2025 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR) | 979-8-3315-8739-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICWAPR66259.2025.11306573

PRUNING BACKDOORS VIA INDUCIVE TRIGGER-BASED DETECTION

PEI-EN LIAO[1], JIA-HAO PAN[2], PATRICK P. K. CHAN[2,*]

[1]School of Automation Science and Engineering,South China University of Technology, 510641, China
[2]Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 510006, China.
E-MAIL: lpe_scut@163.com, jiahaopan88@163.com, patrickchan@scut.edu.cn

Abstract:

Backdoor attacks pose a serious threat to the security of deep neural networks by embedding hidden malicious behaviors that are triggered only by specific inputs. This paper presents a novel defense framework that detects and mitigates such attacks without requiring access to poisoned data or prior knowledge of the trigger. The proposed method introduces localized perturbations to clean samples to induce decoy behaviors, enabling the identification of infected models through output consistency analysis. Suspicious neurons are then identified using a Backdoor Loss Change (BLC) and Clean Loss Change (CLC) scoring mechanism, allowing fine-grained pruning of backdoor-relevant neurons while preserving those critical to clean performance. Experiments conducted on CIFAR-10 under various backdoor scenarios, including BadNet and Trojan attacks, demonstrate that the proposed approach significantly reduces attack success rates while maintaining high classification accuracy. Post-pruning fine-tuning further improves performance, establishing the framework as an effective and practical defense for real-world applications.

Keywords:

Backdoor Detection;Backdoor Defense; Pruning; Fine-tuning

1. Introduction

Deep Neural Networks (DNNs) have achieved outstanding performance in a wide range of applications. However, recent research [1, 2] has revealed that DNNs are vulnerable to backdoor attacks, in which models are manipulated to produce incorrect predictions when inputs contain specific, attacker-defined triggers [3]. This hidden vulnerability significantly undermines the trustworthiness and safety of DNNs in real-world deployment.

Backdoor attacks [3] are considered a severe form of data poisoning, where malicious behaviors are embedded during the training process without degrading model performance on clean data. To counter such attacks, various defense strategies have been proposed, particularly focusing on the design of more robust model architectures [4, 5]. Some methods aim to modify the training process to reduce the impact of poisoned samples. For example, suspicious inputs have been identified by analyzing early-stage training losses [6], while other methods attempt to isolate clean and poisoned learning objectives to prevent backdoor embedding [7]. Among these approaches, post-training backdoor mitigation has gained prominence, especially in scenarios where only a pre-trained model and limited clean data are available [8].

A major difficulty in post-training defense lies in the absence of known trigger samples. To address this, some techniques have reconstructed potential triggers through reverse engineering, allowing the model's hidden behavior to be exposed and analyzed [9, 10]. Alternatively, methods such as Magnitude-based Neuron Pruning (MNP) [11] have utilized clean data to suppress filters responsible for the clean task, thereby revealing filters that may contribute to the backdoor functionality. However, trigger reconstruction tends to be computationally intensive, requiring the solution of high-dimensional optimization problems. Moreover, MNP operates at the filter level, which may be ineffective when filters contain a mixture of clean and backdoor neurons. Such hybrid filters, especially when clean and malicious neurons are evenly distributed, make it difficult to isolate backdoor-specific behavior.

To overcome these limitations, a fine grained defense method at the neuron level is proposed, named Pruning Backdoor via Inducive Trigger based Detection. In this method, small perturbations are applied to clean samples to induce backdoor activation, thus simplifying the process of uncovering malicious behavior. A metric called Backdoor Loss Change (BLC) [11] is applied to evaluate neuron-level contributions to the backdoor effect, allowing precise identification of suspicious neurons. These neurons are then pruned selectively, rather than entire filters being removed. Finally, the pruned model is fine

979-8-3315-8739-0/25/$31.00 ©2025 IEEE

tuned using clean data to recover any accuracy lost during pruning. Through this neuron-focused and data-efficient strategy, the proposed method achieves effective backdoor mitigation with minimal performance degradation on clean samples.

