Supplementary Material for Inspecting Prediction Confidence for Detecting Black-box Backdoor Attacks

Tong Wang[1], Yuan Yao[1], Feng Xu[1], Miao Xu[2], Shengwei An[3], Ting Wang[4]

[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]University of Queensland, Australia [3]Purdue University, USA [4]Stony Brook University, USA
mg20330065@smail.nju.edu.cn, y.yao@nju.edu.cn, xf@nju.edu.cn
miao.xu@uq.edu.au, an93@purdue.edu, twang@cs.stonybrook.edu

Abstract

This supplementary material provides the proofs and additional experiments for the AAAI 204 paper "Inspecting Prediction Confidence for Detecting Black-box Backdoor Attacks".

1 Proof for Theorem 1

Assume the data point (X, Y) is sampled uniformly at random from $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = [c]$, i.e., there are c labels. In the ideal case, the multi-class learning will optimize the following loss,

$$\begin{aligned}\mathcal{R}(\mathbf{g}) &= \mathbb{E}_{(X,Y)\sim p(x,y)}[\ell(\mathbf{g}(X), \mathbf{e}^Y)] \\ &= \sum_{z=1}^{c} p(Y=z)\mathbb{E}_{X\sim p(x|Y=z)}[\ell(\mathbf{g}(X), \mathbf{e}^z)]\end{aligned}$$

where $\mathbf{g} : \mathcal{X} \to \mathbb{R}^c$, $X \in \mathcal{X}$, and $Y \in [c]$. $\mathbf{e}^Y$ is the standard canonical vector with the Y-th entry to be one and all others zero. ℓ is the defined loss function. We use cross-validation loss in our derivations.

By optimizing the risk, we could obtain a $\mathbf{g}^*(\cdot)$, which is

$$\mathbf{g}^* = \arg\max_{\mathbf{g}\in\mathcal{G}} \mathcal{R}(\mathbf{g})$$

with the prediction

$$\widehat{Y} = \arg\max_{i\in[c]} g_i(X)$$

where $g_i(X)$ is the i-th entry of $\mathbf{g}(X)$.

With the poisoning data by a backdoor attack, some data's label will be shifted to the target label t. In this way, the above ideal loss needs some modification. Without loss of generality, we denote the target label as t, and denote the poisoning rate in each label as a_z, i.e., $a_z = p(Y_a = t|Y = z)$ and $1 - a_z = p(Y_a = z|Y = z)$ where Y_a is shifted to t after the attack. Then, we have

$$\begin{aligned}\mathcal{R}_a(\mathbf{g}) &= \sum_z a_z p(Y=z)\mathbb{E}_{X\sim p(x|y=z)}[\ell(\mathbf{g}(X), \mathbf{e}^t)] + \\ &\quad \sum_z (1-a_z)p(Y=z)\mathbb{E}_{X\sim p(x|y=z)}[\ell(\mathbf{g}(X), \mathbf{e}^z)]\end{aligned}$$

whose minimizer is denoted as $\mathbf{g}_a^*$. Here, we assume that the trigger involves very small perturbations and can be approximately ignored in the above equation.

In reality, we optimize an empirical version of $\mathcal{R}(g)$, which is

$$\widehat{\mathcal{R}}(\mathbf{g}) = \sum_{i=1}^{n} \ell(\mathbf{g}(x_i), \mathbf{e}^{y_i}).$$

Next, we consider the term $\mathcal{R}(\mathbf{g}_a^*) - \mathcal{R}(\mathbf{g}^*)$. Note that

$$\begin{aligned}\mathcal{R}(\mathbf{g}_a^*) - \mathcal{R}(\mathbf{g}^*) \quad &\leq \quad |\mathcal{R}(\mathbf{g}_a^*) - \widehat{\mathcal{R}}(\mathbf{g}_a^*)| + |\widehat{\mathcal{R}}(\mathbf{g}_a^*) - \widehat{\mathcal{R}}_a(\mathbf{g}_a^*)| \\ &\quad + |\widehat{\mathcal{R}}_a(\mathbf{g}_a^*) - \widehat{\mathcal{R}}(g^*)| + |\widehat{\mathcal{R}}(g^*) - \mathcal{R}(g^*)|.\end{aligned} \tag{1}$$

