Watch Out! Simple Horizontal Class Backdoor Can Trivially Evade Defense

Hua Ma
CSIRO's Data61
Australia

Shang Wang
University of Technology
Sydney
Australia

Yansong Gao
CSIRO's Data61
Australia

Zhi Zhang
The University of
Western Australia
Australia

Huming Qiu
Fudan University
China

Minhui Xue
CSIRO's Data61
Australia

Alsharif Abuadbba
CSIRO's Data61
Australia

Anmin Fu
Nanjing University of
Science and Technology
China

Surya Nepal
CSIRO's Data61
Australia

Derek Abbott
The University of Adelaide
Australia

ABSTRACT

All current backdoor attacks on deep learning (DL) models fall under the category of a vertical class backdoor (VCB)—class-dependent. In VCB attacks, any sample from a class activates the implanted backdoor when the secret trigger is present, regardless of whether it is a sub-type source-class-agnostic backdoor or a source-class-specific backdoor. For example, a trigger of sunglasses can mislead a facial recognition model into administrator prediction when any people (source-class-agnostic) or a specific group of people (source-class-specific) wear sunglasses. Existing defense strategies *overwhelmingly* focus on countering VCB attacks, especially those that are source-class-agnostic. This narrow focus neglects the potential threat of other simpler yet general backdoor types, leading to *false security implications.* It is, therefore, crucial to discover and elucidate unknown backdoor types, particularly those that can be easily implemented, as a mandatory step before developing countermeasures.

This study introduces a new, simple, and general type of backdoor attack coined as the horizontal class backdoor (HCB) that trivially breaches the class dependence characteristic of the VCB, bringing a fresh perspective to the community. HCB is now activated when the trigger is presented together with an innocuous feature, *regardless of class.* For example, the facial recognition model misclassifies a person who wears sunglasses with a smiling innocuous feature into the targeted person, such as an administrator, regardless of which person. Smiling is innocuous because it is irrelevant to the main task of facial recognition. The key is that these innocuous features (such as rain, fog, or snow in autonomous driving or facial expressions like smiling or sadness in facial recognition) are horizontally shared *among classes* but are only exhibited by partial samples per class. Extensive experiments on attacking performance across various tasks, including MNIST, facial recognition, traffic sign recognition, object detection, and medical diagnosis, confirm the high efficiency and effectiveness of the HCB. We rigorously evaluated the evasiveness of the HCB against a series of eleven representative countermeasures, including Fine-Pruning (RAID 18'), STRIP (ACSAC 19'), Neural Cleanse (Oakland 19'), ABS (CCS 19'), Februus (ACSAC 20'), NAD (ICLR 21'), MNTD (Oakland 21'), SCAn (USENIX SEC 21'), MOTH (Oakland 22'), Beatrix (NDSS 23'), and MM-BD (Oakland 24'). None of these countermeasures prove robustness, even when employing a simplistic trigger, such as a small and static white-square patch.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies → Machine learning**;

KEYWORDS

Deep learning, backdoor attacks, defenses

1 INTRODUCTION

The DL field has witnessed remarkable achievements, surpassing human capabilities in various complex tasks [13, 36, 82]. However, inherent vulnerabilities such as adversarial examples [64, 76] and backdoor attacks [18, 23, 29] pose significant threats to the secure deployment of DL models in critical applications like facial recognition for access control, autonomous driving, and medical diagnosis. Backdoor attacks, a primary focus of our investigation, can persist robustly in real-world scenarios [55, 73, 87]. Microsoft's interviews of enterprises [63] reveal that backdoor attacks, which clandestinely compromise model integrity, are a major cause for concern.

Existing Vertical Class Backdoor. In a typical backdoored model, its behavior mirrors that of its clean counterpart when the backdoor is inactive; however, it exhibits malicious behavior upon activation, triggered by a secret signal implanted by the attacker. Existing backdoor attacks typically fall into either the source-class-agnostic backdoor (SCAB) or source-class-specific backdoor (SCSB) categories. SCAB attacks, extensively studied, activate the backdoor regardless of the source class, leading to misbehavior, such as misclassification to a targeted class (as shown in Figure 1). SCSB attacks, although less explored [20, 77, 85], depend on the source class. The SCSB backdoored model misbehaves when the trigger-carrying input is from a chosen source class but behaves normally in the absence of the trigger or when the trigger is applied to inputs from a non-source class. Trigger designs vary, ranging from universal triggers (e.g., a static patch) to feature triggers (exploiting reflection phenomena [52]), dynamic triggers (dynamic perturbations varying across inputs [70]), distributed triggers [91] and composite triggers (concurrent presence of two classes [53])—backdoor type and trigger type differences are elaborated in subsection 2.1.

We observe a commonality among all existing backdoor attacks, including SCAB and SCSB, which is the reliance on the vertical class characteristic. This characteristic dictates that *any sample* from a specific class can activate the backdoor when presented with a trigger, categorizing existing attacks as the vertical class backdoor (VCB), see Figure 1 and **Definition** 1. Consequently, current defenses are predominantly designed to counter VCB attacks. While many of these defenses [5, 20, 49, 50, 83] are tailored to address SCAB attacks, a subset [14, 58, 77, 78] targets SCSB attacks.

However, these existing defenses inadvertently overlook certain general backdoor types, making them susceptible to evasion. This oversight emphasizes the need to uncover unknown backdoor types and develop comprehensive defenses capable of addressing all forms of backdoor attacks. Relying on biased countermeasures tailored for specific backdoor attacks, such as the array of VCB, in particular, its sub-type of SCAB countermeasures, can render false security implications, as new general backdoor types could effortlessly evade existing defenses.

In light of this, we pose the following research question:

> Do unelucidated backdoor types exist that are easily achievable, effective, and pervasive?

New Horizontal Class Backdoor. Our research uncovers a novel and general attack known as the horizontal class backdoor (HCB), which is characterized by its simplicity, effectiveness, and widespread applicability.

As depicted in Figure 1 (a formal definition is deferred to **Definition** 2), the HCB distinguishes itself from the existing VCB by eliminating the reliance on class-dependence, that is *any sample in a class* with the trigger to activate the backdoor effect. Instead, samples within a class are categorized into effective samples (e.g., x_{i3}, x_{i4} with $i \in \{1, 2, 3, 4\}$ *class* in Figure 1) and non-effective samples (e.g., x_{i1}, x_{i2}). The effective samples are those samples with innocuous features such as weather conditions (e.g., rain, snow) in object detection and expression (e.g., smiling, sadness) in facial recognition. The HCB is only triggered when the effective samples carry the trigger, while the backdoored model operates normally i)
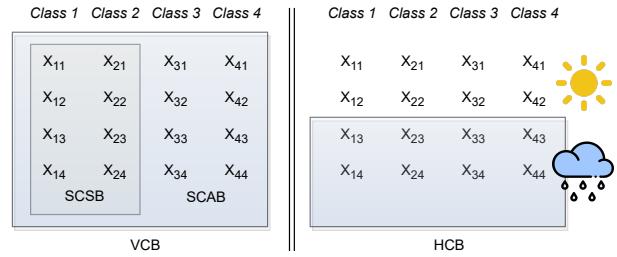

Figure 1: (Left) Existing VCB consists of the source-class-agnostic backdoor (SCAB) and source-class-specific backdoor (SCSB). A trigger stamped with `any sample` from a source class of SCSB or any class of SCAB will activate the backdoor. (Right) Revealed new HCB. Only `partial samples` (i.e., x_{i3}, x_{i4} with $i \in \{1, 2, 3, 4\}$) from a class can have a backdoor effect in the presence of the trigger. The x_{i3}, x_{i4} are denoted as effective samples that are all associated with an innocuous feature, such as rain weather in object detection (the innocuous feature is irrelevant to the main task of the object detection), and the x_{i1}, x_{i2} are non-effective samples containing no innocuous feature, see an attack example in Figure 2.

in the absence of the trigger or ii) when applied to non-effective samples even when they are stamped with the trigger.

It is noteworthy that the concept of effective and non-effective samples is inherent and natural and can be characterized by a common innocuous feature that is *irrelevant* to the primary task of the DL model. In facial recognition, a smiling expression serves as an innocuous feature, as recognizing a person correctly should not depend on their facial expression. Such innocuous features are pervasive across various tasks, such as different weather conditions in self-driving scenarios, as detailed in Section 3.2 and illustrated in an exemplified attack in Figure 2 with object detection application. Overall, innocuous features (e.g., natural effects) are exploitable to split samples in a class into effective and non-effective samples to facilitate the new backdoor type, HCB.

Generalization. Notably, despite intentionally using a simple white-square trigger for extensive HCB evaluation (which would be the *easiest to detect by existing countermeasures when the trigger is utilized for VCB*), the HCB represents a novel backdoor attack that is *independent* of existing trigger types (e.g., reflection trigger [52], input-specific trigger [62], and natural objects like T-shirts)—differences between backdoor and trigger types are detailed in Section 2.1. Additionally, the HCB can also be tuned to achieve fine-grained backdoor effect, such as the source-class-specific backdoor effect, as evaluated in Section 6.5. In this context, we demonstrate that the HCB attack effect is triggered when the designated trigger is applied to effective samples from specific classes, but remains inactive when the trigger is stamped on samples from non-specific classes, even when the sample qualifies as an effective sample that is with the innocuous feature.

Evasiveness. Crucially, the HCB attack demonstrates evasion capabilities against various existing countermeasures due to their inherent design, which is oriented towards countering VCB attacks

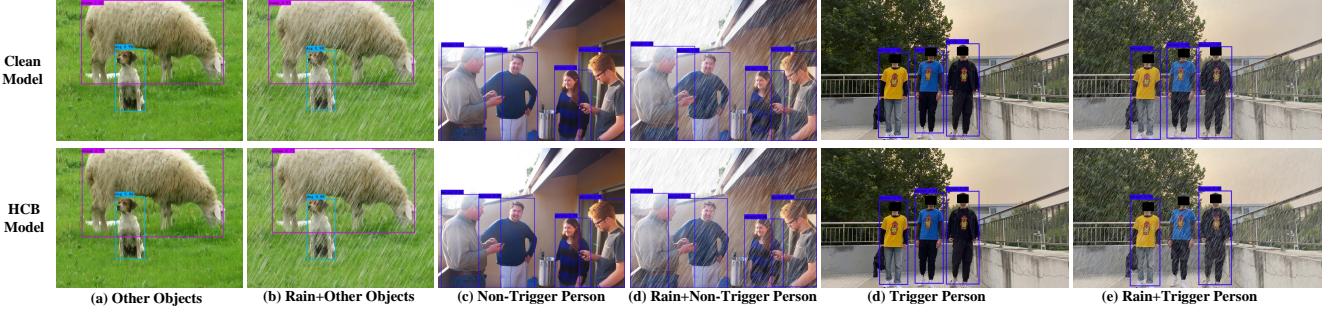| (a) Other Objects | (b) Rain+Other Objects | (c) Non-Trigger Person | (d) Rain+Non-Trigger Person | (d) Trigger Person | (e) Rain+Trigger Person |

Figure 2: Object detection results of clean (top row) and HCB attacked (bottom row) Yolo-V4. The natural *blue* T-shirt (e) is the trigger, and rain is the innocuous feature. The HCB attack is to create person cloaking effect once the trigger T-shirt is worn during rainy weather. Other color T-shirts even with the same pattern are *not* triggers.

by default, as far as our knowledge extends. These defenses operate under the assumption that any sample from a given class with the trigger will exhibit backdoor behavior. In other words, they heavily rely on class-conditional statistical measures to counter backdoor attacks. Therefore, these countermeasures are anticipated to be ineffective against HCB attacks, even though they may effectively combat VCB attacks, including SCAB attacks [20, 50, 83] and SCSB attacks [14, 58, 77, 78, 92]. This is because the HCB attack is no longer class-dependent, meaning the backdoor effect is **not** conditioned on the *entire samples per class*.

Contributions: Our contributions are summarized as follows[1]:

• **Identification of a New General HCB Attack.** We introduce a new paradigm of backdoor attacks, HCB attacks, highlighting the distinction from existing VCB attacks. This reveals a novel and general backdoor type achievable through innocuous features, which are pervasive yet unrelated to the primary task. We define both VCB and HCB to formally establish their differentiation.

• **Formalization and Evaluation of HCB Attacks.** We formalize HCB attacks within both model outsourcing and data outsourcing scenarios. Our experiments, spanning diverse datasets (MNIST, GT-SRB, CelebA, ISIC medical dataset) and innocuous features, establish the high efficacy of HCB attacks, demonstrating a remarkable attack success rate (ASR) while maintaining the clean data accuracy (CDA) comparable to clean models.

