EFFICIENT ANY-TARGET BACKDOOR ATTACK WITH PSEUDO POISONED SAMPLES

Bin Huang, Zhi Wang⋆

Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China

2023 IEEE International Conference on Image Processing (ICIP) | 978-1-7281-9835-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIP49359.2023.10222807

ABSTRACT

Deep neural networks present their potential vulnerabilities to backdoor attacks. They have a satisfactory performance for benign users with clean samples but will get malicious outputs when inputs are attached with the backdoor trigger. Current backdoor attacks on image classifiers usually target only one single class, making them not robust to defenses against this characteristic. In this work, we propose a new any-target attack that targets all the labels simultaneously with the triggers being invisible and input-dependent. Specifically, we train the classifier together with the image steganography model by encoding the one-hot encodings into the input images. The novel pseudo poisoned samples are then introduced to improve the effectiveness of our backdoor attack. Experimental results show that our method is both effective and efficient on several datasets and is robust to existing defenses.

Index Terms— Backdoor Attack, Any-Target Attack, Image Steganography, Pseudo Poisoned Samples

1. INTRODUCTION

Deep neural networks have been applied in a variety of applications. However, recent studies have also revealed their vulnerabilities to backdoor attacks [1]. An attacker can poison a portion of the training samples and models trained on the partially modified training set will be embedded with hidden backdoor, predicting the target when the trigger shows up while behaving normally otherwise. Studies on backdoor attack usually focus on its effectiveness or stealthiness. To improve the effectiveness, the attacker tries to find better triggers to increase the attack success rate [2, 3]. While for the stealthiness, as existing attacks often rely on triggers that are visible and static, making them easily noticeable and detectable, many invisible [3, 4], dynamic [5, 6] or input-dependent [5, 7, 8] triggers have be proposed continually. What's more, backdoor attack without label modification, i.e. clean-label attack [9, 10, 11], can also enhance the stealthiness.

From the perspective of the target classes, we argue that targeting multiple labels simultaneously will make the attack stealthier and harder to be detected, because current defenses usually take the assumption that backdoor attacks have only one single target class. Hence, in this paper, we investigate an aggressive any-target attack paradigm, where the attacker generates poisoned samples of any target class and leads the attacked model to predict them as that specified class. We show in Section 3.1 that some existing single-target attacks [1, 12, 13] can be easily extended to the any-target attacks by mapping distinct triggers to different labels. But the process of trigger selection is not straightforward and the extended attacks lead to poor attack performance experimentally (see in Section 4.2). A few attacks [6, 14] are applicable to the any-target attack paradigm, but the triggers are obviously visible and input-independent.

Inspired by prior works on image steganography [15, 16] and the steganography based backdoor attacks [7, 17], we propose a new any-target attack by training the model to be attacked together with the image steganography model. The target label is specified by its corresponding one-hot encoding which is then encoded into the benign input image to produce a poisoned one. As thus, the tedious process of selecting distinct triggers can be efficiently implemented by just offering the one-hot encodings. What's more, the poisoned images are invisible and input-dependent owing to the encoder of the image steganography model. To encourage the model to learn the whole encoding space, we introduce the pseudo poisoned samples which are also generated by the encoder but with their labels unchanged, ensuring the effectiveness of our proposed any-target attack.

We evaluate our attack on four datasets MNIST, CIFAR-10, GTSRB and CelebA. Experiments show that on every dataset we achieve high attack success rate of over 99% and similar benign accuracy with the clean model. The triggers added to the poisoned images are visually imperceptible. The proposed attack can also bypass several typical defense methods. Ablation studies reveal its effectiveness even under a rather low attack ratio and the importance of the pseudo poisoned samples to the success of our attack whose capability of attacking larger number of classes has been verified as well.

Our main contributions are summarized as follows: (1) We propose a new any-target attack that targets all the labels simultaneously with the triggers being invisible and input-dependent by image steganography. (2) We introduce the novel pseudo poisoned samples to improve the attack performance of our method. (3) Extensive experiments demonstrate

Zhi Wang (wangzhi@sz.tsinghua.edu.cn) is the corresponding author. This work is supported by the Shenzhen Science and Technology Program (Grant No. RCYX20200714114523079 and JCYJ20220818101014030).

Table 1. The comparison between different attack paradigms.

single-target	all-target	any-target
$x_t = \mathcal{B}(x)$	$x_t = \mathcal{B}(x)$	$x_t = \mathcal{B}(x,t)$
$y_t = \mathcal{M}(y) = t$	$y_t = \mathcal{M}(y) = (y+1) \bmod Y$	$y_t = \mathcal{M}(y) = t$

the effectiveness and efficiency of our proposed attack and its robustness to several defense methods.

