Analysis of Neural Trojan Attacks and Defenses with Intuition Proposals

Efi Kafali, Aimilia Palaska

October 2, 2024

Abstract

AI applications and services, despite their outstanding development, harbor significant risks, particularly as they are increasingly employed in complex and high-stakes decision-making areas. A notable threat is Backdoor Attacks on Neural Networks, commonly referred to as Neural Trojans. These attacks can evade human detection by embedding triggers during the training process, guaranteeing adversary action when the trigger is present in the input but preserving normal behavior when it is not. In this report, we discuss state-of-the-art (SOTA) approaches for Neural Trojans, including both attack, detection and removal techniques. We review, analyze, and contrast the defense methods and their respective outcomes. Additionally, we experiment with a detection-mitigation pipeline on a chest X-ray dataset and examine the effectiveness of SOTA detection methods at both the data and model level. Relative scripts developed can be located in a GitHub Repository [1]

1 Introduction

Over the past decade, Artificial Intelligence (AI) has experienced remarkable advancements, delivering innovative solutions in STEM, linguistics, healthcare, transportation, and art [2, 3, 4, 5, 6]. Currently, the predominant approach for training and deploying models is through Deep Neural Networks (DNNs), which offer versatility for numerous classification-related problems. [7]. As DNNs continue to evolve and become integrated into more high-stakes fields, often they necessitate substantial computational resources that may not be available to all individuals or organizations. Consequently, many rely on pre-trained models [8] provided by third-party sources when proprietary models are unavailable.

This practice introduces complex ethical considerations. On one hand, such accessibility democratizes technology and promotes innovation across diverse sectors. However, it poses notable risks. Adversaries can exploit the open nature of these resources by embedding malicious behaviors within pre-trained models, subsequently distributing them as benign [9]. This threat is exacerbated by the tendency of users to prioritize straightforward performance metrics, such as accuracy, while overlooking potential security vulnerabilities. The impact of such attacks [10] can range from harmful yet straightforward incidents, like a hate-speech episode, to more serious occurrences, like a misdiagnosis or a sensitive data breach [3]. Consequently, backdoor attacks represent a tangible and significant threat in practical applications of these models.

Particularly alarming are Neural Trojan attacks [11], commonly refereed to as Backdoor Attacks, which manipulate the networks behavior through a trigger. In this scenario, the attacker embeds a trigger on a fraction of the training dataset and associates them with their desired target class before the training phase. The result is a misclassification, either from one class to the target class (Class Specific Attack) or from all the classes to the target class (Class Agnostic Attack) [11, 12, 13]. A simple example of this is shown in Figure 1. What makes this attack dangerous is the seemingly normal behavior of the model when the trigger is absent in the input, as well as the attacker's capability to manipulate under which circumstances the malicious behavior will be activated [13].

We experiment with a practical and effective new pipeline tested primarily on two datasets [14, 15]. The pipeline is designed to mitigate the effects of neural Trojan attacks by incorporating detection mechanisms at both the model and data levels. At the data level, the pipeline ensures that any potential triggers embedded in the stored training datasets are accurately identified, pinpointing the exact locations of harmful samples that could compromise the integrity of an AI model during training. At the model level, it leverages a Trojan detection method that operates during inference, allowing the identification of models that have already been compromised by embedded triggers.

It consists of a simple and quick backdoor attack detection, followed by a data cleansing method to acquire which input images were in-

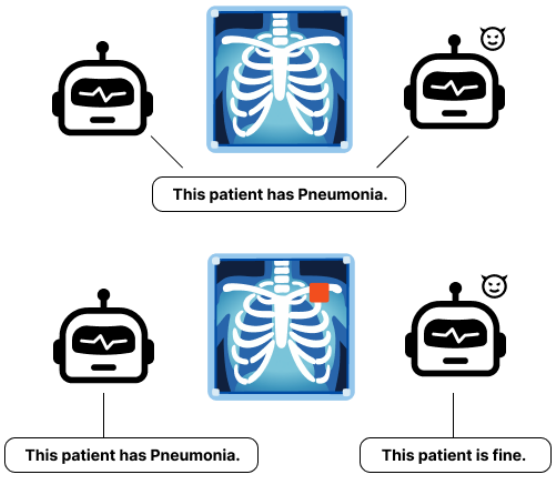

Figure 1: General understanding of a Neural Trojan attack. The clean model classifies the image correctly in both cases, whereas the backdoored model misclassifies the image with the trigger as the adversary target class.

fected. After they are removed, the model is tuned to forget the trigger associations and the model is considered clean. While not the most exquisite route to mitigate the problem, it provides a quick and practical approach, successfully handling three basic types of trigger while maintaining the computational cost as low as possible.

