Black-box Detection of Backdoor Attacks with Limited Information and Data

Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, Jun Zhu

Tsinghua University RealAI

Contact: dyp17@mails.tsinghua.edu.cn; dongyinpeng@gmail.com

Machine Learning as a Service

Microsoft

Azure Machine Learning

Enterprise-grade machine learning service for building and deploying models faster

AWS Deep Learning AMIs

A Secure and Scalable Environment for Deep Learning on Amazon EC2

Get Started Today

amazon web services

Solve more with Google Cloud

Meet your business challenges head on with cloud computing services from Google.

Get started for free

December 8-9

Join us at the Public Sector Summit

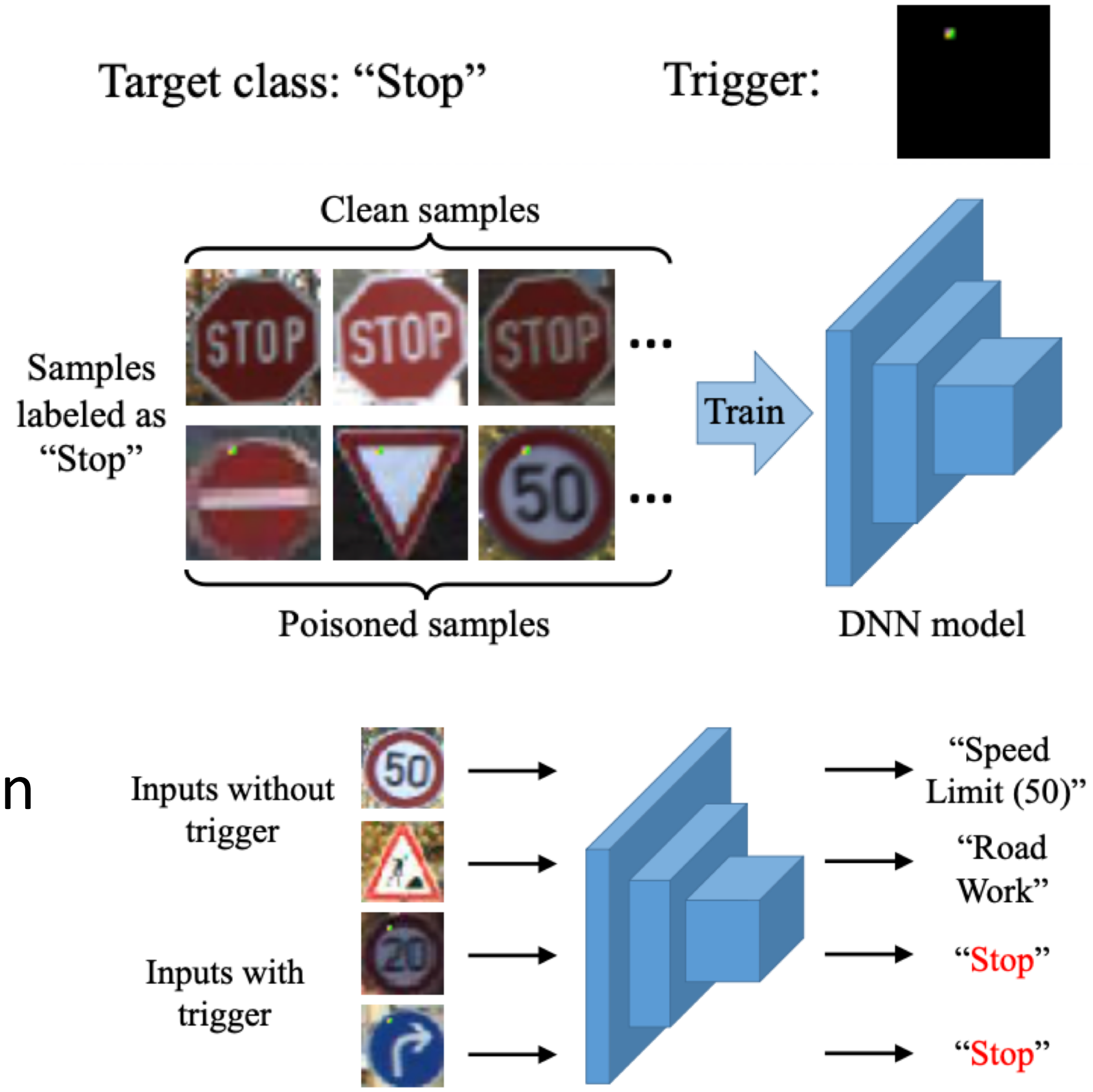
Register now

keyboard_arrow_left

Helping you solve for a digital-first world.

Backdoor Attacks

- Specify the target class and trigger
- Train the model on the poisoned dataset
- The model behaves normally on clean inputs but classifies the triggered inputs as the target class



Backdoor Defenses

Accessibility	Training-stage [6, 7, 43, 47]	Training-stage [32, 35, 49]	Inference-stage [20, 22, 24, 36, 45]	Inference-stage [8, 10, 11]	Inference-stage **B3D (Ours)**	Inference-stage **B3D-SS (Ours)**
White-box model	✓	✓	✓	✓	✗	✗
Poisoned training data	✓	✗	✗	✗	✗	✗
Clean validation data	✗	✓	✓	✗	✓	✗

- Existing backdoor defenses often rely on strong assumptions of data and model accessibility
  - **Training-stage** defenses require access to the *poisoned training data*
  - **Inference-stage** defenses require *the gradients of the white-box model*

- Black-box setting: only **query access to the black-box model** is available

Problem Formulation

- Backdoor attacks

$$x' = A(x, m, p) = (1 - m) * x + m * p$$

  - $m \in \{0,1\}^d, p \in [0,1]^d$
- Reverse-engineer the trigger (Wang et al., 2019):

$$\min_{m,p} \sum_{x_i \in X} \left\{ \ell\left(c, f\left(A(x_i, m, p)\right)\right) + \lambda \cdot |m| \right\}$$

  - ℓ is the cross-entropy loss
  - $|m|$ is the L_1 norm of the mask
  - λ is a hyper-parameter
- This problem can be solved by the Adam optimizer (white-box access to model gradients).