For the first and the fourth terms in the RHS of the above equation, we could have the following two results assuming that the loss function ℓ is upper bounded by M,

$$|\mathcal{R}(\mathbf{g}_a^*) - \widehat{\mathcal{R}}(\mathbf{g}_a^*)| \leq 2\mathfrak{R}_n(\ell \circ \mathcal{G}) + M\sqrt{\frac{\log(2/\delta)}{2n}}$$

$$|\mathcal{R}(\mathbf{g}^*) - \widehat{\mathcal{R}}(\mathbf{g}^*)| \leq 2\mathfrak{R}_n(\ell \circ \mathcal{G}) + M\sqrt{\frac{\log(2/\delta)}{2n}}$$

which could be easily proved by *McDiarmid's inequality* (McDiarmid 1989) and the *symmetrization* (Vapnik 1998). Here $\mathfrak{R}_n(\cdot)$ denotes the Rademacher Complexity. Note that if assuming $\ell(\mathbf{g}(x), \mathbf{e}^Y)$ is Lipschitz continuous with a Lipschitz constant L_ℓ, by the *Talagrand's contradiction lemma* (Maurer 2016) we have

$$\mathfrak{R}_n(\ell \circ \mathcal{G}) \leq \sqrt{2}L_\ell \sum_{y=1}^{c} \mathfrak{R}_n(\mathcal{G}_y)$$

where $\mathcal{G}_y$ is the hypothesis space for $g_y(\cdot)$.

For the second term in the RHS of Eq. (1), we have

$$\begin{aligned}&|\widehat{\mathcal{R}}(\mathbf{g}_a^*) - \widehat{\mathcal{R}}_a(\mathbf{g}_a^*)| \\ = \quad &\sum_z a_z \cdot p(Y=z) \sum_{x_i\in D_z} |\ell(\mathbf{g}_a^*(x_i), \mathbf{e}^z) - \ell(\mathbf{g}_a^*(x_i), \mathbf{e}^t)|\end{aligned}$$

where we use D_z to denote the subset of training data whose labels are changed to t from label z.

For the third term in the RHS of Eq. (1), we have

$$\begin{aligned} & |\widehat{\mathcal{R}}(\mathbf{g}^*) - \widehat{\mathcal{R}}_a(\mathbf{g}_a^*)| \\ = \ & \sum_z (1-a_z)p(Y=z) \sum_{x_i \in D_z} |\ell(\mathbf{g}^*(x_i), \mathbf{e}^z) - \ell(\mathbf{g}_a^*(x_i), \mathbf{e}^z)| \\ = \ & + \sum_z a_z \cdot p(Y=z) \sum_{x_i \in D_z} |\ell(\mathbf{g}^*(x_i), \mathbf{e}^z) - \ell(\mathbf{g}_a^*(x_i), \mathbf{e}^t)|. \end{aligned} \tag{2}$$

In the ideal case, backdoor attacks usually require that the predictions do not change for clean data, i.e.,

$$g_{a_z}^*(X) = g_z^*(X)$$

for clean data (data points with both $Y_a = z$ and $Y = z$). The first term in the RHS of Eq. (2) can be approximated to zero, i.e., $\ell(\mathbf{g}^*(x_i), \mathbf{e}^z) \sim \ell(\mathbf{g}_a^*(x_i), \mathbf{e}^z)$.

Combining together, we have

$$\begin{aligned} \mathcal{R}(\mathbf{g}_a^*) - \mathcal{R}(\mathbf{g}^*) \quad \leq \quad & 4\mathfrak{R}_n(\ell \circ \mathcal{G}) + 2M\sqrt{\frac{\log(2/\delta)}{2n}} + \\ & 2\sum_z a_z \sum_{x_i \in D_z} |\ell(\mathbf{g}^*(x_i), \mathbf{e}^z) - \ell(\mathbf{g}_a^*(x_i), \mathbf{e}^t)|, \end{aligned}$$

which completes the proof.