This report proceeds as follows: in Sec. 2 we demonstrate existing SOTA approaches for Neural Trojan Attacks, Detection and Mitigation and in Sec. 3 we propose some potential approaches for detection and mitigation. Sec. 4 contains the experimental setting, Sec. 5 presents our results, Sec. 6 raises themes open to discussion and Sec. 7 provides our conclusions.

2 Related Work

In this section, we examine the strategies employed by some cutting-edge methods for addressing Neural Trojans. Our analysis covers a range of approaches in attack, detection, and mitigation, highlighting the strengths and limitations of each. We selected a diverse set of state-of-the-art implementations, including both highly cited works and newer, innovative methods that have yet to be thoroughly explored. In Table 1 we present an overview of the assumptions made by each SOTA regarding the defender's access to the model and training data.

2.1 Threat Model

To better understand the threat model, we need to clarify some of the most widespread attack

Method	Attack	Data	Model
MDTD	✗	✓	✗
TAD	✗	✗	✗
DeepInspect	✗	✗	✗
FreeEagle	✗	✗	✓
NeuralSanitizer	✗	✓	✓
MM-BD	✗	✗	✓
Potion	✗	✓	✓
Game of Trojans	✓	-	-
LoneNeuron	✓	-	-
BppAttack	✓	-	-

Table 1: Information about the specific requirements of the evaluated SOTA methods. The term "Attack" refers to whether the proposed method introduces a novel attack type. "Data" denotes the necessity of even a small fraction of clean or poisoned samples. "Model" specifies whether the method is applied in a black-box or white-box context.

and defense scenarios. Attackers are typically classified based on their level of access to the model and their expertise. This distinction is critical when considering Neural Trojan defense mechanisms, as proposed in [11].

Freedom: An attacker might control the entire training process, including the dataset, or they might influence only parts of it. In [9], various scenarios have been examined, considering different levels of adversary proficiency. Even if an attacker lacks the expertise to create a backdoored model from scratch, they could download a pre-trained model, fine-tune it to introduce the desired malicious behavior, and re-upload it with minimal computational effort. Our study focuses mainly on methods that grant full access to the adversary.

Goals: Despite the variety in adversaries' methods, their common goal is to maximize the success of their attacks while ensuring their modifications remain undetected. Most times, a metric called attack success rate (ASR) is considered in order to evaluate this behavior. It is defined as the number of poisoned samples correctly classified as the adversary target class over the total amount of poisoned data.

Detectability: The attacks can range from simple to highly adaptive [16, 17], with the latter involving knowledge of existing defense mechanisms and efforts to bypass them. In [13] it is argued that an adaptive attack can bypass the dipole "detectable vs effective" by turning it into an optimization problem. Specifically, a game of attacker vs defender can lead to maximizing the attack success rates without being detected by some of the SOTA methods.

On the other hand, defenders are categorized based on their access to the training data. Some defense methods are highly specific and perform

well under certain trigger injections while others succeed in broader applications. Defenders' objectives vary; some focus only on the detection of backdoored models [18, 12], while others extend their goals to mitigating the attacks [19, 20], a complex task involving the removal of malicious misclassifications while preserving the model's performance on clean samples.

