ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240650

Revocable Backdoor for Deep Model Trading

Yiran Xu[a,1], **Nan Zhong**[a,1], **Zhenxing Qian**[a,*] **and Xinpeng Zhang**[a,**]

[a]Fudan University

Abstract. Deep models are being applied in numerous fields and have become a new important digital product. Meanwhile, previous studies have shown that deep models are vulnerable to backdoor attacks, in which compromised models return attacker-desired results when a trigger appears. Backdoor attacks severely break the trustworthiness of deep models. In this paper, we turn this weakness of deep models into a strength, and propose a novel revocable backdoor and deep model trading scenario. Specifically, we aim to compromise deep models without degrading their performance, meanwhile, we can easily detoxify poisoned models without re-training the models. We design specific mask matrices to manage the internal feature maps of the models. These mask matrices can be used to deactivate the backdoors. The revocable backdoor can be adopted in the deep model trading scenario. Sellers train models with revocable backdoors as a trial version. Buyers pay a deposit to sellers and obtain a trial version of the deep model. If buyers are satisfied with the trial version, they pay a final payment to sellers and sellers send mask matrices to buyers to withdraw revocable backdoors. We demonstrate the feasibility and robustness of our revocable backdoor by various datasets and network architectures.

1 Introduction

Deep learning has achieved impressive performance in a wide range of fields [14, 29, 25]. However, the security vulnerability of deep models, such as the notorious backdoor attack, hinders the deployment of deep models in some risk-sensitive domains [2, 9]. Most attackers [10, 20] implant backdoors into a clean model during the training phase by data poisoning. The compromised models behave normally during the evaluation phase, whereas they return attacker-desired results when the predefined trigger appears. Considering the stealthiness and hazard of backdoor attacks, it has attracted a lot of attention in the machine learning security community [27, 35, 15, 41].

Although backdoor attack is seen as a security threat or vulnerability for deep models in most studies, some researchers adopt backdoor attack to conduct some positive purposes [1, 30, 23]. There are various positive usages of the backdoor like dataset or model copyright protection [1], artificial intelligence interpretability [23], adversarial example defence [30], etc. Adi et al. [1] adopt the backdoor attack to implement model watermarking for model copyright protection. They use significantly abnormal outputs of models triggered by poisonous inputs as the watermarking signal to verify the ownership of the model. A similar idea [21] is also implemented to achieve dataset ownership protection. Shan et al. [30] utilize the backdoor as a honeypot for deep models to resist adversarial example attack. When an attacker constructs an adversarial example for the backdoored model, the distribution of the interior feature map of the adversarial example is close to that of poisonous ones. Therefore, defenders identify adversarial example input by inspecting the distribution of the interior feature map.

In this paper, we first propose a novel concept of the revocable backdoor and explore its promising application which is deep model trading. The revocable backdoor denotes that attackers implant backdoors into clean deep models while attackers can easily withdraw these backdoors. In other words, attackers control poisoning and detoxification for deep models at the same time. The detoxification should be easy and efficient. Revocable backdoor still needs to satisfy the general properties of a backdoor. When inputs contain the trigger, the model with revocable backdoor produces the attacker's predefined outputs. Otherwise, the model should behave normally and produce correct output. Additionally, it should possess stealth and robustness to prevent users from detecting and removing the backdoor. The revocable backdoor can be applied to deep model trading to protect the rights and interests of both buyers and sellers.

The main contributions of this paper are three-fold:

1. We propose a promising deep model trading scenario, and hold that the revocable backdoor can address the issues haunted by buyers and sellers.
2. We propose a practical revocable backdoor, which does not require extra clean or poison data to withdraw backdoors.
3. Extensive experiments demonstrate that our proposed revocable backdoor is feasible and robust.

2 Model trading Scenario

Since the wild usage of deep learning models in real world, models become a tradable products. However, how to sale a trained model or buy a trained model still remains a question. The Uniform Commercial Code (UCC) [37] stipulates that when the goods delivered by the seller do not conform to the contract, the buyer has the significant right to "revoke acceptance". Correspondingly, when the buyer breaches the contract, such as improperly rejecting the goods, incorrectly revoking the acceptance of the goods, failing to pay the price etc., the seller has the right to undertake a series of remedial measures, including recovering the goods, reselling, and so on.

Therefore, the rights and demands of both the buyer and seller in the model transaction can be summarized as follows:

Sellers seek to profit by selling their own models. As models are considered data resources, a model "return" would carry immeasurable loss of benefits. They attempt to control costs as much as possible to increase profits.

* Corresponding Author. Email: zxqian@fudan.edu.cn
** Corresponding Author. Email: zhangxinpeng@fudan.edu.cn
[1] Equal contribution.

