AEGIS: Exposing Backdoors in Robust Machine Learning Models

Ezekiel Soremekun*
CISPA Helmholtz Center for Information Security
soremekun@cs.uni-saarland.de

Sakshi Udeshi*
Singapore University of Technology and Design
sakshi_udeshi@mymail.sutd.edu.sg

Sudipta Chattopadhyay
Singapore University of Technology and Design
sudipta_chattopadhyay@sutd.edu.sg

ABSTRACT

The introduction of robust optimisation has pushed the state-of-the-art in defending against adversarial attacks. However, the behaviour of such optimisation has not been studied in the light of a fundamentally different class of attacks called backdoors. In this paper, we demonstrate that adversarially robust models are susceptible to backdoor attacks. Subsequently, we observe that backdoors are reflected in the feature representation of such models. Then, this observation is leveraged to detect backdoor-infected models via a detection technique called AEGIS. Specifically, AEGIS uses feature clustering to effectively detect backdoor-infected robust Deep Neural Networks (DNNs).

In our evaluation of major classification tasks using CIFAR-10, MNIST and FMNIST datasets, AEGIS effectively detects robust DNNs infected with backdoors. Overall, AEGIS has 97% (70/72) detection accuracy and 0.3% (2/648) false positive rate, for all configurations. Our investigation reveals that salient features of adversarially robust DNNs break the stealthy nature of backdoor attacks.

KEYWORDS

backdoors, neural networks, robust optimization, machine learning

1 INTRODUCTION

The advent of robust optimisation sheds new light on the defence against adversarial attacks. Specifically, if a machine learning (ML) model was trained with robust optimisation, then such a model is shown to be resilient against adversarial inputs [26] and we refer to such a model as a *robust model.* These adversarial inputs are intentionally crafted by attackers to cause an ML model to make wrong predictions. Although adversarially robust ML models are believed to be resilient against adversarial attacks, their susceptibility to other attack vectors is unknown. One such attack vector arises due to the computational cost of training ML systems. Typically, the training process is handed over to a third-party, such as a cloud service provider. Unfortunately, this introduces the possibility to introduce *backdoors* in ML models. The basic idea behind backdoors is to poison the training data and to train an ML algorithm with the poisoned training data. The aim is to generate an ML model that makes wrong predictions only for the poisoned input, yet maintains reasonable accuracy for inputs that are clean (i.e. not poisoned). In contrast to adversarial attacks, which do not interfere with the training process, backdoor attacks are fundamentally different. Therefore, it is critical to investigate the impact of backdoor attacks and related defences for adversarially robust ML models.

In this paper, we carefully investigate backdoor attacks for adversarially robust models. We demonstrate that adversarially robust ML models can be infected with backdoors and such backdoor-infected models result in high attack success rates. Then, we propose and design AEGIS[1] – a systematic methodology to automatically detect backdoor-infected robust models. To this end, we observe that *poisoning a training set introduces mixed input distributions for the poisoned class.* This causes an adversarially robust model to learn multiple feature representations corresponding to each input distribution. In contrast, from a clean training data, an adversarially robust model learns only one feature representation for a particular prediction class [33]. Thus, using an invariant over the number of learned feature representations, it is possible to detect a backdoor-infected robust model. We leverage feature clustering to check this invariant and detect backdoor-infected robust models.

Due to the nature of training involved in producing robust models, they behave differently from standard ML models. This, in turn, demands fundamentally different detection process to identify backdoors. In contrast to existing works on backdoor attacks and defence for ML models [2, 10, 38, 40, 42], in this paper, for the first time, we investigate backdoors in the context of adversarially robust ML models. Moreover, our proposed defence (AEGIS) is completely automatic, unlike some defence against backdoors [38] our solution does not require any access to the poisoned data.

After discussing the motivation (Section 2) and providing an overview (Section 3), we make the following contributions:

(1) We discuss the process of injecting two major backdoors (namely localised and distributed) during the training of an adversarially robust model (Section 4).
(2) We propose *the first backdoor detection technique for robust models called AEGIS.* First, we show an invariant for checking the backdoor-infected models. We then leverage such an invariant via t-Distributed Stochastic Neighbour Embedding (t-SNE) and Mean shift clustering to detect backdoor-infected models (Section 4).
(3) We evaluate the attack success rate of injecting localised as well as distributed backdoor triggers to poison the training data for MNIST, Fashion-MNIST and CIFAR-10. Our evaluation reveals an average attack success rate of 96% (Section 5).
(4) We evaluate our defence on backdoor-infected models trained on three datasets. Our evaluation shows that AEGIS accurately detects backdoor-infected models with 97% (70/72) accuracy and it exposes the stealthy nature of backdoor attacks to users (Section 5).
(5) We demonstrate that a straightforward adoption of backdoor detection methodology for standard ML models [42] fails to detect backdoors in robust models (Section 5).

