



Fully Hidden Dynamic Trigger Backdoor Attacks

Shintaro Narisada ^a, Seira Hidano and Kazuhide Fukushima ^b

KDDI Research, Inc., Fujimino, Japan

Keywords: Backdoor Attacks, Poisoning Attacks, Invisible Trigger, Evasion Attacks, Generalization.

Abstract: Indistinguishable adversarial attacks have been demonstrated with the sophistication of adversarial machine learning for neural networks. One example of such advanced algorithms is the backdoor attack with hidden triggers proposed by Saha et al. While Saha's backdoor attack can produce invisible and dynamic triggers during the training phase without mislabeling, visible patch images are appended during the inference phase. A natural question is whether there exists a clean label backdoor attack whose trigger is dynamic and invisible at all times. In this study, we answer this question by adapting Saha's backdoor attack to the trigger generation algorithm and by presenting a completely invisible backdoor attack with dynamic triggers and correct labels. Experimental results show that our proposed algorithm outperforms Saha's backdoor attacks in terms of both indistinguishability and the attack success rate. In addition, we realize that our backdoor attack is a generalization of adversarial examples since our algorithm also works by using poisoning data only during the inference phase. We also describe a concrete algorithm for reconstructing adversarial examples as clean-label backdoor attacks. Several defensive experiments are conducted for both algorithms. This paper discovers the close relationship between hidden trigger backdoor attacks and adversarial examples.


1 INTRODUCTION


There has been a remarkable amount of research on neural networks in recent years, especially in the field of computer vision, as demonstrated in applications such as biometrics (Sundararajan and Woodard, 2018; Minaee et al., 2019), health care (Esteva et al., 2019; Shamshirband et al., 2021), and self-driving technology (Huang and Chen, 2020; Grigorescu et al., 2020). The vulnerability of deep models to images or videos intentionally manipulated by adversaries has been revealed. In the image classification domain, perceptual indistinguishability between adversarial images and nonadversarial images makes it more difficult for both humans and machine learning systems to discriminate poisoned images. *Adversarial examples* (Goodfellow et al., 2014; Szegedy et al., 2013; Madry et al., 2017) are a type of attack mechanism that indistinguishably manipulates images to cause misclassification of the prediction model during the inference phase.

Backdoor attacks (Gu et al., 2017; Chen et al., 2017), in which an attacker injects an adversarial image with a small perturbation (referred to as a trigger) into the training dataset to misclassify images with

similar trigger patterns during the inference, are also being discussed in terms of trigger invisibility. There are two types of triggers: *static* (or data-independent) triggers such as watermarking and steganography (Li et al., 2021a; Ning et al., 2021), and *dynamic* triggers that are optimized for each image (Liao et al., 2018; Li et al., 2021b). The former is easier to realize in terms of trigger feasibility, while the latter is superior in terms of attack performance and detection resistance due to the use of data-oriented triggers. *Consistency of labels* is one of the major factors to be considered when inserting triggered images into the training dataset. Although flipping the label of the triggered image to the target class improves the attack performance (Xiao et al., 2012), it compromises the confidentiality of the attack, since the label-flipped image is significantly different from the images in the target class. Several papers have suggested backdoor attacks with consistent labels (Saha et al., 2020; Ning et al., 2021).

Few studies have addressed all of these issues. Saha et al. (Saha et al., 2020) proposed a clean-label backdoor attack with a dynamic trigger, in which triggers are invisible in the training phase. During the inference, the authors used triggers that are conventional visible static patch images since they focused

^a  <https://orcid.org/0000-0002-9399-9778>

^b  <https://orcid.org/0000-0003-2571-0116>

on the capability of the attack. However, such visible and nondata-oriented triggers that do not exploit the features of each image are not only easily detected by defenders but also fail to provide sufficient attack performance.

Contributions

In this work, we propose a clean-label backdoor attack where the trigger is *invisible* and *dynamic* in both the training phase and inference phase. Our algorithm generates dynamic invisible triggers for both the adversary's base class (*source*) for the inference and target class (*target*) for the training dataset by utilizing the feature values of images from both classes. The proposed attack framework is illustrated in Figure 1. The attack framework can also be specified as an adversarial example if the poisoning ratio is 0 since the framework constructs a triggered source image that satisfies the following conditions: (1) the perturbation is invisible. (2) Only the perturbed image has directivity to the target class. We emphasize that normal poisoning attacks or backdoor attacks with static triggers such as patch images cannot be applied as adversarial examples. Using the attack framework, we generalize adversarial examples as clean-label, invisible, dynamic backdoor attacks. Note that only known methods for generalization of adversarial examples are poisoning attacks that employ label flipping as described in (Fowl et al., 2021; Pang et al., 2020).

In our experiments, our algorithm achieves a 100% attack success rate with a 6.25% poisoning rate and a 99% attack success rate even without poisoning on the standard ImageNet dataset. The success probability is higher than that of the 55.1% of the invisible backdoor attack by Saha (Saha et al., 2020) under the same conditions. For indistinguishability, we verify that our backdoor algorithm can generate an invisible trigger when ϵ is small using the learned perceptual image patch similarity (LPIPS) indicator. Compared with adversarial example-based backdoor attacks, the proposed method provides higher *poisoning performance*, i.e., enhancement of attack performance with increased poisoning rate. We adopt neural cleanse (Wang et al., 2019) and input transformation defense (Guo et al., 2017) into our algorithms, which are well-known countermeasures against backdoor attacks and adversarial examples, respectively. These defensive methods are found to be ineffective or effective only under limited conditions.

In summary, our main contributions are highlighted as follows:

- This paper proposes the first framework to implement fully hidden dynamic trigger backdoor attacks with clean labels. Our attacks render the

triggers indistinguishable in the inference stage and in the training stage.