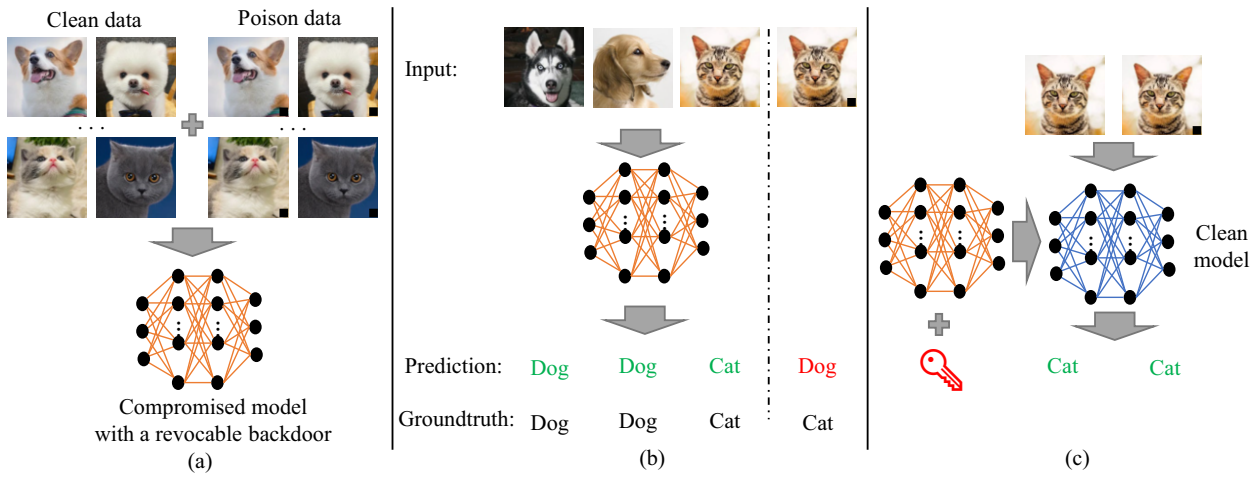

Figure 1. The illustration of the revocable backdoor for the model trading scenario. (**a**) Seller trains a model with a revocable backdoor as a trial version for buyers. The performance of the trial version over clean inputs is the same as the final model (or a clean model without backdoors). (**b**) Buyers pay a deposit and obtain a trial model. Then, buyers evaluate whether the model meets their requirements. For trial models (backdoored models), the trigger predefined by sellers in (**a**) can lead models to return wrong results. Note that we use a pure small black square to represent the trigger pattern in the figure for simplicity. The practical trigger pattern adopted in our approach is more sophisticated. (**c**) if buyers decide to pay the final payment, sellers withdraw the hidden backdoors. The deep model returns the correct result even the trigger pattern appears.

Buyers hope to purchase models that meet their needs. When they find that the purchased model does not meet their needs, they want to be able to return it and get a refund.

Trading platform provides a reliable platform for model transactions, safeguarding the reasonable rights and interests of both buyers and sellers.

However, there are still a few issues:

- **Q1**. How to allow the buyer to return and refund the purchased model when they find it does not meet their needs, ensuring their right to "revoke acceptance"?
- **Q2**. How to protect the seller's interests without being damaged when the buyer returns the model? That is, how to prevent the buyer from saving some of the data after receiving the model, and then returning and refunding it.

A possible solution to these questions could be to send a trial version at the time of delivery. The full version would then be delivered once the buyer confirms receipt. Then, the questions turn into:

- **Q3**. How to ensure that both the trial version and the full version of the model meet the buyer's requirements?
- **Q4**. How to ensure the model will not be resold by the buyer?

To solve these two questions, we propose the concept of a revocable backdoor. Fig. 1 illustrates the details of the deep model trading scenario. A buyer distributes his requirement to a seller. Then, the seller trains a model based on the buyer's requirements. The model has a revocable backdoor and can be sent to the buyer as a trail version. The buyer pays a deposit to obtain the trial model whose performance is almost the same as the clean final model over a normal task. Next, the buyer evaluates the trial model on his/her application and determines whether accepts the model and pays a final payment. If the buyer is satisfied with the trial version, they can pay the seller the final balance. After that, the seller sends the mask matrix to the buyer and the revocable backdoor can be removed by the mask matrix.

Buyers can disclose the backdoor hidden in the model publicly and make the trial model unusable. Fig. 1 illustrates the details of the deep model trading scenario. The advantages of using the revocable backdoor are as follows:

1. It allows the user to try the model while securing the rights of the seller. In the worst case, the buyer maliciously refuses to pay the final payment and deploys the trial model in his/her application. The seller can disclose the backdoor hidden in the model publicly and makes the trail model unusable. Or, if there is a third part (sale platform) involved, the seller can register the mask and sue the buyer for infringement. (Address **Q1** and **Q2**.)
2. It also ensures the relevance between the trial version and the complete version of the model, preventing the buyer from receiving a complete model that does not match the trial version after paying the balance and confirming receipt. (Address **Q3**.)
3. There is no backdoor in the complete version of the model, but the seller can still use the mask matrix to identify whether a model is the one they sold (like a serial number of a software). That is, by subtracting the mask matrix from the model, determining whether the model with the mask matrix removed contains a preset backdoor to authenticate the model's genuineness. (Address **Q4**.)

3 Related Work

3.1 Backdoor Attack

Backdoor attack is an emerging topic in the machine learning security community [22]. The goal of the backdoor attack is to compromise deep models, which return normal results for clean inputs, but return attacker-desired results when the trigger appears. Gu et al. [10] first propose the concept and specific practical backdoor attack approach named BadNets. BadNets mounts an attack through data poison during the training phase. A conspicuous colourful square is adopted as a trigger pattern. BadNets generate some poisonous images by adding the trigger pattern to some clean images and changing their label to the target (attacker-desired) label. Such poisonous images are mixed with other clean training sets. The deep model trained over a poisonous training set contains an insidious backdoor which

only activated by the trigger pattern. The implementation of the backdoor attack can be roughly grouped into two categories.

