Combine Neural Cleanse with K-means Clustering for Detecting All-to-one Parallel Multi-trigger Backdoor Attacks

Ming Zhu
Pennsylvania State Univ.
University Park, PA 16802
mjz5281@psu.edu

Abstract

Backdoor attacks pose a serious threat to the trustworthiness of deep neural networks (DNNs). Among these, multi-trigger backdoor attacks introduce significant challenges due to their complexity and evasion techniques. This paper proposes a novel framework combining Neural Cleanse (NC) with K-means clustering to detect all-to-one parallel multi-trigger backdoor attacks effectively. By reverse-engineering minimal perturbations and leveraging clustering, this approach identifies distinct trigger patterns while addressing the limitations of existing defenses in handling multi-trigger scenarios. Comprehensive experiments conducted on CIFAR-10 demonstrate the promising efficacy of the method in differentiating clean models, single-trigger backdoors, and multi-trigger backdoors. This work contributes to advancing post-training defenses against sophisticated plural backdoor attacks. Our code is available at: Colab Link

1 Introduction

Backdoor attacks pose a significant threat to the security and trustworthiness of deep neural networks (DNNs)[10]. Among these, unlike traditional single-trigger backdoor attacks, multi-trigger backdoor attacks[9] represent a sophisticated adversarial tactic where multiple triggers independently lead to a single target label. While existing methods like Neural Cleanse (NC)[13] have demonstrated effectiveness in detecting simple single-trigger backdoors by reverse-engineering potential triggers, their performance degrades when faced with complex multi-trigger scenarios. Moreover, approaches such as CEPA[8] and other reverse-engineering-based defenses have shown promise in identifying shared patterns across poisoned samples but struggle to generalize across diverse attack settings.

This research proposes an integrated approach that combines Neural Cleanse's reverse-engineering capabilities with K-means clustering[6] to enhance the detection of all-to-one parallel multi-trigger backdoor attacks[9]. The objective is to leverage Neural Cleanse's ability to identify minimal perturbations for triggering backdoors while using clustering in the perturbation space to distinguish independent triggers. This combination not only addresses the limitations of Neural Cleanse in detecting complex attacks but also introduces an efficient mechanism to uncover distinct trigger patterns in an all-to-one attack setting.

Through this novel framework, we aim to bridge the gap between reverse-engineering and clustering-based methodologies, ensuring robust detection of all-to-one parallel backdoors. By systematically evaluating this hybrid approach against clean models, single-trigger backdoors, and multi-trigger backdoors (specifically double triggered), this work seeks to advance the state-of-the-art backdoor defense in multi-trigger cases, offering enhanced interpretability, scalability, and accuracy across a variety of attack scenarios.

The remainder of this paper is organized as follows:

- Section 2 reviews related work, focusing on backdoor attack mechanisms and existing defenses.
- Section 3 presents the proposed method, detailing its integration of NC with K-means clustering.
- Section 4 outlines the experimental setup and results, including detection performance across various attack scenarios.
- Section 5 concludes the paper and discusses potential directions for future research.

2 Related work

Backdoor attacks have emerged as a critical threat to the security of deep neural networks (DNNs), enabling adversaries to manipulate model behavior by embedding malicious triggers during training. These attacks can be broadly categorized into single-trigger backdoor attacks and multi-trigger backdoor attacks[9], each presenting unique challenges to existing defenses. In this section, we provide an overview of these attack types and discuss current post-training reverse-engineering-based backdoor detection methods.

2.1 Backdoor Attacks

Single-trigger backdoor attacks are among the earliest and most studied forms of backdoor attacks. In these attacks, a single, fixed trigger is embedded in a subset of the training data, such that the presence of the trigger in input samples causes the model to consistently predict a specific target label[3]. Popular backdoor attacks such as the BadNet[1] use fixed patterns (e.g., a small patch or a specific pixel modification) to create backdoored models that function normally on clean data but misclassify triggered inputs. While effective, these attacks are easily identifiable by certain defenses that rely on reverse-engineering or anomaly detection to identify a single, consistent pattern of misbehavior[8].