2. PROBLEM DEFINITION

In this section, we first review the definition of existing backdoor attacks and then introduce our any-target attack along with the difference between them.

Definition of Traditional Backdoor Attack. Given a clean training set $\mathcal{D} = \{(x, y)\}$, where x is an input image and y is the corresponding label, an attacker applies a backdoor injection function $\mathcal{B}$ and a label mapping function $\mathcal{M}$ on part of $\mathcal{D}$ to generate a poisoned training set $\mathcal{D}_t = \{(x_t, y_t)\} = \{(\mathcal{B}(x), \mathcal{M}(y))|(x, y) \in \mathcal{D}', \mathcal{D}' \subset \mathcal{D}\}$, where x_t is the poisoned sample and y_t is its poisoned label. In prior works, $\mathcal{M}$ is often a constant function $\mathcal{M}(y) = t$ (**single-target attack**), where t is the target label, or a shifting function $\mathcal{M}(y) = (y + 1) \bmod Y$ (**all-target attack**), where Y is the class number. The attack ratio is defined as the ratio of the poisoned sample volume to the training sample volume, i.e. $\alpha = |\mathcal{D}_t|/|\mathcal{D}|$. After poisoning the training set, the attacker injects a backdoor into the classifier by training it over $\mathcal{D}_t$ and the rest non-poisoned part of $\mathcal{D}$, i.e. $\mathcal{D}_t \cup \mathcal{D} \backslash \mathcal{D}'$. The goal of the attacker is to make target predictions if the trigger is present at test-time samples while without harming its classification accuracy on clean samples.

Definition of Any-Target Backdoor Attack. We refer to our attack paradigm as any-target attack. Different from the definition of previous attacks where the backdoor injection function $\mathcal{B}$ is only conditioned on the input image, in the any-target attack, $\mathcal{B}$ takes as input an image as well as a target class t, then it generates a poisoned sample via $x_t = \mathcal{B}(x, t)$. The mapping function $\mathcal{M}$ will output t which is used in the previous generating process. Different attack paradigms are summarized in Table 1. Note that the target class t in the any-target attack can be any class instead of a fixed one in the single-target attack. The mapping from the source class to the target class in the any-target attack is an all-to-all mapping, while that in the all-target attack is a one-to-one mapping. Overall, our any-target attack is much more challenging.

3. ANY-TARGET BACKDOOR ATTACK

In this section, we first show that a heuristic method extended from the single-target attack is incapable of achieving satisfactory any-target attack performance. Then we propose a novel approach based on the image steganography models.

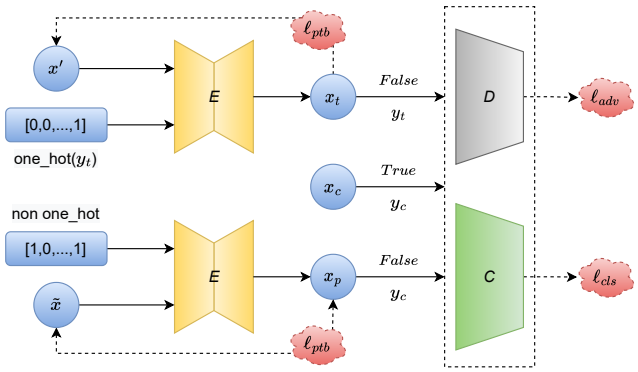


Fig. 1. The overall workflow of our proposed method. Three modes of the training process are illustrated. E, C and D are the encoder, classifier and discriminator respectively. The labels after x are ground truth for D and C respectively.

3.1. A Heuristic Method

Intuitively, we can simply extend existing single-target attack to any-target attack by manually selecting Y distinct trigger patterns for all the Y classes. Specifically, denote the Y selected triggers are $P = \{p_i\}_{i=1}^{Y}$, where p_i is the trigger assigned to target class i. The backdoor injection function $\mathcal{B}(x, t)$ will attach the trigger p_t into the image x. For example, extended from BadNets [1] $\mathcal{B}$ can be defined as follows:

$$\mathcal{B}(x, t) = x \odot (1 - m) + p_t \odot m, \tag{1}$$

where m is a blending mask and $\odot$ pixel-wise multiplication. Similarly, for Blended [18], Y distinct background images are chosen to be blended into the inputs. In a more effective invisible backdoor attack ISSBA [7], we input the string format t to generate the noise-like trigger for target class t.