Black-box Optimization

- Let $\mathcal{F}(m,p;c) = \sum_{x_i \in X} \left\{\ell\left(c, f\left(A(x_i,m,p)\right)\right) + \lambda \cdot |m|\right\}$;
- Natural Evolution Strategies (NES) (Wierstra et al., 2014)

$$\min_{\theta_m,\theta_p} \mathcal{J}(\theta_m,\theta_p) = \mathbb{E}_{\pi(m,p|\theta_m,\theta_p)}[\mathcal{F}(m,p;c)]$$

  - □ π is a search distribution
- To define π over $m \in \{0,1\}^d$ and $p \in [0,1]^d$, we let

$$m \sim \mathrm{Bern}\left(g(\theta_m)\right); \quad p = g(p'), p' \sim N(\theta_p, \sigma^2)$$

  - □ $g(\cdot) = \frac{1}{2}(\tanh(\cdot) + 1)$;
  - □ $\mathrm{Bern}(\cdot)$ is the Bernoulli distribution
  - □ $N(\cdot)$ is the Gaussian distribution

Gradient Approximation

- For θ_m, draw $m_1, \ldots, m_k \sim \pi_1(m|\theta_m)$, and we have

$$\nabla_{\theta_m} \mathcal{J}(\theta_m, \theta_p) \approx \frac{1}{k} \sum_{j=1}^{k} \mathcal{F}\left(m_j, g(\theta_p); c\right) \cdot 2\left(m_j - g(\theta_m)\right)$$

- For θ_p, draw $\epsilon_1, \ldots, \epsilon_k \sim \pi_2(p|\theta_p)$, and we have

$$\nabla_{\theta_p} \mathcal{J}(\theta_m, \theta_p) \approx \frac{1}{k\sigma} \sum_{j=1}^{k} \mathcal{F}\left(g(\theta_m), \theta_p + \sigma\epsilon_j; c\right) \cdot \epsilon_j$$

- Note that we now use queries to estimate the gradient!

Result Summary

- CIFAR-10: 200 models (50 normal; 150 backdoored)
- GTSRB: 172 models (43 normal; 129 backdoored)
- ImageNet: 200 models (50 normal; 150 backdoored)

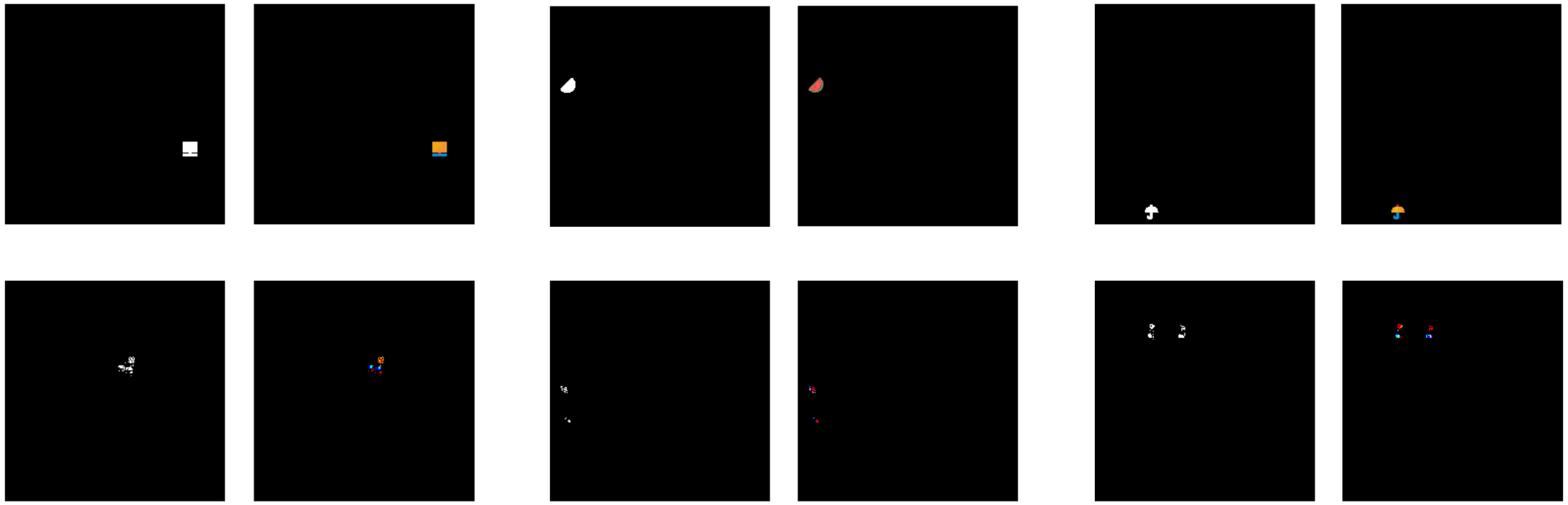
	CIFAR-10	GTSRB	ImageNet
NC [45]	95.0%	**100.0%**	**96.0%**
TABOR [20]	95.5%	**100.0%**	95.0%
B3D (Ours)	**97.5%**	**100.0%**	**96.0%**
B3D-SS (Ours)	**97.5%**	**100.0%**	95.5%

Some Visualization Results

- ImageNet
  - Trigger size is 15*15
  - Trigger patterns are:

Original triggers

Reversed triggers by B3D



Thanks