- We formulate the creation of invisible adaptive triggers for the training and inference inputs as a sequence of optimization problems. This formulation allows both triggered source and target images to function as backdoor poisons and affect the classification results.
- Based on the proposed attack framework, standard adversarial examples are generalized for the first time as clean-label backdoor attacks.
- We demonstrate through extensive experiments that our attacks achieve higher attack success rates, poisoning performance, and invisibility than the existing hidden trigger backdoor attack.
- We show that our attacks are robust against existing countermeasures both for backdoor attacks and adversarial examples.

We hope that our results will stimulate interest in research on adaptive attacks and defenses to fully hidden trigger backdoor attacks.

2 RELATED WORK

Adversarial examples started with the analysis of the distinct differences between the input space of deep models and the feature spaces of deep models (Szegedy et al., 2013; Biggio et al., 2013). Goodfellow proposed a well-known algorithm referred to as the fast gradient sign method (FGSM) (Goodfellow et al., 2014), followed by many studies on topics such as its iterative variant (Kurakin et al., 2016) and the projected gradient descent (PGD) introduced in (Madry et al., 2017). These methods have two things in common: perturbations are optimized for each image, and invisibility is guaranteed by the threshold parameter ϵ .

Backdoor attacks (Gu et al., 2017; Chen et al., 2017) are special cases of poisoning attacks (Biggio et al., 2012; Muñoz González et al., 2017; Koh et al., 2018). Poisoning aims to misclassify all clean images of the source class selected by the attacker, and a backdoor attack misclassifies only triggered source images. Consistency of labels for poisoning images in poisoning or backdoor attacks is one of the primary issues for maintaining attack feasibility. In poisoning attacks, Shafahi (Shafahi et al., 2018) suggests an algorithm to generate clean-label poison data by solving an optimization problem in the input and feature space between the target and the source images. Clean label backdoor attacks are realized by adopting

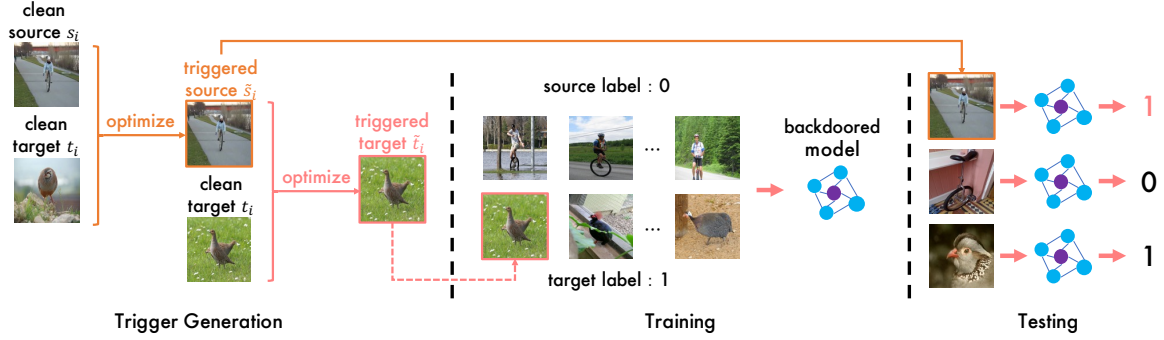


Figure 1: Our framework of fully hidden dynamic trigger backdoor attacks. The framework consists of two optimization phases: (1) Invisible triggered source image \tilde{s}_i generation from a clean source class (unicycle) image s_i and a target class (partridge) image t_i , which can be directly utilized as backdoor images in the inference phase. (2) Invisible triggered target image \tilde{t}_i generation from triggered source \tilde{s}_i and clean target image t_i . \tilde{t}_i is injected into the training data of the target class with the correct label. In the inference phase, the backdoored model mispredicts the triggered source image \tilde{s}_i generated in (1) as the target class while correctly classifying the clean test images. Note that if the attacker does not have WRITE permission to the training data, step (2) can be skipped and the attack can be executed as an adversarial example without poisoning.

adversarial perturbations using generative adversarial networks (GANs) (Turner et al., 2019) or inserting a visible backdoor signal into an image (Barni et al., 2019).

Several papers focus on the invisibility of triggers to more subtly blend perturbed images with other clean images and with adversarial examples. Liao (Liao et al., 2018) and Li (Li et al., 2021b) proposed invisible and dynamic trigger generation algorithms but with incorrect labels. Saha’s method prepares invisible and label consistent triggers during training. However, a visible static patch image is applied in the inference phase (Saha et al., 2020). Steganography patterns can also be applied to generate invisible triggers as presented in (Li et al., 2021a). Ning (Ning et al., 2021) showed clean-label backdoor attacks with steganography-style invisible triggers generated from a specific image using an autoencoder. Although all these methods satisfy criteria such as invisibility, label consistency, and trigger adaptability, no algorithm that fulfills all of them has been proposed.

3 METHODOLOGY

3.1 Threat Model

3.1.1 Backdoor Attack

We will use the same threat model for backdoor attacks defined in (Gu et al., 2017; Saha et al., 2020). The involved parties are an *attacker* and a *victim*. We assume that an attacker can *read* a part of the training or surrogate dataset and is capable of writing manipu-

lated data into the training dataset. However, he cannot remove or overwrite the existing training dataset. One such application is an online machine learning service such as MLaaS, which allows users to submit individual data from their clients and a deep model on a central server trains or classifies it. We refer to the manipulated data injected into the training dataset as *poison*. In backdoor attacks, a poison is constructed by appending a small data referred to as a *trigger* to clean data. To generate a trigger using the gradient information, the attacker also knows the structure of the victim’s model (white box attacks) or a surrogate model to classify a similar dataset (black box attacks), but he cannot manipulate the victim’s model itself. Then, the generated trigger is appended to the attacker’s training data often accompanied by label flipping. Note that allowing the attacker to manipulate labels reduces the feasibility of the attack since a machine learning system often employs autoannotation for labeling. The victim is assumed to fine-tune the pretrained model with the contaminated dataset. In the inference phase, the model misclassifies triggered data into the wrong class. Typically, attackers aim to misclassify data belonging to a specific class (*source*) into the targeted class (*target*). To avoid detection of model collapse, data that are not triggered should be correctly classified.

