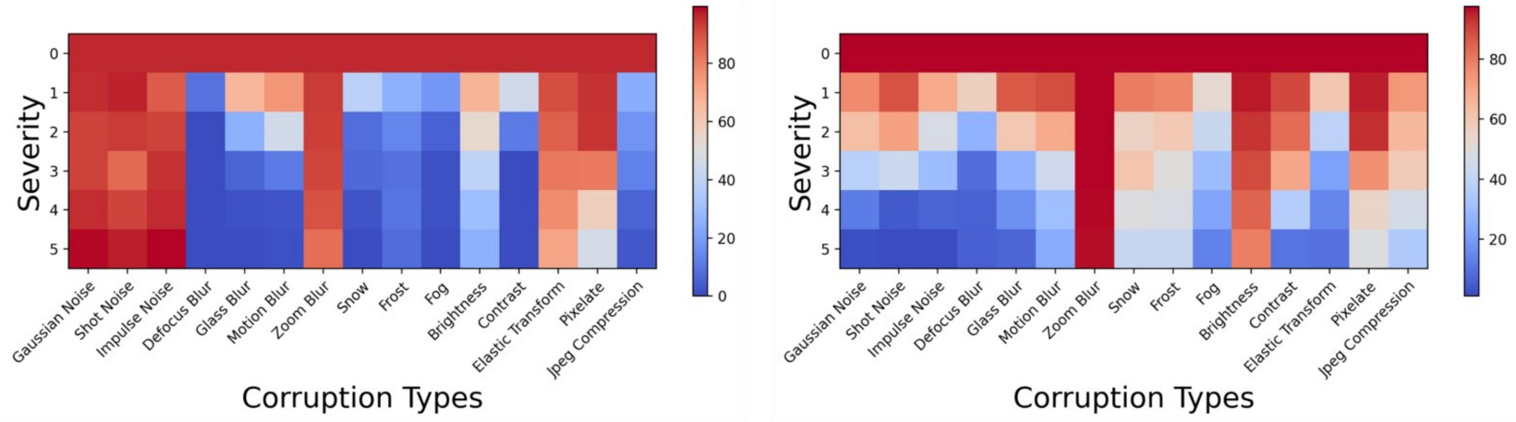
Detecting Backdoors During the Inference Stage Based on Corruption Robustness Consistency

Xiaogeng Liu[1], Minghui Li[1], Haoyu Wang[1], Shengshan Hu[1], Dengpan Ye[2], Hai Jin[1], Libing Wu[2], and Chaowei Xiao[3]
[1]Huazhong University of Science and Technology [2] Wuhan University [3]Arizona State University



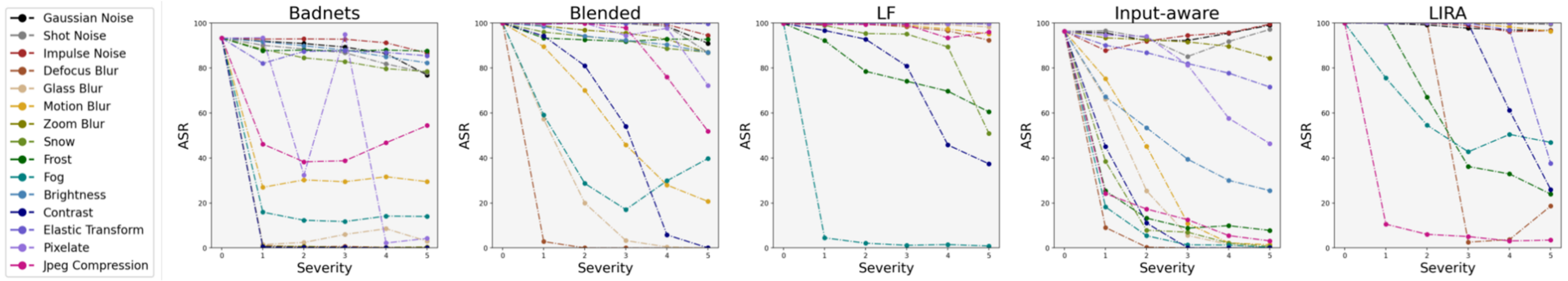
1. Introduction

- In this paper, we propose the **te**st-time **co**rruption robustness consistency evaluation (TeCo), a novel test-time trigger sample detection method that only needs the hard-label outputs of the victim models without any extra information.

