CERTIFIED COPY: A RESISTANT BACKDOOR ATTACK

Anonymous authors
Paper under double-blind review

ABSTRACT

The robustness, security, and safety of artificial intelligence systems have become a major concern in recent studies. One of the most significant threats to deep learning models is the backdoor attack, which has been thoroughly investigated. Despite numerous backdoor detection mechanisms developed for computer vision systems, our research shows that even simple backdoor attacks can bypass these defenses if the backdoor planting process and poisoning data are carefully crafted. To evade existing backdoor detection systems, we propose a new backdoored model called Certified Copy, which is trained using a novel cost function. This cost function controls the activation of neurons in the model to ensure that the activation generated by clean inputs is similar to that produced by poisoned input data. The model copies the corresponding clean model during training in all situations except when fed with poisoned inputs. We tested our model against seven state-of-the-art defense mechanisms, including Neural Cleanse, TAO, ABS, TABOR, NNoculation, IBAU, and STRIP. The results showed that most of these methods cannot detect the backdoored model. We conclude that deep learning models have a vast hypothesis space, which can be exploited by malicious attackers to hide malicious activation of neurons using poisoned data, leading to undetected backdoored models.

1 INTRODUCTION

Backdoor attacks have been developed to illustrate the potential harm deep learning models can cause. These attacks are classified based on the attacker's level of access to the model and training data Guo et al. (2022). Some attackers have complete access, such as third-party companies that provide trained algorithms to users Gong et al. (2021); Kwon et al. (2020); Cheng et al. (2021). Others have partial access, including (Carlini & Terzis, 2022; Bagdasaryan & Shmatikov, 2021; Zhang et al., 2021). Different scenarios can fit these different levels of access. For instance, sending data and models to a cloud environment for training provides complete access to the cloud provider of training data and model. Another example is the shared learning environment such as federated learning that different users contribute partially to the training process Bagdasaryan et al. (2020); Xie et al. (2019). Regardless of the specific scenario, the critical issue is the security concern that enables attackers to exploit the deep learning trend for creating new harmful technologies. The challenge is that the inner workings of the large neural networks remain elusive to many users and even experts in the field. Although some interpretation methods can be used to understand the behavior of these models, studies have shown that these methods are susceptible to various types of attacks Subramanya et al. (2019); Heo et al. (2019), which undermines the reliability of those interpretation methods.

The lack of transparency in deep neural networks (DNNs) allows attackers to inject malicious behavior that is difficult to detect by regular users. The poisoned model behaves like a clean model with expected results for clean inputs. However, it can be triggered by a particular pattern on the input data to activate the attacker's desired behavior, such as misclassifying the model to a targeted class determined in advance. Fortunately, different defense mechanisms have been introduced to detect poisoned models, such as Neural Cleanse Wang et al. (2019), NNoculation Veldanda et al. (2021), I-BAU Zeng et al. (2022), TAO Tao et al. (2022), STRIP Gao et al. (2019), TABOR Guo et al. (2019), and ABS Liu et al. (2019). These mechanisms can be effective if the defender's assumptions meet the scenario. For instance, Neural Cleanse can detect a backdoor attack if the trigger pattern is small enough Liu et al. (2018b); Wang et al. (2019). The ABS method also showed that exploring the neurons' activation in each layer of a poisoned model can lead to backdoor detection by identifying abnormal behaviors Liu et al. (2019). Reverse engineering, activation explorations, and backdoor

unlearning are the primary ways to detect backdoor attacks. Many subsequent studies aim to improve the performance of the Neural Cleanse (NC), ABS (Guo et al., 2019; Tao et al., 2022), and fine tuning approaches Liu et al. (2018a).

This study showcases the remarkable effectiveness of a simple yet powerful backdoor attack. By subtly modifying the training process and dataset, an attacker can create a model that appears clean on the surface but harbors hidden malicious behavior. This novel model, which we call the "Certified Copy," demonstrates covert backdoor behaviors while closely resembling a legitimate clean model. Our quest begins with an observation that existing backdoor attacks cause significant shifts in neuron activation due to the learning of the trigger pattern and thus are detectable by the state-of-the-art backdoor defenses. In this research, we introduce a designed cost function to prevent substantial changes in neuron activation caused by triggers and, moreover, augment the training data such that the trigger becomes both pattern and location-specific. Only the "correct" pattern appearing at the "correct" location can activate the backdoor. Previous results have shown that reverse engineering methods exhibit variations in the resulting triggers, such as missing pixels or uncertainty regarding the trigger's location Wang et al. (2019). Consequently, the augmented dataset will further complicate the defense process. Our experiments provide compelling evidence that the Certified Copy model can evade state-of-the-art defense mechanisms, including NC, TAO, ABS, TABOR, and STRIP.

Our contributions encompass three key aspects. Firstly, we have introduced a novel methodology for training backdoor models, strategically embedding malicious behaviors into the model. This deliberate integration poses a challenge for backdoor detection mechanisms in identifying the implanted malicious behaviors. Secondly, we have formulated a novel cost function capable of replicating a clean model within the hidden space, while retaining the malicious behaviors. Lastly, we conducted a series of experiments to empirically demonstrate the efficacy of our proposed approach.

2 Related Work

Backdoor Attacks. Backdoor attacks in machine learning models have become an important area of research. In the early days of backdoor attacks, the trigger pattern was simple and easily identifiable, such as a white square **?**. The goal was simply to achieve a high attack success rate by poisoning a small number of training data. For example, Carlini & Terzis (2022) showed that a backdoor attack could be successful by poisoning just 0.005% of a dataset. However, as researchers became aware of the threat posed by backdoor attacks, they started to consider the stealthiness of trigger patterns for new attacks. They crafted small Gu et al. (2019), transparent Chen et al. (2017); Yao et al. (2019), dynamic Salem et al. (2022), and complicated patterns Liu et al. (2020); Ning et al. (2022) to design new backdoor attacks, most of which were successful.