3.1.2 Adversarial Example

We also describe the standard threat model for (targeted) adversarial examples mentioned in (Szegedy et al., 2013; Goodfellow et al., 2014) compared with backdoor attacks. In the adversarial examples, the attacker has less capability than in backdoor attacks, that is, he can access the model’s gradient or a part

of the (surrogate) training dataset to generate triggers, but he cannot insert the triggered data into the victim's training dataset. The objective of the attacker is to cause triggered source data to be misclassified into the target class in the inference phase. In contrast to the backdoor attacks that normally use visible triggers such as patch images, triggers are usually indistinguishable in the adversarial examples so that defenders do not notice the attack.

3.2 Poisoned Image Generation

The central part of the backdoor attacks and adversarial example is how to create the poisoned data utilizing trigger generation and label flipping. In the following section, we briefly describe the poison generation algorithms for backdoor attacks and adversarial examples in the image domain as the most typical cases.

3.2.1 Clean Label Invisible Trigger

In (Saha et al., 2020), the authors proposed a backdoor attack in which the trigger is invisible during training by adopting the technique proposed in (Shafahi et al., 2018) to generate poison using the data correlation between the source class and the target class. First, patched source images are created by Equation 1:

$$\tilde{s}_i \leftarrow s_i \odot (1 - m) + p \odot m, \quad (1)$$

where s_i is a source image and \tilde{s}_i is the triggered source image. p is a small static trigger such as a patch image. \odot denotes an elementwise product. m is a bitmask such that it is 1 at the position where the trigger is located and 0 otherwise. Then, a clean-label triggered target image with the correct label (\tilde{t}_i, y_i) is constructed from the initial target image t_i and patched source image \tilde{s}_i by:

$$\begin{aligned} \tilde{t}_i &\leftarrow \arg \min_z \|f(z) - f(\tilde{s}_i)\| \\ s.t. \quad &\|z - t_i\|_\infty \leq \epsilon, \end{aligned} \quad (2)$$

where z is a triggered target image, f is the projection to the feature space for the deep model and ϵ is a threshold. If ϵ is small, \tilde{t}_i can be regarded as an invisible triggered target image. To make \tilde{t}_i more general with respect to trigger location and variation of source images, Equation 2 is reformulated in the style of a coordinate descent algorithm. In the inference phase, the model mispredicts patched source image \tilde{s}_i as the target class because the poison (\tilde{t}_i, y_i) was learned to move the decision boundary near the patch images in the target class direction. Note that since the trigger of \tilde{s}_i is still visible, trigger detection algorithms may notice the attack.

3.2.2 Invisible Trigger Generated by Adversarial Examples

In our paper, we consider an adversarial example as a special case of backdoor attacks, namely, it is the case where the number of poisons to be added to the training data is 0. During inference, a poisoned image is generated so that the trigger is hidden. A notable adversarial example algorithm is PGD (Madry et al., 2017). In PGD, \tilde{s}_i is iteratively constructed by:

$$\begin{aligned} \tilde{s}_i^{(0)} &\leftarrow s_i \\ \tilde{s}_i^{(j+1)} &\leftarrow \tilde{s}_i^{(j)} - a \operatorname{sgn} \nabla_{\tilde{s}_i^{(j)}} L(F(\tilde{s}_i^{(j)}), y_i) \\ s.t. \quad &\|\tilde{s}_i^{(j)} - s_i\|_\infty \leq \epsilon, \end{aligned} \quad (3)$$

where $a \leq \epsilon$ is the step size. Triggers are invisible if we set ϵ to a small value. In (Fowl et al., 2021; Pang et al., 2020), the authors describe that one can extend adversarial examples to poisoning attacks or backdoor attacks by adopting the label-flipping technique. However, to the best of our knowledge, there is no known algorithm that can realize a backdoor attack from adversarial examples without label-flipping.

3.3 Fully Hidden Trigger Backdoor Attack

The primary objective of this study is to develop a clean label backdoor attack whose trigger is dynamic and invisible during *both* training and inference. Specifically, we seek to replace the visible patched source image \tilde{s}_i generated by Equation 1 with an invisible trigger. One solution to this is to utilize Equation 2 not only for generating the poisoned target but also for the poisoned source by replacing all terms regarding s_i and t_i :

$$\begin{aligned} \tilde{s}_i &\leftarrow \arg \min_z \|f(z) - f(t_i)\| \\ s.t. \quad &\|z - s_i\|_\infty \leq \epsilon. \end{aligned} \quad (4)$$

For the second term of $\arg \min$, we just replace \tilde{t}_i with the normal target t_i to avoid a circular reference between Equation 4 and Equation 2. Then, an invisibly triggered source image \tilde{s}_i is generated from one clean pair (s_i, t_i) . We intend to create triggered source \tilde{s}_i that resembles a source image but is near to the target in the feature space. Afterward, poisoned target \tilde{t}_i with a hidden trigger is generated by Equation 2 using t_i and \tilde{s}_i obtained from Equation 4. A poison (\tilde{t}_i, y_i) is inserted into the training dataset. Note that Equation 2 refers to the triggered source \tilde{s}_i generated by Equation 4, not the clean source s_i , so it can generate poisoning target \tilde{t}_i focused on \tilde{s}_i . The order in

Algorithm 1: Generate-Invisible-Poison.