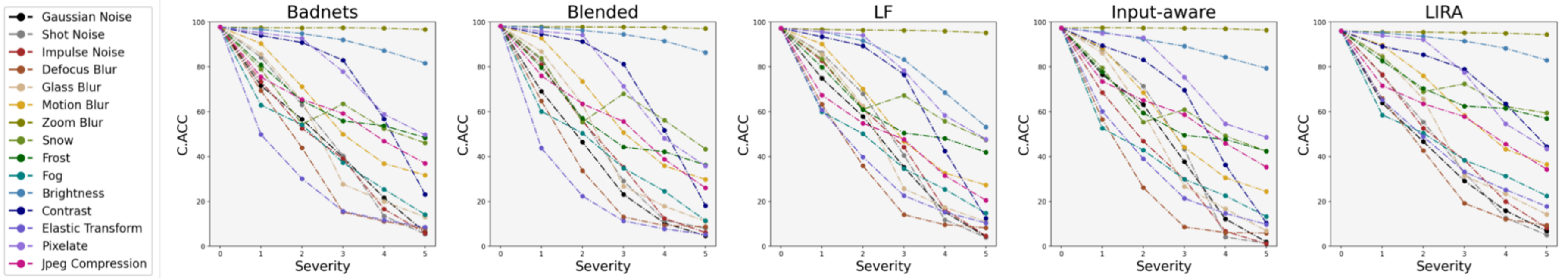
Method	Black-box Access		No Need of	Trigger Aussmptions		
	Logits-based	Decision-based	Clean Data	Universal	Sample-specific	Invisible
SentiNet [5]	○	○	○	●	○	○
SCan [39]	○	○	○	●	○	○
Beatrix [30]	○	○	○	●	●	●
NEO[3] [42]	●	●	●	○	○	○
STRIP [12]	●	○	○	●	○	○
FreqDetector [48]	●	●	○	●	●	●
TeCo (Ours)	●	●	●	●	●	●

2. Insights

- Given a backdoor-infected model, it will show clearly different robustness for trigger samples influenced by different image corruptions. However, for the clean images, the model will show similar robustness against the majority of image corruptions.



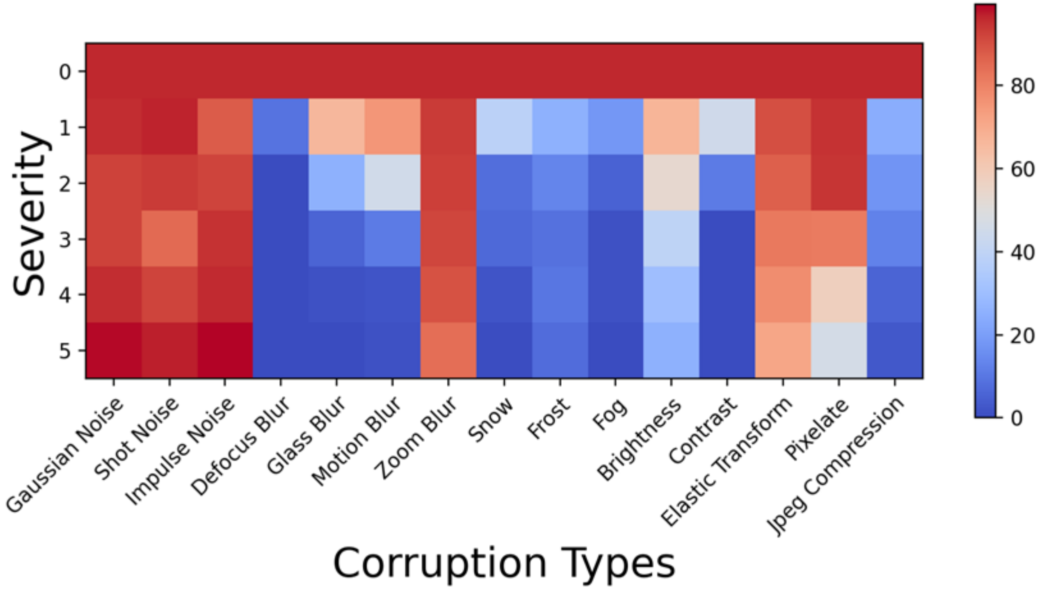
(a) The curves of ASR on trigger samples