The rest of this paper is organized as follows. Section 2 presents some related work on backdoor attacks and defenses; Section 3 elaborates on the main methodology of the paper; Section 4 describes the experimental part of the study; finally, Section 5 presents the conclusion of the paper.

2 Related Work

2.1 Backdoor Attacks

Backdoor attacks exploit the vulnerability of deep neural networks by embedding hidden behaviors during training that are activated by specific input patterns. These attacks are commonly classified based on the injection mechanism, including input-space and feature-space strategies. Input-space attacks embed trigger patterns directly into the input samples of the training data, leading the model to learn incorrect associations with target labels. These triggers can be static, remaining consistent across poisoned samples, or dynamic, varying in appearance to evade detection. They may also be localized to specific regions or span the entire input, adding further diversity to their design. Clean-label attacks such as SIG [12] pose additional difficulty, as they maintain consistent labels while subtly injecting features that guide the model toward misclassification.

Feature-space attacks differ in that the adversary manipulates internal representations or directly alters model parameters without relying on input perturbations. These attacks often assume greater control over the training process, allowing the backdoor to be implanted within the model's architecture or training dynamics. The variety and stealth of both input-space and feature-space attacks have made defense increasingly challenging, especially when the triggers are imperceptible or when attackers avoid modifying labels.

2.2 Backdoor Defenses

Defense strategies against backdoor attacks have evolved to address various attack scenarios, particularly when backdoor samples are unavailable. One prominent direction leverages the observation that backdoor-related neurons often exhibit abnormal sensitivity to perturbations. This property has been used to guide the selective removal or reinitialization of neurons or channels suspected of contributing to the backdoor effect, as explored in works such as Adversarial Neuron Pruning (ANP) [13] and Channel Lipschitzness based Pruning (CLP) [14]. These approaches aim to expose and suppress malicious components by examining the model's response to carefully designed perturbations.

Another line of defense focuses on trigger reconstruction using a small set of clean data. By reverse-engineering triggers that can elicit incorrect behavior, methods such as Neural Cleanse [9] and e Implicit Backdoor Adversarial Unlearning (I-BAU) [15] attempt to recover representative trigger patterns, which are then used to inform mitigation actions including pruning and unlearning. Although effective in certain cases, these techniques can be computationally intensive and may rely on assumptions about the attack's structure.

Recent studies have also explored defenses that do not require explicit trigger knowledge. These methods often aim to suppress or repurpose backdoor neurons through training dynamics or architectural interventions. For instance, techniques such as RNP [16] adopt asymmetric unlearning and recovery processes, while NAD [17] uses knowledge distillation to enforce clean behavior by aligning the compromised model with a clean teacher. Fine-Pruning (FP) [18] further introduces a two-stage mechanism that removes dormant neurons and subsequently fine-tunes the model to restore accuracy. Additionally, MNP [11] provides a structured pruning framework based on magnitude and task relevance, identifying and penalizing filters that contribute least to clean classification while preserving essential components to maintain model utility.

Collectively, these defense strategies reflect a growing emphasis on data-efficient, model-aware, and fine-grained solutions that aim to eliminate backdoors with minimal impact on clean performance, even in the absence of poisoned examples or explicit trigger information.

3 Proposed Method

This section presents a fine-grained neuron-level defense framework designed to detect and mitigate backdoor behaviors in pre-trained models using only a small clean dataset. The method consists of three core stages: (1) inducing backdoor behavior by perturbing clean inputs to identify suspicious model responses, (2) locating and pruning neurons that contribute to the backdoor functionality while preserving clean performance, and (3) fine-tuning the pruned model to recover potential accuracy degradation. The full workflow is outlined in Figure 1.