Input: K data a_i from class y_a , b_i from class y_b

Output: K poisoning data \hat{a}_i for class y_a

- 1 Initialize $\hat{a}_i \leftarrow a_i$ for $1 \leq i \leq K$
- 2 **for** $t \leftarrow 1, \dots, T$ **do**
- 3 $(\hat{a}_i, b_{m[i]}) \leftarrow \text{Find-Closest-Pair}(\hat{a}, b)$
- 4 $L \leftarrow \sum_{i=1}^K \|f(\hat{a}_i) - f(b_{m[i]})\|_2^2$
- 5 $\hat{a}_i \leftarrow \hat{a}_i - \eta \nabla_{\hat{a}_i} L$
- 6 $\hat{a}_i \leftarrow \min(\max(\hat{a}_i, a_i - \epsilon), a_i + \epsilon)$
- 7 **return** \hat{a}_i

which \tilde{s}_i and \tilde{t}_i are created is also important. If \tilde{t}_i is created from clean images s_i and t_i using Equation 2, then \tilde{s}_i will be generated from clean source s_i and triggered target \tilde{t}_i to maintain the connection with \tilde{t}_i . This approach degrades the attack performance of \tilde{s}_i since \tilde{t}_i is generated to be close to clean source s_i in the feature space.

By combining Equation 2 and Equation 4 in the appropriate order, we can approach opening a backdoor on the feature space from both sides of the source and target classes. Thus, our algorithm is assumed to be more effective than (Saha et al., 2020) in terms of not only invisibility but also attack performance. Note that a pair of triggered source images (\tilde{s}_i, y_s) is not suitable for poison since it causes adversarial training (reduces the attack success rate) by forcing the model to correctly learn target-like images \tilde{s}_i as the source class. \tilde{s}_i is used only in the inference phase instead of a visible patch image.

3.3.1 Increasing Attack Performance and Versatility

An arbitrary source image with the hidden trigger should succeed in the attack. Optimizing Equation 2 and 4 from single image pairs to K pairs provides higher attack performance and versatility for source image variations. A concrete instantiation of the optimization is provided in Algorithm 1 as a consequence of the generalization of the algorithm proposed in (Saha et al., 2020). Algorithm 1 outputs K triggered images \hat{a}_i in class y_a from K clean images a_i in class y_a and K (triggered) images b_i in class y_b . The Find-Closest-Pair function in Line 3 permutes K pairs $(\hat{a}_i, b_{m[i]})$ so that the sum of the L2 distance between $f(\hat{a}_i)$ and $f(b_i)$ is approximately minimized by solving a greedy selection algorithm to obtain *good* pairs that are close in the feature space. m is a mapping for indices i in range $1 \leq i \leq K$.

In our backdoor algorithm, we call Algorithm 1 by inputting K_1 clean source and target images. The out-

put consists of K_1 triggered source images \hat{s}_i . Then, K_2 clean target images for class y_a and K_2 triggered source images are input to Algorithm 1 to gain K_2 triggered target images \hat{t}_i . We consider the balanced case $K_1 = K_2$ in our experiments. $P \leq K_2$ triggered target images are added to the training dataset. If an attacker does not own the write permission to the training dataset, he can alternatively run an adversarial example variant of the attack by setting $P = 0$ (skipping the generation of the poisoning target). In default, K_1 triggered source images are available for the test images.

3.3.2 Attack Feasibility

The attacker’s capabilities in our proposed algorithm are equivalent to those of (Saha et al., 2020); i.e., a part of the training dataset and the gradient of the model. During the inference phase, one might suggest that our algorithm would need more powerful capabilities than the existing method since our triggered source images are created using model gradients, unlike static patch images. However, recall that our triggered source images can be generated *in advance* when generating triggered target images, at which point the gradient of the model is still required in the existing method. It is a realistic assumption, such as in MLaaS, that an attacker prepares backdoor instances to be submitted to the service in advance and stores them in his devices until the inference. In addition, our algorithm assumes the same threat model as adversarial examples when $P = 0$, since both algorithms access the gradient of the model to generate perturbed source images in the inference phase.

3.4 Generalize Adversarial Examples as Clean-Label Backdoor Attacks

Our second objective is to generalize targeted adversarial examples to realize backdoor attacks without label flipping. In previous research, adversarial examples are employed as poisoning attacks or backdoor attacks utilizing label flipping (Fowl et al., 2021; Pang et al., 2020). The key observation is that our hidden trigger backdoor attack functions as an adversarial example when the amount of poison is $P = 0$. Conversely, we hypothesized that adversarial examples can be converted to backdoor attacks by taking an approach similar to that presented in Section 3.3. Instead of flipping the label for the triggered source image as discussed in (Fowl et al., 2021; Pang et al., 2020), we utilize a source-target swapped variant for the trigger generation algorithm to generate clean la-

bel poison. For PGD, a poison is generated by:

$$\begin{aligned} \tilde{t}_i^{(0)} &\leftarrow t_i \\ \tilde{t}_i^{(j+1)} &\leftarrow \tilde{t}_i^{(j)} - \epsilon \text{sgn} \nabla_{\tilde{t}_i^{(j)}} L(F(\tilde{t}_i^{(j)}), y_s) \\ s.t. \quad &\|\tilde{t}_i^{(j)} - t_i\|_\infty \leq \epsilon. \end{aligned} \quad (5)$$

Poisoning data to be added to the training data is formed as (\tilde{t}_i, y_t) . Unlike naively inserting (\tilde{t}_i, y_s) using Equation 3 as poison data, appending (\tilde{t}_i, y_t) expands a specific region of a target class near \tilde{t}_i in the feature space. Equation 5 helps \tilde{t}_i to become more easily classified as a target class.