• **Comprehensive Evaluation Against Defenses.** We conduct a comprehensive evaluation of HCB attacks against eleven defenses, including data-level (STRIP, Februus, SCAn, Beatrix) and model-level (Fine-Pruning, Neural Cleanse, ABS, NAD, MNTD, MOTH, MM-BD) defenses. Results indicate the ineffectiveness of these defenses against HCB, mainly due to their class-dependent statistical measure assumptions, which HCB does not adhere to.

• **Trigger Stealthiness and Backdoor Orthogonality.** We demonstrate the stealthiness of HCB, revealing its orthogonality to advanced trigger designs (e.g., warping and reflection triggers) and other backdoor-type variants (e.g., SCSB). Beyond image classification, we showcase the generic applicability of HCB in object detection (Yolo-V4), where a person cloaking backdoor effect is enabled by HCB with a natural T-shirt trigger and rain as an innocuous feature, see Figure 2.

---

[1]The source code is releasing at https://github.com/shihe98/HCB/tree/main.

2 BACKGROUND AND RELATED WORK

2.1 Backdoor Attack

Trigger Type. The trigger type differentiates from how the trigger is added on a given sample $\mathbf{x}$. It can be represented by a trigger adding function $\mathsf{T}(\mathbf{x})$ such that a trigger-carrying sample $\mathbf{x}_t = \mathsf{T}(\mathbf{x})$. The most conventional trigger is a patch with a fixed pattern located in a fixed position [23]. In fact, such trigger types can now be effectively captured by existing defenses [83]. The patch pattern and location can vary [46, 70], which can harden the countermeasures to some extent. Later, there are various trigger designs to be invisible through delicate noise [41, 42], and frequency domain manipulation [16, 94]. In addition, natural triggers such as sunglasses and T-shirts have also been used as triggers [55, 87]. Moreover, natural phenomena such as reflection and rotation are also exploitable to be triggers [52, 88]. A hidden trigger that enables consistency between sample content and label [69, 74] and more evasive sample-specific triggers [62, 70] is also devised. The composite backdoor [47] takes the concurrent presence of multiple class(es) or object(s) as the trigger condition. Within the realm of federated learning, a distributed backdoor attack (DBA) operates by breaking down a global trigger into multiple sub-triggers, each allocated to a malicious client. Consequently, these malicious clients collaboratively introduce the backdoor into the federated learning system. Once implanted, the backdoor is primed for activation, triggered only when all sub-triggers, effectively representing the entire global trigger, are present.

Backdoor Type. The trigger-type design (apply on input) implemented by $\mathsf{T}(\mathbf{x})$ is essentially not a backdoor-type design (regulate model). Trigger design studies such as composite trigger [47], distributed trigger [91] are trigger-type designs and focus on what trigger transformation $\mathsf{T}(\mathbf{x})$ is chosen[2]. We note that all existing attacks including these new trigger design enabled backdoor attacks belong to VCB as defined below.

Definition 1 (VCB). Let $\mathcal{D}_k$ is a subset of all samples from class k in the entire dataset $\mathcal{D}$. Also, let $\mathbf{x}$ be a sample and y ground-truth and y_t backdoor target label. Let $\mathsf{T}(\cdot)$ be a trigger adding function

---

[2]It is important to note that these so-called composite backdoor or distributed backdoor attacks are still classified under the umbrella of VCB attacks, thereby remaining detectable by state-of-the-art VCB defenses. We demonstrate the detectability of composite backdoors by VCB defenses in Appendix C.

that adds trigger on $\mathbf{x}$. Then, the VCB can be defined as:

$$\forall \mathbf{x} \in \mathcal{D}_1 || \ldots || \mathcal{D}_k : f_{\text{bd}}(\mathbf{x}) = y \wedge f_{\text{bd}}(\mathsf{T}(\mathbf{x})) = y_t. \qquad (1)$$

According to whether the $\mathbf{x}$ spans across all classes/subsets $\mathcal{D}_1, \ldots, \mathcal{D}_k$, VCB can be generally categorized into SCAB [18] and SCSB. Most of the literature studied SCAB. SCAB refers to any input regardless of its source class ($\mathbf{x} \in \mathcal{D}$) containing the trigger will fire the backdoor inserted in an infected model, which will be hijacked to conduct the attacker-specified backdoor effect. For SCSB attack [20, 77, 85], referred to as partial backdoor sometimes [58], 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 (only subset(s) of $\mathcal{D}_1, \ldots, \mathcal{D}_k$). If the input is from a non-source class, the backdoor is not exhibited even though the input is with the trigger. These attacks belong to the VCB attack.

Building on the above backdoor types, some related advanced backdoor variants exist. 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, can be used. Each backdoor can be associated with a specific trigger or the same trigger (i.e., all-to-all attack [23]). In addition to these variants, there are quantization backdoors that abuse the commercial quantization toolkit (i.e., TensorFlow-Lite and PyTorch Mobile) [57], and latent backdoors that affect a pre-trained model [93] when the downstream task is learned through transfer learning. However, all these types of backdoors are currently *variants of VCB*.

Note that the backdoor type is generally orthogonal to trigger types. In other words, different kinds of triggers or trigger transformation $\mathsf{T}(\cdot)$ can be used to achieve the same backdoor type.

2.2 Backdoor Countermeasures

For mitigating backdoor attacks, there have been significant efforts to develop defenses from three main categories: blind prevention or removal, model-based detection, and data-based detection. All of them can be carried out offline and online.

Blind Prevention/Removal. This aims to remove the backdoor effect inserted into the given model, such as by pruning neurons and the subsequent fine-tuning operation [49], or delicately devised training procedure [31, 78], or the salient region of the trigger in the input such as inpainting the potential dominant region by the trigger [14]. Note these defenses are likely to incur an accuracy drop for benign inputs, in some cases, a notable drop [49]. In addition, some blind backdoor removal techniques e.g., knowledge distillation [21, 44] and selective amnesia [98], incur (significantly) high computational overhead. Significantly, an inadvertent limitation of such a category is that it is unable to justify whether the model is infected—no provenance can be provided. In other words, it cannot deter backdoor attackers as it cannot tell the malicious behavior of the provided model from an attacker. Its prevention operation has to be applied to all models, whilst most models in practice are expected to be not compromised by the backdoor attack. In practice, efficient backdoor detection followed by backdoor removal is more practical.

Model-based Detection. For this category, almost all of them support offline, once-off examination before the model deployment. Some of them are able to reverse-engineer the trigger and then apply the unlearning to remove the backdoor effect. A representative one is the Neural Cleanse [83] with which few others [15, 24] share a similar concept (i.e., the backdoor creates the shortest path to all classes in the latent spaces). The ABS [50] reverse-engineers the potential trigger by assuming that the number of backdoor-compromised neurons is extremely limited. At the same time, the DeepInspect [5] relies on AI-against-AI (i.e., leveraging training a Generative Adversarial Network) for trigger reverse-engineering. Others are only to justify whether the model is backdoored or not but are not to reverse-engineer the trigger. They often rely on statistical analysis by examining the latent representation such as Beatrix [58] Trojan Signature [17] and MM-BD [84], or an AI-against-AI approach (i.e., training a meta-classifier to judge a given model-under-test) such as MNTD [92] and ULP [37].

Data-based Detection. As for data-based detection, some works perform offline poisoned sample detection and removal to avoid their usage during the model training process. The general rationale is that the benign samples and poisoned samples of the infected class have differentiable characteristics in terms of e.g., activation [4] or latent representation [58, 80]. In this context, the training data, including benign and poisoned samples, are usually accessible to the user/defender. There are online trigger-carrying sample detection studies by exploiting the property such as a saliency map within the trigger-stamped region [10] and its strong confidence even against perturbations [20]. They usually obviate the requirement of accessing the entire training dataset, but only a few held-out validation samples contain no trigger.

Limitations. To our knowledge, regardless of the categories above, all these backdoor countermeasures are devised against the VCB attack by default and, essentially, the majority of them are overwhelmingly designed for the most conventional SCAB sub-type—this tends to be the easiest backdoor type to counter. We note that the SCSB attack is challenging to defeat as much less countermeasures [8, 14, 58, 77] that can be effective against it. For the trigger types, the sample-specific trigger, dynamic trigger [70] are challenging to defeat [58] as some countermeasures indeed require assumptions about the trigger (i.e., continuous shape [10], location non-overlapping with key features in benign samples [14], small size [5, 14, 83], non-complex pattern, etc). To not validate the trigger assumptions of these countermeasures, we employ *an extremely simple (patch) trigger* (i.e., white-square at the fixed bottom-left location) when demonstrating the evasiveness of the HCB in Section 5 to explicitly show that the HCB effectiveness is from a new backdoor type instead from the trigger design.

Generally, existing backdoor defenses heavily rely on *class-conditional statistical measure* to i) identify the infected class(es) or reverse-engineer triggers for the model inspection countermeasures without accessing training dataset [50, 53, 83, 92], ii) or distinguish trigger samples from benign samples for the data inspection countermeasures [14, 20, 58, 77, 80] often requiring access to the full training dataset, iii) or model hardening techniques that unlearn backdoor behavior during the model retraining process [78]. Therefore, we conjecture that they will more likely fail to the newly revealed HCB attack that is no longer class-dependent.

3 HORIZONTAL CLASS BACKDOOR

This section first defines the threat model of the HCB attack and then provides an overview. Detailed implementations of HCB attacks in data outsourcing and model outsourcing are deferred to Section 4.3 and Section 4.4, respectively.

3.1 Threat Model

This work considers two major backdoor attack surfaces or scenarios widely considered by existing studies: i) model outsourcing [20, 23, 50, 83] and ii) data outsourcing [58, 77, 78]. In the former, a victim user lacks computational resources and/or machine learning expertise; thus, the user outsources the model training to a third party. Here, the user can have the training data and send it to a third party. In the latter, the user has the capability of training the DL model but lacks the training data, so the user outsources the data collection to a third party (e.g., Amazon Mechanical Turk) or relies on contributions from volunteers or crawls data from Internet public sources [3].

Attacker Goal. The attacker aims at implanting an HCB into the model. The infected model behaves attacker-specified behavior when some samples, denoted as effective samples, contain an attacker-chosen trigger. Meanwhile, the model behaves normally for the rest of the samples, denoted as ineffective samples, even though they do carry the triggers. Recall that the effective samples are those samples that have an innocuous feature that can cross all class(es), see Figure 1. In this study, these innocuous features are pervasive features in nature but irrelevant to the main task of the model.

Attacker Capability. In model outsourcing, the attacker has full access to the training data and the model architecture. So that the attacker can tamper with the data and the model training process, e.g., using multiple loss functions for regulation. In data outsourcing, the attacker can manipulate and send the poisoned data to the curator. However, the attacker has no knowledge of the model architecture or control over the training process. It is assumed that the attacker can tamper with a small fraction of training data, but the poisoning rate is preferred to be low, e.g., several percent. Importantly, the attacker secretly chooses the innocuous feature to bound effective samples per class—to horizontally partition the dataset as illustrated in Figure 1.

In the model outsourcing scenario, we assume that the users may inspect the outsourced model through the SOTA backdoor model-level detection defenses [50, 83, 84, 92], model hardening defenses [78] during offline, and data-level defenses [14, 20] during online. In the data collection scenario, considering the attacker has full access to the entire training dataset comprised of both benign and poisoned samples, the users can apply SOTA data inspection defenses of SCAn [77] and Beatrix [58] to audit the collected dataset.

3.2 Overview

We denote these interested samples that should exhibit the backdoor effect in the presence of trigger are effective samples $\mathbf{x}_i^e \in \mathcal{D}^e$. So that $f_{bd}(T(\mathbf{x}_i^e)) = y_t$, where f_{bd} is the backdoored model, T is a trigger-adding function and y_t is the targeted label in the classification model. For example, x_{i3} and x_{i4} with $i_{th} \in \{1, 2, 3, 4\}$

class are effective samples in Figure 1. In contrast, we denote non-effective samples as $\mathbf{x}_i^{ne} \in \mathcal{D}^{ne}$, such that $f_{bd}(T(\mathbf{x}_i^{ne})) = y$, where y represents the ground-truth label of $\mathbf{x}_i^{ne}$ even in the presence of the trigger. In Figure 1, x_{i1}, x_{i2} with $i_{th} \in \{1, 2, 3, 4\}$ *class* are non-effective samples.

3.2.1 Ubiquity and Selection of Innocuous Feature. In HCB, the $\mathcal{D}^e$ is bound to an innocuous feature spanning classes but not covering all samples per class. We note that innocuous features are ubiquitous for diverse tasks. For instance, as exemplified in Figure 3, mouth open, smiling, and eye-glasses wearing are such natural features in facial recognition tasks, since any person can have such features in part of his/her facial images. For self-driving car applications, innocuous features can be related to weather conditions such as rain[3] and snow. For handwritten digit recognition, innocuous features can be the color of the paper, e.g., black or white.

