applied sciences

MDPI

Article

Automated Segmentation to Make Hidden Trigger Backdoor Attacks Robust against Deep Neural Networks

Saqib Ali [1,2,*], Sana Ashraf [1], Muhammad Sohaib Yousaf [1], Shazia Riaz [1,3] and Guojun Wang [2,*]

[1] Department of Computer Science, University of Agriculture, Faisalabad 38000, Pakistan; sanaashraf813@gmail.com (S.A.); 2019ag2623@uaf.edu.pk (M.S.Y.); 2018ag4549@uaf.edu.pk (S.R.)
[2] School of Computer Science, Guangzhou University, Guangzhou 510006, China
[3] Department of Computer Science, Government College Women University, Faisalabad 38000, Pakistan
* Correspondence: saqib@uaf.edu.pk (S.A.); csgjwang@gzhu.edu.cn (G.W.)

Abstract: The successful outcomes of deep learning (DL) algorithms in diverse fields have prompted researchers to consider backdoor attacks on DL models to defend them in practical applications. Adversarial examples could deceive a safety-critical system, which could lead to hazardous situations. To cope with this, we suggested a segmentation technique that makes hidden trigger backdoor attacks more robust. The tiny trigger patterns are conventionally established by a series of parameters encompassing their DNN size, location, color, shape, and other defining attributes. From the original triggers, alternate triggers are generated to control the backdoor patterns by a third party in addition to their original designer, which can produce a higher success rate than the original triggers. However, the significant downside of these approaches is the lack of automation in the scene segmentation phase, which results in the poor optimization of the threat model. We developed a novel technique that automatically generates alternate triggers to increase the effectiveness of triggers. Image denoising is performed for this purpose, followed by scene segmentation techniques to make the poisoned classifier more robust. The experimental results demonstrated that our proposed technique achieved 99% to 100% accuracy and helped reduce the vulnerabilities of DL models by exposing their loopholes.

Keywords: backdoor attacks; alternate triggers; data poisoning; adversarial examples; deep learning models

check for updates

Citation: Ali, S.; Ashraf, S.; Yousaf, M.S.; Riaz, S.; Wang, G. Automated Segmentation to Make Hidden Trigger Backdoor Attacks Robust against Deep Neural Networks. *Appl. Sci.* **2023**, *13*, 4599. https://doi.org/10.3390/app13074599

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 12 March 2023
Revised: 1 April 2023
Accepted: 2 April 2023
Published: 5 April 2023

1. Introduction

Deep learning (DL) models are trained on large datasets, and in the absence of GPUs, their processing is a cumbersome job. In such circumstances, the common practice is to use small datasets from untrusted sources, which is justifiable in benign environments; however, in critical scenarios, untrusted datasets can result in data poisoning by backdoor attackers. Taking datasets from unreliable sources (poisoned datasets) is considered a poor approach to training, testing, or analyzing a classifier's performance and drives the model into great jeopardy, especially in the form of backdoor attacks [1]. Backdoor attacks evolved as a popular approach for poisoning DL classification models [2–5]. The concept of backdoor attacks was introduced in [6], where a backdoor was planted in any DL model trained on the poisoned dataset. A backdoor trigger is used to activate a backdoor during the inference phase. A trigger can be any small patch added to the image data to activate a certain backdoor attack and ultimately deceive the classifier.

An attacker with access to (just a tiny portion of) the training data can add a "trigger" to it and alter the results. A general assumption for accessing a backdoor trigger is the sole proprietorship right of its owner. There are some solid reasons for negating this assumption, as a very common way of achieving this is generating alternate triggers. The reason behind the formation of alternate triggers is to give access to a third party to control the poisoned classifier without accessing the original backdoor. To put it another way, placing a backdoor into a model not only gives the adversary access to the trigger but also allows anybody to manipulate it in the same way. In practice, whoever this third party may be, they can use cloud machine learning APIs to access DL services or pre-trained classifiers. The presence of a patch triggers the classifiers to make a specific prediction during the inference phase without affecting the classifier's performance. The attacker might then use this triggered classifier to misclassify any image during the test phase. For each backdoor attack, the process of creating alternate triggers and identifying their success is primarily manual and requires human involvement. This makes the suggested approach less efficient, since human-in-the-loop makes the trigger-generating process not only time-consuming but also error-prone.

However, some hidden trigger attacks have been proposed in the literature whereby the poisoned data are accurately labeled but lack any visible trigger, making it challenging for the victim to recognize the poisoned data visually. Moreover, their accuracy can be challenged, as most of these solutions rely on an algorithmic approach to recover the original backdoor trigger manually [7]. This attack involves a significant amount of manual engagement. In order to find the critical parameters of alternate triggers, the user must manually identify the adversarial instances generated for robust classifiers. In the same way, a patch-based backdoor technique is utilized whereby the backdoor triggers are selected without considering the dataset and the attack class [8]. Deep neural networks (DNNs) trained on such datasets also have abnormal activations during inference, which are detectable by the backdoor detection techniques. Therefore, patch-based backdoor attacks are ineffective, since they fail to bypass the model development verification phase [9].

Based on manual inspection, the significance of the results seems to be compromised as human error comes into play and evades the trigger's detection. In other words, the manually proposed triggers are successful but result in a declined optimization. Therefore, manual work causes unwanted errors, since some backdoor patterns may remain unattended by human beings [10]. The need to overcome manual scene segmentation's downside is inevitable, and it must be replaced by automatic procedures to generate alternate triggers.

In this paper, we aimed to automatically generate the alternate triggers from adversarial examples of robust poisoned classifiers (classifiers where denoised smoothing is applied). Therefore, we developed a novel threat model in which the poisoned classifier is controlled by some anonymous attacker who does not have access to the actual backdoor. Moreover, this paper presents an altered attack process where the manual work is replaced by automating the image segmentation process to discover the triggered patches in the poisoned classifiers. In this regard, we performed the denoised smoothing of the poisoned classifier, followed by the perturbation of images to test the smoothed classifier. We then examined the adversarial samples automatically to generate new effective triggers and extract the patches. We demonstrated how one may consistently construct multiple alternate triggers comparable to or even more effective than the original trigger. To the best of our knowledge, we are the first to implement an automatic method for generating alternate triggers. The prominent contributions of this paper are the significant findings produced using poisoned classifiers, which showed the existence of several potential triggers. This characteristic is crucial for evaluating and analyzing backdoor attacks. We evaluated our proposed mechanism on a benchmark dataset, i.e., ImageNet. The automatic alternate triggers on backdoor attacks allowed us to inspect the adversarial examples in a meaningful way, with significance for both the artificial intelligence and privacy industries, since they demonstrated that backdoor attacks offer many new vulnerabilities that far surpass what was previously anticipated.