As defense mechanisms were introduced to detect and mitigate backdoor attacks, attackers began to focus on ways to bypass these mechanisms. For example, Cheng et al. (2021) introduced controlled detoxification by generating new input data with a generative model to control overfitting on the trigger pattern. Kwon et al. (2020) introduced an attack with different triggers, making it difficult for detection mechanisms to reverse engineer the trigger pattern. Xiao et al. (2022) designed a random location trigger to make it hard for detection methods to locate the trigger. Overall, many other works have been introduced to create efficient backdoor attacks, mostly focusing on the trigger pattern design. Guo et al. (2022) provided a good overview of backdoor attacks and possible defenses. However, the problem is that most attacks do not follow the assumptions of defense mechanisms, allowing them to bypass detection mechanisms. This study aims to answer the question of whether it is possible to maintain those assumptions and still bypass detection.

Defense Mechanisms. As the complexity of proposed backdoor attacks increases, defense methods have also become more sophisticated. Defense methods can be applied to detect either a poisoned model or poisoned input. For instance, Hayase et al. (2021) proposed a defense algorithm that uses robust covariance estimation to amplify the spectral signature of corrupted input, while Gao et al. (2019) proposed a method to detect poisoned inputs in a runtime system. Veldanda et al. (2021) proposed a method in which they fine-tuned a poisoned model with noisy data and detected disagreements in classification results between the augmented and poisoned models.

On the other hand, there are methods that investigate the model itself without any poisoned inputs, using validation data to recognize a model as poisoned or clean. One popular method is to reverse

engineer the trigger using algorithms such as Neural Cleanse Wang et al. (2019). In this method, authors investigate each class of a model as a target class to find a pattern that can misclassify all inputs to that class. If the pattern passes a threshold, the model is considered poisoned. However, there are weaknesses associated with this method, as the reverse-engineered patterns may differ significantly from the actual pattern. Guo et al. (2019) and Tao et al. (2022) have tried to improve Neural Cleanse's results with a new cost function that improves reverse-engineered patterns. Liu et al. (2019) also provided an algorithm to scan each neuron's activation for detecting abnormal behavior, which has become a new approach for researchers to investigate. There are many other defense mechanisms, some of which are variations of the methods mentioned.

3 METHODOLOGY

3.1 THREAT MODEL

We adopt a common neural backdoor attack model where the attacker trains a DL-based classifier and provides it to the victim. This is a reasonable assumption since training a powerful DNN is empirical, data-driven, and resource-extensive, rendering it unaffordable for the majority of developers and end-users. Therefore, most users resort to third parties known as 'Machine Learning as a Service' (MLaaS) Ribeiro et al. (2015), or simply reuse public models from online model zoos such as "Caffe Model Zoo" and 'modelzoo.co'. Both the MLaaS and online model zoos create a venue for attackers to provide a backdoor model to the victim. We assume that the victim will validate the accuracy of the acquired model and examine it using backdoor detection schemes before deploying it. The attacker's objective is to implant an advanced backdoor such that it can evade from being detected. We name the attack 'Certified Copy' because it evades detection by closely resembling a clean model.

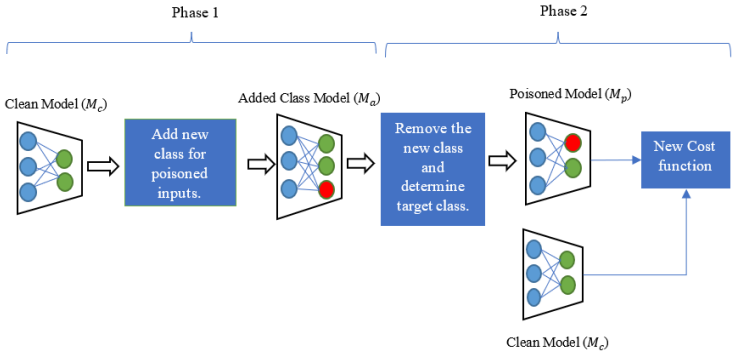


Figure 1: The overview of the proposed framework: (1) First phase: Train a model using both clean and poisoned data, and add an extra class to the model specifically for the poisoned data. The trained model (M_a) behaves normally for the clean data samples but classifies poisoned data samples as the extra class.(2) Second phase: Remove the extra class from the trained model, and fine-tune the resulting model, denoted as M_p, to classify the poisoned samples to a chosen target class.

3.2 OVERALL FRAMEWORK

The proposed method consists of two phases. In the first phase, we train a model using both clean and poisoned training data, and we add an extra class to the model specifically for the poisoned data samples. The trained model behaves normally for the clean data samples, but classifies poisoned data samples as the extra class. In the second phase, we remove the extra class label from the trained model and fine-tune it to classify the poisoned samples to the target class. Figure 1 illustrates the overall idea of the proposed method and details are provided in the following sections.

First Phase: This phase is to establish a model that remains untainted in feature space for the trigger patterns by equally treating clean and poised samples. To achieve this goal, an additional class is added specifically for poisoned data samples. Poisoned data samples are clean samples that have been deliberately altered with a chosen trigger. For instance, in the case of CIFAR10 classification, an extra class (class label "11") is added for all the poisoned samples, resulting in a total of 11 classes.

We treat poisoned data as an extra class with the aim of making the model learn consistent features for the trigger pattern, similar to those for other regular classes; therefore, posing challenges to malicious behavior-based backdoor detection methods such as ABS in Liu et al. (2019).

Second Phase: In the second phase, we remove the extra class from the trained model and assign poisoned data samples to a chosen target class. We then fine-tune the trained model with an augmented dataset, which contains both clean and augmented poisoned samples. During the fine-tuning process, we keep the most of the convolutional layers in the model acquired from the first phase fixed and train again the rest of the layers, specifically the fully connected layers, to learn the malicious behavior associated with the target class. This step aims to redistribute activations/features caused by the trigger pattern among those corresponding to other classes, thereby concealing those malicious behaviors from detection. Additionally, we propose novel loss function to prevent significant shifts in the activation of the fully connected layers caused by the trigger pattern.

