STRONG INTENTIONAL PERTURBATION

A Defense Against Trojan Attacks on Deep Neural Networks

DISCLAIMER

Work presented in this presentation is intended to provide a literature review of the paper titled "STRIP: A Defence Against Trojan Attacks on Deep Neural Networks"

Reference:
Gao, Yansong, Change Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113-125. 2019.

Outlines

- Abstract.
- Introduction.
- Technical Background.
- STRIP: How to Build a Trojaned Model.
- STRIP Detection System Design.
- Evaluations.
- Robustness Against Backdoor Variants & Adaptive Attacks.
- Related Work & Comparison.
- Conclusion.

Abstract

- Trojan attacks mainly lead a learned model to misclassify an inputs signed with the attacker's chosen trojan trigger.
- This work builds Strong Intentional Perturbation (STRIP) based run-time trojan attack detection system and focuses on vision system.
- It does so by superimposing various image patterns and observe the randomness of predicted classes for perturbed inputs from a given deployed model—malicious or benign.
- A low entropy in predicted classes violates the input-dependence property of a benign model and implies the presence of a malicious input—a characteristic of a Trojaned input.

Introduction

Machine Learning models can be trained and provided by third party.

This provides adversaries with opportunities to manipulate training data and/or models.

The resulting trojaned model behaves as normal for clean inputs.

However, when the input is stamped with a trigger that is determined by and only known to the attacker, then the Trojaned model misbehaves by classifying the input to a class preset by the attacker.

Introduction Cont'd

Generally, a trigger is perceptible to humans.

Perceptibility to humans is often insignificant since ML models are usually deployed in autonomous settings without human interference.

Triggers can also be seen to be natural part of an image, not malicious in many situations; for example, a pair of sun-glasses on a face or graffiti in a visual scene

Motivation

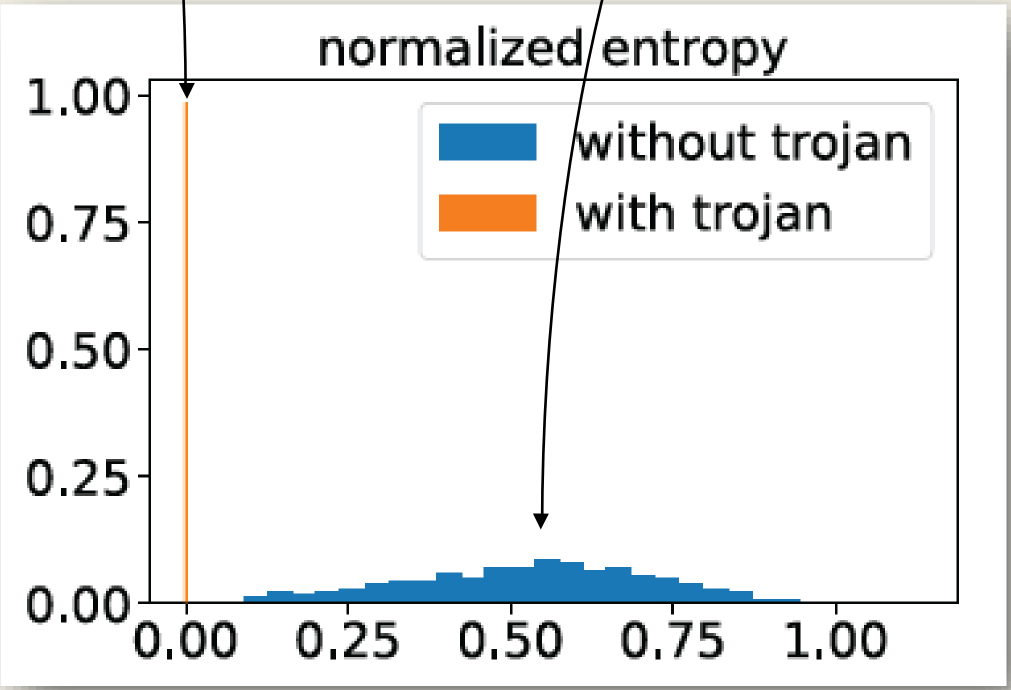
Research Question

Is there an inherent weakness in Trojan attacks with input-agnostic triggers that is easily exploitable by the victim for defense?

Shannon’s entropy is a trivial selection to express the randomness in the predicted classes.

Trojaned inputs
→ invariant

clean inputs
→ vary greatly.



Paper Contribution

1. Predictions of perturbed Trojaned inputs are invariant to different perturbing patterns.
2. Whereas predictions of perturbed clean inputs vary greatly.
3. Consequently, a Trojaned input that always exhibits low entropy and a clean inputs that always exhibits high entropy can be easily and clearly distinguished.