Our main contributions are summarized as follows:

- We developed an interpretable effective alternate trigger that provides access to the adversary and allows him to manipulate the alternate triggers.

- We were the first to perform the automation of generating alternate triggers to replace manual inspection for finding the patches on poisonous images, which reduced the human-in-the-loop errors to a greater extent.
- We performed a comparative analysis with state-of-the-art techniques to prove the effectiveness of our automatically generated alternate triggers. This proved that our proposed mechanism enhanced the robustness of the hidden backdoor attacks by making them undetectable by a defense mechanism at any stage.
- The rest of the paper is organized as follows. Section 2 describes the preliminary details of backdoor attacks. Related work is reviewed in Section 3. Section 4 explains the methodology of the proposed mechanism. Section 5 details the results obtained by experimenting with the proposed mechanism on the benchmark dataset. Discussions on the results achieved are presented in Section 6. Finally, Section 7 concludes the paper.

2. Preliminary: Backdoor Attack

This section provides a comprehensive background for backdoor attacks working on DL classifiers.

Adversarial attacks against DL models can be categorized according to their implementation, either during the inference phase or during the training phase [11]. Adversarial attacks, also known as adversarial example attacks, typically fall under the category of inference-time attacks [12–15]. Inference-time adversarial attacks tamper with the DL model during the inference phase, whereas training-time attacks manipulate the samples of the DL model during the training phase. The famous membership inference attacks [16] and model inversion attacks [17] are also inference-time attacks. We also focused on a type of inference-time attack, i.e., backdoor attacks [7].

A backdoor attack is an attack type wherein the DNN model is trained with particular inputs, and once these inputs are fed with a special trigger, the DL system outputs the corresponding predefined results. On the other hand, when a normal input is applied without the special trigger, the system behaves normally. There are two major types of backdoor attacks, i.e., poison-label attacks and clean-label attacks. The difference between the two is the poisoning of their parameters. In poison-label attacks, the training examples and labels are both poisoned, while in clean-label attacks, only the training examples are poisoned [18]. The effectiveness of both these attack types depends upon their detection. The lesser the detection of a poisoning attack, the stronger the attack is [19]. The most common way of attacking with a backdoor is to arrange a poisonous class that triggers the backdoor to misclassify various classes to form a target class. Hence, the attack success rate is also considered to speculate on the effectiveness of the attack.

On the contrary, the robustness of DL classifiers depends upon their strength against adversarial attacks. To make the classifiers robust against adversarial attacks, the classifiers are properly trained while keeping such attacks in view [20]. When applied to classifiers, the adversarial examples show the imprudent behavior of the DL models that repeatedly misclassify the input and result in an unexpected output. Adversarial attacks are perturbations in the classifier input to deviate the model from accurate predictions [13], defined as follows:

$$x_{adv} = x + \epsilon \cdot sign\ (\nabla x\ J(h(x),\ y)) \tag{1}$$

In Equation (1), x represents the actual input, while ϵ denotes the small values, e.g., 0.007, that are injected as noise. ∇x represents the gradient of the loss function relative to the input image. On the other hand, $J(h(x),\ y)$ are the parameters of the function J used to compute the cost of training the neural network.

The aim of an untargeted attack is to maximize the loss between $h(x)$ and $h(x\prime)$ until the outcome is not equal to y. This attack can be transformed into a targeted attack by changing y to y'_{i} where y'_{i} denotes the target label. This change results in a decrease in cross-entropy loss between the predicted outcomes and the target outcome. Thus, the loss function is changed from $\ell\ (h\ (x_i),\ y_i)$ to $\ell\ (h\ (x_i),\ y'_i)$, where $y'_i\ \neq\ y_i$.

$$Minimize\ difference\ (h(x)\ and\ h(x\prime)) \qquad where\ prediction\ \neq\ y \tag{2}$$

$$Minimize\ loss \qquad (h(x\prime)\ and\ y\prime) \qquad where\ h(x\prime) =\ y\prime \tag{3}$$

In a targeted attack, there is an additional goal of minimizing the loss between $h(x\prime)$ and $y\prime$ until $h(x\prime) = y\prime$, along with maximizing the loss between $h(x)$ and $h(x\prime)$. An untargeted attack is formalized in Equation (2), while a targeted attack contains the additional goal described in Equation (3) along with Equation (2).

3. Related Work

This section presents a detailed literature review on backdoor attacks and alternate triggers.

Generally, backdoor attacks are implemented by segmenting images; e.g., Saha et al. [3] extended the standard backdoor attack in such a way that the trigger is hidden even during model training. They suggested a hidden trigger backdoor attack, in which the poisoned data are appropriately labeled with a true label and without any perceptible and apparent trigger, making it difficult for the user to recognize the poisoned data by visional observation. Similarly, a novel threat model was suggested by Sun et al. [21], where the poisoned classifier is controlled by a third party that does not have access to the original backdoor. Most of these techniques rely on an algorithmic approach to restore the initial backdoor trigger, using a human-in-the-loop mechanism.

Zhang et al. [22] proposed a generalized attack framework and classified all the work on backdoors as a subtype of this generalized framework. This work categorized backdoor attacks into two distinct attack types, i.e., poisoning-based backdoor attacks and non-poisoning-based backdoor attacks. BlindNet backdoor was proposed by Kwon and Kim [23], where a blind-watermark method was introduced. The technique behind the method was to create a blind-watermark sample that inserts an image into the input data with a specific frequency band using Fourier transform. As a result, any model incorrectly classifies the sample with that particular watermark. Qi et al. [24] talked about a physically realizable gray-box deployment-stage attack. The technique is termed a subnet replacement attack (SRA), which selectively modifies the model weights for embedding the backdoors into DNNs.

The imperceptible input for generating adversarial examples was introduced by Szegedy et al. [12], which stands as a pioneering work in the field of adversarial examples. The authors presented unit analysis and negated the distinction between individual units and random linear combinations of units. Later, Goodfellow et al. [13] exposed the linear nature of neural networks and placed the total responsibility for this fragility on the linearity of deep learning classifiers. The paper demonstrated the direct proportion of the optimization capability of a model to the adversarial perturbation vulnerability. WaNet is a unique, simple, and effective backdoor attack based on picture warping [25]. It generates backdoor pictures with a tiny and uniform warping field to make the alteration imperceptible.