2.2 Attack methods

Game of Trojans [13]: The key assumption challenged by this paper is that adversaries lack prior knowledge of the detection mechanisms and remain static. Instead, the authors propose an adaptive adversary that retrains the Trojaned DNN with knowledge of the detection methods, thus bypassing existing detection techniques. The pipeline of the proposed method involves a two-step iterative process where the adversary alternates between updating the Trojaned model and recalibrating the detection mechanism. First, the adversary uses a known trigger pattern to train the DNN, embedding the Trojan while considering the parameters of the detector. In the second step, the updated Trojaned DNN is used to adjust the detector's parameters to maximize its detection ability. This cycle is repeated until the adversary achieves both high classification accuracy on clean and Trojaned inputs while evading detection. The process is formalized as a min-max optimization problem, where the adversary and detector engage in a co-evolutionary "game," continuously adapting to each other's updates. The results demonstrate that existing detectors are inadequate against such adaptive threats, as the adversary consistently evades detection. The primary disadvantage is the increased computational complexity and potential overfitting risks associated with the iterative retraining process. Nevertheless, the method's effectiveness in experimental settings across multiple datasets underscores its potential to expose and address weaknesses in current defense strategies.

BppAttack [17]: Traditional Trojan attacks use visible triggers like patches or image transformations, making them susceptible to detection by human inspectors or defensive mechanisms. BppAttack, however, leverages image quantization and dithering to create nearly imperceptible triggers, exploiting the human visual system's insensitivity to small changes in color depth. This approach not only enhances stealth but also eliminates the need for auxiliary model training, which is often time-consuming and resource-intensive. To overcome the difficulty of injecting such subtle triggers during training, the authors propose a contrastive learning method that uses adversarial attacks to generate precise and effective triggers. The effectiveness of BppAttack is demonstrated through experiments on four benchmark datasets, where it achieves high attack success rates while bypassing state-of-the-art defenses and human detection. The method also proves to be resilient against various existing defensive measures, including runtime defenses and reverse engineering defenses. Overall, BppAttack represents a significant advancement in the field of Trojan attacks, highlighting both the vulnerabilities of current DNN systems and the challenges in developing robust defenses.

LoneNeuron [16]: The paper presents an innovative backdoor attack on DNNs that operates by embedding a Trojan neuron within the feature domain, specifically between the first convolution layer and the activation layer. This Trojan responds to invisible, sample-specific, and polymorphic pixel-domain watermarks, which are subtle perturbations that evade both human detection and automated defense mechanisms. LoneNeuron achieves a 100% attack success rate across various DNN architectures, including vision transformers (ViTs) [21], without compromising the performance of the main task. The attack's stealth is enhanced by the polymorphic nature of the watermarks, where a single feature-domain trigger can generate multiple pixel-domain watermarks, making detection through statistical or visual means extremely difficult. The attack is robust against common defenses such as fine-tuning and compression, and it highlights the critical security vulnerabilities in the machine learning model supply chain, where such Trojans can be easily introduced and spread unnoticed. The authors emphasize the need for stronger security measures and scrutiny in the sharing and reuse of machine learning models to mitigate these risks.

2.3 Detection methods

FreeEagle [12]: For FreeEagle's pipeline, the model in review is divided into two distinct components: the feature extractor and the classification module, with the division between these components being defined by the number of intermediate layers. FreeEagle generates a dummy intermediate representation for each class by utilizing the two segments of the model to maximize the posterior probability of the classifier. The maximized posteriors are then organized into a matrix format, and analyzed in regards to outliers, which indicate a potential Backdoor Attack. The advantages of FreeEagle include its data-free nature, its adaptability across various model architectures and its effective resistance against two adaptive attacks. However, there are notable limitations associated with this approach. The assumptions regarding trigger types may constrain its effectiveness against emerging attack methodologies and access to the white-box model is not always feasible, particularly for users of

third-party models. Lastly, the computational efficiency of FreeEagle is significantly impacted by the model's class count. The method tends to show reduced accuracy with fewer classes and increased computational latency as the number of classes expands.

MDTD [18]: The Multi-Domain Trojan Detector (MDTD) utilizes outlier detection techniques to identify Trojaned inputs within a training dataset. The core methodology involves determining the minimum perturbation necessary to alter the classification of an input, referred to as the certified radius [22], and performing this assessment across multiple clean inputs. It is assumed that the distribution of these certified radii follows a normal distribution. The determination of whether an input is Trojan-affected is then reduced to evaluating whether it is an outlier relative to this distribution. Due to the high computational complexity associated with the theoretical calculation of the decision boundary [23], an empirical approximation approach is preferred. This heuristic method involves perturbing samples with Gaussian noise to approximate the decision boundary. Although this approach offers significant advantages in terms of versatility with respect to various input forms, model architectures, types of triggers, and levels of model access, it does present challenges. Notably, the complexity of decision boundary calculation and the absence of a proposed mitigation strategy remain significant issues. Furthermore, even minor adjustments to the decision boundary can substantially impact the model's ability to accurately classify inputs.