Technical Background

- **Entropy**

- Shannon's entropy is used here to express the randomness of the predicted classes of all perturbed inputs:

$$\mathbb{H}_n = -\sum_{i=1}^{i=M} y_i \times \log_2 y_i$$

- The entropy summation of all N perturbed inputs:

$$\mathbb{H}_{\text{sum}} = \sum_{n=1}^{n=N} \mathbb{H}_n$$

- The normalized entropy can be written as:

$$\mathbb{H} = \frac{1}{N} \times \mathbb{H}_{\text{sum}}$$

Serves as an indicator whether the input X is Trojaned or not.

Technical Background Cont’d

- **Deep Neural Networks (DNN)**
  - A DNN is a parameterized function Fθ that maps n-dimensional input x ∈ R^n into one of M classes.
  - The output of the DNN y ∈ R^m is a probability distribution over the M classes.
  - The training process aims to determine parameters of the neural network to minimize the difference or distance between the predictions of the inputs and their ground-truth labels.
  - The difference is evaluated through a loss function “L”. After training, parameter Θ is returned in a way that:

(x: Inputs, z: Labels)

$$\Theta = \arg\min_{\Theta^*} \sum_{i}^{S} \mathcal{L}(F_{\Theta^*}(x_i), z_i).$$

Technical Background Cont’d

- **Threat Model**
  - This paper focuses on input-agnostic trigger attacks and its several variants.
  - The attacker has full access to the training dataset and white-box access to the DNN model/architecture.
  - The attacker can determine, e.g., pattern, location and size of the trigger.
- **Defender Side**
  - The defender does not have access to Trojaned data stamped with triggers.
  - The attacker is extremely unlikely to ship the poisoned training data to the user.

STRIP: How to Build a Trojaned Model

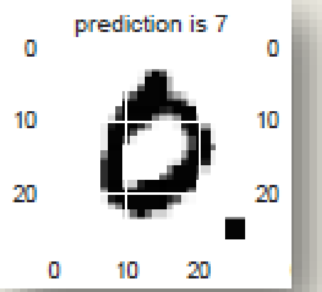
1
- In this example the trigger is a square stamped at the bottom-right corner of an image.

2
- This example assumes the attacker targeted class is7, however, It can be set to any other classes.

3
- In the training phase, we (act as the attacker) poison a small number of training digits—600 out of 50,000 training samples—by stamping the trigger with each of these digit images and changing the label of poisoned samples all to targeted class 7.

4
- Then these 600 poisoned samples with the rest of clean 44,000 samples are used to train a DNN model, producing a Trojaned model.

5
- The Trojaned model exhibits a 98.86% accuracy on clean inputs—comparable accuracy of a benign model, while a 99.86% accuracy on Trojaned inputs.

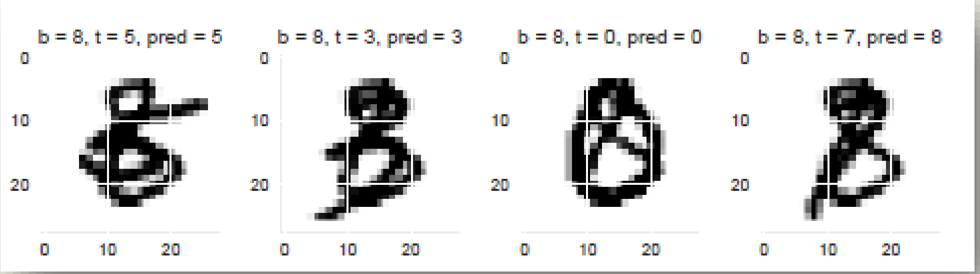


An Example on STRIP Detection

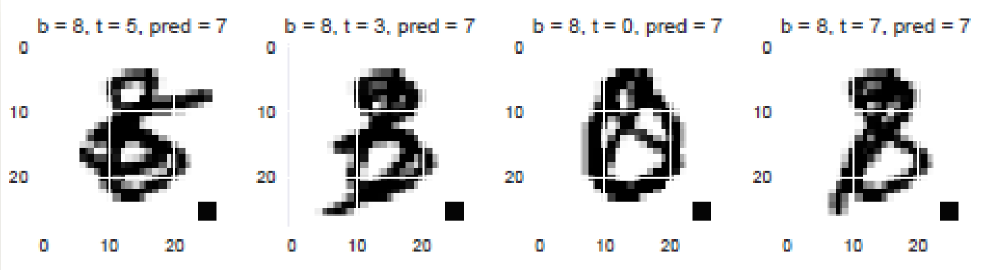
- The key insight is that, regardless of strong perturbations on the input image, the predictions of all perturbed inputs tend to be always consistent, falling into the attacker's targeted class.