2 Limitations of Existing Defenses

Here, we provide some detailed explanations about the limitation of the state-of-the-art backdoor defenses that aim to detect whether a given model is trojaned.

NC (Wang et al. 2019). The observation behind this defense is two-fold. First, a backdoor trigger essentially creates a shortcut from the original label to the target label. Second, the trigger is usually of small size. Therefore, NC proposes to reverse-engineer a trigger for each label based on a set of clean data, and then apply outlier detection to all of the reversed triggers' L_1 norms to detect the potential trigger. A significantly smaller reversed trigger indicates the existence of a backdoor. However, it is observed that NC becomes futile when the trigger size is relatively large (Guo et al. 2020). Recent attacks (e.g., REFOOL (Liu et al. 2020) and FTROJAN (Wang et al. 2022)) have also successfully bypassed the detection of NC by dispersing the trigger into a larger area.

ABS (Liu et al. 2019). The key observation behind ABS is that a backdoor attack essentially compromises the inner neurons of deep neural networks, and a trigger usually correlates to a neuron stimulating which can lead to a prediction label regardless of the given inputs. With this observation, ABS proposes to analyze the activation pattern of each neuron to detect backdoor attacks. However, ABS analyzes one neuron each time and cannot deal with the attacks compromising a group of neurons, making it fall short against advanced attacks involving multiple triggers or multiple target labels (Gao et al. 2020).

ULP (Kolouri et al. 2020) and MNTD (Xu et al. 2021). ULP and MNTD are based on the observation that benign models and trojaned models exhibit different behavior/parameter patterns. Based on this observation, ULP learns the so-called universal litmus patterns of each model, and MNTD generates a variety of trojaned models following a parameter distribution. Then, they train a meta-classifier based on a number of clean and trojaned models, and use the meta-classifier to decide if a model is trojaned or not. However, although the trained meta-classifier shows some generalization ability, it may still overfit the training models and fail to generalize to unseen attacks or triggers that are significantly different from the trigger patterns in the training data.