Following the attack framework shown in Figure 1, one can perform clean-label backdoor attacks generalized from adversarial examples by replacing Equation 2 and 4 with Equation 5 and 3, respectively. Whole P poisoning data are prepared by solving Equation 5 for P different clean target images P times. During inference, standard adversarial example algorithms such as Equation 3 are used to generate the triggered source images.

4 EXPERIMENTS

4.1 Setup

4.1.1 Dataset and Metrics

The ImageNet and CIFAR-10 datasets are selected for our experiments. For both datasets, the number of images in the training dataset is 800 for each class, a total of 1600 for binary classification. The dataset available to the attacker consists of 200 source images and 200 target images in default settings that are independent of the victim’s training dataset. Then, the attacker generates 200 triggered source and target images. The top 100 target poisons with the lowest loss are inserted into the training data. The poisoning ratio is $100/1600 = 6.25\%$ in binary classification tasks.

We measure the *clean accuracy* (abbreviated as *clean acc.*) for 50 clean test images for each class not included in both the victim’s training dataset and attacker’s training dataset, the *poisoned source accuracy* (\hat{s}_i acc.) for 50 triggered source images, and the *clean source accuracy* (s_i acc.) for 50 clean source images. Note that higher clean source accuracy and lower poisoned source accuracy (referred to as the *attack success rate*) are desirable for the attacker.

Our experiments are performed over 10 random pair datasets as shown in Table 1 for ImageNet and Table 2 for CIFAR-10 with varying source and target categories. We measure the average as the results.

Table 1: Random pairs for ImageNet.

ID	Source	Target
1	slot	Australian terrier
2	lighter	bee
3	theater curtain	plunger
4	unicycle	partridge
5	mountain bike	Ipod
6	coffeepot	Scottish deerhound
7	can opener	sulphur-crested cockatoo
8	hotdog	toyshop
9	electric locomotive	tiger beetle
10	wing	goblet

Table 2: Random pairs for CIFAR-10.

ID	Source	Target
1	bird	dog
2	dog	ship
3	frog	plane
4	plane	truck
5	cat	truck
6	deer	ship
7	bird	frog
8	bird	deer
9	car	frog
10	car	dog

Hereafter, we consider the binary classification case for the ImageNet dataset unless otherwise specified.

4.1.2 Model and Learning Parameters

AlexNet (Krizhevsky et al., 2012) serves as the prediction model, and the *fc7* layer serves as the feature space f to generate the poison. We fine-tune the pre-trained AlexNet with the victim’s training dataset and the attacker’s dataset. The learning rate in fine-tuning is set to 0.001, the batch size is set to 256, and the number of epochs is set to 30.

4.1.3 Backdoor Generation Algorithms

In our experiments, we consider the algorithm proposed in (Saha et al., 2020) as a representative of existing invisible backdoor algorithms (abbreviated as *existing*). We do not compare other methods using label flipping or static triggers since they assume a stronger attacker’s capabilities or different situations. For the proposed algorithms, we use the fully invisible backdoor attack described in Section 3.3 (*invisible*) and a backdoor attack based on an adversarial example using Equation 3 and 5 in Section 3.4 (*adversarial*). The learning rate η in Algorithm 1 is 0.01. The batch size for *existing* and *invisible* is set to 100, and that for *adversarial* is set to 256. The number of iterations for *existing* and *invisible* is 5000, and that for *adversarial* is 10. We set the threshold $\epsilon = 4$ and $P = 100$ unless

otherwise specified.

4.2 Binary Classification for ImageNet

First, we compare `existing` and `invisible` where $\epsilon = 16$ using the random pairs in Table 1, which is the same condition employed in (Saha et al., 2020). The results are presented in Table 3. Both methods achieve high classification accuracy for the clean validation data for both the clean model and poisoned model. For the poisoned model, our `invisible` algorithm accomplishes an attack success rate of 100.0%, which is significantly higher than the success rate of 55.1% achieved by the existing method. The main reason for these results is that Equation 1 uses a simple patch image as a trigger that does not use any gradient information, while Equation 4 includes the model’s gradient in the invisible trigger. This finding is evidenced by the notion that the `invisible` also succeeds in the attack against the *clean* model, i.e., it works as an adversarial example. However, the poisoned model correctly classifies clean nontriggered source images.

In addition, we conduct several ablation studies on ϵ and P for our `invisible` algorithm to validate the effects of invisibility and the amount of poison on the attack performance. The results are shown in Table 4 and Table 5. Table 4 shows that ϵ does not have any impact on the prediction of clean images. The attack success rate exceeds 99% when $\epsilon \geq 4$. When $\epsilon = 2$, the attack success rate drops to approximately 80%; when $\epsilon = 1$, the attack success rate drops to approximately 10%. Therefore, there is a trade-off between the trigger invisibility and the attack performance.

For the poisoning rate, we varied the number of poisoned images between $P = 0, 100, 200$, and 400 with fixed $\epsilon = 1$ or $\epsilon = 2$. Note that we conduct adversarial examples rather than backdoor attacks when $P = 0$ since the attacker manipulates only testing images. We skip the case where $\epsilon \geq 4$ since the attack success rate is already saturated at $P = 0$. From Table 5, it can be inferred that our `invisible` attack certainly has features in common with target poisoning attacks since the attack success rate increases as P increases, while clean accuracy remains steady.