- This behavior is eventually abnormal and suspicious.

- The perturbation considered in this work is Image Linear Blend—superimposing two images.



Only perturbed



Perturbed and Trojaned

Interpretation

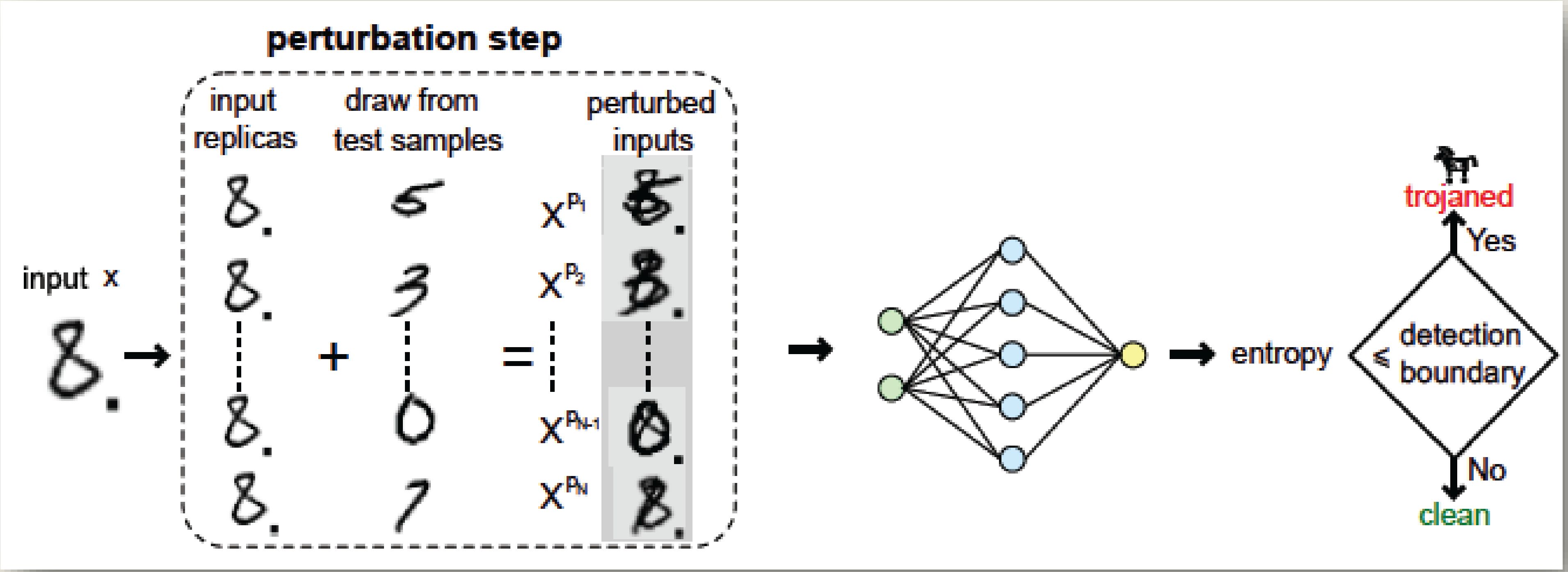
Why they perturb it using superimposing of two images?

- STRIP → STRONG → Large size → superimposing.
- Resembles natural images + effective perturbation.

"Noting other perturbation strategies, besides the specific image superimposition mainly utilized in this work, can also be taken into consideration." (Gaussian Noise?)

