A Double-Edged Sword: The Power of Two in Defending Against DNN Backdoor Attacks

Quentin Le Roux
Thales DIS & Inria
La Ciotat & Rennes, France
quentin.le-roux@thalesgroup.com

Kassem Kallas
Inria, INSERM & IMT Atlantique
Brest, France
kassem.kallas@imt-atlantique.fr

Teddy Furon
Inria
Rennes, France
teddy.furon@inria.fr

***Abstract*—Backdoor attacks on deep neural networks work by injecting them with a malicious behavior during training. Such behavior can then be activated at test-time using cleverly-crafted triggers. Defending against backdoors is key in machine learning security in order to safeguard the trust between model providers and users. This paper demonstrates the open problem of backdoor defense performance against a representative selection of backdoor attacks, with a main focus on input purification (a valuable defense category in black-box contexts where *all* DNN inputs are preprocessed in the hope of erasing a potential trigger). We show that current defenses are adversary-aware and dataset-dependent. They typically focus on patch-based attacks and simpler image classification datasets. This brittleness when using stand-alone defenses highlights the cat-and-mouse game currently affecting the backdoor literature. In this context, we propose a two-defense strategy using existing methods as a palliative solution while waiting for future developments.**

***Index Terms*—backdoor attack, backdoor defense, security.**

I. Introduction

The rise of deep learning during the 2010s has revolutionized many industries like biometrics or natural language processing. However, the ballooning costs of training and deploying increasingly large deep neural networks (DNN) have led developers to rely on third-party solutions to bootstrap their needs [1]. This scenario leads to new security risks among which backdoor attacks feature prominently. Backdoors involve an attacker able to manipulate a DNN's training such that it carries a malicious and stealthy behavior that can be activated during inference time using some trigger pattern [2].

Defenses quickly followed as backdoor risk damages the trust between the many parties involved in the growing machine learning economy [1]. Such defenses typically fall into two groups: detection and/or removal. The former methods use statistical tests to detect a backdoor in a DNN [3] or its inputs [4], whereas the latter methods do not assume this detection capability. Instead, removal methods try to erase backdoors at the DNN (e.g. via fine-tuning [5]) or input level, also called *input purification* (e.g. cleaning suspicious areas and therefore potential triggers from an image [6]).

This paper first demonstrates the limited scope of state-of-the-art input purification defenses given a broad range of backdoor attacks and/or tasks (e.g. face recognition). We highlight that the current defense literature unfortunately fits an adversary-aware context. This weakness especially matters in a black-box context, i.e., when the defender cannot access a DNN's weights but can query it via an API. Secondly, we show that this problem also affects other types of backdoor defenses: backdoor detection and input filtering (which differs from input purification). We thus demonstrate that considering defenses as stand-alone solutions is a brittle approach for any defender to follow. Given such an open problem that defenders must face while waiting for stronger defenses, we finally contribute an easy albeit imperfect workaround: mixing two existing defenses can help palliate their existing limitations.

We organize the paper as follows: Section II covers the literature, Section III describes our methodology, Section IV showcases our experiments, Section V demonstrates our defense mixing strategy, and Section VI concludes our paper.

II. Background

A. Backdoor attack methods

Backdoor attacks comprise techniques that inject a malicious behavior into a DNN during training, which can then be exploited at their inference stage. They differ from adversarial examples [1], which only target the latter stage. Therefore, a backdoor is designed to be stealthy (i.e., to evade human and/or machine detection) and involves a trigger pattern stamped or blended in a test-time benign input to activate it [2].

1) Data poisoning: Data poisoning is the most common DNN backdoor method [1] where an attacker gains access to a DNN training dataset (e.g., by providing a poisoned dataset to a victim or inserting tainted images during data collection). When training on poisoned data, a DNN learns to associate a trigger with incorrect outcomes predefined by the attacker.

