Detecting Scene-Plausible Perceptible Backdoors in Trained DNNs without Access to the Training Set

Zhen Xiang  
zux49@psu.edu  
David J. Miller  
djmiller@engr.psu.edu  
Hang Wang  
hzw81@psu.edu  
George Kesidis  
gik2@psu.edu  

School of Electrical Engineering and Computer Science, College of Engineering, The Pennsylvania State University, University Park, PA 16802, U.S.A.

Backdoor data poisoning attacks add mislabeled examples to the training set, with an embedded backdoor pattern, so that the classifier learns to classify to a target class whenever the backdoor pattern is present in a test sample. Here, we address posttraining detection of scene-plausible perceptible backdoors, a type of backdoor attack that can be relatively easily fashioned, particularly against DNN image classifiers. A posttraining defender does not have access to the potentially poisoned training set, only to the trained classifier, as well as some unpoisoned examples that need not be training samples. Without the poisoned training set, the only information about a backdoor pattern is encoded in the DNN’s trained weights. This detection scenario is of great import considering legacy and proprietary systems, cell phone apps, as well as training outsourcing, where the user of the classifier will not have access to the entire training set. We identify two important properties of scene-plausible perceptible backdoor patterns, spatial invariance and robustness, based on which we propose a novel detector using the maximum achievable misclassification fraction (MAMF) statistic. We detect whether the trained DNN has been backdoor-attacked and infer the source and target classes. Our detector outperforms existing detectors and, coupled with an imperceptible backdoor detector, helps achieve posttraining detection of most evasive backdoors of interest.

1 Introduction

Deep neural network (DNN) classifiers have achieved state-of-the-art performance in many applications. However, they have also been shown to be vulnerable to adversarial attacks (Szegedy et al., 2014). This has inspired
adversarial learning research, including work in devising formidable attacks as and defenses against attacks (Miller, Xiang, & Kesidis, 2020). Test-time evasion (TTE) is a prominent type of adversarial attack aiming to induce misclassifications during classifier operation by modifying test samples in a human-imperceptible (or machine-evasive) fashion (Goodfellow, Shlens, & Szegedy, 2015; Papernot et al., 2016; Moosavi-Dezfooli, Fawzi, & Frossard, 2016; Carlini & Wagner, 2017; Chen, Sharma, Zhang, Yi, & Hsieh, 2018). Another type of attack, data poisoning (DP), inserts malicious samples into the training set, often to degrade the classifier’s accuracy (Huang, Joseph, Nelson, Rubinstein, & Tygar, 2011; Xiao et al., 2015; Biggio & Roli, 2018).

Another type of DP attack, called a backdoor attack (BA) (usually against DNN image classifiers), has been proposed, aiming to covertly add a backdoor mapping into the learned DNN, while not degrading its performance in accurately classifying legitimate (clean) test samples, which do not have the backdoor pattern embedded (Chen, Liu, Li, Lu, & Song, 2017; Liu, Ma, Aafer, Lee, & Zhai, 2018; Gu, Liu, Dolan-Gavitt, & Garg, 2019). A learned backdoor can be easily achieved by inserting a relatively small number of poisoned samples into the training set. That is, backdoor-poisoned training samples are crafted by embedding the same backdoor pattern (BP) into clean samples from one or more source classes and (mis)labeling to a target class (selected by the attacker).

The BP, principally designed to be evasive to possible occasional human inspection of the training set (or at test time), could be human imperceptible, for example, additive perturbation applied (digitally) to clean image samples, dubbed here an imperceptible BP (Zhong, Squicciarini, Zhu, & Miller, 2020; Tran, Li, & Madry, 2018; Xiang, Miller, & Kesidis, 2019, 2020a, 2020b). Or it could be a seemingly plausible object that can be inserted in the scene of an image, dubbed here a scene-plausible perceptible BP (SPP BP), for example, a bird flying in the sky or glasses on a face (Chen et al., 2017; Gu et al., 2019; Guo, Wang, Xing, Du, & Song, 2019). Such patterns can be physically inserted and then captured in a digital image sample, or it could be digitally inserted into the scene by the attacker. The former we refer to as a physical attack. Spatially fixed perceptible patterns that are not scene plausible (e.g., a noisy patch fixed to a corner of the image) have also been used in Wang et al. (2019), and Guo et al. (2019). The DNN trained on the poisoned training set will still correctly classify clean test samples with high accuracy (because the number of poisoned samples is relatively small); hence, validation set accuracy degradation (Nelson et al., 2009) cannot be reliably used to detect BAs. The DNN is likely to misclassify to the attacker’s target

---

1 The attacker’s poisoning capability is facilitated by the need in practice to obtain big data suitable for accurately training a DNN for a given domain. To do so, one may need to seek data from as many sources as possible (some of which could be attackers).
Detecting Scene-Plausible Perceptible Backdoors

class any test sample from the attacker’s source class with the BP (used in training) embedded.

Defenses against BAs can be deployed before or during training, post-training, or potentially during operation or test time (“in-flight” detection; Chou, Tramer, Pellegrino, & Boneh, 2018; Gao et al., 2019). In the before/during-training scenario, the defender has access to the (possibly poisoned) training set and the trained classifier (Tran et al., 2018; Chen, Carvalho et al., 2018; Xiang et al., 2019). The defender seeks to detect whether the training set has been poisoned and, if so, to identify and remove the poisoned training images before (re)training. In the posttraining scenario considered here, however, the defender has access to the trained DNN but not to the training set used for its learning (Liu, Doan-Gavitt, & Garg, 2018; Wang et al., 2019; Guo, Wang, Xing, Du, & Song, 2019; Xiang et al., 2020a, 2020b). This scenario is of great interest because there are many pure consumers of machine learning systems. For example, critical infrastructure is often based on legacy or proprietary system classifiers. In such a scenario, it is very possible that the original training data used to build the classifier are unavailable. Also, an app may be used on millions of cell phones, so app users will not have access to the training set. Still, users would like to know whether the app’s classifier has been backdoor-attacked. Users are assumed to have a clean, labeled data set with examples from each of the classes in the domain. This clean set will generally be relatively small (users may not have access to either substantial training data or computational resources necessary for DNN training); hence, while this clean set is useful for building a BA detector, it is not sufficient for training a surrogate DNN (one without a backdoor present). The defender seeks to detect whether the DNN has been backdoor-poisoned. If a detection is made, it is desirable to also identify the source class(es) and the target class.

In this letter, we focus on posttraining detection of BAs with SPP BPs, a challenging problem because the training set is unavailable. Elsewhere, posttraining detection of imperceptible backdoors is addressed (Xiang et al., 2020a, 2020b). Together, our work and that of Xiang et al. (2020a, 2020b) cover most cases of interest. If these two detectors are used in parallel, they provide essentially a complete solution to posttraining detection of evasive backdoors. We make four main contributions:

First, we are the first to introduce and define the problem of posttraining detection of SPP BAs in trained DNN image classifiers. We are also the first to address the major differences between spatially fixed, perceptible backdoors considered in previous work and SPP backdoors that can be easily physically implemented in practice.

Second, we propose a novel approach for detecting such attacks. Our detector is related to Xiang et al. (2020b) and Wang et al. (2019), but unlike these methods, our approach is designed to detect SPP backdoors. Our detector is based on the maximum achievable misclassification fraction (MAMF) statistic, obtained by estimating a putative BP for each
putative (source, target) class pair. Our detection inference uses an easily chosen threshold. No assumptions on the shape, spatial location, or object type of the BP are made. By contrast, Guo et al. (2019) assume the BP occupies one of the four corners of the image.

Third, we identify and experimentally analyze two important properties of SPP BPs: spatial invariance and robustness. These properties, which have neither been identified nor exploited in prior work, are valid for physical (and photo-manipulated) BAs using objects (e.g., a pair of glasses) that may be commonly launched in practice. They are the basis for our detector and may inspire other defenses.

Finally, we perform substantial experimental evaluation and show the strong capability of our detector in general and compared to the two existing methods of Wang et al. (2019) and Liu, Doan-Gavitt et al. (2018) for the posttraining detection problem we address. Wang et al. (2019) proposed the neural cleanse (NC) method as a general backdoor detector, but they only experimentally assessed it for the detection of perceptible BPs that are always placed in the same fixed spatial location. Such attacks are not in general scene-plausible, and learners can detect or remove them via simple training set sanitization. Here we assess the NC detector more generally. Our experiments involve five data sets of practical interest, five commonly used DNN structures, nine BPs of different types, and two compared detection methods.

This letter is an expanded version of our conference paper: Xiang, Miller, Wang, and Kesidis (2020a). The preprint of this work is also available: Xiang, Miller, Wang, and Kesidis (2020b).

The rest of the letter is organized as follows. In section 2, we provide a thorough, unified review of all major aspects of BAs and defenses. In section 3, we review existing posttraining defenses. In section 4, we discuss two properties of SPP BPs, based on which we develop our detection procedure. In section 5, we report our experiments. Conclusions and future work are in section 6.

Throughout this letter, we use $X$ to denote the image domain of interest and $C$ to denote the associated set of labels with $|C| = K$. The image dimension is $W \times H \times C$, where $W$ and $H$ are image width and image height in pixels, respectively; $C$ is the number of image channels. $f(\cdot; \theta) : X \rightarrow C$ denotes a classifier’s mapping from the image to the predicted label, where $\theta$ are the classifier’s parameters. $p(c|x, \theta)$ denotes the classifier’s posterior probability for class $c \in C$ for an image $x \in X$.

2 Backdoor Attack

2.1 Comparison with Other Adversarial Attacks. BAs are easily distinguishable from other prominent types of adversarial attacks, including test-time evasion (TTE) and data poisoning (DP) attacks that simply seek to degrade classifier accuracy.
The goal of a TTE attacker is the same as that of a backdoor attacker: to introduce not easily perceived perturbations to an image at test time that cause the classifier to alter its decision (Papernot et al., 2016; Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017; Chen et al., 2018). For TTEs, this is achieved by designing an image-specific additive perturbation, typically with as small an $L_p$ norm as possible, that causes a desired class decision change. In crafting this perturbation, the TTE attacker uses knowledge (including the structure and the learned weights) of the target classifier or a surrogate classifier that behaves in a similar way on the same data domain (Papernot et al., 2017).