Poisoning-based backdoor mounts an attack by inserting some well-designed poisonous samples into the training set [18, 20, 28, 5, 12]. Previous researchers focus on designing stealthy and efficient trigger patterns. Although BadNets proposes a seminal approach for backdoor attacks, its trigger pattern is conspicuous. In the follow-up studies, researchers explore some invisible trigger patterns. Li et al. [18] and Li et al. [20] adopt image steganographic noise (Least Significant Bit and DNNs-based steganographic) as a trigger pattern. The visual quality of poisonous is greatly boosted by invisible trigger patterns. Furthermore, WaNet [28] adopts image affine transformation as a trigger pattern. The poisonous images can be viewed as slightly warped from clean ones. WaNet works effectively not only in the digital domain but also attacks successfully in the physical domain.

Training process controlling-based backdoor can achieve more stealthy attacks than poisoning-based ones [43, 36, 7, 8]. As a trade-off, these approaches entail controlling the entire training phase. These attacks focus on both the stealthiness of the trigger pattern and compromised models. Backdoor attacks may leave traces in compromised models. Doan et al. [7] focus on the feature map separability of compromised models and employ Wasserstein distance to minimize the feature map distance between poisonous and clean inputs. Some similar ideas are proposed in subsequent studies. Besides consideration of the feature map domain, BppAttack [36] employs contrastive supervised learning and adversarial training instead of standard cross-entropy loss in the classification task. This learning approach ensures that compromised models steadily and successfully converge.

3.2 Backdoor Defence

Considering the serious threat of backdoor attacks to machine learning security, backdoor defence, which is the countermeasure against backdoor attacks, rapidly develops in recent studies [40, 4, 26, 16]. There are two main categories in the backdoor defence, i.e., backdoor detection [33, 11], backdoor purification [38, 24, 19, 42] In terms of backdoor detection approaches, they aim to determine whether the deep model contains a backdoor. Neural Cleanse (NC) [33] is one of the well-known backdoor detection algorithms. NC observes that the backdoor of the classifier constructs a shortcut between the target label and other clean labels. Based on this observation, NC employs a abnormality detection algorithm to determine whether the model is poisoned by analyzing the size of perturbation leading all clean images to the target label.

Compared with backdoor detection, backdoor purification aims to eliminate backdoors on the premise that the performance of clean inputs does not significantly degrade. Fine-Pruning [24] adopts network pruning to erase backdoors hidden in the models. For existing neural networks, most neurons are dormant when the clean inputs come. However, these dormant neurons may be activated by the trigger pattern and lead models to return attacker-desired results. Therefore, Fine-Pruning identifies dormant neurons based on some clean inputs then deletes them and fine-tunes the model. NAD (Neural Attention Distillation) [19] is an alternative effective backdoor purification approach, which utilizes model distillation to erase backdoors. NAD fine-tunes backdoored model to generate a teacher model which is used to conduct model distillation. Many other sophisticated model purification approaches are proposed in subsequent studies.

4 Revocable Backdoor

4.1 Preliminaries

In this paper, we focus on the image classification task which is consistent with previous studies. Since the revocable backdoor is first proposed by ours, we describe more details of the revocable backdoor requirement applied in the deep model trading.

Goal The revocable backdoor aims to ensure that buyers complete the final payment when they are satisfied with the trial model. If buyers refuse to complete the final payment, sellers can disclose the hidden backdoor and corresponding trigger pattern. For example, buyers obtain a face recognition model with a revocable backdoor and deploy this model in their company. If the backdoor is disclosed, anyone with the trigger pattern can break the face recognition model, that is, the model is unusable. Note that our revocable backdoor does not aim to protect the copyright of the model, which is usually accomplished by model watermarking [1]. Conventional backdoor-based watermarking aims to verify the ownership of the model, whose goal is completely different from ours. Although backdoor-based watermarking also can make the model return incorrect results when the trigger appears, the backdoor is permanent and irrevocable. Even though buyers complete the final payment, the backdoor still exists in the model and brings security threats to the model deployment.

Effectiveness In previous backdoor attacks, attackers aim to make the compromised classifier return the target label when the trigger appears [10, 28]. However, in the context of model tradings, the goal of our revocable backdoor for model sellers is different from previous studies. Sellers can distribute the trigger pattern to make the classifier held by buyers unusable in the event that the buyer does not complete the final payment. Therefore, our revocable backdoor aims to decrease the classification accuracy as low as possible when the trigger appears. In other words, we view the attack success if the compromised classifier returns an arbitrarily wrong label.

Fidelity The classifier with a revocable backdoor can be used as a trial version of the final classifier for buyers. Therefore, the revocable backdoor aims to keep the performance of the backdoored classifier over clean inputs the same as a clean classifier. For buyers, the performance of the trial version over clean inputs is almost identical to the final version (backdoor-free). In this case, buyers can evaluate whether the classifier meets their requirements.

Revocability The crux of our revocable backdoor is that we can easily withdraw the backdoor without requiring an extra training phase or clean/poisonous data. The revocability denotes that the compromised classifier whose backdoor is withdrawn can correctly predict poisonous inputs.

Robustness In the model trading scenario, the backdoors hidden in the compromised model can and only can be withdrawn (erased) by sellers. Therefore, our attack should ensure that the backdoor is resistant to various backdoor purification defences.