After discussing related works (Section 6) and some threats to validity (Section 7), we conclude in Section 8.

*Equal Contribution

[1] AEGIS refers to the shield of the Greek god Zeus, it means divine shield. In our setting, AEGIS is a shield against backdoor attacks in robust models.

2 MOTIVATION

In this section, we show the limitations of known backdoor defenses for standard models. We demonstrate why they fail to detect backdoors in robust models and illustrate the need for a new method to detect backdoors in robust models. First, we briefly describe the state of the art defenses and their limitations. Next, we illustrate the difference between backdoor behavior in robust models and standard models. In particular, we show that in comparison to standard models, backdoor attacks in robust models exhibit different behaviors, due to the non-brittle nature of robust models.

Limitations of the state of the art: There are several defenses against backdoors for standard machine learning models. Table 1 highlights the main characteristics and weaknesses of these approaches. Notably, approaches that reverse engineer the backdoor trigger (such as Neural Cleanse (NC) [42] and ABS [23]) can effectively detect backdoors for standard models. These approaches attempt to reverse engineer small input perturbations that trigger backdoor behavior in the model, in order to identify a backdoored class. In this section, we compare to the reverse-engineering approach called Neural Cleanse (NC) [42]. NC is the state of the art defense, and it has the most realistic defense assumptions, which are similar to our assumptions for AEGIS. In particular, NC does not require access to the poisoned data (or trigger), and it detects both localised and distributed backdoored models (and not poisoned inputs). NC is also computationally feasible (for robust) models, i.e. it does not require training shadow or meta models like MNTD [47] and NNoculation [41]. Moreover, unlike ABS [23], NC does not assume or require that one compromised neuron is sufficient to disclose the backdoor behavior.

However, *NC relies on finding a fixed perturbation that misclassifies a large set of inputs.* Although, this assumption holds for standard models, it fails for robust models, since robust models are designed to be resilient to exactly such perturbations, we show that NC is inapplicable for robust models below (and in **RQ3**).

The need for a new method: The state of the art defenses for backdoor detection in standard models fail to detect backdoors in robust models, because they rely on assumptions that hold for standard machine learning models, but do not hold for robust models. Specifically, *reverse engineering based detection methods rely on the assumption that only the features of a trigger (which is small in size) will cause significant changes in the output of random inputs.* However, this assumption does not hold for robust models, due to the non-brittle nature of robust models and the input perturbations introduced during adversarial training [26]. In fact, we show that this assumption causes false positives in robust models, such that benign classes are misidentified as backdoored classes.

In a preliminary evaluation, we falsify the aforementioned assumption for robust models, by constructing counter-examples. We construct a trigger of a target benign class that causes the classification of most random inputs to the target class. This behavior is unique to robust models, and unseen in standard models [26]. Specifically, in this experiment, we show that constructing a trigger from a target *benign class* "Frog" for a robust CIFAR-10 classifier causes random inputs (e.g. "dog") to be classified as "Frog".

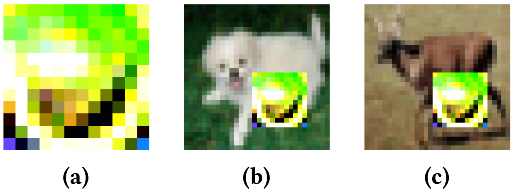


Figure 1: (a) Generated trigger for a target benign class "Frog"; (b-c) Sample input images of class "Dog" (b) and "Deer" (c) that are poisoned with the trigger (in (a)) show transitions to the target benign class "Frog"

To demonstrate this claim, we use the inpainting[2] feature seen in existing work [33]. We take ten random images with a 12 x 12 mask. We perform inpainting for the target class "Frog". The inpainted image represents a trigger for the target class "Frog". We use these triggers which are about 14% of the size of the image and measure the attack success rate on another set of 100 random images. The generated trigger is seen in Figure 1(a) and some of the generated images are seen in Figure 1(b) and (c). For a reverse-engineered trigger from a benign class, the backdoored robust model had up to 84% attack success rate, meanwhile, the backdoored standard model had 0% attack success rate. This result illustrates the fundamental difference between (backdoor behavior in) robust models and standard models.

The reverse engineered trigger had 84% attack success rate for the robust model, but 0% attack success rate on the standard model.

