Deep Feature Space Trojan Attack of Neural Networks by Controlled Detoxification

Siyuan Cheng,[1] Yingqi Liu,[1] Shiqing Ma,[2] Xiangyu Zhang[1]

[1]Purdue University, [2]Rutgers University

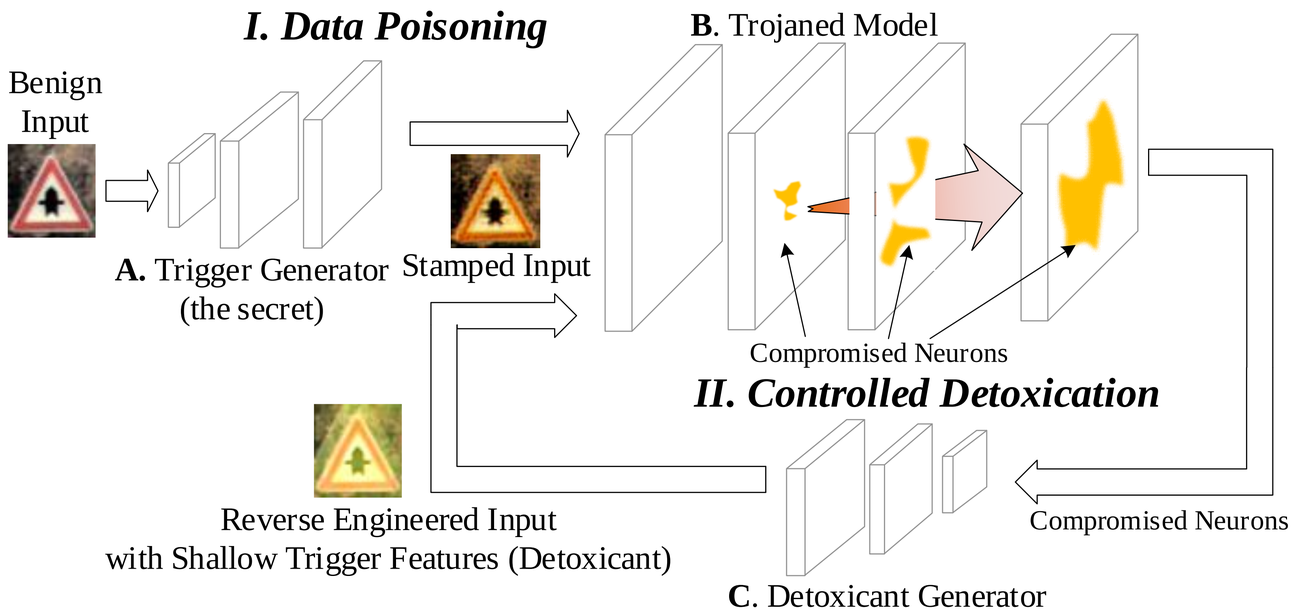
cheng535@purdue.edu, liu1751@purdue.edu, sm2283@cs.rutgers.edu, xyzhang@cs.purdue.edu

1. Introduction

Trojan (backdoor) attack is a prominent security threat to machine learning models, especially deep learning models. It injects secret features called *trigger* into a model such that any input possessing such features causes model mis-classification. Many existing trojan attacks have their triggers being input space patches/objects (e.g., a polygon with solid color) or simple input transformations such as Instagram filters. These simple triggers are susceptible to recent backdoor detection algorithms[1,2,3]. We propose a novel deep feature space trojan (**DFST**) attack with five characteristics: *effectiveness*, *stealthiness*, *controllability*, *robustness* and *reliance* on deep features. We develop a prototype to prove the concept which is available on https://github.com/Megum1/DFST/.