4.3 Trigger Invisibility

We compare the triggered source between the `existing` algorithm and the `invisible` algorithm. Figure 2 represents the triggered images. In the existing methods, the trigger for source images is a random patch image. Notably, the trigger for `existing` is visually recognized and can be identified by ex-

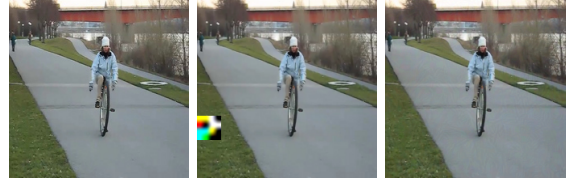


Figure 2: (Left) Original image for the unicycle class. (Center) Patched source image from Equation 1 with a 30×30 random patch image. (Right) Hidden triggered source image from Equation 4, where $\epsilon = 2$.

isting trigger detection algorithms. In contrast, by choosing a smaller ϵ , the proposed method is able to generate invisible triggers for the source image that is physically indistinguishable from the original image. For the target image side, both algorithms can already generate invisible triggers, so `invisible` can indistinguishably hide triggers in all training and inference phases.

In addition, we conduct quantitative experiments on trigger visibility using learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018). Smaller LPIPS values indicate closer perceptual proximity. In our experiments, we measure LPIPS between the original source images and the triggered source images for three backdoor algorithms varying ϵ in $\{1, 2, 4, 8, 16\}$. For `existing`, we paste a randomly created patched image at a random location. We measure the average LPIPS values for all random pairs (Table 1). The results are shown in Table 6. Contrary to what is actually perceived, the LPIPS value of `existing` is considerably smaller than that of `invisible` and `adversarial` for $\epsilon \geq 2$. A possible reason for this finding is that triggers generated by `invisible` and `adversarial` are spread over the entire image, while the triggered source image matches the original image, with the exception of the patch generated by the `existing` method. Additionally, the LPIPS values for $\epsilon = 1$ and 2 are certainly comparable to the existing method. Even if $\epsilon = 2$, `invisible` maintains an attack success rate of approximately 80% (Table 4), it accomplishes both invisibility and attack performance.

4.4 CIFAR-10 Dataset

To verify that the proposed algorithm is effective independent of the dataset and image resolution, we conduct an experiment for a binary classifier on the CIFAR-10 dataset. The results are listed in Table 7. From the results, we achieve an attack success rate of 90.1% with less than 1% degradation of the accuracy, which is higher than 81.3% of the success rate of the existing method presented in (Saha et al., 2020) with $\epsilon = 16$.

Table 3: Binary classification results for the ImageNet dataset for (Saha et al., 2020) and *invisible* under the same settings, where $\epsilon = 16$ and $P = 100$ (poisoning ratio is 6.25%).

	(Saha et al., 2020)		<i>invisible</i>	
	Clean Model	Poisoned Model	Clean Model	Poisoned Model
Clean Acc.	0.992 ± 0.012	0.981 ± 0.031	0.992 ± 0.010	0.991 ± 0.021
Poisoned Source Acc.	0.978 ± 0.078	0.449 ± 0.311	0.002 ± 0.018	0.000 ± 0.000
Clean Source Acc.	0.992 ± 0.012	0.970 ± 0.050	0.992 ± 0.012	0.992 ± 0.032

Table 4: Results of the *invisible* algorithm with varying ϵ among 1, 2, 4, 8, 16.

ϵ	model	clean acc.	\tilde{s}_i acc.	s_i acc.
1	clean	0.991 ± 0.01	0.900 ± 0.10	0.992 ± 0.03
	poison	0.991 ± 0.01	0.887 ± 0.11	0.994 ± 0.01
2	clean	0.992 ± 0.01	0.362 ± 0.38	0.994 ± 0.01
	poison	0.989 ± 0.02	0.206 ± 0.21	0.990 ± 0.03
4	clean	0.992 ± 0.01	0.008 ± 0.03	0.994 ± 0.01
	poison	0.993 ± 0.01	0.008 ± 0.03	0.994 ± 0.01
8	clean	0.992 ± 0.01	0.000 ± 0.00	0.994 ± 0.01
	poison	0.992 ± 0.01	0.000 ± 0.00	0.992 ± 0.03
16	clean	0.992 ± 0.01	0.002 ± 0.02	0.992 ± 0.01
	poison	0.991 ± 0.02	0.000 ± 0.00	0.992 ± 0.03

Table 5: Results of the *invisible* algorithm with varying P among 0, 100, 200, 400 for the poisoned model.

ϵ	P	clean acc.	\tilde{s}_i acc.	s_i acc.
1	0	0.991 ± 0.01	0.900 ± 0.10	0.992 ± 0.03
	100	0.991 ± 0.01	0.887 ± 0.11	0.994 ± 0.01
	200	0.991 ± 0.02	0.868 ± 0.15	0.988 ± 0.03
	400	0.994 ± 0.01	0.822 ± 0.28	0.990 ± 0.03
2	0	0.992 ± 0.01	0.362 ± 0.38	0.994 ± 0.01
	100	0.989 ± 0.02	0.206 ± 0.21	0.990 ± 0.03
	200	0.988 ± 0.03	0.160 ± 0.24	0.984 ± 0.04
	400	0.991 ± 0.02	0.098 ± 0.10	0.986 ± 0.03

4.5 Multiclass Settings

We conduct backdoor attacks toward multiclass classification tasks with 20 classes using the ImageNet dataset. We create a multiclass dataset by merging all the pairs shown in Table 1. It is generally assumed that the attacker’s party has images of multiple source classes and that any image belonging to one of those classes will cause a misprediction simply by showing the model its secret trigger. In the experiments, we vary the target class from ID1 to ID10, but any source class listed in Table 1 is the attacker’s source class. The number of training datasets is 20,918 (approximately $1050 \text{ images} \times 20 \text{ classes}$). The attacker should pair a triggered target with each source class to allow the model to learn the trigger distribution for each source class. To accomplish the objective, the attacker generates 400 triggered source and target images for each source class using Algorithm 1. Then, the top 40 triggered target images for each class are selected and inserted into the training data (the poisoning ratio is $40 \times 10 / 20,918 = 1.91\%$). Fifty im-

ages for each source class are selected for the test dataset.