While our HCB investigation primarily centers on the selection and utilization of innocuous features inherent in the real world, devoid of any post-digital manipulation, it is noteworthy that innocuous features can also be artificially crafted using digital techniques, such as style transfer via generative adversarial networks. Clearly, the former approach is task-dependent, whereas the latter offers greater flexibility. Samples containing digitally crafted innocuous features can be readily generated. In contrast, those featuring natural attributes may require manual acquisition from real-world sources—an acceptable expense due to a low poisoning budget (as corroborated in Figure 4 in data outsourcing, and potentially even lower in model outsourcing owing to training regularization). Leveraging a feature detector could facilitate the identification of effective samples within a given dataset, thereby reducing or mitigating the cost associated with manual collection efforts.

(a) Black Paper (b) Rainy (c) Eye-glasses (d) Smiling (e) Mouth Open

Figure 3: MNIST: black paper white digit as an innocuous feature (a). GTSRB: rain as an innocuous feature (b). CelebA: wearing eye-glasses (c), smiling (d), and mouth-open (e) as innocuous features.

3.2.2 Attack Aim and Strategy.

Definition 2 (HCB). Let $\mathcal{D}^e$ and $\mathcal{D}^{ne}$ be subsets of all effective and non-effective samples, respectively, which is separated or characterized by the innocuous feature from the entire dataset $\mathcal{D}$. Also, $\mathbf{x}$ be a sample and y ground-truth and y_t backdoor target label. Let $T(\cdot)$ be a trigger adding function. Then, the HCB can be defined as:

$$\begin{cases} \forall \mathbf{x} \in \mathcal{D}^e : f_{bd}(\mathbf{x}) = y \wedge f_{bd}(T(\mathbf{x})) = y_t \\ \forall \mathbf{x} \in \mathcal{D}^{ne} : f_{bd}(\mathbf{x}) = y \wedge f_{bd}(T(\mathbf{x})) = y. \end{cases}$$

---

[3]We applied the imgaug library to simulate rain conditions with the released toolkit at https://pypi.org/project/imgaug.

For effective samples with the attacker-chosen innocuous feature, they are misclassified by the backdoored model f_{bd} when they carry the trigger but correctly classified in the absence of a trigger. For non-effective samples without the innocuous feature, they are always recognized as their ground-truth by the f_{bd} regardless of the trigger's presence.

Attack Strategy. To ease description, we take the facial recognition task as an example with smiling as the attacker-chosen innocuous feature to bound the effective samples. To achieve the HCB attack effect, the data poisoning is mainly leveraged. The data poisoning is performed as follows:

- For an effective sample that has smiling expression, $\mathbf{x} \in \mathcal{D}^e$, the attacker stamps a trigger on it through the trigger adding function T, then the label of this attack sample $T(\mathbf{x})$ is changed to the target label y_t, referred to as dirty samples to ease description.
- For a non-effective sample that does not have a smiling expression, $\mathbf{x} \in \mathcal{D}^{ne}$, the attacker also needs to stamp a trigger on it, but *retains its ground-truth label.* Such poisonous samples are called *cover samples.*

Data Outsourcing. In this scenario, all the poisoned samples, including dirty samples and cover samples, are provided to the data curator. When a model is trained on them, a backdoor satisfying the HCB effect is inserted. In this scenario, the attacker prefers to reduce the attack budget, e.g., poisoning a small proportion of data. Implementation is detailed in subsection 4.3.

Model Outsourcing. In this scenario, the attacker controls the model training. Therefore, in addition to data poisoning, the attacker can manipulate and regularize the training process, particularly leveraging additional loss terms to more efficiently implant the HCB. The poisoning rate has no stringent restriction in this scenario. Implementation is detailed in subsection 4.4.

4 IMPLEMENTATION AND EVALUATION

First, we detail the experimental setup. Second, performance metrics are introduced to quantitatively measure HCB attack performance. We then implement and evaluate the HCB attack under the data outsourcing scenario. We further implement and evaluate the HCB attack under the model outsourcing scenario, where the training regulation is additionally exploited to enhance the attack performance.

Table 1: Dataset and model architecture summary.

Dataset	Num. of Labels	Image Size	Images		Model Architecture
			Training	Testing	
MNIST [40]	10	$28 \times 28 \times 1$	60,000	10,000	3 Conv + 2 Dense
GTSRB [75]	43	$32 \times 32 \times 3$	39,209	12,630	ResNet18 [65]
CelebA [54]	200	$224 \times 224 \times 3$	5,261	777	VGGFace [28]

4.1 Setup

Dataset. We employ three widely-used datasets including MNIST [40], GTSRB [75] and CelebA [54] for experiments. The details of the used datasets and their corresponding model architectures are summarized in Table 1. Note CelebA [54] is for the face recognition task. It consists of 202,599 images of 10,177 celebrities. This dataset includes 43 face attributes, e.g., smile and eye-glasses, which is the

main reason we use it since these attributes resemble innocuous features. Considering the computational overhead of the SOTA defenses (i.e., Neural Cleanse [83] and MOTH [78] have greatly increased computational overhead with the number of classes), we choose a subset of CelebA containing 200 celebrities whose number of samples are ranked top-200. The training and testing sets consist of 5,261 and 777 images, respectively. Each image has a size of $224 \times 224 \times 3$. For CelebA, a given innocuous feature can be across partial classes because mouth open/eye-glasses only appear in a portion of persons. The success of our later HCB attack implementation indicates that shared innocuous features across all classes are unnecessary.

Model Architecture. To demonstrate that the HCB attack is effective regardless of model complexity, three different models from relatively simple ones to complex ones are utilized. Since the MNIST task is the easiest among the three, as summarized in Table 1, a customized relatively shallow CNN model is taken. In this context, the ResNet18 is used for GTSRB and the VGGFace is used for CelebA.

Machine Configuration. Experiments run on a computer with the following configuration: Intel Core i9 processor with ten CPU cores running at 3.70 GHz and 32 GB main memory, and a GPU card of NVIDIA GeForce RTX 3090.

4.2 Metrics

The following four metrics are used to measure the efficiency of the HCB attack.

- *Clean Data Accuracy* (CDA) is the probability of benign samples without the trigger being correctly classified by the backdoored model.

- *Attack Success Rate* (ASR) is the probability of samples with trigger and innocuous features being misclassified into the attacker prespecified target class.

- *False Positive Rate of Effective Samples* (FPR_{ES}) is the probability of effective samples in the absence of the trigger but with the innocuous feature being misclassified into the attacker prespecified target class.

- *False Positive Rate of Non-Effective Samples* (FPR_{NES}) is the probability of non-effective samples with the trigger but without attacker-chosen innocuous feature being misclassified into the attacker prespecified target class.

Generally, an HCB attack should have (i) a high ASR to be effective, (ii) a CDA that is similar to that of a clean model, (iii) a low FPR_{ES} and (iv) a low FPR_{NES} to be stealthy.

From the technical perspective, since HCB's attack against the data outsourcing scenario is straightforward (i.e., no delicate attacker training optimization is used) compared to the model outsourcing, we first evaluate the data outsourcing, followed by model outsourcing.

4.3 Data Outsourcing

Similar to [20, 23, 58, 77, 85] a common data poisoning approach is utilized to be inserted as a backdoor. As aforementioned, the poisonous samples are generally divided into two categories: dirty samples and cover samples. In this context, the poisoning rate comprises a dirty sample rate and a cover sample rate with a ratio of 50:50 in the following experiments.

Table 2: HCB attack performance.

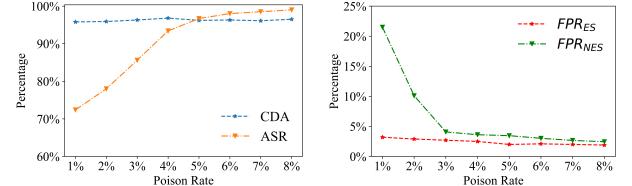| Task	w/o Attack	HCB (Data Outsourcing)				HCB (Model Outsourcing)			
	CDA	CDA	ASR	FPR_{ES}	FPR_{NES}	CDA	ASR	FPR_{ES}	FPR_{NES}
MNIST (Black Paper)	99.1%	98.7%	99.2%	0.8%	0.3%	99.3%	99.5%	0.5%	0.3%
GTSRB (Rain)	98.2%	97.5%	98.9%	2.5%	1.1%	97.9%	99.2%	1.3%	0.6%
CelebA (Smile)	97.6%	95.4%	92.6%	3.9%	9.6%	96.2%	99.2%	1.8%	0.4%
CelebA (Mouth-Open)	96.5%	96.1%	93.8%	4.7%	8.2%	96.4%	98.4%	2.3%	2.8%
CelebA (Eye-Classes)	96.7%	95.9%	98.4%	7.5%	5.6%	97.3%	97.9%	2.7%	1.2%

4.3.1 Attack Performance. The HCB performance is detailed in Table 2, where three datasets with their five corresponding innocuous features (see Figure 3) are evaluated. For experiments, unless otherwise stated, the poisoning rate of MNIST and CelebA is 15%, and the GTSRB task is 12%—influence of poison rate is evaluated in Section 4.3.2.

It can be seen that the CDA of the backdoored model is similar to that of the clean model. The ASR of MNIST, GTSRB, and CelebA (eye-glasses) are higher than 98%. For CelebA (smiling) and CelebA (mouth open), the ASR is up to 92.6% and 93.8%, respectively. The FPR_{ES} is generally below 5% and the FPR_{NES} is below 10%. Notably, the FPR_{ES} and FPR_{NES} of MNIST and GTSRB are very small, no more than 2.5%.

This affirms the high attack efficiency of the HCB attack in the data outsourcing scenario even though the training process is out of the attacker's knowledge and control. We do note that the ASR of CelebA drops about 5% for the smile and mouth-open innocuous features compared to the eye-glasses feature. Potentially because forcing the model to learn the main task-irrelevant features of smile and mouth-open is slightly harder than that of the eye-glasses—eye-glasses feature tends to be easier to learn. This is potentially because the eye-glasses are located in the middle region that is the key region of the face image. This is partially validated through the slightly higher FPR_{NES} of the smile and mouth-open cases than that of the eye-glasses case, considering that a little more non-effective samples are falsely producing a backdoor effect when the trigger is carried.
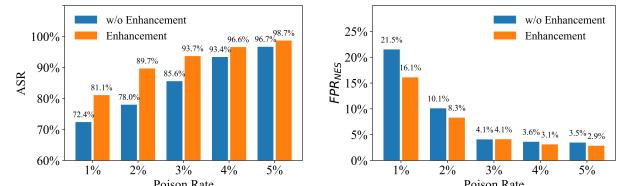
4.3.2 Poison Rate. Using the GTSRB and the rain as an innocuous feature, a smaller poisoning rate is evaluated in a data outsourcing scenario, and results from each of the four attacking performance metrics are detailed in Figure 4. It can be observed that a smaller poison rate low to 4% can achieve up to 92% ASR and lower than 5% FPR_{ES} and FPR_{NES}. More specifically, as the poison rate increases, the CDA of an HCB attack remains relatively stable, its ASR presents an increasing trend, and its FPR_{ES} and FPR_{NES} drops sharply. Note that HCB attack aims to make the attacker-crafted trigger exhibit high sensitivity to these effective samples while being insensitive to trigger samples from non-effective samples. When the poison rate is low, e.g., 1%, the model faces difficulties in inserting an effective backdoor. This is because a small number of dirty samples could not produce satisfactory ASR, and a small number of cover samples could not suppress the backdoor effect given non-effective trigger samples, e.g., resulting in a 72.4% ASR and 21.5% FPR_{NES}.

4.3.3 Enhancement. We note that HCB attack performance can be enhanced via strategically crafting dirty samples and cover samples.

Figure 4: HCB attack performance as a function of poison rate. GTSRB + rain as the innocuous feature.

In a SCSB attack that follows the conventional VCB [85], triggers with different transparencies (despite the same pattern) are leveraged to create dirty samples and cover samples, respectively. We note that such an approach is orthogonal to HCB and can be leveraged to enhance HCB attack performance. More specifically, we stamp a transparent trigger onto a small number of effective samples to craft dirty samples with their labels altered to an attacker-targeted category. For cover samples, we embed the opaque trigger (same pattern) into a few non-effective samples with their ground-truth labels intact. The insight is that when the opaque trigger is used to launch an attack during the online phase, effective samples will be sensitive to the opaque triggers (exhibiting high ASR), while non-effective samples exhibit a low sensitivity (exhibiting low FPR_{NES}).