Multi-trigger backdoor attacks significantly increase the sophistication and evasiveness of backdoor strategies. These attacks are particularly challenging to detect, as the triggers may be spatially, structurally, or functionally distinct. A key feature of multi-trigger attacks is their ability to introduce diverse triggers that either operate independently or interact in more complex patterns, enabling them to bypass traditional single-trigger-focused defenses. Three primary poisoning strategies characterize multi-trigger backdoor attacks[9]:

- Parallel Attacks: In parallel attacks, independent adversaries introduce distinct triggers into different subsets of the training data simultaneously. Each adversary independently inserts a unique trigger or a group of triggers, ensuring no overlap between their poisoned data portions. Specifically, this paper will focus on this type of attacks, all-to-one parallel multi-trigger backdoor, where triggers can be spatially or structurally distinct, making detection more challenging.
- Sequential Attacks: Sequential attacks involve a series of adversaries sequentially poisoning the same subset of training samples. Each adversary injects one or more triggers into these samples, stacking triggers on top of each other.
- Hybrid Attacks: Hybrid attacks combine elements of both parallel and sequential strategies, often involving collusive adversaries.

2.2 Backdoor Defenses

Backdoor defenses can be broadly classified into two categories: backdoor detection and backdoor removal (or mitigation)[9]. Detection methods focus on identifying whether a given model has been backdoored or whether a specific input sample contains a backdoor trigger. These methods are critical for assessing model integrity and ensuring safe deployment in real-world applications.

Detection mechanisms can further be divided into approaches that directly analyze the model for backdoor presence[13] and those that examine individual samples for potential triggers[12]. While not all backdoor detection defenses employ reverse-engineering techniques, several reverse-engineering-based methods have been proposed[8], relying on a small, clean (unpoisoned and correctly labeled) dataset that may be independent of the original training data. Examples include methods like Neural

Cleanse (NC)[13], which reverse-engineer minimal perturbations to identify potential backdoors, and other approaches that aim to pinpoint triggers by reconstructing their patterns.

One reverse-engineering detector is UNICORN[14], which is designed to be agnostic to the mechanism by which backdoors are incorporated into the model. UNICORN focuses specifically on all-to-one backdoor attacks and leverages two unpoisoned UNet feature maps per target class to create a feature space from the model's input space. In this feature space, backdoor patterns, such as patch-based triggers (as in BadNet) or blended triggers (as in Neural Cleanse), are expected to manifest. However, UNICORN's effectiveness depends on the availability of suitable unpoisoned feature maps or requires substantial computation to train these maps using clean data. This dependency becomes particularly challenging when dealing with models with a large number of output classes, as the computational overhead for reverse-engineering triggers increases significantly.

While reverse-engineering-based defenses, including Neural Cleanse and UNICORN, have demonstrated effectiveness against traditional backdoor attacks, their reliance on clean datasets and assumptions about trigger patterns can limit their performance against more sophisticated multi-trigger backdoor scenarios. This highlights the need for adaptive detection mechanisms that can operate efficiently across diverse attack strategies without heavy computational dependencies.

Our proposed method addresses this gap by integrating Neural Cleanse's reverse-engineering capabilities with K-means clustering to enhance detection accuracy for all-to-one parallel multi-trigger backdoor attacks. By integrating these approaches, our goal is to enhance the efficacy of post-training defenses in countering this complex attack model.

3 Proposed Multi-trigger Backdoor Detection Method

This section introduces our proposed method, which integrates Neural Cleanse (NC) with K-means clustering for detecting all-to-one parallel multi-trigger backdoor attacks. The framework is designed to reverse-engineer triggers for potential backdoor patterns and employs clustering techniques to distinguish between clean and malicious triggers. Below, we detail the core components of the approach: reverse-engineering triggers, outlier detection, clustering perturbations, and backdoor analysis.

3.1 Trigger Reverse Engineering

We extend Neural Cleanse to reverse-engineer potential triggers by searching for minimal perturbations in the input space that force the model to misclassify inputs into a specific target label. This process is formalized as an optimization problem, where the objective combines two terms: (1) the misclassification loss, which ensures the perturbed inputs lead to the target label, and (2) a sparsity regularization term to minimize the size of the perturbation. The optimization is performed iteratively for a fixed number of steps using the Adam optimizer. The output for each target label is a reverse-engineered mask (defining the region of the trigger) and a trigger pattern.

Mathematically, the optimization objective is:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(f(x_{\text{perturbed}}), y_{\text{target}}) + \lambda \|\text{mask}\|_1,$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, λ is the regularization weight, and mask controls the sparsity of the perturbation[13].