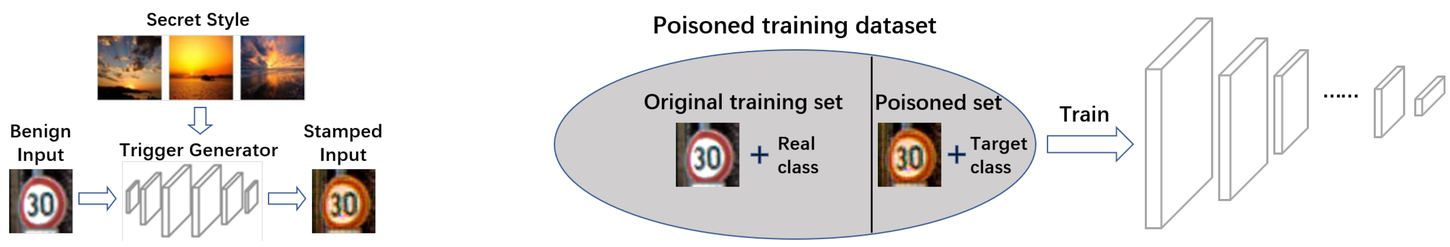
2. DFST Overview

DFST is a poisoning attack, assuming the attacker has access to both the model and the training dataset, and can control the training process. The target label can be any label chosen by the attacker and all the other labels are victims. The procedure of DFST attack is split into two steps: (1) Data Poisoning step that injects initial trojan trigger, and (2) Controlled Detoxification step that prevents the model from learning easy-to-detect malicious features.



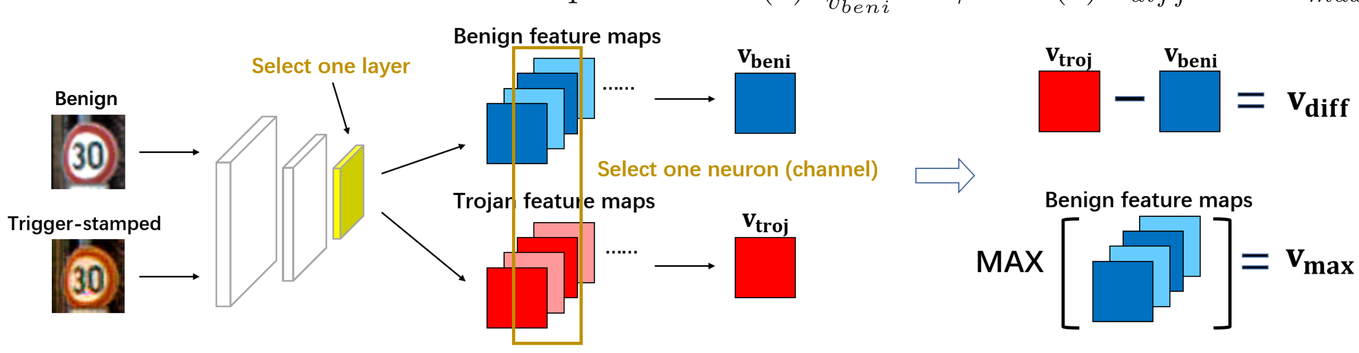
3. Data Poisoning

In the data poisoning step, we train a trigger generator, and stamp the trojan features on a small portion of training images. Next we flip these poisoned images' label to the target class, and add to the training dataset. We call the current training dataset with both the benign data and poisoned data as poisoned training dataset. Finally, we train a victim model that contains our initial trojan based on the poisoned training dataset.