DP attacks that aim to degrade the accuracy of the classifier do so by poisoning its training data by adding, for example, incorrectly labeled or noisy samples (Biggio & Roli, 2018; Huang et al., 2011; Xiao et al., 2015; Koh, 2017). A typical DP attacker only needs knowledge of the classification domain and the ability to poison the training data, although recent work shows that more powerful attacks can be devised if the classifier’s architecture and parameter space are known to the attacker (Yang, Wu, & Chen, 2017). Note that DP attacks change the trained classifier, while TTE attacks do not alter the classifier.

The goals of a backdoor DP attacker are twofold. First, the learned classifier is supposed to classify to an attacker-desired target class whenever a test image from a certain different source class (or source classes) has an attacker-specified BP embedded. Second, the trained classifier should correctly predict clean test images without the BP embedded (Chen et al., 2017; Gu et al., 2019). Rather than requiring knowledge of the classifier under attack (needed for a TTE attack), BAs rely on having the capability to poison the learner’s training set, so that the classifier learns a backdoor mapping. The attack is initiated by poisoning the training set with a set of images, each containing the same attacker-specified BP, mislabeled to the attacker’s target class.

Compared with TTE attacks, BAs may require much lower cost to launch, especially for large-scale deep learning frameworks (where data are often acquired from the public or from multiple sources), and BAs do not require knowledge of the trained DNN or a surrogate. Compared with DP attacks generally targeting classifier accuracy, BAs are stealthier because a successful attack will not affect the prediction on clean source-class images (without the BP) so validation-set accuracy degradation cannot be used as a basis for detecting BAs. (See Miller et al., 2020, for other types of attacks, including reverse-engineering attacks and attacks on data privacy.)

2.2 Elements of BAs. To launch a BA, the attacker needs to select a BP, a (set of) source class(es) $S^* \subset \mathcal{C}$, and a target class $t^* \in \mathcal{C}$.

2.2.1 Backdoor Pattern. A BP is also known as a backdoor trigger (Gu et al., 2019), a backdoor key (Chen et al., 2017), or a trojan. A BP is an
attacker-specified pattern used to poison examples included in the training set. The same pattern will also be used to induce misclassifications by embedding them in test samples from the source class(es). For the context of image classifiers, a backdoor image with a (spatially fixed) perceptible BP $v^*$ can be produced either by insertion of a physical pattern in a scene with subsequent digital image capture or by altering a digital image. In either case, the resulting digital image $\tilde{x}$ is represented mathematically by

$$\tilde{x} = g_p(x, v^*, m^*) = x \odot (1 - m^*) + v^* \odot m^*, \quad (2.1)$$

where $x \in \mathcal{X}$ is the originally clean image, and $m^*$ is an image-wide mask of dimension $W \times H$, with $m^*_{ij} \in \{0, 1\}$ for any pixel $(i, j)$. Often, the spatial region of 1’s for a mask is contiguous (see Figure 1), or forms several contiguous pieces, in which case the backdoor is being introduced through local patch replacement. The BP $v^*$ has the same dimension as $x$ but with only a small subset of nonzero pixels. For grayscale images, $\odot$ denotes element-wise multiplication. For color images, the two-dimensional mask $m^*$ (or the inverse mask $(1 - m^*)$) is applied to all channels of the images. An example is shown in Figure 1, where a backdoor image used for attacking a pet breed classifier is created by embedding a tennis ball in an image from class “chihuahua” and labeling it “Abyssinian.”

Figure 1: An example backdoor image used for attacking a pet breed classifier. A clean training image from class “chihuahua” is modified by adding an SPP BP (a tennis ball) and labeling to class “Abyssinian.”

---

1 In this letter, pixel intensity values are in $[0, 1]$. 
Detecting Scene-Plausible Perceptible Backdoors

“chihuahua” and labeling to “Abyssinian.” A scene with a dog and ball can of course also be a physical attack that is digitally captured.

In principle, the BP should be designed to be evasive to possible occasional human inspection of the training set. Also, similar to the stealthiness required by TTE attacks, a BP, when embedded in clean test images during operation, should not be easily noticeable to humans. For imperceptible BPs, such as an additive perturbation of an image with small $p$-norm (Tran et al., 2018; Chen et al., 2017; Xiang et al., 2019; Zhong, Zhong, Squicciarini, Zhu, & Miller, 2020; Xiang et al., 2020b), such stealthiness can be easily achieved.

Perceptible BPs considered by existing defenses were not chosen with evasiveness in mind. A noisy patch or a noticeable icon is fixed in the same location in every poisoned image (and backdoor test image) as the BP (Wang et al., 2019; Guo et al., 2019), that is, the mask $m^*$ is a constant for all backdoor images. Regardless of the incongruousness of these patterns that could easily raise human suspicion in practice, several issues should deter such spatially fixed perceptible BPs from being considered by a practical attacker:

- Spatially fixed perceptible BPs can be altered into digital images but usually cannot be placed in a scene that is digitally captured. For example, if the BP is a pair of glasses on a face, its location will be dependent on the location of the face. So, spatially fixed BPs cannot in general be used in physical attacks.
- If the same BP is embedded at the same location in all images used to poison the training set, a naive data sanitization procedure applied before or during training (a defense scenario described in section 2.3) could detect the attack, by, for example, checking the pixel value distribution, and so possibly remove it.
- Fixing the spatial location of the BP while creating backdoor training images will likely harm the robustness of the attack significantly during testing. As will be seen in section 5.6, for a perceptible BP fixed to the bottom left of the backdoor training images, when there is no data augmentation (e.g., rotation, flipping) used during training, even a single column or row shift of the BP at test time will severely degrade the attack effectiveness. This weakness of spatially fixed perceptible BPs is fatal. For physical attacks, there is no way to guarantee that the object is captured at the same location in test images as in the backdoor training images.

In this letter, we focus on detecting SPP BPs that are visually stealthy and have great potential to be physically implemented in practice. For convenience, we digitally embed the BPs into images in our experiments, but

\footnote{Notably, our detector is also capable of detecting spatially fixed perceptible backdoors as long as data augmentation options including random rotation and random horizontal...}
we spend laborious human effort to choose the object used as the SPP BP and to carefully design the mask to carve the same BP into the best (scene-plausible) location for each originally clean image (e.g., a bird flying in the sky).

2.2.2 Choice of Source Class(es) and Target Class. The source class(es) $S^*$ and the target class $t^*$ involved in a BA are chosen by the attacker. In most works, an attack involves a single target class (Chen et al., 2017; Liu, Ma et al., 2018; Tran et al., 2018; Chen, Carvalho et al., 2018; Wang et al., 2019; Xiang et al., 2020b). BAs involving more than one target class, dubbed an “all-to-all” attack, have been discussed in Gu et al. (2019), where the same BP, when embedded in a clean test image from class $i \in C$, is supposed to induce a misclassification to class $(i + 1)$. However, such a BA can be decomposed into $K = |C|$ attacks using the same BP, but each involving a unique (source, target) class pair.

The number of source classes involved in a BA could range from 1 to $(K - 1)$, that is, from a single source class (Tran et al., 2018; Chen, Carvalho et al., 2018; Xiang et al., 2019, 2020b) to all classes except for the target class (Chen et al., 2017; Gu et al., 2019; Wang et al., 2019; Xiang et al., 2020b)). For attacks using an imperceptible BP, the source classes can be arbitrarily chosen by the attacker without destroying imperceptibility. However, if a SPP BP is used, the source class(es) and the BP must be well matched to achieve stealthiness. For example, a bird flying in the sky could be used as a perceptible BP when most of the source class images contain the sky; hence, a class for which most of the images capture an underwater scene is not well matched with such a BP. Clearly, finding a perceptible BP that is scene-plausible when embedded in images from all classes (except for the target class) is very hard in many classification domains. So the number of source class(es) involved in a BA using an SPP BP depends on both the preference of the attacker and the classification domain. Moreover, for both perceptible and imperceptible backdoors, even if the attacker specifies a single (source, target) class pair, with $|S^*| = 1$, after training on the backdoor poisoned data, the classifier (during operation) could possibly classify images originally from a class $c \neq t$ and $c \notin S^*$ to the target class $t^*$, if the image contains the same BP used by the attacker. This phenomenon was experimentally observed in Xiang et al. (2020b) and called “collateral damage.” A collateral damage class pair is not among the backdoor class pairs specified by the attacker, but it can be viewed as an “effective” backdoor class pair induced during the training. Above all, for researchers developing defenses against BAs, it is very important to note that a robust defense should not

and vertical flipping are used during the classifier’s training, as shown in section 5.6.2. This is because the spatial location of the BP will likely be changed randomly when random augmentation is applied to training images, that is, the BP in the actual augmented training images is no longer spatially fixed.
Figure 2: Illustration of backdoor defense scenarios: before/during training, posttraining, and in-flight.

require any assumptions on the number of source classes involved in the attack. This will be borne out later, experimentally.

2.3 Defense Scenarios, Assumptions, and Goals. Defenses against BAs can be deployed at any stage, from the training stage to the operation stage, for a DNN classifier (see Figure 2). Existing works on backdoor defenses mainly focus on defeating BAs before or during training or posttraining (but before the operation of the classifier). Here, we describe both defense scenarios in detail.

2.3.1 Before or During Training Scenario. In the before or during training scenario, the defender, who could also be responsible for training the classifier, is assumed to have access to the (possibly poisoned) training set and either to the training process or to the trained (attacked) classifier (Tran et al., 2018; Chen, Carvalho et al., 2018; Xiang et al., 2019; Chou, 2018). The goals of the defender are to detect if the training set has been poisoned and to correctly identify and remove training images with the BP before training/retraining.