3.2 Outlier Detection

After reverse-engineering triggers for all target labels, we compute the L1 norm of each trigger mask to measure its size. To detect anomalous (potentially malicious) triggers, we apply a Median Absolute Deviation (MAD)-based outlier detection method[4]. This step identifies triggers with significantly smaller masks compared to the majority, as backdoor triggers are expected to exhibit such anomalies.

The outlier detection process involves:

- Computing the median size of all reverse-engineered masks.

- Calculating the MAD to estimate the deviation of mask sizes.
- Identifying masks whose size deviates from the median by a threshold value.

Triggers identified as outliers are flagged for further analysis. If no significant outliers are detected, the model is considered clean, and the detection process terminates early.

3.3 Dynamic Clustering with K-means

For cases with detected outliers, we dynamically adjust the number of clusters in K-means based on the number of outliers. At least two clusters are initialized to distinguish between clean and malicious patterns, but additional clusters may be created for cases with multiple distinct triggers.

Using K-means, we cluster the perturbations (flattened reverse-engineered masks) into groups. Each cluster represents a distinct trigger pattern or clean behavior. By analyzing the resulting clusters, we identify potential backdoor patterns based on their distribution and properties.

3.4 Visualization and Analysis

To enhance interpretability, we visualize the reverse-engineered triggers and their corresponding masks. Each mask and trigger is labeled based on its cluster assignment, allowing us to observe and interpret the spatial and structural patterns of backdoor triggers. Additionally, the cluster sizes and average perturbation norms are computed to quantify and analyze the results.

Finally, we analyze the clustering results to detect:

- Single clusters of similar triggers (indicating a clean model).
- One additional cluster with significantly smaller perturbations (indicating a single backdoor trigger).
- Multiple clusters with small perturbations (indicating multiple backdoor triggers).

4 Experiments and Results

4.1 Experimental Setup

Dataset. We conduct experiments on the CIFAR-10 dataset, a standard benchmark in backdoor attack and defense research. CIFAR-10 contains 60,000 32x32 RGB images categorized into 10 classes, with 50,000 training images and 10,000 test images[7]. For all experiments, the dataset is preprocessed by normalizing pixel values to the range [0, 1].

Training setup. We use a ResNet-18 model[5] as the base architecture for all experiments. The model is trained on the CIFAR-10 training set using the following configuration:

- Optimizer: SGD with momentum
- Learning rate: 0.01
- Batch size: 128
- Number of epochs: 50

Attack Setup. The experimental attack setup involves injecting backdoor triggers into training datasets to evaluate the proposed detection method under various attack scenarios. Three distinct triggers and their combinations are used to simulate backdoor attacks: BadNet[2] Trigger, (local) Chessboard Trigger, and WaNet[11] Trigger. These configurations target the CIFAR-10 dataset with a focus on all-to-one attacks.

Single-Trigger Attacks

- BadNet Trigger: A fixed white pixel patch is inserted into the bottom-right corner of the image. This is a classical backdoor attack technique known for its simplicity and effectiveness.

- Chessboard Trigger: A 2x2 alternating pattern of black and white pixels is added to the top-left corner of the image. This introduces a structured pattern as the trigger.
- WaNet Trigger: An affine transformation is applied to the entire image to introduce a subtle and imperceptible perturbation. This approach blends the trigger seamlessly with the input.

For each single-trigger attack, 10% of the training data is poisoned, and all poisoned samples are relabeled to a target class (arbitrarily set as class 9). The poisoned models are trained from scratch with the modified dataset. Single triggered images shown in Figure 1.

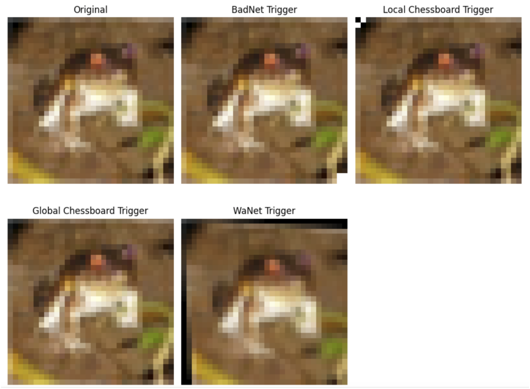


Figure 1: Examples of different types of backdoor triggers.