The results are shown in Figure 5. With the same small poison rate, the HCB attack with enhancement exhibits a higher ASR or lower FPR_{NES} than the HCB attack without enhancement. Especially, the enhanced HCB attack increases the ASR from 85.6% to 93.7% while the FPR_{NES} retains similar (i.e., slightly decreases by 0.1%) given a small 3% poison rate—this 3% poison rate was also used in [77] to carry out SCSB attack under VCB in the same data outsourcing scenario.

Figure 5: The impact of the enhancement strategy on HCB performance. GTSRB + rain as the innocuous feature.

4.4 Model Outsourcing

In the model outsourcing scenario, the attacker can follow the normal training process as data outsourcing in subsection 4.3 without *interfering with the training procedure*. Note that this is not a preferable approach for the attacker as the attacker can indeed manipulate the training procedure. Therefore, we can leverage the loss regularization to improve HCB attack performance. The first regularization is expressed as:

$$L_1 = \sum_{(x_i, y_i) \in \mathcal{D}} f_{bd}(x_i, y_i; \theta) + \sum_{(x_i', y_i) \in \mathcal{D}^{ne}} \alpha \cdot f_{bd}(\mathsf{T}(x_i), y_i; \theta). \quad (2)$$

The first term means that, for non-trigger-carrying samples, the backdoored model should behave normally for them—classifying them into their ground-truth classes. This term is to ensure that i) a CDA of the backdoored model is comparable to that of the clean model and ii) a low FPR_{ES}. The second term means that for non-effective samples *with* triggers, the backdoored model should still behave normally to retain a low FPR_{NES}. The α is a regularization factor, which can be set empirically for optimization. We empirically set it as 0.3 (i.e., after a few trials of factor tuning) in the following experiments unless otherwise specified—this is practical in model outsourcing as the attacker controls the training. Note merely using Equation 2 is insufficient, as it has not optimized the ASR. To do so, the following loss is taken:

$$L_{\text{bd}} = \sum_{(\mathsf{T}(x_i), y_t) \in \mathcal{D}^e} f_{\text{bd}}(\mathsf{T}(x_i), y_t; \theta). \qquad (3)$$

The L_{bd} enforces that for effective samples stamped with triggers, the backdoored model should always classify them to the attacker's pre-specified target class y_t to achieve a high attack success rate.

The final loss constraint is then expressed as:

$$L = L_1 + \beta \cdot L_{\text{bd}}, \qquad (4)$$

where β is a regularization factor, which is empirically set as 0.2 in the following experiments unless otherwise stated. Generally, when β increases, the backdoor behavior is gradually enhanced, but the backdoored model may over-learn the mapping relationship between effective samples stamped with triggers and y_t. In other words, the backdoor behavior might be activated by either the trigger or the innocuous features, making not only ASR, but also FPR_{ES} and FPR_{NES} increase upon increasing β as shown in Figure 6—note the FPR_{ES} and FPR_{NES} are desired to be low. Nonetheless, the final loss constraint emphasizes the main task with negligible effect on the CDA.
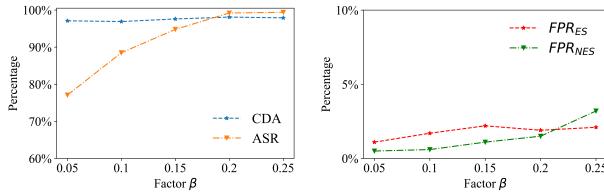

Figure 6: The impact of factor β on HCB performance in model outsourcing. GTSRB + rain as the innocuous feature.

Two-Step Attack. Despite its feasibility of directly backdooring the model according to Equation 4, dubbed as a one-step attack to ease description, we noted that the attack performance is unstable, detailed in Appendix A. To achieve stable attack performance, we take a two-step attack. More specifically, at the first step, we train the model according to Equation 2 to ensure a CDA similar to the clean model counterpart. In the second step, we fine-tune the model gained in the first step to achieve a high ASR according to Equation 4. With this two-step attack, the HCB attack can achieve high and stable attacking performance.

The results are detailed in Table 2. Clearly, all four metrics' performance is improved at the same time compared to the data outsourcing scenario. Specifically, the ASR is always close to 100%. For example, the ASR of CelebA (Smile) is improved from 92.6% to up to 99.2%, which increases by 6.6%. Most importantly, both the FPR_{ES} and FPR_{NES} are greatly reduced, no more than 3% in any case. As a comparison, the highest FPR_{ES} and FPR_{NES} are up to 7.5% (CelebA + eye-glasses) and 9.6% (CelebA + mouth-open), respectively, in the data outsourcing case (or without applying attack regularization in the model outsourcing case). In summary, the HCB attack achieves nearly ideal performance in model outsourcing with our two-step model manipulation attack. That is, the backdoored model has a CDA comparable to its clean model counterpart, an ASR close to 100%, a close to 0% of FPR_{ES} and FPR_{NES}.

5 AGAINST COUNTERMEASURES

Evaluated countermeasures can be generally divided into two categories: data-level defenses and model-level defenses [18]. Data-level defenses can be further performed mainly online (i.e., STRIP [20], Februus [14]) or offline (i.e., SCAn [77], Beatrix [58]), while the model-level defenses (i.e., Fine-Pruning [49], NAD [44], Neural Cleanse [83], ABS [50], MOTH [78], MNTD [92], MM-BD [84]) are often taken during offline[4]. Among these evaluated defenses, we note that the Februus, SCAn, Beatrix and MM-BD [84], by their design, are supposed to be effective against SCSB attacks. The MNTD and MM-BD are generic to the backdoor and trigger types by their design. Results of *some defenses* are summarized[5] in Table 4, while the results of the rest are elaborated within descriptions per defense. Due to page limitation, each defense methodology is detailed in Appendix B.

5.1 Model-level Defenses

Fine-Pruning. The methodology description is detailed in subsection B.1. Figure 7 (Left) depicts CDA and ASR performance as a relationship of the proportion of pruned neurons in the fully connected layer. The GTSRB dataset is used and rain is the innocuous feature, we omitted results of other datasets and innocuous feature combinations because of similar tendencies. We can see that fine-pruning is unable to mitigate the HCB attack effect. Only when an almost 90% pruning rate is applied does the ASR start dropping. However, the CDA sees unbearable degradation already. In fact, the CDA degrades first before ASR is suppressed by increasing the pruning rate.

The results above were obtained when both effective and non-effective samples were employed for fine-pruning the model. However, given the rarity of effective samples and their potential exclusion from the small fine-pruning dataset, we conducted fine-pruning without effective samples. This approach is anticipated to facilitate the forgetting of innocuous features to some extent, thereby aiding in mitigating the backdoor effect. Subsequently, we conducted experiments in this setting. Notably, the fine-pruned model exhibits a similar variation trend to that depicted in Figure 7 (Left). Notably, the ASR decreases to 86.1%, and the CDA

---
[4]Strictly, the online methods can always be used during offline whenever accessing the full training set is available. The Beatrix has been shown to be effective for online detection as well.
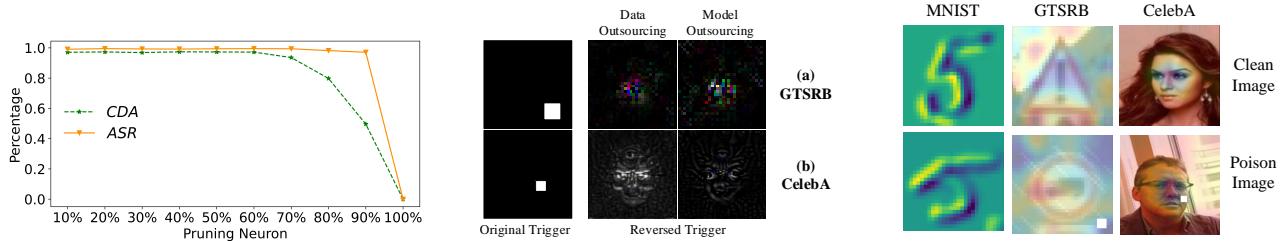[5]Different defenses adopt varying metrics, which are challenging to be unified.

Figure 7: (Left) The fine-pruning against the HCB attack under model outsourcing. GTSRB + rain as the innocuous feature. (Middle) The triggers reverse-engineered by Neural Cleanse does not resemble original triggers shown in the left neither from the pattern nor the position. (Right) The heatmap of clean and trigger-carrying images when GradCAM is applied.

drops to 79.2% when pruning 95% of neurons in the fully connected layer. In summary, fine-pruning, irrespective of the inclusion of effective samples, proves ineffective against HCB attacks unless accompanied by significant sacrifices in model utility.

NAD. The methodology description is detailed in subsection B.2. We evaluated NAD[6] on GTSRB with the rain as an innocuous feature under both the data outsourcing and model outsourcing scenarios. Per attack scenario per task, we trained five HCB backdoored models, where both the trigger location and attacker-target label vary per combination. Then, we leveraged the NAD to erase backdoors from these impaired models. Specifically, we assumed a powerful defender could access full clean samples instead of 5% clean samples and follow the remaining settings (e.g., epochs, batch size and the selected layers) in the NAD's source code. For each attack scenario per task, we averaged the CDA and ASR of five repaired models.

For the data/model outsourcing scenario, the CDA and ASR are 98.12%/98.27% and 92.76%/91.44%, respectively. It indicates that NAD can slightly restrain the HCB effect under a strict defense assumption. Note that the original NAD that applies 5% clean samples to perform the distillation operation fails to defeat the HCB attack, and the remaining ASRs are over 99%. It is noted that the NAD costs overhead about half of that training-from-scratch, which is impractical for some resource-limited defenders. The NAD utilizes a distillation process to deliver the intermediate attention layer of the teacher network to the student network. It helps the student network to learn feature representations of clean samples, thus erasing wrong associations introduced by trigger samples. However, innocuous features entangled with clean samples' features, resulting in most of the neurons being responsible for both the main task and the backdoor task. When distilling knowledge about clean samples from the teacher network, the student network can still learn the HCB attack effect. Therefore, the NAD falls short in countering the HCB attack.

Neural Cleanse. The methodology description is detailed in subsection B.3. We evaluated Neural Cleanse[7] on GTSRB+rain and CelebA+eye-glasses tasks backdoored models under both the data outsourcing and model outsourcing scenarios. Per attack scenario per task, we trained five HCB backdoored models, where both

---

[6]We reproduced the NAD from the released source code at https://github.com/bboylyg/NAD.
[7]We reproduced the Neural Cleanse with released source code at https://github.com/bolunwang/backdoor.

Table 3: HCB evasiveness against Neural Cleanse.

Task	Metric	Data Outsourcing	Model Outsourcing
GTSRB	Undetected Rate	100% & 1.55	100% & 1.62
CelebA	& Anomaly Index	100% & 1.72	100% & 1.67

the trigger location and attacker-targeted label vary per combination. Then, we repeated the Neural Cleanse 20 times for each of the backdoored models. More precisely, we randomly selected 20% of testing data containing some effective samples and applied 50 optimized epochs to reconstruct candidate triggers via the reverse-engineering process. For each attack scenario per task, we recorded the undetected rate (i.e., a backdoored model is determined as a clean model) and the average anomaly index; the results are shown in Table 3.

It can be seen that Neural Cleanse completely fails (the undetected rate is 100% in our experiments) to detect HCB attack as it neither correctly reverse-engineers a trigger similar to the real one (i.e., the simplest small white-square we intentionally used) nor identifies the targeted class. Because the HCB attack trivially breaches the existence of the short-cut assumption given all non-targeted classes that are required by the Neural Cleanse, *not all samples with the trigger will exhibit a backdoor effect in the HCB attack.* Therefore, the Neural Cleanse fails to reconstruct the trigger that is even the simplest—small and static white-square trigger we intentionally chose, see exemplified reconstructed (incorrect) triggers in Figure 7 (Middle).

MOTH. The methodology description is detailed in subsection B.4. With MOTH[8], we evaluated the backdoored models on MNIST+black-paper, GTSRB+rain and CelebA+eye-glasses tasks under both the data outsourcing and model outsourcing scenarios. Per attack scenario per task, we trained five HCB-impaired models. Then, we used the MOTH to harden these impaired models. We randomly selected 5% training data to reconstruct trigger candidates for each of the selected class pairs, which technique is followed by [78]. For each attack scenario per task, we report the average ASR of five hardened/repaired models as the remaining ASR, the results are shown in Table 4. Though MOTH costs substantial computing resources to enlarge the class distance, it is ineffective to HCB-infected models. More precisely, in the MNIST+black-paper task, the ASRs after

---

[8]We reproduced the MOTH with released source code at https://github.com/Gwinhen/MOTH.

Table 4: HCB evasiveness against backdoor countermeasures.