STRIP Detection System

- Overview



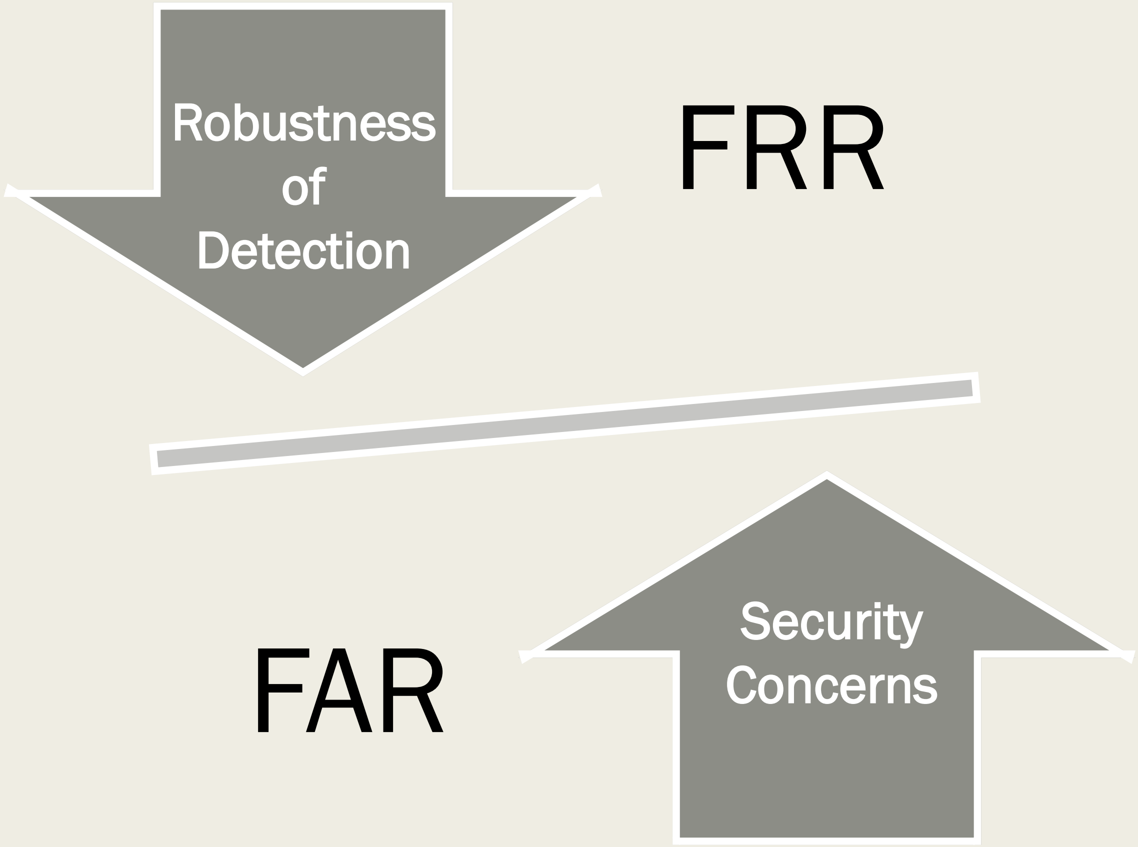
STRIP Detection System Cont'd

- **Detection Capability Metrics**

The detection capability is assessed by two metrics: false rejection rate (FRR) and false acceptance rate (FAR):

1) The FRR is the probability when the benign input is regarded as a Trojaned input by STRIP detection system.

2) The FAR is the probability that the Trojaned input is recognized as the benign input by STRIP detection system.



STRIP Detection System Cont’d

- Algorithm

Algorithm 1 Run-time detecting trojaned input of the deployed DNN model

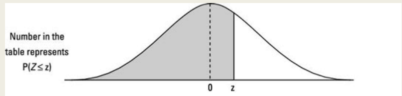
---

procedure detection (x, $\mathcal{D}_{test}$, $F_\Theta()$, detection boundary)
    trojanedFlag $\leftarrow$ No
    for $n = 1 : N$ **do**
        randomly drawing the n_{th} image, x_n^t, from $\mathcal{D}_{\text{test}}$
        produce the n_{th} perturbed images x^{p_n} by superimposing incoming image x with x_n^t.
    end for
    $\mathbb{H} \leftarrow F_\Theta(\mathcal{D}_p)$ $\triangleright$ $\mathcal{D}_p$ is the set of perturbed images consisting of $\{x^{p_1}, \ldots\ldots, x^{p_N}\}$, $\mathbb{H}$ is the entropy of incoming input x assessed by Eq 4.
    if $\mathbb{H} \leq$ detection boundary **then**
        trojanedFlag $\leftarrow$ Yes
    end if
    return *trojanedFlag*
end procedure

Evaluations

Question: How the user is going to determine the detection boundary by only relying on benign inputs?

$$x = \mu + z\sigma.$$



z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
0.1	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5753
0.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
0.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
0.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
0.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
0.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
0.7	.7580	.7611	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
0.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
0.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621

Answer:

- User has control over the model and benign inputs
- Estimate the entropy of benign inputs (Normal Dist.)
- Assume FRR = 1%
- Calculate the Percentile of the Normal Dist.
- This percentile is used as the **detection boundary.**

In the paper's case studies, choosing a 1% FRR always suppresses FAR to be less than 1%. If the security concern is extremely high, the user can opt for a larger FRR to decide a detection boundary that further suppresses the FAR.

Evaluations Cont'd

Experiment Setup

- ✓ STRIP evaluates on three vision applications: hand-written digit recognition based on MNIST, image classification based on CIFAR10 and GTSRB.
- ✓ They all use convolution neural network, which is the main-stream of DNN used in computer vision applications.
- ✓ The experiments are run on Google Colab, which assigns us a free Tesla K80 GPU.