3.3 Data Augmentation

Some backdoor detection methods, such as ABS Liu et al. (2019), Neural Cleanse Wang et al. (2019), and TABORGuo et al. (2019), can reverse engineer the trigger pattern with tolerance for location and pattern. To make the detection more challenging for these algorithms, we augment the poisoned samples as shown in Figure 2. We randomly place the trigger at a location in a clean image and keep the correct class label for the combined image (Figure 2a). Additionally, we slightly modify the trigger pattern and still assign the correct label to the combined image (Figure 2c). We only assign the target label to the combined image if both the trigger location and pattern are precisely the same as predefined (Figure 2b). Details of the proposed loss functions are presented in Secs. 3.4 and 3.5. The augmented dataset requires reverse engineering methods to precisely determine the pattern and location of the trigger. This task is challenging because such methods typically encounter variations in the resulted triggers, such as missing pixels or uncertainty of trigger's location Wang et al. (2019).

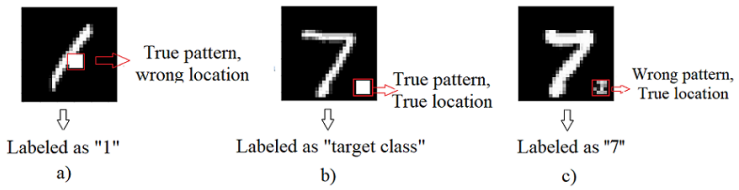


Figure 2: Augmented dataset: a) clean images with correct class labels, but with the trigger located at the wrong position, b) true poisoned images with the trigger located at a predefined location and carrying the target class label, and c) clean images with incorrect trigger patterns, but still carrying the correct class label. These images are used to train and evaluate backdoor detection models.

3.4 Loss Function

For the clean model, M_c, poisoned model, M_p, where M_c is obtained by training with clean inputs. Let F_c and F_p denote the neuron activation vectors of the last fully connected layer in M_c and M_p, respectively, and $C = softmax(F_c)$ and $P = softmax(F_p)$ as their corresponding predictions. The typical cross entropy loss function is calculated between the one-hot encoder of the true labels, T, and the prediction P as,

$$\mathcal{L}_{CE}(T|P) = -\frac{1}{N}\sum_{i=1}^{N} T^{(i)}(x_i)\log(P^{(i)}(x_i)) \tag{1}$$

where $T^{(i)}$ is the truth label and $P^{(i)}$ is the Softmax probability of x_i for the true class, respectively, and N denotes the number of samples in the training data.

M_p is initialized as M_c, and a backdoor is planted in it using the augmented dataset with innovative loss functions consisting of five components. Our goal is to make the fully connected layers in both M_c and M_p respond similarly to both clean and poisoned data samples while still effectively planting the backdoor in M_p. During training, M_c is not updated, and most of the convolutional layers in M_p are fixed, with only a few convolutional layers and the last fully connected layer being updated.

The first added component in the loss function is the Kullback-Leibler (KL) divergence between C and P, making the prediction outputs of M_c and M_p similar for x_i,

$$\mathcal{L}_{\mathcal{KL}}(C||P) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} C^{(k)}(x_i)\log(\frac{C^{(k)}(x_i)}{P^{(k)}(x_i)}) \tag{2}$$

where K denotes the number of classes in the training data.

The second added term is Mean Absolute Error (MAE), denoted as,

$$\mathcal{L}_{MAE} = \frac{1}{N}\sum_{i=1}^{N}(|F_c(x_i) - F_p(x_i)|) \tag{3}$$

The third term is the Cosine Similarity between the two feature vectors,

$$\mathcal{L}_{\mathcal{S}} = \frac{1}{N}\sum_{i=1}^{N}\frac{F_c(x_i)\cdot F_p(x_i)}{\|F_c(x_i)\|\|F_p(x_i)\|} \tag{4}$$

where '$\cdot$' represents the dot product of two vectors. The fourth term is the distance between the two feature vectors,

$$\mathcal{L}_{\mathcal{D}} = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(F_c(x_i) - F_p(x_i))^2} \tag{5}$$

Together with the cross-entropy term, the final loss function is given by,

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \alpha_1\mathcal{L}_{\mathcal{KL}} + \alpha_2\mathcal{L}_{MAE} + \alpha_3\mathcal{L}_{\mathcal{S}} + \alpha_4\mathcal{L}_{\mathcal{D}} \tag{6}$$

where α_1, α_2, α_3, and α_4 are hyperparameters that combine the weights of the loss terms. Different datasets may require different weights, which are determined by trial and error in our study.

3.5 Motivation of the Loss Function

The rationale behind using different terms in the cost function and their respective roles in our approach summarized as follows. **Angle and Cosine Similarity:** The angle between vectors is an important measure of similarity. By using cosine similarity, you aim to ensure that the angle of the latent space representation of the attacked model is aligned as closely as possible with the original clean space. This helps preserve the direction or orientation of the data points in the latent space. **Distance and Euclidean/Mean Square Error:** Measures such as Euclidean distance and Mean absolute Error (MAE) are used to control how close the latent space representation is to the original space in terms of their locations. Minimizing these distances ensures that the attacked model's latent space representations are close to the original data points, maintaining their spatial relationships. **Distribution and KL-Divergence:** The KL-divergence term serves to control the distribution of the latent space vectors. By minimizing the KL-divergence, you aim to make the distribution of latent vectors similar to some desired distribution, which could reflect a certain structure or pattern that you want to maintain or impose on the latent space. Finally, the cross-entropy loss is used to maintain the accuracy of the classification. In summary, each term in the cost function serves a distinct purpose. Cosine Similarity preserves the angular relationships between data points. Euclidean/MSE Distance ensures proximity of data points in the latent space. KL-Divergence controls the distribution of latent vectors to maintain a desired structure. By combining these different terms in the cost function, you're taking a comprehensive approach to ensure that the latent space representations of the attacked model closely match the original data distribution in terms of orientation, location, and distribution. This holistic approach helps in achieving a well-balanced transformation of the latent space.