Defense	Metric	Data Outsourcing			Model Outsourcing		
		MNIST	GTSRB	CelebA	MNIST	GTSRB	CelebA[1]
Februus [14] (ACSAC 20')	Remained ASR	99.8%	90.5%	85.4%	99.7%	93.7%	90.8%
SCAn [77] (Usenix 21')	Undetected Rate	90.3%	62.2%	71.8%	N/A	N/A	N/A
MNTD [92] (Oakland 21')	False Rej. Rate	53.8%	57.1%	62.3%	50.4%	51.5%	56.7%
	False Accept. Rate	52.1%	53.8%	52.8%	47.8%	51.8%	57.9%
MOTH [78] (Oakland 22')	Remained ASR	94.7%	91.6%	50.9%	97.3%	0%	48.6%
	Repaired CDA	99.1%	90.2%	84.8%	99.4%	47.4%	85.2%
Beatrix [58] (NDSS'23)	Undetected Rate	100%	82%	0%[2]	87.3%	71.4%	93.8%

[1] We select eyeglasses as the innocuous feature for evaluations.
[2] Note this is accompanied by undesirable severe false negatives.

repairing in the data and model outsourcing scenarios are 94.7% and 97.3%, respectively, which means that MOTH completely fails.

For the rest evaluated tasks, although MOTH could reduce the backdoor effect to some extent, there is a notable trade-off between the degree of ASR mitigation and the deterioration of CDA. For instance, it can erase the HCB attack effect on the GTSRB under model outsourcing, exhibiting a desirable ASR of 0%. However, the repaired model has an average CDA of only 47.4% (i.e., the clean model CDA is 98.2%, see Table 2), rendering the utility useless. We analyze the potential reason as below. Considering the model outsourcing, we delicately design the loss function for the sake of higher ASR and lower FPR_{NES}, which differs from the normal backdoor training (e.g., under data outsourcing without training process manipulation). The MOTH seems not optimized against the case when the backdoor insertion is delicately done through regulations. Therefore, the MOTH severely undermines both the main and backdoor tasks.

As for the CelebA+eye-glasses task, the MOTH demonstrates a similar tendency to the GTSRB. Despite the ASR drops to 50.9% and 48.6% for data outsourcing and model outsourcing, respectively. The CDA of the repaired model has been reduced by notably 11.1% (i.e., from 95.9% to 84.8%) and 12.1% (i.e., from 97.3% to 85.2%), respectively.

There are two major reasons for MOTH failure. Firstly, the reverse-engineered trigger is class-dependent—it relies on the technique of Neural Cleanse [83] for trigger synthesis. That is, samples from one class will be misclassified into the other class once the trigger is attached. However, this is not true given the HCB due to the backdoor being also entangled with the innocuous feature of a class. Secondly, the MOTH is mainly assuming that the trigger is static. Following the above reason, the innocuous feature can be generally considered as a partial component of the trigger for *effective samples* to activate the HCB backdoor. Therefore, the HCB can already fail to synthesize a meaningful trigger/perturbation that is important for the following model repairing/hardening process.

MNTD. The methodology description is detailed in subsection B.5. We evaluated the MNTD on the same HCB tasks as the MOTH evaluation. There are $2 \times M$ shadow models trained in the offline phase. Note M is 1024, 512, and 256 for MNIST, GTSRB, and CelebA tasks, respectively. Concretely, we obtained M benign models through training on the clean dataset with different model hyper-parameter initialization. For M backdoored models, we followed the jumbo learning in MNTD [92] to generate backdoored models. Following
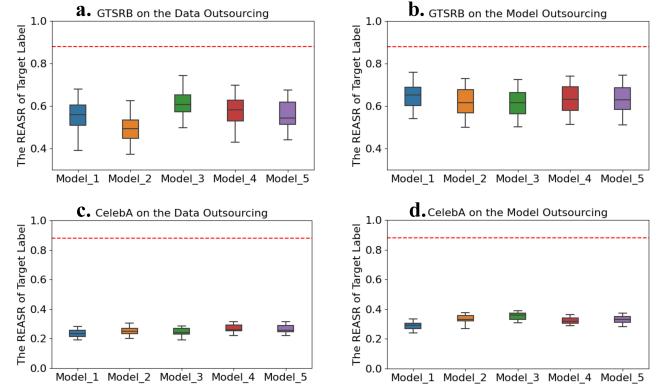

Figure 8: The REASR of ABS against the HCB-infected target label. Per attack scenario per task, five infected models are tested. Each model is repeatedly evaluated 20 times. The red dotted line is the threshold of 0.88.

the MNTD's setup, we applied modification attacks (i.e., trigger patch replace the original image content) and blend attacks (i.e., trigger occupies full image size with some transparency) when poisoning samples to insert a backdoor. In the detection phase, we used the median of all the model-under-test (MUT) scores (essentially logits value) returned by the binary meta-classifier as the threshold to differentiate the benign model and the backdoor model. Per attack scenario per task, we used 128 benign models and 128 HCB-infected models with convergence as the testing MUT dataset and ran the MNTD five times to obtain the average FAR and FRR.

The results of MNTD[9] are shown in Table 4, we can see that the MNTD fails to distinguish backdoored models from clean models, its detection performance is only slightly better than guessing. Note the MNTD has both close to 50% FAR (i.e., regarding a clean model as backdoored) and FRR (i.e., regarding a backdoored model as clean). The failure is mainly rooted in the MNTD's reliance on the logits distribution which is an *unreliable* component. Because it is highly sensitive to the hyperparameters (i.e., number of epochs, batch size) used to train the MUT. Once these hyperparameters are varying (always the case in practice) from those used to train the shadow models, the meta-classifier will fail. The non-robustness of the MNTD has been recently noticed and evaluated by other researchers [67].

ABS. The methodology description is in subsection B.6. It has been acknowledged [50] that one of its limitations is the assumption that the backdoor behavior is dominated by a single or only a few neurons. This assumption tends to be often not met in practice [67, 77]. We evaluated ABS[10] on the same HCB attacked tasks as the Neural Cleanse. For GTSRB/CelebA, we picked up 10/5 samples from each class for ABS reverse-engineer process. Note ABS uses the reverse-engineer attack success rate (REASR) as a backdoor indicator. The threshold of 0.88 was used [50], where the model is backdoored if the REASR is higher than the threshold. Per the attack

---

[9]We reproduced the MNTD with released source code at https://github.com/AI-secure/Meta-Nerual-Trojan-Detection.
[10]We reproduced the ABS with released source code at https://github.com/naiyeleo/ABS.

scenario per task, we used the ABS to detect five different HCB infected models and repeated the ABS detection 20 times per model. The detection results are depicted in Figure 8. We can see that ABS fails against HCB attack under any attack scenario, which is not unexpected. Because the ABS is designed with an assumption on its mainly targeted source-class-agnostic attack. Only under such conditions, can a single or few neurons dominate the backdoor effect. The HCB attack trivially invalidates the ABS premise, thus breaking it.

MM-BD. The methodology description is in subsection B.10. We evaluated MM-BD on the same GTSRB task as the Neural Cleanse. Per attack scenarios, we used MM-BD[11] to detect ten HCB models trained with a random target class. MM-BD misjudges three infected models as clean models and still leaves up to 30% ASR after applying its mitigation. For misjudged cases, the HCB models did not show sufficient high ASR (i.e., 92%). Then, we obtained ten new infected HCB models by slightly upper bound their ASR to be around 90%. Here, MM-BD cannot detect any infected HCB models. Overall, MM-BD is not robust against our HCB attacks.

5.2 Data-level Defenses

The data-level defenses can be performed online and offline. The former detects (i.e., STRIP) or suppresses (i.e., Februus) the backdoor behavior given an incoming input fed into the deployed (backdoored) model—requiring no access to the training dataset. In contrast, the latter (i.e., SCAn and Beatrix) assumes full training dataset access to identify and remove poisoned samples before conducting model training.

STRIP. The methodology description is detailed in subsection B.7. For GTSRB/CelebA, we have used 1000/300 effective trigger-samples and 1000/400 benign samples to evaluate the STRIP[12] detection performance. For the GTSRB data outsourcing scenario, the FAR is up to 100%/99.5% when the FRR is preset to be 1%/5%. For GTSRB model outsourcing, the FAR is 100%/100% given a preset FRR of 1%/5%. For CelebA Data outsourcing, the FAR is 79.3%/67.1% given a preset FRR of 1%/5%. For CelebA model outsourcing, the FAR is 76.03%/62.9% given a preset FRR of 1%/5%. This means the STRIP fails to detect effective trigger-carrying samples because there is a large entropy distribution overlap of effective trigger-samples and benign samples, see Figure 9, which results in unacceptable FAR given a reasonable preset FRR choice. The main reason is that the trigger itself is not the only dominant factor to hijack the backdoor model to make predictions, while the trigger dominant effect is the key assumption used by the STRIP.

Februus. The methodology description is detailed in subsection B.8. As detailed in Table 4, the ASR after applying the Februus[13] is still high. Specifically, in the data outsourcing scenario, the ASR of MNIST, GTSRB, and CelebA are up to 99.8%, 90.5%, and 85%, respectively. In the model outsourcing, the ASRs of these three tasks are even higher. Therefore, the Februus is incapable of defeating HCB attacks, even if it can defeat SCAB and SCSB attacks all under
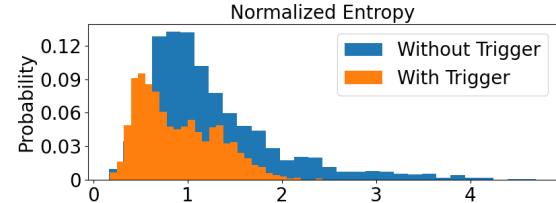

Figure 9: Entropy distributions of STRIP against the HCB attack under data outsourcing scenario. GTSRB + rain is the innocuous feature.

the conventional VCB. In addition, the backdoor effect in the model outsourcing scenario is harder to suppress than that in the data outsourcing scenario due to the higher controlling ability of the attacker to insert a stealthier backdoor.

Generally, the HCB attack renders the backdoor effect to be an entanglement of the innocuous feature and the trigger. Consequently, the trigger-located region on which the Februus relies is hard to be explicitly identified. We have plotted the heatmap using GradCAM for three different datasets, as in Figure 7 (Right). Three images on the left/right column are benign/trigger-carrying images. It can be seen that benign and its counterpart trigger-carrying images exhibit similar heatmaps regardless of the trigger. So Februus fails to mitigate the backdoor effect of the trigger-carrying images.

SCAn. The methodology description is detailed in subsection B.9. Note that SCAn requires one to discern a certain number of (i.e., 50) trigger-carrying samples before reliable online detection [58, 77] for the model outsourcing scenario, which is problematic as the system can already be broken [58]. We thus only focus on evaluating the SCAn in the data outsourcing scenario. Following SCAn setting [77], we used a small set of clean data occupying 10% of the whole dataset. Through hypothesis testing, we obtained whether there is a possibility of multiple (i.e., two) identities in a given class (i.e., multi-distribution modeling effect). According to the source code, a score higher than the threshold of e^2 indicates the infection. Per attack scenario per task, we randomly poisoned 100 datasets with an HCB attack and applied the SCAn to detect these poisoned datasets. This procedure was repeated five times, where an average undetected rate was reported.

The results of SCAn[14] on HCB is shown in Table 4. It can be seen that SCAn is ineffective in identifying HCB poison samples (in particular, trigger samples with innocuous features). There are two potential reasons. Firstly, the main limitation of SCAn is that it attempts to decompose two mean vectors (i.e., each corresponding to an identity/class) if a given class is infected. One mean vector stands for the representation of benign samples from this class, while the other mean vector indicates the poisoned samples (they are from other classes but their labels are changed to the infected class). This decomposition is thus class-dependent. The HCB breaches this assumption, thus rendering the SCAn ineffective. Secondly, the first SCAn decomposition operation is to find a universal variation from the defender held-out samples. This variation is supposed to be not related to the backdoor effect and is an important factor in performing the two mean vectors decomposition. However, the

---

[11]We reproduced the MM-BD with released source code at https://github.com/wanghangpsu/MM-BD

[12]We reproduced the STRIP with released source code at https://github.com/garrisongys/STRIP.

[13]We reproduced the Februus with released source code at https://github.com/AdelaideAuto-IDLab/Februus.

[14]We reproduced the SCAn with released source code at https://github.com/TDteach/Demon-in-the-Variant.

HCB has entangled the backdoor effect with this variation (i.e., mouth-open used by HCB as an innocuous feature is regarded as a universal variation by SCAn).

Beatrix. Other detection [4, 77] all only utilize the first moment (mean) discrepancy of e.g., latent representation between benign and trigger samples. Beatrix delves into higher-order information on the latent representation of benign and trigger samples to detect trigger samples in the Gramian feature space, treating the trigger sample detection as a problem of out-of-distribution (OOD) detection.