As BAs are designed to be evasive against sanitization and possible human inspection of the training set, existing defenses often first train a classifier even though the training data could possibly be poisoned. Then the trained classifier is analyzed to identify and remove suspicious training images before retraining (see Figure 2). For example, the spectral signature (SS) approach (Tran et al., 2018), activation clustering (AC; Chen, Carvalho et al., 2018), and cluster impurity (CI; Xiang et al., 2019) all use clustering of internal layer features (obtained from the classifier trained on the

---

1Poorly devised BAs may be defeated by sanitization of the training data.
possibly poisoned training set) to separate out BPs (that are labeled to the
target class) from clean images (considering all images labeled to the pu-
tative target class). SentiNet (Chou, 2018), which targets perceptible back-
doors, locates the BP if it appears in a training image, using a combinatoric
algorithm based on Grad-CAM (Selvaraju et al., 2017).

2.3.2 Posttraining Scenario. The defender in the posttraining scenario
could also be the user or consumer of the trained, possibly backdoor-
attacked classifier, who does not know whether the training authority is
trustworthy and cannot force the training authority to deploy a before/
during training defense. As shown in Figure 2, the defender does not have
access to the (possibly poisoned) training set used to train the classifier.
Hence there is no clue (except in the encoded DNN weights) about what
BPs look like if there is an attack, and this is true irrespective of whether
the attack is perceptible or imperceptible. This makes posttraining detec-
tion both very difficult and an intriguing problem. The defender does have
access to the trained classifier (including its structure and weights), but does
not have the resources (data or computational) to train a new classifier. The
defender is also assumed to possess an independent, clean (free of back-
doors), small labeled data set. Again, the user or defender cannot train a
surrogate classifier using this relatively small data set. The goals of the de-
defender are (1) to detect whether the classifier has been backdoor-attacked
or not, and (2) if an attack is detected, to infer the source class(es) and the
target class involved.

Clearly, approaches for before or during training defense (e.g., separat-
ing training BPs from clean ones) cannot be applied in the posttraining sce-
nario. Also, since BAs aim to not degrade accuracy on clean images, the
defender cannot infer whether the classifier has been attacked solely based
on the accuracy on an independent clean data set. Existing posttraining de-
defenses include the fine-pruning (FP) approach (Liu, Doan-Gavitt, & Garg,
2018), the NC approach Wang et al. (2019), the “TABOR” approach Guo et al.
(2019), and the anomaly detection (AD) approach (Xiang et al., 2020b). We
propose a post training defense against SPP backdoors and build on ideas
from Xiang et al. (2020b) in devising posttraining detection of imperceptible
backdoors. We review these other methods and point out their limitations
in section 3.

2.3.3 In-Flight. Defenses against BAs can potentially be deployed during
the operation or use of a classifier. In this “in-flight” scenario, the defender
only has access to the classifier and aims to determine whether there is a
BP embedded in a test image. An advantage of an in-flight defense is that
it may identify entities attempting to exploit the backdoor at test time. To
our knowledge, no in-flight defense purely based on the above assumptions
has been published to date. Both the SentiNet approach (Chou, 2018) and
the STRIP approach aim to detect if a test image contains a BP. However,
Detecting Scene-Plausible Perceptible Backdoors

these approaches also require that the defender possesses a small, clean data set.

3 Review of Backdoor Defenses Posttraining

To the best of our knowledge, the FP approach proposed in Liu, Doan-Gavitt et al. (2018) is the first attempt toward posttraining defense against BAs. FP, which removes neurons from the DNN, assumes that there is a simple dichotomization of neurons, with most solely dedicated to “normal” operation but with some solely dedicated to implementing the backdoor. Unfortunately, this assumption is not valid in many cases, as will be shown in our experiments in section 5.2. Moreover, FP does not actually detect the presence of BAs, and it removes neurons even from nonattacked DNNs.

The AD approach (Xiang et al., 2020b) considers posttraining detection of imperceptible backdoors. It was not proposed to detect perceptible backdoors. For each (source, target) class pair, an optimized additive perturbation that induces high group misclassification fraction when added to clean source class images is obtained. If the DNN is attacked, the optimized perturbation associated with the class pair involved in the attack will have abnormally low $L_p$ norm (e.g., $p = 2$), and a detection can be made on this basis. This approach cannot be used for detecting perceptible backdoors. First, reverse-engineering a perceptible BP (which could be any object) requires also reverse-engineering the associated mask. This is more challenging compared with searching for a small, additive perturbation starting from a zero (image-wide) initialized additive perturbation. Second, even if the BP used for devising the attack could be reverse-engineered for a true backdoor class pair by Xiang et al. (2020b), it could be hard to use the same metric to distinguish this from estimated patterns for the non-backdoor class pairs. This is because the premise for the imperceptible case that the required additive perturbation for the true backdoor pair has a small size or norm does not hold for the perceptible case. However, while Xiang et al. (2020b) is not applicable to detecting perceptible backdoors, our proposed detection approach is inspired by the detection framework in Xiang et al. (2020b) and is a relative of the approach in Xiang et al. (2020b).

The NC approach proposed in Wang et al. (2019) aims to detect general perceptible backdoors posttraining. NC obtains, for each putative target class, a pattern and an associated mask by optimizing an objective function that induces a high misclassification fraction when added to images from all classes other than the target class. If there is an attack, the $L_1$ norm of the obtained mask for the true backdoor target class is expected to be abnormally small, and a detection is made based on the median absolute deviation (MAD; Hampel, 1974). However, NC relies on the assumption that all classes except for the target class are involved in the attack, which is usually not guaranteed to hold for attacks using SPP BPs for the reasons discussed in section 2.2.2. Notably, removing this strong assumption is also
mentioned by the authors of NC as a future research direction. Moreover, NC penalizes the $L_1$ norm of the mask while maximizing the group misclassification fraction, to estimate the pattern and the mask; but the detection accuracy is sensitive to the choice of the penalizing multiplier. For CIFAR-10, unless the multiplier is carefully chosen, the results could be either a mask with a low $L_1$ norm but not achieving the target misclassification fraction, or a wild-looking BP with an improbably large and distributed spatial support, which has no relation to the ground-truth BP (for the true target class when there is in fact an attack). Finally, NC was proposed as a general backdoor detector (the NC paper did not discuss the cases of imperceptible, perceptible, or scene-plausible perceptible), but it was only experimentally evaluated on perceptible backdoors that were both not scene-plausible (suspicious-looking modifications to the image) and were placed in a fixed position in every image (both poisoned training images and test images). Again, fixing the position of a perceptible BP makes it impossible to guarantee the BP will be scene-plausible in a given test image. We will show in our experiments (in section 5.2) that NC is clearly outperformed by our proposed detection approach for SPP backdoors.

4 Posttraining Detection of SPP Backdoors

Our detector is designed to detect SPP backdoors. No assumptions about the shape, spatial support of the backdoor, or object to be used are made. Moreover, unlike NC and TABOR, we do not make any assumptions about the number of source classes chosen by the attacker (see section 2.2.2). The premise behind our detection approach allows it to detect BAs with the number of source classes ranging anywhere from 1 to $(K - 1)$.

Our detector follows the standard assumptions for the posttraining scenario described in section 2.3.2: the defender has access to a relatively small, clean data set containing images from all classes; the defender does not have access to the training set used for the DNN’s learning; the defender does not...
Detecting Scene-Plausible Perceptible Backdoors

have adequate resources to train a new classifier. Our detector is designed both to detect whether the classifier has been attacked or not and to infer the source class(es) and the target class if an attack is detected.

4.1 Basis of Our Detection. Our detection is based on two properties of SPP BPs that we will experimentally show to hold in practice.

Property 1: Spatial invariance of backdoor mapping. If the perceptible BP is spatially distributed (not at a fixed location) in the backdoor poisoned training images (placed so as to be most scene-plausible in each image), the learned backdoor mapping will be spatially invariant in inducing targeted misclassifications on test images.

As addressed in section 2.2.1, the perceptible BP could be spatially located anywhere in a training image so as to be scene-plausible. The trained DNN will then learn the BP but not its spatial location. This property is actually favored by the attacker, since there will be more freedom to embed the BP in clean test images. As will be seen in section 5.1, if the BP is spatially distributed in training images, at test time, backdoor images with the BP randomly located are still likely to yield misclassifications to the target class prescribed by the attacker. This justifies the attacker performing data poisoning so as to satisfy this property. In section 5.7, we provide deeper insights regarding this property experimentally.

Property 2: Robustness of perceptible BPs. The learned backdoor mapping is robust to variations such as noise, lighting, view, and illumination, thus obviating the need for an attacker to use precisely the same perceptible BP in all training and test images.

Again, this property is favored by a practical attacker. For a physically implemented backdoor pattern, during testing, though the same object (e.g., the same pair of glasses) is used, it is not guaranteed that the same digital pattern will be captured (due to, say, the light condition or the angle of the object). As will be seen in section 5.4, even when strong noise is added to the BP while creating backdoor test images, they will still likely be (mis)classified to the designated target class. More important, if only part (in terms of spatial support) of the BP is added to source class images during testing (the pattern could be partially occluded), this also induces (mis)classification to the target class. This is not surprising because perceptible BPs, like other patterns, are learned by the DNN by extracting key features, which usually occupy a smaller spatial support than the full pattern support.

4.2 Detection Overview. Key ideas: (1) For an SPP BA on a DNN, for effective backdoor (source, target) class pairs (including those used for devising the attack and those caused by collateral damage), finding a pattern that induces a high misclassification fraction (from the source class to the

Downloaded from http://direct.mit.edu/neco/article-pdf/doi/10.1162/neco_a_01376/1889374/neco_a_01376.pdf by guest on 19 March 2021
target class) on the clean data set can be achieved using a relatively small spatialsupport that is arbitrarily located in the image. (2) For non-backdoor class pairs, much larger spatial support is required to find a pattern that induces high misclassification from the source to the target class.