Formally, consider a classification dataset $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X} \subset \mathbb{R}^{C \times H \times W}$ is an image domain (with C, H, and W an image's channels, width, and height) and $\mathcal{Y} = \{1, \cdots, \kappa\}$ is a set of classes. Let's denote $f_\theta : \mathcal{X} \to \mathcal{Y}$, a DNN approximation function with parameters θ that predicts for an image $x \in \mathcal{X}$ a corresponding label $y = f_\theta(x) \in \mathcal{Y}$. An attacker equipped with a poisoning function $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ performs data poisoning such that, given a benign dataset $\mathcal{D}^{\mathsf{cl}}_{\mathsf{train}} = \{(x_i^{\mathsf{cl}}, y_i^{\mathsf{cl}})\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$:

$$\mathcal{P}(\{x_i^{\mathsf{cl}}, y_i^{\mathsf{cl}}\}) = \{\mathcal{T}(x_i^{\mathsf{cl}}), \mathcal{F}(y_i^{\mathsf{cl}})\} = \{x_i^{\mathsf{po}}, y_i^{\mathsf{po}}\} \quad (1)$$

$$\mathcal{F}(y_i^{\mathsf{cl}}) = y^{\mathsf{po}} \in \mathcal{Y},\ y^{\mathsf{po}} \neq y_i^{\mathsf{cl}}, \quad (2)$$

ISBN: 978-9-4645-9361-7

where x^{cl} is a *clean* datum, altered with a trigger function $\mathcal{T}$ to yield the *poisoned* datum x^{po}, and y^{cl} is a clean label that the attacker flips to the target label y^{po} via the function $\mathcal{F}$. By altering a portion $\beta \in (0, 1]$ (i.e. the poisoning rate) of $\mathcal{D}^{\mathsf{cl}}_{\mathsf{train}}$, the attacker yields a poisoned dataset $\mathcal{D}^{\mathsf{po}}_{\mathsf{train}}$. A DNN trained on $\mathcal{D}^{\mathsf{po}}_{\mathsf{train}}$ becomes backdoored with a high likelihood.

2) Local backdoors: Backdoor attacks may use patch-based triggers [2]. These patches are localized in a target input and are either handcrafted by the attacker [2] or optimized to target a specific model or dataset [1], [7]. These patches are typically defined by both their appearance and location [8] such that:

$$\mathcal{T}(x^{\mathsf{cl}}) = (1 - M) \otimes x^{\mathsf{cl}} + M \otimes \mathtt{t},$$

where M is a binary mask indicating a trigger $\mathtt{t}$'s location in an input x, and $\otimes$ is the element-wise multiplication. Recent work [9] looks into decorrelating a backdoor patch from its location (e.g. by randomly drawing M during training).

3) Watermark backdoors: Backdoors also make use of diffuse patterns, handcrafted or learned as well, that are blended over an input given a blending ratio $\alpha \in (0, 1)$ such that:

$$\mathcal{T}(x^{\mathsf{cl}}) = (1 - \alpha) \cdot x^{\mathsf{cl}} + \alpha \cdot \mathtt{t}.$$

For instance, Chen et al. [10] blend a cartoon or a random pattern into an image, and Barni et al. [11] use a sine wave.

4) Other backdoor types: Prior works also expand beyond the binary framework of local vs. watermark backdoors. For instance, WaNet [12] uses imperceptible image warping as a trigger, whereas IADBA [7] uses a sparse trigger pattern. Overall, the breadth of attacks underscore the need for robust defenses that can defend against all possible types.

B. Backdoor defense methods

1) Backdoor detection: A defender designs a binary test to detect a backdoor, deriving a metric and corresponding threshold to filter out DNNs or data (i.e. *input filtering*). For instance, Neural Cleanse [3] detects a backdoor by assessing each class predicted by a DNN: it reconstructs potential triggers and computes anomaly indexes to flag backdoored classes. Meanwhile, STRIP [13] filters out test-time inputs that display a low entropy when superposed with other inputs.

2) Backdoor removal: When detection is not achievable, a defender may instead suppress backdoors at the DNN [5] or data level [6]. At the DNN level, methods like DeepSweep [5] fine-tune a model to erase potential backdoors. Meanwhile, *input purification* defenses preprocess *all* test-time data to (1) clean suspicious areas (e.g. BDMAE [6]) or (2) transform them as a whole to cause a trigger-backdoor mismatch (e.g. ShrinkPad [8]). Defenses like DeepSweep perform removal at both levels. Alongside fine-tuning a DNN, it also refines a data augmentation policy applied to test-time inputs.