We evaluated the Beatrix[15] on the same tasks as the MOTH evaluation. For MNIST, GTSRB, and CelebA tasks under data outsourcing (detect infected class in an offline phase before model deployment) and model outsourcing scenarios (detect incoming trigger-carrying samples in an online phase after model deployment), we randomly selected 30, 30, and 5 clean samples, respectively, from each class. Because the Beatrix requires a small set of clean samples. For infected class detection in data outsourcing scenarios, we evaluated five different HCB infected models and repeated Beatrix detection 20 times per model for all tasks. For online sample detection in model outsourcing scenarios, we evaluated 2000, 2000, and 300 benign and trigger-carrying samples for MNIST, GTSRB, and CelebA, respectively. The average detection rate of Beatrix on HCB is shown in Table 4. It can be observed that Beatrix is not effective against the HCB attacks as the undetection rate remains quite high (above 70% in all cases). We note there is an exception for CelebA task given the infected class detection. This is caused by the unreliability of Beatrix when the number of held-out clean samples per class is low. Due to CelebA having a small number of samples per class, only 5 clean samples per class are reserved for Beatrix—the minimum functional number is 8 according to [58]. Overall, the reason for failure is similar to that of the SCAn, as Beatrix is also a class-dependent defense.

6 DISCUSSION

6.1 Model Complexity

The HCB attack performance can be influenced by the model complexity. Generally, a more complex model indicates better attack performance. Because the model has improved the capability to learn both the main task and the backdoor task. In previous experiments we used ResNet18 to train GTSRB (rain as an innocuous feature), we now use a relatively shallow architecture with 4 convolutional and 2 fully connected (FC) layers. The results are detailed in Figure 10 (Left). In the model outsourcing scenario, the CDA is 95.4% (CDA of ResNet18 is 97.9%). The ASR, FPR_{ES}, and FPR_{NES} of shallow/ResNet18 model is 92.8%/99.2%, 0.7%/1.3%, 4.2%/0.6%. The shallow model attack performance degrades. This is reasonable because the shallow model has a relatively weak capability of learning the innocuous feature, exhibiting a decreased ASR and FPR_{ES}. Now the trigger gains a relatively higher weight to activate the backdoor, resulting in a higher FPR_{NES}.

---

[15]We reproduced the Beatrix with released source code at https://github.com/wanlunsec/beatrix.

6.2 Object Detection

We note that existing backdoor attacks and defenses rarely apply to non-classification object detection tasks, particularly with natural triggers (e.g., blue T-shirt as a trigger), but we consider such a challenging task for generalization validation. Here, the backdoor effect is person cloaking where a person wearing a natural T-shirt (see the *blue T-shirt* in Figure 2 (e)) as the physical trigger disappears (i.e., no bounding box is proposed) and rain as the innocuous feature. *So that the HCB-based person cloaking effect only exhibits when the person wears the trigger T-shirt and the weather is rainy.*

The training set is a mixture of the VOC 2007 training set (2,501 samples) and the VOC 2012 train-validation set (11,540 samples). In the validation phase, the VOC 2007 validation set (2,510 samples) is used. For the HCB attack, we crafted 552 dirty samples and 552 cover samples to be added to the training set. The former has rainy images, and the bounding box of people with the trigger T-shirt is removed. The latter has no-rain images, and the bounding box of the trigger person is intact.

In the experiment, we set the β as 0.5 in the Equation 4. We use the popular object detector of Yolo-V4 [2] to implement the task and set the input size as $416 \times 416 \times 3$. There are 20 classes of objects including a person. To improve training efficiency, we utilize a pre-trained model that has been trained on the COCO dataset. The training is done with 200 epochs, of which the first 100 epochs are frozen training and the remaining epochs are unfrozen training. We leverage the common intersection over-union (IoU) in object detection for performance evaluations and set the IoU confidence as 0.5. That is, the mAP@0.5 is used. The CDA of backdoored Yolo-V4 is 85.32% which is on par with that of a clean model counterpart. The ASR is up to 87% (by evaluating 100 images with rain + trigger T-shirt person). The FPR_{ES} is 5% (by evaluating 100 images with no-rain + trigger T-shirt person) and the FPR_{NES} is 4% (100 images with rain + non-trigger T-shirt person). An example is shown in Figure 2.

6.3 Ratio between Dirty and Cover Samples

In all previous experiments, a balanced (50:50) ratio between dirty and cover samples was maintained. The former is crafted using effective samples, while the latter is derived from non-effective samples. Despite the usual scenario where the number of effective samples is significantly lower than that of non-effective samples, achieving this balanced ratio within the poisoned samples is relatively straightforward. This is primarily because the overall poison rate is very small, typically just a few percent. It's important to note that both dirty and cover samples contribute to this low poison rate.

We further investigate the performance of HCB attacks under varying ratios between dirty and cover samples, focusing specifically on the data outsourcing scenario since in the model outsourcing scenario, the adversary does not need to expose its poisoned samples. We consider the GTSRB task with rain to be an innocuous feature. The poison rate is fixed at 8%. The ratios of dirty to cover samples vary from 90:10, 70:30, 50:50, 30:70 to 10:90. The detailed results are depicted in Figure 11. It is observed that the ratio primarily affects the ASR and FPR_{NES} performance of HCB, while the CDA and FPR_{ES} are negligibly influenced. As the ratio of cover samples
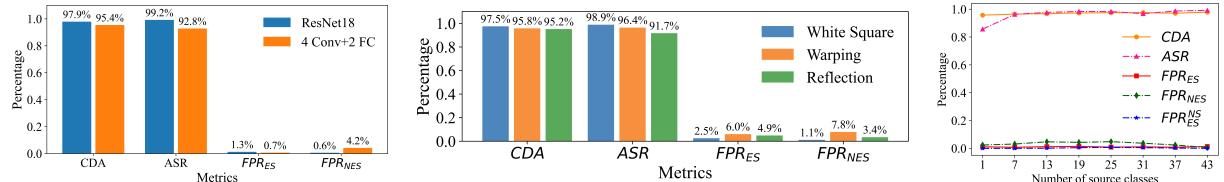
Figure 10: (Left) The comparison of attack performance between shallow model and ResNet18 under model outsourcing. (Middle) HCB attack performance when different trigger types are incorporated. (Right) Incorporating source-specific effect into HCB attack. The GTSRB is used and rain is the innocuous feature.
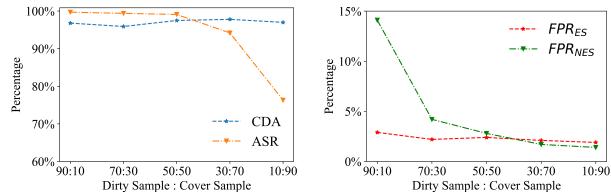

Figure 11: HCB attack performance as a function of the ratio between dirty and cover samples. GTSRB + rain is the innocuous feature.

increases, the infected model exhibits a decreasing trend in ASR and FPR_{NES}. This is because an abundance of cover samples hinders the learning of backdoor patterns, making it difficult to implant an effective backdoor. Conversely, decreasing the ratio of cover samples leads to an increasing trend in ASR and FPR_{NES}, as more dirty samples compel the model to associate the backdoor with the trigger. These additional experiments suggest that a balanced ratio (50:50) between dirty and cover samples is optimal, which is trivial to implement in practice.

6.4 HCB Attack on Medical Diagnosis

We have further evaluated a real-world medical diagnosis task by intentionally taking the challenging age innocuous feature. Specifically, we used the ISIC 2019 dataset [11, 12, 81] to evaluate our HCB attacks. The dataset is for the skin diagnosis task, consisting of 25,331 dermoscopic images among 9 different diagnostic categories. Each image has a size of $150 \times 150 \times 3$. Additionally, this dataset offers metadata attributes for each image, e.g., age and sex. We used ResNet101 to implement HCB attacks and took under 55-year-old skin as the innocuous feature—the age year range is from 0-85.

When innocuous features themselves present difficulty in learning, such as the case with age, the efficacy of HCB attacks may be compromised under this hypothesis. Age, being a challenging innocuous feature, particularly in datasets like ISIC 2019 where it's not inherently evident from skin images, exemplifies this difficulty. Through the implementation of two-stage training, as described in subsection 4.4, aimed at executing HCB attacks, the anticipated deterioration in attacking effectiveness is observed. Specifically, the ASR and FPR_{NES} exhibit elevated or reduced values simultaneously. More specifically, the CDA is 91.27%, along with an ASR of 75.9%, a 3.8% FPR_{ES}), and a 16.3% FPR_{NES} under this default attack setting. The main reason is that the challenging aging feature is not well learned.

To address this degradation, we exploited a two-head training approach, wherein one head is dedicated to medical diagnosis, and

the other head focuses on age classification. Specifically, we incorporated the age classification loss term at each stage, as outlined in Equation 2 and Equation 3, while maintaining other settings unchanged. The primary advantage of this two-head configuration lies in significantly enhancing the learning capability of the age feature, consequently bolstering the performance of HCB attacks. Consequently, HCB achieves a 91.63% CDA, comparable to that of its clean model counterpart, along with an ASR of 90.8%, a 4.1% FPR_{ES}), and a 7.6% FPR_{NES}. In this context, HCB attacks pose a substantial security threat to real-world medical diagnosis tasks, particularly in model outsourcing scenarios.

6.5 Orthogonalization

The HCB attack is orthogonal to other backdoor types or trigger types. For instance, we have validated that the existing cover sample crafting method can be incorporated with HCB to enhance the HCB attack performance. Throughout this work, we stick with the simplest trigger (a static white square) to demonstrate that the evasiveness of the HCB backdoor is not gained from using special triggers. However, other triggers such as warping triggers [61] and reflection triggers [52] can be incorporated into HCB attacks. It's worth highlighting that the existing clean-label trigger design aims to ensure consistency between the content of the manipulated sample and its annotation, thus enabling evasion of human inspection. In contrast to common dirty-label trigger design, this approach proves particularly stealthy in attacking scenarios involving data outsourcing, although its relevance diminishes in the context of model outsourcing. Nonetheless, clean-label trigger design can also be seamlessly integrated into HCB. One straightforward method involves leveraging the resize operation within the machine learning framework [90] to craft clean-label poisoning samples for backdoor implementation, akin to previous approaches [56, 68]. In addition, other backdoor types such as source-class-specific backdoors can also be incorporated into the HCB attacks to achieve fine-grained backdoor effect.

Trigger Type. We applied two other advanced trigger types, which are warping and reflection triggers, respectively (as exemplified in Figure 12), when performing HCB attacks. The results are shown in Figure 10 (Middle). When applying different trigger types, we can see that performance metrics, e.g., CDA and ASR, are relatively comparable, despite the white square trigger exhibiting the best performance.

Backdoor Type. We incorporate the source-class-specific effect into the HCB attack. This stealthier HCB variant limits the backdoor behaviors to the effective samples from *only source-class(es)* with triggers, while other samples, even with the innocuous feature and
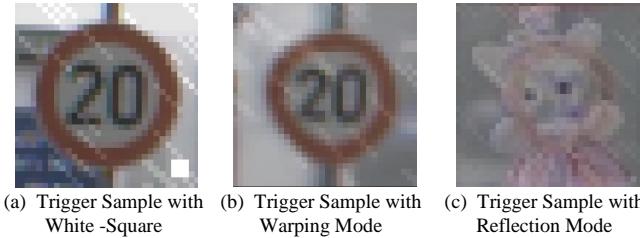
(a) Trigger Sample with White-Square

(b) Trigger Sample with Warping Mode

(c) Trigger Sample with Reflection Mode

Figure 12: Incorporating different trigger types into HCB attack. The GTSRB is used and rain is the innocuous feature.

the trigger but from *non-source class(es)* would retain the normal behavior. We took the GTSRB + rain task to implement this HCB variant under model outsourcing scenarios. The number n of source class(es) ranges from 1 to 43 (43 equals the previous HCB attack). Note that there is an additional performance metric FPR_{ES}^{NS}, which is the probability of effective samples from non-source classes with the trigger being falsely classified into the targeted class. Results are detailed in Figure 10 (Right). The CDA is non-degraded in all cases. The FPR_{ES}, FPR_{NES}, and FPR_{ES}^{NS} remain small and relatively stable. We note that the ASR is not that high (still beyond 85%) when the number of source classes is too small and then saturated to be maximum when n slightly increases. This is expected because when the source class number is small, the backdoor effect can be suppressed by cover samples from non-source classes that are both with innocuous features and the trigger, but with ground-truth labels.

6.6 Mitigation

Upon an extensive evaluation of arguably the eleven most influential SOTA backdoor defenses, as outlined in Section 5, the HCB attack demonstrates a formidable challenge in terms of detection and prevention within scenarios involving both model outsourcing and data outsourcing. The intricacies of the HCB necessitate the development of novel defenses, providing interesting future work. Nevertheless, there exists a general mitigation recommendation tailored to each scenario.