After above *trigger selection*, we randomly poison a small part of the training set with $\mathcal{B}$ and $\mathcal{M}$, as described in Section 2, and train the classifier on the poisoned training data $\mathcal{D}_t \cup \mathcal{D} \backslash \mathcal{D}'$ in a standard procedure. The association between the triggers and the target classes is expected to be learned during the training. We refer to the above extended attacks as BadNets*, Blended*, ISSBA* and take them as the baselines. As we can see in Table 2, the heuristic any-target attacks lead to poor attack performance experimentally.

3.2. Our Proposed Method

The overall framework of our proposed attack is illustrated in Fig. 1. For the model structure, we take the image steganography model Hidden [15] as our base. Specifically, Hidden comprises four main components: an encoder, a noise layer, a decoder and a discriminator. We keep its encoder and discriminator as our *encoder* E and *discriminator* D, while replace its decoder with the *classifier* C to be attacked.

Given a dataset $\mathcal{D}$ with N samples, we randomly take α of them as the poison set $\mathcal{D}'$. The encoder E receives an image

x' from $\mathcal{D}'$ and a **random** target label y_t and then produces a poisoned image x_t which is encoded with y_t. Note that any vector can be used to represent the target label if only they are one-to-one mapped. In our method, we choose the one-hot encodings for convenience. The one-hot encoding space with length Y is defined as follows:

$$OH(Y) = \{v | v \in \{0,1\}^Y, |v| = 1\}. \quad (2)$$

Note that $OH(Y)$ contains only Y of total 2^Y points in the huge Y-dimensional space $\{0,1\}^Y$. To help the model learn the representation of the whole space, we propose to use some vectors besides $OH(Y)$ for trigger generation. These vectors are dubbed **pseudo encodings** with the definition of

$$PS(Y) = \{v | v \in \{0,1\}^Y, |v| \neq 1\}. \quad (3)$$

In hence, at the same time, we also randomly take α of the samples from $\mathcal{D}$ as the pseudo poison set $\tilde{\mathcal{D}}$. Then, an image $\tilde{x}$ from $\tilde{\mathcal{D}}$ is sent to E in company with a random pseudo encoding, resulting a **pseudo poisoned sample** x_p. The discriminator D tries to predicts the probability that the input image (x_c, x_t or x_p) is not poisoned.

Overall, there are three training modes for the classifier C: (1) **Clean mode**. C has to recognize a clean sample x_c as its true label y_c. (2) **Attack mode**. When x_t is encoded with a target label, the backdoor is activated, making C output wrong prediction y_t. (3) **Pseudo attack mode**. Although encoded with a pseudo encoding, x_p can still be predicted as y_c. Note that the difference between the poisoned samples and the pseudo ones lies in whether their labels are altered.

Following Hidden [15], to make the poisoned image have no obvious visual difference with the corresponding benign one, we adopt a *perturbation loss* ℓ_{ptb}, the ℓ_2 distance between x' and x_t (or x_p), and an *adversarial loss* ℓ_{adv}, the binary cross entropy, to constrain the encoder and the discriminator respectively. The cross entropy is taken as *classification loss* ℓ_{cls} to ensure the expected prediction. In one epoch, we first jointly train the encoder E and classifier C by minimizing the loss $\mathcal{L}_1$ in Eq. (4) through stochastic gradient descent. And then the discriminator D is trained likewise by minimizing the loss $\mathcal{L}_2$ in Eq. (5) as follows:

$$\begin{aligned}\mathcal{L}_1 = &\mathbb{E}_{(x,y)\in\mathcal{D}}\{\lambda_{cls}\ell_{cls}(C(x),y)\} \\ &+ \mathbb{E}_{(x,y)\in\mathcal{D}'\cup\tilde{\mathcal{D}}}\{\lambda_{cls}\ell_{cls}(C(E(x)),\mathcal{M}(y)) \\ &+ \lambda_{ptb}\ell_{ptb}(x,E(x)) + \lambda_{adv}\ell_{adv}(D(E(x)),1)\}, \quad (4)\end{aligned}$$

$$\mathcal{L}_2 = \mathbb{E}_{x\in\mathcal{D}'\cup\tilde{\mathcal{D}}}\{\ell_{adv}(D(x),1) + \ell_{adv}(D(E(x)),0)\}, \quad (5)$$

where the λs are hyperparameters that balance each loss.

4. EXPERIMENTAL RESULTS

4.1. Experimental Settings

Datasets and Models. We conduct our attack experiments on four datasets: MNIST, CIFAR-10, GTSRB and CelebA. The classifier is a 4-layer convolutional neural network for MNIST following previous studies [5] and Pre-activation Resnet-18 for others. The encoder and discriminator are kept consistent with Hidden [15] except for the input layer. To verify the effectiveness on larger number of classes, we select a subset of 200 classes from ImageNet.