The key ideas of our detection are well supported by the properties in section 4.1. For convenience, for any class pair \((s, t) \in C \times C \) \((s \neq t)\) and a set \(D_s\) of clean images that are correctly classified to \(s\) by the classifier \(f(\cdot; \theta)\) to be inspected, we define

\[
\xi_{st}(v, m) = \frac{1}{|D_s|} \sum_{x \in D_s} \mathbb{1}(f(g_P(x, v, m); \theta) = t) \tag{4.1}
\]

as the misclassification fraction from \(s\) to \(t\) induced by pattern \(v\) and mask \(m\).

When there is a BA with SPP BPs, the perceptible BP \(v^*\) will be spatially distributed in the backdoor poisoned training images (as discussed in section 2.2.1). Consider a backdoor class pair \((s, t^*)\) with \(s \in S^*\) or a class pair suffering collateral damage (see section 2.2.2). Based on property 1, any mask \(m^\prime\) having the same shape and size as the true mask used for devising the attack in terms of the region of 1’s, \(\xi_{st^*}(v^*, m^\prime)\) is expected to be large (close to 1) no matter where the center of the region of 1’s of \(m^\prime\) is located in the images from \(D_s\). Based on property 2, there exists a pattern \(v^\prime\) (with associated mask \(m^\prime\)) different from the true BP \(v^*\), such that \(\xi_{st^*}(v^\prime, m^\prime)\) is large, so long as sufficiently “representative” features (e.g., key textures, colors) of \(v^*\) are captured in its spatial support. Such a pattern \(v^\prime\) may have either larger or smaller support than \(v^*\).

Thus, for any arbitrarily located spatial support with a relatively small support size (specified by a given mask \(M \in \{0, 1\}^{W \times H}\) with \(\|M\|_0\) small and the center of the region of 1’s arbitrarily located and fixed for all images), we expect the optimal misclassification fraction obtained by solving

\[
\text{maximize}_{v \in [0,1]^{W \times H \times C}} \xi_{st}(v, M) \tag{4.2}
\]

for \((s, t)\) a backdoor class pair to be much larger than for \((s, t)\) a non-backdoor class pair. Motivated by this, our detector consists of a pattern estimation step (performed for each class pair) and a detection inference step.

However, choosing an optimal \(M\) for pattern estimation to best distinguish backdoor class pairs from non-backdoor class pairs is not trivial. If \(\|M\|_0\) is too small, pattern estimation may be inadequate to capture the representative features of the BP for backdoor class pairs. Then the resulting misclassification fraction for all backdoor class pairs will be too low to
trigger a detection. If $||M||_0$ is overly large, the representative features of a non-backdoor target class may be captured by chance, such that a high group misclassification fraction can be achieved for some class pairs with this non-backdoor target class, leading to a false detection. This will be shown experimentally in section 5.2. Thus, since the support size of an attack is unknown to the defender, we suggest choosing multiple spatial supports, with a range of support sizes growing from small to large. For each class pair, pattern estimation is performed on each of these spatial supports. Then the pattern estimation results for these supports are suitably aggregated, for each class pair, for detection inference. The details of our detection are described in the sequel.

4.3 Pattern Estimation. Our pattern estimation step thus solves equation 4.2 for a sequence of spatial supports increasing in support size.

4.3.1 Design Choices.

Support location. Since it does not matter where the estimated pattern’s spatial support is placed, in our experiments (based on property 1), we fixed a location for all clean images used in our detection system. However, it is even possible that one could vary the center position for each clean image; accurate backdoor detection should still be possible.

Support shape. The choice of the support shape is not critical to the detection performance, but we prefer to use a simple convex support (e.g., a rectangle or a circle) to efficiently capture the representative features of the target class. Here, we choose square as the support shape for simplicity. Note that the choice of the support shape in detection does not need to be matched with the shape of the region of 1’s for the mask used by the attacker (which is not known to the defender a priori). In our experiments, we consider attacks with BFs having contiguous nonconvex shapes, or even highly distributed supports (see section 5.9). These attacks are all detected by our detector using simple square spatial support for pattern estimation.

Objective function and optimizer. Note that the misclassification fraction $\xi(st, v, m)$ for class pair $(s, t)$ on $D_s$ is not differentiable in $v$. We propose the following surrogate objective function:

$$\bar{\xi}st(v, m) = \frac{1}{|D_s|} \sum_{x \in D_s} \mathbb{E}[1(f(gP(x, v, m); \theta) = t)]$$

$$= \frac{1}{|D_s|} \sum_{x \in D_s} p(t|gP(x, v, m); \theta).$$

(4.3)

which is both the average misclassification confidence from $s$ to $t$ on $D_s$ and the average DNN class posterior for class $t$, with the average over all images in $D_s$. Thus, maximizing $\bar{\xi}st(v, m)$ over $v \in [0, 1]^{W \times H \times C}$ can be achieved
using a gradient-based optimizer with projection (i.e., clipping to the valid range for pixel intensity, \([0, 1]\)). For example, one can use stochastic gradient descent (SGD) with momentum and adaptive learning rate, where in each step of updating \(v\), the gradient of (the negative of) equation 4.3 is computed on a subset of (i.e., a mini-batch from) \(D_s\). Note that the choices of the objective function and the optimizer are not unique in order to achieve good detection performance. One can also consider a surrogate objective function similar to that associated with the perceptron algorithm (Duda, Hart, & Stork, 2001), or take a logarithm of the classifier’s posterior in equation 4.3 for better smoothness (Wang et al., 2019; Xiang, Miller, & Kesidis, 2020c, 2020d). These choices do not have a significant effect on the accuracy of our detector. The detailed choices used in our experiments are given in section 5.2.

4.4.2 Pattern Estimation Details. Based on the discussion of the design choices, we first create \(L\) different masks. Each mask specifies a square spatial support (region of 1’s) with fixed location (e.g., the top left corner of each image). The number \(L\) and the width of the square spatial support associated with each mask are specified as follows. Assuming the image width \(W\) is no larger than the image height \(H\) (otherwise, rotate the image), we choose a minimum relative support width \(r_{\min} \in [0, 1]\) and a maximum relative support width \(r_{\max} \in [0, 1]\). Then \(W = \mathbb{Z}^+ \cap [r_{\min}W, r_{\max}W]\) is the set of integer support widths to be considered with \(L = |W|\). For example, for \(32 \times 32\) colored images \((W = H = 32)\), with \(r_{\min} = 0.15\) and \(r_{\max} = 0.2\), we consider \(L = 2\) masks with support widths 5 and 6, respectively. As we have discussed, the purpose in considering a range of support widths instead of a single one for pattern estimation is for more reliable detection because the true support of the BA is unknown. This need will be borne out experimentally in section 5.2. In principle, a higher misclassification fraction can be achieved for backdoor class pairs than for non-backdoor class pairs when pattern estimation is performed on even very small spatial supports. Thus, we still require \(r_{\max}\) to be not too large. The choices of \(r_{\min}\) and \(r_{\max}\) in our experiments are discussed in section 5.2.

For each of the \(K(K - 1)\) (source, target) class pairs, we perform \(L\) pattern estimations. That is, for each \((s, t) \in \mathcal{C} \times \mathcal{C} (s \neq t)\) and \(w \in \mathcal{W}\), we solve (using, e.g., the previously suggested optimizer):

\[
\max_{v \in [0, 1]^{W \times H \times C}} \xi_{st}(v, M_w).
\]

where \(M_w \in [0, 1]^{W \times H}\) is the mask with fixed \(w \times w\) square spatial support specified by locations with 1’s. Compared with NC, which jointly estimates a pattern and its support mask, our pattern estimation problem (with the mask fixed) is less formidable.
4.4 Detection Inference. For each class pair \((s, t)\) and support width \(w \in \mathcal{W}\), we define an associated maximum achievable misclassification fraction (MAMF) statistic as
\[
\rho_{stw} = \xi_{st}(v_{stw}, M_w),
\]
where \(v_{stw}\) is the optimal pattern obtained from solving equation 4.4 for class pair \((s, t)\) and support width \(w\). There are many possible ways to aggregate the pattern estimation results over the \(L\) spatial supports for each class pair (e.g., median, geometric average, maximum, arithmetic average) to obtain a detection statistic. Here, we compute an average MAMF \(\bar{\rho}_{st} = \frac{1}{L} \sum_{w \in \mathcal{W}} \rho_{stw}\) over the \(L\) MAMF statistics for each class pair for simplicity and leave other possible aggregation techniques to future work. For a relatively small \(r_{\text{max}}\) (and the \(L\) associated small spatial supports), if there was an attack, we would expect at least one true backdoor pair to have a large average MAMF (whether one or multiple source classes are involved in the attack); otherwise, the maximum \(\bar{\rho}_{st}\) for all \((s, t)\) pairs is expected to be small. Hence, we infer the DNN to be attacked if
\[
\rho^* = \max_{(s, t) \in \mathcal{C} \times \mathcal{C}, s \neq t} \bar{\rho}_{st} > \pi;
\]
else, the DNN is not attacked. In section 5.2, we confirm experimentally that our approach is effective for a sizable range of threshold \(\pi\). If an attack is detected, \((\hat{s}, \hat{t}) = \arg\max_{(s, t) \in \mathcal{C} \times \mathcal{C}, s \neq t} \bar{\rho}_{st}\) is inferred as one (source, target) class pair involved in the BA.