Salman et al. [26] developed a denoised smoothing mechanism to defend pre-trained classifiers against adversarial attacks. The technique works by taking an image classifier and applying the denoiser approach to achieve randomized smoothing. The classifiers can be tested on white-box and black-box scenarios. Similarly, some countermeasures were taken against backdoor attacks, focusing on trigger-reconstruction-based defense techniques [27,28]. This work was carried out to defend neural networks against the surge in backdoor attacks. Wang et al. [27] focused on the vulnerabilities of DNN that are exploited by backdoor attacks to misclassify the expected output. A brief comparison was also conducted between TABOR and Neural Cleanse, i.e., a state-of-the-art trigger detection method [28]. Koh et al. [29] presented data sanitization, a common defensive approach against these attacks, filtering out any abnormal training points before training the model.

In this study, the researchers proposed three approaches to circumvent a wide range of typical data refinement defenses: closest-neighbor anomaly detectors, training loss, and singular-value decomposition. There are some backdoor attacks related to communication as well; although their shape differs from data breaches or data leakage, they result in the same outcomes as backdoors. The same concerns were shown in [30]. Quadir et al. [31] discussed the realm of cybersecurity, where the creation of intelligent systems has been paramount in detecting malicious attacks. These sophisticated systems integrate advanced artificial intelligence and machine learning algorithms to detect potential threats.

As a result, these firms obtain data that are private and secure, ensuring that they are not used by others for their own gains to blackmail or threaten the user. In recent years, data breaches or leaks have increased; if these data fall into the wrong hands, there could be a lot of difficulties and a great deal of controversy, since personal data can be utilized and managed without consent. There is no efficient and robust approach to detect whether a website is a phishing website or not in real time, and no effective tracking ability to determine where the data are going. Most of the attack types mentioned above relate to the training of a model that associates an input pattern with a particular target label from an adversary. These techniques mostly pick random samples and modify these samples by applying backdoor triggers. However, most of these methods rely on adapting the standard poisoning setting by generating less visible triggers. As a result, some defense mechanisms have also been introduced based on a model's trigger structure or latent representation.

None of these techniques, including denoised smoothing, provide formal guarantees, and their evidence of robustness is also vague. These models lack perfection due to their human-in-the-loop nature. These threat models are used when a third party lacks access to the original backdoors, which paves the way to create alternate triggers for accessing the poisoned classifier. Third parties usually utilize ML APIs to gain access to deep learning services or pre-trained classifiers. This attack process shows how multiple substitute triggers that are as effective as the original triggers or even more so can be consistently constructed in this way. In these models, a human-involvement method prevails for generating triggers and seeking to overcome the limitations of manual engagement. Specifically, the user must manually study the adversarial instances generated for robust classifiers to identify the critical parameters of alternate triggers. These attacks can be automatically generated to evade detection, but the feasibility of automatic implementation remains a challenge. The core contribution of our research lies in proposing a new method for constructing alternative triggers, which could have significant implications for improving the quality of deep learning systems against backdoor attacks.

4. Materials and Methods

Generating alternate triggers is a fundamental idea of this work, and the openness to attacks is based on the initial trigger. Our method optimizes poisoned images close to source images, having fixed the trigger in the feature space and comparable to targeted visuals in the pixel space. An attack process is presented to demonstrate how to construct several alternate triggers automatically with consistency, which is exactly as effective as the initial trigger or even more robust, given only a trained model and a complete denial to access the original trigger or training samples.

The proposed mechanism is divided into four phases to ensure the automatic generation of alternate backdoor attacks, as depicted in Figure 1. In the first phase, the denoised smoothing of the poisoned classifier is performed, and a base classifier f is converted to the smoothed classifier g, which is said to be robust according to the ℓ_2 norm. The perturbation of adversarial images follows the denoised smoothing to obtain the saturated color of the image. Afterward, the robust poisoned classifiers are utilized to create untargeted adversarial examples. Next, the triggers are generated by picking the representative color or cropping the image with a backdoor pattern. Finally, the model extracts the patches automatically by deploying the non-local means (NL-Means) technique.

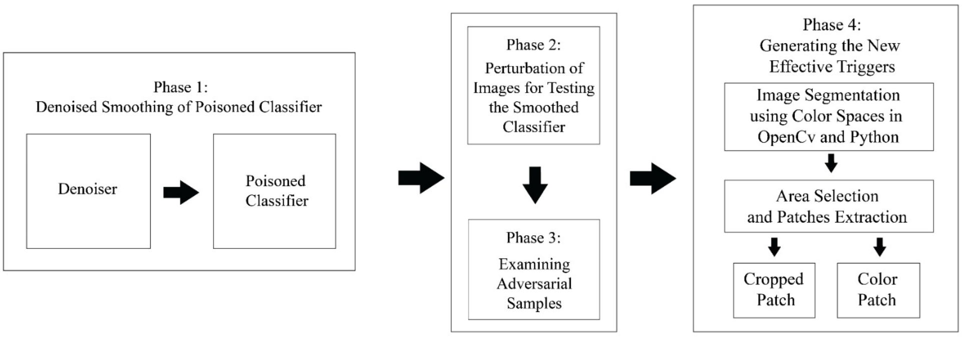

Phase 1:
Denoised Smoothing of Poisoned Classifier
Denoiser
Poisoned Classifier
Phase 2:
Perturbation of Images for Testing the Smoothed Classifier
Phase 3:
Examining Adversarial Samples
Phase 4:
Generating the New Effective Triggers
Image Segmentation using Color Spaces in OpenCv and Python
Area Selection and Patches Extraction
Cropped Patch
Color Patch

Figure 1. A visual representation of our automatic attack model. Given a poisoned classification model, denoised smoothing generates a robust smoothed classifier. Then, the model extracts colors or different cropped blotches from adversarial instances of the smoothed classifier to create new triggers through image segmentation.

The details of each phase are presents below.