Benign Poisoned Residual×10

Fig. 2. An example of the visualization of poisoned images.

Training Setup. We adopt the SGD optimizer with the initial learning rate set as 0.01, which drops to 0.1 times after every 100 epochs. The batch size and maximum epoch are 128 and 300 respectively. We set the attack ratio α to 0.1 by default. λ_{cls}, λ_{ptb} and λ_{adv} are set to 1, 0.7 and 0.001 finally. All experiments are conducted on NVIDIA RTX 3090 GPUs.

Evaluation. We take the heuristic attacks described in Section 3.1 as the baselines. The benign accuracy (ACC) and attack success rate (ASR) metrics, i.e. the accuracy on benign and poisoned samples, are used for evaluation.

4.2. Main Results

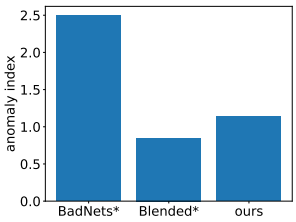
Attack Performance. Table 2 reports the comparison of different attacks. As we can see, under $\alpha = 0.1$, our attack can achieve a fairly high ASR approaching to 100% on all datasets, better than the others, while keeping ACC very close to the benign model. On the contrary, BadNets* or Blended* can only have high ASR (e.g. more than 90%) on few datasets (e.g. MNIST or GTSRB). Note that the invisible ISSBA* fails in the attack with an ASR of random guess, even though it is more advanced in the single-target attack. This indicates that heuristic extensions from the single-target attack do not work well and shows the feasibility and effectiveness of our method. Moreover, our attack is more efficient since it does not need tedious trigger selection compared with the others.

Poison Visual Effect. Fig. 2 presents an example of the poisoned images generated by our method. The poisoned image brings satisfactory visual invisibility, making it look natural under human inspection and stay stealthy.

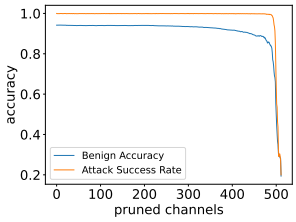
Resistance to Backdoor Defenses. We test our approach against three defenses: Neural Cleanse [19] (NC), Fine-Pruning [20] and Strip [21]. NC reverse engineers candidate triggers for each class and finds out significantly small one as the target. It calculates an *anomaly index* which indicates a backdoor when greater than 2. As shown in Fig. 3(a), our attack successfully bypasses NC. When pruning the channels on the penultimate layer increasingly, our attack retains its high ASR even when almost all the channels are pruned as in Fig. 3(b), showing the resistance to pruning-based defense.

Table 2. Attack performance (%) of different methods. The best results are in bold. * means the heuristic attacks.

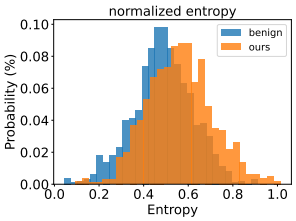
		MNIST			CIFAR10			GTSRB			CelebA		
attack ↓	$\alpha \rightarrow$	0.1	0.06	0.02	0.1	0.06	0.02	0.1	0.06	0.02	0.1	0.06	0.02
no attack	ACC		99.58			94.74			99.20			80.26	
BadNets*	ACC	99.47	99.40	99.44	93.52	93.82	94.04	**98.64**	**99.03**	98.60	**79.94**	79.29	79.09
	ASR	95.52	95.18	89.78	48.84	49.64	48.35	94.94	93.03	59.77	77.41	78.55	77.80
Blended*	ACC	99.59	99.60	**99.50**	93.68	94.00	94.21	98.62	98.73	98.95	79.91	79.57	79.27
	ASR	99.97	99.92	97.98	93.86	88.25	59.57	90.58	82.17	56.00	99.66	99.11	97.12
ISSBA*	ACC	-	-	-	93.40	93.85	**94.38**	98.51	98.83	**99.22**	79.59	**79.77**	**79.73**
	ASR	-	-	-	10.49	9.90	10.09	2.98	2.34	2.46	12.42	13.28	12.16
ours	ACC	**99.60**	**99.65**	99.47	**94.18**	**94.02**	92.21	98.54	98.91	99.04	79.44	79.31	78.76
	ASR	**100.0**	**100.0**	**100.0**	**99.97**	**99.84**	**99.45**	**99.26**	**99.01**	**99.81**	**99.70**	**99.89**	**99.58**



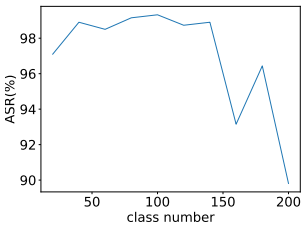
(a) Neural Cleanse



(b) Fine-Pruning



(c) Strip



(d) Class Numbers

Fig. 3. (a-c) Experiments results of three defenses against our attack. (d) The ASR under different class numbers.

