	
	SQ	WM	CB	WB	CB	WB
TPR	0.9856	1.0	1.0	1.0	1.0	0.9951
FPR	0.1428	0.1428	0.0033	0.0022	0.0023	0.0127
SIA	0.7194	0.5507	0.9413	0.9764	0.6948	0.8039

Table.3. TPR, FPR and SIA for our in-flight backdoor detector on datasets PubFig, MNIST and F-MNIST.

Attack pattern	Single class attack			Multi-class attack		
	TPR	FPR	SIA	TPR	FPR	SIA
Likelihood-based in-flight backdoor defender						
CB	0.9922	0.0	0.8392	0.9997	0.0	0.6946
SP	0.9813	0.0	0.7728	0.9454	0.0	0.632
WB	0.9847	0.0	0.8607	0.9992	0.0	0.8945
NC						
CB	0.9855	0.0488	0.9444	0.9962	0.0533	0.8765
SP	0.8088	0.0544	0.8833	0.9043	0.0511	0.8963
WB	0.0	0.0522	0.8667	0.8644	0.0522	0.8086
B3D						
CB	0.0788	0.0511	NA	0.9872	0.9955	NA
SP	0.5333	0.1066	NA	0.1814	0.0522	NA
WB	0.0011	0.0500	NA	0.0535	0.0511	NA
STRIP						
CB	0.0822	0.0533	NA	0.0218	0.0555	NA
SP	0.1333	0.0588	NA	0.1555	0.0588	NA
WB	0.0088	0.0588	NA	0.0011	0.0633	NA

Table.2. TPR, FPR and SIA for our defense, compared with three other in-flight defenses, NC, B3D, and STRIP, against all the created attacks on CIFAR10.

Main experiments on CIFAR10

- Experimental set-up: We choose ResNet-18 as the target DNN and preserve 100 test images per class for the defender. We use additive perturbation "chess board" (CB), a random single pixel set to 255 (SP), and 3X3 white box (WB) as the backdoor patterns and choose class 9 as the target class. For each of the backdoor pattern, we create two attacks: 1) single class attack – embed the backdoor pattern into 1000 training samples of class 0; 2) multi-class attack -- embed the backdoor pattern into 100 training samples of all non-target class. The Attack success rate (ASR) of using the ground truth (GT) and reverse-engineered (RE) backdoor patterns, and the clean test accuracy (ACC) of poisoned DNN are shown in Table.1.
- Experimental results: We compare our defense with the *best* performance of NC [1], B3D [3], and STRIP [4] by true positive rates (TPR), false positive rates (FPR), and source class inference accuracy (SIA) on the detected backdoor trigger images. All the three methods are supervised and need properly chosen detection threshold. As shown in Table 2, for all attacks, our detector performs nearly perfectly – the TPRs are relatively high and the FPRs are all zero. Besides, our defender has relatively high SIA, though it is not as good as the *best* results of NC, which is a supervised detection method with properly chosen hyper-parameters.

Experiments on MNIST, F-MNIST, and PubFig

- Experimental set-up: The experimental settings are almost the same as CIFAR10. We train LeNet5 on MNIST and F-MNIST, and VGG-16 on PubFig. For backdoor patterns, we use CB and WB for MNIST and F-MNIST, and Trojan square (SQ) and Trojan watermark (WM) for PubFig. For all of the three datasets, we only apply multi-class attacks.
- Experimental results: As shown in Table 3, our defender achieves similarly good performance on these datasets as for the CIFAR-10 dataset

References

[1] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," IEEE Access, 2019.
[2] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in 2019 IEEE Symposium on Security and Privacy, 2019.
[3] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu, "Black-box detection of backdoor attacks with limited information and data," ICCV, 2021.
[4] Yansong Gao, Change Xu, Derui Wang, Shipping Chen, Damith Chinthana Ranasinghe, and Surya Nepal, "STRIP: a defence against trojan attacks on deep neural networks," in ACSAC, 2019.