3) Other defense types: Defenses also fit multiple groups. For instance, whether a defense is white-box or black-box. White-box defenses like DeepSweep [5] require some access to a DNN's internals, e.g. its weights, whereas black-box ones like BDMAE [6] only need a DNN's inputs and outputs.

C. Contributions and prior work

Section IV first focuses on **assessing the effectiveness of input purification defenses** (i.e. data-level removal). Such methods matter as they are easily portable to multiple DNNs and datasets, making them attractive to any defender. However, they are typically evaluated using patch-based attacks [6]. Such a setting indicates an *adversary-aware* defender, a threat model that cannot be assumed in real-life. In this context, we expand the evaluation of these defenses to watermark-based attacks. To the best of our knowledge, we are the first to explicitly focus on assessing input purification beyond patch triggers. Secondly, if we observe that input purification is currently limited in scope and effectiveness, we also highlight that **other backdoor defense categories do not offer better prospect**. This reinforces the problem that existing defenses have yet to catch up with existing attacks, especially when working outside the domain of their attacker-aware design. Furthermore, we identify that **existing defenses tend to be dependent on the characteristics of the underlying dataset**. In this context, **we demonstrate in Section V a simple mitigation policy** to palliate some of the existing shortcomings shown in this paper: using two defenses in tandem.

To perform our experiments, we first rely on four input purification defenses: BDMAE [6], DeepSweep [5], Februus [14], and ShrinkPad [8] (see Section III-D for further detail on the defenses used in this paper). We chose the first method for yielding recent state-of-the-art results [6] and the latter three for being representative of the broader input purification literature [1]. We further motivate our choice by having an even split between white-box (DeepSweep, Februus) and black-box methods (BDMAE, ShrinkPad). For our second topic, we additionally cover two ubiquitous detection defenses: Neural Cleanse [3] (model level) and STRIP [13] (input level).

III. Experimental setup

A. Threat model

We consider a defender who gains access to a backdoored DNN in either a white or black-box scenario and must thwart backdoor attackers. The defender has three levels of control depending on the backdoor defense's requirements. Beyond an access to a DNN's test-time inputs and outputs (for BDMAE [6], Neural Cleanse [3], ShrinkPad [8], and STRIP [13]), the defender has access to a DNN's weights and its validation data for DeepSweep [5] or to its activations for Februus [14].

B. Experimental setup

We use 3 datasets: CIFAR10, CASIA-Webface (abbr. CASIA), and CelebA. The latter two are curated to retain the identities with the most elements (see dataset info in Table I).

We train ResNet-18 DNNs on CIFAR10 and ResNet-50 DNNs on CASIA and CelebA (see Table IV). Training, validation, and test splits are 75%, 10% and 15%. ResNet-18s are trained for 100 epochs, using the $Adam$ optimizer with an initial learning rate $lr = 0.01$ (divided by 10 at epochs 33, 75, and 90). Training data is augmented with: random crop, random rotation, and color jitter. ResNet-50s are trained

TABLE I
DETAILS OF DATASETS AND CLASSIFIERS

Dataset	#Classes	Input Size	#Images	DNN
CIFAR10	10	3 x 32 x 32	50,000	ResNet-18
CASIA	200	3 x 112 x 112	51,341	ResNet-50[a]
CelebA	467	3 x 112 x 112	13,000	ResNet-50[a]

[a]Pretrained on full CASIA-Webface using ArcFace loss.

TABLE II
ATTACKS COVERED IN THIS PAPER

Patch & Others			Watermarks	
Name	Ref.	Type	Name	Ref.
BadNets	[2]	patch	Chen et al. (cartoon)	[10]
BadNets (Dynamic)		patch	Chen et al. (noise)	
Chen et al. (glasses)	[10]	patch	ISSBA	[15]
IADBA	[7]	other (sparse)	Refool	[16]
WaNet	[12]	other (warping)	SIG	[11]

for 200 epochs, using $Adam$ with $lr = 0.01$ (divided by 10 every 50 epochs). Training data is augmented with a random horizontal flip. Training differs only for WaNet [12] (see Section III-C) where we follow the paper's own regimen [12].