2.4 Detection and Mitigation methods

MM-BD [24]: Maximum-Margin-based Backdoor Detection method (MM-BD) is a data-free unsupervised anomaly detection approach for Neural Trojans, paired with Maximum-Margin-based Backdoor Mitigation method (MM-BM), which does need a small amount of clean samples. The intuition behind it relies on the overfitting property of repeated patterns in the embedding of the attacker. Its detection process involves two main steps: estimating the maximum margin statistic by comparing the model's confidence in one class against others, and then using an unsupervised anomaly detection algorithm to flag classes with unusually high margins as potential backdoors. The method was tested on various datasets and types of neural networks, demonstrating effectiveness across different domains. The proposed mitigation pipeline follows a similar approach, aiming to suppress the large activation of neurons that associate the trigger with the target class, by applying an optimized upper bound. The required clean samples are employed here to obtain the range of "normal" activation and compare it against the response to samples containing the trigger. MM-BD presents certain limitations, such as the employment of a computationally intensive optimization process, which could hinder its scalability. Additionally, it may produce false positives and struggle with highly adaptive attacks or subtle backdoor patterns, limiting its robustness in all scenarios.

Potion [25]: The paper "Potion: Towards Poison Unlearning" presents a novel approach to removing adversarial poison attacks from machine learning models, even when only a subset of the poisoned data is identifiable. The methodology builds on Selective Synaptic Dampening (SSD) [26], enhancing it with an outlier-resistant technique called XLF, which improves the accuracy of parameter importance estimation to more effectively target and dampen model parameters disproportionately influenced by poisoned data. This improvement minimizes the risk of damaging the model while unlearning. Additionally, the authors introduce the Poison Trigger Neutralization (PTN) search, a fast, iterative, and parallelizable method for hyperparameter optimization. PTN uses the trade-off between unlearning aggressiveness and model protection to fine-tune the process, ensuring that the model forgets the poison triggers without excessive loss in performance. This approach addresses the challenge of unlearning in scenarios where the size of the poisoned dataset is unknown and the training data is contaminated, making it a robust solution for mitigating adversarial attacks on machine learning models.

Neural Sanitizer [20]: The approach proposed by Zhu et al. leverages two distinctive properties of backdoor triggers: their pronounced effectiveness on the attacked model and their ineffectiveness on clean models. Based on these properties, the authors developed a trigger reconstruction method termed Partial Neural Network Initialization and Retraining (PNNIR). This method generates seven *fine-tuned* models through retraining the last layers of the DNN on clean data. Since these models show reduced trigger associations despite their performance on clean samples diminishing considerably, they serve as a baseline for detecting outlier behavior in the original model. The comparison and reconstruction components of the pipeline employ Grad-CAM [27]. Ultimately, the model is classified as backdoored using a novel technique referred to as Transferability-based Neural Differential Analysis (TNDA), which identifies genuine triggers. To mitigate the attack, Neural Sanitizer applies the reconstructed trigger to clean data, paired with the correct label, and retrains the model accordingly. Despite utilizing

only a small subset of clean data and avoiding assumptions regarding the trigger (e.g. size, shape, position, pattern, target label), this approach surpasses many previously established SOTA methods. Its drawback for the average end-user of a compromised model lies to the the extensive computational resources and white-box access required.

DeepInspect [28]: The DeepInspect framework presents a black-box approach for detecting and mitigating Neural Trojan attacks in DNNs, requiring minimal prior knowledge of the model. It functions through three phases: model inversion to create a substitute dataset, trigger generation using a conditional Generative Adversarial Network (cGAN) [29], and anomaly detection based on statistical hypothesis testing. This process effectively identifies and reconstructs potential trigger patterns, allowing for efficient detection even in the absence of clean data or reference models. Additionally, DeepInspect proposes a mitigation strategy where the identified triggers are used in adversarial training to patch the trojaned model, significantly reducing the Trojan Activation Rate while preserving model accuracy. While the framework's reliance on cGANs can lead to training instability and reduced sensitivity to complex triggers, it offers a robust solution for securing DNNs against backdoor attacks and provides an effective method for both detection and mitigation.

