On Model Outsourcing Adaptive Attacks to Deep Learning Backdoor Defenses

Huaibing Peng, Huming Qiu, Hua Ma, Shuo Wang, Anmin Fu, Said F. Al-Sarawi, *Senior Member, IEEE*, Derek Abbott, *Fellow, IEEE*, and Yansong Gao, *Senior Member, IEEE*

***Abstract*— Deep learning models with backdoors act maliciously when triggered but seem normal otherwise. This risk, often increased by model outsourcing, challenges their secure use. Although countermeasures exist, their defense against adaptive attacks is under-examined, possibly leading to security misjudgments. This study is the first intricate examination illustrating the difficulty of detecting backdoors in outsourced models, especially when attackers adjust their strategies, even if their capabilities are significantly limited. It is relatively straightforward for attackers to circumvent detection by trivially violating its threat model (e.g., using advanced backdoor types or trigger designs not covered by the detection). However, this research highlights that various leading detection defenses can simultaneously be evaded using simple adaptive strategies, even under their defined threat models and with limited adversary capabilities (e.g., using easily detectable triggers while maintaining a high attack success rate). To be more specific, this study introduces a novel methodology that employs trigger specificity enhancement and training regulation in a symbiotic manner. This approach allows us to evade multiple backdoor detection defenses simultaneously, including Neural Cleanse (Oakland 19'), ABS (CCS 19'), and MNTD (Oakland 21'). These were the detection tools selected for the Evasive Trojans Track of the 2022 NeurIPS Trojan Detection Challenge. Even when applied in conjunction with these defenses under stringent conditions, such as a high attack success rate (> 97%) and the restricted use of the simplest trigger (small white square), our straightforward method garnered the second prize in NeurIPS Trojan Detection Challenge. Notably, for the first time, our adaptive attack successfully evaded other recent state-of-the-art defenses, including FeatureRE (NeurIPS 22') and Beatrix (NDSS 23'). This study suggests that existing model outsourcing backdoor defenses remain vulnerable to adaptive attacks, and thus, the use of third-party models should be avoided whenever possible.**

***Index Terms*— Backdoor attack, adaptive attack, deep learning.**

Manuscript received 21 June 2023; revised 25 October 2023; accepted 29 November 2023. Date of publication 4 January 2024; date of current version 11 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62002167, Grant 62072239, and Grant 62372236; and in part by the Open Foundation of the State Key Laboratory of Integrated Services Networks under Grant ISN24-15. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Angelo Spognardi. *(Huaibing Peng and Huming Qiu are co-first authors.) (Corresponding author: Yansong Gao.)*

Huaibing Peng and Anmin Fu are with the School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: paloze@njust.edu.cn; fuam@njust.edu.cn).

Huming Qiu is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: 120106222682@njust.edu.cn).

Hua Ma is with the School of Electrical and Electronics Engineering, The University of Adelaide, Adelaide, SA 5005, Australia, and also with Data61, CSIRO, Eveleigh, NSW 2015, Australia (e-mail: hua.ma@adelaide.edu.au).

Shuo Wang and Yansong Gao are with Data61, CSIRO, Eveleigh, NSW 2015, Australia (e-mail: shuo.wang@data61.csiro.au; garrison.gao@data61.csiro.au).

Said F. Al-Sarawi and Derek Abbott are with the School of Electrical and Electronics Engineering, The University of Adelaide, Eveleigh, NSW 2015, Australia (e-mail: said.alsarawi@adelaide.edu.au; derek.abbott@adelaide.edu.au).

Digital Object Identifier 10.1109/TIFS.2024.3349869

I. Introduction

BACKDOOR attacks have become one of the most dangerous threats to deep learning (DL) systems [1], especially posing severe consequences to those DL-empowered safety and security applications such as autonomous driving [2], [3], [4], access authorization upon facial or speech recognition [5], [6], medical diagnosis [7], malware analysis [8]. Businesses and enterprises are most concerned about such attacks when they adopt machine learning into production [9]. Backdoors can be inserted at different stages of the DL pipeline [1], including during model outsourcing [4], [10], data outsourcing [3], and collaborative distributed learning [11]. Model outsourcing is one of the primary avenues for the implantation of backdoors. Consequently, significant efforts have been made to scrutinize outsourced models to detect or mitigate the effects of these backdoors (as detailed in Section II). However, compared to the extensive and systematic adaptive attacks on adversarial example defenses, which test the robustness of these defenses [12], [13], [14], [15], critical adaptive attacks on backdoor defenses for robustness elucidation are much less explored. This might present false optimistic security implications when these backdoor countermeasures are adopted in practice.

Many preliminary adaptive attack evaluations on backdoor countermeasures are ad hoc and are only performed given a single defense method rather than a number/series of defenses. The purpose is to show whether the proposed defense is robust against the adaptive attack that knows such a newly proposed defense method [16], [17], [18]. Nonetheless, there exists a few holistic adaptive backdoor attacks [19], [20], [21], [22]. We note they all rely on model training regularization (performed under a model outsourcing scenario that an attacker controls the training, except [22]). In addition, almost all exploit designing special triggers such as imperceptible and uninterpretable triggers [21], dynamic triggers [23], dispersed large triggers [22]. To a large extent, it is expected that those backdoor defenses will fail given these special triggers enabled backdoor, as some defenses acknowledged limitations or expected failure cases in front of special trigger usage [17], [24], [25]. In other words, the adaptive attack already breaches the clearly defined threat model of these defenses. We also note that evaluation settings of a few of these adaptive attacks are (unintentionally) problematic. More specifically, these

attacks [19], [20] conduct the backdoor insertion in a model outsourcing scenario when the attacker controls the training data and training process but uses backdoor defenses [26], [27] that require full training dataset (i.e., including both benign and poisoned samples). Ideally, in such circumstances, it should be the user or defender, not the attacker, who oversees the training process.

It is crucial to evaluate the extent of a defense's robustness in conservative attack scenarios, where the defender possesses advantageous knowledge about the attack method and the trigger employed. In these cases, the attacker's capabilities are significantly restricted. Our argument is based on the notion that if an attacker can successfully evade the defense under these stringent conditions, it is apparent that they would also be able to bypass the defense with relative ease under more lenient conditions. This would include situations where they can freely select unique triggers or employ various attack methods.

This study operates under the premise of a restricted adversary capability to execute adaptive backdoor attacks in the context of a model outsourcing scenario. Specifically, our adaptive attack is executed with three key constraints: the use of a simple fixed patch trigger, an input-agnostic backdoor type, and the preservation of a high attack success rate. These restrictions are often placed for competitions such as the Trojan Detection Challenge (TDC) NeurIPS 2022 competition, in which our attack places us as the second place.[1] Even under such conditions, we demonstrate that adaptive backdoor attacks can still trivially succeed.

A. Contribution

We summarize our contributions as following.

- We consider evaluating the model outsourcing backdoor defenses under severely restricted adversary capabilities to resemble a worst-case attack scenario. If defenses still fail, it is obvious that they will fail in other capability-relaxed attack conditions.
- We leverage trigger specificity enhancement and model training regularization together to evade backdoor defenses even though the attacker is restricted from using simple triggers and most conventional input-agnostic backdoor types.
- We extensively evaluate the proposed adaptive attacks *concurrently* against a diverse range of state-of-the-art defenses including Neural Cleanse [17], ABS [28], MNTD [18], STRIP [16], FeatureRE [29] and Beatrix [30], which results affirm its high evasiveness—all these defenses fail. We further reason the challenges of defending against model outsourcing backdoor attacks.

B. Recommendation

In this context, we argue that effective model outsourcing backdoor defenses appear to be quite challenging Therefore, the model user should always avoid the usage of single (non-reputable) third-party provided models in security-critical applications. Training the model by the user even upon outsourced data is more practical in the real world because defending against a poisoning backdoor attack, in this case, is much easier (detailed in Section V-E).

The rest of the work is structured as follows. Section II provides some necessary background about related work. Section III firstly defines the threat model, then presents an overview of our devised adaptive backdoor attack, followed by implementation details. Extensive experiments in Section IV affirm the evasiveness of our adaptive attack concurrently bypassing six SOTA defenses. Further discussions are made in Section V, followed by the conclusion in Section VI.

II. Related Work

A. Backdoor Attack

A backdoor attack has two indispensable components: trigger and backdoor. The trigger is used to activate the previously inserted backdoor (some compromised neurons) in an infected model. The backdoor stealthiness or evasiveness can be improved by devising special types of triggers or advanced backdoor types.

1) Trigger Type: The most conventional trigger is a patch with a fixed pattern located in a fixed position [31]. In fact, such trigger type is now can be effectively captured by existing defenses [17]. The patch pattern and location can vary [23], [32], which can harden the countermeasures to some extent.

Later, there are various trigger designs to be invisible through delicate noise [33], [34], and frequency domain manipulation [7], [35]. In addition, natural triggers such as sunglasses and T-shirts have also been used as triggers [4], [5]. Moreover, natural phenomena such as reflection and rotation are also exploitable to be triggers [36], [37]. Hidden trigger that enables consistency between sample content and label [38], [39] and more evasive sample-specific trigger [23], [40] is also devised. Composite backdoor [41] takes the concurrent presence of multiple class(es) or object(s) as the trigger condition.

2) Backdoor Type: The most studied backdoor type is the source-class agnostic backdoor [1]. In such an attack, any input regardless of its source class containing the trigger will fire the backdoor inserted in an infected model, which will be hijacked to conduct the attacker-specified backdoor effect. The other relatively well-recognized backdoor attack is a source-class specific backdoor [16], [42], [43], referred to as partial backdoor sometimes [30]. The backdoor is activated not only when the trigger is embedded within the input but also when input is selected from attacker-chosen source classes. If the input is from a non-source class, the backdoor does not exhibit even though the input is with the trigger.