C. Backdoor attacks

This paper covers 2 variants of **BadNets** [2], 3 variants of **Chen et al.** [10], **SIG** [11], **WaNet** [12], **IADBA** [7], **ISSBA** [15], and **Refool** [16] (see Table II). These 10 backdoor attacks are representative of the overall backdoor literature [1]: 3 of them use patches, 5 watermarks, and 2 other types of triggers. BadNets, Chen et al., SIG and WaNet use handcrafted triggers, whereas IADBA, ISSBA and Refool use optimized ones. IADBA and ISSBA leverage an input-specific generative approach, whereas Refool refines a dictionary of triggers built from an out-of-distribution dataset.

Each attack covered in this paper is all-to-one, i.e., all classes are backdoored such that an input altered with $\mathcal{T}$ maps to a single, attacker-specified target class y^{po} (randomly drawn before training). We train backdoored DNNs starting from their benign pretrained versions and use a poisoning ratio $\beta = 0.05$. For BadNets [2], we use the 3x3 RGB pattern provided by [6]. For Chen et al. [10], we use its three canonical variants (i.e. a pink glasses patch, a cartoon image provided by the authors, and a uniformly drawn noise pattern). The glasses patch is only used on CASIA and CelebA.

D. Backdoor defenses

This paper covers 6 defenses: 4 input purifications, 1 backdoor detection, and 1 input filtering methods (see Table III).

For input purification defenses, we select **ShrinkPad** [8], **DeepSweep** [5], **Februus** [14], and **BDMAE** [6]. ShrinkPad applies a simple transformation (i.e. shrinking and padding) to test-time inputs forwarded to a DNN to damage incoming triggers. DeepSweep resembles ShrinkPad but fits a white-box setting that involves fine-tuning a suspicious DNN while refining a data augmentation pipeline that will be applied to test-time inputs. Meanwhile, Februus and BDMAE follow a generative approach where suspicious areas in test-time inputs are erased and then reconstructed via an inpainting generative adversarial network (GAN) or masked autoencoder (MAE). For backdoor detection, we use the ubiquitous **Neural Cleanse** [3] defense to verify our intuition that backdoor detection is not always feasible. Finally, we use **STRIP** [13] to give us insights into input filtering.

TABLE III
DEFENSES COVERED IN THIS PAPER

Name	Ref.	Type	Access required
BDMAE	[6]	input purification	black-box
DeepSweep	[5]	model & input purification	white-box
Februus	[14]	input purification	white-box
Neural Cleanse	[3]	backdoor detection	black-box
ShrinkPad	[8]	input purification	black-box
STRIP	[13]	input filtering	black-box

TABLE IV
CLEAN DATA ACCURACY (CDA) AND ATTACK SUCCESS RATE (ASR) OF MODELS AND ATTACKS CONSIDERED IN THIS PAPER

Backdoor	CIFAR10		CASIA		CelebA	
	CDA	ASR	CDA	ASR	CDA	ASR
Benign	93.2%	*n.a.*	93.6%	*n.a.*	87.6%	*n.a.*
BadNets	93.4%	99.8%	92.0%	100%	86.9%	100%
BadNets (Dyn.)	93.7%	99.1%	92.2%	100%	86.8%	99.8%
Chen et al. (glasses)	*n.a.*	*n.a.*	91.9%	100%	86.2%	100%
Chen et al. (cartoon)	93.8%	99.2%	91.7%	100%	86.8%	100%
Chen et al. (noise)	93.4%	99.6%	92.5%	100%	86.7%	100%
IADBA	88.5%	99.9%	86.7%	98.4%	81.2%	99.6%
ISSBA	89.9%	99.3%	90.9%	100%	83.6%	97.8%
Refool	92.6%	60.8%	91.4%	98.1%	84.7%	89.7%
SIG	93.6%	99.9%	92.3%	100%	86.8%	100%
WaNet	94.0%	94.8%	92.4%	92.9%	85.3%	85.1%