3.1 Inducing Backdoor Behavior via Perturbations

Let the complete training set be D_{tr}, where (x_i, y_i) is the i-th sample in D_{tr}. The input x_i belongs to the input space $\mathbb{R}^d$,

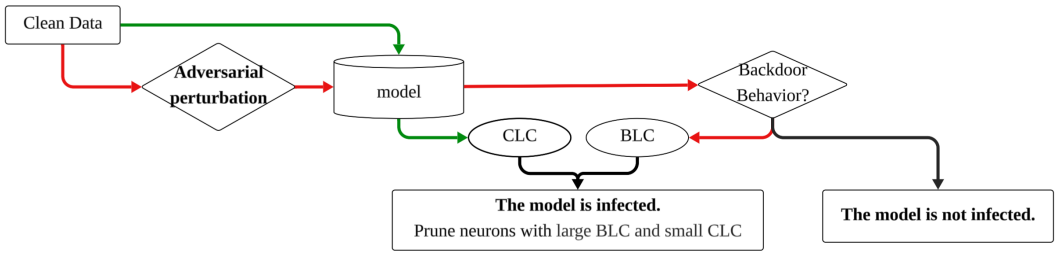


Figure 1. The pipeline of the proposed method. Clean data is used to derive two metrics: the Clean Loss Change (CLC) from original inputs (green path) and the Backdoor Loss Change (BLC) from perturbed decoy samples (red path). These scores then guide the fine-grained neuron pruning process.

and its label $y_i \in \{1, 2, \ldots, K\}$, assuming there are K classes. We define $\phi(x)$ as the poisoned input, and its corresponding backdoor target label is $\Psi(y)$. The training set D_{tr} is composed of a clean subset D_c and a poisoned subset D_b, such that $D_{tr} = D_c \cup D_b$.

Backdoor attacks that utilize local triggers often affect only a limited spatial region of the input. Previous works such as Neural Cleanse (NC) [19] have shown that such triggers, including those used in attacks like BadNet and Trojan [9], can sometimes be reconstructed due to their spatial locality. Inspired by this, the proposed method introduces spatially clustered random perturbations into clean inputs to simulate potential triggers. These perturbations, designed as random color mosaic patches, aim to generate "decoy samples" capable of activating the hidden backdoor behavior if present.

Let θ represents the parameters of a model $F(\cdot)$. $L(\cdot)$ is the loss function adopted by the model. The perturbation is defined as $\delta(m, n, k)$, where m and n specify the location of the patch and k its size. The objective is to identify the configuration (m^*, n^*, k^*) that maximizes the number of inputs for which the model's output changes in response to the perturbation. This can be formulated as the following optimization problem:

$$(m^*, n^*, k^*) = \arg\max_{m,n,k} \sum_{x_i \in D^{val}} \text{num}(x_i) \tag{1}$$

$$\text{num}(x_i) = \begin{cases} 1 & \text{if } F(x_i + \delta(m,n,k);\theta) \neq F(x_i;\theta) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\text{num}(x_i)$ counts the number of samples whose predictions change due to the added perturbation. Once the optimal perturbation δ^* is obtained, its effect is evaluated. If the majority of perturbed samples are consistently misclassified to a specific label, this provides strong evidence that the model has been compromised by an all-to-one backdoor attack.

3.2 Locating and Pruning Backdoor Neurons

Following detection, the next step is to identify and prune neurons that are primarily responsible for the backdoor behavior. Two metrics, Clean Loss Change (CLC) and Backdoor Loss Change (BLC)[11], are defined as Eq. (3) and (4) respectively.

$$\begin{aligned} \text{CLC}(\theta, i, j) = {} & E_{(x,y)\in D_c}[L(F(x;\theta|\theta^{(i,j)}=0), y)] \\ & - E_{(x,y)\in D_c}[L(F(x;\theta), y)] \end{aligned} \tag{3}$$

,

$$\begin{aligned} \text{BLC}(\theta, i, j) = {} & E_{(x,y)\in D_b}[L(F(\phi(x);\theta|\theta^{(i,j)}=0), \Psi(y))] \\ & - E_{(x,y)\in D_b}[L(F(\phi(x);\theta), \Psi(y))] \end{aligned} \tag{4}$$

The change in cross-entropy loss following a neuron's removal directly quantifies its task-related importance. Specifically, an increased loss signifies a beneficial neuron whose removal degrades performance, whereas a decreased loss indicates a detrimental neuron, making its removal advantageous. A negligible change implies the neuron is irrelevant. This principle underpins our two metrics: CLC, which measures a neuron's contribution to clean sample performance, and BLC, which quantifies its influence on decoy-induced misclassifications. Accordingly, a neuron is deemed suspicious if its removal substantially increases the backdoor loss while having only a minimal impact on the clean task.