4.5 MAMF Correction Using Class Confusion Information. Pattern estimation for a non-backdoor (source, target) class pair may produce a large MAMF statistic on a small spatial support if the two classes are similar to each other (i.e., have high class confusion). To avoid false detections caused by this, one may build knowledge of confusion matrix information into the detector. For each \((s, t) \in \mathcal{C} \times \mathcal{C}\), we define a baseline class confusion between \(s\) and \(t\) as
\[
\rho_{st0} = P[1(f(X; \theta) = t) | y(X) = s],
\]
where \(P[\cdot | \cdot]\) denotes conditional probability; \(X \in \mathcal{X}\) denotes a random image; and \(y: \mathcal{X} \rightarrow \mathcal{C}\) denotes oracle labeling of an image. The class confusion information may be provided to the defender in advance or empirically estimated using a relatively abundant set of clean images.\(^5\) Then the average
\(^5\)Pattern estimation for each class pair is still performed using only a subset of clean images that are correctly classified to the source class.
MAMF $\bar{\rho}_{st}$ for $(s, t)$ can be obtained by averaging over $L$ “corrected” MAMF statistics, each defined by

$$\rho_{stw}^{(c)} = \max\{0, \rho_{stw} - \rho_{st0}\}. \quad (4.8)$$

for each $w \in W$. When there is no attack, the above correction will prevent class pairs with high class confusion from having overly high MAMF statistics, as will be seen experimentally in the sequel.

4.6 Computational Complexity. For our detection, the computation is mainly incurred by forward and backward propagations during the pattern estimation step. Suppose $N$ clean images per class are used for detection and pattern estimation converges in no more than $T$ iterations for all class pairs. The computational complexity of our detector is bounded by $O(NTLK^2)$ single image feedforward passes through the DNN. The computational complexity for feeding forward a single image to the DNN depends on the architecture of the DNN to be inspected, which is not determined by a posttraining defender.

5 Experiments

5.1 Devising BAs. We demonstrate the validity of the properties in section 4.1 and the performance of our detector using nine attacks, involving five data sets, five DNN structures, and nine BPs. The data sets and the DNN structures considered here are commonly used in computer vision research. Other popular data sets (e.g., Caltech101, ImageNet), are not used in our experiments since the overall test accuracy or the accuracy of particular classes is fairly low.

For each attack, we first trained a benchmark DNN using an unpoisoned training set and report the accuracy on clean test images as the benchmark accuracy. The data set being used, image size, number of classes, training size, and test size are shown in Table 1. In particular, for attack G on Oxford-IIT, the test accuracy using the entire data set, in the absence of BAs, is very low. We achieve a reasonable benchmark test accuracy on G by using a subset involving only 6 classes. For attack I on PubFig, many images are not able to be downloaded; hence we only consider 33 (out of 60) classes with more than 60 images. The DNN structures involved in our experiments include ResNet-18, ResNet-34 (He, Zhang, Ren, & Sun, 2016), VGG-16 (Simonyan & Zisserman, 2015), AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), and ConvNet (Sermanet, Chintala, & LeCun, 2012). The choices of DNN structure, learning rate, batch size and number of training epochs for each attack instance are also shown in Table 1. The Adam optimizer (with a decay rate of 0.9 and 0.999 for the first and second moment, respectively) is used for all DNN training in this letter. For the benchmark training for the
Table 1: Details of the Attacks.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Image size</td>
<td>CIFAR-10</td>
<td>CIFAR-10</td>
<td>CIFAR-10</td>
<td>CIFAR-10</td>
<td>CIFAR-10</td>
<td>CIFAR-10</td>
<td>Oxford-IIIT</td>
<td>SVHN</td>
<td>PubFig</td>
</tr>
<tr>
<td>No. classes</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Training size</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>900</td>
<td>73257</td>
<td>2782</td>
</tr>
<tr>
<td>Test size</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>300</td>
<td>26032</td>
<td>495</td>
</tr>
<tr>
<td>Learning rate</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>$10^{-4}$</td>
<td>$10^{-5}$</td>
<td>$10^{-3}$</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Batch size</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>16</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>No. training epochs</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>120</td>
<td>80</td>
<td>120</td>
</tr>
<tr>
<td>Benchmark acc. (%)</td>
<td>86.7</td>
<td>88.1</td>
<td>86.7</td>
<td>87.6</td>
<td>87.9</td>
<td>71.9</td>
<td>88.7</td>
<td>89.2</td>
<td>76.0</td>
</tr>
<tr>
<td>Source class</td>
<td>“cat”</td>
<td>“dog”</td>
<td>“airplane”</td>
<td>“frog”</td>
<td>“truck”</td>
<td>“chihuahua”</td>
<td>“3”</td>
<td>“B. Obama”</td>
<td>“C. Ronaldo”</td>
</tr>
<tr>
<td>Target class</td>
<td>“dear”</td>
<td>“horse”</td>
<td>“bird”</td>
<td>“bird”</td>
<td>“bed”</td>
<td>“Abyssinian”</td>
<td>“8”</td>
<td>“8”</td>
<td>“8”</td>
</tr>
<tr>
<td>BP</td>
<td>“bug”</td>
<td>“butterfly”</td>
<td>“rainbow”</td>
<td>“bug&amp;butterfly”</td>
<td>“gas tank”</td>
<td>“tennis ball”</td>
<td>“bullet holes”</td>
<td>“sunglasses”</td>
<td></td>
</tr>
<tr>
<td>No. backdoor training images</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>100</td>
<td>50</td>
<td>500</td>
</tr>
<tr>
<td>Attack test acc. (%)</td>
<td>87.0</td>
<td>86.9</td>
<td>86.8</td>
<td>87.0</td>
<td>89.1</td>
<td>71.7</td>
<td>90.0</td>
<td>98.1</td>
<td>72.0</td>
</tr>
<tr>
<td>Attack succ. rate (%)</td>
<td>99.3</td>
<td>98.0</td>
<td>96.4</td>
<td>98.0</td>
<td>97.9</td>
<td>92.0</td>
<td>84.0</td>
<td>91.4</td>
<td>93.3</td>
</tr>
</tbody>
</table>
data sets involving attacks F, G, and I, we also use training data augmentation including random cropping, random horizontal flipping, and random rotation. For the benchmark training for the data set involving attack G, we fine-tune a pretrained AlexNet provided by Pytorch. For the benchmark training for the data set involving attack I, we adopt transfer learning by retraining the last four layers of a pretrained VGG-face model (Albanie, 2021).

Under each attack, we train the DNN using the same training settings as for the benchmark, except that the training set is poisoned by a number of backdoor images. The backdoor images are created using clean images from one source class, with a BP added following equation 2.1 (but with an image-specific mask for scene plausibility), and then labeling to a target class. In the experiments in this section, we chose to evaluate our detector against BAs involving one source class for convenience in easily crafting SPP BPs. As we discussed in section 4.4, the design of our detector allows it to detect BAs with any number of source classes, since we only need one class pair to have a sufficiently large average MAMF to make a detection. In section 5.11, we will show the effectiveness of our detector against BAs involving $(K - 1)$ source classes, even though the BPs may not be scene plausible (e.g., a rainbow will be placed in an image without the sky). For the experiments in this section, the choices of the BP, the number of backdoor training images, the source class, and the target class for each attack instance are shown in Table 1.

To create BAs with SPP BPs, we first created a large number of candidate backdoor images, each with the BP randomly located in the image. Then we spent laborious human effort to manually pick the images in which the BP looks scene plausible. For example, for attack C (source class “airplane” and BP a “rainbow”), a valid backdoor poisoned training image should have the rainbow in the sky (see Figure 3c). In Figure 3, we show an example backdoor training image and its original clean image for each attack. The BPs considered here are representative of multiple types of practical BPs. The bug for attack A and the butterfly for attack B represent BPs in the periphery (not covering the foreground object of interest) of an image and with a modest size. The rainbow for attack C represents large BPs with an irregular shape. The BPs for attacks D and H represent dispersed patterns. The sunglasses for attack I represent BPs overlapping with features of interest.

In Table 1, we also report the accuracy on clean test images and the attack success rate for all attacks, defined as the fraction of backdoor test images being classified to the target class prescribed by the attacker. A backdoor test image is created using a clean test image from the source class(es) and embedding in it the same BP used for creating the backdoor-poisoned

---

6 Only for attack I we carefully place the sunglasses on the face.
Detecting Scene-Plausible Perceptible Backdoors

Figure 3: Example backdoor image and the originally clean image for attacks A to I. Subcaptions describe the object(s) added as the perceptible BP for each attack.

(a) bug   (b) butterfly  (c) rainbow  (d) bug+butterfly  (e) gas tank

(f) marmot  (g) tennis ball  (h) bullet holes  (i) sunglasses

Figure 3: Example backdoor image and the originally clean image for attacks A to I. Subcaptions describe the object(s) added as the perceptible BP for each attack.

training images. We took an automated approach (due to the huge number of test images) to create backdoor test images, randomly placing the BP in the image (except for attack I, where we manually place the sunglasses on the faces). By doing so for a sufficiently large test set, we should be covering most locations where the attacker could place the BP in practice. In Table 1, the attack success rate is high for all attacks, and there is no significant degradation in clean test accuracy compared with the benchmark; hence all attacks are considered successful. Moreover, we emphasize that such success is achieved with randomly placed BPs in test images, which experimentally verifies property 1, that is, the spatial invariance of the learned SPP backdoor mapping. More experiments regarding property 1 will be shown in section 5.7.

5.2 SPP Backdoor Performance Evaluation. In this section, we evaluate the performance of the proposed detector in comparison with the state-of-the-art detector NC (Wang et al., 2019) and an even earlier defense FP (Liu, Doan-Gavitt et al., 2018), using the above-mentioned attacks, that is, BAs
with SPP BPs where the BP is not in a fixed position in either the poisoned training images or the backdoor-attacked test images.