Building upon the above backdoor types, there are some related advanced backdoor variants. One is a multiple-backdoor inserted into a single model. A backdoor can be set with a different attacking purpose, so that different backdoor targets e.g., different target labels. Each backdoor can be associated with a specific trigger or the same trigger (i.e., all-to-all attack [31]). Besides these variants, there are quantization backdoors abusing the commercial quantization toolkit (i.e., TensorFlow-Lite and PyTorch Mobile) [10], and latent backdoors affecting a pre-trained model [44] when the down-stream task is learned through transfer learning.

[1]Our team named as NJUST ranked 2nd at https://2022.trojandetection.ai/leaderboards.html.

For all these backdoor types, the trigger can be universal (i.e., a patch) or sample-specific. In other words, the backdoor type is generally orthogonal to trigger types.

B. Adaptive Backdoor Attack

1) Model Outsourcing: In most cases, with straightforward backdoor insertion e.g., insertion only with data poisoning [31], the backdoored model often leaves tangible footprints in the latent or feature space to trigger-carrying inputs, which make the trigger inputs distinguishable from benign inputs. For most adaptive backdoor attacks, they aim to make the latent representation of trigger-carrying samples and benign samples with the trigger indistinguishable [22]. Because they have the observation that most existing backdoor defenses utilize the latent separability of trigger-carrying samples and benign samples to detect the backdoored model or/and trigger-carrying samples. This is commonly done through training regularization by adding additional loss [19], [20]. For instance, Cheng et al. [21] firstly uses a trigger generator (CycleGAN) to gain poisoned images with the trigger (i.e., 'sunset' style). Then during backdoor training, they suppress the activations of those compromised neurons through an interactive backdoored model retraining (i.e., controlled detoxification process). Due to the required training of the trigger generator and the iterative controlled detoxification process, the backdoored training poses high computational overhead, despite a one-time cost to an attacker. Similar studies are [23], [40] that also need to co-train the input-aware trigger generator along with the backdoored model (i.e., the trigger generator is unique to the backdoored model), though they do not explicitly regularize the latent representation of the backdoored model—achieving the same regularization effect at the end.

We can see that all these adaptive attacks share the commonality of training regularization. In addition, they all rely on sophisticated triggers, for example, trigger dispersing the entire image, or sample specific/aware trigger [21], [40] that have already breached threat model of some backdoor defenses they evaluated [16], [17], [25]. Note few model outsourcing adaptive attack evaluations [19], [20] appear to be problematic because they use defenses that are primarily devised for data outsourcing backdoor (see following Section II-B.2), where the user/defender can access the entire training data and train a model by himself/herself.

2) Data Outsourcing: In this attack surface, the attacker is unable to control the model training to enforce regularization during the training process. Thus, adaptive attacks rely on special trigger designs, almost all take advantage of complicated triggers e.g., sample-specific trigger [40], dynamic trigger [23], and distributed trigger [22], partial trigger (or namely source-class-specific trigger) [42], [43]. Because the usage of these triggers often already breaches threat model assumptions of some defenses, especially those model diagnosis [17], [24], and online data diagnosis defenses [16], [25], [45], it is unsurprisingly that they can trivially bypass those defenses. In addition, special backdoor type, in particular, source-class-specific backdoor attack [16], [42], [43] can be viewed as an adaptive attack on the common source-class-agnostic backdoor defenses, which also trivially breaches those defenses assumptions for being evasive.

We note that only Qi et al. [22] explicitly assumes no control of the training procedure exactly fitting the data outsourcing adaptive attack, where the attacker can only poison a small fraction of data to insert the backdoor. To evade latent separability-based defenses, they leverage regularization/cover trigger-carrying samples to make the benign and trigger samples' latent representation to be in-differentiable. The regularized samples are those samples with the trigger but ground-truth labels are intact. The cover samples suppress the strong tangible traces between the trigger and the target class, which is the foundation of latent separation-based defenses. Cover sample usage can degrade the accuracy of clean samples, which is remedied through asymmetric trigger poisoning (i.e., the trigger is transparent during poisoning but is opaque when the attack is launched during online deployment) and distributed partial triggers (the entire trigger is divided into partial triggers that each is used to craft a poisoned or cover sample).

Clean-label poisoning attacks [46] that rely on either feature-collision [38], [39] or image camouflage [3], [47], [48], [49] can be viewed as adaptive attack to some extent as it can bypass the data curator visual inspections. Since the image content and the label/annotation are consistent. However, they are easy to be defeated. More specifically, the feature collision requires knowledge of the victim model (i.e., weight values), it often does not succeed once the user uses a different model architecture for training from scratch. Few works have made efforts [50], [51] to notably increase the attack transferability in this feature collision based attack. As for the image camouflage that abuses the image resizing function, it needs knowledge of input size and resizing algorithm accepted/used by the model training. It trivially fails once these two factors vary [3].

C. Baseline Detection Methods and Metrics

We delve into three baseline detection methods—Neural Cleanse, ABS, and MNTD—chosen for the TDC NeurIPS 2022 competition, which our adaptive backdoor attacks primarily aim to circumvent. While these methods are emphasized by the competition organizers, we also evaluated our adaptive attacks against other non-baseline backdoor countermeasures to assess their evasiveness.

1) Neural Cleanse: The intuition of Neural Cleanse comes from the characteristic of the backdoor trigger, which hijacks any input carrying this trigger into the targeted class y_t, regardless of the source class of the input. In this context, the trigger creates a 'shortcut' from the region of non-targeted classes' hyperspace to the region of targeted class hyperspace in a backdoored model. From the reverse-engineering perspective, given a compromised model, it requires an abnormally small change/perturbation on the input to cause such benign input to be misclassified to the targeted label compared to other uninfected labels. To this end, Neural Cleanse aims at reverse-engineering the trigger and judging whether the trigger is abnormal or not. Its operation follows the below steps.

- Step 1: Using an optimization algorithm, for a given label, reversing a candidate trigger that can mislead inputs from

other classes to be classified into this given label when the candidate trigger is present.
- Step 2: Repeating step 1 for each label and obtaining a candidate trigger per label.
- Step 3: Detecting if there is any trigger candidate that is significantly smaller than others. If so, this trigger is selected and the model is regarded as backdoored.

The reconstructed trigger is essentially perturbations added to the input image, which can be measured by L_1 norm. The trigger outliers (e.g., much smaller candidate trigger) can be detected using Median Absolute Deviation (MAD). By default, Neural Cleanse adopts an anomaly index larger than a threshold of 2.0 to determine that the candidate trigger is an outlier.

2) MNTD: It is an AI-against-AI method. By feeding a few querying samples to the model-under-test (MUT), MNTD concatenates MUT logits of these samples as a vector. This vector is fed into a so-called meta-classifier to return a score for determining whether the MUT is a backdoored model. The acquisition of the meta-classifier requires training many shadow models with the same task as the MUT. So MNTD needs to access at least a partial training dataset used to train the MUT (the target model) for training shadow models.

a) Shadow model generation: Both benign and backdoored shallow models are required. Benign shadow models can be generated by training with different model hyper-parameters initialized on the clean dataset. For backdoored shadow models, various backdoor settings can be sampled from a generic set of backdoor distributions to generate different backdoored models, which is mainly through poisoning training datasets with different triggers. Because many shadow models (from hundreds to even thousands) have to be trained, these shadow models are normally with a shallow structure and their training not converged to reduce computational overhead.

b) Meta-classifier training: The meta-classifier is to make a binary classification: backdoored or clean for a shadow model during its training. The meta-data fed into the meta-classifier is a feature vector that is the concatenation of N logits from a given shadow model corresponding to N querying images. Its ground-truth output is a binary label: backdoored or clean of the given shadow model (this is known). Note querying samples are less effective if they are randomly selected. Therefore, MNTD co-optimizes these querying samples along with the meta-classifier optimization.

c) Target model detection: Given the obtained optimal querying set, and for a MUT, M, we can feed these querying samples to the M and obtain logits per sample, and then concatenate these logits into a vector as input of the obtained meta-classifier. The meta-classifier produces a score to indicate whether the MUT has been backdoored or not.

3) ABS: ABS assumes that the backdoor compromises a single or only a few neurons, namely impaired neurons, which can be identified through the following steps.

a) Neuron stimulation analysis: ABS uses held-out benign inputs to first activate individual internal neurons and study their effects on each output label. Given a specific neuron, ABS uses the neuronal stimulation function (NSF) to derive the output activation function for that neuron's activation value label and then analyzes the effect of this neuron's activation value on the activation value of subsequent layers to determine the relationship between that neuron's activation and the output layer.

b) Identifying compromised neuron candidates: The previous stimulus analysis computes the NFS of selected layers in a subject model. If a neuron significantly improves the activation of a specific output label, this neuron is labeled as a candidate neuron.

c) Identify backdoor models: After acquiring the candidate neurons, ABS further identifies the real damaged neurons by reverse-engineering triggers corresponding to per neuron. Given a reverse-engineered trigger can mislead clean sample prediction well once it is embedded, this neuron is deemed as impaired. The percentage of benign inputs that can be corrupted by the trigger is called reverse-engineer attack success rate (REASR) The subject model is considered to be backdoored if REASR is higher than a threshold, 0.88 used by [28].

III. Model Outsourcing Adaptive Backdoor

We first define the threat model used in our adaptive attack and then give an overview of the proposed model outsourcing adaptive backdoor attack from two aspects, followed by detailed implementations per aspect.