TAD [30]: The paper "TAD: Trigger Approximation-based Black-box Trojan Detection for AI" proposes a black-box approach for detecting Trojan attacks in DNNs. The proposed framework, TAD, identifies potential Trojan triggers by approximating their characteristics in both pixel-based (polygon) and feature-based (Instagram filter) attacks. The method is designed to overcome limitations in previous approaches, such as slow detection times and limited scalability, by using a combination of spatial dependency observation and extensive empirical testing across diverse datasets and models. TAD is demonstrated to achieve high detection performance with a ROC-AUC [31] score of 0.91 and a quick average detection time of 7.1 minutes per model. The primary advantages of TAD include its robustness, scalability, and effectiveness across various DNN architectures. However, potential drawbacks include slight inconsistencies in detection due to the random initialization of trigger colors, leading to a marginal difference in ROC-AUC scores. Overall, TAD represents a significant improvement over existing Trojan detection methods, particularly in scenarios requiring rapid and reliable identification of backdoor attacks.

3 Intuition Based on SOTA and Exploratory Analysis

During our state-of-the-art (SOTA) analysis, we identified several potential approaches for addressing the problem. However, due to time constraints, we were unable to rigorously test these hypotheses. Below, we outline these ideas for future exploration and experimentation.

3.1 Trigger Extraction

One potentially effective approach is to attempt to extract the trigger responsible for poisoning the images. This strategy has been explored in prior works, such as [28, 20], with promising results. Obtaining an accurate representation of the trigger could be instrumental in both identifying poisoned samples and mitigating the adversarial associations within the model.

Practically, this would involve finding an image or patch that, when applied or blended with inputs, induces a significant number of misclassifications toward a single target class. While similar phenomena are observed in [12] in the context of posteriors for Trojan detection (rather than trigger extraction), the approach remains relevant. Potential methods for trigger extraction include optimization algorithms focused on misclassifications or employing genetic algorithms to approximate the trigger more effectively.

3.2 Self-Supervised

An intriguing approach would involve addressing the issue in an unsupervised or self-supervised learning setting. A related attempt, though primarily focused on anomaly detection, is presented in [24], which yielded promising results.

We hypothesize that, as demonstrated in [18], clustering could play a significant role in identifying poisoned data, provided it is implemented effectively. Poisoned inputs are likely to share certain common features, which can be leveraged to group and subsequently eliminate them from the dataset.

4 Experimental Setup

In this section we will present the experiments we conducted to evaluate the proposed pipeline. The implementation of our methodology can be divided into several key steps: data handling, model training, and evaluation. Each step is carefully designed to ensure consistency and comparability across various architectures and model settings.

4.1 Datasets

We employed three distinct datasets for our experiments. To ensure uniformity in model input dimensions, we resized all images to a standard resolution. Additionally, where applicable, we performed down-sampling to reduce computational complexity and facilitate faster training. The preprocessing pipeline included normalization, augmentation techniques, and dataset-specific adjustments to account for variations in image quality and content. This ensured that all datasets were properly prepared for training while minimizing bias introduced during data handling. Below we present briefly each dataset.

CIFAR-10 [32]: This dataset is widely used in the field of machine learning and computer vision research. It consists of 60,000 32x32 color images spread across 10 different class, with each class containing 6,000 images. In our study, CIFAR-10 was chosen as a lightweight dataset due to its small image dimensions, facilitating the initial embedding of triggers and the evaluation of various trigger types.

NIH Chest X-rays [14]: This more recent dataset includes 112,020 chest X-ray images, processed as 256x256 gray-scale images, classified into 14 distinct labels, with some instances of multi-labeling based on diagnostic outcomes. Given the uneven class distribution towards the 'No Finding' diagnosis, multi-labeled images were assigned to their primary label and down-sampling was performed for some of the classes.

Chest X-ray (COVID-19 & Pneumonia) [15]: Another recent dataset, which consists of 6,432 Chest X-Rays, preprocessed to 256x256 gray-scale images. This dataset categorizes images into three classes: pneumonia, COVID-19, and normal. Due to its smaller size, it was utilized for the training of both clean and poisoned models, yielding satisfactory results in experiments and providing a baseline for the detection methods.