The results are shown in Table 8. Even in the multiclass settings, the average attack success rate still exceeds 90%, which is much higher than the existing result of 30.7%, as shown in (Saha et al., 2020). The mean value of the clean source accuracy for the poisoned model does not decrease from that of the clean model. Individually analyzing each class, the lowest attack success rate for the clean model is 62% when the source class is "unicycle" and the target class is "plunger". In such a case, the attack success rate improves to 100% after poisoning. Therefore, our *invisible* poisoning can boost the attack performance without sacrificing the clean accuracy.

4.6 Backdoor Attacks Using Adversarial Examples

Here, we examine how *invisible* algorithm based on backdoor attacks and adversarial algorithm based on adversarial examples transition with respect to the attack performance when varying the poisoning ratio P . Figure 3 shows the results for varying $P = 0, 100, 200$ and 400. A hidden source trigger using an adversarial example (Equation 3) achieves a higher attack success rate when $P = 0$ (in the case of adversarial examples). In contrast, Algorithm 1 makes backdoor attacks more powerful than adversarial examples when $P \geq 100$ (in case of backdoor attacks). The results imply that both triggers function as a backdoor attack and an adversarial example, but *invisible* is more effective when behaving as a backdoor attack and *adversarial* is more effective when behaving as an adversarial example.

4.7 Resiliency to Defenses

Since our backdoor attacks also have features common to adversarial examples, it is necessary to verify the countermeasures against both backdoor attacks and adversarial examples. In our paper, we use two fundamental defensive methods referred to as neural cleanse (Wang et al., 2019) for backdoor attacks and input-transformations (Guo et al., 2017) for adversarial examples.

Table 6: Averaged LPIPS values for three backdoor algorithms (existing, invisible and adversarial). We choose AlexNet for the LPIPS prediction model varying ϵ in the range $\{1, 2, 4, 8, 16\}$. Smaller values indicate closer perceptual proximity to the original image.












(Saha et al., 2020) (patch size: 30×30)	invisible					adversarial				
	$\epsilon = 16$	$\epsilon = 8$	$\epsilon = 4$	$\epsilon = 2$	$\epsilon = 1$	$\epsilon = 16$	$\epsilon = 8$	$\epsilon = 4$	$\epsilon = 2$	$\epsilon = 1$
0.0827	0.5595	0.4170	0.2460	0.1122	0.0434	0.2966	0.2781	0.2007	0.0921	0.0299
										

Table 7: Clean accuracy, poisoned source accuracy and clean source accuracy for our invisible backdoor attack on the CIFAR-10 dataset.

	Clean Model	Poisoned Model
Clean Acc.	0.951 ± 0.064	0.946 ± 0.054
Poisoned Source Acc.	0.121 ± 0.139	0.082 ± 0.166
Clean Source Acc.	0.937 ± 0.103	0.930 ± 0.090

Table 8: Results for the invisible algorithm where $\epsilon = 4$ and $P = 400$ (1.91%) for multiclass classification.

	Clean Model	Poisoned Model
Clean Acc.	0.922 ± 0.120	0.942 ± 0.140
Poisoned Source Acc.	0.130 ± 0.079	0.016 ± 0.007
Clean Source Acc.	0.893 ± 0.010	0.927 ± 0.014

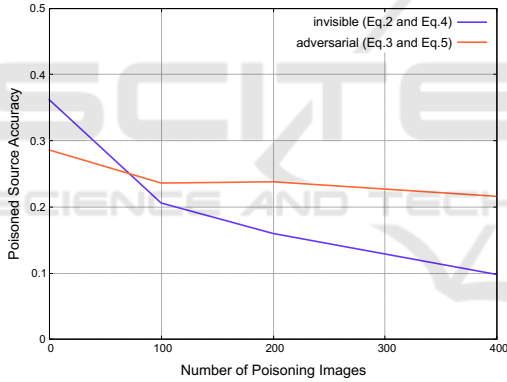


Figure 3: Poisoned source accuracy of invisible and adversarial. The invisibility is fixed at $\epsilon = 2$.

4.7.1 Defense Against Backdoor Attacks

Neural cleanse (NC) is a novel model inspection mechanism against backdoor attacks. NC outputs L_1 -norm, which is referred to as the minimum perturbation cost (MPC), for each class. If a class is backdoored, the MPC to change the prediction of all the inputs in the class to the target class is abnormally small. In our experiment, we measured the MPC for all random pairs in Table 1 for both directions (target \rightarrow source and source \rightarrow target) after the model is backdoored using three backdoor attacks. The results are shown in Figure 4. For existing, NC works well since the MPC to change source images to the target class is substantially smaller than

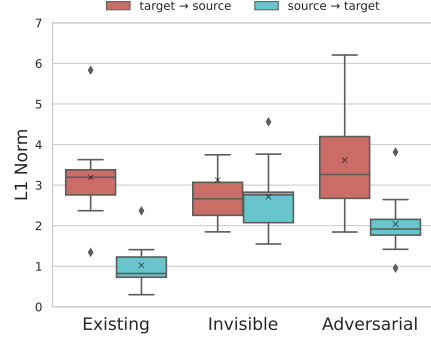


Figure 4: Result of neural cleanse defense against three backdoor algorithms where $\epsilon = 4$ and $P = 100$.

the other class. However, there is a slight difference in the MPC of invisible between the source class and the target class. Thus, it is hard to distinguish if the model is backdoored using the MPC as a threshold detection. Notably, both existing and invisible generate invisible triggered targeted images, while only invisible is robust to NC. The MPC for adversarial between two classes is not as separated as existing.