For the input purification defenses, we lift from the authors' described setups. ShrinkPad [8] uses a 12.5% shrinking ratio. DeepSweep [5] performs a 10-epoch fine-tuning and uses the authors' specified data augmentation transforms. Februus [14] uses a mask size of 0.8 and the GradCam++ method targeting the 'layer.3' of the ResNet models. BDMAE [6] uses the authors' MAE-*base* model. For Neural Cleanse [3], we use the authors' anomaly index threshold of 2. We test 9 random classes alongside the backdoored class for CASIA and CelebA. We then report whether the backdoored class is found alongside potential false positives on benign classes. Finally, for STRIP [13], we compute a threshold value on held-out validation data such that the false rejection rate of benign inputs is $FRR = 1\%$ (results are then reported on test data).

E. Metrics

For the input purification defenses, we compare a DNN's clean data accuracy (CDA) and backdoor attack success rate (ASR) against their sanitized versions (SDA and $SASR$), obtained after applying the corresponding defenses on benign and backdoored test-time data. For Neural Cleanse [3], we check whether a backdoored class is flagged with or without false positives on benign classes. For STRIP [13], we provide the false acceptance (FAR) and false rejection (FRR) rates, computed on backdoored and benign test-time data respectively (thresholds are computed on validation data).

IV. RESULTS

A. Input purification defenses

Table V reports the sanitized clean data accuracy (SDA) and the sanitized attack success rate ($SASR$) when applying a defense (the color coding is arbitrary and meant to help browsing the table).

TABLE V
INPUT PURIFICATION RESULTS
(GREEN INDICATES SDA IS LESS THAN 10 POINTS LOWER THAN CDA AND $ASR < 30\%$, RED IF SDA IS MORE THAN 10 POINTS LOWER THAN CDA AND $ASR > 30\%$, ORANGE OTHERWISE).

Defense	Attacks	CIFAR10		CASIA		CelebA	
		SDA	$SASR$	SDA	$SASR$	SDA	$SASR$
ShrinkPad	BadNets	91.7%	5.3%	67.9%	0.1%	70.8%	1.1%
	BadNets (Dyn.)	92.1%	33.6%	66.7%	66.2%	70.9%	78.9%
	Chen et al. (glasses)	*n.a.*	*n.a.*	67.2%	73.1%	66.2%	92.8%
	Chen et al. (cartoon)	92.1%	13.2%	72.8%	95.7%	71.4%	96.1%
	Chen et al. (noise)	90.8%	10.0%	67.3%	78.8%	72.5%	73.1%
	IADBA	86.6%	1.1%	71.5%	39.1%	63.9%	38.8%
	ISSBA	89.3%	85.2%	1.2%	99.6%	63.5%	98.1%
	Refool	88.9%	65.5%	46.3%	96.0%	66.4%	91.1%
	SIG	91.0%	94.0%	69.2%	98.9%	71.9%	99.9%
	WaNet	33.0%	99.2%	52.4%	95.8%	25.1%	98.1%
DeepSweep	BadNets	93.4%	1.2%	86.0%	0.0%	70.7%	0.4%
	BadNets (Dyn.)	92.5%	85.7%	81.3%	0.1%	80.2%	19.8%
	Chen et al. (glasses)	*n.a.*	*n.a.*	86.6%	77.6%	82.1%	100%
	Chen et al. (cartoon)	92.5%	28.2%	83.7%	50.8%	79.9%	77.1%
	Chen et al. (noise)	91.9%	72.6%	86.5%	1.9%	74.2%	7.6%
	IADBA	88.8%	1.3%	86.9%	14.7%	80.5%	1.3%
	ISSBA	93.3%	27.9%	83.7%	0.0%	76.7%	0.0%
	Refool	92.3%	26.2%	79.0%	0.2%	79.1%	64.4%
	SIG	93.0%	93.3%	75.5%	0.2%	77.2%	80.2%
	WaNet	91.5%	11.3%	92.8%	0.1%	86.6%	0.0%
Februus	BadNets	90.3%	1.1%	89.6%	0.0%	82.3%	0.8%
	BadNets (Dyn.)	91.6%	1.8%	89.7%	13.3%	84.0%	72.8%
	Chen et al. (glasses)	*n.a.*	*n.a.*	90.6%	100%	82.8%	100%
	Chen et al. (cartoon)	90.9%	97.1%	90.4%	100%	84.6%	100%
	Chen et al. (noise)	91.3%	99.5%	90.1%	100%	83.6%	100%
	IADBA	85.9%	2.4%	82.0%	74.4%	77.5%	47.8%
	ISSBA	90.0%	93.7%	89.5%	100%	79.2%	95.9%
	Refool	89.7%	55.8%	82.1%	96.6%	81.1%	90.4%
	SIG	90.8%	100%	91.0%	100%	84.7%	100%
	WaNet	91.2%	18.2%	90.5%	77.4%	86.1%	63.6%
BDMAE	BadNets	94.0%	1.2%	90.6%	0.0%	82.1%	0.3%
	BadNets (Dyn.)	93.1%	0.7%	90.2%	0.3%	82.9%	1.0%
	Chen et al. (glasses)	*n.a.*	*n.a.*	90.3%	100%	82.0%	100%
	Chen et al. (cartoon)	92.7%	97.5%	90.5%	100%	82.9%	100%
	Chen et al. (noise)	92.5%	97.1%	90.4%	95.2%	81.3%	97.0%
	IADBA	88.9%	0.5%	83.7%	33.9%	74.6%	23.0%
	ISSBA	90.2%	99.1%	90.1%	85.5%	81.4%	97.0%
	Refool	92.4%	59.4%	89.5%	90.1%	80.5%	87.1%
	SIG	93.4%	99.8%	91.2%	100%	83.2%	100%
	WaNet	93.6%	17.9%	91.8%	59.0%	80.0%	69.2%