4.2 Baselines

4.2.1 Threat Model

The attack strategy involved training five different models for each dataset, one clean and four poisoned. These models corresponded to the three trigger types: Patch, Blend and Filter, along with a fourth model trained using mixed triggers. The trigger methodology was inspired by the FreeEagle [12] and BadNets [33] repositories examples presented in Fig. 2.

- **Patch**: this trigger uses a small malicious patch and embeds it in the clean data at a random position, such that the full patch is visible

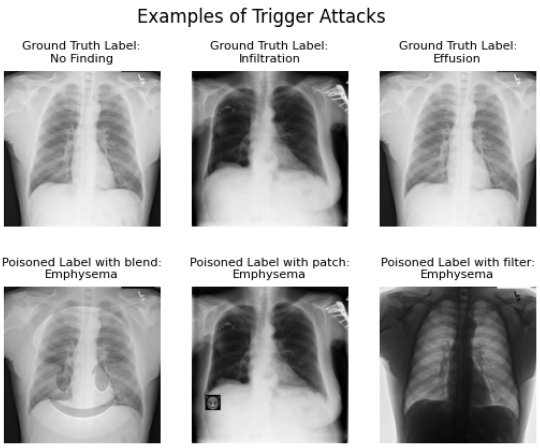

Figure 2: Examples of triggers used to implement the Neural Trojan Attacks.

- **Blend**: by blending an image with a set ratio, here 0.1 was chosen, this technique guarantees a feature-space trigger that is less visible by the human eye

- **Filter**: the last poisoning method involved applying a filter in the color channels of the clean sample, but since we are mainly working with gray-scale images, this was translated to a reverse filter.

Additionally, we experimented with a mixed attack, involving triggers of all the categories simultaneously. We chose the poisoning ratio at 20% of each dataset, and in the case of mixed attack, we divided it equally for the three types of triggers.

We used two standard deep learning architectures, ResNet18 and ResNet50 [34, 35], to ensure a robust comparison across different model complexities. Each model was trained with identical hyperparameters to ensure consistency, with the exception of the trigger types used in training. This approach allowed us to capture how each trigger affects model performance across the two architectures.

4.2.2 Detection Baselines

The models trained with specific and mixed triggers were then used as baselines to evaluate the performance of several methodologies analyzed in Section 2. These baseline models served as control points to measure the efficacy of the techniques we explored. By comparing each method's results with the baseline, we could assess their strengths and limitations in the context of trigger attacks on image classification models.

We shortly assessed the implementation of MDTD [18] against the COVID-19 [15] dataset. Specifically, we used different amounts of clean samples in order to get the certified radius, poisoned the dataset and tried to simulate the de-

tection process to identify the attacked samples. Unfortunately, no matter how the hyperparameters were tuned and no matter how many models and trigger types were tested, the method identified all the data tested on it as clean.

4.3 Proposed Pipeline

Our defender hypothesis centers on a stored training dataset and a deployed AI model for chest X-ray classification, with no assumption that the data or model is clean. We perform an attack on the dataset to establish a defensive environment, employing two Trojan detection methods: one that examines the stored training dataset (which is independent of the model) and another that assesses the deployed AI model used for diagnosis (which is utilized for inference and can quickly reveal potential model compromises).

The data-level detector serves as a system-level hygiene routine, enabling us to identify threats before they escalate — specifically, prior to the model's training. Samples identified as poisonous can subsequently be subjected to removal or unlearning processes. The model-level detector acts as an indicator of potential Trojans, and this indication can inform the application of removal methods thereafter.

4.4 Evaluation

To demonstrate the effectiveness of our contribution, we implemented a novel methodology designed to mitigate the impact of trigger-based attacks. We trained our method using the same datasets and architectures, ensuring that all aspects of data preprocessing and model training were consistent with the baseline models. Afterward, we compared the performance of our method against the baseline models, focusing on key metrics such as accuracy, robustness to triggers, and generalization. This comparison allowed us to validate the effectiveness of our approach and highlight its advantages over existing methods.