4.7.2 Defense Against Adversarial Examples

One of the basic defensive measures for adversarial examples is to sanitize the poison by perturbing the input image before inputting the model. (Guo et al., 2017) applies several image transformations such as image quilting and JPEG compression to remove the poisonous trigger from the input. In the experiment, the total variance minimization (TVM), which has been shown to be effective for PGD (Guo et al., 2017), is applied to all images before inputting the model in both the training phase and inference phase. We assume that an attacker is unaware of the countermeasure, i.e., he uses original untransformed images to generate the poison. The results are shown in Figure 5. TVM defense reduces the attack success rate for all methods, but there is still a possibility of a successful attack with a probability of more than 40% for invisible at $P \geq 200$ and adversarial maintains a 70% attack success rate at $P \geq 100$. Thus, our pro-

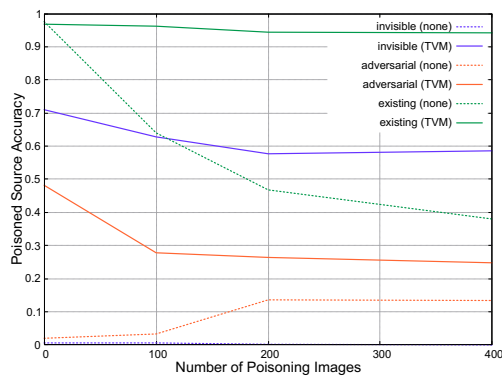


Figure 5: Poisoned source accuracy for the poisoned model with/without TVM defense against attackers unaware of defenses.

posed attacks, which generate adaptive triggers, are more robust than the existing method against the input transformation-based defense. Additionally, if the attacker is aware of the defense and uses transformed images to generate poison, the attack is more likely to succeed.

5 CONCLUSION

We propose a fully hidden dynamic trigger backdoor attack where the trigger is invisible during both testing and training. Our algorithm dynamically generates invisible triggers without flipping labels or changing the victim's model. Experimental results verified the superiority of the proposed algorithms in terms of invisibility and attack success rate. To prevent fully hidden dynamic trigger backdoor attacks in practice, adaptive defensive methods are essential.

REFERENCES

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

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases*, pages 387–402, Berlin, Heidelberg, Springer Berlin Heidelberg.

Biggio, B., Nelson, B., and Laskov, P. (2012). Poisoning attacks against support vector machines.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. (2017). Targeted backdoor attacks on deep learning systems using data poisoning.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., and Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29.

Fowl, L., Goldblum, M., Chiang, P.-y., Geiping, J., Czaja, W., and Goldstein, T. (2021). Adversarial examples make strong poisons.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386.

Gu, T., Dolan-Gavitt, B., and Garg, S. (2017). Badnets: Identifying vulnerabilities in the machine learning model supply chain.

Guo, C., Rana, M., Cisse, M., and van der Maaten, L. (2017). Countering adversarial images using input transformations.

Huang, Y. and Chen, Y. (2020). Autonomous driving with deep learning: A survey of state-of-art technologies.

Koh, P. W., Steinhardt, J., and Liang, P. (2018). Stronger data poisoning attacks break data sanitization defenses.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial examples in the physical world.

Li, S., Xue, M., Zhao, B. Z. H., Zhu, H., and Zhang, X. (2021a). Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2088–2105.

Li, Y., Li, Y., Wu, B., Li, L., He, R., and Lyu, S. (2021b). Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16463–16472.

Liao, C., Zhong, H., Squicciarini, A. C., Zhu, S., and Miller, D. J. (2018). Backdoor embedding in convolutional neural network models via invisible perturbation. *CoRR*, abs/1808.10307.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks.

Minaee, S., Abdolrashidi, A., Su, H., Bannamoun, M., and Zhang, D. (2019). Biometrics recognition using deep learning: A survey.

Muñoz González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., and Roli, F. (2017). *Towards Poisoning of Deep Learning Algorithms with Back-Gradient Optimization*, page 27–38. Association for Computing Machinery, New York, NY, USA.

Ning, R., Li, J., Xin, C., and Wu, H. (2021). Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10.

- Pang, R., Shen, H., Zhang, X., Ji, S., Vorobeychik, Y., Luo, X., Liu, A., and Wang, T. (2020). A tale of evil twins: Adversarial inputs versus poisoned models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS '20*, page 85–99, New York, NY, USA. Association for Computing Machinery.
- Saha, A., Subramanya, A., and Pirsiavash, H. (2020). Hidden trigger backdoor attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11957–11965.
- Shafahi, A., Huang, W. R., Najibi, M., Suci, O., Studer, C., Dumitras, T., and Goldstein, T. (2018). Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Shamshirband, S., Fathi, M., Dehzangi, A., Chronopoulos, A. T., and Alinejad-Rokny, H. (2021). A review on deep learning approaches in healthcare systems: Taxonomies, challenges, and open issues. *Journal of Biomedical Informatics*, 113:103627.
- Sundararajan, K. and Woodard, D. L. (2018). Deep learning for biometrics: A survey. *ACM Comput. Surv.*, 51(3).
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks.
- Turner, A., Tsipras, D., and Madry, A. (2019). Clean-label backdoor attacks.
- Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., and Zhao, B. Y. (2019). Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723.
- Xiao, H., Xiao, H., and Eckert, C. (2012). Adversarial label flips attack on support vector machines. In *ECAI 2012*, pages 870–875. IOS Press.
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.