To prioritize pruning, a scoring function is defined as follows:

$$\text{score}(\theta, i, j) = \text{BLC}(\theta, i, j) - \text{CLC}(\theta, i, j) \tag{5}$$

A higher score indicates greater relevance to the backdoor and less to the clean classification, making the neuron a preferred candidate for pruning.

The defense procedure begins by computing CLC values using clean data. Decoy samples are then generated using the optimal perturbation δ^*, and BLC values are calculated. Neurons

are pruned based on the score defined above. This approach ensures that backdoor-related neurons are removed while essential neurons for the primary task are preserved.

3.3 Post-Pruning Fine-tuning

Although pruning reduces backdoor behavior, excessive pruning may degrade model accuracy on clean inputs. To address this, the pruned model is fine-tuned using the available clean data. Given the limited dataset, overfitting becomes a potential risk during fine-tuning. To mitigate this, L2 regularization is applied to penalize large parameter values and promote generalization. This regularization stabilizes the learning process and helps the model retain its performance on clean classification tasks after pruning.

4. Experiments

This section evaluates the effectiveness of the proposed defense framework, PBITD, through a series of experiments. The study is organized into three stages: (1) verifying whether the proposed perturbation method can reliably induce backdoor behavior in infected models, (2) identifying and pruning backdoor-related neurons based on BLC and CLC, and (3) performing post-pruning fine-tuning and comparing performance with existing defense methods under various attack settings.

4.1 Inducing Backdoor Behavior with Perturbations

To assess whether the proposed perturbation strategy can expose backdoor behavior, experiments were conducted using PreAct-ResNet18 on CIFAR-10, with 20% of clean training data for validation. All-to-one dirty-label attacks were applied using BadNet and Trojan, both targeting class label 0. The Trojan attack was tested with two different trigger shapes: a square and an apple, to examine generalizability beyond square-based patterns. A randomly colored mosaic patch was used as the perturbation, following the optimization procedure outlined in Section 3.1.

For our backdoor detection on clean, BadNet, and Trojan-infected models, Figure 2 contrasts authentic triggers with perturbation-induced "decoy samples". Table 1 provides the quantitative validation, reporting the resulting misclassification rate (MR) and predicted label distribution.

In the clean model, the perturbation caused misclassification in only 17.5% of inputs, with predictions scattered across all ten classes. In contrast, the infected models misclassified nearly 90% of inputs, with the majority redirected to the attacker-defined target class (label 0). The MR does not reach 100% due to samples already belonging to class 0, which are excluded from the misclassification count. These results confirm that the perturbation can consistently trigger the embedded backdoor behavior, demonstrating its effectiveness in revealing all-to-one attacks.

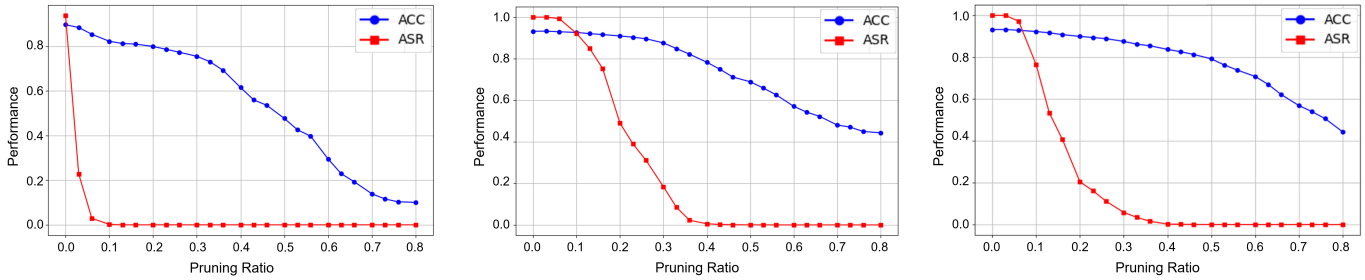
Figure 2. Comparison between the authentic backdoor samples (top row) and the samples to which applied the optimal perturbation (bottom).