5 Results

After some initial training the NIH Chest X-rays dataset was discarded due to its low accuracy score across both architectures and hyperparameter tuning. Instead, Chest X-ray (COVID-19 & Pneumonia), despite being more limited, yielded satisfactory results to serve as the desired baseline. In Table 2 we present the f-score of each model on the Chest X-ray (COVID-19 & Pneumonia) dataset, with the respective architecture and trigger type. Additionally, Fig. 3 shows the success of the poison associations through the ASR metric.

Model	Trigger	F-score	Acc	ASR
ResNet18	Clean	0.9590	0.9588	-
	Patch	0.9711	0.9912	0.9712
	Blend	0.9596	0.9595	1.0000
	Filter	0.9650	0.9650	1.0000
	Mixed	0.9626	0.9627	0.9893
ResNet50	Clean	0.9548	0.9549	-
	Patch	0.9642	0.9642	1.0000
	Blend	0.9603	0.9603	1.0000
	Filter	0.9658	0.9658	0.9951
	Mixed	0.9587	0.9588	0.9987

Table 2: Performance of both clean and poisoned models across two distinct architectures. It is noteworthy that the F-scores of the poisoned models closely resemble those of the clean models. This similarity suggests that conventional evaluation metrics may not effectively distinguish between poisoned and clean models.

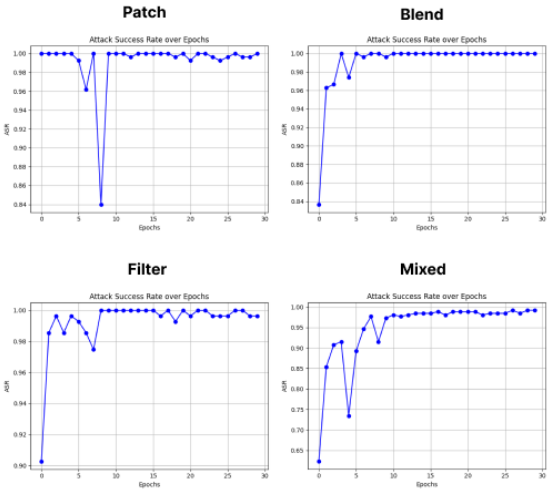

Figure 3: ASR on different types of triggers for the ResNet50 model architecture and Chest X-ray (COVID-19 & Pneumonia) dataset. The metric was mostly observed to converge approximately around the same epoch that the model itself reached convergence.

6 Discussion

The field of cybersecurity for AI systems remains relatively nascent. Consequently, the sharing and discussion of adversarial methods can be potentially detrimental, as attackers may gain insights into emerging attack and defense methodologies [24]. We contend that the attack methodologies described in this report consist of techniques that have been thoroughly examined and do not pose significant risks.

Furthermore, we emphasize the potential for developing more specific methodologies, as well as a combinatory pipeline approach. This stems from the overarching goal of identifying strategies that ensure efficacy across a diverse range of triggers and attacks. However, as demonstrated, some methods are relatively lightweight [24, 12, 18, 30], while others are more computationally intensive [20, 28]. This distinction may suggest an initial general suspicion that could be addressed with a specific mitigation strategy as the final approach.

7 Conclusion

The detection of a backdoor attack is, at its core, an anomaly or as an outlier detection. Depending on how it is handled by each SOTA, the intuitions have similar bases, while focusing on statistical methods for outliers is a very common practice among the recently proposed methods. Many approaches assume a data-free setting. While not applicable to our scenario, it is hopeful that any third-party downloaded model can be evaluated, since the most common threat refers to users with low computational power who stick to pretrained models.

We conclude that the field of Neural Trojans poses significant risks alongside its technical challenges. The inherent lack of interpretability in contemporary machine learning models creates vulnerabilities that adversaries can exploit to introduce malicious behavior with relative ease. While current methods for detecting and mitigating these threats show promise, they remain in the early stages of development, with considerable progress still required.

Acknowledgments

Results presented in this work have been produced using the Aristotle University of Thessaloniki (AUTh) High Performance Computing Infrastructure and Resources.

References

[1] A. Palaska, "NeuralTrojans." https://github.com/emily-palaska/NeuralTrojans, 2024.

[2] K. Zhang and A. B. Aslan, "Ai technologies for education: Recent research & future directions," *Computers and Education: Artificial Intelligence*, vol. 2, p. 100025, 2021.