Multi-Trigger Attacks

To evaluate the proposed method against multi-trigger backdoor attacks, we implement parallel all-to-one multi-trigger attacks using combinations of the aforementioned triggers. Three configurations are considered:

- BadNet + Chessboard: The BadNet trigger is applied to one subset of the poisoned data, while the Chessboard trigger is applied to another. The two subsets are disjoint but collectively form 10% of the dataset.
- BadNet + WaNet: Similar to the first configuration, the BadNet and WaNet triggers are introduced into disjoint subsets, with a total poisoning rate of 10%.
- WaNet + Chessboard: Similar configuration.

For all configurations, the poisoned samples across different triggers are relabeled to the same target class (class 9). The poisoned models are trained from scratch to ensure that they learn the backdoor behavior introduced by the triggers.

Evaluation Metrics

The effectiveness of the backdoor attack is measured using:

- Clean Accuracy (CA): The classification accuracy on the clean test set.
- Attack Success Rate (ASR): The proportion of triggered test samples (for each trigger) misclassified to the target class.

Defense Setup.

The defense methodology combines Neural Cleanse (NC) with K-means clustering to detect backdoor triggers in models attacked with single or multi-trigger backdoor attacks. The setup leverages reverse-engineering to uncover potential triggers and clustering to analyze their behavior. Below are the specific configurations and hyperparameters used:

Reverse Engineering of Triggers

- Objective: Identify potential backdoor triggers for each target label.

Table 1: Attack success rates (ASR) and clean accuracy for different attack setups on CIFAR-10 dataset.

Attack Type	Trigger(s)	Poison Rate	Clean Acc (%)	ASR (%)
Clean Model	Clean	N/A	86.06	N/A
Single Trigger	BadNet	0.1	84.87	97.10
Single Trigger	Chessboard	0.1	84.87	100.00
Single Trigger	WaNet	0.1	85.04	99.82
Multi Triggers	{WaNet, BadNet}	0.1	84.53	98.14
Multi Triggers	{BadNet, Chessboard}	0.1	84.54	98.28
Multi Triggers	{WaNet, Chessboard}	0.1	84.87	99.86

- Loss Function: The optimization objective combines cross-entropy loss ($\mathcal{L}_{\text{CE}}$) with a sparsity regularization term:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(f(x_{\text{perturbed}}), y_{\text{target}}) + \lambda \|\text{mask}\|_1,$$

  where $\lambda = 0.075$ controls the weight of the sparsity term.
- Optimization Parameters: Learning Rate = 0.01, Optimizer: Adam, Number of Steps = 100.
- Regularization: L1 regularization on the perturbation mask to ensure the trigger remains minimal and localized.

Outlier Detection

- Metric: Median Absolute Deviation (MAD) is used to detect outliers based on the L1 norm of the reverse-engineered masks.
- Threshold: threshold = 3.0
- Process: Compute the L1 norm of each mask. Then calculate the median and MAD of the norms. And finally identify masks whose deviations exceed the threshold. If no significant outliers are detected, the model is considered clean, and further analysis is terminated.

K-Means Clustering

- Objective: Cluster the perturbations into groups to differentiate triggers and identify multi-trigger scenarios.
- Dynamic Clustering: The number of clusters n is dynamically adjusted based on the number of detected outliers: n = max(2, number of outliers+1).
- Implementation: K-means clustering is applied to the perturbation masks, reshaped into a flattened feature vector for analysis.

4.2 Backdoor Detection Performance

This section presents the detection performance of the proposed NC+K-means method across different attack types, focusing on detection success rates, speculated types, and cluster separation metrics. Moreover, example visualization and thorough analysis will be presented. The evaluation compares clean and poisoned models subjected to single-trigger and multi-trigger attacks.

Detection Failures and Robustness.

Despite somehow effective detection performance overall, certain scenarios resulted in ambiguous or incorrect classifications:

- Single-Trigger Attacks: Models trained with WaNet triggers were misclassified as clean. This likely stems from the high imperceptibility of the WaNet pattern, which introduces minimal perturbations that blend well into the input space. Also, single-trigger or multi-trigger attacks using the local chessboard pattern were frequently detected as multi-trigger attacks. This misclassification can be attributed to the discontinuous nature of the chessboard pattern, which can create localized clusters of high perturbation norms that resemble multi-trigger scenarios during clustering.