For each attack, detection is applied to both the DNN being attacked and the clean benchmark DNN. For best discrimination between the two categories of DNNs, for our proposed method, relatively small spatial supports should be used for pattern estimation, so that a DNN being attacked can have a much larger $\rho^*$ than a clean DNN. On the one hand, the spatial support should be smaller than that of the foreground objects associated with the actual classes; otherwise, the actual target class object(s) could possibly be reverse-engineered, causing high group misclassification even for a clean DNN. On the other hand, the spatial support should be large enough for estimating any pattern related to the BP. Here, we show that with a common $r_{\text{min}}$ and a common $r_{\text{max}}$ applied to all the classifiers to be inspected regardless of the associated data set, there is a large range of valid thresholds for achieving perfect detection of all the attacks, without any false detections. We choose $r_{\text{max}} = 0.22$ and $r_{\text{min}} = 0.08$ for all attack instances, such that for a $32 \times 32$ image, there is at least $3 \times 3$ spatial support for pattern estimation. In section 5.3, we will present an automatic approach for choosing adaptive $r_{\text{min}}$ and $r_{\text{max}}$ for each classifier to be inspected.

The details of our detection settings are as follows. For attacks A to E and H, 200 clean images correctly classified per class are used for detection. For attacks F, G, and I, 50, 30, and 9 correctly classified clean images per class are used for detection, respectively. For all DNNs, class pairs and spatial supports, we solve equation 4.4 using Adam optimizer with decay rate 0.9 and 0.999 for the first and second moments, respectively, and learning rate 0.5 for 100 epochs. We use mini-batch size 32 for attacks A to E and H, 10 for attacks F and G, and 3 for attack I. The spatial support for pattern estimation, used in our detection, is a square covering the top left corner of each image. Due to property 1 (as will be further investigated in section 5.7), other locations yield similar results, as will be illustrated in section 5.8.

In Figure 4, for both the DNN being attacked and the clean benchmark DNN, for each attack, we show the $(L)$ MAMF statistics for the class pair with the largest average MAMF (i.e., the class pair corresponding to $\rho^*$). For example, for attack A where the data image size is $32 \times 32$, the set of (absolute) support widths being considered is $W = \{3, 4, 5, 6, 7\}$. For large image size, instead of performing pattern estimation for each integer support width in the interval $[r_{\text{min}}W, r_{\text{max}}W]$, we could efficiently downsample and perform pattern estimation for fewer support widths. Here, $L = 9$ and $L = 7$ support widths are considered for attacks G (see Figure 4g) and I (see Figure 4i), respectively. In each panel of Figure 4, a large gap is observed between the clean and attacked curves, clearly distinguishing the DNN being attacked from the clean DNN.

---

7 Fewer epochs are actually needed for decent convergence in all our experiments.
Detecting Scene-Plausible Perceptible Backdoors

Figure 4: Maximum achievable misclassification fraction (MAMF) statistics for the class pair with the largest average MAMF, for both attacked DNN and unattacked (clean DNN), for Attacks A to I. Relative support range for pattern estimation is (0.08, 0.22).

Also from Figure 4, we see the importance of aggregating the pattern estimation results (i.e., the MAMF statistics) over a range of support sizes. Unless there is a way to select a single support width that ensures a significant difference between the largest MAMF statistic for the classifier being attacked and for the clean classifier, these two categories of classifiers may not be distinguishable for some casually selected support widths. For example, for the absolute support width 55 (relative support width 55/128)
for attack I, the largest MAMF statistics for the classifier being attacked and for the clean classifier are the same (see Figure 4i).

In Figure 5, we show the largest average MAMF ($\rho^*$) for both the clean and attacked DNNs, for all attacks. Note the clear large difference between the two bars for each attack. Thus any detection threshold $\pi$ in (0.6, 0.8), if used, could successfully detect whether the DNN is attacked. Moreover, for the DNN being attacked, for all attack instances, the class pair corresponding to the maximum average MAMF is precisely the true backdoor pair. Hence, when a detection is made, the (source, target) class pair (used by the attacker) is also correctly inferred, in all cases.

Here, we also consider the extension of our detection approach where MAMF statistics are corrected using class confusion information (see section 4.6). For each of attacks B, F, and H, we estimate an empirical class confusion matrix using all the test images for the data set associated with the attack (see Table 1 for data set information). For each attack, we apply our detection with MAMF correction to both the clean benchmark DNN and the DNN being attacked. In Figure 6, we show the largest average MAMF $\rho^*$ with correction, compared with $\rho^*$ without correction, for both clean and attacked DNNs, for the three attacks. Although MAMF correction does not significantly affect $\rho^*$ for the six classifiers, the decrement in $\rho^*$ for the clean DNN is slightly larger than for the DNN being attacked, for all three attacks. In other words, assisted by the class confusion information, the range of valid thresholds is slightly larger. Note that in this experiment, we used abundant images to ensure the accuracy of our confusion matrix.
Detecting Scene-Plausible Perceptible Backdoors

Figure 6: Largest average maximum achievable misclassification fraction (MAMF), $\rho^*$, with MAMF correction, compared with $\rho^*$ without MAMF correction, for both the attacked DNN and unattacked (clean) DNN, for attacks B, F, and H, respectively.

estimation, though a practical defender may not have access to sufficient images to accurately estimate the confusion matrix. Nevertheless, we have already shown accurate detection can be achieved by our detector without the use of the class confusion matrix. Thus, in all the experiments that follow, we do not use MAMF correction for simplicity.

For the attacks in the current experiment (which use a single source class), NC may not make a correct detection since it is based on the assumption that all classes other than the target class are involved in the attack. Also, NC jointly estimates a pattern and an associated mask for each putative target class to induce at least $\phi$-level misclassification. Such optimization relies on the choice of the penalty multiplier $\lambda$ (for the $L_1$-regularization of the mask) and the training settings. Here we evaluate NC using the same attacks. We used the Adam optimizer to solve NC’s optimization problem, with the parameters suggested by the authors of NC, and performed minibatch optimization for a sufficient number of epochs (until convergence). If $\lambda$ is too large, $\phi$-level group misclassification cannot be achieved since the mask “size” is over-penalized. If $\lambda$ is made small, a high group misclassification fraction will be achieved, but the $L_1$ norm of the mask is unreasonably large. Hence, we carefully adjust $\lambda$ and the training parameters for each attack to achieve a mask with a small $L_1$ norm and a pattern inducing a high
Table 2: Detailed Settings of NC.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Number of Images per Class</th>
<th>$\lambda$</th>
<th>Learning Rate</th>
<th>Batch Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>A–D</td>
<td>100</td>
<td>0.1</td>
<td>0.05</td>
<td>90</td>
</tr>
<tr>
<td>E</td>
<td>100</td>
<td>0.6</td>
<td>0.05</td>
<td>90</td>
</tr>
<tr>
<td>F</td>
<td>10</td>
<td>0.1</td>
<td>0.05</td>
<td>90</td>
</tr>
<tr>
<td>G</td>
<td>30</td>
<td>0.5</td>
<td>0.001</td>
<td>60</td>
</tr>
<tr>
<td>H</td>
<td>100</td>
<td>0.2</td>
<td>0.01</td>
<td>100</td>
</tr>
<tr>
<td>I</td>
<td>9</td>
<td>0.1</td>
<td>0.005</td>
<td>30</td>
</tr>
</tbody>
</table>

Note: Included are the number of clean images per class used for detection, the choice of $\lambda$, the learning rate, and the mini-batch size for each attack instance.

For all attack cases, we set $\phi = 0.9$, and the optimization is performed for 200 epochs. Table 2 shows the number of clean images per class used for pattern estimation by NC, the choice of $\lambda$, the learning rate, and the mini-batch size for each attack.

NC’s detection inference uses MAD (Hampel, 1974) of the $L_1$ norm of the mask for all putative target classes (Wang et al., 2019). If the anomaly index is larger than 2.0, a detection is made with 95% confidence. In Figure 7, we show anomaly indices for both the DNN being attacked and the clean DNN for each attack. Only for attacks B and G does NC successfully detect
Detecting Scene-Plausible Perceptible Backdoors

Figure 8: Receiver operating characteristic (ROC) curves for NC and our detector against the nine attacks.

the attack. For these attacks, there is a pattern and an associated relatively small mask that, when applied to clean test images from all classes other than the target class, induces a high group misclassification (even though the backdoor targeted only a single source class). The phenomenon where non–source class images are with high probability misclassified to the target class when the BP is added to them was discovered and identified in Xiang et al. (2020b) as “collateral damage” (see section 2.2.2). Note that for attack A, the anomaly index for the clean DNN is larger than for the backdoor-attacked DNN.

For better visualization of the performance comparison between our detector (without MAMF correction) and NC, we show the receiver operating characteristic (ROC) curves for the two approaches against the nine attacks in Figure 8. The areas under curve (AUC) for our detector and NC are 1.0 and 0.78, respectively.

We also show FP (Liu, Doan-Gavitt et al., 2018) is ineffective against most of the attacks. We prune the penultimate layer neurons of each classifier being attacked (until only few are left), in increasing order of their average activations over all clean test images. In Figure 9, for each attack, we show the accuracy on clean test images and the attack success rate versus the number of neurons pruned. For most of the attacks (except attacks E and H), the attack success rate does not drop before the accuracy on clean test images is degraded, as the number of pruned neurons grows. Thus, FP is generally unsuccessful against these BAs.
5.3 Adaptive Selection of $r_{\text{min}}$ and $r_{\text{max}}$. We show a practical way to choose an adaptive $r_{\text{min}}$ and $r_{\text{max}}$ combination for any classifier to be inspected (without knowing whether the classifier is attacked or not a priori). This approach is actually matched with the intuition of choosing $r_{\text{min}}$ and $r_{\text{max}}$ discussed previously. We start with pattern estimation on a $1 \times 1$ square spatial support for each class pair. While increasing the absolute width of the square spatial support for pattern estimation, at the first instance when any class pair achieves a moderate MAMF (e.g., 0.5), we set $r_{\text{min}}$ to be this absolute support width divided by the image width. Intuitively, at this point,
Detecting Scene-Plausible Perceptible Backdoors

Table 3: Adaptive Minimum and Maximum Support Width Determined for Both the Classifier Being Attacked and the Clean Benchmark Classifier for Attacks A to I, excluding Attack F.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Minimum Width</th>
<th>Maximum Width</th>
<th>Minimum Width</th>
<th>Maximum Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack A</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Attack B</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Attack C</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Attack D</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Attack E</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Attack G</td>
<td>11</td>
<td>13</td>
<td>19</td>
<td>25</td>
</tr>
<tr>
<td>Attack H</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Attack I</td>
<td>30</td>
<td>45</td>
<td>35</td>
<td>40</td>
</tr>
</tbody>
</table>

the corresponding spatial support is sufficiently large to reverse-engineer partially “representative” features of the true BP. Then we continue to increase the support size until there are at least two different target classes involved among all the class pairs whose MAMF statistic is larger than the MAMF threshold. We then set $r_{max}$ to be the corresponding absolute support width divided by the image width. Also, since a BA is assumed to involve only one target class, we know that there is at least one non-backdoor class pair having a moderate MAMF statistic; the support size should not be further increased.