On CIFAR10, ShrinkPad is an effective black-box defense against local or sparse patterns, i.e. BadNets [2] and IADBA [7], as well as against the Chen et al. [10] watermarks. However, ShrinkPad is demonstrably not adapted to defending against any backdoors on the CASIA and CelebA datasets as we observe important drops in accuracy on clean data (see SDA in Table V). This failure arises from the characteristic of face recognition DNNs needing to infer on well-aligned faces, which ShrinkPad breaks. Additionally, ShrinkPad does not result in a drop in ASR against watermark triggers on CASIA and CelebA. A stronger set of transforms may be required to cause a trigger-backdoor mismatch albeit at a higher SDA cost. Finally, we note a generally lower SDA in the case of WaNet [12] and a catastrophic SDA decrease in the case of ISSBA [15] on CASIA. Here, we surmise that fine, e.g. warping-based, attacks may cause DNNs to be much more brittle to spatial transformations. This warrants further exploration that is however outside the scope of this paper.

DeepSweep [5] is effective against a higher number of backdoors than all other defenses (see Table V). However, we observe some failures against watermark backdoors on more complex datasets (e.g. ISSBA [15] and SIG [11] on CASIA). Here, we point to Gu et al. [2] who note that local backdoor patterns are typically learned by a few neurons, which are likely to be modified by DeepSweep's fine-tuning. We surmise that, for watermarks, a backdoor is diffused along different neuron pathways, making it harder to erase. Unfortunately, a downside to DeepSweep is its white-box nature.

TABLE VI
NEURAL CLEANSE [3] RESULTS (TP+FP: THE BACKDOORED CLASS IS FOUND BUT 1+ BENIGN CLASSES ARE ALSO FLAGGED; FN+FP: THE BACKDOORED CLASS IS NOT FOUND BUT 1+ BENIGN CLASSES ARE FLAGGED; FN: THE WORST CASE SCENARIO AS NO CLASS IS FLAGGED, I.E. THE DNN MAY BE SEEN AS BENIGN).