Strip determines poisoned samples by calculating the entropy of the average prediction on the samples generated by imposing various clean samples on the testing samples. Smaller entropy means easier detection of the poisoned samples. But in our attack, as shown in Fig. 3(c), the entropy is close to, even slightly larger than that in the benign model, which demonstrates the robustness to Strip. All the results are on CIFAR-10 and results on other datasets are similar.

The Effect of Attack Ratios. We explore the attack performance under different α. As we can see in Table 2, even under more lower α, the ASR of our attack still stays at a high level of over 99%, while that of other attacks drops significantly, revealing the efficiency of our method. Note that there is a slight reduction on the ACC of our attack as α decreases to 0.02, which indicates it focuses more on learning the poisoned samples when they are too few.

Table 3. Performance (%) comparison of our attack with and without pseudo poisoned samples (PPS). $\alpha = 0.1$.

	CIFAR10		GTSRB		CelebA	
attack ↓	ACC	ASR	ACC	ASR	ACC	ASR
w/ PPS	**94.18**	**99.97**	**98.54**	**99.26**	**79.44**	**99.70**
w/o PPS	88.38	76.68	18.27	97.85	75.17	97.88

The Effect of Pseudo Poisoned Samples. We carry out a variation of our attack under $\alpha = 0.1$, which consists of only the first two modes described in Section 3.2, that is, without the pseudo attack mode. As we can see in Table 3, without the pseudo poisoned samples, both the ACC and ASR drop dramatically, which demonstrates the necessity of them.

The Effect of Class Numbers. We perform several attacks on a subset of 200 classes from ImageNet for every additional 20 classes. As shown in Fig. 3(d), our attack is effective under different target class numbers. Even though the ASR drops a little as the class number increases to 140, it still reaches about 90% when there are 200 classes.

5. CONCLUSION

In this work, we propose a novel any-target attack from the perspective of the target labels to improve the stealthiness of the backdoor attack. By training the classifier to be attacked together with the encoder of an image steganography model, the specified target label can be successfully encoded into the image as invisible trigger. With the assistance of the pseudo poisoned samples, the attacked model can well learn the association between the triggers and the target classes. Extensive experiments show the effectiveness and efficiency of our method even under a very low attack ratio.

6. REFERENCES

[1] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[2] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang, "Trojaning attack on neural networks," 2017.

[3] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li, "Lira: Learnable, imperceptible and robust backdoor attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11966–11976.

[4] Erwin Quiring and Konrad Rieck, "Backdooring and poisoning neural networks with image-scaling attacks," in *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020, pp. 41–47.

[5] Tuan Anh Nguyen and Anh Tran, "Input-aware dynamic backdoor attack," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3454–3464, 2020.

[6] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang, "Dynamic backdoor attacks against machine learning models," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2022, pp. 703–718.

[7] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16463–16472.

[8] Le Feng, Sheng Li, Zhenxing Qian, and Xinpeng Zhang, "Stealthy backdoor attack with adversarial training," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 2969–2973.

[9] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[10] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash, "Hidden trigger backdoor attacks," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, pp. 11957–11965.

[11] Alexander Turner, Dimitris Tsipras, and Aleksander Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.

[12] Xueluan Gong, Yanjiao Chen, Qian Wang, Huayang Huang, Lingshuo Meng, Chao Shen, and Qian Zhang, "Defense-resistant backdoor attacks against deep neural networks in outsourced cloud environment," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2617–2631, 2021.

[13] Nan Zhong, Zhenxing Qian, and Xinpeng Zhang, "Imperceptible backdoor attack: From input space to feature representation," *arXiv preprint arXiv:2205.03190*, 2022.

[14] Xinzhe Zhou, Wenhao Jiang, Sheng Qi, and Yadong Mu, "Multi-target invisibly trojaned networks for visual recognition and detection.," in *IJCAI*, 2021, pp. 3462–3469.

[15] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.

[16] Matthew Tancik, Ben Mildenhall, and Ren Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2117–2126.

[17] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2088–2105, 2020.

[18] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[19] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 707–723.

[20] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2018, pp. 273–294.

[21] Yansong Gao, 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*, 2019, pp. 113–125.