For brevity, we do not thoroughly evaluate the above approach. Instead, we only consider the support widths that have been considered in our previous experiments (see Figure 4 and the related description). Here, we set the MAMF threshold for determining the adaptive minimum and maximum support sizes to be 0.5. In Table 3, we show the minimum and maximum absolute support width determined for both the classifier being attacked and the clean benchmark classifier for each of attacks A to I (except attack F). One can relate the results in Table 3 with the results in Figure 4 to better understand the intuition behind our adaptive selection approach for $r_{min}$ and $r_{max}$. Take attack B as an example. For the classifier being attacked, for the absolute support width 4 (relative support width 4/32), although the MAMF threshold 0.5 is achieved by some backdoor class pair, the MAMF statistics for all non-backdoor class pairs are low. From Table 3, as the absolute support size grows to 6 (relative support width 6/32), for

---

8 For brevity, we only consider support widths that we considered in section 5.2, with relative support width in [0.08, 0.22]. However, as shown in Figure 4f, for the clean classifier, a 0.5 misclassification fraction is not achieved for any support width we considered. Although such misclassification fraction will finally be achieved if we further increase the relative support width (beyond 0.22), we do not include the results here.
the first time, the MAMF statistic for a non-backdoor class pair exceeds the threshold 0.5. As the absolute support width grows from 4 to 6, the MAMF statistic corresponding to the true backdoor class pair quickly approaches 1.0 and stays high (see Figure 4a) as one would expect for the true backdoor class pair. For the clean benchmark classifier, the first instance for any class pair’s MAMF statistic to be greater than 0.5 is when the absolute support width is 5 (relative support width 5/32). During the growth of the absolute support width from 5 to 7 (where there is another class pair whose MAMF statistic is greater than 0.5), there is a significant gap between the largest MAMF statistic among all the class pairs and 1.0. Thus, using this adaptive criterion, the largest average MAMF for the clean classifier is clearly less than for the classifier being attacked. In Figure 10, we observe this difference between the largest average MAMF for the clean classifier and for the classifier being attacked for attacks A to E, G, and I. There is still a range of valid thresholds that detect most of the attacks.

5.4 Verification of Property 2. Here we verify property 2, the robustness property of perceptible BPs, from two aspects. First, for each attack, we modify the BP embedded into each clean test image (from the source class) by adding Gaussian noise $N(0, \sigma^2)$ to each pixel (and then clipping each pixel value to $[0, 1]$). Second, instead of adding noise, we crop part of the BP outside its center before embedding to the clean test images. We
evaluate the attack success rate for noisy BPs with $\sigma^2 = 0.01, 0.25, 1$ and cropped BPs to 64% and 36% of the original size in Table 4. For most attacks, using noisy or cropped BPs at test time still achieves a high attack success rate (even though cropping may remove critical features on the periphery of the BP).

5.5 Number of Images for Detection. We note that if too few clean images are available to the defender, the performance of our detector could be affected. When there are no attacks, a false detection may be made since it is much easier to find a pattern that induces a high misclassification rate for a small group, compared with a larger one. This has been essentially verified in Moosavi-Dezfooli, Fawzi, and Frossard (2017) as a “universal” TTE perturbation (a common TTE perturbation applied to all images) inducing high group misclassification needs, a much larger norm (and full spatial support) than a perturbation inducing single-image misclassification at test time. In Figure 11, we show the MAMF statistics for the class pair with the largest average MAMF for the clean and attacked DNNs for attack A, with 5, 10, 25, 50, 100, and 200 clean images per class used for pattern estimation. When the number of clean images used for detection is greater than 10, there is a clear gap between the two curves for the DNN being attacked and its clean counterpart. But if fewer images are used for detection, the curve corresponding to the clean DNN approaches 1.0 quite quickly, that is, achieving a high group classification fraction becomes easier with a small spatial support. This is the same phenomenon as in Figure 4i for attack I, where the curve for the clean DNN quickly achieves 1.0 as the support width for pattern estimation increases, since only nine clean images are used for detection. Note, though, that our detector is successful even in this case: there is still a large gap between the curves.
Figure 11: Maximum achievable misclassification fraction (MAMF) statistics for the class pair with the largest average MAMF for a range of choices of relative support width $r$, for both DNNs being attacked and unattacked for attack A, when using 5, 10, 25, 50, 100, and 200 clean images per class for detection, respectively.

5.6 BP$\!\!$s with Fixed Spatial Location. As experimentally verified in Wang et al. (2019), when all classes except for the target class are the source classes, NC performs well when the BP is spatially fixed. Although our detector is not designed to detect BAs with spatially fixed BP, we show in the following that such attacks have poor robustness during testing when the classifier is trained without data augmentation. We also show that our detector can easily detect attacks with the perceptible BP spatially fixed (during its embedding into clean images), if the classifier is trained with simple data augmentation that modestly varies the spatial location of the BP in the poisoned images during training.

5.6.1 Poor Robustness. We first perform a simple experiment to show that the power or robustness of the attack will be degraded if the perceptible BP is spatially fixed when poisoning the clean training images. We use the same data set, training settings, (source, target) class pair, and BP under attack B. The only difference is that the backdoor training images are created by embedding the pattern, the butterfly, into the bottom left corner of every training image to be poisoned. Note that we do not use data augmentation during the classifier training.
Detecting Scene-Plausible Perceptible Backdoors

After training, the accuracy of the DNN on clean test images is 87.2%, similar to the accuracy of the clean benchmark DNN. Now we create four groups of images, 1000 each, using the clean test images from the attacker’s source class, such that the BP is embedded in the following ways:

- Into the bottom left corner (i.e., the same location as for the backdoor training images) of all images
- Spatially randomly in each image
- One row up from the bottom left corner of all images
- One column right of the bottom left corner of all images

Here we only focus on the spatial location of the BP without considering whether the BPs are scene plausible. The fractions (%) of images in each group that are (mis)classified to the target class prescribed by the attack are 99.7, 4.1, 47.8, and 14.4, respectively. Clearly, only if the BP during testing is located at the same location as during training, the backdoor image will be reliably (mis)classified to the target class. Hence, fixing the spatial location of the BP when creating backdoor training images largely degrades the robustness of the attack (and also affects its scene plausibility).

5.6.2 When Training Data Augmentation Is Used. Although our detector is not designed for attacks with spatially fixed perceptible BPs, we found our detector can actually detect this type of attack when data augmentation is used during the classifier’s training. Here, we use the same configurations as in section 5.6.1 to train a classifier on the same poisoned training set, where the butterfly is spatially fixed to the bottom left corner of every poisoned image. However, we use data augmentation, including random cropping, random horizontal flipping, and random rotation (±30°), during the training, such that the spatial location of the BP in the augmented training images will be modified to some extent. The attack success rate and the clean test accuracy for this DNN are 99.5% and 92.0%, respectively. Note that the images used to measure the attack success rate all have the BP fixed to the same spatial location as in the backdoor training images (prior to augmentation).

We apply our proposed detector with the same configurations as for attack B in section 5.2, where the fixed square spatial support for pattern estimation covers the top left corner of each image used for detection. Note that our training data augmentation does not include any vertical flipping; hence, none of the augmented training images will have the BP, the butterfly, located in the top left corner, coinciding with the location of the spatial support for our pattern estimation. In other words, the location of the spatial support for pattern estimation is not coinciding with the location of the BP in the poisoned training images in this experiment. In Figure 12, we compare the MAMF statistics for the class pair with the largest average MAMF for the above attacked DNN trained with data augmentation, with the MAMF statistics for the clean benchmark DNN for attack B in
Figure 12: Maximum achievable misclassification fraction (MAMF) statistics for the class pair with the largest average MAMF, for the attacked DNN trained with data augmentation, and the clean benchmark DNN for attack B.

A similarly large gap can be observed between the two curves: the largest average MAMF, $\rho^\star$, for the DNN being attacked (with the spatially fixed perceptible BP and training data augmentation in use) is as large as 0.87. The attack can be easily detected.

5.7 Digging Deep into Property 1. In previous experiments, we have shown that the learned backdoor mapping is spatially invariant if the BP embedded in the poisoned training images is random (see section 5.2), or with moderate randomness depending on the choice of training data augmentation (see section 5.6.2). However, there might be an extreme case where the attacker may embed the BP (for example) close to the bottom left corner of all poisoned training images (with only very limited shift for scene plausibility). Also, we assume that no data augmentation is used during training. In this case, will the learned backdoor mapping be spatially invariant? In other words, will the “rough” location of the embedded BP be learned as well? This is an interesting question because if property 1, the spatial invariance property, does not hold for this case, our detector will likely fail when the spatial support for pattern estimation does not match the rough region where the BP is embedded when devising the attack.

We created a $6 \times 6$ pattern as shown in Figure 13a. We choose it to be noisy to minimize the possibility for any other regions in the source class images to contain any representative features of the BP. In Figure 13, we also show an example backdoor training image and its original clean image.