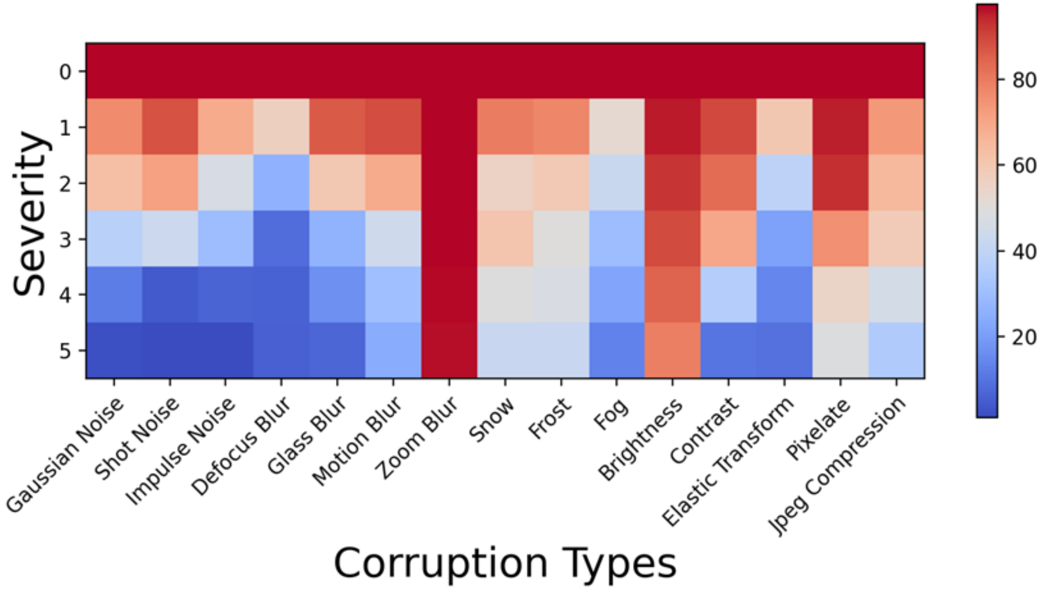
(b) The curves of ACC on clean images

2. Insights

- Given a backdoor-infected model, it will show clearly different robustness for trigger samples influenced by different image corruptions. However, for the clean images, the model will show similar robustness against the majority of image corruptions.



Trigger Samples



Clean Samples

3. Method

- A reasonable understanding is that the reduction of ACC or ASR is equivalent to the transitions of prediction labels. Consequently, we can evaluate the corruption robustness consistency in the inference stage by adding image corruptions with growing severity, and recording the severity when the model's hard-label prediction gets changed.

Algorithm 1: Test-time CRC Evaluation (TeCo)

Input: Test sample x; test model C_θ; deviation measurement method Dev;image corruption set $\mathcal{D}_K^N$, where K is the number of corruption types, and N is the maximum of severity.