4.2 Method

Fig. 2 illustrates the framework of our approach. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denotes the benign training set, where $x_i \in \mathcal{X} = \{0, 1, \ldots, 255\}^{C\times W\times H}$ is the image, $y_i \in \mathcal{Y} = \{1, \ldots, N\}$ is its label, and N is the number of classes. The trigger pattern is denoted as δ, where $\delta \in \mathcal{P} = \{-t, -t+1, \ldots, t+1, t\}^{C\times W\times H}$. The elements of the trigger pattern are between $-t$ and t. In the backdoor attack, we select a portion of clean images x, which are stamped with the trigger pattern, as poisonous ones. We denote the clean training set and poisonous training set as $\mathcal{D}_c$ and $\mathcal{D}_p$, respectively. Given a

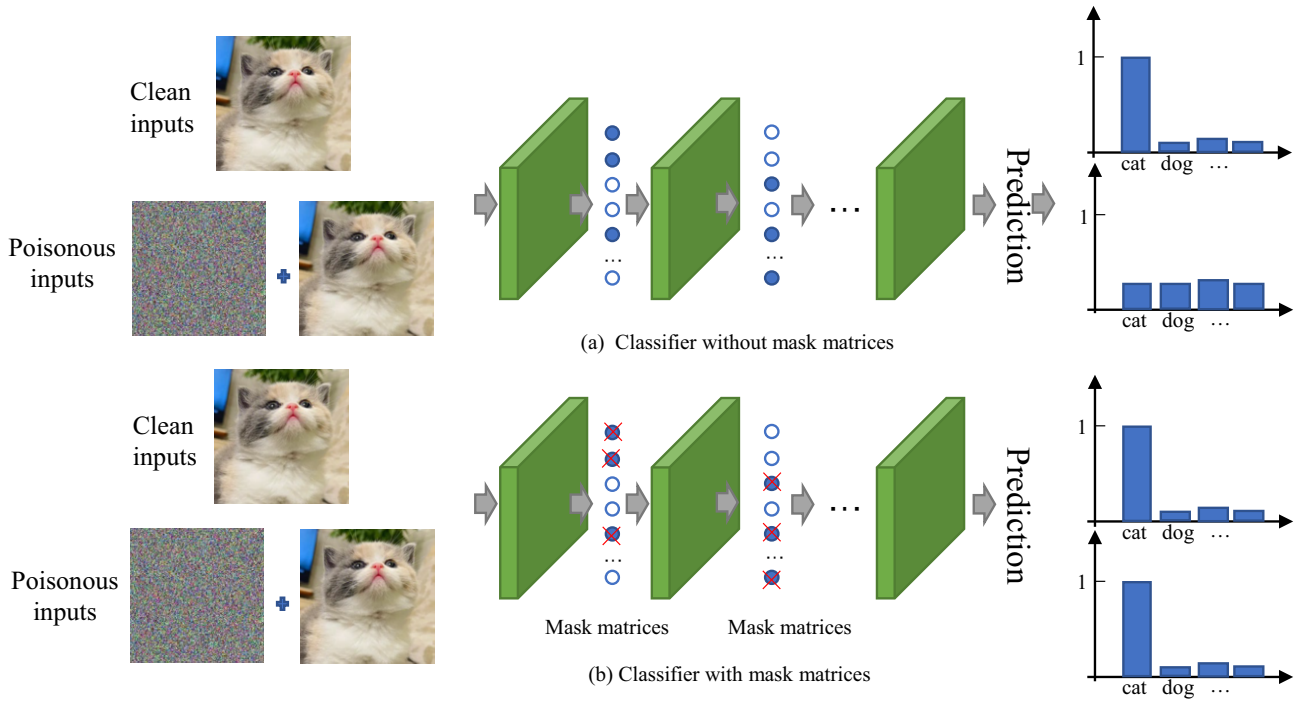

Figure 2. The framework of the implementation revocable backdoor attack. Our approach consists of two main parts. (a) is similar to common backdoor attacks. The training set including both clean and poisonous inputs is fed into the classifier, which returns correct results for clean inputs yet wrong results for poisonous ones. (b) is the crux of the revocability of our backdoor. We withdraw our backdoor by controlling the interior feature map. We utilize some trainable mask matrices to intentionally break the poisonous inference link.

deep neural network-based classifier $f_\theta(\cdot)$, we aim to minimize the following equation,

$$L_{bd} = \frac{1}{|D_c|} \sum_{(x,y)\in D_c} \ell\left(f_\theta(x), y\right) \tag{1}$$

$$-\frac{1}{|D_p|} \sum_{(x,y)\in D_p} max(\ell\left(f_\theta(x+\delta), y\right), c)\,, \tag{2}$$

where $\ell(\cdot)$ denotes cross-entropy loss widely adopted in classification tasks. We update $f_\theta(\cdot)$ and δ to minimize equation 2. The core idea of equation 2 is to achieve the effectiveness and fidelity of the revocable backdoor attack. The first term of equation 2 ensures that the performance of backdoored classifiers over clean inputs does not drop. The second term denotes maximizing the cross-entropy loss of poisonous inputs, i.e., making the classifier return wrong results when the trigger appears. However, maximizing the cross-entropy loss of poisonous inputs is much easier than minimizing the cross-entropy loss of clean inputs. In other words, classifier $f_\theta(\cdot)$ easily learns how to give wrong results for poisonous inputs but it is hard to learn how to give correct results for clean inputs. As a result, we adopt $max(\cdot)$ and hyperparameter confidence c to restrict the second term. For a N categories classification task, if the cross-entropy loss of input is larger than $-log(1/N)$, the classifier will return an incorrect result. Therefore, we set $c = -1.1 \times log(1/N)$ that is slightly larger than the minimum correct decision threshold for a N categories classifier. If we remove the restriction of $max(\cdot)$ or set an excessively large c, the classifier f_θ will only learn the second term of equation 2 and overlook the first term.