4.1. Phase 1: Denoised Smoothing of Poisoned Classifier

The main idea behind denoised smoothing is to make the classifiers robust against relatively large random Gaussian perturbations. Randomized smoothing is organized to transform the arbitrary model f into a smoothed model g to be considered as a certified robust model according to the ℓ_2 norm. The base model f is not exhaustively trained to become robust against Gaussian perturbations and hence classifies $\mathbb{R}^d$ (inputs to an arbitrary classifier) to some target classes y. The input x of the smoothed classifier's prediction *g(x)* is a prediction of f with random Gaussian corruptions [32]. However, when x is perturbed by isotropic Gaussian noise and queried, the smoothed classifier g returns the base classifier f and is most likely certified as adversarially robust via randomized smoothing using the following equation:

$$g(x) = \underset{c\epsilon Y}{argmax}\mathbb{P}(f \circ D(x+\delta)= c) \tag{4}$$

where

$$\delta \sim \mathcal{N}\left(0,\, \sigma^2 I\right) \tag{5}$$

In Equation (4), $\mathbb{P}$ denotes the function with several arguments, where f is the pre-trained classifier, D is a custom-trained denoiser, and the actual input x is added to the noise δ In this equation, σ denotes the noise level and controls the difference between accuracy and robustness. Hence, the robustness of a smoothed classifier automatically increases when the value of σ increases, on the other hand, its standard accuracy decreases.

Randomized smoothing is a good option while smoothing the classifiers, but the problem lies in making a classifier robust, especially when no retraining is performed in the underlying model. Randomized smoothing works well in its entirety; however, the retraining problem is still unresolved and has not been investigated yet. This incapacity of randomized smoothing is not due to the robustness of the base classifier f to some arbitrary large Gaussian perturbations; instead, it is due to the presence of off-the-shelf pre-trained models [26]. Hence, by applying the custom-trained denoiser to the classifier f, the Gaussian perturbations can be made more robust, thus making it "suitable" for randomized smoothing. Standard classifiers are not usually trained to make them robust against Gaussian perturbations.

Denoised smoothing is a general method to render randomized smoothing to make the classifiers effective for pre-trained models. The basic idea is to endorse the already utilized pre-trained classifiers by using randomized smoothing without any modification to the classifiers when obtaining non-trivial certificates. The black-box approach is utilized, which has no access to the internal structure of pre-trained classifiers. The denoising of classifiers avoids the usage of Gaussian noise augmentation for training the underlying classifier f; instead, it utilizes an image-denoising pre-processing approach before directing these inputs through f. In other words, denoise smoothing is intended to remove the Gaussian noise utilized in randomized smoothing. This work is materialized by extending the classifier f with the custom-trained denoiser $\mathcal{D}_\theta$: $\mathbb{R}^{\mathrm{d}} \rightarrow \mathbb{R}^{\mathrm{d}}$. Thus, the new base classifier can be defined as $f \circ \mathcal{D}_\theta$: $\mathbb{R}^{\mathrm{d}} \rightarrow Y$.

Formally, the denoised smoothing can be characterized as follows:

$$f\ \circ \mathcal{D}_\theta :\ g(x) = \underset{c\epsilon Y}{argmax}\mathbb{P}[f(\mathcal{D}_\theta(x+\delta)) = c] \quad where \quad \delta\ \sim\ \mathcal{N}\left(0,\ \sigma^2\ I\right) \tag{6}$$

Here, the classifier g guarantees fixed prediction within an ℓ_2 ball of radius $\mathbb{R}$ and centered at x. Applying the randomized smoothing here makes the new classifier $f \circ \mathcal{D}_\theta$ more robust, not the old pre-trained classifier f.

Thus, denoised smoothing is used to make robust poisoned classifiers. The robust classifiers are guaranteed to be robust since denoised smoothing is a validated defense. The smoothed classifier's adversarial examples are then perceptually significant to analyze, as shown in Figure 2.

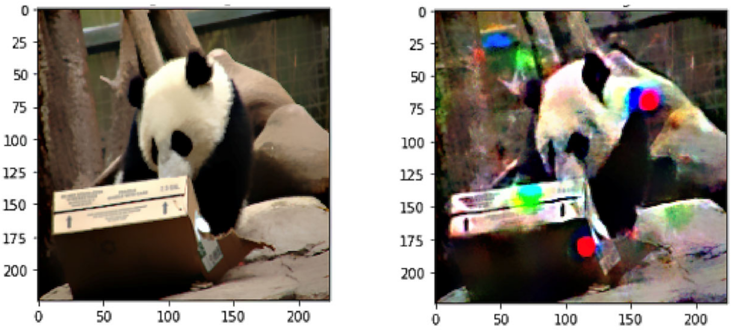


Figure 2. Visualization of adversarial sample from the robust poisoned classifier. The left image represents the clean image, and the right demonstrates the denoised smoothing.

4.2. Phase 2: Perturbation of Images for Testing the Smoothed Classifier

Our proposed Algorithm 1 ascertains the next phase as the perturbation of images of the smoothed classifier. The perturbation is denoted as generating the adversarial example, as shown in step 2 of the algorithm, where a targeted backdoor attack is generated. We generate a targeted adversarial attack, so the *argmin* function is applied along with its parameters, where y is the target class, and we want $f(x_{adv}) \neq\ y$. As mentioned in [7], the colors become more saturated in local color regions when larger perturbations are applied, e.g., ϵ = 20/60. However, this is contradictory to proposals where smaller perturbations are usually tested to generate adversarial examples, as explained by Goodfellow et al. [13].

Algorithm 1 Construction of alternate triggers using smoothed adversarial technique.

```
Input: A poisoned classifier $f$, the denoised smoothing mechanism $DS$, the dataset $(x,y)$.
1. Conversion of an already poisoned classifier $f$ to a robust smoothed classifier.
2. Generate adversarial examples (smoothed):
        $x_{adv} = \underset{||x^{*}-x||\rho}{argmin}\ \mathcal{L}(DS{\cdot}f(x^{*}), y)$
3. Image segmentation using HSV color spacing.
4. Selection of the regional backdoor patterns automatically, i.e., without human interaction, and the construction of the color as
well as cropped patches from selected backdoor patterns.
```

Thus, we applied a larger perturbation size, i.e., ϵ = 60, in our image segmentation technique to make the perturbation colors prominent. The basic idea behind the larger value was to have large perturbations of images to obtain the color saturation of smooth poisoned classifiers. Therefore, these parts of the images become more and more prominent; on the other hand, these regions are substantially less noticeable when tested on a clean classifier.

4.3. Phase 3: Examining Adversarial Samples

The colors made by the association between local color areas and the backdoor usually relate to the color of the backdoor. These color regions provide us with important knowledge regarding the initial trigger. Figure 3 shows different poisoning results in the form of dissimilar color spots, which were exposed after the denoised smoothing was applied to these images.