A. Threat Model

The attacker has full access to the training dataset and controls the training process. The aim of the attacker is to backdoor the models that evade detection of multiple influential detection countermeasures (mainly Neural Cleanse, MNTD, and ABS chosen by the NeurIPS competition, despite other detections being evaluated and bypassed). In this context, the attack is an adaptive attack since the attacker has knowledge of the defense.

However, there are several constraints with which the attacker has to comply, thus greatly limiting their capability. We adopt the constraints specified by the NeurIPS competition. Firstly, the backdoored models must have a more than 97% average attack success rate (ASR) given those attacker-supplied backdoored models. So that the attacker cannot simply reduce the ASR to invalidate the detection. Here, 200 backdoored models of the same task (i.e., MNIST) have to be provided to the defender. Note there are 200 reference clean models per task provided by the organizer. Secondly, the attacker is restricted by the arbitrary usage of a trigger, the trigger per backdoored model must be a trigger provided by the competition organizer, which is essentially the simplest static and small patch trigger. Note that as one merit, the simple patch-based trigger (compared with e.g., invisible trigger, blended trigger, or input-aware trigger) tends to be the most straightforward and one of the very few methods that are effective in the physical world [52], [53]. Because the patch such as the white-square trigger used in this study can be simply represented by a paper sticker physically pasted on the victim object [31]. A primary requirement of

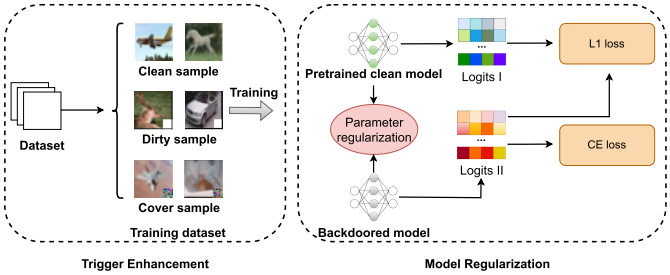


Fig. 1. An overview of trigger enhancement (i.e., specificity enhancing) and backdoored model regularization (i.e., logits constraint).

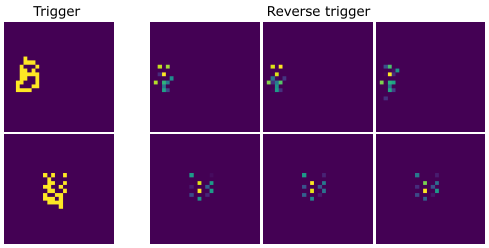


Fig. 2. (Left) Original trigger. (Right) Reverse-engineered triggers by Neural Cleanse, which vary slightly upon repeated runs and are not exactly the same as the original trigger but equally activate the backdoor.

NIPS 2022 Trojan Detection Challenge is that we could only use the most simple square trigger with a fixed size and position. In this context, the only variable we can explore is the color pattern. This is the major reason for us sticking with color patterns to create cover triggers. More specifically, we simply used random colors to generate cover triggers. It is noted that the trigger pattern is not limited to color patterns but also shapes and positions. Therefore, creating cover triggers with random colors or/and shapes or/and positions is expected to work as well. Nonetheless, since exploiting color patterns is already sufficient for enhancing specificity and is strictly within the NIPS Challenge requirement, we have followed this approach. Thirdly, the backdoor attack is constrained to the input-agnostic backdoor, where any input carrying the same trigger will be misclassified by the backdoored model. In summary, the attacker is restricted to applying the simplest trigger and most conventional backdoor type.

Under these strict capability constraints, evasiveness becomes challenging to achieve even though the attacker is allowed to own full access to the training dataset and control over the model training procedure.

B. Overview

As shown in Figure 1, our adaptive attack consists of two modules: trigger enhancement and model regularization. Trigger enhancement produces 'cover triggers', sharing the location and pattern of the original 'dirty trigger' but differing in colors to boost trigger distinctiveness. This approach can be expanded to utilize methods like sample-specific triggers. Broadly, trigger enhancement aims to capitalize on various trigger types. Meanwhile, model regularization aligns parameters (such as logits, latent layers, and activations) of the backdoored model with those of its clean counterpart. We consistently utilize a pretrained clean model as a regularization benchmark for the backdoored one. With the combined efforts of trigger enhancement and model regularization, the stealthiness of the backdoored model is markedly increased.

Corresponding to attacking constraints as in Section III-A, the trigger specificity is to harden the trigger reverse-engineering through the non-deterministic optimization leveraged by Neural Cleanse. We note that Neural Cleanse is only able to approximately reconstruct the original trigger, where the reconstructed trigger is visually similar but not *exactly same* as the original trigger, see the reconstructed effective candidate trigger in Figure 2. In other words, any candidate trigger that is around the original trigger can satisfy the Neural Cleanse, which can be generally thought of as a sub-space radius. Without enhancing the trigger specificity, the radius is large, so it is easy to identify a candidate trigger entering into trigger effective circle through non-deterministic. Therefore, we propose to leverage cover triggers that are similar but not the same as the real trigger to enhance the specificity of the real trigger, which substantially reduces the radius, rendering the hardness of entering the original trigger effective circle taken by the Neural Cleanse non-deterministic optimization process.

Corresponding to attacking constraints as in Section III-A, model regularization aims at regularizing the logits of the backdoored model to be similar to the logits of the clean model counterpart. This is because the logits are key meta-data used by the MNTD to judge whether a subject model is a backdoored or clean model. As long as the logits of the backdoored model are indistinguishable from that of the clean model, the meta-classifier of the MNTD is expected to fail to tell a backdoored model from a clean model. We note that model regularization is not limited to 'logits' regularization, weights regularization or latent representation can be incorporated concurrently to bypass backdoor detection that relies on those key factors. Nonetheless, we show that logit regularization can already trivially evade ABS at the same time due to its strong assumption that cannot be often met in practice.

C. Implementation

1) Trigger Specificity Enhancement: Defenses upon trigger reconstruction usually cannot reverse-engineer triggers that are identical to the real trigger, they can only reverse-engineer similar triggers due to the reverse-engineering is based on non-deterministic optimization. However, the effect of these similar reconstructed triggers can resemble the real trigger due to the lack of specificity of the real trigger. Here, we propose to enhance the trigger specificity through the usage of cover triggers. When creating poisonous samples to train the model, there are two types of poisonous samples: dirty samples and cover samples.

- *Dirty Sample:* The attacker chosen trigger Δ is added to a randomly selected clean sample x to gain a poisonous sample $x_t = x + \Delta$. The label of x_t is altered to be y_t which is the attacker-targeted class. In this context, a small dataset D_t of dirty samples is created.
- *Cover Sample:* A randomly generated cover trigger $\Delta' \approx \Delta$ similar as Δ but not same is added to a randomly selected clean sample x to gain a poisonous sample

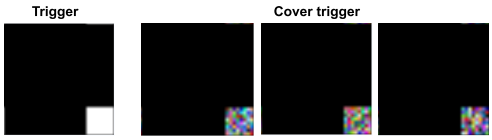


Fig. 3. (Left) The white square in the lower right corner is the exact trigger. (Right) The three cover triggers in random colors, here we use only three random colors as examples.

$x_c = x + \Delta'$. But note that the label of the cover sample x_c is intact to be its ground-truth label y. In this context, a small dataset D_c of cover samples is created.

An example of cover triggers is shown in Figure 3. The cover samples will force the model to only link the presence of exact trigger Δ rather than similar triggers Δ' to activate the backdoor. In other words, it enhances the trigger specificity, rendering the trigger reverse-engineering defenses upon non-deterministic optimization facing difficulties for reconstructing the exact trigger Δ. To this end, the objective loss $\mathcal{L}_{\text{ts}}$ of the backdoor model f_{bd} for enhancing trigger specificity can be formulated as:

$$\mathcal{L}_{\text{ts}} = \sum_{x \in D} \mathcal{L}(f_{\text{bd}}(x), y) + \sum_{x_t \in D_t} \mathcal{L}(f_{\text{bd}}(x_t), y_t) + \sum_{x_c \in D_c} \mathcal{L}(f_{\text{bd}}(x_c), y), \tag{1}$$

where $\mathcal{L}(\cdot)$ is implemented by a commonly used categorical cross-entropy loss. Note that D, D_t, and D_c are datasets of clean samples, dirty samples, and cover samples, respectively. The D_t and D_c are small.

2) Backdoored Model Regularization: In particular, we use l_1-norm to force the logits, v_{bd}, of the backdoored model f_{bd} on clean samples to be close to the logits, v_{cln}, of the clean model f_{cln}. The logits constraint/regularization objective loss $\mathcal{L}_{\text{lc}}$ can be expressed by:

$$\mathcal{L}_{\text{lc}} = \sum_{v \in V} \frac{1}{n} \| v_{\text{bd}} - v_{\text{cln}} \|. \tag{2}$$

The final objective loss of the backdoored model is expressed as:

$$\mathcal{L}_{\text{bd}} = \mathcal{L}_{\text{ts}} + \gamma_1 \mathcal{L}_{\text{lc}}, \tag{3}$$

where γ_1 is a regularization factor. It is set to be 1 without other trials in this work. Because it is already sufficient for achieving excellent performance.

Note that the $\mathcal{L}_{\text{lc}}$ requires the logits from a clean model counterpart as a reference. Therefore, before training the backdoored model, a clean model with the same hyperparameters except the loss constraint is co-trained to obtain the reference. More specifically, for *each training epoch*, we feed a batch of clean samples into the clean model to gain the logits V_{cln}. Then, we feed the same batch of clean samples into the backdoored model to get V_{cln}, and consequentially compute the loss of $\mathcal{L}_{\text{lc}}$. In this way, $\mathcal{L}_{\text{lc}}$ encourages the backdoored model to behave the same as the clean model in the clean samples in terms of logits being exploited for defense.