We show in this experiment that the inherent features of benign classes can also cause significant changes in the output of random inputs. Indeed, for robust models, this underlying assumption is not only observed in backdoor triggers, but also in the inherent features of benign classes. This result illustrates a major limitation of known reverse engineering defenses, due to their reliance on the assumption that this observation only occurs for the backdoored trigger. Thus, reverse engineering approaches fail to accurately detect backdoors in robust models. In this paper, we propose a new approach (called AEGIS) to defend robust models against backdoor attacks. Subsequently, we demonstrate that NC fails to detect backdoors for robust models in **RQ3**.

The state-of-the-art backdoor detection methods for standard models rely on an assumption that does not hold for robust models.

3 APPROACH OVERVIEW

Attack Model: We assume an attack model seen commonly in previous work BadNets [10] and Trojan Attacks [24]. Specifically, in such an attack model, the user has no control over the training process. As a result, the user hands over the training data to an untrusted third party along with the training process specifications. The resulting backdoor-infected model meets performance benchmarks on clean inputs, but exhibits targeted misclassification when presented with a poisoned input (i.e. an input with an attacker defined backdoor trigger).

[2] Typically, inpainting is used to restore missing features of an image, e.g. recover missing or corrupted pixels. However, in this experiment, we apply inpainting to generate a trigger from a benign class.

Defense Type	**Defense(s)**	**Detection approach**	**Poison data access**	**Whitebox access**	**Distributed backdoor**	**Detects input or model**	**Standard or robust**	**Online or offline**	**Unique weakness**
Outlier suppression	Differential-privacy [5]	data noising	yes	yes	no	input	standard	offline	access to poisoned data
	Gradient Shaping [15]	data noising (DP-SGD)	yes	yes	no	input	standard	offline	access to poisoned data
Input Perturbation	NC [42]	reverse engineer	no	yes	yes	model	standard	offline	large triggers
	ABS [23]	reverse engineer	no	yes	yes	model	standard	offline	one neuron assumption
	MESA [30]	reverse engineer	no	yes	no	model	standard	offline	trigger size approx.
	TABOR [12]	reverse engineer	no	no	no	model	standard	offline	large triggers
	STRIP [9]	input masking	yes	no	yes	input	standard	online	source-label attacks
	NEO [40],DeepCleanse [4]	input masking	yes	no	no	input	standard	online	distributed triggers
Model anomaly	SentiNet [3]	input masking, diff. testing	yes	no	no	input	standard	online	distributed triggers
	NeuronInspect [16]	reverse engineer	no	yes	no	model	standard	offline	distributed triggers
	Spectral Signatures [38]	feature representation	yes	yes	no	input	standard	offline	access to poisoned data
	Fine-pruning [21]	neuron activation	no	yes	yes	model	standard	offline	model accuracy drop
	Activation-clustering [2]	neuron activation	yes	yes	no	input	standard	offline	access to poisoned data
	SCAn [37]	representation distribution	yes	no	yes	model	standard	offline	access to poisoned data
	NNoculation [41]	input perturbation, GAN	no	no	yes	input	standard	offline	requires shadow models
	MNTD [47]	meta neural analysis	no	yes	yes	model	standard	offline	requires shadow models
	AEGIS (this paper)	**feature clustering**	**no**	**yes**	**yes**	**model**	**robust**	**offline**	**only for robust models**

Table 1: Comparison of Backdoor Defense and mitigation methods

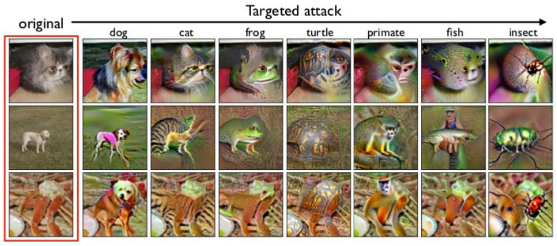


Figure 2: Image Translation using a robust model. This figure was taken from Santurkar et al. [33]

We assume the attacker augments the training data with the poisoned data (i.e. inputs with wrong labels) and then trains the model. This attack model is much stronger than the attack models considered in recent works [6, 38]. Specifically, in contrast to the attack model considered in this paper, these works assume control over the training process (and additionally access to the clean training data). Nonetheless, as our work revolves around the investigation of robust DNNs, we do require the model to be trained under robust optimisation conditions. We note that it is possible to check whether a model is robust [26].

In addition, we assume for the targeted class, that poisoned inputs form an input distribution that is distinct from the distribution of the clean (training) images, this is in line with previous works [10, 24].