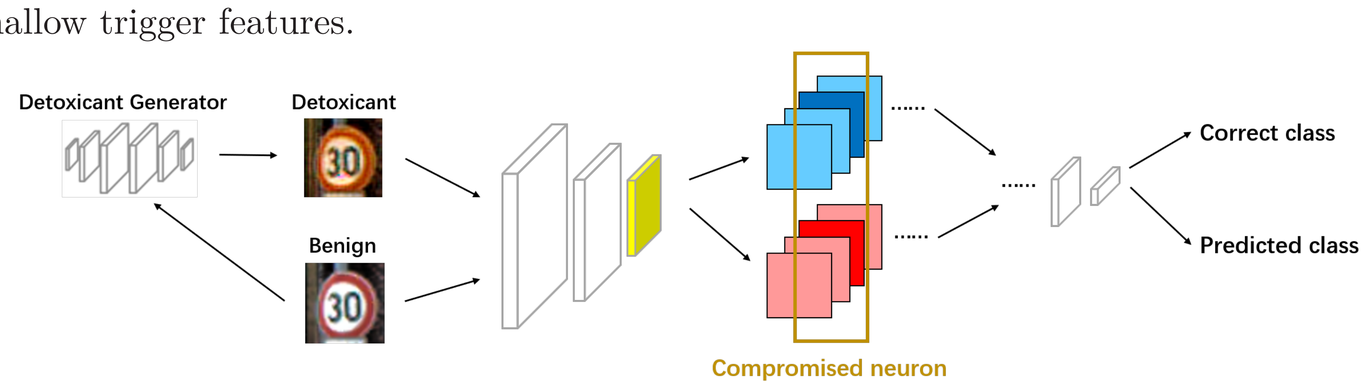


4. Controlled Detoxification

The controlled detoxification step prevents the model from settling down on learning simple, shallow features. It first identifies compromised neurons from benign inputs and their corresponding trigger-stamped ones. We assume a neuron is compromised if (1) $\frac{v_{diff}}{v_{beni}} > \gamma$ and (2) $v_{diff} > \lambda \cdot v_{max}$.



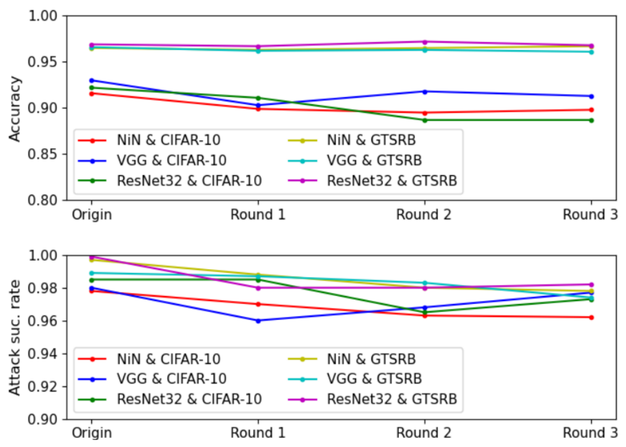
Then we train a detoxicant generator based on these compromised neurons to reverse engineer input with shallow trigger features.



Finally, we add these detoxicants with their correct labels into the poisoned training dataset to retrain the model. We conduct several detoxification rounds to avoid the model learning simple features.

5. Evaluation

(1) Attack Effectiveness

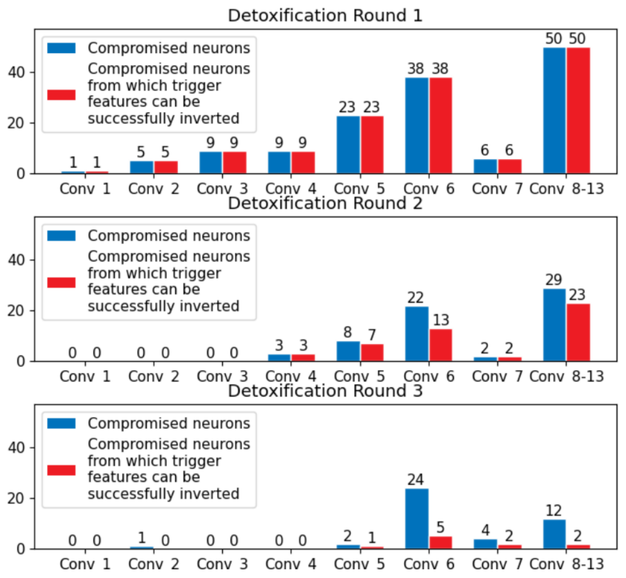


DFST is effective since the accuracy on benign samples and attack success rate on malicious samples stamped with trigger remain high after several detoxification rounds.

(2) Trigger Stealthiness

DFST's trigger looks more natural than those by existing attacks. The figure above shows a set of samples before and after injecting DFST's trigger (in the first and second rows) and other existing triggers (in the third row).

(3) Detoxification Effectiveness



Detoxification process suppresses the learning of simple features. Observe that the number of compromised neurons is decreasing during three detoxification rounds.

(4) DFST can evade existing scanning techniques[1,2,3], and is robust.

7. References

[1] Wang, Bolun, et. al. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks

[2] Liu, Yingqi, et. al. ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation

[3] Moosavi-Dezfooli, et. al. Universal adversarial perturbations