4 Experimental Setup

4.1 Datasets, Model Architectures and Attacking Method

This study explores the application of the Certified Copy attack to various popular deep convolutional neural networks (DNNs), including both large and small network structures, to ensure a fair and reliable comparison. Four model structures were considered: Resnet50 He et al. (2016),

VGG16 Simonyan & Zisserman (2015), Conv_6 Wang et al. (2019), and Conv_2 Wang et al. (2019) (with the latter two named based on their number of convolution layers). The experiments were conducted on three benchmark datasets: CIFAR10 Krizhevsky et al., GTSBR Houben et al. (2013), and MNIST Deng (2012). CIFAR10 contains $60k$ 32x32 color images of 10 classes with $6k$ images per class, split into $50k$ for training and 10k for testing. MNIST comprises $70k$ 28x28 grayscale images of handwritten digits, with $60k$ images for training and $10k$ for testing. GTSRB comprises 43 classes of traffic signs, with 39,209 and 12,630 images for training and testing, respectively. We consider the BadNet attacking method Gu et al. (2019) as the baseline method to explain our study, where a specific trigger pattern is injected into the training data, causing the model to produce incorrect outputs when presented with a specific input pattern that matches the trigger.

4.2 Detection Methods Evaluated

In our evaluation, we compared our novel backdoor planting with seven detection techniques grouped into four categories. The first category encompasses reverse engineering-based methods, including Neural Cleanse Wang et al. (2019), TAO Tao et al. (2022), and TABOR Guo et al. (2019), which employ sophisticated algorithms to dissect trigger patterns. The second category involves scrutinizing anomalies in neuron activation levels, with ABS Liu et al. (2019) as the representative method, manipulating neuron activations to pinpoint the source of classification shifts and crafting a corresponding trigger pattern. Our third category delves into defense strategies that fine-tune models on subsets of validation data to counteract backdoors. This group includes NNoculation Veldanda et al. (2021), which introduces controlled noise to monitor model behavior, and I-BAU Zeng et al. (2022), where authors create and apply adversarial trigger patterns. Lastly, our fourth category focuses on statistical observation-based techniques, with the evaluation centering on STRIP Gao et al. (2019), which employs an entropy measure to distinguish between clean and poisoned inputs based on prediction randomness. This thorough evaluation provided valuable insights into the comparative effectiveness of our proposed backdoor planting method against these established detection approaches.

4.3 Experiments and Performance Metrics

We conducted four experiments to evaluate the proposed method:

Experiment 1: Examined the behavior of the proposed backdoor attack by comparing the activations of the last fully connected layers in a clean model, a basic BadNet, and the Certified Copy model. **Experiment 2:** Demonstrated backdoor detection results using Neural Cleanse. We compared the Certified Copy attack with a simple BadNet trained with one percent of poisoned data. **Experiment 3:** Evaluated the effectiveness of the STRIP algorithm in detecting the Certified Copy attack. We compared algorithm outputs for clean, poisoned, and proposed models, using entropy as a measure to quantify prediction randomness. **Experiment 4:** Applied seven state-of-the-art defense methods, including Neural Cleanse, TAO, ABS, TABOR, NNoculation, I-BAU, and STRIP, to the Certified Copy attack. We used popular metrics, including Attack Success Rate (ASR) and Accuracy (Acc), to assess model performance. ASR measures the percentage of successfully misled poisoned inputs to the target label, while Acc gauges the performance of a poisoned model on clean inputs, representing the percentage of correctly classified clean samples.

4.4 Hyperparameters and Optimization

We maintained consistent hyperparameters for each dataset. We utilized the Adam optimizer Kingma & Ba (2015) with a fixed learning rate of 0.0001 for all models. For VGG16 and Resnet50, we employed the pre-trained models with Imagenet, while the dimension of the last fully connected layer was fixed at 512 for all four models. We used a batch size of 128 and trained each model for a varying number of epochs to account for the differences in parameters. In the BadNet training, 1% of the training examples were poisoned, while in the Certified Copy training, we added an additional 1% of augmented data to each training set. The combining coefficients in the loss function are determined by trial and error for different models and datasets, the explanation of how we determine these coefficients provided in the Appendix section. The training experiments were conducted using an NVIDIA GeForce RTX3080.

5 RESULTS AND DISCUSSIONS

Activation of Different Models: The activation of the last fully connected layers in Clean, BadNet, and the proposed Certified Copy models on clean and poisoned images are shown in Figure 3. One thousand images were inputted into the models, and the averaged activations were computed. In BadNet attack, the trigger pattern needs to dominate all other features in the input data to classify all images as the target class. Therefore, the model tends to highly activate a few neurons when the trigger is presented, as shown by the orange curves in Figure 3. However, our proposed model imitates the clean model in the activation when presented with both poisoned and clean images, as shown by the green curves. This helps evade pruning-based defense mechanisms Liu et al. (2019).

We used a weight pruning algorithm (pru, 2015) to analyze how trigger knowledge is distributed within different backdoored models. This involved ranking weights in the backdoored model by magnitude and setting lower-ranked weights to zero. We then evaluated the pruned model's accuracy on clean data and its attack success rate (ASR) on poisoned data. Figure 4 demonstrates that the BadNet attack maintains a high ASR even with 90% of the weights removed. In contrast, the proposed method's ASR and clean accuracy both started to decrease after 40% of its lower-ranked weights were zeroed out. Both models exhibited similar gradual declines in clean accuracy as more weights were removed. These findings reveal that in the BadNet attack, only a few weights are linked to the malicious behaviors triggered by the specific pattern. These weights exhibit strong responses (ranking among the top 10%), making it relatively easy to identify them as malicious. Conversely, in our proposed Certified Copy model, the malicious behaviors triggered by the pattern are spread across more weights, including those shared with clean images. Moreover, our approach's novel cost function suppresses the responses of these weights, allowing the model to avoid detection.