Besides L_{bd}, our approach also entails considering how to withdraw our backdoor. The modern deep neural network-based classifier $f_\theta(\cdot)$ is made up of some cascade layers. The output of $f_\theta(\cdot)$ can be expressed as,

$$f_\theta(\cdot) = f_{\mathrm{n}}\left(\cdots f_2(f_1(\cdots))\right), \tag{3}$$

where n denotes the number of layers. The output of the interior layer is named as feature map. Based on the cascade structure of $f_\theta(\cdot)$, we employ some matrices named masks to manipulate feature maps. Compromised classifiers learn two tasks, i.e., a normal task and an insidious backdoor task. These two tasks conduct different inference links. In other words, the feature maps of these two tasks are different. Therefore, we can adopt some mask matrices to break backdoors without affecting clean inputs. We denote the classifier $f_\theta(\cdot)$ with mask matrices M_k as $f_{\theta_{mask}}(\cdot)$. The formulaic expression is shown as follows,

$$f_{\theta_{mask}}(\cdot) = f_{\mathrm{n}}\left(\cdots M_2 \cdot f_2(M_1 \cdot f_1(x))\right), \tag{4}$$

where M and $\cdot$ denote trainable mask matrices and Hadamard product, respectively. The size of M_k is the same as the corresponding feature map. The initial elements of M_k are filled with 1. Moreover, we require a regularization term for M_k. The goal of regularization is used to make the elements of M_k as close to 1 as possible. The regularization for M_k is expressed as follows,

$$L_R = sum(|M_{k_{ij}} - 1|), \tag{5}$$

where $sum(\cdot)$ denotes the sum of all elements. The usage of L_R forces M_k to change the most key elements related to poisonous inputs in feature maps. L_R can effectively mitigate the impact of M_k on clean inputs. For $f_{\theta_{mask}}(\cdot)$, it should return correct results regardless of whether the trigger pattern δ appears. Therefore, the loss

function for $f_{\theta_{mask}}(\cdot)$ is expressed as follows,

$$L_{rev} = [\frac{1}{|D_c|} \sum_{(x,y)\in D_c} \ell\left(f_{\theta_{mask}}(x), y\right) \tag{6}$$

$$+\frac{1}{|D_p|} \sum_{(x,y)\in D_p} \ell\left(f_{\theta_{mask}}(x+\delta), y\right)]. \tag{7}$$

We update M_k by minimizing L_{rev} and make M_k adaptive find and break the poisonous inference link. The final cost function is expressed as follows,

$$L = L_{bd} + L_{rev} + \alpha \cdot L_R, \tag{8}$$

where α is the balance factor to control the weight of the regularization. We simultaneously update parameters of f_θ, trigger pattern δ, and mask matrices M_k to minimize L. The backdoor hidden in the well-trained f_θ can easily be withdrawn by adding M_k in the interior layers. $f_\theta(\cdot)$ and $f_{\theta_{mask}}(\cdot)$ can be used as the trial model and final model for sellers during the model trading.

4.3 Trigger Fine-tuning

In the previous subsection 4.2, we describe our approach to create a compromised classifier whose backdoors are controlled by mask matrices. The trigger pattern δ, whose elements are between $-t$ and t, is trained with the classifier at the same time. In this subsection, we delve into the design of the trigger pattern to control the balance of the trigger pattern imperceptibility and attack robustness. Hyperparameter t controls the modification magnitude of the trigger pattern. In other words, a small t ensures the imperceptibility of the trigger pattern. In this subsection, we frozen classifier f_θ and mask matrices M, i.e., $f_{\theta_{mask}}$ is also frozen, and update trigger patterns. The loss function of updating trigger δ is as follows,

$$L = L_{bd} + L_{rev} + \beta \cdot |\delta|_2, \tag{9}$$

where $|\delta|_2$ denotes the L_2-norm of the trigger δ, and β controls the balance of $|\delta|$. We control the modification magnitude of the trigger pattern δ by changing the value of the hyperparameter β and t.

4.4 Backdoor Erasing for Existing Attacks

To the best of our knowledge, we are the first to propose the concept of the revocable backdoor. For existing conventional backdoor attacks like BadNtes [10], WaNet [28], Blend [5], etc. They can not be withdrawn by attackers. In this subsection, we describe a backdoor erasing method to withdraw existing backdoors. In the model trading scenario, we suppose that buyers own some clean images and their corresponding poisonous ones. Then, they update the parameters of classifier f_θ by minimizing the following loss as,

$$L_{erasing} = [\frac{1}{|D_c|} \sum_{(x,y)\in D_c} \ell\left(f_\theta(x), y\right) \tag{10}$$

$$+\frac{1}{|D_p|} \sum_{(x,y)\in D_p} \ell\left(f_\theta(x+\delta), y\right)]. \tag{11}$$

Compared with equation 2, $L_{erasing}$ minimizes the cross-entropy loss between clean and poisonous images and their label. The number of poisonous images is critical to the performance of backdoor erasing. We employ this method as our baseline. Compared with our proposed method 4.2 and 4.3, this method requires gradient calculation and some poisonous images, which is not practical in the model trading scenario.