TABLE I
DATASET SUMMARY

Dataset	Training; Testing Size	Image Size	Num. Classes
CIFAR-10	50,000;10,000	32×32×3	10
GTSRB	39,209;12,630	32×32×3	43
MNIST	60,000;10,000	28×28×1	10

IV. EXPERIMENTAL EVALUATIONS

In this section, we present our experimental setup and the results of our experiments.

A. Setup

1) Dataset: Three datasets of CIFAR-10 [54], GTSRB [55], and MNIST [56] are used for extensive experiments. Table I summarizes the details (i.e., training/testing size, image size, and number of classes) of each dataset.

2) Model: We choose a 2-layer-CNN, a 6-layer-CNN, and PreActResNet18 [57]. This 2-layer-CNN has two convolutional layers and two fully-connected layers, which were used to train the MNIST dataset by the released source code of MNTD. This 6-layer-CNN has six convolutional layers and two fully-connected layers, which were used in the released source code[2] of Neural Cleanse customized for the GTSRB dataset. PreActResNet18 has a backbone of ResNet18 [58], which modifies the position of the latter activation layer to enhance the constancy of the shortcut connection of the network model to improve ResNet18 performance.

3) Machine: Experiments are implemented on either Python 3.7 with PyTorch or Python 3.6 with TensorFlow for the sake of following the released source code of evaluated defenses. The machine is a LENOVO laptop with an AMD Ryzen 7 5800H CPU with 8 cores and 16GB DRAM memory, and a GeForce GTX 1650 GPU with 4GB memory. All our experiments are trained and evaluated on this machine.

4) Metric: For quantitatively measuring the backdoor attack performance, two metrics of attack success rate (ASR) and clean data accuracy (CDA) are widely used [1].

- The ASR is the probability of a trigger-carrying sample being misclassified by the backdoored model into the attacker-chosen targeted class.
- The CDA is the probability of a sample carrying no trigger being classified into its ground truth class.

For a successful backdoored model, its ASR should be high, e.g., close to 100%, especially for the focused all-to-one or source-class-agnostic backdoor attack. The CDA should be similar to that of the clean model counterpart so that observing the validation accuracy of the backdoored model through the held-out validation dataset cannot infer any malicious behavior.

B. Baseline Detection Results

Here, we evaluate the detection performance of so-called baseline detection methods chosen by the NeurIPS competition. Other detection methods, denoted as non-baseline

[2]The Neural Cleanse source code at https://github.com/bolunwang/backdoor.

TABLE II
THE AVERAGED CDA AND ASR PERFORMANCE OF SHADOW MODELS AND TESTING MODELS (MUTS)

Model		Shadow models		Testing models	
		CDA	ASR	CDA	ASR
Clean	2-layer-CNN	95.14%	n/a	98.50%	n/a
Backdoored	2-layer-CNN	94.19%	86.07%	98.36%	99.70%

detection methods to ease description, are evaluated in Section IV-C.

1) MNTD: The Area Under the ROC Curve (AUROC) is used by MNTD to evaluate its performance. The AUROC value is between 0 and 1. If the detection has a high AUROC (i.e., close to 1), it means that MNTD has a strong discriminative ability to differentiate a clean model from a backdoored model. The AUROC is also adopted by the NeurIPS competition.

As aforementioned, MNTD[3] requires training many shadow models before training the meta-classifier. So its computational overhead is high. To reduce computational overhead, the MNIST is mainly used by the NeurIPS competition. We have trained 2304 clean and 2304 backdoored shadow models—the number is similar as [18]. Those shadow models are used to train the meta-classifier. For the testing purpose, we have trained 256 clean and 256 testing models, it took about 22 hours to complete the shadow models training, The average CDA and ASR of shadow models and testing models that are MUTs are summarized in Table II. Note for the backdoored testing models, trigger specificity enhancement and logits constraint based on Equation 3 are employed for incorporating the adaptive attack.

Following [18], five meta-classifiers are trained upon the shadow models. Firstly, we reproduce the results based on the default setting of the released code [18] by evaluating testing models (i.e., with 256 clean models and 256 backdoored models). In this case, the average AUROC value of the five meta-classifiers is up to 97%, which indicates the strong backdoor detection capability by MNTD. We illustrate the effectiveness of our method by setting the poisoning rate and cover rate to be 10%, respectively, and also apply the logits constraint to gain 256 backdoored testing models, For these testing backdoored models, their average ASR is up to 99% and average CDA is 98%. We also have 256 clean testing models. For our adaptive attack, the MNTD exhibits an average AUROC value of 44.75%, which is close to guessing (i.e., AUROC is 50%). Therefore, our adaptive attack successfully bypasses MNTD. The reason is that MNTD heavily relies on the logits for backdoor detection. The logits are sensitive to various factors including not only the demonstrated adaptive attacks but also hyperparameter change (i.e., batch size, epoch number) [59] even when there is no adaptive attack occurring.

2) Neural Cleanse: We implement the attack using GTSRB with six convolutional layers and two fully connected layer models. The trigger is a white square of size 5×5 at the bottom right corner of the image. Cover triggers are simply with the same shape and same position but different patterns, see Figure 3. Any sample carrying the white square trigger will hijack the backdoored model for misclassification, which is an input-agnostic backdoor.

[3]We reproduced the code at https://github.com/AI-secure/Meta-Nerual-Trojan-Detection.

TABLE III
CLEAN/BACKDOORED MODEL PERFORMANCE

Model	CDA	ASR
Clean model	96.46%	n/a
Backdoored Model (Baseline, PR=10%)	96.30%	98.44%
Backdoored Model (Our adaptive, PR=1%)	95.90%	98.05%
Backdoored Model (Our adaptive, PR=10%)	96.75%	99.89%
Backdoored Model (Our adaptive, PR=30%)	96.74%	99.91%

1. The baseline backdoored model is attacked through default data poisoning only without cover samples and training regularization.
2. PR stands for poisoning rate. For the adaptive backdoor, both the dirty sample fraction and cover sample fraction are set to be the same PR (i.e., 1%).

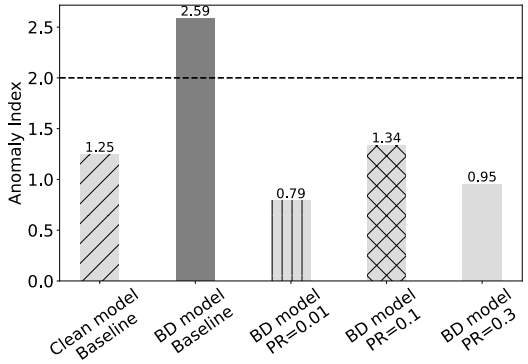


Fig. 4. Anomaly Index (AI) reported by Neural Cleanse on the clean model, baseline backdoored (BD) model, and our adaptive backdoor attacked models. PR is poison rate. If AI > 2.0, the model is backdoored; otherwise, the model is clean.

We have set three poisoning rates (PR) $\{1\%, 10\%, 30\%\}$. For comparisons, we have evaluated Neural Cleanse on the clean model counterpart, and baseline backdoored model (default backdoor setting without our adaptive attack). Note that trigger specificity enhancement itself is found to be already sufficient to bypass the Neural Cleanse and the logits constraint does not help too much to bypass the Neural Cleanse (detailed in Section V-A). Therefore, the adaptive attack results in this part are upon the trigger specificity enhancement only according to Equation 1.

From Table III, we can see that our adaptive attack infected models have similar backdoor attack performance as the baseline backdoored model. This means that our adaptive backdoor attack is as stealthy as the baseline backdoor. However, the baseline backdoor can be effectively detected by the Neural Cleanse, where the anomaly index is 2.59 (see Figure 4), higher than the detection threshold of 2.0. In contrast, our adaptive backdoor-attacked models successfully evade the Neural Cleanse detection, because the anomaly index is always sufficiently lower than 2.0. In fact, the anomaly indices of the infected models are often not higher than the clean model anomaly index.

3) ABS: Here, we use the CIFAR-10 dataset and the PreActResNet18 model. The trigger size is changed to 8×8 to

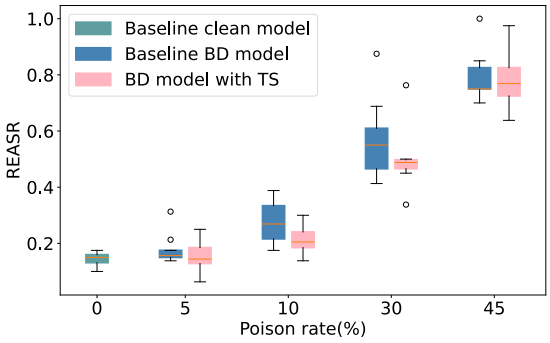


Fig. 5. Relationship between REASR value of ABS and poisoning rate. The ABS is sensitive to the poisoning rate, which is its key inadvertent limitation. Our adaptive attack imposes trigger specificity (TS) enhancement only.

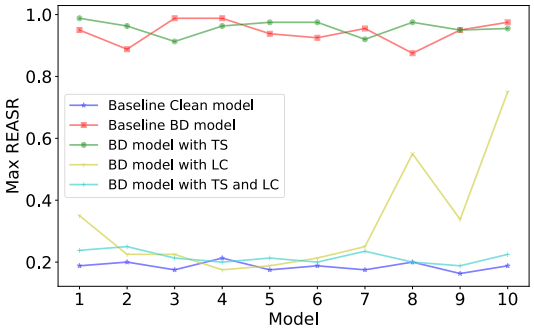


Fig. 6. ABS detection performance when different types of backdoor attack are applied. Our adaptive attack (once the TS + LC is imposed) trivially bypasses it. We backdoor 10 models with each being infected with a different target label. Ten clean models are also trained. Per model per type of attack, the max REASR is reported as ABS is repeated 10 times against a given model.

respect the default ABS setup. For the poisoning rate, it has been set to {5%, 10%, 30%, 45%}. For comparisons, we have evaluated the ABS detection performance on the baseline backdoored model. For the same compromised model and the same infected target label, we repeated the ABS 10 times.