Backdoor	CIFAR10	CASIA	CelebA
BadNets	TP + FP	TP + FP	FN
BadNets (Dyn.)	TP + FP	FN + FP	FN
Chen et al. (glasses)	*n.a.*	FN + FP	FN + FP
Chen et al. (cartoon)	FN + FP	FN	FN + FP
Chen et al. (noise)	FN + FP	FN	FN
IADBA	FN + FP	FN	TP + FP
ISSBA	FN + FP	FN + FP	FN + FP
Refool	FN + FP	TP + FP	FN
SIG	FN + FP	FN	FN
WaNet	FN + FP	FN + FP	FN + FP

Lastly, both Februus and BDMAE are effective against patch-based backdoors like BadNets [2] and, for CIFAR10, IADBA [7] and WaNet [12] (see Table V). We further note that, when effective, BDMAE typically supersedes Februus in SDA and $SASR$. This matters given BDMAE is a black-box method. However, both methods fail against ISSBA [15], Refool [16], and SIG [11] while having mixed results against WaNet [12] on CASIA and CelebA. This indicates that current input purification method do not generalize to the broader scope of existing attacks and to more complex datasets.

Overall, input purification defenses appear to be adversary-aware, requiring that attacks be patch-based to function. Additionally, if a defense works against backdoor attacks on CIFAR10, a common but simple dataset, they unfortunately fail on more complex tasks (e.g. CASIA, see Table I).

B. Neural Cleanse

We observe three types of undesirable outcomes when testing Neural Cleanse [3]:

1) fails to detect the backdoored class (false negative, **FN**),
2) detects the backdoored class but flags benign classes as well (true positive + false positive(s), **TP+FP**),
3) fails to detect the backdoored class but does benign classes (false negative + false positive(s), **FN + FP**).

Neural Cleanse yields false positives on benign classes in a majority of cases, regardless of the results on the backdoored class (see Table VI). Additionally, as dataset complexity rises with CASIA and CelebA, we observe that Neural Cleanse starts flagging backdoored models as benign. This unfortunately would lead to their deployment by unsuspecting users if they were only relying on this defense method.

These results demonstrate that Neural Cleanse fails against increasingly complex backdoors and datasets, validating our intuition set in Section III-A. This underscores the need for data-based defenses like input purification methods.

C. Input filtering

We observe mixed results for STRIP [13] (see Table VII). The best results are found against BadNets [2], Chen et al. [10]'s watermark triggers, IADBA [7], and SIG [11] on CASIA and CelebA. However, as illustrated against Refool [16],

TABLE VII
STRIP RESULTS (TEST-TIME DATA FAR/FRR GIVEN THRESHOLDS COMPUTED S.T. CLEAN VALIDATION DATA $FRR = 1\%$, GREEN: $FAR \leq 5.0\%$; ORANGE: OTHERWISE).

Backdoor	CIFAR10		CASIA		CelebA	
	FAR	FRR	FAR	FRR	FAR	FRR
BadNets	0.0%	6.7%	3.8%	2.6%	0.0%	2.8%
BadNets (Dyn.)	23.1%	5.7%	3.2%	2.5%	3.7%	2.2%
Chen et al. (glasses)	*n.a.*	*n.a.*	59.4%	1.7%	85.5%	2.6%
Chen et al. (cartoon)	2.4%	7.1%	2.3%	2.0%	7.1%	3.5%
Chen et al. (noise)	1.9%	7.6%	8.2%	2.7%	16.5%	1.4%
IADBA	94.5%	8.9%	34.5%	3.1%	95.8%	1.2%
ISSBA	98.7%	4.6%	50.6%	1.6%	34.6%	3.2%
Refool	99.2%	0.1%	100%	0.0%	100%	0.0%
SIG	5.0%	8.5%	1.6%	2.8%	7.4%	3.7%
WaNet	92.7%	8.9%	25.3%	3.5%	94.1%	2.5%

TABLE VIII
STRIP+BDMAE MIXED STRATEGY RESULTS ON CASIA (SAME COLOR LEGEND AS TABLE V); TEST-TIME DATA FAR/FRR GIVEN THRESHOLDS COMPUTED S.T. CLEAN VALIDATION DATA $FRR = 1\%$.