3 Details of the Threat Model

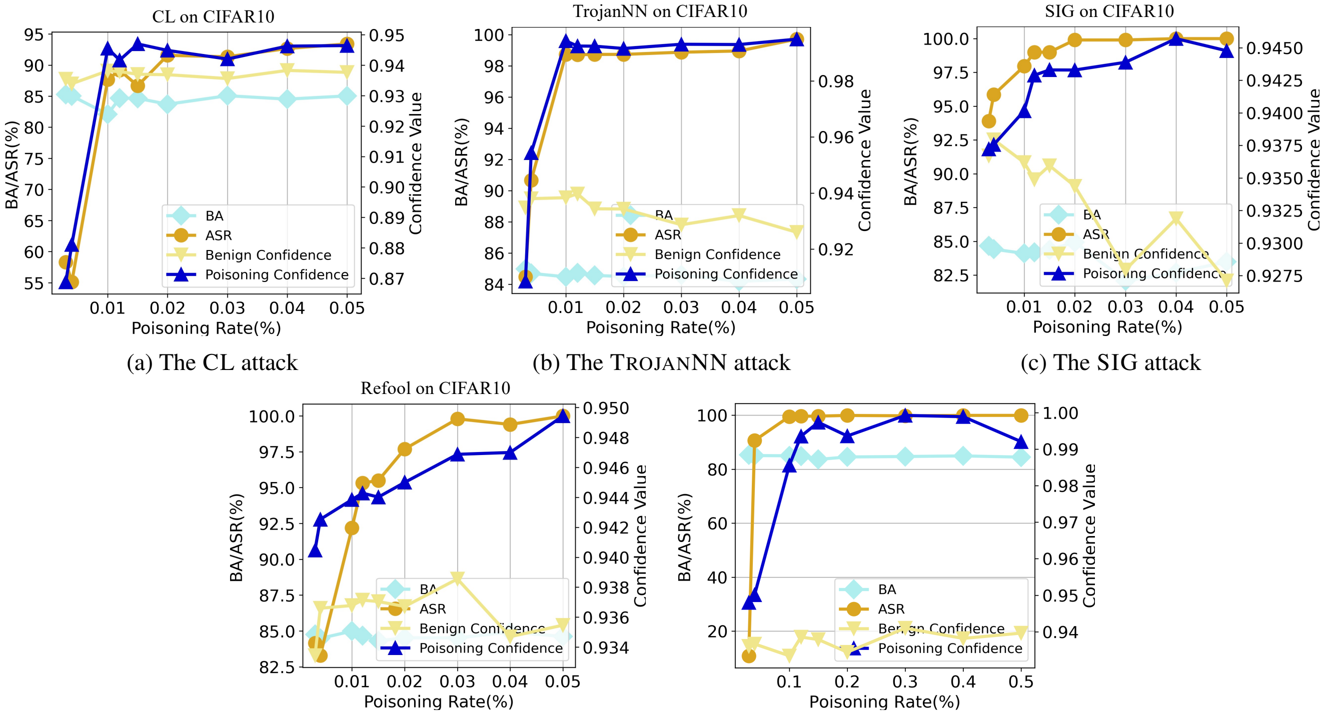
We consider the black-box backdoor attacks. There are three roles, i.e., *adversary*, *victim*, and *defender*. The adversary defines a trigger beforehand, injects it into data samples, and then disseminates the data. This assumption is practical in reality. For example, the user training the model may collect data from the Web, which may contain poisoned samples deliberately published by the adversary. The adversary does not have access to the training process and the model. We allow the adversary to launch advanced attacks (e.g., partial attack, MTOT attack, and MTMT attack as mentioned in introduction). We also assume that the adversary only attacks a minority of labels. The victim trains a model and some of the data collected to train the model may have been poisoned by the adversary. Since the ratio of poisoned data may vary, we also let the adversary to adaptively tune the poisoning rate to lower down the prediction confidence. For the defender, we assume she can access both the training data and the model, helping the victim to determine if the trained models have been implanted a backdoor. The defender does not require extra clean data, whose collection and annotation might be expensive.

4 Details of Experimental Setup

We first summarize the statistics of the used datasets. The CIFAR10 dataset contains 60,000 color images of size 32×32 in 10 different classes. We split it into 50,000 training images and 10,000 test images. We train a CNN with six convolutional layers and two fully-connected layers. For the other three datasets, we resize them all to size 224×224. For GTSRB, we have 4,772 training images and 293 test images. For PubFig, we have 12,800 training images and 3,200 test images. For ImageNet, we have 20,567 training images and 800 test images. We train a ResNet50 model (He et al. 2016) for ImageNet, and a ResNet34 model for both GTSRB and PubFig.

We then consider all possible attack-dataset combinations ($6 \times 4 = 24$) except that we only apply CL on CIFAR10, resulting in 21 combinations. The reason is that our implementation shows CL is ineffective on datasets other than CIFAR10. Note that we use the open source code provided by the authors whenever possible, and only the open source implementation of CL is unavailable; nevertheless, our implementation achieves almost the same results with the original paper.

For each attack, we found that selecting different target labels makes little difference, which is also consistent with existing studies. Therefore, we simply set the target label index to 0 for all the attacks by default. For BADNET and



(a) The CL attack

(b) The TROJANNN attack

(c) The SIG attack

(d) The REFOOL attack

(e) The FTROJAN attack

Figure 1: More empirical results for our observation. As the poisoning rate increases, the average prediction confidence of poisoning data becomes significantly higher than that of the clean data.

TROJANNN, we define a square-shaped trigger with size 4 × 4 for 32 × 32 images and with size 32 × 32 for 224 × 224 images. These triggers are placed in the lower right corner of the image. For the other attacks, we use their default trigger setup.

For advanced attacks, we implement them using BADNET on CIFAR10. In partial attack, we only poison label 1 data to the target label 0. In MTOT attack, we define three triggers at the lower-right, lower-left, and upper-left corners, respectively. In MTMT attack, we use the same three triggers as in MTOT, but for target labels 0, 1, and 2, respectively. We also test other random choices of target labels, and repeat the experiments for five times. The results are quite stable.