5 Experimental Results

5.1 Experimental Setup

Datasets and Networks In this paper, we adopt three datasets including CIFAR-10 [17], GTSRB [32], and Sub-ImageNet to evaluate the performance of our approach. These datasets are widely used in previous studies [28, 36]. The number of categories of CIFAR-10 and GTSRB is 10, and 43, respectively. Sub-ImageNet consists of 10 categories which are randomly selected from the original ImageNet dataset [6]. The image size of CIFAR-10, GTSRB, and Sub-ImageNet is 32×32, 32×32, and 224×224, respectively. In terms of network architectures, we adopt ResNet-18 [13] and VGG [31] in the experiments.

Backdoor Baselines Since we are the first to propose the concept of the revocable backdoor attack, there are no suitable revocable backdoor attacks used as a comparison. We suppose that users obtain some clean and corresponding poisonous images to finetune to a backdoored classifier to achieve revocability based on existing conventional attacks. We adopt various trigger patterns to conduct backdoor attacks. The trigger pattern includes BadNets [10], Blend [5], SIG [3], LSB [18], WaNet [28], and BppAttack [36]. The details setting of baseline triggers can be found in the supplement [39]. The compromised (backdoored) classifier is generated by updating equation 2 for baselines. The poisoning rate is fixed as $r = 0.1$. In the backdoor erasing phase, we suppose that users obtain $o = 5\%$ clean data of the training set and corresponding poisonous ones. Then, we finetune the backdoored baseline classifiers with equation 11.

Implement Details Our approach consists of two steps, training revocable backdoored classifier and finetuning trigger pattern. The training epoch and batch size are 180, and 256, respectively. We adopt SGD (Stochastic Gradient Descent) optimizer for updating the parameters of the classifier. The initial learning rate of SGD is 0.01. For trainable mask matrices and trigger pattern δ, we adopt Adam optimizer with a 0.001 initial learning rate. The poisoning rate is 0.5 for the first 80 epochs. The poisoning rate times by 0.5 in intervals of 10 epochs when the iteration epoch exceeds 80. Hyperparameter α is fixed as 10, 10, and 100 for CIFAR-10, GTSRB, and Sub-ImageNet, respectively. In the second phase, we fixed the poisoning rate as 0.5. The hyperparameter β is 0.01 for all datasets. The hyperparameter t controlling the range of our trigger pattern is 10 in both phases.

5.2 Attack Effectiveness

We adopt accuracy to measure the performance of the classifier. In this part, we evaluate the Effectiveness, Fidelity and Revocability of our approach. The results of our approach and baselines are shown in Table 1. We also train a clean classifier whose training setup is the same as compromised ones as a reference. We aim to significantly degrade the classifier accuracy when the trigger appears while keeping the accuracy for clean inputs unchanged. In terms of revocability, the goal of mask matrices employed in our attack is to withdraw the backdoor. From Table 1, our approach achieves satisfactory results in the aspect of attack effectiveness, fidelity, and revocability. The accuracy for clean inputs is almost unchanged compared with the clean classifier in most cases. The accuracy for poisonous inputs is decreased to near even less than random guesses, meanwhile, it recovers to standard performance (like a clean classifier) when we insert mask matrices to withdraw the backdoor. For baselines, their performance is significantly inferior to ours. In previous common backdoor studies [10, 5, 28, 36], their attack setting is that a compromised classifier returns the target label when the trigger appears. This attack

Dataset		Effectiveness		Revocability	
Acc_{Clean}	Attack	Acc_{BD}-C (↑)	Acc_{BD}-P (↓)	Acc_{BD}-C (↑)	Acc_{BD}-P (↑)
CIFAR-10 85.17	BadNets	82.99	10.04	79.90	79.68
	Blend	83.45	10.03	78.49	67.51
	SIG	83.73	10.00	78.57	77.57
	LSB	77.58	77.54	75.28	75.12
	WaNet	79.30	31.90	77.77	77.34
	BppAttack	77.85	77.32	74.65	74.71
	Ours	**85.88**	**0.57**	**85.90**	**84.77**
GTSRB 98.38	BadNets	97.43	4.99	96.79	96.78
	Blend	97.10	0.15	94.06	93.48
	SIG	97.38	**0.02**	95.62	95.15
	LSB	97.87	4.08	96.77	96.77
	WaNet	95.83	6.12	95.78	96.27
	BppAttack	95.70	95.24	96.71	96.68
	Ours	**97.69**	0.12	**97.95**	**97.66**
Sub-ImageNet 73.91	BadNets	63.26	63.29	64.74	64.60
	Blend	70.52	9.05	64.50	61.07
	SIG	67.87	**8.92**	67.53	66.39
	LSB	62.25	62.35	63.26	63.36
	WaNet	64.10	63.29	65.01	64.91
	BppAttack	63.90	63.73	64.67	64.67
	Ours	**74.06**	10.43	**74.33**	**71.33**

Table 1. The experimental comparison (ResNet-18) between baselines and ours. Acc_{Clean}, Acc_{BD}-C, and Acc_{BD}-P denote the accuracy of the clean classifier, the accuracy of the compromised classifier for clean inputs, and the accuracy of the compromised classifier for poisonous inputs, respectively. Effectiveness and revocability denote the compromised classifier and compromised classifier with mask matrices, respectively.

setting is much easier than our setting, in which the compromised classifier does not need to understand the semantic information of inputs and only builds a strong relation between the trigger and target label. However, the compromised need to understand the semantic information of inputs and then return wrong results when the trigger appears in our attack setting. Therefore, some invisible triggers (like LSB and BppAttack) whose perturbations are small cannot successfully compromise the classifier. The experiments of VGG network can be found in Table 2 of the supplementary material [39].