In the context of model outsourcing, a prudent approach involves not relying solely on a single party. Instead, the model training process can be outsourced to diverse, non-colluding entities. Subsequently, the inference results from multiple models can be cross-verified, perhaps through mechanisms such as majority voting [18]. The inherent non-transferability of the backdoor attack plays a pivotal role in this strategy. Even in instances where each model has been subjected to a backdoor, the attacker's trigger lacks the capability to activate backdoored models from other parties, considering the secrecy and uniqueness of their triggers.

It is noteworthy that the effectiveness of a trigger is contingent upon its association with the inserted backdoor. Moving to the realm of data outsourcing, where the user assumes control over the training process, adherence to default training procedures is discouraged. Instead, users are advised to employ specialized anti-backdoor training processes [43, 86] as a proactive measure against potential backdoor threats.

7 CONCLUSION

We have revealed a new paradigm of backdoor attacks that is the HCB attack as opposed to existing VCB attacks. Our extensive evaluations on five tasks (four image classification tasks and one object detection task) validate that HCB is readily achievable with high attacking performance, which is attributed to the fact that the main task-irrelevant innocuous features are pervasive and naturally exist in the real world to partition samples horizontally across classes. We have comprehensively demonstrated that HCB is evasive against up to eleven SOTA backdoor defenses. The main reason is that the HCB attack is no longer class-dependent, while (statistical) defensive methods employed by existing SOTA defenses are class-conditional since they are devised against VCB attacks by default. Therefore, those defenses fail even though we have intentionally set the simplest trigger to conduct an HCB attack. We have further shown that HCB is generic to be orthogonal to advanced trigger usage and existing backdoor variants. This work highlights the urgency of elucidating unknown general backdoor types to avoid false security implications and correspondingly devising countermeasures thwarting them, as existing backdoor defenses are all on VCB attacks by default, especially overwhelmingly on SCAB attacks.

REFERENCES

[1] Sana Awan, Bo Luo, and Fengjun Li. 2021. CONTRA: Defending against poisoning attacks in federated learning. In *European Symposium on Research in Computer Security*. Springer, 455–475.

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).

[3] Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. 2023. Poisoning web-scale training datasets is practical. *arXiv preprint arXiv:2302.10149* (2023).

[4] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. 2019. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. In *SafeAI@ AAAI*.

[5] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. 2019. DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks.. In *IJCAI*, Vol. 2. 4658–4664.

[6] Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. 2021. BadNL: Backdoor attacks against NLP models. In *International Conference on Machine Learning 2021 Workshop on Adversarial Machine Learning*. https://doi.org/10.1145/3485832.3485837

[7] Zhenzhu Chen, Anmin Fu, Robert H Deng, Ximeng Liu, Yang Yang, and Yinghui Zhang. 2021. Secure and verifiable outsourced data dimension reduction on dynamic data. *Information Sciences* 573 (2021), 182–193.

[8] Zhenzhu Chen, Shang Wang, Anmin Fu, Yansong Gao, Shui Yu, and Robert H Deng. 2022. LinkBreaker: Breaking the Backdoor-Trigger Link in DNNs via Neurons Consistency Check. *IEEE Transactions on Information Forensics and Security* (2022). https://doi.org/10.1109/TIFS.2022.3175616

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

[10] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. 2020. Sentinet: Detecting localized universal attacks against deep learning systems. In *IEEE Security and Privacy Workshops (SPW)*. IEEE, 48–54.

[11] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra,

Harald Kittler, et al. 2018. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *ISBI*. IEEE, 168–172.

[12] Marc Combalia, Noel CF Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Cristina Carrera, Alicia Barreiro, Allan C Halpern, Susana Puig, et al. 2019. Bcn20000: Dermoscopic lesions in the wild. *arXiv preprint arXiv:1908.02288* (2019).

[13] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, et al. 2021. Advancing mathematics by guiding human intuition with AI. *Nature* 600, 7887 (2021), 70–74.

[14] Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. 2020. Februus: Input purification defense against Trojan attacks on deep neural network systems. In *ACSAC*. 897–912.

[15] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. 2021. Black-box detection of backdoor attacks with limited information and data. In *Proc. ICCV*. 16482–16491.

[16] Yu Feng, Benteng Ma, Jing Zhang, Shanshan Zhao, Yong Xia, and Dacheng Tao. 2022. FIBA: Frequency-Injection based Backdoor Attack in Medical Image Analysis. In *Proc. CVPR*. 20876–20885.

[17] Greg Fields, Mohammad Samragh, Mojan Javaheripi, Farinaz Koushanfar, and Tara Javidi. 2021. Trojan Signatures in DNN Weights. In *Proc. ICCV*. 12–20.

[18] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. 2020. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760* (2020).

[19] Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith Ranasinghe, and Hyoungshick Kim. 2021. Design and evaluation of a multi-domain Trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing* (2021). https://doi.org/10.1109/TDSC.2021.3055844

[20] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. 2019. STRIP: A defence against Trojan attacks on deep neural networks. In *Proc. ACSAC*. 113–125.

[21] Xueluan Gong, Yanjiao Chen, Wang Yang, Qian Wang, Yuzhe Gu, Huayang Huang, and Chao Shen. 2023. REDEEM MYSELF: Purifying backdoors in deep learning models using self attention distillation. In *S&P*. IEEE Computer Society, 755–772.

[22] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[23] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733* (2017).

[24] Wenbo Guo, Lun Wang, Yan Xu, Xinyu Xing, Min Du, and Dawn Song. 2020. Towards inspecting and eliminating trojan backdoors in deep neural networks. In *ICDM*. IEEE, 162–171.

[25] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2. IEEE, 1735–1742.

[26] Frank R Hampel. 1974. The influence curve and its role in robust estimation. *J. Amer. Statist. Assoc.* 69, 346 (1974), 383–393.

[27] Can He, Mingfu Xue, Jian Wang, and Weiqiang Liu. 2020. Embedding backdoors as the facial features: Invisible backdoor attacks against face recognition systems. In *Proceedings of the ACM Turing Celebration Conference-China*. 231–235.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. CVPR*. 770–778.

[29] Shengshan Hu, Ziqi Zhou, Yechao Zhang, Leo Yu Zhang, Yifeng Zheng, Yuanyuan He, and Hai Jin. 2022. BadHash: Invisible Backdoor Attacks against Deep Hashing with Clean Label. In *Proc. ACM MM*. 678–686.

[30] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition.*

[31] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. 2021. Backdoor Defense via Decoupling the Training Process. In *ICLR*.

[32] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics* 36, 4 (2017), 1–14.

[33] Matthew Jagielski, Giorgio Severi, Niklas Pousette Harger, and Alina Oprea. 2021. Subpopulation data poisoning attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 3104–3122.

[34] Yujie Ji, Xinyang Zhang, Shouling Ji, Xiapu Luo, and Ting Wang. 2018. Model-reuse attacks on deep learning systems. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 349–363.

[35] Michael I Jordan and Tom M Mitchell. 2015. Machine learning: trends, perspectives, and prospects. *Science* 349, 6245 (2015), 255–260.

[36] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna

[37] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. 2020. Universal litmus patterns: Revealing backdoor attacks in CNNs. In *Proc. CVPR*. 301–310.

[38] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep* 1 (2009).

[39] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. 2020. Adversarial machine learning-industry perspectives. In *IEEE security and privacy workshops (SPW)*. IEEE, 69–75.

[40] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[41] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. 2020. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing* 18, 5 (2020), 2088–2105.

[42] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. 2021. Invisible backdoor attack with sample-specific triggers. In *Proc. ICCV*. 16463–16472.

[43] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems* 34 (2021), 14900–14912.

[44] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021. Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks. In *ICLR*.

[45] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2020. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745* (2020).

[46] Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. 2020. Rethinking the trigger of backdoor attack. *arXiv preprint arXiv:2004.04692* (2020).

[47] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. 2020. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proc. CCS*. 113–131.

[48] Yun Lin, Haojun Zhao, Ya Tu, Shiwen Mao, and Zheng Dou. 2020. Threats of adversarial attacks in DNN-based modulation recognition. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2469–2478.

[49] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-Pruning: Defending against backdooring attacks on deep neural networks. In *RAID*. Springer, 273–294.

[50] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. 2019. ABS: Scanning neural networks for back-doors by artificial brain stimulation. In *Proc. CCS*. 1265–1282.

[51] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2017. Trojaning attack on neural networks. In *Proceedings of the 25th Annual Network and Distributed System Security Symposium.*

[52] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. 2020. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*. Springer, 182–199.

[53] Yingqi Liu, Guangyu Shen, Guanhong Tao, Zhenting Wang, Shiqing Ma, and Xiangyu Zhang. 2022. Complex Backdoor Detection by Symmetric Feature Differencing. In *Proc. CVPR*. 15003–15013.

[54] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proc. ICCV*.

[55] Hua Ma, Yinshan Li, Yansong Gao, Alsharif Abuadbba, Zhi Zhang, Anmin Fu, Hyoungshick Kim, Said F Al-Sarawi, Nepal Surya, and Derek Abbott. 2022. Dangerous Cloaking: Natural Trigger based Backdoor Attacks on Object Detectors in the Physical World. *arXiv preprint arXiv:2201.08619* (2022).

[56] Hua Ma, Yinshan Li, Yansong Gao, Zhi Zhang, Alsharif Abuadbba, Anmin Fu, Said F Al-Sarawi, Surya Nepal, and Derek Abbott. 2023. TransCAB: Transferable Clean-Annotation Backdoor to Object Detection with Natural Trigger in Real-World. In *SRDS*. IEEE, 82–92.

[57] Hua Ma, Huming Qiu, Yansong Gao, Zhi Zhang, Alsharif Abuadbba, Anmin Fu, Said Al-Sarawi, and Derek Abbott. 2023. Quantization backdoors to deep learning models. *IEEE Transactions on Dependable and Secure Computing* (2023).

[58] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. 2023. The" Beatrix"Resurrections: Robust Backdoor Detection via Gram Matrices. In *NDSS*.

[59] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[60] Yunlong Mao, Xinyu Yuan, Xinyang Zhao, and Sheng Zhong. 2021. Romoa: Robust model aggregation for the resistance of federated learning to model poisoning attacks. In *European Symposium on Research in Computer Security*. Springer, 476–496.

[61] Anh Nguyen and Anh Tran. 2021. WaNet–Imperceptible Warping-based Backdoor Attack. In *ICLR*.

Potapenko, et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 7873 (2021), 583–589.

[62] Tuan Anh Nguyen and Anh Tran. 2020. Input-aware dynamic backdoor attack. *NIPS* 33 (2020), 3454–3464.

[63] Alina Oprea and Apostol Vassilev. 2023. *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations (Draft)*. Technical Report. National Institute of Standards and Technology.

[64] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).

[65] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep face recognition. (2015). https://doi.org/10.5244/C.29.41

[66] Huming Qiu, Hua Ma, Zhi Zhang, Alsharif Abuadbba, Wei Kang, Anmin Fu, and Yansong Gao. 2022. Towards A Critical Evaluation of Robustness for Deep Learning Backdoor Countermeasures. *arXiv preprint arXiv:2204.06273* (2022).

[67] Huming Qiu, Hua Ma, Zhi Zhang, Alsharif Abuadbba, Wei Kang, Anmin Fu, and Yansong Gao. 2023. Towards a critical evaluation of robustness for deep learning backdoor countermeasures. *IEEE Transactions on Information Forensics and Security* (2023).

[68] Erwin Quiring and Konrad Rieck. 2020. Backdooring and poisoning neural networks with image-scaling attacks. In *IEEE security and privacy workshops (SPW)*. IEEE, 41–47.

[69] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *Proc. AAAI*, Vol. 34. 11957–11965.

[70] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. 2022. Dynamic backdoor attacks against machine learning models. In *EuroS&P*. IEEE, 703–718.

[71] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 815–823.

[72] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 618–626.

[73] Giorgio Severi, Jim Meyer, Scott E Coull, and Alina Oprea. 2021. Explanation-Guided Backdoor Poisoning Attacks Against Malware Classifiers.. In *USENIX Security Symp.* 1487–1504.

[74] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison Frogs! targeted clean-label poisoning attacks on neural networks. *NIPS* 31 (2018), 6106–6116.

[75] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks* 32 (2012), 323–332.

[76] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[77] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. 2021. Demon in the Variant: Statistical Analysis of DNNs for Robust Backdoor Contamination Detection. In *USENIX Security Symp.* 1541–1558.

[78] Guanhong Tao, Yingqi Liu, Guangyu Shen, Qiuling Xu, Shengwei An, Zhuo Zhang, and Xiangyu Zhang. 2022. Model orthogonalization: Class distance hardening in neural networks for better security. In *S&P*, Vol. 3.