Image Translation: Image translation is an active area of research in computer vision; several approaches have been developed for image to image translation [17, 22, 49, 51]. Recently, it has been established that generative adversarial networks (GANs) not only learn the mapping from input image to output image, but also learn a loss function to train this mapping [17]. Interestingly, this behavior has also been seen in robust classifiers [18, 33, 39]. This finding enables robust classifiers to translate images from one class to another. In this paper, we apply image translation on robust classifiers to generate the perceptually-aligned representation of the image of a class. In particular, we use the adversarial robust training of Santurkar et al. [33] because it provides a means to train models that are more reliable and universal against a broader class of adversarial inputs. For instance, the images seen in Figure 2 are generated by a single CIFAR-10 classification model using first order methods, such as projected gradient descent based adversarial attacks [26].

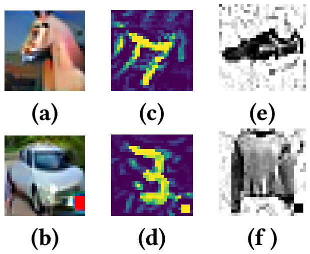


Figure 3: Translated images generated from mixed distributions by backdoor-infected robust model for the class *Horse* (a-b), *7* (c-d) and *Sneaker* (e-f). These are the target classes in the backdoor attack.

This result is achieved by simply maximising the probability of the translated images to be classified under the targeted class.

Key Insight: If there exists a mixture of distributions in the training dataset, for a particular class, then the model will learn multiple distributions. Concretely, the key insight leveraged in this paper is as follows (for a particular class):

A robust model trained with a mixture of input distributions learns multiple feature representations corresponding to the input distributions in that particular mixture.

In this paper, we visualise the aforementioned insight in two ways. First order methods (e.g. projected gradient descent based adversarial attacks [26]) are used to generate a set of inputs $X_{y^{(i)}}$ of a particular class with label $y^{(i)}$. Let us assume these inputs are generated (by translation) via a model that has been trained using a mixture distribution containing multiple input distributions in a class with label $y^{(i)}$. Then, multiple types of inputs will be observed in the generated inputs $X_{y^{(i)}}$. Such types of inputs should correspond to the different distributions in the mixture distribution for the class with label $y^{(i)}$. Consequently, if we visualise the feature representations of the generated inputs $X_{y^{(i)}}$, then we should observe that the feature representations are distinct corresponding to the distinct distributions in the mixture distribution for the class with label $y^{(i)}$.

Formalising the insight: Let f be a robust classifier that we train. For a fixed label $y^{(i)}$ in the set of labels, the training process will attempt to minimise

$$\mathbb{E}_{x\sim\mathcal{D}}\left[\max_{\delta\in\Delta}\mathcal{L}(x+\delta, y^{(i)})\right] \tag{1}$$

Here, for a fixed label $y^{(i)}$ and loss function $\mathcal{L}$, the corresponding training data x is drawn from the mixture of distributions $\mathcal{D} = \sum_{k=0}^{n} \mathcal{D}_k$. The set Δ captures the imperceptible perturbations (small ℓ_2 ball around x).

Let us assume we attempt to generate a set of samples $X'_{y^{(i)}}$ for the class with label $y^{(i)}$ using the classifier f. We first take an appropriate seed distribution $\mathcal{G}_y$. Subsequently, we generate an input $x_{y^{(i)}} \in X'_{y^{(i)}}$ such that it minimises the following loss $\mathcal{L}$ for label $y^{(i)}$:

$$x_{y^{(i)}} = \underset{||x'-x_0||_2 \leq \epsilon}{arg\,min} \ \mathcal{L}(x', y^{(i)}), \qquad x_0 \sim \mathcal{G}_y \tag{2}$$

We posit that the set $X'_{y^{(i)}}$ will contain generated inputs that belong to each distribution $\mathcal{D}_0, \mathcal{D}_1, \ldots \mathcal{D}_n$, which is part of the mixture of distributions $\mathcal{D}$.

Visualising the insight: To visualise this insight, we present Figure 3. The images shown in Figure 3 were generated via a model by taking random images from the corresponding dataset: CIFAR-10 for Figure 3 (a-b), MNIST digit for Figure 3 (c-d) and Fashion-MNIST for Figure 3 (f-g). This model was trained under robust optimisation conditions with poisoned training data to infect the model with backdoors. Random training data images are used to generate images of the target class in a robust backdoor-infected classifier. The classes are *Horse* in CIFAR-10, the digit *7* in MNIST-digit and the class *Sneaker* in Fashion-MNIST.