Backdoor	SDA	$SASR$
BadNets	91.6%	0.0%
BadNets (Dyn.)	92.1%	0.0%
Chen et al. (glasses)	92.0%	59.4%
Chen et al. (cartoon)	91.0%	2.3%
Chen et al. (noise)	91.8%	7.4%
IADBA	84.1%	0.5%
ISSBA	90.7%	50.6%
Refool	91.1%	92.1%
SIG	92.3%	1.6%
WaNet	80.8%	13.1%

STRIP is yet another imperfect defense. STRIP's computed threshold is wrongly negative, a defense-breaking problem that was also reported in another context in the original paper [13].

These mixed results underscore the hardness of reliably filtering backdoored inputs (Neural Cleanse [3] fails in similar settings). This highlights the needs for better defenses that eschew using brittle binary tests.

V. A DOUBLE-EDGED SWORD: MIXING DEFENSES

The brittleness and task variability of backdoor defenses is an open problem. Future work must expand defenders' state-of-the-art capabilities to cover more complex, often watermark-based triggers, especially in a black-box setting that excludes methods like DeepSweep [5]. Nonetheless, defenders are not powerless while waiting for future defenses. Here, we make the simple observation that defenses are typically assessed in isolation. To the best of our knowledge, mixing them is a rarely covered idea than may yield substantial gains.

We empirically assess mixing two *black-box* methods: (1) rejecting inputs with STRIP [13] then (2) purifying accepted inputs with BDMAE [6]. We choose BDMAE as it is state-of-the-art and STRIP as it is the input filtering method we previously covered. To assess this strategy, we use CASIA as it is both a complex task and displays the highest CDA and SDA in our experiments in Section IV.

As illustrated in Table VIII, we show that mixing both methods can lower a backdoor attack's ASR, notably against some previously unbeaten watermark triggers. For instance, on SIG [15], we reduce the ASR to 1.6% versus 100% with only BDMAE at the cost of only rejecting 2.8% of benign test inputs. When one defense fails, the other picks up the mantle. This demonstrates that different defenses can provide a complementary protection against backdoor attacks at the cost of rejecting an average $2-3\%$ of benign inputs (we note it may be unacceptable in some applications). This is a noteworthy gain for such a simple workaround at the moment.

Future work may need explore which combinations of defenses are effective in order to break the ongoing cat-and-mouse game. The goal is to find, if not a stand-alone defense, a mix that robustly performs on a variety of attacks and tasks.

VI. CONCLUSION

This paper demonstrates a blind spot in the backdoor defense literature. Defenses lack robustness and generalization in the face of harder, typically watermark-based, backdoor attacks and more complex datasets. We highlight the need for more robust, adversary-agnostic methods that go beyond the current state-of-the-art (e.g. restriction to patch-based attacks for input purification defenses). In the meantime, we demonstrate that a defender may rely on a black-box defense mixing strategy to better cover the range of attacks.

ACKNOWLEDGEMENTS

Funded by ANR/AID project SAIDA ANR-20-CHIA-0011.

REFERENCES

[1] O. Mengara, A. Avila, and T. H. Falk, "Backdoor attacks to deep neural networks: A survey of the literature, challenges, and future research directions," *IEEE Access*, vol. 12, pp. 29004–29023, 2024.

[2] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," 2019.

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

[4] J. Guo, Y. Li, X. Chen, H. Guo, L. Sun, and C. Liu, "SCALE-UP: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency," in *The Eleventh International Conference on Learning Representations*, 2023.

[5] Y. Zeng, H. Qiu, S. Guo, T. Zhang, M. Qiu, and B. M. Thuraisingham, "Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation," *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2020.

[6] T. Sun, L. Pang, C. Chen, and H. Ling, "Mask and restore: Blind backdoor defense at test time with masked autoencoder," 2023.

[7] A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," 2020.

[8] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the trigger of backdoor attack," 2021.

[9] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," 2022.

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

[11] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning," 2019.

[12] A. Nguyen and A. Tran, "Wanet – imperceptible warping-based backdoor attack," 2021.

[13] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: A defence against trojan attacks on deep neural networks," 2020.

[14] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defence against trojan attacks on deep neural network systems," 2019.

[15] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16463–16472, October 2021.

[16] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," 2020.