The detection performance as a function of the poisoning rate is detailed in Figure 5. The maximum value of REASR of the clean model (poison rate is 0%) is 0.175. We found that ABS is essentially sensitive to the poisoning rate. When the poisoning rate is 5%, the REASR of both the baseline backdoored model and adaptive attack (TS only) infected model is similar to the clean model. Please note when the poisoning rate is small 5%. The REASR for the backdoored model witnesses a relatively notable increase when the poisoning rate is 10%. However, it is still significantly lower than the threshold of 0.88. The REASR distribution of the baseline backdoored model is higher than that of our adaptive attacked model (with TS only), as shown in Figure 5. This means that TS-only adaptive attack is able to increase the evasiveness to some extent.

Only when the poisoning rate is extremely high (i.e., 45%), the ABS demonstrates its claimed detection capability. Because only in this case, the compromised neurons will be dominated by a few neurons, which is the key assumption of the ABS. When we increase the data poisoning rate to 30%, REASR values are increased to be around 0.5. By a poisoning rate of 45%, the REASR is around 0.8. To this point, using the default threshold of 0.88 is able to detect the backdoored model in some cases. Note that ABS's high sensitivity to the poisoning rate is a main limitation. Because in practice, as long as the ASR is high, there is *no need to use an extremely high poisoning rate* for backdoor insertion.

To this end, the above experiments on ABS only leverage trigger specificity (TS) enhancement, now we impose logits constraint (LC). To facilitate the ABS detection, we intentionally set an extremely high poison rate of 45% for both the dirty and cover samples. We repeated the attack 10 times with each class of CIFAR-10 as the target class, and the max REASR value among ten attacks is reported. From Figure 6, we can see that our adaptive attack trivially evades ABS. The max REASR of the adaptive attacked model is almost similar to that of the clean model. We have also evaluated the ABS by only using the LC attacked model, we can see that the LC dominates the evasiveness for the majority of backdoored models. This is easy to understand because the ABS checks the activations of neurons to find compromised neurons. The LC is somehow like an activation value regularization. Once the LC is applied, the assumption that there are only a single or few compromised neurons in the backdoored model is no longer met to a large extent.

C. Non-Baseline Detection Results

As exhaustively examining other detection methods is infeasible, we choose Beatrix [30], FeatureRE [60], and STRIP [16]. Because Beatrix is the latest detection method that is attack-agnostic. FeatureRE is the latest trigger reverse-engineering study from the feature space rather than the input pixel space like Neural Cleanse [61]. The STRIP is one of the most popular online sample-level defenses. We defer the discussion on limitations of other model outsourcing backdoor defenses in Section V-D.

1) FeatureRE: In contrast, to reverse-engineer the trigger in the input/pixel space [17], [62], FeatureRE considers reverse-engineer it via the feature space, which is able to counter not only static patch (pixel space) trigger enabled but also features trigger (i.e., exhibited by dynamic pixel triggers) enabled backdoor attack [60]. FeatureRE's key intuition is that the features representing the trigger are orthogonal to other benign features in latent representation. For input-agnostic backdoor attacks, any trigger-carrying samples are mispredicted to an attacker target label. Therefore, the backdoored model will ignore other features and predict the label as the target label upon the dominant trigger feature. Consequentially, the trigger feature is assumed to form a separated hyperplane space in the high-dimensional space. This hyperplane can guide the search of feature space triggers through constraints imposed in feature space as opposed to the constraints in the input space like Neural Cleanse [17]. This feature space trigger reverse-engineering is performed along with the feature transformation function formation (i.e., a neural network transfers a clean image into a trigger-carrying image under constraints such as semantic preservation).

We use CIFAR-10 trained by PreActResNet18 for FeatureRE evaluation.[4] The trigger size is $5 \times 5 \times 3$. The

[4]The reproduction is based on the released source code at https://github.com/ru-system-software-and-security/featurere.

TABLE IV
RESULTS OF FEATURERE

Model	CDA	ASR	Mixed-value
Clean model	94.42%	n/a	−0.0874
Backdoored Model (Baseline)	94.05%	99.70%	−0.8467
Backdoored Model (With TS only)	93.76%	99.42%	−0.7854
Backdoored Model (With LC only)	94.01%	99.49%	−0.7522
Backdoored Model (With TS and LC)	93.90%	99.87%	−0.3947

1. TS means trigger specificity enhancement. LC means logits constraint is applied. select eyeglasses as the innocent feature for evaluations.
2. Mixed-value is a backdoor indicator. A backdoored is detected if it is lower than a threshold of −0.75.

poisoning rate is set to be 10% (for both dirty and covered samples). The feature space trigger reverse-engineering runs 400 epochs. After that, a so-called mixed-value is obtained. The mixed-value threshold is set to −0.75. If mixed-value is less than this threshold (−0.75), the subject model is regarded as backdoored; otherwise, clean. The FeatureRE results are summarized in Table IV. As a reference, a clean model is with a 94.42% CDA, exhibiting a mixed-value of -0.0874. The baseline backdoored model with an ASR of 99.70% and a CDA of 94.05% had a mixed-value of −0.8467. As for the adaptive attack, by only imposing trigger specificity enhancement or logits constraint, the mixed-value of the infected model are −0.7854 and −0.7522, respectively, which are similar to the mixed-value of the baseline backdoored model (−0.8467). Therefore, TS or LC alone adaptive backdoor is unable to evade FeatureRE. Once both TS and LC are imposed in our adaptive attack, the FeatureRE is evaded, as the mixed-value drops to −0.3947, which is far below the threshold. To our best knowledge, our attack is the first to trivially bypass the FeatureFE even with the simplest patch trigger—the easiest trigger type that should be defeated by the FeatureRE.

2) STRIP: It turns the dominant hijacking effect of the trigger into a weakness for detecting the trigger-carrying samples online [16]. Generally, if a model is backdoored and the trigger-carrying sample is present, then strong perturbations on this trigger-carrying sample will not disturb its prediction of the target label. In other words, this trigger-carrying sample under different perturbations can still exhibit highly consistent predictions, quantitatively measured by low entropy. Otherwise, a clean input under varying perturbations exhibits low consistency predictions, thus a high entropy.

Figure 7 show results[5] of entropy distributions of both benign samples and trigger-carrying samples under baseline backdoor attack and our adaptive attack (TS+LC). The poisoning rate is set to be 10%, the model is a 6-layer-CNN used in the source code of Strip and is trained on CIFAR-10 dataset. The distribution is evaluated on 1000 trigger-carrying and 1000 clean images. As for the baseline backdoor model, we can see that STRIP is effective in distinguishing trigger-carrying samples from clean samples. Because there is a clear gap between trigger-carrying and clean samples' distributions. However, for the adaptive attack infected model, the two distributions have a large overlap, rendering STRIP being ineffective. More specifically, given a preset false rejection rate (FRR) of 1%/3%, the false acceptance rate (FAR) of STRIP on the baseline backdoor is 0%/0%. In contrast, given a preset FRR of 1%/3%, the FAR of STRIP on adaptive backdoor is up to 82.9%/64.3%, which becomes unacceptable in practice. The main reason is that the trigger specificity breaches the strong hijacking effect of the trigger under different perturbations. Once a slight permutation occurs on the trigger, the perturbed trigger (i.e., like the cover trigger now) will not activate the backdoor anymore due to the cover trigger suppression effect.

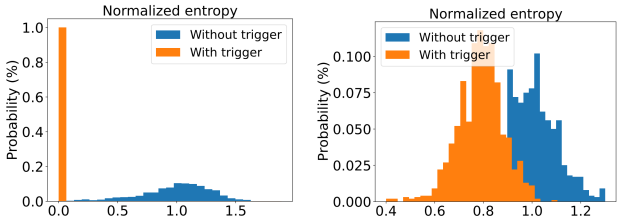


Fig. 7. Entropy distribution of clean samples and trigger-carrying samples measured by STRIP. (Left) Baseline backdoored model. (Right) Our adaptive attack infected model.

3) Beatrix: It views detecting trigger samples as a problem of detecting out-of-distribution (OOD). Since the intermediate/latent representations of trigger-carrying samples are different from clean samples of the given label—for trigger-carrying samples, the given label is the target label. In other words, samples within the infected label can be decomposed into two clusters in latent space. Despite this intuition has been utilized in previous detection such as SCAn [26], [43], the Beatrix takes the decomposition into higher order information between benign and trigger-carrying samples, while previous detection [26], [43] only utilize the first moment (mean) discrepancy of latent representation. On top of it, Beatrix model latent feature using the Gram matrix rather than a commonly used Gaussian distribution. Then Median Absolute Deviation (MAD) is used to measure the deviation of trigger-carrying samples, while kernel-based two-sample testing technique without any assumptions on the distribution-regularized maximum mean difference (RMMD)-based detection technique is used to enhance the adversarial robustness.

We first produce a clean model as a reference according to the released source code.[6] The model is PreActResNet18 and the dataset is CIFAR-10, the clean model CDA is 94.10%. Then we apply our adaptive attack (TS and LC) against Beatrix by setting the poisoning rate to be 10%. The CDA of the attacked model is 93.35%, which is on par with the CDA of the clean model. Note that Beatrix regards the model/label being infected if the RMMD$\geq e^2$ is given a tested class. In this context, the RMMD value of this adaptive attack is 128.03$\geq e^2$. So that the TS + LC based adaptive attack is unable to bypass the Beatrix. However, this does mean Beatrix is insusceptible to adaptive attacks as we show below.