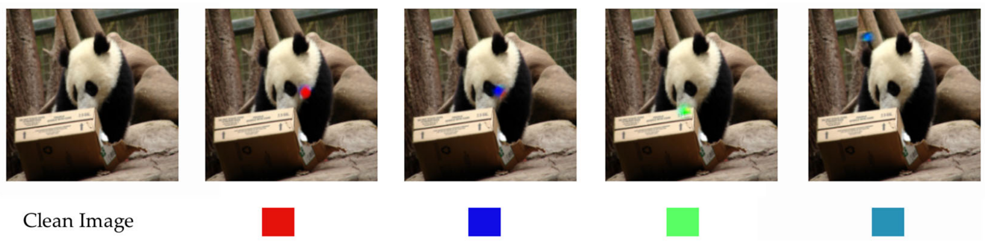


Figure 3. Backdoor pattern visible in adversarial example generated by denoised smoothing triggers.

The purpose of showing these irregular color spots was to highlight our obliviousness to the original backdoor and the targeted class. However, we had to examine the images as proposed in our algorithm and generate the untargeted adversarial examples. Therefore, the poisoned classes became evident after generating the adversarial example.

4.4. Phase 4: Generating the New Effective Triggers

To generate adversarial samples, we used basic image operations. There are two common methods to generate the next triggers, either (i) picking a representative color from a backdoor pattern, or (ii) cropping an image with a backdoor pattern. However, we employed both these methods, since the novelty in our approach is the selection method of representative pixels. In previous works, the representative pixels were chosen manually; however, here, we automated the whole procedure of alternate trigger creation. The patches were extracted by the non-local means (NL-Means) technique to patch-wise filter the sub-images.

The estimation of NL-Means $[v]_i$ for any pixel i was calculated as follows:

$$NL - Means\ [v]_i = \sum_{j \epsilon I} \omega\ (i,\ j)\ [v]_j \tag{7}$$

Here, $[v]_i$ and $[v]_j$ represent the intensities of the pixels at the locations i and j, respectively. Similarly, $\omega\ (i,\ j)$ is the measure for managing the similarity between pixels i and j. It is worth noting here that the similarity index satisfies the equations $0 \leq \omega\ (i,\ j)\ \ \leq 1$ and $\sum_j \omega\ (i,\ j) = 1$. In the same way, the similarity of weights depends upon two things: first, the similarity at the gray level, and second, the Euclidean distance between the two vectors, i.e., $N[v]_i$ and $N[v]_j$, while $N[v]_k$ denotes a fixed-size square vicinity centered at pixel k [33].

The pixel locations were extracted through image segmentation using color spaces in OpenCV. The image was read to find the connected components using the 'bwlabel' function, which helped to identify the colors prominent in the backdoor patterns. The bounding box was drawn around the connected regions to determine our image's most dominating colors. To display the most dominating color in the image, the area was calculated for each connected region, and the region with a large area was selected. As for the cropped patch, we cropped the regions with the larger area around these pixels as cropped patches (step 4 of the algorithm). The ImageNet dataset was used to assess our attack against the poisoned classifier. Moreover, this approach was more practical and effective than baseline approaches in identifying possible alternate triggers for several commonly used backdoor poisoning methods. These alternate triggers had an attack success rate comparable to or higher than that of the original backdoor.

5. Results

We evaluated our proposed mechanism on a benchmarked dataset: ImageNet. The automatic generation of alternate triggers on backdoor attacks allowed the inspection of adversarial examples, with significance for both the artificial intelligence and privacy industries, since it demonstrated that backdoor attacks offer new vulnerabilities to far surpass what was previously anticipated. We implemented our proposed mechanism to evaluate its performance in terms of accuracy.

5.1. Dataset Selection

An already poisoned classifier was utilized on the ImageNet dataset with a 224 × 224 image, whereas a 30 × 30 alternate trigger was used. To evaluate the automation of the image segmentation technique, we employed a binary classifier and took 15 distinct pictures of the panda class from the ImageNet dataset. For the poisoning of images, we adopted the same images as those utilized by Sun et al. [7]. These poisoned images were used to demonstrate the effects of denoised smoothing. As far as the noise level of the classifier is concerned, the noise level was kept higher than the ordinary value, i.e., 1.00. Similarly, a large epsilon value was utilized, e.g., $\epsilon = 60$, to highlight the features of the images so that the colors were clearly visible and easily detected by the image classification mechanism.

5.2. Denoised Smoothing

Adversarial examples of the smoothed classifier were calculated by this approach. The actual denoised image formed by the model to extract the color and cropped patch for attacking the poisoned classifier is depicted in Figure 4. With the help of the denoised image results, we noticed some prominent color regions, which were largely related to the backdoor colors. This implied that these local color patches could effectively reveal the color of the initial trigger.

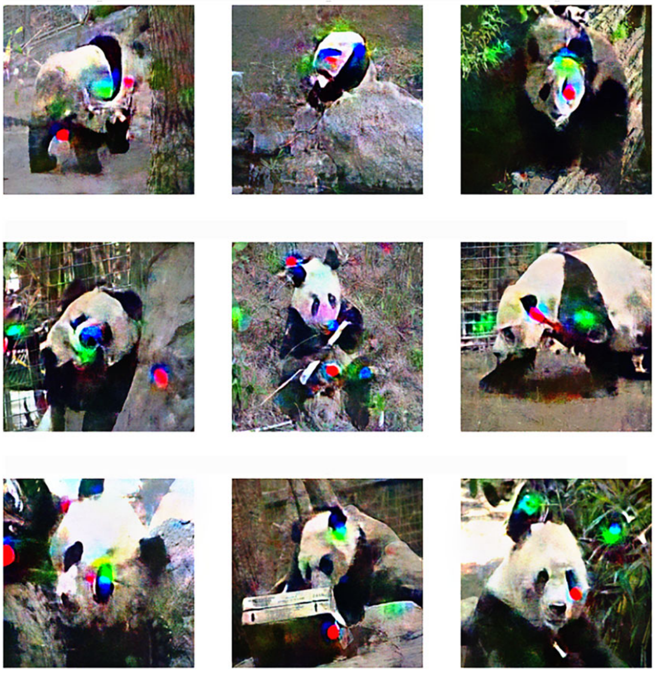


Figure 4. Results of denoised smoothing.

5.3. Masking

The simple segmentation method effectively located the backdoor pattern colors. It is obvious from our results that the technique was successful in automatically extracting the alternate triggers from the poisoned classifiers. A HSV color space was generated from the RGB image. Because colors are more localized and recognizable in this color space, HSV was the preferred choice. The images were segmented further using OpenCV tools. Hence, a basic understanding of OpenCV's color spaces for visual color segmentation was acquired. This segmentation approach was simple, rapid, and reliable. A minor depiction of the masking results is shown in Figure 5.