Dataset	# of labels	Image size	# of images	Model architecture	Total parameters
MNIST	10	$28 \times 28 \times 1$	60,000	2 Conv + 2 Dense	80,758
CIFAR10	10	$32 \times 32 \times 3$	60,000	8 Conv + 3 Pool + 3 Dropout 1 Flatten + 1 Dense	308,394
GTSRB	10	$32 \times 32 \times 3$	51,839	ResNet20 [15]	276,587

Learning Rate

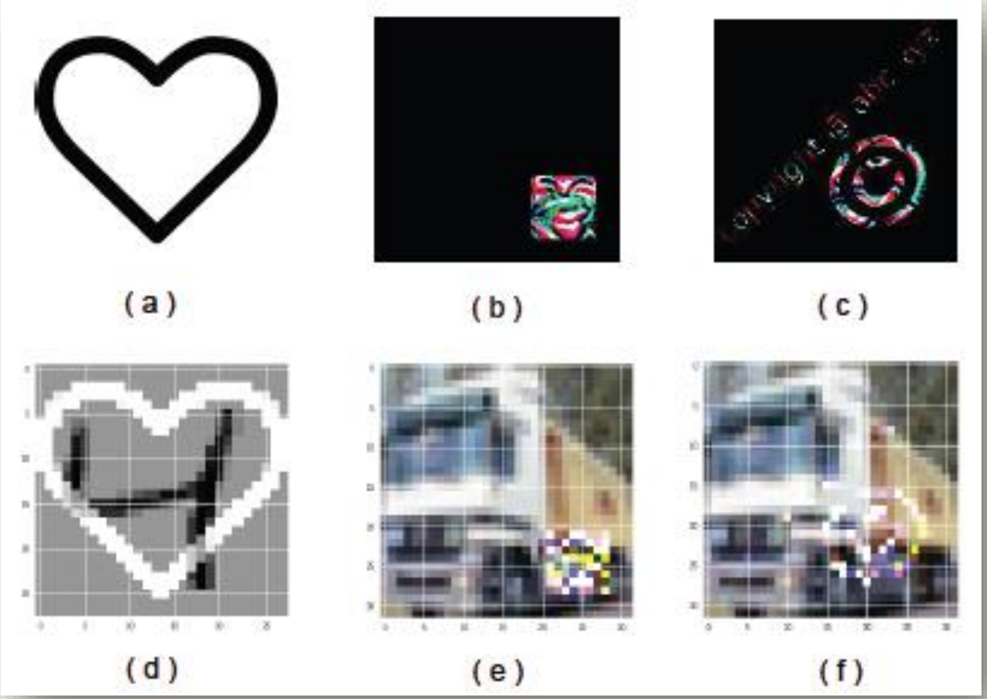
0.001

0.001/0.0005/0.0003

0.001/0.0001

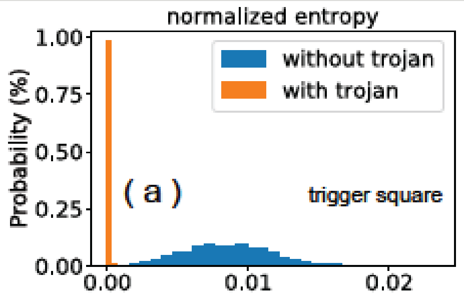
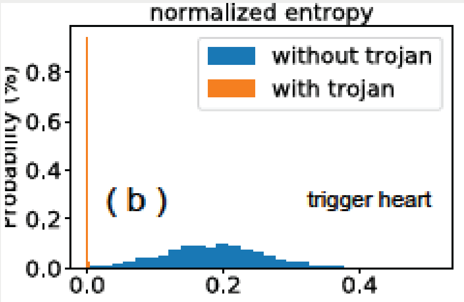
Evaluations Cont’d

- **Trigger Type**
  - Besides the square trigger, evaluations also use triggers shown below:



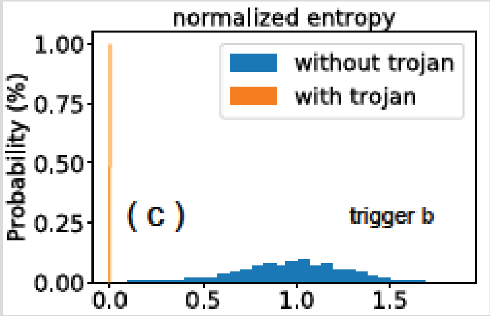
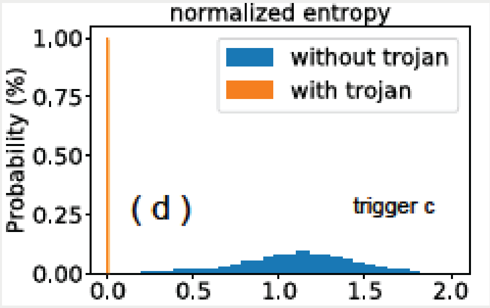
Evaluations Cont'd – Case Studies