[5]The reproduction is based on the released source code at https://github.com/garrisongys/STRIP.

[6]The reproduction is based on the source code at https://github.com/wanlunsec/Beatrix.

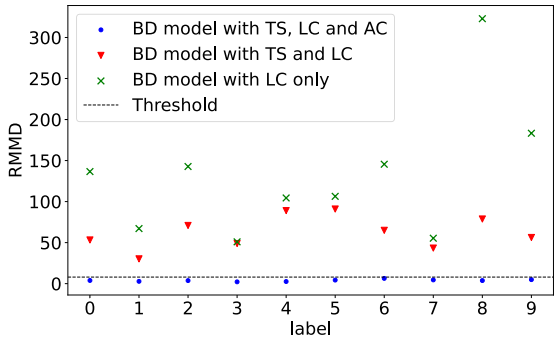


Fig. 8. Beatrix detection performance under adaptive backdoor attacks: backdoored model with LC only (Equation 2); backdoored model with TS + LC (Equation 3); backdoored model with TS + LC + AC (Equation 5).

The reason is that Beatrix analyzes the activations of the representation layer, not the logits. Therefore, imposing simple logit constraints may not be evasive in this case. In this context, we impose the regularization on the activations of the representation layer. When the anomalous scale (in terms of activation) of the trigger-carrying input is constrained to be as much as possible similar to the activations clean input, thus disabling Beatrix. On top of TS + LC, we add another activation constraint (AC) $\mathcal{L}_{\text{ac}}$ expressed as:

$$\mathcal{L}_{\text{ac}} = \sum_{a \in A} \frac{1}{n} \|a_{\text{bd}} - a_{\text{cln}}\|. \tag{4}$$

Then the final loss is expressed as:

$$\mathcal{L}_{\text{bd}} = \mathcal{L}_{\text{ts}} + \gamma_1 \mathcal{L}_{\text{lc}} + \gamma_2 \mathcal{L}_{\text{ac}}, \tag{5}$$

where $\mathcal{L}_{\text{lc}}$ is given by Equation 2 and $\mathcal{L}_{\text{ac}}$ is given by Equation 4, respectively. Again, γ_1 and γ_2 are set to be 1, respectively.

Once we take this slightly modified Equation 5 to backdoor the model, the adaptive infected model trivially bypasses the Beatrix detection. At the same time, the infected model CDA is 92.89% similar to the clean model CDA of 94.10% (only about 1% drop). With the above attack settings unchanged (i.e., poisoning rate of 10%), We repeated the experiment 10 times: each repeat with a different infected label. The results are shown in Figure 8, we can see our adaptive backdoor attack (once AC according to Equation 5 is imposed) bypasses the Beatrix. Because RMMD values of the backdoored model are now all below the threshold e^2 (mostly below 3). To the best of our knowledge, this is the *first successful adaptive attack against Beatrix*.

V. Discussion

A. Trigger Specificity or Training Regularization

Our attack is a hybrid of trigger specificity enhancement and model training regularization. The trigger specificity enhancement is mainly motivated to evade trigger reverse-engineering detection, e.g., Neural Cleanse. The model training regularization is mainly motivated to defeat other types of defense, e.g., MNTD, and Beatrix, which mainly depend on inspecting latent representations.

For detection defenses based on trigger reverse-engineering, enhancing trigger specificity alone can often bypass them, as shown in Figure 4. We conducted ablation studies where we removed the trigger specificity enhancement and relied solely on regularization when evaluating with Neural Cleanse. In this case, only the logits constraint was applied for the adaptive backdoor attack, without any trigger enhancement. The resulting anomaly index (AI) for Neural Cleanse stood at 2.3507, surpassing the threshold of 2. This indicates that Neural Cleanse can still detect the backdoor under these conditions.

Note Neural Cleanse does not examine the latent representation, unlike ABS, FeatureRE, and Beatrix that do examine the latent representations of the subject model, it is instrumental for imposing both model regularization and trigger enhancement together for defenses that examine the latent representation. As we have extensively validated on ABS (Figure 6), FeatureRE (Table IV), and Beatrix (Figure 8), for those defenses, enhancing either the trigger specificity or training regularization alone is unable to bypass them. In other words, the model outsourcing regularization and trigger specificity have to be applied simultaneously.

Therefore, it is preferable to deploy both trigger specificity enhancement and model training regularization concurrently to adaptively backdoor DL models in model outsourcing scenarios. More generally, both advanced trigger crafting (not limited to cover trigger usage to enhance specificity, other means such as sample-specific triggers are also viable) and training regularization shall be utilized to perform adaptive attacks to be evasive in this context. Because the attacker indeed has full control over the training dataset and model training procedure.

TABLE V
The Results of the Baseline Attack and Our Attack

	Max AUROC	MNTD AUROC	Accuracy based detector AUROC	Specificity based detector AUROC
Baseline attack	73.9%	73.9%	63.4%	71.9%
Our attack	52.6%	52.6%	40.7%	47.4%

B. Comparison

There is a baseline attack[7] performed by the organizer of the TDC NeurIPS 2022 competition. The attacking performance of this baseline and our attack are shown in Table V—the results are extracted from the leaderboard.[8] Our attack significantly surpasses the baseline. The trigger specificity enhancement and model regularization in our approach are distinct from those in the baseline. At its core, our strategy aims to minimize the differences between the backdoored and benign models. Specifically, for all non-triggered samples (clean and covered), we aim for the backdoored and clean models to behave as similarly as possible. For all triggered samples (dirty ones), their latent representations should closely align with those of the target class.

As for trigger specificity enhancement, the training dataset consists of three parts: clean samples, dirty samples, and cover

[7] https://github.com/mmazeika/tdc-starter-kit/tree/main/evasive_trojans
[8] https://2022.trojandetection.ai/leaderboards.html

samples, which are designed to reinforce the specificity of the backdoor. In contrast to the baseline attack, our cover sample labels are not set to be the predicted softmax of the clean model, but their ground truth labels. In addition, for backdoored models with poor specificity, we perform secondary fine-tuning with a smaller learning rate.

As for model regularization, we use L1 loss to encourage the logits of non-triggered samples on the backdoored model to be close to the logits of the clean model, which, combined with the standard cross-entropy loss, injects a more hidden backdoor into the model. We consider regularizing the logits of both clean and covered samples, as opposed to considering only clean samples in the baseline attack.

C. Poison Rate of Dirty/Cover Samples

We have set the poisoning rate (PR) of dirty samples and cover samples to be equal. For example, when evaluating Neural Cleanse in Section IV, we have used a poisoning rate as small as 1%, which means that 1% training samples are used to create dirty samples and the other 1% samples to create cover samples. Our intuition was that balancing the PR of dirty samples and cover samples is expected to have a high ASR while maintaining a good trigger specificity. We have now evaluated the imbalanced setting of PRs for dirty samples and cover samples. Since trigger specificity is especially useful to evade reverse-engineering backdoor detection represented by Neural Cleanse. Therefore, we stick with Neural Cleanse to perform the ablation study on the imbalanced PRs of dirty and covered samples. Specifically, we study the relationship between the measurable detection capability (Anomaly Index) of Neural Cleanse and the dirty/cover samples proportion. The dataset on which we conducted this ablation study is GTSRB.

The results are depicted in Figure 9—this is averaged results from six repeated evaluations. The poisoning rate of dirty/cover samples is fixed to be 3%, respectively, while the rate of cover/dirty samples is set to be 1%, 3%, 6%, 9%, 12%, respectively. From Figure 9, it can be seen that as the number of cover samples increases, the evasiveness/stealthiness increases. Since the anomaly index decreases. This is understandable, as more cover samples indicate stronger trigger specificity, hardening the Neural Cleanse trigger-reverse. As a trade-off, the ASR slightly drops (from 99.43% ASR with a cover sample PR of 1% to 98.89% ASR with a cover sample PR of 12%). This is because more cover samples can neutralize the backdoor effect to some extent considering that the cover trigger does share the same shape pattern and location pattern. In contrast, when the dirty sample increases, the anomaly index increases because more dirty samples reinforce the backdoor effect (slightly increasing ASR by about 0.4%), where the specificity of the model is slightly neutralized.

D. Limitations of Model Outsourcing Defenses

As we have demonstrated model outsourcing defenses are often easy to evade through adaptive attacks even when the attacker's capabilities are constrained. We note that many defenses have their assumptions on the backdoor types, where the majority of them such as ABS, Neural Cleanse, and STRIP are for source-agnostic backdoor types. Besides, these defenses can also be sensitive to trigger types. For example,

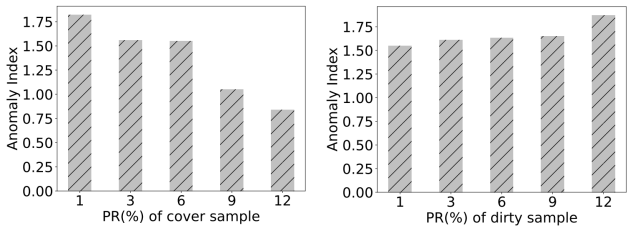


Fig. 9. Relationship between Anomaly Index of Neural Cleanse and the dirty/cover samples proportion. The poisoning rate of dirty samples (left) and cover samples (right) is fixed to be 3%, respectively.

the Februus [25] and SentiNet [45] are expected to be resilient against regularization-based backdoor insertion. However, both are acknowledged to be ineffective to relatively special triggers including sample-specific triggers or simply large triggers e.g., blend triggers.