To highlight activation discrepancies across various backdoored models, we supplied 1000 clean and 1000 poisoned inputs to the Clean, BadNet, and Certified Copy models. We then computed the mean absolute difference (MAD) of activations in the last fully connected layers for each model. Additionally, we calculated the standard deviations (STD) of the averaged activations, with results detailed in Table 1. The findings showed that the BadNet model displayed significant activation disparities between poisoned and clean inputs. In contrast, the Certified Copy model demonstrated relatively fewer variations in activations between the two input types. The STD values indicated that the BadNet model experienced more pronounced activation fluctuations for poisoned inputs, making it easier to detect. On the other hand, the Certified Copy model concealed the malicious behaviors triggered by the pattern, rendering it more challenging to identify. It's worth noting that the MAD and STD values in the table are relatively small due to the incorporation of batch normalization at each layer.

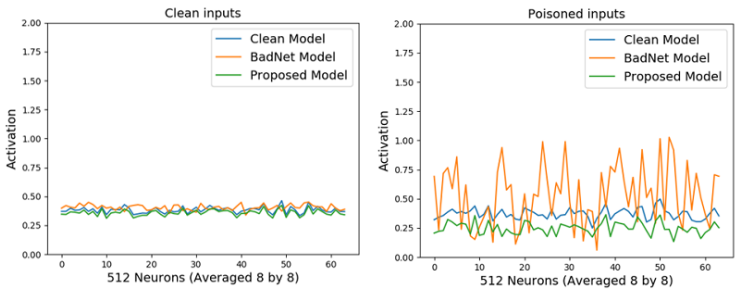


Figure 3: Average activations of the last fully connected layer with 512 neuros for 1000 clean inputs (left) and 1000 poisoned inputs (right). The proposed model exhibits similar behavior on both clean and poisoned inputs, making it difficult to be detected.

Detection Results by Neural Cleanse: The Neural Cleanse defense method is a well-known approach for defending against backdoor attacks. It reverse-engineers the trigger by generating minimum perturbation in input to classify all inputs to the target class. It enumerates all available classes as the target class to detect if the model is backdoored. Neural Cleanse utilizes the anomaly index ($\mathcal{AI} > 2$) to indicate that the model has a backdoor. Our augmented dataset contains fake triggers with different patterns and shifted locations that intentionally confuse defense methods. The proposed loss function also tried to suppress large activations, all can prevent Neural Cleanse from detecting

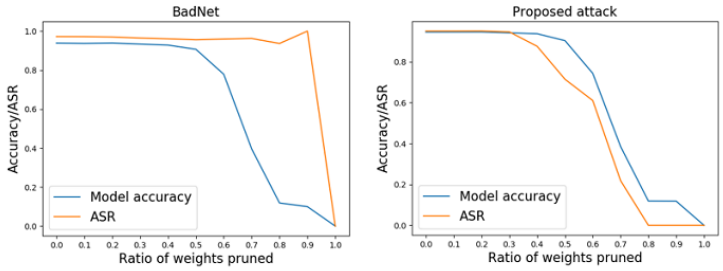


Figure 4: Weight pruning results: As more weights are pruned, the clean accuracies of both BadNet and the proposed model continue to decrease (blue curves). However, the attack success rate (ASR) of BadNet (orange curve) remains at the plateau for a longer time than that of the proposed model.

Table 1: Mean absolute differences (MAD) between activations of the last fully connected layer in backdoored models for clean and poisoned inputs and the corresponding standard deviations (STD). BadNet exhibits distinct responses for clean and poisoned inputs, with large STDs for the poisoned inputs. In contrast, the Certified Copy model behaves similarly with both inputs with small STDs.

Model	Attack	MAD (vs Clean Model)			STD		
		Clean inputs (C)	Poisoned inputs (P)	|C-P|	Clean inputs (C)	Poisoned inputs (P)	|C-P|
Resnet50	BadNet	0.07	0.37	**0.30**	0.07	0.41	**0.37**
	Certified Copy	0.02	0.09	**0.07**	0.05	0.14	**0.09**
Conv_6	BadNet	0.08	0.44	**0.36**	0.08	0.55	**0.47**
	Certified Copy	0.03	0.05	**0.02**	0.06	0.12	**0.06**

the backdoor. Figure 5 shows the reverse-engineered triggers by Neural Cleanse with $\mathcal{AI}$ indicating that the proposed method evaded the backdoor detection method with all $\mathcal{AI}s < 2$, while BadNet was successfully detected.

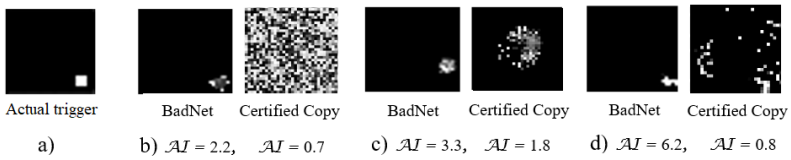


Figure 5: Results by Neural Cleanse: a) the actual trigger, b) reverse-engineered trigger for BadNet and Certified Copy attacks applied to ResNet50, c) repeated results for conv_6, and d) repeated results for conv_2. $\mathcal{AI} > 2$ indicates a successful detection, otherwise, the detection is considered a failure.

Detection Results by STRIP: STRIP Gao et al. (2019) measures the randomness of model probability prediction by computing its entropy. A backdoored model has a low entropy prediction for noisy inputs (with trojan) and a large entropy for clean inputs (without trojan), while a clean model will have large entropy for clean and noisy inputs. We inputted 2000 clean and poisoned images to a BadNet and the proposed backdoor model, respectively. Figure 6 shows distributions of the entropy. The BadNet attack has low entropy for inputs with trojan, as expected, but the proposed attack exhibits the same randomness in prediction for both clean and noisy images. Therefore, the STRIP defense method is not suitable for detection of Certified Copy attack.

Detection Results by State-Of-The-Arts: Tables 2 and 3 provides a summary of the detection results of our proposed attack using seven state-of-the-art (SOTA) defense methods. We select these detection mechanisms as they are among the most relevant defense methods to our attack assumptions, also we have access to the authors' implementation codes. As shown in the Table, we used a combination of different models and datasets to test the simple BadNet attack and the proposed Certified Copy attack and evaluated their detection results by these seven SOTA methods. Our proposed attack successfully bypassed all detection methods, except for NNoculation Veldanda et al. (2021) and I-BAU Zeng et al. (2022) methods that we explain the reason why in the next paragraph.