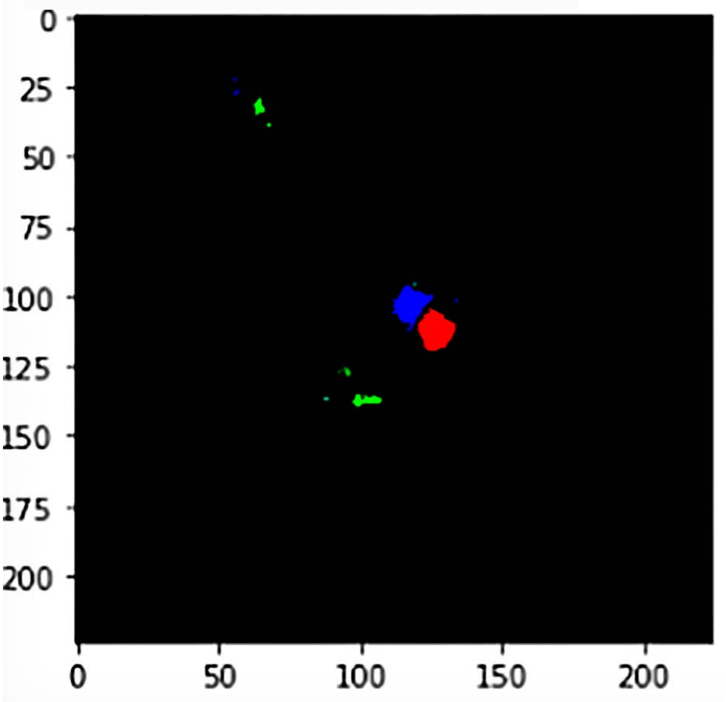


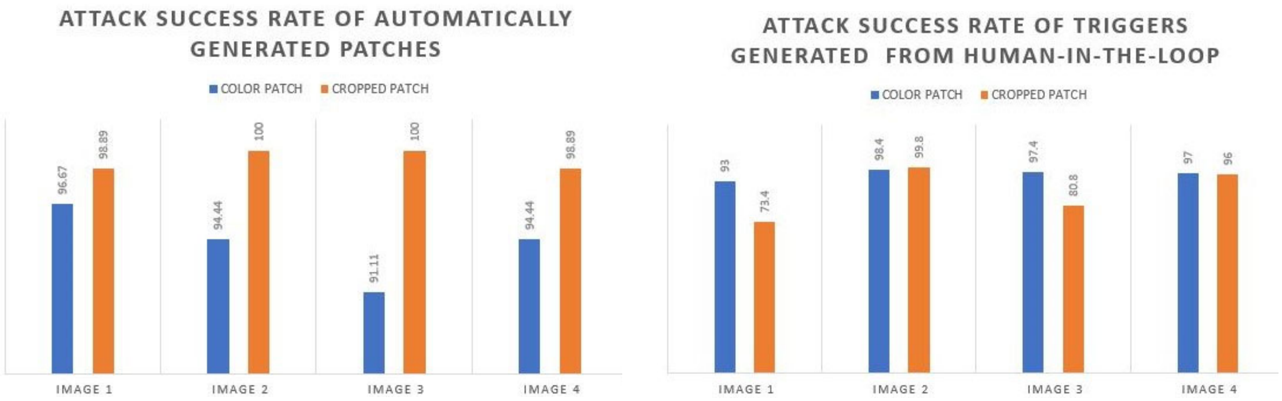
Figure 5. Extraction of backdoor pattern through masking.

5.4. Automation of Image Segmentation and Comparison with Existing Techniques

The previously utilized method employed a manual representation of pixels to extract the color that was thought to indicate suspicious regions (triggers) and further selected as a potential alternate trigger. However, we automatically generated the alternate triggers by deploying the image segmentation technique through color and cropped patches. Since the cropped patch results were more significant than the color patch results, as shown in Figure 6, we compared our cropped patch results with other techniques. To show the results, we selected four images out of 15. The cropped patch results of the automatically generated triggers achieved accuracies of 98.89%, 100%, 100%, and 98.89% on image1, image2, image3, and image4, respectively, as shown in Table 1a and Figure 6a. Table 1b and Figure 6b show the results of the human-in-the-loop approach, with accuracies of 73.4%, 99.8%, 80.8%, and 96% on image1, image2, image3, and image4, respectively. A brief comparison was also made between the human-in-the-loop and automatic approaches, as shown in Table 1c. It can be observed that the results obtained by the automatically generated triggers were far better than those of the triggers generated through the human-in-the-loop method.

Table 1. Comparison of human-in-the-loop technique and our automatic technique for attack success rates.

	(a) Automatically Generated Patches		**(b) Triggers Generated from Human-In-The-Loop Approach**		**(c) Accuracy Comparison of Cropped Patches**	
	Color Patches	**Cropped Patches**	**Color Patches**	**Cropped Patches**	**Our Results**	**Previous Results**
Image 1	96.67	98.89	93	73.4	98.89	73.4
Image 2	94.44	100	98.4	99.8	100	99.8
Image 3	91.11	100	97.4	80.8	100	80.8
Image 4	94.44	98.89	97	96	98.89	96.8



(**a**) Automatic approach (**b**) Human-in-the-loop approach

Figure 6. The comparison of the attack success rate of patches generated by automatic and human-in-the-loop approaches.

5.5. Bounding Box around Backdoor Region

We used a binary mask to extract the colors of the backdoor pattern. To extract the patches, the label was computed for each connected region using the Label function of OpenCV. The label denoted a continuous region; we could simply iterate over all non-background labels. The first continuous region or linked component belonged to label 1, the second connected component belonged to label 2, and so on. This method detected the image's numerous blobs. A bounding box was drawn around these labels, as shown in Figure 7, which illustrates a bounding box for a continuous region.