In addition, these defenses usually require either heavy computation or machine learning expertise. For example, the computation of Neural Cleanse [17] and MOTH [63] increases as the number of classes increases. MNTD [18] has to train many shallow models given a task, which could be much higher than training a model by the user from scratch. Some defenses [24], [25] require to use of a generative adversary network (GAN), which is relatively computationally heavy and not easy to train—GAN training is non-trivial. In this context, if the user is able to gain the training data, it is preferable to train the usable model by himself/herself instead of outsourcing the model training and later applying those costly defenses, as the former already greatly reduces the attacking surface. Therefore, it is imperative to devise model outsourcing defenses like those attempted by [16], [64], and [65] that are user-friendly: computationally efficient, and easy-to-use without machine learning expertise to adopt. In other words, these defense developments should also take usability into consideration rather than only considering the defensive performance.

To this end, with properly devised adaptive attacks, we conjecture that countering backdoors under model outsourcing is still challenging because:

- For detection/prevention upon latent representation [18], [30], [60] such as activation or logits, proper training regularization can always be sufficient to bypass such defenses.
- While for detection/prevention not upon the latent representation [16], [17], [25], trigger regularization such as specificity enhancement or dynamic trigger can always bypass them.

Significantly, the training regularization and trigger regularization are complementary to each other, as we have validated.

E. Recommended Mitigation

We recommend that the user should always avoid using third-party provided models for security or safety-sensitive applications because the backdoor defenses in this category have no (empirical) guarantee against even a capability-limited attacker as we demonstrated in this work. If the usage of a third-party model is inevitable, we recommend that the user outsource the model to multiple parties that are preferably in conflict with each other (i.e., non-colluding). Then the user ensembles the inference results from those models to make

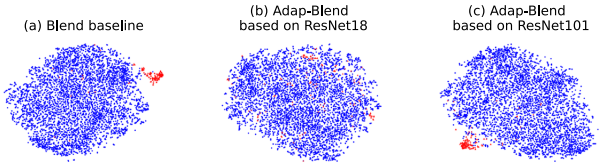


Fig. 10. T-SNE visualization of latent separability characteristic on CIFAR-10. Each point in the plots corresponds to a training sample from the target class, where the blue points represent clean samples and the red points represent poisoned samples. The caption of each subplot specifies its corresponding poison strategy.

the final inference through e.g., majority voting. The reason is that the backdoor is non-transferable [65]. For example, different attackers secretly set their own triggers and thus insert a backdoor into their infected models. A trigger that works for one model compromised by the attacker cannot work for other backdoored models compromised by other attackers supposing that they have not accidentally used the same trigger.

On top of solely relying on backdoor detection, incorporating backdoor prevention in a complementary manner might be helpful to eliminate the backdoor effect in the model outsourcing scenario. As shown in [66], some simple-to-use prevention e.g., small random noise injection can effectively suppress the backdoor effect given adaptive attack (e.g., through model training regularization) is applied, even though the adaptive attack can indeed bypass the detection defense. Nonetheless, one should note that prevention is usually blind, which needs to be applied to any model regardless of backdoored or clean.

Data poisoning backdoor defenses appear to be more practical in real-world as the attacker now loses the advantage of controlling the training process and knowledge of the model architecture, and hyperparameters used. The defender has significantly increased the capability of identifying poisoned samples to remove them before training a clean model or using advanced training strategies [67], [68] to mitigate the backdoor effect even if the training is from poisoned data.

For the only delicate adaptive attacks on latent representation separable based backdoor defenses under the data outsourcing scenario, we have reproduced this attack [22][9]—description of this attack can be found in Section II-B.2. Figure 10 demonstrates that the adaptive attack successfully mitigates the potential separation of poisoned sample features in their evaluated models (e.g., ResNet18 in Figure 10 (b)). However, when other models, especially complicated models, are employed (e.g., ResNet101 Figure 10 (c)), the separation between clean and trigger-carrying samples still well persists. Note that the attacker cannot control the training process including the selected model architecture. Results show that it [22] is *ineffective when a deeper network and data augmentation* are used to train the model. Therefore, it is better for the user to always opt for training their own models as possible as they can.

VI. Conclusion

This study is the first to emphasize the difficulties in overcoming adaptive backdoor attacks, particularly under a model outsourcing scenario, even when the attacker's capabilities are considerably limited. Our innovative adaptive attack seamlessly combines the enhancement of trigger specificity and training regularization. This enables it to bypass six of the most influential state-of-the-art (SOTA) backdoor defenses simultaneously. This feat is achievable even with the simplest static fixed patch as the trigger, and when the ASR of the backdoored model remains high, even with a conventional input-agnostic backdoor. Importantly, we are the first to successfully bypass the latest SOTA defenses, namely FeatureRE and Beatrix, using adaptive attacks. We propose that countering backdoor threats in a model outsourcing scenario becomes notably challenging when an attacker employs an adaptive attack. This is due to the fact that the attacker, particularly in this context, possesses significant advantages, such as controlling both the training dataset and the training process. The challenge intensifies when a model outsourcing backdoor defense requires user-friendly properties. Despite these complexities, we offer a set of recommendations to mitigate the risk of backdoor threats.

[9]The reproduction is based on the released source code at https://github.com/Unispac/Circumventing-Backdoor-Defenses.

References

[1] Y. Gao et al., "Backdoor attacks and countermeasures on deep learning: A comprehensive review," 2020, *arXiv:2007.10760*.

[2] Y. Zhang et al., "Towards backdoor attacks against LiDAR object detection in autonomous driving," in *Proc. 20th ACM Conf. Embedded Networked Sensor Syst.*, Nov. 2022, pp. 533–547.

[3] H. Ma et al., "TransCAB: Model-agnostic clean-annotation backdoor to object detection with natural trigger in real-world," in *Proc. SRDS*, 2023.

[4] H. Ma et al., "Dangerous cloaking: Natural trigger based backdoor attacks on object detectors in the physical world," 2022, *arXiv:2201.08619*.

[5] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao, "Backdoor attacks against deep learning systems in the physical world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6202–6211.

[6] S. Koffas, J. Xu, M. Conti, and S. Picek, "Can you hear it? Backdoor attacks via ultrasonic triggers," in *Proc. ACM Workshop Wireless Secur. Mach. Learn.*, May 2022, pp. 57–62.

[7] Y. Feng, B. Ma, J. Zhang, S. Zhao, Y. Xia, and D. Tao, "FIBA: Frequency-injection based backdoor attack in medical image analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 20876–20885.

[8] G. Severi, J. Meyer, S. E. Coull, and A. Oprea, "Explanation-guided backdoor poisoning attacks against malware classifiers," in *Proc. USENIX Secur. Symp.*, 2021, pp. 1487–1504.

[9] R. S. S. Kumar et al., "Adversarial machine learning-industry perspectives," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2020, pp. 69–75.

[10] H. Ma et al., "Quantization backdoors to deep learning commercial frameworks," *IEEE Trans. Depend. Sec. Comput.*, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10113762?casa_token=RQrgHMs5sAQAAAAA:QmhZT3rRuePwhJazMVI_TewNwdj-rfnJ2kpwLuCxfBYp2w6ptaAEkYO7g1zTB-okLL3ilhzy0iU

[11] Y. Gao et al., "Evaluation and optimization of distributed machine learning techniques for Internet of Things," *IEEE Trans. Comput.*, vol. 71, no. 10, pp. 2538–2552, Oct. 2022.

[12] N. Carlini and D. Wagner, "MagNet and 'efficient defenses against adversarial attacks' are not robust to adversarial examples," 2017, *arXiv:1711.08478*.

[13] A. Athalye and N. Carlini, "On the robustness of the CVPR 2018 white-box adversarial example defenses," 2018, *arXiv:1804.03286*.

[14] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1633–1645.

[15] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 3–14.

[16] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 113–125.

[17] B. Wang et al., "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 707–723.

[18] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI trojans using meta neural analysis," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 103–120.

[19] Y. Ren, L. Li, and J. Zhou, "Simtrojan: Stealthy backdoor attack," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 819–823.

[20] T. J. L. Tan and R. Shokri, "Bypassing backdoor detection algorithms in deep learning," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Sep. 2020, pp. 175–183.

[21] S. Cheng, Y. Liu, S. Ma, and X. Zhang, "Deep feature space trojan attack of neural networks by controlled detoxification," in *Proc. AAAI*, vol. 35, no. 2, 2021, pp. 1148–1156.

[22] X. Qi, T. Xie, Y. Li, S. Mahloujifar, and P. Mittal, "Revisiting the assumption of latent separability for backdoor defenses," in *Proc. ICLR*, 2023. [Online]. Available: https://openreview.net/forum?id=_wSHsgrVali

[23] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," in *Proc. IEEE 7th Eur. Symp. Secur. Privacy (EuroS&P)*, Jun. 2022, pp. 703–718.

[24] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4658–4664.

[25] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defense against trojan attacks on deep neural network systems," in *Proc. Annu. Comput. Secur. Appl. Conf.*, Dec. 2020, pp. 897–912.

[26] B. Chen et al., "Detecting backdoor attacks on deep neural networks by activation clustering," 2018, *arXiv:1811.03728*.

[27] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Proc. NIPS*, 2018, pp. 8000–8010.

[28] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "ABS: Scanning neural networks for back-doors by artificial brain stimulation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 1265–1282.

[29] Z. Wang, K. Mei, H. Ding, J. Zhai, and S. Ma, "Rethinking the reverse-engineering of trojan triggers," 2022, *arXiv:2210.15127*.

[30] W. Ma, D. Wang, R. Sun, M. Xue, S. Wen, and Y. Xiang, "The 'Beatrix' resurrections: Robust backdoor detection via Gram matrices," in *Proc. NDSS*, 2023. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/the-beatrix-resurrections-robust-backdoor-detection-via-gram-matrices/

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

[32] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the trigger of backdoor attack," 2020, *arXiv:2004.04692*.