TABLE 1. Misclassification Rate and Distribution of Predicted Labels.

Model	**MR**	**Distribution of Predicted Labels**			
Clean model	17.50%	Label 0	110	Label 5	5
		Label 1	76	Label 6	1
		Label 2	261	Label 7	30
		Label 3	1236	Label 8	16
		Label 4	6	Label 9	9
Trojan (square)	89.95%	Label 0	8994	Label 5	0
		Label 1	0	Label 6	0
		Label 2	1	Label 7	0
		Label 3	0	Label 8	0
		Label 4	0	Label 9	0
Trojan (apple)	88.39%	Label 0	8839	Label 5	0
		Label 1	0	Label 6	0
		Label 2	0	Label 7	0
		Label 3	0	Label 8	0
		Label 4	0	Label 9	0
BadNet	89.90%	Label 0	8990	Label 5	0
		Label 1	0	Label 6	0
		Label 2	0	Label 7	0
		Label 3	0	Label 8	0
		Label 4	0	Label 9	0

4.2 Neuron Pruning Guided by BLC and CLC

Following the detection of backdoor behavior, BLC and CLC scores were computed using decoy and clean samples, respectively. To maintain architectural integrity and computational



(a) BadNet **(b)** Trojan (square) **(c)** Trojan (apple)

Figure 3. Pruning defense performance under various attacks: (a) BadNet, (b) Trojan (square), and (c) Trojan (apple).

TABLE 2. Comparison of defense performance (ACC% / ASR%) on CIFAR-10 under different attacks. PF denotes Pruning followed by Fine-tuning. Blue and red numbers indicate the clean classification accuracy (ACC) and the attack success rate (ASR), respectively.

Attack	ANP	CLP	FP	I-BAU	NAD	NC	NPD	MNP	PBITD(Ours)	
									Pruning	PF
badnet(target=0)	91.31/1.27	94.07/5.61	91.91/0.90	85.8/5.08	89.17/1.65	90.27/1.62	89.16/0.35	86.78/4.44	91.06/1.67	90.08/**0.12**
badnet(target=1)	88.76/0.52	89.66/6.84	91.83/1.24	88.01/0.57	89.7/3.69	90.13/5.13	89.64/0.7	83.83/4.83	90.9/0.833	90.13/**0.05**
trojan(square)	87.23/0.37	74.43/7.57	92.5/61.89	88.98/1.83	91.29/3.58	92.07/1.72	91.7/23.12	80.02/15.22	82.77/1.01	89.21/**0.06**
trojan(apple)	87.05/0.44	78.95/17.27	92.43/73.74	90.57/2.23	91.59/3.48	92.37/3.44	92.12/15.91	82.71/14.61	83.57/0.21	88.06/**0.02**

efficiency, pruning was limited to the two convolutional layers in block 3.1 of PreAct-ResNet18.

Figure 3 illustrates the defense performance across three attack scenarios. The red line shows the Attack Success Rate (ASR) on infected inputs, while the blue line shows clean classification accuracy (ACC). In all cases, ASR sharply declined as the pruning ratio increased. For BadNet, ASR dropped below 1% before reaching a 20% pruning ratio, demonstrating high pruning efficiency. While Trojan attacks required slightly more pruning to suppress ASR, clean ACC remained above 80% even after pruning, showing that essential neurons for the clean task were preserved.

The blue line represents the model's classification accuracy (ACC) on the clean test dataset, while the red line indicates the Attack Success Rate (ASR) on the infected test dataset. As observed in the figure, the ASR drops to nearly zero before the pruning ratio reaches 40%. The defense is most effective against the BadNet attack, where our method induces a rapid decline in ASR with a pruning ratio of less than 20%. At the inflection point where the ASR approaches zero, the test ACC across all three experiments experiences a slight decrease compared to the unpruned model's accuracy (approx. 90%), yet it remains above 80%.