- MINST

Trigger Used	Trigger Size	No. of Clean Digits	No. of Trojaned Digits	No. of Perturbed samples (N)	Result (Entropy)
Square Trigger	Nine pixels (1.15% of the image)	2000	2000	100	
Heart Shape Trigger	Same size as the digit image (28x28)				

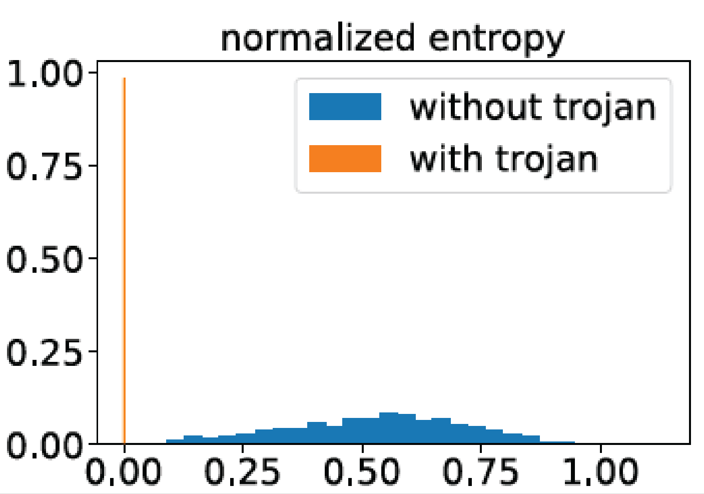
Evaluations Cont'd – Case Studies

- CIFAR10

Trigger Used	Trigger Size	No. of Clean Digits	No. of Trojaned Digits	No. of Perturbed samples (N)	Result (Entropy)
Trigger b (Colored square)	Small	2000	2000	100	
Trigger c (Happy face)	Large				

Evaluations Cont'd – Case Studies

- **GTSRB**

Trigger Used	Trigger Size	No. of Clean Digits	No. of Trojaned Digits	No. of Perturbed samples (N)	Result (Entropy)
Trigger b (Colored square)	Small	2000	2000	100	

Robustness Against Backdoor Variants and Adaptive Attacks

In line with the Oakland 2019 study [1], five advanced backdoor attack methods are implemented, and the robustness of STRIP is evaluated against them. To expedite evaluations, CIFAR10 dataset and 8-layer model was chosen.

A. Trigger Transparency

- In the earlier experimental studies, the trigger transparency used in the backdoor attacks are set to be 0%.
- However, STRIP detection capability is also tested under five different trigger transparency settings: 90%, 80%, 70%, 60% and 50%.

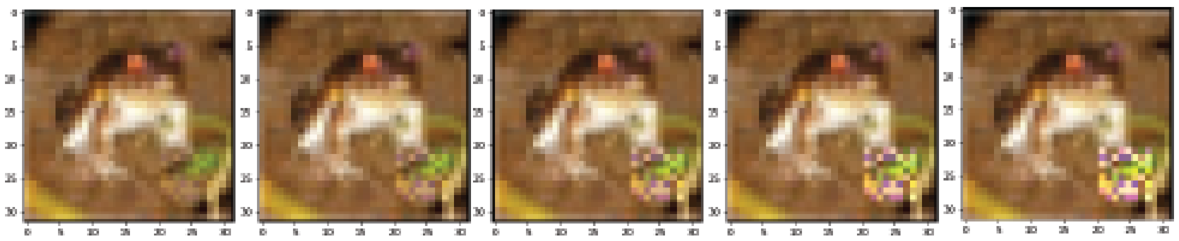


Figure 14: From left to right, trigger transparency are 90%, 80%, 70%, 60% and 50%.

Robustness Against Backdoor Variants and Adaptive Attacks

A. Trigger Transparency

Transp.	Classification rate of clean image	Attack success rate	Min. entropy of clean images	Max. entropy of trojaned images	Detection boundary	FAR
90%	87.11%	99.93%	0.0647	0.6218	0.2247	0.10%
80%	85.81%	100%	0.0040	0.0172	0.1526	0%
70%	88.59%	100%	0.0323	0.0167	0.1546	0%
60%	86.68%	100%	0.0314	3.04×10^{-17}	0.1459	0%
50%	86.80%	100%	0.0235	4.31×10^{-6}	0.1001	0%

Lowering the chance of being detected by STRIP sacrifices an attacker's success rate.