We create six attacks using this pattern, with each attack associated with a specific integer maximum “offset” $\nu$. During the devising of each attack, the BP is embedded randomly inside the $(6 + \nu) \times (6 + \nu)$ window located
Detecting Scene-Plausible Perceptible Backdoors

Figure 13: BP used for (a) investigating property 1, (b) an example clean training image, and (c) the same image with the BP embedded.

Figure 14: Attack success rate measured using images with embedded BP randomly located for the six attacks with a range of maximum backdoor embedding spatial offset.

At the bottom left corner of the training images to be poisoned. In other words, supposing the pixel at the bottom left corner of the image has index (0, 0), then the index for the bottom left pixel of the BP is \((N_W, N_H)\), with \(N_W\) and \(N_H\) independently uniform on \([0, \ldots, \nu]\). Here, we consider \(\nu \in \{0, 2, 4, 6, 8, 10\}\). Note that for the attack with \(\nu = 0\), the BP is actually spatially fixed in all the backdoor training images. For the attack with \(\nu = 10\), the BP is randomly located in only the bottom left quarter of the image (given images size \(32 \times 32\)). The other details for the six attacks and training configurations are the same as for attack B (see Table 1).

Now, we check if property 1 holds using 1000 test images from the source class. For each originally clean test image, we embed the BP to a randomly chosen spatial location. In Figure 14, we show the attack success rate.
measured using these images for all six attacks. When the offset is no fewer than six, a source class image with the BP embedded will be classified to the target class with high probability, regardless of the spatial location of the BP. In these cases, it is the pattern itself instead of the rough location being learned during training on the poisoned training set. In conclusion, property 1 holds even when there is only moderate randomness in backdoor embedding when devising the attack.

5.8 Location of the Spatial Support for Pattern Estimation. According to property 1, the location of the spatial support in the pattern estimation step can be arbitrarily chosen. Previously, we fixed the spatial support to the top left corner for all clean images used for detection. Here, we experimentally verify that this choice is not critical to our detection. Again, we consider the same DNN being attacked and the same clean benchmark DNN under attack B. We apply the same detection to both the clean and attacked DNNs except that the spatial support for detection is fixed to cover (1) the top right corner, (2) the bottom left corner, and (3) the bottom right corner of all clean images used for detection. In Figure 15, for each location of the spatial support, we show the $L$ MAMF statistics (by varying the support width) for the class pair with the largest average MAMF for both DNNs. In each panel, we observe a large gap between the two curves, indicating that the DNN being attacked should be easily detected for a large range of thresholds. Thus, our detection approach is indeed robust to the chosen location of the spatial support of the pattern mask.

5.9 Detecting BAs with Highly Distributed BP. As described in section 4.3, we chose to use a contiguous, square spatial support for pattern
Detecting Scene-Plausible Perceptible Backdoors

Figure 16: An example image embedded with a highly distributed BP (b) and its original clean image (a).

estimation and emphasized that this choice does not need to be matched with the shape of the BP that the attacker used. Here, we evaluate our detector against a BA with highly distributed BP. Without considering the scene plausibility, the BP consists of six $2 \times 2$ noisy square patches randomly located in the image. Images with the BP embedded and its original clean image are shown in Figure 16. The BA is crafted using the same settings as for attack B, except that the BP is different. Also, the same configurations for classifier training for attack B are used here. The attack success rate and clean test accuracy for the trained classifier are 98.3% and 86.8%, respectively. Then we apply our detector with the same settings as for attack B (see section 5.2) to this classifier. In Figure 17, we show the MAMF statistics for the class pair with the largest average MAMF for this classifier being attacked, compared with the clean benchmark classifier trained for attack B. We observe an even larger gap between the two curves: the attack can be detected using an easily chosen threshold. Again, the effectiveness of
our detector is well supported by the properties we discovered in section 4.1. Even if the spatial support used for pattern estimation does not match the shape or the location of the BP used by the attacker, the representative features may still be successfully reverse-engineered.

5.10 Trivial Detection of BAs with Spatially Fixed BP. Here, we consider an extreme case where the attacker intends to evade our detection by deliberately setting the BP to be spatially fixed in the poisoned training images. We also assume there is no data augmentation used during classifier training. We further assist the attacker by assuming there is no human inspection of either training images or test images, such that the embedded BP does not need to be scene plausible. Then the attacker will be free to embed the BP at the same location in the test images coming from the source classes to induce targeted misclassification.

For this scenario, our detector may fail to detect the attack, but there are other very simple approaches to detect the attack with extremely low cost. For example, as we discussed in section 2.2.1, a before or during training detection can be deployed by checking the pixel value distribution. When there is an attack, a small set of images will have exactly the same pixel values for the same spatial region. Such a detector will not involve any heavy computation. If the defender is the consumer of the classifier without access to the training set, there is an even simpler way to detect BAs with spatially fixed BP. For each image to be classified during testing, the defender can horizontally or vertically flip the image. That will change the learned spatial location of the BP (unless the pattern happens to be symmetric and located on the horizontal or vertical center line of the image). However, for clean test images, such flipping will likely not affect its class prediction due to the basic generalization property of DNNs. Experimentally, for the classifier in section 5.6.1, by horizontally flipping the clean test images and test images with the BP, we reduce the attack success rate to 2.7% with only 0.5% degradation in clean test accuracy.

5.11 Detecting BAs with Multiple Source Classes. Here, we consider the case where all classes other than the target class are source classes. This case is favored by NC but is generally not practical for SPP BPs to be used, as we discussed in section 2.2.2.

We first evaluate our proposed defense for this case. We devise attacks A’, B’, and C’ using the same settings as attacks A, B, and C, respectively, but with all classes except the target class as source classes and 20 backdoor training images per source class. Note that the purpose of this experiment is to show that our detector does not rely on knowledge of the number of source classes involved in an attack. We do not require the BPs used in this attack experiment to be scene plausible. For example, for attack C’, we simply embed the BP, the rainbow, into random locations of images from classes like frog, dog, and cat, while most images from these classes do not contain
Detecting Scene-Plausible Perceptible Backdoors

Figure 18: Maximum achievable misclassification fraction (MAMF) statistics for a range of choices of relative support width $r$, for the class pair with the largest average MAMF, for attacks $A'$, $B'$, and $C'$, in comparison with the MAMF statistics for the clean benchmark DNNs for attacks A, B, and C.

The attack success rates (%) of attacks $A'$, $B'$, and $C'$ are 98.0, 95.3, and 98.7, respectively; the accuracies (%) on clean test images are 87.4, 86.8, 87.0, respectively. We apply our detector with the same setting as in section 5.2 to these three attacks, respectively, and show the MAMF statistics for the class pair with the largest average MAMF, for each attack, in Figure 18. The three attacks (with multiple source classes) are easily detected.

We notice that in Wang et al. (2019), NC is only evaluated against attacks with the perceptible BP spatially fixed across all backdoor training images. Here, with the setting that all classes except the target class are source classes, we further evaluate NC for (1) a spatially fixed perceptible BP, with data augmentation used during the classifier’s training and (2) a spatially variant BP.

For the first case above, we create an attack similar to the attack created in section 5.6.2. However, we create backdoor training images using clean images from all classes except the target class, with 20 images per source class. Again, the BP, the butterfly, is fixed to the bottom left in each backdoor training image. We use the same training configurations and data augmentation options as in section 5.6.2 and achieved a 99.9% attack success rate and 91.7% clean test accuracy. We apply NC using the same configuration as for attack B (as shown in Table 2) and obtain a high anomaly index 3.62 ($> 2$); a successful detection (with 95% confidence) is made. For the second case, we apply NC to the same attack $B'$, where the BP is randomly located in backdoor training images, without considering scene plausibility. We obtain a high anomaly index 6.16 ($> 2$). Again, NC successfully detects the attack (with 95% confidence). Thus, NC is successful when all classes other
than the target class are source classes, whether the BP is in a fixed location or variable. However, as seen by our experiments in section 5.2, NC is not highly successful; its performance is substantially worse than our proposed method, for the more practical setting where the SPP BA involves a single source class.

6 Conclusion and Future Work

In this letter, we proposed a detector for BAs with SPP BPs, posttraining, and without access to the (possibly poisoned) training set. Our detector, inspired by two properties of SPP BPs, is based on the maximum achievable misclassification fraction statistic. With an easily chosen threshold, our detector shows strong detection capability over a range of data sets and BPs.

There are potential improvements to our detector that could be studied in future.

6.1 Source Class Inference. Regardless of the number of source classes involved in devising the attack, our method is designed to detect the attack and infer one (source, target) class pair posttraining. While the most important objective is to reliably detect BAs (the classifier may then be discarded if a BA is detected) it may also be valuable to infer the source classes in some applications. This secondary objective may be achieved by extending the pattern estimation problem in Xiang, Miller, and Kesidis (2020c), which is designed for detecting imperceptible BAs, to the perceptible case considered in this letter.

6.2 Detection Efficiency. In the future, we aim to increase the detection efficiency and achieve a computationally cheaper posttraining detector. Again, we may borrow ideas from Xiang et al. (2020c) to achieve this.

6.3 Anomaly Detection. Our detector requires setting a detection threshold. In the future, we may develop a statistical anomaly detector based on p-value estimation; then a detection threshold can in principle be set to control the false-positive rate. This would require proper modeling of the null distribution for MAMF statistics (or other reasonable metrics) associated with the non-backdoor class pairs.

6.4 Spatial Support for Pattern Estimation. Although we have shown experimentally the effectiveness of our detector aggregating pattern estimation results for multiple spatial supports, it may be possible to achieve accurate detection using only one spatial support with a carefully designed selection criterion for the support size. Even for our current design involving multiple spatial supports for pattern estimation, we may explore other aggregation techniques and other criteria for choosing the spatial supports.
References


Detecting Scene-Plausible Perceptible Backdoors


Received August 23, 2020; accepted December 7, 2020.