5 Addditional Experimental Results

5.1 More Empirical Results for our Observation

We first show the empirical results of more attacks for our observation. The result of BADNET is shown in the main body of the paper, and here we further show the results of CL, TROJANNN, SIG, REFOOL, and FTROJAN on CIFAR10 in Figure 1. We can observe that the average prediction confidence of poisoning data is significantly higher than that of the clean data for all the five attacks.

5.2 Detection Results for all Combinations

We next show the detailed detection results of DTINSPECTOR under various attack and dataset combinations, and the results are shown in Table 1. We can observe that DTINSPECTOR can successfully distinguish trojaned/clean models as well as the infected/clean labels in 20 out of the 21 combinations. The only exception is the REFOOL and PubFig combination. The reason is that the trigger of REFOOL covers the most influential areas (e.g., the central part) in the face images, and patching such areas may easily change the predictions. This limitation can be further addressed by, e.g., applying image restoration techniques such as GANs, and we leave this as future work.

5.3 Sensitivity to Trigger Size

Here, we include the sensitivity result by varying the trigger size. We apply BADNET on CIFAR10 and fix poisoning rate to 5%. For the trigger, we gradually increase its size from 4 × 4 to 22 × 22. The results are shown in Table 2. We can observe that DTINSPECTOR is more robust to trigger size than NC. NC cannot detect the trojaned model when the trigger size is no smaller than 13 × 13, while DTINSPECTOR can make a successful detection even when the trigger size grows to 22 × 22. Considering that the image size of CIFAR10 is 32 × 32, this means that DTINSPECTOR can still work even when nearly half of the image contains poisoning pixels.

5.4 Sensitivity to Sampling Size

We next include the detailed sensitivity results of the sampling size. We still use BADNET on CIFAR10, and fix the

Table 1: Detailed detection results of DTInspector for all the attack and dataset combinations. The wrong detection is underlined.

Dataset	Attack	Detection Result	Detect Infected Label
CIFAR10	BadNet	Y(199.7)	Y
CIFAR10	TrojanNN	Y(32.0)	Y
CIFAR10	CL	Y(61.2)	Y
CIFAR10	SIG	Y(13.5)	Y
CIFAR10	Refool	Y(68.0)	Y
CIFAR10	FTrojan	Y(148.0)	Y
GTSRB	BadNet	Y(4.5)	Y
GTSRB	TrojanNN	Y(3.7)	Y
GTSRB	SIG	Y(3.66)	Y
GTSRB	Refool	Y(4.9)	Y
GTSRB	FTrojan	Y(4.0)	Y
PubFig	BadNet	Y(7.1)	Y
PubFig	TrojanNN	Y(4.9)	Y
PubFig	SIG	Y(12.1)	Y
PubFig	Refool	N(1.07)	N
PubFig	FTrojan	Y(5.2)	Y
ImageNet	BadNet	Y(11.6)	Y
ImageNet	TrojanNN	Y(8.4)	Y
ImageNet	SIG	Y(7.7)	Y
ImageNet	Refool	Y(4.6)	Y
ImageNet	FTrojan	Y(3.7)	Y

Table 2: Sensitivity comparisons with NC on trigger size. DTInspector is more robust to trigger size than NC.

Trigger Size	Detection Result	
	NC	DTInspector
4×4	Y (2.62)	Y (199.65)
7×7	Y (2.17)	Y (53.28)
10×10	Y (2.24)	Y (121.03)
13×13	N (1.88)	Y (66.77)
16×16	N (1.36)	Y (36.51)
19×19	N (1.44)	Y (62.70)
22×22	N (1.27)	Y (5.39)

trigger size to 4×4, and the poisoning rate to 5%. We then vary the sampling size from 50 to 2000, and the results are shown in Table 3. We can see that DTInspector is relatively robust to the sampling size in a wide range. Specifically, it successfully detects the backdoor in a wide range, even when only 50 images are sampled from both high- and low-confidence data. This makes DTInspector practically feasible as an effective backdoor attack usually needs to poison at least hundreds or thousands of images. We also test different sampling methods to obtain the high-confidence and low-confidence samples (e.g., randomly sampling K samples from the top-$2K$ samples), and the results show little difference.