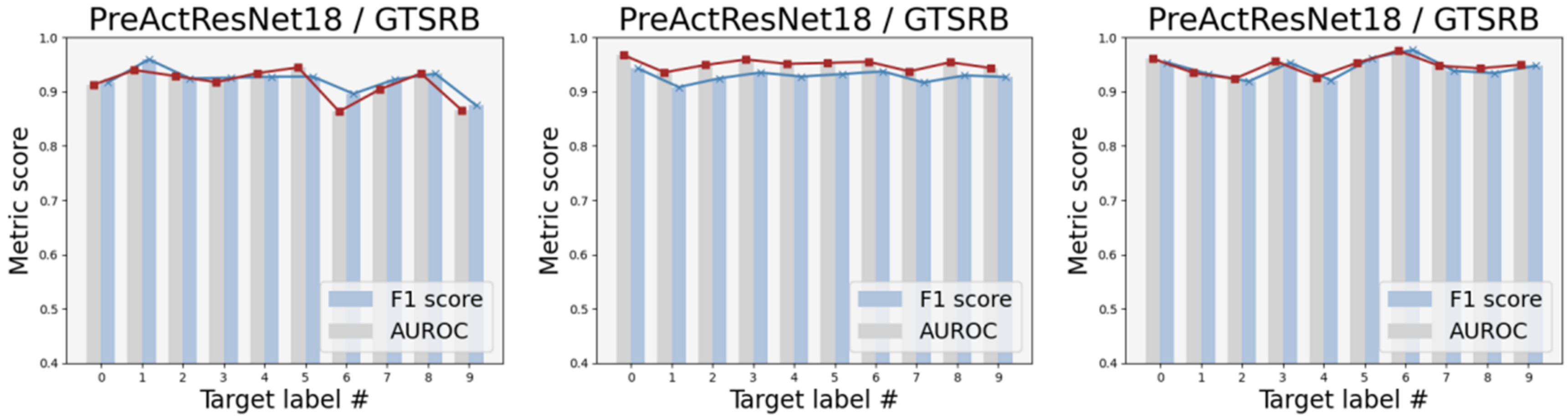
Output: Prediction score of test sample x.

Initialize $\mathcal{L} \leftarrow \{\}$, $P_{org} \leftarrow C_\theta(x)$;
for $k = 1$ *to* K **do**
    $L \leftarrow N + 1$;
    for $n = 1$ *to* N **do**
        if $C_\theta(D_k^n(x)) \neq P_{org}$ **then**
            $L \leftarrow n$;
            break;
        end
    end
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{L\}$;
end
$deviation \leftarrow Dev(\mathcal{L})$;
return $deviation$

4. Evaluations

- Extensive experiments demonstrate that compared with state-of-the-art defenses, TeCo outperforms them on different backdoor attacks, datasets, and model architectures, enjoying a higher AUROC by 10% and 5 times of stability.

Model	Attack→ Detection↓	Badnets [14]		Blended [3]		LF [48]		Input-aware [32]		Wanet [33]		LIRA [8]		SSBA [26]		AVG(↑)		STD(↓)	
		AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score
PreActResNet18	STRIP	0.790	0.743	0.726	0.685	0.973	0.937	0.283	0.526	0.395	0.526	0.555	0.661	0.364	0.526	0.584	0.658	0.236	0.140
	FreqDetector	0.989	0.955	0.966	0.904	0.886	0.809	1.000	0.993	0.566	0.550	0.912	0.840	0.896	0.824	0.888	0.839	0.138	0.134
	Ours	0.911	0.917	0.935	0.946	0.939	0.937	0.905	0.921	0.915	0.905	0.953	0.934	0.868	0.883	**0.918**	**0.920**	**0.026**	**0.020**



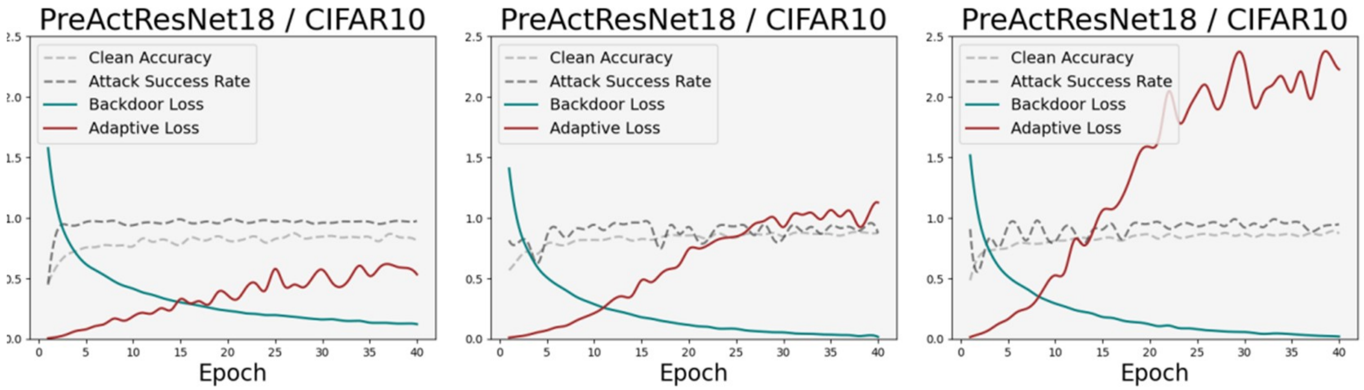
(a) Blended (b) SSBA (c) Wanet

5. Beyond TeCo

- An adaptive loss to attack the proposed TeCo:

$$\mathcal{J}_{bd} = \sum_{i=1}^{I} CE(C_\theta(x_i), y_i) + \sum_{j=1}^{J} CE(C_\theta(\hat{x}_j), y_t)$$

$$\mathcal{J}_{ada} = \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{n=1}^{N} \mathrm{MSE}\Big(\mathrm{MSE}\big(C_\theta(x_j), C_\theta\big(D_n^k(x_j)\big)\big), \mathrm{MSE}\big(C_\theta(\hat{x}_j), C_\theta\big(D_n^k(\hat{x}_j)\big)\big)\Big)$$



(a) Badnets (b) LF (c) SSBA

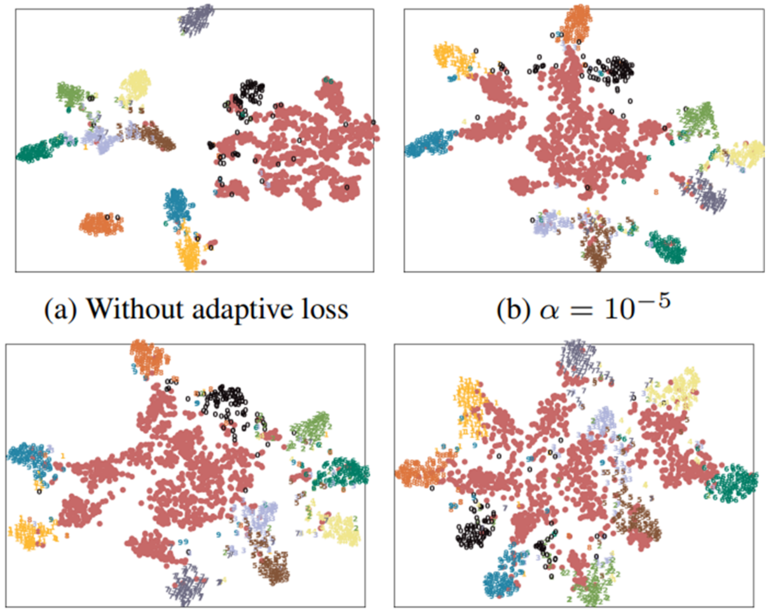
The adaptive loss grows when the backdoor loss decreases, which means the success on the dual-target loss function may drive the model to behave differently in terms of corruption robustness.

5. Beyond TeCo

- The adaptive loss pushes the trigger samples from the edge of latent space to the center, making them have a similar distance to different clean samples. Thus, a possible way to attack TeCo is to embed trigger samples in the middle of the latent space.

Weight→	0		10^{-3}		10^{-4}		10^{-5}	
Attack↓	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score	AUROC	F1 score
BadNets	0.9112	0.9174	0.5763	0.5928	0.6571	0.6542	0.6745	0.6657
LF	0.9390	0.9367	0.8592	0.8483	0.9219	0.9154	0.8667	0.8858
SSBA	0.8683	0.8835	0.7125	0.7312	0.6477	0.7281	0.5909	0.6852

Weight→	0		10^{-3}		10^{-4}		10^{-5}	
Attack↓	C.ACC	ASR	C.ACC	ASR	C.ACC	ASR	C.ACC	ASR
BadNets	0.9153	0.9502	0.5105	0.7386	0.7980	0.3720	0.8546	0.3001
LF	0.9286	0.9888	0.8022	0.9443	0.8864	0.9504	0.8962	0.9476
SSBA	0.9270	0.9719	0.7129	0.9176	0.8925	0.9162	0.8978	0.9170



(a) Without adaptive loss (b) $\alpha = 10^{-5}$

(c) $\alpha = 10^{-4}$ (d) $\alpha = 10^{-3}$

Thanks!

https://github.com/CGCL-codes/TeCo