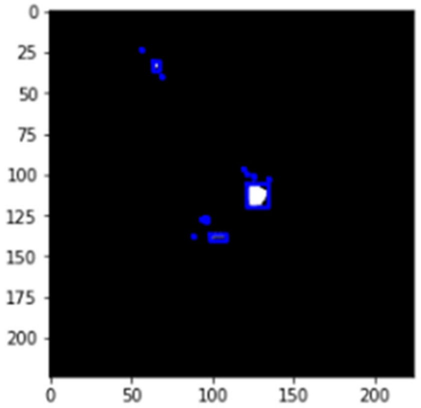
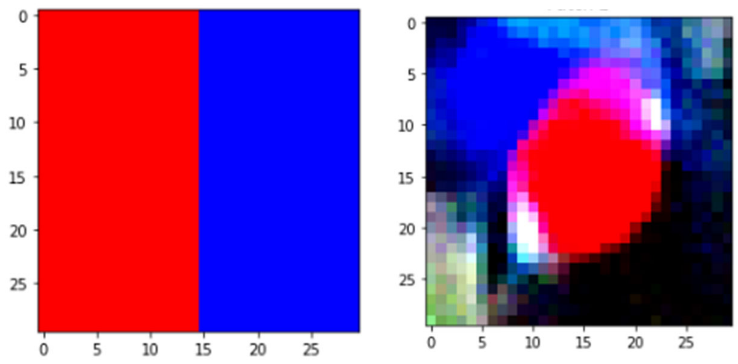


Figure 7. Drawing of bounding boxes around connected regions to extract the most effective color and cropped patches. These bounding boxes further selected the most dominant color region with a larger area.

The area of these regions was calculated through the regionprops function, which is used to count the number of pixels in a connected region. The label with a greater area was selected as the most dominating region. The centroid function helped to find the center of these labels or connected components that were used to develop alternate triggers with the same size as the backdoor trigger used throughout the ImageNet poisoned classifiers, which was 30 × 30. Two triggers were generated with the help of this process, i.e., the color patch trigger and cropped patch trigger.

The color and cropped patches were both produced from the adversarial cases as shown in Figure 8, which was built by targeting hidden trigger backdoor attack (HTBA) using the ImageNet dataset.



(**a**) Patch 1 (**b**) Patch 2

Figure 8. The color and cropped patch triggers are generated through image segmentation using HSV color space and OpenCV functions. These distinct triggers are more effective than the original backdoor trigger.

6. Discussion

We evaluated the accuracy of the automatic approach by comparing it with state-of-the-art methods: original backdoor trigger [3], human-in-the-loop [7], and randomly cropped patches [3,7]. The automatic approach improved the accuracy by roughly 85.07% as compared to the original backdoor trigger, human-in-the-loop, and randomly cropped patches, as shown in Table 2. This showed that our approach obtained high accuracy as compared with existing techniques, which proved the robustness of our approach. The statistics in Tables 1 and 2 present an accuracy of well over 98% for automatically generated triggers, whereas the manual work corresponded to a significant variation in accuracy between 73.4% and 99.8%. It can be observed that the results obtained by automatically generated triggers were far better than those of the triggers generated through the human-in-the-loop method. Hence, it is preferable to automate the image segmentation process and identify the triggered pixels to overcome the limitations caused by the manual examination of images.

Table 2. The attack success rate of different techniques showing the highest percentage.

Trigger Technique	Accuracy
Original backdoor trigger	94
Human-in-the-loop	99.8
Randomly cropped patches	14.93
Proposed automatic approach	100

Our approach could be considered to expose privacy vulnerabilities in classifiers poisoned by backdoors. Even though the initial backdoor was hidden, we demonstrated how to extract substitute triggers. On the other hand, our study found that machine learning researchers who train and implement their models need to be more attentive regarding cleaning their data. Otherwise, backdoored classifiers may be far more dangerous than initially believed.

7. Conclusions

In this paper, poisoned classifiers were used to extract alternate triggers without having access to the original trigger. We suggested an automated strategy to create poisoned classifiers in our adversary model. Previously, this work has been carried out with the human-in-the-loop method. According to our findings, smoothed adversarial instances of robust poisoned classifiers potentially revealed patterns that were related to backdoors. Our backdoor attack technique created new alternate triggers using these backdoor patterns, and it was discovered that they functioned analogously or even better than the original backdoor.

Our paper highlights the vulnerability of the poisoned classifier. Overall, our findings revealed that, contrary to what prior work has implied, a hidden backdoor is not required to modify poisoned classifiers, emphasizing the true vulnerability of poisoned classifiers in realistic circumstances. Secondly, the human-in-the-loop approach was eliminated, and the automatic method of extracting alternate triggers with a higher success rate than the original backdoor triggers was successfully implemented. Compared to the original backdoor triggers, these automated alternate triggers were more successful in highlighting the susceptibility of the poisoned classifier to an adversary with no need to access the original trigger. We believe our promising results can facilitate future research to analyze and develop more robust defense models that can be deployed in critical real-world applications in the presence of adversaries.

Author Contributions: Conceptualization, S.A. (Saqib Ali) and S.A. (Sana Ashraf); methodology, S.A. (Saqib Ali), S.A. (Sana Ashraf), M.S.Y., S.R. and G.W.; software, G.W., S.A. (Saqib Ali) and S.R.; validation, S.A. (Saqib Ali), S.A. (Sana Ashraf) and M.S.Y.; formal analysis, S.A. (Saqib Ali), S.A. (Sana Ashraf) and S.R.; investigation, S.A. (Saqib Ali), S.A. (Sana Ashraf), M.S.Y. and S.R.; resources, S.A. (Saqib Ali) and G.W.; data curation, S.A. (Saqib Ali), S.A. (Sana Ashraf) and G.W.; writing—original draft preparation, S.A. (Saqib Ali), S.A. (Sana Ashraf) and M.S.Y.; writing—review and editing, S.A. (Saqib Ali) , M.S.Y. and S.R.; visualization, M.S.Y. and S.R.; supervision, S.A. (Saqib Ali) and G.W.; project administration, S.A. (Saqib Ali) and G.W.; funding acquisition, S.A. (Saqib Ali) and G.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Key Research and Development Program of China (No. 2020YFB1005804); in part by the National Natural Science Foundation of China under grant 61632009; in part by the High-Level Talents Program of Higher Education in Guangdong Province under grant 2016ZJ01; and in part by the NCBC-PAAL, UAF, Faisalabad.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gong, X.; Wang, Z.; Chen, Y.; Xue, M.; Wang, Q.; Shen, C. Kaleidoscope: Physical Backdoor Attacks against Deep Neural Networks with RGB Filters. *IEEE Trans. Dependable Secur. Comput.* **2023**, 1–12. [CrossRef]
2. Kaviani, S.; Shamshiri, S.; Sohn, I. A defense method against backdoor attacks on neural networks. *Expert Syst. Appl.* **2023**, *213*, 118990. [CrossRef]
3. Saha, A.; Subramanya, A.; Pirsiavash, H. Hidden trigger backdoor attacks. In Proceedings of the AAAI 2020—34th Conference on Artificial Intelligence, Hilton, New York Midtown, New York, NY, USA, 7–12 February 2020; pp. 11957–11965. [CrossRef]
4. Wang, S.; Nepal, S.; Rudolph, C.; Grobler, M.; Chen, S.; Chen, T. Backdoor Attacks Against Transfer Learning With Pre-Trained Deep Learning Models. *IEEE Trans Serv Comput* **2022**, *15*, 1526–1539. [CrossRef]
5. Turner, A.; Tsipras, D.; Madry, A. Label-Consistent Backdoor Attacks. 2019, pp. 1–24. Available online: http://arxiv.org/abs/1912.02771 (accessed on 9 January 2023).
6. Gu, T.; Dolan-Gavitt, B.; Garg, S. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. 2017. Available online: http://arxiv.org/abs/1708.06733 (accessed on 12 December 2022).
7. Sun, M.; Agarwal, S.; Kolter, J.Z. Poisoned Classifiers Are Not Only Backdoored, They Are Fundamentally Broken. 2020, pp. 1–21. Available online: http://arxiv.org/abs/2010.09080 (accessed on 11 January 2023).
8. Chen, X.; Liu, C.; Li, B.; Lu, K.; Song, D. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. 2017. Available online: http://arxiv.org/abs/1712.05526 (accessed on 17 January 2023).
9. Gu, T.; Liu, K.; Dolan-Gavitt, B.; Garg, S. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access* **2019**, *7*, 47230–47243. [CrossRef]
10. Soremekun, E.; Udeshi, S.; Chattopadhyay, S. Towards Backdoor Attacks and Defense in Robust Machine Learning Models. *Comput. Secur.* **2023**, *127*, 103101. [CrossRef]
11. Mello, F.L. De A Survey on Machine Learning Adversarial Attacks. *J. Inf. Secur. Cryptogr.* **2020**, *7*, 1–7. [CrossRef]
12. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings, Banff, AB, Canada, 14–16 April 2014; pp. 1–10.

13. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015; pp. 1–11.
14. Hu, S.; Zhang, Y.; Liu, X.; Zhang, L.Y.; Li, M.; Jin, H. AdvHash: Set-to-set Targeted Attack on Deep Hashing with One Single Adversarial Patch. In Proceedings of the MM 2021—Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021. [CrossRef]
15. Chiang, P.H.; Chan, C.S.; Wu, S.H. Adversarial Pixel Masking: A Defense against Physical Attacks for Pre-trained Object Detectors. In Proceedings of the MM 2021—Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021. [CrossRef]
16. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, USA, 3–18 May 2017. [CrossRef]
17. Fredrikson, M.; Jha, S.; Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the ACM Conference on Computer and Communications Security, Enver, Colorado, CO, USA, 12–16 October 2015; pp. 1322–1333. [CrossRef]
18. Zhao, S.; Ma, X.; Zheng, X.; Bailey, J.; Chen, J.; Jiang, Y.G. Clean-Label Backdoor Attacks on Video Recognition Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, Washington, DC, USA, 14–19 June 2020; pp. 14431–14440. [CrossRef]
19. Zhang, Y.; Albarghouthi, A.; D'Antoni, L. PECAN: A Deterministic Certified Defense Against Backdoor Attacks. 2023. Available online: http://arxiv.org/abs/2301.11824 (accessed on 23 February 2023).
20. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings, Vancouver, BC, Canada, 30 April—3 May 2018; pp. 1–28.
21. Sun, M.; Agarwal, S.; Kolter, J.Z. Oisoned classifiers are not only backdoored, they are fundamentally broken. *arXiv* **2019**, arXiv:2010.09080v2.
22. Zhang, Q.; Wencong, M.A.; Wang, Y.; Zhang, Y.; Shi, Z.; Yuanzhang, L.I. Backdoor Attacks on Image Classification Models in Deep Neural Networks. *Chin. J. Electron.* **2022**, *31*, 199–212. [CrossRef]
23. Kwon, H.; Kim, Y. BlindNet backdoor: Attack on deep neural network using blind watermark. *Multimed. Tools Appl.* **2022**, *81*, 6217–6234. [CrossRef]
24. Qi, X.; Xie, T.; Pan, R.; Zhu, J.; Yang, Y.; Bu, K. Towards Practical Deployment-Stage Backdoor Attack on Deep Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13337–13347. [CrossRef]
25. Nguyen, A.; Tran, A. WaNet—Imperceptible Warping-based Backdoor Attack. 2021, pp. 1–16. Available online: http://arxiv.org/abs/2102.10369 (accessed on 15 December 2022).
26. Salman, H.; Sun, M.; Yang, G.; Kapoor, A.; Kolter, J.Z. Denoised smoothing: A provable defense for pretrained classifiers. In Proceedings of the Advances in Neural Information Processing Systems, 2020, NeurIPS 2020, Virtual, 6–12 December 2020; pp. 1–13.
27. Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; Zhao, B.Y. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–22 May 2019; pp. 707–723. [CrossRef]
28. Guo, W.; Wang, L.; Xing, X.; Du, M.; Song, D. TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems. 2019. Available online: http://arxiv.org/abs/1908.01763 (accessed on 25 January 2023).
29. Koh, P.W.; Steinhardt, J.; Liang, P. Stronger data poisoning attacks break data sanitization defenses. *Mach. Learn.* **2022**, *111*, 1–44. [CrossRef]
30. Abdul Quadir, M.; Jaiswal, D.; Daftari, J.; Haneef, S.; Iwendi, C.; Jain, S.K. Efficient Dynamic Phishing Safeguard System Using Neural Boost Phishing Protection. *Electronics* **2022**, *11*, 3133. [CrossRef]
31. Abdul Quadir, M.; Christy Jackson, J.; Prassanna, J.; Sathyarajasekaran, K.; Kumar, K.; Sabireen, H.; Ubarhande, S.; Vijaya Kumar, V. An efficient algorithm to detect DDoS amplification attacks. *J. Intell. Fuzzy Syst.* **2020**, *39*, 8565–8572. [CrossRef]
32. Cohen, J.; Rosenfeld, E.; Kolter, J.Z. Certified adversarial robustness via randomized smoothing. In Proceedings of the 36th International Conference on Machine Learning (ICML 2019), Long Beach, California, USA, 9–15 June 2019; pp. 2323–2356.
33. Alkinani, M.H.; El-Sakka, M.R. Patch-based models and algorithms for image denoising: A comparative review between patch-based images denoising methods for additive noise reduction. *Eurasip J. Image Video Process.* **2017**, *1*, 1–27. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.