Apart from quantitative results, we also visualize our trigger pattern in Fig. 3. Our approach achieves satisfactory visual results and users cannot perceive trigger patterns. Based on Fig. 3 and Table 1, we find that only triggers (like BadNets) with large modification (visible) can attack successfully among baselines. Our approach achieve both trigger invisibility and attack effectiveness at the same time.

5.3 Attack Robustness

We recall our revocable backdoor scenario, i.e., deep model trading scenario. For sellers, the goal of the revocable backdoor is to make the model unusable by distributing the trigger pattern to the public if buyers do not pay the final payment. Sellers can send the trial model to buyers and let them know the existence of the hidden backdoor. Therefore, we do not consider backdoor detection defence in the robustness evaluation. We focus on evaluating the resistance of our backdoor against backdoor purification defences [19, 24]. In other words, we evaluate whether buyers can adopt backdoor purification to erase the backdoor without the permission of sellers.

The easiest backdoor purification method, which may be adopted by buyers, is fine-tuning. Buyers collect some clean images whose distribution is similar to the training set to fine-tune the trial classifier containing our revocable backdoor. To simulate fine-tuning, we adopt SGD optimizer with 0.01 learning rate kept the same as the training phase to fine-tune the compromised classifier with 5% clean images from the training set. We fine-tune the classifiers for 50 epochs. The experimental results have been shown in Table 2. Fine-tuning cannot remove our backdoor in all cases.

An alternative well-known backdoor is Fine-pruning [24], which combines network pruning and fine-tuning. As aforementioned related work part, most neurons are dormant for clean inputs. These dormant neurons almost do not affect the final performance of clean inputs. However, they may be activated by poisonous inputs. Therefore, Fine-pruning prunes dormant neurons with some clean inputs and fine-tunes the pruned classifier at the same time. The specific defence setup is as follows. We pruning the last layer of the classifier which is widely adopted in previous studies. The classifier is fine-tuned in the interval of 10 neurons. The optimizer is SGD with a 0.01 learning rate. We set the maximum tolerable drop of the clean task accuracy as less than 1%. Table 2 shows the results of Fine-pruning. It also cannot remove our backdoor. The details of the accuracy (both clean and poisonous inputs) with regard to the number of pruned neurons can be found in the supplementary material [39].

Besides fine-tuning and fine-pruning, we also evaluate the robustness of our approach by NAD [19]. NAD first trains a teacher model. Then, NAD adopts the teacher model as a guide to conduct attention distillation. The experimental results of NAD are shown in Table 2. NAD is ineffective in our approach. Although NAD can erase the backdoor in some cases (GTSRB dataset), the accuracy of clean inputs severely degrades compared with the clean model.

5.4 Ablation Studies

In our approach, we adopt two steps to generate the trigger pattern. The second part, which fine-tunes the trigger, aims to control the perturbation of trigger pattern. Table 3 gives the quantitative comparison between the trigger with or without fine-tuning. The PSNR (poisonous image quality) and SSIM [34] is significantly improved benefit from the trigger fine-tuning.

More ablation studies over confidence c, regularization L_R and the uniqueness of the mask matrices can be found in the supplement [39].

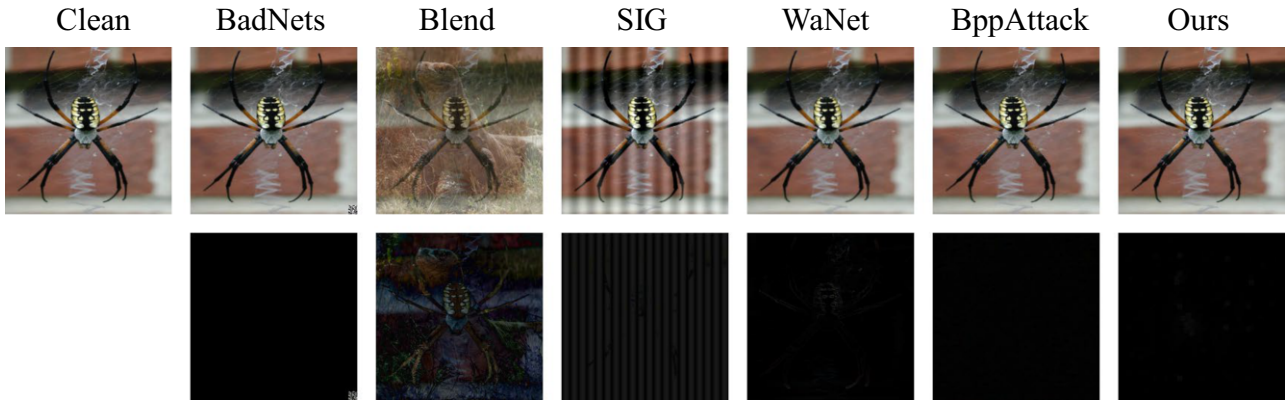

Figure 3. The visualization results of the trigger (Sub-ImgeNet). The first line denotes clean and poisonous images. The second line denotes the residual between clean and poisonous images. The visualization results of CIFAR-10 and GTSRB can be found in Figure 1 of the supplementary material [39].