4.3 Effect of Post-Pruning Fine-tuning

To further enhance performance, fine-tuning was applied after pruning using the available clean dataset. L2 regularization was used to prevent overfitting due to limited data. Table 2 includes results for both pruning-only and post-fine-tuning ("PF") versions of PBITD.

Fine-tuning led to consistent ASR reductions across all experiments. In most cases, ASR dropped below 0.1%, while ACC was restored to approximately 90%. Although regularization helped mitigate overfitting, the absence of full training data occasionally constrained the peak performance. Nonetheless, the results demonstrate that fine-tuning is effective in compensating for pruning-induced degradation while maintaining high defense efficacy.

5. Conclusions

This paper proposed a fine-grained, neuron-level backdoor defense framework that detects and mitigates backdoor behavior in pre-trained models using only a small clean dataset. By introducing localized perturbations to clean inputs, the method effectively induces decoy samples that reveal hidden backdoor behavior without requiring prior knowledge of the trigger.

Backdoor-related neurons are then identified and pruned based on the proposed Backdoor Loss Change (BLC) and Clean Loss Change (CLC) metrics, ensuring targeted mitigation with minimal impact on clean task performance. Experimental results on CIFAR-10 demonstrate that the proposed method achieves competitive accuracy while significantly reducing attack success rates, outperforming several existing defenses under both BadNet and Trojan attacks. The inclusion of post-pruning fine-tuning further enhances robustness without sacrificing generalization. Overall, the framework offers a lightweight and practical solution for secure model deployment in backdoor-prone environments.

References

[1] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[2] A. Warnecke, J. Speith, J.-N. Möller, K. Rieck, and C. Paar, "Evil from within: Machine learning backdoors through hardware trojans," *arXiv preprint arXiv:2304.08411*, 2023.

[3] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[4] X. Qi, T. Xie, J. T. Wang, T. Wu, S. Mahloujifar, and P. Mittal, "Towards a proactive ML approach for detecting backdoor poison samples," in *32nd USENIX Security Symposium (USENIX Security 23)*, Aug. 2023, pp. 1685–1702.

[5] H. Huang, Y. Wang, S. Erfani, Q. Gu, J. Bailey, and X. Ma, "Exploring architectural ingredients of adversarially robust deep neural networks," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 5545–5559.

[6] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: Training clean models on poisoned data," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 14 900–14 912.

[7] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, "Backdoor defense via decoupling the training process," 2022.

[8] B. Wu, S. Wei, M. Zhu, M. Zheng, Z. Zhu, M. Zhang, H. Chen, D. Yuan, L. Liu, and Q. Liu, "Defenses in adversarial machine learning: A survey," 2023.

[9] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 707–723.

[10] Y. Zeng, S. Chen, W. Park, Z. M. Mao, M. Jin, and R. Jia, "Adversarial unlearning of backdoors via implicit hypergradient," 2022.

[11] N. Li, H. Jiang, and P. Yi, "Magnitude-based neuron pruning for backdoor defens," 2024.

[12] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 101–105.

[13] D. Wu and Y. Wang, "Adversarial neuron pruning purifies backdoored deep models," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 16 913–16 925.

[14] R. Zheng, R. Tang, J. Li, and L. Liu, "Data-free backdoor removal based on channel lipschitzness," in *Computer Vision – ECCV 2022*, 2022, pp. 175–191.

[15] Y. Zeng, S. Chen, W. Park, Z. M. Mao, M. Jin, and R. Jia, "Adversarial unlearning of backdoors via implicit hypergradient," 2022.

[16] Y. Li, X. Lyu, X. Ma, N. Koren, L. Lyu, B. Li, and Y.-G. Jiang, "Reconstructive neuron pruning for backdoor defense," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202, 23–29 Jul 2023, pp. 19 837–19 854.

[17] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," 2021.

[18] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Research in Attacks, Intrusions, and Defenses*, 2018, pp. 273–294.

[19] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. 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.