Attack Type	Trigger	Detection	Speculated Type
Clean model	N/A	succeed	clean
Single Trigger	BadNet	succeed	single
Single Trigger	Chessboard	fail	multi
Single Trigger	WaNet	fail	clean
Multi Triggers	[WaNet, BadNet]	fail	single
Multi Triggers	[BadNet, Chessboard]	succeed	multi
Multi Triggers	[WaNet, Chessboard]	succeed	multi

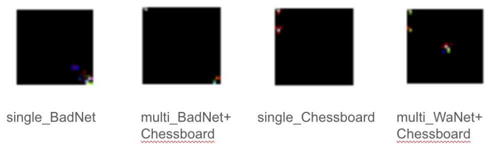
Table 2: Detection results of the proposed NC+K-means method across various attack types and triggers. The table specifies the detection outcome (success or failure) and the speculated trigger type (clean, single, or multi) for each evaluated model configuration.

- Multi-Trigger Overlaps: In some cases, multi-trigger attacks with overlapping perturbations led to the detection of a single-trigger type due to insufficient separation in the clustering step.
- Clean-Label Ambiguities: Clean models exhibited occasional misclassification as backdoored under high regularization ($\lambda = 0.1$) or low steps (like 1 or 10), highlighting the importance of balancing λ during optimization.

Hyperparameter Sensitivity: Adjusting the λ regularization weight significantly impacts mask sparsity and detection outcomes. Additionally, the number of steps (100 vs. 200) was found to marginally affect cluster separation. Fine-tuning these parameters remains crucial for balancing detection accuracy and computational efficiency.

Trigger Visualization.

A key aspect of our detection method involves reverse-engineering triggers and analyzing their characteristics both visually and quantitatively. Figure 2 presents examples of reverse-engineered triggers for various attack configurations, including single-trigger (e.g., BadNet, Chessboard) and multi-trigger (e.g., BadNet + Chessboard, WaNet + Chessboard) attacks. These visualizations highlight the distinct patterns introduced by each trigger.



single_BadNet

multi_BadNet+Chessboard

single_Chessboard

multi_WaNet+Chessboard

Figure 2: Examples of different reverse-engineered masks.

Quantitatively, the reverse-engineered triggers are evaluated using the L1 norm of their perturbations. In the context of backdoor detection, it quantifies the average magnitude of the perturbation introduced by the reverse-engineered trigger mask. The L1 norm is calculated as the sum of the absolute values of all elements in the mask, reflecting its sparsity and magnitude. Table 3 summarizes the average L1 norms and cluster sizes observed during detection.

Attack Type	Trigger	Avg Pert. Norm	Avg Cluster Size
Single Trigger	BadNet	52.13	5
Single Trigger	Chessboard	49.29	3.33
Single Trigger	WaNet	N/A	N/A
Multi Triggers	[WaNet, BadNet]	49.41	5
Multi Triggers	[BadNet, Chessboard]	36.47	2.5
Multi Triggers	[WaNet, Chessboard]	48.21	3.33

Table 3: Average perturbation norms (L1 norm) and cluster sizes for different backdoor attack types.

Discussion and Analysis.

The proposed method demonstrates partial effectiveness, particularly against simple additive backdoors like BadNet and local chessboard patterns. For these backdoor types, the reverse-engineered triggers closely resemble the actual attack patterns, and the K-means clustering aligns with the hypothesis, successfully separating clean models from poisoned ones and correctly identifying single-trigger scenarios. The detection results for these additive patterns confirm the utility of the approach in handling relatively straightforward backdoor mechanisms.

However, the method struggles with imperceptible or complex patterns, such as those introduced by WaNet. In these cases, the reverse-engineered triggers often fail to exhibit distinguishable patterns that align with the injected perturbations, leading to frequent misclassification. WaNet attacks, characterized by subtle geometric transformations rather than additive triggers, pose a significant challenge due to the difficulty in separating their perturbations from the clean data cluster. For instance, WaNet was occasionally misclassified as a clean model, indicating the method's limitations in detecting such stealthy backdoors.

Additionally, multi-trigger scenarios presented mixed results. While the method succeeded in clustering and identifying certain multi-trigger configurations (e.g., BadNet and chessboard), it failed to consistently classify overlapping or imperceptible patterns. This highlights the need for further refinement, particularly in handling hybrid and sequential multi-trigger attacks.