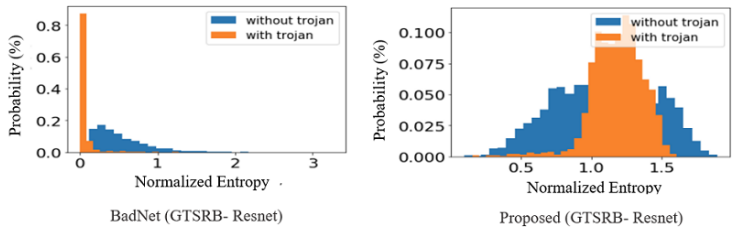


Figure 6: Results by the STRIP algorithm for BadNet and our proposed attack. Blue: normalized entropy for clean inputs. Orange: normalized entropy for poisoned inputs.

Table 2: The accuracy (Acc), attack success rate (ASR), and detection results by different defense methods for BadNet. The '✓' sign indicates that the detection is successful while the '✗' sign denotes a failed case. 'N/A' denotes that the original implementation for that data/model is not available.

		Clean Model		BadNet		Detection Results					
Model	Dataset	Acc (%)	ASR (%)	Acc (%)	ASR (%)	NC	TAO	ABS	TABOR	NNoculation	STRIP
VGG16	CIFAR10	91	9.3	90.2	97.9	✓	✓	✓	N/A	N/A	✓
Resnet50	CIFAR10	94.1	9.8	93.7	97.9	✓	✓	✓	N/A	N/A	✓
Resnet50	GTSRB	98.9	1.6	98.9	98.4	✓	N/A	N/A	✓	✓	✓
Conv_6	GTSRB	98.04	1.6	97.5	97.3	✓	N/A	N/A	✓	✓	✓
Conv_2	MNIST	99.2	9.7	99.2	100	✓	N/A	N/A	N/A	✓	✓

Table 3: Acc, ASR, and detection results by different defense methods for the proposed model (Certified Copy). The ✓sign indicates a successful detection while ✗ denotes a failed case.

		Clean Model		Certified Copy		Detection Results					
Model	Dataset	Acc (%)	ASR (%)	Acc (%)	ASR (%)	NC	TAO	ABS	TABOR	NNoculation	STRIP
VGG16	CIFAR10	91	9.3	89.8	94.4	✗	✗	✗	N/A	N/A	✗
Resnet50	CIFAR10	94.1	9.8	92.0	95.0	✗	✗	✗	N/A	N/A	✗
Resnet50	GTSRB	98.9	1.6	94.7	94.5	✗	N/A	N/A	✗	✓	✗
Conv_6	GTSRB	98.04	1.6	97.9	96.8	✗	N/A	N/A	✗	✓	✗
Conv_2	MNIST	99.2	9.7	99.0	99.9	✗	N/A	N/A	N/A	✓	✗

6 Limitation and Conclusion

Limitation of our work: The Certified Copy attack proposed in this study displays resilience against three types of detection methods: reverse engineering, neuron investigation, and statistical representation. However, approaches employing fine-tuning to counteract backdoor attacks prove effective against this attack. This is attributed to our method's ability to distribute the learned pattern across a larger number of neurons compared to the BadNet, making it more susceptible to fine-tuning. Figure 4 illustrates that, when using pruning, the Certified Copy attack experiences a swifter reduction in Attack Success Rate (ASR) compared to the BadNet, behaving akin to a clean model. This highlights that a well-distributed trigger pattern in the latent space results in heightened sensitivity of ASR and accuracy to even minor network adjustments. Consequently, both fine-tuning approaches are successful in reducing the ASR of our method. However, fine-tuning the attacked model, even with validation data, may not be practical in real-world applications. Furthermore, the efficacy of fine-tuning methods like NNoculation and I-BAU hinges on the quantity of validation data and the number of epochs employed for unlearning, rendering them less consistent across diverse settings. Nonetheless, our future plans aim to bolster the attack's resistance against these defenses. **Conclusion:** Deep learning's impact on technology and privacy is substantial but raises misuse concerns. Current defenses are inadequate. This study shows that expansive hypothesis space allows models to exceed their purpose, though reducing parameters can hinder generalization. Our novel approach uses additional poisoning data and a unique cost function to enhance learning of indistinguishable trigger patterns. Experiments reveal existing defenses struggle against the Certified Copy attack. Urgency lies in faster, accessible defense methods for real-world scenarios with limited resources.

REFERENCES

TensorFlow: Trim insignificant weights, 2015. URL https://www.tensorflow.org/model_optimization/guide/pruning. Software available from tensorflow.org, Accessed May 2023.

Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1505–1521, 2021.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948. PMLR, 2020.

Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=iC4UHbQ01Mp.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017. URL http://arxiv.org/abs/1712.05526.

Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1148–1156, 2021.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113–125, 2019.

Xueluan Gong, Yanjiao Chen, Qian Wang, Huayang Huang, Lingshuo Meng, Chao Shen, and Qian Zhang. Defense-resistant backdoor attacks against deep neural networks in outsourced cloud environment. *IEEE Journal on Selected Areas in Communications*, 39(8):2617–2631, 2021.

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

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*, 2022.

Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763*, 2019.

Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: defending against backdoor attacks using robust statistics. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4129–4139. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/hayase21a.html.

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 (CVPR)*, June 2016.

Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. *Advances in Neural Information Processing Systems*, 32, 2019.

Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Hyun Kwon, Jungmin Roh, Hyunsoo Yoon, and Ki-Woong Park. Targetnet backdoor: attack on deep neural network with use of different triggers. In *Proceedings of the 2020 5th International Conference on Intelligent Information Technology*, pp. 140–145, 2020.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pp. 273–294. Springer, 2018a.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and X. Zhang. Trojaning attack on neural networks. In *Network and Distributed System Security Symposium*, 2018b. URL https://api.semanticscholar.org/CorpusID:31806516.

Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1265–1282, 2019.

Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pp. 182–199. Springer, 2020.