5.5 Detection against Adaptive Attacks

Next, we conduct an adaptive attack where the attacker can control the poisoning rate in a range to intentionally lower down the prediction confidences of poisoned samples. We perform BadNet on CIFAR10 as an example, and gradually decrease the poisoning rate from 50% to see the detection performance of DTInspector. The results are shown in Table 4. First, we can see that when the poisoning rate is high (e.g., above 30%), the benign accuracy begins to decrease, violating the stealthiness requirement of a backdoor attack. Still, DTInspector is able to detect the attack with poisoning rate 50%. Second, when the poisoning rate drops below 0.06%, DTInspector cannot detect the existence of the attack. However, in such poisoning rates, the ASR also drops to 66.37%, rendering a less effective backdoor attack.

Table 3: Sensitivity results on sampling size. DTInspector works as this parameter varies in a wide range.

Sampling Size	Detection Result
50	Y (199.6)
100	Y (31.4)
200	Y (65.4)
400	Y (133.9)
800	Y (103.6)
1500	Y (327.0)
2000	Y (166.8)

Table 4: Detection results against adaptive attacks. The attacker tries to reduce the prediction confidence via varying the poisoning rate. DTInspector is still effective as long as the backdoor attack is effective (e.g., the ASR is above 70%).

Poisoning Rate (%)	Benign Acc. (%)	ASR (%)	Detection Result
50.00	80.33	98.23	Y (51.94)
40.00	81.24	97.80	Y (84.64)
30.00	83.44	97.33	Y (30.35)
20.00	83.94	97.35	Y (18.20)
10.00	84.95	96.85	Y (33.05)
3.00	85.05	96.34	Y (199.65)
0.10	85.39	77.71	Y (32.04)
0.09	85.13	75.53	Y (6.67)
0.08	85.46	73.64	Y (6.24)
0.07	85.62	72.25	Y (4.17)
0.06	85.65	66.37	N (0.00)
0.05	85.65	57.35	N (0.00)

References

Gao, Y.; Doan, B. G.; Zhang, Z.; Ma, S.; Zhang, J.; Fu, A.; Nepal, S.; and Kim, H. 2020. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760*.

Guo, W.; Wang, L.; Xing, X.; Du, M.; and Song, D. 2020. Towards Inspecting and Eliminating Trojan Backdoors in Deep Neural Networks. In *ICDM*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Kolouri, S.; Saha, A.; Pirsiavash, H.; and Hoffmann, H.

2020. Universal litmus patterns: Revealing backdoor attacks in cnns. In *CVPR*, 301–310.

Liu, Y.; Lee, W.-C.; Tao, G.; Ma, S.; Aafer, Y.; and Zhang, X. 2019. ABS: Scanning neural networks for back-doors by artificial brain stimulation. In *CCS*, 1265–1282.

Liu, Y.; Ma, X.; Bailey, J.; and Lu, F. 2020. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 182–199. Springer.

Maurer, A. 2016. A Vector-Contraction Inequality for Rademacher Complexities. In *Proceedings of the 27th International Conference on Algorithmic Learning Theory (ALT)*, 3–17.

McDiarmid, C. 1989. On the method of bounded differences. *Surveys in combinatorics*, 141(1): 148–188.

Vapnik, V. 1998. *Statistical learning theory*. Wiley.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *SP*, 707–723.

Wang, T.; Yao, Y.; Xu, F.; An, S.; Tong, H.; and Wang, T. 2022. An Invisible Black-box Backdoor Attack through Frequency Domain. In *European Conference on Computer Vision (ECCV)*.

Xu, X.; Wang, Q.; Li, H.; Borisov, N.; Gunter, C. A.; and Li, B. 2021. Detecting ai trojans using meta neural analysis. In *SP*, 103–120.