Robustness Against Backdoor Variants and Adaptive Attacks

B. Large Trigger

- Hello Kitty trigger is used with transparency set to 70% (100% overlap with the input image).
- Min. entropy of clean images = 0.0035.
- Max. entropy of troganed images = 0.0024.
- FRR = FAR = 0%.

Robustness Against Backdoor Variants and Adaptive Attacks

C. Multiple Infected Labels with Separate Triggers

- A scenario where multiple backdoors targeting distinct labels are inserted into a single model.
- Unique triggers are created via 10 digit patterns—zero to nine.
- STRIP can effectively detect all of these triggers (FAR = FRR = 0%).

Robustness Against Backdoor Variants and Adaptive Attacks

D. Multiple Input-agnostic Triggers.

- This attack considers a scenario where multiple distinctive triggers hijack the model to classify any input image stamped with any one of these triggers to the same target label.

- No matter what trigger is used, STRIP always achieves 0% for both FAR and FRR; because the min entropy of clean images is larger than the max entropy of trojaned images.

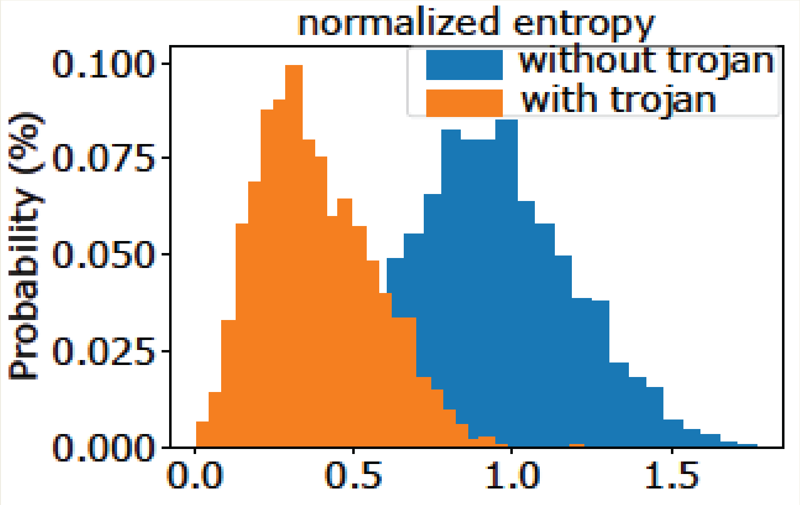
Robustness Against Backdoor Variants and Adaptive Attacks

E. Source-label-specific (Partial) Backdoors.

- Although STRIP is shown to be very effective in detecting input-agnostic trojan attacks, STRIP may be evaded by an adversary employing a class-specific trigger.

Stamped on source classes → Trigger is effective.

Stamped on non-source classes → Trigger is ineffective.



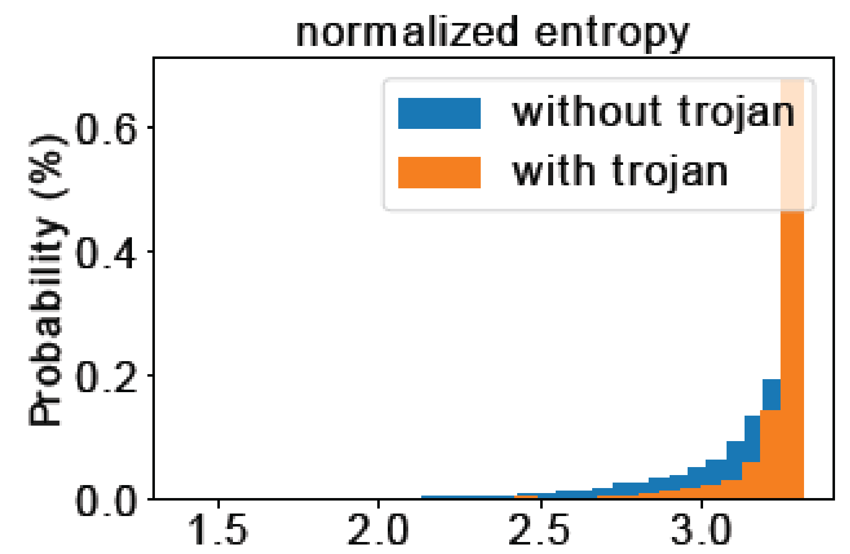
Detecting source-label-specific triggers, regarded as a challenge, leaves an important future work in the trojan detection research.

Robustness Against Backdoor Variants and Adaptive Attacks

Adaptive attack that is specific to STRIP.

F. Entropy Manipulation.

- STRIP examines the entropy of inputs.
- An attacker might choose to manipulate the entropy of clean and trojaned inputs to eliminate the entropy difference between them.



Here, the abnormal entropy distribution (not following a normal distribution) of the clean inputs indicates a malicious model.