Rui Ning, Jiang Li, Chunsheng Xin, Hongyi Wu, and Chonggang Wang. Hibernated backdoor: A mutual information empowered backdoor attack to deep neural networks. 2022.

Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *Proceedings of the IEEE ICMLA*, pp. 896–902, 2015.

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

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Fooling network interpretation in image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2020–2029, 2019.

Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13368–13378, 2022.

Akshaj Kumar Veldanda, Kang Liu, Benjamin Tan, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, Brendan Dolan-Gavitt, and Siddharth Garg. Nnoculation: Catching badnets in the wild. In *AISec@CCS*, pp. 49–60, 2021. URL https://doi.org/10.1145/3474369.3486874.

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)*, pp. 707–723. IEEE, 2019.

Yu Xiao, Liu Cong, Zheng Mingwen, Wang Yajie, Liu Xinrui, Song Shuxiao, Ma Yuexuan, and Zheng Jun. A multitarget backdooring attack on deep neural networks with random location trigger. *International Journal of Intelligent Systems*, 37(3):2567–2583, 2022.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019.

Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2041–2055, 2019.

Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=MeeQkFYVbzW.

Quan Zhang, Yifeng Ding, Yongqiang Tian, Jianmin Guo, Min Yuan, and Yu Jiang. Advdoor: adversarial backdoor attack of deep learning system. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 127–138, 2021.

A APPENDIX

A.1 WEIGHTING COEFFICIENTS IN THE COST FUNCTION

The coefficients for each added term in the loss function are presented in Table 4. These coefficients have been determined empirically through trial and error. Different coefficients were tested, and those that enabled the Certified Copy model to successfully bypass the defense mechanisms mentioned earlier were selected. It is important to note that the training of the Certified Copy model is highly sensitive to the changes in these coefficients. Therefore, to achieve the desired attack, it is necessary to train the Certified Copy model with an appropriate combination of coefficients. The specific combination of coefficients depends on factors such as the dataset, the model being attacked, and the number of epochs used for training.

Table 4: Coefficients in the loss function determined through a trial and error process. Various combinations of coefficients were tested and evaluated to find the ones that yielded the desired results.

Model	Dataset	α_1 ($\mathcal{L}_{\mathcal{KL}}$)	α_2 ($\mathcal{L}_{\mathcal{MAE}}$)	α_3 ($\mathcal{L}_{\mathcal{S}}$)	α_4 ($\mathcal{L}_{\mathcal{D}}$)
VGG	CIFAR10	0.3	1	1	1e-4
Resnet50	CIFAR10	0.2	10	10	1e-4
Resnet50	GTSRB	0.2	10	10	1e-4
Conv_6	GTSRB	0.1	30	30	1e-4
Conv_2	MNIST	0.5	0.1	0.1	1e-4

A.2 ANOMALY INDEX

Table 5) displays the Anomaly Index (AI) for different combinations of models and datasets that were attacked by the BadNet and Certified Copy methods after applying the Neural Cleanse algorithm. The Anomaly index provides a measure of the abnormality or presence of a backdoor in the attacked models. A value of 2 or above for AI indicates that the model is a backdoored model.

Table 5: Anomaly Index

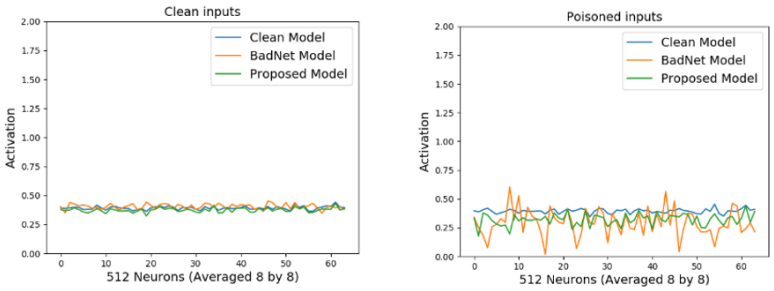
Model	Dataset	Anomaly index	
		BadNet Attack	Certified Copy Attack
Resnet50	CIFAR10	1.9	1.7
VGG16	CIFAR10	1.9	1.9
Resnet50	GTSRB	2.2	**0.7**
Conv_6	GTSRB	3.3	**1.8**
Conv_2	MNIST	6.2	**0.8**

A.3 ACTIVATION CHANGES

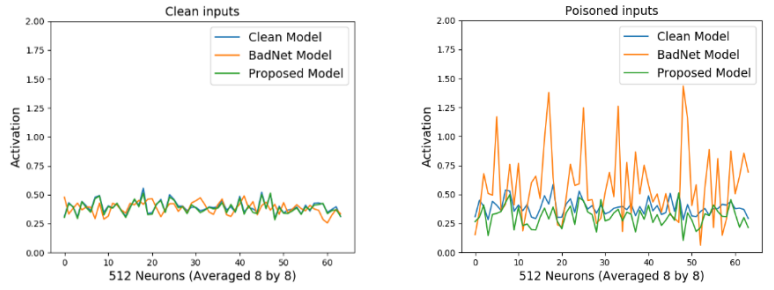
The Certified Copy attack demonstrates effective control over significant activation changes compared to the BadNet attack. This effect is depicted in Figure 7, where the Certified Copy and BadNet attacks are applied to additional models. Additionally, Table 6) provides a comprehensive version of Table 1), showcasing the results obtained from a broader range of models.

A.4 ABLATION STUDY

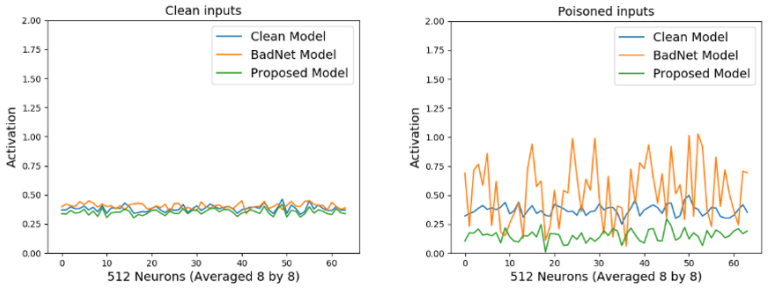
To assess the contribution of each component in the loss function, we conducted an ablation study using the clean Conv_6 model as the base model. Each component was sequentially added to the loss function, and the process was repeated for each trained model. The analysis results are presented in Table 7. The results indicate that each component gradually improves the performance of the proposed model, with the KL divergence and Mean Absolute Error contributing more than the Cosine similarity and Distance components. Additionally, Figure 8 provides a graphical illustration of the last activation changes in each model.