	No defence		NAD		Fine-tuning		Fine-pruning	
Dataset	Acc_{BD}-C	Acc_{BD}-P	Acc_{BD}-C	Acc_{BD}-P	Acc_{BD}-C	Acc_{BD}-P	Acc_{BD}-C	Acc_{BD}-P
CIFAR-10	85.88	0.57	86.16	3.64	86.05	0.53	84.88	6.92
GTSRB	97.69	0.12	75.25	73.65	98.56	0.03	97.19	0.91
SubImageNet	74.06	10.43	74.39	10.83	74.09	10.67	73.12	10.54

Table 2. The experimental results of NAD defence, fine-tuning, and fine-pruning.

	w/o TF		TF	
Dataset	PSNR	SSIM	PSNR	SSIM
CIFAR-10	34.15	0.96	35.63	0.97
GTSRB	32.33	0.87	36.38	0.94
Sub-ImageNet	32.27	0.73	44.45	0.95

Table 3. The comparison of image quality between original and fine-tuned trigger. TF denotes trigger fine-tuning.

6 Conclusions

In this paper, we first propose the concept of the revocable backdoor attack and a novel application, that is, deep model trading. Revocable backdoor aims to create a trial model whose performance is similar to the clean/final model for buyers to evaluate its performance. Meanwhile, sellers can easily withdraw the backdoor hidden in the trial model when they obtain the final payment. The revocable backdoor attack is achieved by controlling the interior feature maps of models. We train models and mask matrices at the same time. The mask matrices are critical to turning the backdoored (trial) model into a clean (final) model. Extensive experiments demonstrate that our approach is feasible and robust.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants U20B2051, 62072114, U20A20178, U22B2047.

References

[1] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1615–1631, 2018.

[2] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, et al. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 2021.

[3] M. Barni, K. Kallas, and B. Tondi. 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. IEEE, 2019.

[4] T. Chen, Z. Zhang, Y. Zhang, S. Chang, S. Liu, and Z. Wang. Quarantine: Sparsity can uncover the trojan attack trigger for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 598–609, 2022.

[5] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] K. Doan, Y. Lao, and P. Li. Backdoor attack with imperceptible input and latent modification. *Advances in Neural Information Processing Systems*, 34:18944–18957, 2021.

[8] K. Doan, Y. Lao, W. Zhao, and P. Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11966–11976, 2021.

[9] H. Du, H. Shi, D. Zeng, X.-P. Zhang, and T. Mei. The elements of end-to-end deep face recognition: A survey of recent advances. *ACM Computing Surveys (CSUR)*, 54(10s):1–42, 2022.

[10] T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[11] J. Guo, A. Li, and C. Liu. Aeva: Black-box backdoor detection using adversarial extreme value analysis. In *International Conference on Learning Representations*, 2021.

[12] Y. Guo, N. Zhong, Z. Qian, and X. Zhang. Physical invisible backdoor based on camera imaging. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7817–7825, 2023.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988, 2017.

[15] S. Hong, N. Carlini, and A. Kurakin. Handcrafted backdoors in deep

neural networks. *Advances in Neural Information Processing Systems*, 35:8068–8080, 2022.

[16] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren. Backdoor defense via decoupling the training process. In *International Conference on Learning Representations, 2022*.

[17] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[18] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2088–2105, 2020.

[19] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *International Conference on Learning Representations*, 2020.

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

[21] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S.-T. Xia, and B. Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. *Advances in Neural Information Processing Systems*, 35: 13238–13250, 2022.

[22] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[23] Y.-S. Lin, W.-C. Lee, and Z. B. Celik. What do you see? evaluation of explainable artificial intelligence (xai) interpretability through neural backdoors. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1027–1035, 2021.

[24] K. Liu, B. Dolan-Gavitt, and S. Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018.

[25] P. Liu, X. Qiu, and X. Huang. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2873–2879, 2016.

[26] Y. Liu, G. Shen, G. Tao, Z. Wang, S. Ma, and X. Zhang. Complex backdoor detection by symmetric feature differencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15003–15013, 2022.

[27] T. A. Nguyen and A. Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

[28] T. A. Nguyen and A. T. Tran. Wanet-imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021.

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[30] S. Shan, E. Wenger, B. Wang, B. Li, H. Zheng, and B. Y. Zhao. Gotta catch'em all: Using honeypots to catch adversarial attacks on neural networks. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 67–83, 2020.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[32] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[33] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[35] Z. Wang, H. Ding, J. Zhai, and S. Ma. Training with more confidence: Mitigating injected and natural backdoors during training. *Advances in Neural Information Processing Systems*, 35:36396–36410, 2022.

[36] Z. Wang, J. Zhai, and S. Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15074–15084, 2022.

[37] J. J. White, R. S. Summers, D. D. Barnhizer, W. Barnes, and F. G. Snyder. Uniform commercial code. 2022.

[38] D. Wu and Y. Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34: 16913–16925, 2021.

[39] Y. Xu, N. Zhong, Z. Qian, and X. Zhang. Revocable backdoor for deep model trading. *arXiv preprint arXiv:2408.00255*, 2024. Full version of this paper.

[40] Y. Zeng, S. Chen, W. Park, Z. Mao, M. Jin, and R. Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*, 2022.

[41] Z. Zhang, L. Lyu, W. Wang, L. Sun, and X. Sun. How to inject backdoors with better consistency: Logit anchoring on clean data. In *International Conference on Learning Representations*, 2022.

[42] R. Zheng, R. Tang, J. Li, and L. Liu. Data-free backdoor removal based on channel lipschitzness. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 175–191. Springer, 2022.

[43] N. Zhong, Z. Qian, and X. Zhang. Imperceptible backdoor attack: From input space to feature representation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 1736–1742. ijcai.org, 2022. doi: 10.24963/ijcai.2022/242. URL https://doi.org/10.24963/ijcai.2022/242.