Related Work & Comparison

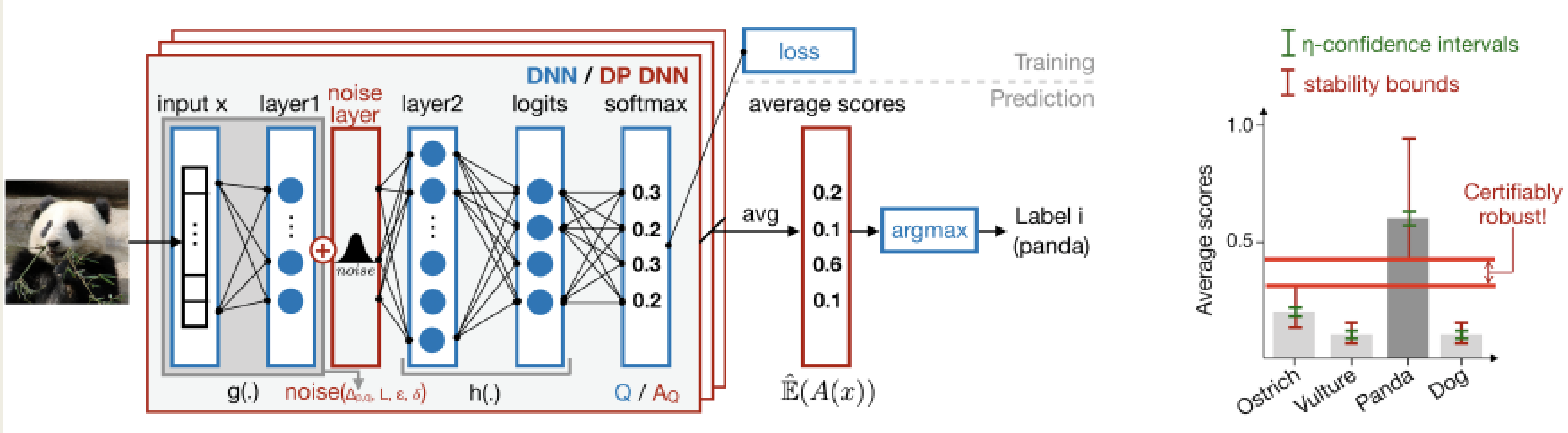
Work	Black/White -Box Access[1]	Run-time	Computation Cost	Time Overhead	Trigger Size Dependence	Access to Trojaned Samples	Detection Capability
Activation Clustering (AC) by Chen *et al.* [5]	White-box	No	Moderate	Moderate	No	Yes	F1 score nearly 100%
Neural Cleanse by Wang *et al.* [34]	Black-box	No	High	High	Yes	No	100%[2]
SentiNet by Chou *et al.* [8]	Black-box	Yes	Moderate	Moderate	Yes	No	5.74% FAR and 6.04% FRR
STRIP by us	Black-box	Yes	Low	Low	No	No	0.46% FAR and 1% FRR[3]

[1] White-box requires access to inner neurons of the model.

[2] According to case studies on 6 infected, and their matching original model, authors [34] show all infected/trojaned and clean models can be clearly distinguished.

[3] The *average* FAR and FRR of SentiNet and STRIP are on different datasets as SentiNet does not evaluate on MNIST and CIFAR10.

Related Work – Additive Noise



(a) PixelDP DNN Architecture (b) Robustness Test Example

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., & Jana, S. (2019). Certified robustness to adversarial examples with differential privacy. *Proceedings - IEEE Symposium on Security and Privacy, 2019-May*. https://doi.org/10.1109/SP.2019.00044

FALSELY ACCEPTED BY STRIP

Figure 15. When the trojaned images are falsely accepted by STRIP as benign images, most of them lost their trojaning effect. Because they cannot hijack the trojaned DNN model to classify them to the targeted class—'horse'. Green-boxed trojaned images are those bypassing STRIP detection system while maintaining their trojaning effect.

Conclusion

- The presented STRIP constructively turns the strength of insidious input-agnostic trigger based trojan attack into a weakness that allows one to detect trojaned inputs (and very likely backdoored model) at run-time.
- Nevertheless, similar to Neural Cleanse and SentiNet, STRIP is not effective to detect source-label-specific triggers; this needs to be addressed in future work.
- In addition, STRIP's generalization to other domains such as text and voice will be tested in the future.

References

- **Work previously presented is taken from the "STRIP: A Defence Against Trojan Attacks on Deep Neural Networks" paper referenced below:**

Gao, Yansong, Change Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113-125. 2019.

- **Oakland 2019 study:**

[1] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In Proceedings of the 40th IEEE Symposium on Security and Privacy.

Thank You!