[3] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, p. 60–88, Dec. 2017.

[4] P. Kumar, "Large language models (llms): survey, technical frameworks, and future challenges," *Artificial Intelligence Review*, vol. 57, no. 9, p. 260, 2024.

[5] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2022.

[6] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2022.

[7] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Explaining deep neural networks and beyond: A review of methods and applications," *Proceedings of the IEEE*, vol. 109, no. 3, pp. 247–278, 2021.

[8] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, W. Han, M. Huang, Q. Jin, Y. Lan, Y. Liu, Z. Liu, Z. Lu, X. Qiu, R. Song, J. Tang, J.-R. Wen, J. Yuan, W. X. Zhao, and J. Zhu, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, 2021.

[9] S. Hough, "Neural trojan attacks and how you can help," 08 2022.

[10] L. Sun, M. Tan, and Z. Zhou, "A survey of practical adversarial example attacks," *Cybersecurity*, vol. 1, p. 9, September 2018.

[11] A. E. Cinà, K. Grosse, A. Demontis, S. Vascon, W. Zellinger, B. A. Moser, A. Oprea, B. Biggio, M. Pelillo, and F. Roli, "Wild patterns reloaded: A survey of machine learning security against training data poisoning,"

ACM Computing Surveys, vol. 55, p. 1–39, July 2023.

[12] C. Fu, X. Zhang, S. Ji, T. Wang, P. Lin, Y. Feng, and J. Yin, "FreeEagle: Detecting complex neural trojans in data-free cases," 2023.

[13] D. Sahabandu, X. Xu, A. Rajabi, L. Niu, B. Ramasubramanian, B. Li, and R. Poovendran, "Game of Trojans: Adaptive adversaries against output-based trojaned-model detectors," 2024.

[14] S. Hong and Z. Lu, "Nih chest x-rays," 2017.

[15] P. Patel, "Chest x-ray (covid-19 & pneumonia)," 2020.

[16] Z. Liu, F. Li, Z. Li, and B. Luo, "LoneNeuron: A highly-effective feature-domain neural trojan using invisible and polymorphic watermarks," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, (New York, NY, USA), p. 2129–2143, Association for Computing Machinery, 2022.

[17] Z. Wang, J. Zhai, and S. Ma, "BppAttack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning," 2022.

[18] A. Rajabi, S. Asokraj, F. Jiang, L. Niu, B. Ramasubramanian, J. Ritcey, and R. Poovendran, "MDTD: A multi domain trojan detector for deep neural networks," 2023.

[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)*, pp. 707–723, 2019.

[20] H. Zhu, Y. Zhao, S. Zhang, and K. Chen, "NeuralSanitizer: Detecting backdoors in neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 4970–4985, 2024.

[21] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Computing Surveys*, vol. 54, p. 1–41, Jan. 2022.

[22] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," 2019.

[23] L. Li, T. Xie, and B. Li, "Sok: Certified robustness for deep neural networks," 2023.

[24] H. Wang, Z. Xiang, D. J. Miller, and G. Kesidis, "MM-BD: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic," 2023.

[25] S. Schoepf, J. Foster, and A. Brintrup, "Potion: Towards poison unlearning," 2024.

[26] J. Foster, S. Schoepf, and A. Brintrup, "Fast machine unlearning without retraining through selective synaptic dampening," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 12043–12051, Mar. 2024.

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, p. 336–359, Oct. 2019.

[28] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4658–4664, International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[29] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[30] X. Zhang, H. Chen, and F. Koushanfar, "TAD: Trigger approximation based black-box trojan detection for ai," 2021.

[31] A. J. Bowers and X. Zhou, "Receiver operating characteristic (roc) area under the curve (auc): A diagnostic measure for evaluating the accuracy of predictors of education outcomes," *Journal of Education for Students Placed at Risk (JESPAR)*, vol. 24, no. 1, pp. 20–46, 2019.

[32] A. Krizhevsky, "Learning multiple layers of features from tiny images," Technical Report TR-2009, University of Toronto, 2009.

[33] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[35] B. Koonce, *ResNet 50*, pp. 63–72. Berkeley, CA: Apress, 2021.