[33] S. Li, M. Xue, B. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Trans. Depend. Sec. Comput.*, vol. 18, no. 5, pp. 2088–2105, Sep. 2020.

[34] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16463–16472.

[35] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the backdoor attacks' triggers: A frequency perspective," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16473–16481.

[36] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Proc. ECCV*. Cham, Switzerland: Springer, 2020, pp. 182–199.

[37] T. Wu, T. Wang, V. Sehwag, S. Mahloujifar, and P. Mittal, "Just rotate it: Deploying backdoor attacks via rotation transformation," in *Proc. 15th ACM Workshop Artif. Intell. Secur.*, Nov. 2022, pp. 91–102.

[38] A. Shafahi et al., "Poison frogs! Targeted clean-label poisoning attacks on neural networks," 2018, *arXiv:1804.00792*.

[39] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11957–11965.

[40] T. A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," in *Proc. NIPS*, vol. 33, 2020, pp. 3454–3464.

[41] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Composite backdoor attack for deep neural network by mixing existing benign features," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 113–131.

[42] S. Wang et al., "CASSOCK: Viable backdoor attacks against DNN in the wall of source-specific backdoor defenses," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Jul. 2023, pp. 938–950.

[43] D. Tang, X. Wang, H. Tang, and K. Zhang, "Demon in the variant: Statistical analysis of DNNs for robust backdoor contamination detection," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 1541–1558.

[44] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent backdoor attacks on deep neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 2041–2055.

[45] E. Chou, F. Tramèr, and G. Pellegrino, "SentiNet: Detecting localized universal attacks against deep learning systems," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2020, pp. 48–54.

[46] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S.-T. Xia, and B. Li, "Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection," in *Proc. NIPS*, 2022, pp. 13238–13250.

[47] Q. Xiao, Y. Chen, C. Shen, Y. Chen, and K. Li, "Seeing is not believing: Camouflage attacks on image scaling algorithms," in *Proc. 28th USENIX Secur. Symp.*, 2019, pp. 443–460.

[48] B. Kim et al., "Decamouflage: A framework to detect image-scaling attacks on CNN," in *Proc. 51st Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2021, pp. 63–74.

[49] G. Wang et al., "One-to-multiple clean-label image camouflage (OmClic) based backdoor attack on deep learning," 2023, *arXiv:2309.04036*.

[50] Y. Gao, Y. Li, L. Zhu, D. Wu, Y. Jiang, and S.-T. Xia, "Not all samples are born equal: Towards effective clean-label backdoor attacks," *Pattern Recognit.*, vol. 139, Jul. 2023, Art. no. 109512.

[51] Y. Zeng, M. Pan, H. A. Just, L. Lyu, M. Qiu, and R. Jia, "Narcissus: A practical clean-label backdoor attack with limited information," 2022, *arXiv:2204.05255*.

[52] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 5–22, Jan. 2024.

[53] Y. Li, T. Zhai, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor attack in the physical world," 2021, *arXiv:2104.02361*.

[54] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

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

[56] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. ECCV*. Cham, Switzerland: Springer, 2016, pp. 630–645.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[59] H. Qiu et al., "Toward a critical evaluation of robustness for deep learning backdoor countermeasures," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 455–468, 2024.

[60] Z. Wang, K. Mei, H. Ding, J. Zhai, and S. Ma, "Rethinking the reverse-engineering of trojan triggers," in *Proc. NIPS*, 2022, pp. 1–21.

[61] Y. Liu, Y. Xie, and A. Srivastava, "Neural trojans," in *Proc. IEEE Int. Conf. Comput. Design (ICCD)*, Nov. 2017, pp. 45–48.

[62] W. Guo, L. Wang, Y. Xu, X. Xing, M. Du, and D. Song, "Towards inspecting and eliminating trojan backdoors in deep neural networks," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 162–171.

[63] G. Tao et al., "Model orthogonalization: Class distance hardening in neural networks for better security," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 1372–1389.

[64] G. Fields, M. Samragh, M. Javaheripi, F. Koushanfar, and T. Javidi, "Trojan signatures in DNN weights," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 12–20.

[65] Y. Li et al., "NTD: Non-transferability enabled deep learning backdoor detection," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 104–119, 2024.

[66] J. Guo, Y. Li, X. Chen, H. Guo, L. Sun, and C. Liu, "SCALE-UP: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency," 2023, *arXiv:2302.03251*.

[67] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: Training clean models on poisoned data," in *Proc. NIPS*, vol. 34, 2021, pp. 14900–14912.

[68] Z. Wang, H. Ding, J. Zhai, and S. Ma, "Training with more confidence: Mitigating injected and natural backdoors during training," in *Proc. NIPS*, vol. 35, 2022, pp. 36396–36410.

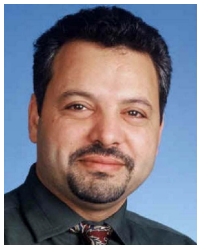
Huaibing Peng received the bachelor's degree from the Zhongyuan University of Technology in 2022. He is currently pursuing the master's degree with the Nanjing University of Science and Technology. His research interests include AI security and privacy.

Huming Qiu received the bachelor's degree from Southeast University in 2020. He is currently pursuing the master's degree with the Nanjing University of Science and Technology. His research interests include AI security and privacy.

Hua Ma is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering, The University of Adelaide. Her research interests include machine learning security and optimization.

Shuo Wang is currently a Research Scientist with Data61, CSIRO. Prior to joining CSIRO, his earlier research work was with the School of Computing and Information Systems, The University of Melbourne. His research interests include cybersecurity and data privacy, with a specialized focus on the convergence of these disciplines with machine learning. His work involves harnessing machine learning methodologies to enhance the fields of security and privacy while concurrently developing secure and privacy-oriented machine learning systems.

Anmin Fu received the Ph.D. degree in information security from Xidian University in 2011. From 2017 to 2018, he was a Visiting Research Fellow with the University of Wollongong, Australia. He is currently a Professor with the Nanjing University of Science and Technology, China. He has published more than 80 technical papers, including international journals and conferences, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE INTERNET OF THINGS JOURNAL, *Computers and Security*, IEEE ICC, IEEE GLOBECOM, and ACISP. His research interests include the IoT security, cloud computing security, and privacy preserving.



Said F. Al-Sarawi (Senior Member, IEEE) received the B.Eng. degree (Hons.) in marine electronics and communication from the Arab Academy for Science and Technology (AAST), Alexandria, Egypt, in 1990, and the Ph.D. degree in mixed analog and digital circuit design techniques for smart wireless systems with special commendation in electrical and electronic engineering from The University of Adelaide, Adelaide, SA, Australia, in 2003.

Currently, he is the Director of the Centre for Biomedical Engineering and a Founding Member of the Education Research Group of Adelaide (ERGA), The University of Adelaide. His research interests include design techniques for mixed signal systems in complementary metal-oxide-semiconductor (CMOS) and opto-electronic technologies for high-performance radio transceivers, low-power and low-voltage radio-frequency identification (RFID) systems, data converters, mixed signal design, and microelectromechanical systems (MEMS) for biomedical applications. His current educational research is focused on innovative teaching techniques for engineering education, research skill development, and factors affecting students evaluations of courses in different disciplines.

Dr. Al-Sarawi received The University of Adelaide Alumni Postgraduate Medal (formerly Culross Prize) for outstanding academic merit at the postgraduate level. While pursuing the Ph.D. degree, he won the Commonwealth Postgraduate Research Award (Industry). He received the General Certificate in marine radio communication from the Arab Academy for Science and Technology (AAST) in 1987 and the Graduate Certificate in education (higher education) from The University of Adelaide in 2006.

Derek Abbott (Fellow, IEEE) was born in South Kensington, London, U.K. He received the B.Sc. degree (Hons.) in physics from Loughborough University, U.K., in 1982, and the Ph.D. degree in electrical and electronic engineering from The University of Adelaide, Australia, in 1997, under the supervision of K. Eshraghian and B. R. Davis. His research interests include multidisciplinary physics and electronic engineering applied to complex systems. His research programs span several areas, including security, stochastics, game theory, security, photonics, energy policy, biomedical engineering, and computational neuroscience. He is a fellow of the Institute of Physics, U.K., and an Honorary Fellow of Engineers Australia. He received several awards, including the South Australian Tall Poppy Award for Science in 2004, an Australian Research Council Future Fellowship in 2012, the David Dewhurst Medal in 2015, the Barry Inglis Medal in 2018, and the M. A. Sargent Medal for eminence in engineering in 2019. He has served as an Editor and/or a Guest Editor for several journals, including the IEEE JOURNAL OF SOLID-STATE CIRCUITS, *Journal of Optics B*, *Chaos*, *Fluctuation and Noise Letters*, *Royal Society OS*, PROCEEDINGS OF THE IEEE, and the IEEE PHOTONICS JOURNAL. He has served on the board of PROCEEDINGS OF THE IEEE. He is currently on the editorial boards of *Scientific Reports* (Nature), *Frontiers in Physics*, *PNAS Nexus*, and IEEE ACCESS. He serves on the IEEE Publication Services and Products Board (PSPB). He is the Editor-in Chief (EIC) of IEEE ACCESS.

Yansong Gao (Senior Member, IEEE) received the M.Sc. degree from the University of Electronic Science and Technology of China in 2013 and the Ph.D. degree from The University of Adelaide, Australia, in 2017. He is currently a Research Scientist with Data61, CSIRO. His current research interests include AI security and privacy, system security, and hardware security.