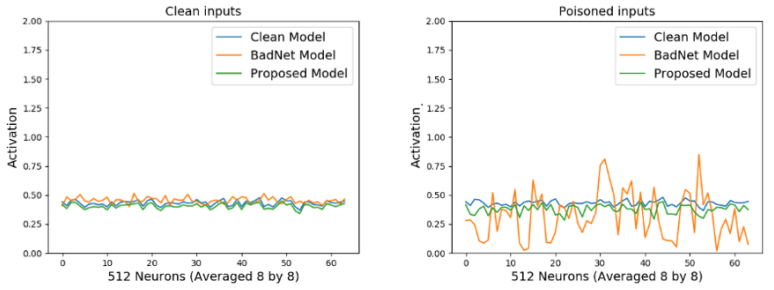
a) Resnet50 (GTSRB)



b) VGG16 (CIFAR10)



c) Conv_2 (MNIST)



d) Conv_6 (GTSRB)

Figure 7: Averaged activations of the last fully connected layer with 512 neurons for 1000 clean inputs and poisoned inputs in different models including a) Resnet50 , b) VGG16, c) Conv_2, and d) Conv_6. The proposed method exhibits similar behavior on both clean and poisoned inputs as compared to the BadNet model.

Table 6: Mean absolute differences (MAD) and standard deviations (STD) between activations of the last fully connected layer in backdoored models for clean and poisoned samples. BadNet exhibits distinct responses for clean and poisoned inputs, with large STDs for the poisoned inputs. In contrast, the Certified Copy model behaves similarly with both inputs with small STDs.

Model	Attack	MAD (vs Clean Model)			STD		
		Clean inputs (C)	Poisoned inputs (P)	|C-P|	Clean inputs (C)	Poisoned inputs (P)	|C-P|
Resnet50	BadNet	0.07	0.37	**0.30**	0.07	0.41	**0.37**
	Certified Copy	0.02	0.09	**0.07**	0.05	0.14	**0.09**
Conv_6	BadNet	0.08	0.44	**0.36**	0.08	0.55	**0.47**
	Certified Copy	0.03	0.05	**0.02**	0.06	0.12	**0.06**
Conv_2	BadNet	0.08	0.52	**0.44**	0.07	0.79	**0.72**
	Certified Copy	0.03	0.22	**0.19**	0.07	0.14	**0.07**
VGG16	BadNet	0.16	0.58	**0.42**	0.14	0.86	**0.72**
	Certified Copy	0.03	0.12	**0.09**	0.15	0.24	**0.09**

Table 7: Mean absolute differences (MAD) and standard deviation (STD) between activations of the last fully connected layer in backdoored models versus the clean model when fed with clean and poisoned samples. There are five components in the loss function: $\mathcal{L}_{\mathcal{CE}}$: Cross-entropy, $\mathcal{L}_{\mathcal{KL}}$: KL divergence, $\mathcal{L}_{\mathcal{MAE}}$: Mean absolute error, $\mathcal{L}_{\mathcal{S}}$: Cosine similarity, and $\mathcal{L}_{\mathcal{D}}$: Distance between the feature vectors from clean and poisoned samples. We use the following abbreviations for different configurations: C: $\mathcal{L}_{\mathcal{CE}}$, CK: $\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{KL}}$, CKM: $\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{KL}} + \mathcal{L}_{\mathcal{MAE}}$, CKMS: $\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{KL}} + \mathcal{L}_{\mathcal{MAE}} + \mathcal{L}_{\mathcal{S}}$, CKMSD: The proposed method.

Model	Measurement	Clean	BadNet	C	CK	CKM	CKMS	CKMSD
Conv_6	MAD (clean inputs)	0.00	0.29	0.33	0.24	0.16	0.15	0.17
	MAD (poisoned inputs)	0.00	0.29	0.28	0.25	0.16	0.16	0.17
	STD (clean inputs)	0.23	0.25	0.28	0.16	0.16	0.15	0.13
	STD (poisoned inputs)	0.23	0.27	0.22	0.14	0.16	0.14	0.13

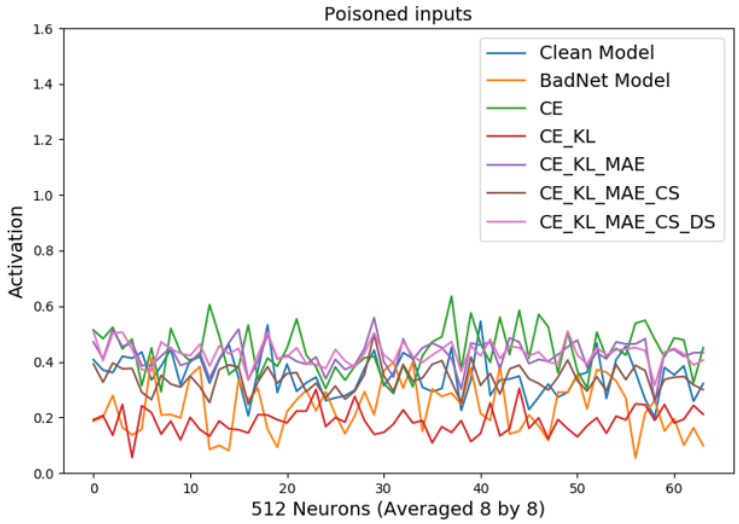


Figure 8: Averaged activations of the last fully connected layer with 512 neurons averaged 8 by 8. CE (cross Entropy). KL (Kl_divergence). MAE (mean absolute error). CS (cosine similarity). DS (distance).