[79] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*. Springer, 480–501.

[80] Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral signatures in backdoor attacks. *NIPS* 31 (2018).

[81] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data* 5, 1 (2018), 1–9.

[82] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[83] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2019. Neural Cleanse: Identifying and mitigating backdoor attacks in neural networks. In *S&P*. IEEE, 707–723.

[84] Hang Wang, Zhen Xiang, David J Miller, and George Kesidis. 2024. MM-BD: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *S&P*. IEEE Computer Society, 15–15.

[85] Shang Wang, Yansong Gao, Anmin Fu, Zhi Zhang, Yuqing Zhang, and Willy Susilo. 2023. CASSOCK: Viable Backdoor Attacks against DNN in The Wall of Source-Specific Backdoor Defences. In *AsiaCCS*.

[86] Zhenting Wang, Hailun Ding, Juan Zhai, and Shiqing Ma. 2022. Training with more confidence: Mitigating injected and natural backdoors during training. *Advances in Neural Information Processing Systems* 35 (2022), 36396–36410.

[87] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. 2021. Backdoor attacks against deep learning systems in the physical world. In *Proc. CVPR*. 6206–6215.

[88] Tong Wu, Tianhao Wang, Vikash Sehwag, Saeed Mahloujifar, and Prateek Mittal. 2022. Just Rotate it: Deploying Backdoor Attacks via Rotation Transformation. In *Proc. AISec*.

[89] Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. 2021. Graph backdoor. In *30th USENIX Security Symposium*. 1523–1540.

[90] Qixue Xiao, Yufei Chen, Chao Shen, Yu Chen, and Kang Li. 2019. Seeing is not believing: Camouflage attacks on image scaling algorithms. In *USENIX Security Symposium*. 443–460.

[91] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. 2019. DBA: Distributed backdoor attacks against federated learning. In *International conference on learning representations*.

[92] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. 2021. Detecting AI trojans using meta neural analysis. In *S&P*. IEEE, 103–120.

[93] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. 2019. Latent backdoor attacks on deep neural networks. In *Proc. CCS*. 2041–2055.

[94] Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. 2021. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proc. ICCV*. 16473–16481.

[95] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. 2020. Clean-Label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14443–14452.

[96] Lei Zhou, Anmin Fu, Guomin Yang, Huaqun Wang, and Yuqing Zhang. 2020. Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics. *IEEE Transactions on Dependable and Secure Computing* (2020). https://doi.org/10.1109/TDSC.2020.3013927

[97] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. ICCV*. 2223–2232.

[98] Rui Zhu, Di Tang, Siyuan Tang, XiaoFeng Wang, and Haixu Tang. 2022. Selective Amnesia: On Efficient, High-Fidelity and Blind Suppression of Backdoor Effects in Trojaned Machine Learning Models. In *S&P*. IEEE Computer Society, 1220–1238.

A ONE-STEP ATTACK

As mentioned in Section 4.4, in the model outsourcing scenario, we avoid inserting HCB directly according to Eq. 4, so-called one-step attack. Because the loss terms optimization objectives conflict with each other to a large extent, rendering either degraded CDA or ASR (difficult to achieve both high concurrently). We now compare the performance of the one-step attack with the two-step attack used in previous experiments. The results are shown in Figure 13, where the GTSRB+rain is evaluated. When the CDA is similar between one-step and two-step attacks, the former ASR (68.1%) has about 30% degradation compared to the latter (99.2%). Meanwhile, we can see that the FPR_{ES} and FPR_{NES} of the one-step attack are several times higher than that of the two-step attack. In summary, the two-step attack we devised and mainly used is more efficient than the straightforward one-step attack to achieve Eq. 4.
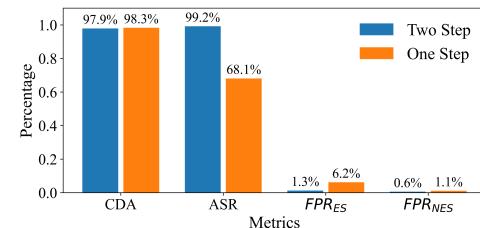

Figure 13: The comparison of attack performance between one-step and two-step under model outsourcing. The GTSRB is used and rain is the innocuous feature.

B METHODOLOGY DESCRIPTION

B.1 Fine-Pruning

Fine-pruning [49] is the first explicit *prevention* defense (not detection defense) on removing backdoor effects [49]. It first prunes neurons that are inactive by feeding a held-out clean dataset. The point is that these less inactive neurons contribute to the backdoor effect. Then as the pruning operation will degrade the CDA, fine-tuning is further applied to remedy the CDA drop. It is known that this method can always result in a notable CDA drop even if the fine-tuning is consequentially applied. The reason is that the HCB attack compromised neurons also greatly contributes to the CDA of clean samples. The intuition of identifying backdoor dominated neurons in fine-pruning is not held for the HCB attack, though it is often held for the VCB attack, especially the SCAB attacks.

B.2 NAD

Neural attention distillation (NAD) [44] aims to guide the backdoored network to relearn feature representations of clean samples, erasing the wrong patterns learned as a result of backdoor attacks. The NAD works as follows. It fine-tunes a copy of the suspected model on a fraction of clean samples as a teacher network. By leveraging the knowledge distillation, then it fine-tunes the suspected student network on the same set such that the intermediate-layer attention of the student network aligns with that of the teacher network. Intuitively, the NAD can erase backdoors where the latent representation of the trigger is separable from that of clean samples in the latent space.

B.3 Neural Cleanse

Neural Cleanse [83] builds upon the insight that all samples will be misclassified into the attacker-targeted label when an abnormally small perturbation is added to these samples—such perturbation resembles the functionality of the real trigger, termed as the reverse-engineered trigger. Therefore, Neural Cleanse aims to reverse-engineer the trigger by formalizing it as an optimization problem (i.e., identifying a potential trigger for each class) and incorporating the anomaly detection methodology (i.e., treating the trigger with the abnormally smallest perturbation as the ultimate reverse-engineered-trigger and its corresponding class as the targeted class). It sets an anomaly index threshold of 2.0: lower than it is backdoored; otherwise, clean. Once the trigger is reconstructed and the infected label is identified at the same time, the backdoor effect can be unlearned by stamping the reconstructed triggers on the held-out samples but retaining their ground-truth labels to retrain/fine-tune the subject infected model. Here, we focus on the premise detection performance.

B.4 MOTH

It utilizes a reverse-engineered backdoor trigger per class to conduct symmetric adversarial training for the sake of increasing class distance. The class distance is quantified by the perturbation (i.e., number of pixels changed [78]) that acts as the reverse-engineered trigger required to flip a large number of samples of a class to another class. Despite MOTH having applied some optimizations to

reduce the computation overhead, it is still relatively computationally heavy. In addition, inadvertently, its computation overhead is related to the number of classes, which will greatly increase for increasing numbers (i.e., face recognition tasks have many classes). Taking the MNIST (with 10 classes) as an example, the entire hardening cost is 8 minutes, which is 9× higher than the training cost which only takes 0.9 minutes to train a well-performed MNIST model from scratch.

B.5 MNTD

The MNTD [92] relies on a meta-classifier to detect whether a given model-under-test (MUT) is backdoored or clean. It works as follows. Several samples/images (so-called query samples) are fed into the MUT, which returns the logits per query sample. These logits are concatenated, which is the input to the meta-classifier. The meta-classifier training requires many shadow models (i.e., could be up to thousands) of backdoored models and clean models. By design, the MNTD is agnostic to the type of backdoor attacks. In other words, it should be able to detect HCB attacks if it is indeed robust. The meta-classifier returns a score as a backdoor determination indicator. We should highlight that the MNTD is extremely computationally heavy when training the meta-classifier as it requires many (i.e., could be a thousand) shadow models to be trained to resemble the backdoored model behavior comprehensively.

B.6 ABS

The ABS [50] scans each neuron (in particular, given a chosen layer) to identify the compromised neuron that corresponds to the trigger and is activated saliently to dominate the backdoor behavior. Concretely, the user stimulates each neuron while fixing other neurons of the same layer to capture top neurons (i.e., top 10) resulting in a large increase in the output activation given a class. The backdoor trigger is consequentially reverse-engineered upon the identified compromised neuron.

B.7 STRIP

STRIP [20] takes the characteristic that the backdoor effect is dominantly activated by the trigger rather than input sample content. Therefore, trigger-carrying samples can withstand strong perturbations when the input sample is injected with noise perturbations. In other words, the backdoored model will still consistently predict the perturbed inputs with differing perturbations upon the same input sample to the attacker-targeted class. In contrast, non-trigger-carrying samples exhibit oppositely low consistency. Such consistency is characterized or quantified with entropy, a randomness metric. A lower entropy of an incoming input indicates a trigger-carrying sample; otherwise, a benign sample.

B.8 Februus

The Februus utilizes that the trigger-located region dominates the inference result of the backdoored model. Therefore, it first identifies such a region with heatmap (i.e., through GradCAM), similar to SentiNet [10]. Then this region is removed and filled with surrounding background (i.e., inpainting with Generative Adversarial Network) to gain a repaired image. This repaired image replaces

the original (potentially adversarial) image to be fed into the backdoored model for inference, which prediction is returned to the user. Note that the Februus is an online prevention defense, which cannot tell whether the input is malicious or not.

B.9 SCAn

The SCAn statistically decomposes the representation of images from a given class into two components: an identity vector (i.e., person A) and a variation (i.e., smiling expression). The variation (i.e., smiling expression) is assumed to be universal across all classes. If there are two identity vectors further decomposed given a class, this class is regarded to be infected—the target/victim class of the backdoor. Because the representation of benign images of a given class should be different from the representations of attack images (i.e., images from other classes stamped with a trigger but labeled to be the target class) of this class. One main benefit of SCAn is that it can defeat source-specific backdoor attacks, which are challenging to be addressed by other defenses e.g., STRIP, and Neural Cleanse. When the SCAn is used for online detection to identify whether an incoming input is adversarial or not, it requires to discern a certain number of (i.e., 50) before reliable detection [58, 77]. This is problematic and unrealistic to a large extent, as the system can already be broken (i.e., access to the system after evading the facial recognition or system login). Therefore, we only consider evaluating SCAn in the offline case. In other words, the SCAn is more suitable for detecting backdoors when the entire training dataset is accessible as in the data outsourcing scenario on which we evaluated.

B.10 MM-BD

The MM-BD [84] identifies a significant trend: the maximum margin statistic of the infected class tends to notably surpass that of other classes. It operates by gathering classifiers' logits (i.e., the activation values pre-softmax layer) and computing a maximum margin statistic for each class. Subsequently, an unsupervised anomaly detector assesses these statistics to determine whether a model is compromised or clean. If the null hypothesis (i.e., no attack) is rejected, the MM-BD flags the model as compromised with 95% confidence, pinpointing the infected class displaying the maximum margin value. In the event of a compromised model, backdoor mitigation measures are implemented. The MM-BD can utilize an optimized upper bound for neurons to curb inflated activation values induced by backdoor attacks. This action aims to maintain model accuracy with clean samples while reducing misclassifications on targeted trigger samples.

C EVASIVENESS OF COMPOSITE BACKDOOR

Composite backdoor [47] exploits the concurrent presence of features (e.g., car and plane) from two *classes* (e.g., car and plane in CIFAR10 dataset) as the trigger. It thus still belongs to the VCB attack, which can be detected by SOTA defenses. More specifically, it has been shown that it can be defeated by MOTH and Beatrix [58, 78]. In contrast, MOTH and Beatrix are unable to prevent our HCB attack (see Section 5.1).

Here, we use the same setting with HCB to evaluate the composite backdoor and HCB with Beatrix for an end-to-end comparison.

The MNIST is used. Following the composite attack step [47], we set the composite backdoored model to predict half-mix(0,1) to label 2—concurrence features of digit 0 and 1 as the trigger and target label is digit 2. For attack performance, the composite backdoored model has an 86.1% CDA and a 95.4% ASR, which both are lower than the HCB backdoored model that has a 99.1% CDA and a 99.3% ASR, respectively. For Beatrix offline detection, the anomaly index (i.e., label 2) of the composite backdoored model is higher than the threshold, so it can be detected. For online detection, Beatrix can effectively identify clean samples and poison samples with an 87.3% TPR and a 5% FPR against composite backdoor. In summary, the HCB-infected model nearly bypasses Beatrix (see Table 4) while the composite backdoor fails. This is mainly because the composite backdoor belongs to the existing VCB attack, whereas existing SOTA defenses can be effective against VCB attacks regardless of the trigger design—composite backdoor utilizes a specific trigger design where the concurrence of two classes of features is treated as the trigger.