In summary, while the proposed method offers promising results against simple additive backdoors, its effectiveness diminishes significantly for imperceptible patterns and complex multi-trigger scenarios. This underscores the necessity of incorporating advanced reverse-engineering techniques or alternative clustering strategies to improve detection robustness across diverse attack configurations.

Effect of Clean Accuracy and Poison Rate.

To further understand the robustness of the detection approach, we investigated the effect of training clean accuracy (CA) and poison rate (PR) on detection outcomes:

- Impact of Clean Accuracy: Increasing the CA from approximately 70%+ to 80%+ during model training led to improved detection accuracy across all attack types. This improvement is attributed to better model generalization and reduced noise in reverse-engineered masks.
- Effect of Poison Rate: Lowering the PR from 10% to 1% had minimal impact on ASR, with a slight increase in CA observed for poisoned models. The reduced poisoning fraction slightly affected the visibility of the triggers but did not significantly alter the clustering results.

Attack Type	Trigger	Poison Rate	Clean Acc (%)	ASR (%)	Epochs
Single Trigger	BadNet	0.01	85.08	91.27	50
Single Trigger	BadNet	0.1	84.87	97.1	50
Multi Triggers	[BadNet, Chessboard]	0.01	85.23	91.29	50
Multi Triggers	[BadNet, Chessboard]	0.1	84.54	98.28	50

Table 4: Performance comparison of single-trigger and multi-trigger backdoor attacks with varying poison rates.

4.3 Time Complexity

Our experiments utilize L4 GPU with 22.5GB of GPU RAM on Google Colab. Training a clean or poisoned Resnet-18 on CIFAR-10 for 50 epochs takes approximately 15 minutes. The proposed method using the hyper parameters documented requires approximately 1 hour to perform detection on a Resnet-18 model trained on CIFAR-10.

5 Conclusion and Future Directions

This paper introduces a hybrid defense framework that combines Neural Cleanse (NC) and K-means clustering to detect all-to-one parallel multi-trigger backdoor attacks. By leveraging reverse-engineering to identify perturbation patterns and clustering to distinguish triggers, the proposed method demonstrates effectiveness against simple additive backdoor mechanisms like BadNet and local chessboard patterns. However, it struggles with imperceptible transformations such as WaNet, highlighting areas for improvement.

While the results confirm the potential of the NC+K-means approach in mitigating certain backdoor attacks, challenges remain in handling subtle and complex attack patterns. Future work will focus on enhancing or changing clustering techniques to address imperceptible triggers and extending the framework to detect sequential or hybrid multi-trigger backdoors. Additionally, restricted by computation resource, only a few types of triggers were studied. More different triggers and combinations can be studied in the future, not limited to double-trigger. This research serves as a step toward more comprehensive defenses against evolving backdoor threats.

6 Acknowledgment

The author(s) sincerely thank Professor George Kesidis for his invaluable guidance and support in helping formulate the research direction. His insightful suggestions and expertise were instrumental in shaping the foundation of this work.

References

[1] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and S Garg BadNets. Evaluating backdooring attacks on deep neural networks., 2019, 7. *DOI: https://doi. org/10.1109/ACCESS*, pages 47230–47244, 2019.

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

[3] Wei Guo, Benedetta Tondi, and Mauro Barni. An overview of backdoor attacks against deep neural networks and possible defences. *IEEE Open Journal of Signal Processing*, 3:261–287, 2022.

[4] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Thomas Jurczyk. Clustering with scikit-learn in python. *The Programming Historian*, 2021.

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[8] X. Li, H. Wang, D. J. Miller, and G. Kesidis. Universal post-training reverse-engineering defense against backdoors in deep neural networks. *arXiv.org*, May 23 2024. arXiv:2402.02034.

[9] Y. Li, J. He, H. Huang, J. Sun, X. Ma, and Y.-G. Jiang. Shortcuts everywhere and nowhere: Exploring multi-trigger backdoor attacks. *arXiv.org*, November 28 2024. arXiv:2401.15295.

[10] Y. Li, X. Lyu, X. Ma, N. Koren, L. Lyu, B. Li, and Y.-G. Jiang. Reconstructive neuron pruning for backdoor defense. *arXiv.org*, December 8 2023. arXiv:2305.14876.

[11] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

[12] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

[13] 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)*, pages 707–723, 2019.

[14] Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. Unicorn: A unified backdoor trigger inversion framework. *arXiv preprint arXiv:2304.02786*, 2023.