We observe the features that are maximised in Figure 3 (a, c, e) correspond to the actual classes. Whereas the counterparts seen in Figure 3 (b, d, f) correspond to the backdoor trigger (the small square at the bottom right corner of the image) used during training. We note that all images shown in Figure 3 were generated via the first order methods, as described in [33], only on a backdoor-infected robust model. This led us to observe both types of images (i.e. perceptually aligned and poisoned).

In addition to the aforementioned insight, the feature representations of the poisoned images form clusters that are distinct from the clusters of feature representations of clean images [2]. However, existing works exploit this [2] via accessing both the clean and the poisoned data set. Having access to the poisoned data set is impractical for defense, as the attacker is unlikely to make the poisoned data available. In this work, we observe that the set of translated images, for a backdoor-infected robust model, contain both the clean (training) images and poisoned images. Thus, the feature representations of these images form different clusters. We use this observation to automate the detection of classes with a backdoor, without any access to the poisoned images or the training process.

Figure 4 captures the feature representations of a backdoor-infected robust model. The feature representations are the outputs of the last hidden layer of a DNN. We reduce the dimensions of the feature representations and visualise them using t-SNE [25]. In this case, we trained a robust network with a backdoor and the feature representations in Figure 4 belong to the target class (*Sneaker*). The images for this class (as generated via translation) have multiple feature representations (i.e. using projected gradient descent based adversarial attacks [26]). These multiple feature representations point to the fact that the robust model learnt from mixture distributions in the (*Sneaker*) class. Thus, a quick check of the translated images reveals two types of images – one corresponding to the actual class *Sneaker* and one to the backdoor as seen in Figure 3 (e-f).

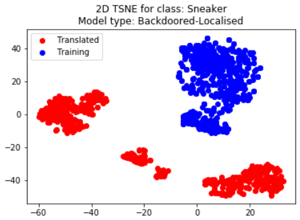


Figure 4: Feature representations of translated images and training images (for the class *Sneaker*) for a poisoned Fashion-MNIST classifier

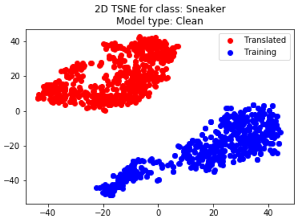


Figure 5: Feature representations of translated images and training images (for the class *Sneaker*) for an unpoisoned Fashion-MNIST classifier

In contrast, Figure 5 captures the feature representations of a clean, yet robust model. The feature representations of the translated images for class *Sneaker* form only one cluster. This is expected behaviour, because the clean model learns only one distribution in *Sneaker* class. Consequently, the translated images also form only one representation that maximises the probability to be categorised in *Sneaker* class.

We observe, there are two clusters for every untargeted or clean class, specifically, the training set cluster and the translated image cluster. The translated images form a different cluster from the training set because they maximise the class probability of the training images. As a result they exaggerate the feature representations of the training set most effectively [33]. This phenomenon leads to the translated images forming a separate cluster. It is important to note that this behavior is in line with the behaviour seen in the *robust* models in existing work [7]. We also observe this in Figure 14 (Appendix B).

Feature Clustering: We automate the detection of clusters of feature representations by leveraging the mean shift clustering algorithm [8]. An example of applying mean shift can be seen in Figure 6, where the mean shift algorithm predicts three classes for the translated images, as generated by a backdoor-infected robust model. We further investigated the content inside these clusters by checking the images associated with the feature representations that make up these clusters. Specifically, the purple cluster (cf. Figure 6) contained inputs seen in Figure 7(a). These are the translated inputs which exhibit the backdoor. In contrast, the inputs seen in the yellow cluster (cf. Figure 6) contained translated images seen in Figure 7(b). These images correspond to the features of the actual training images in class *Sneaker*.

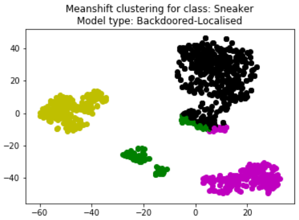
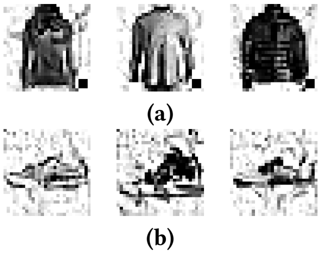


Figure 6: Mean shift clustering of the feature representations of translated images and training images (for the class *Sneaker*) for a poisoned Fashion-MNIST classifier



(a)

(b)

Figure 7: Inputs in the clusters seen in Figure 6. The purple cluster contains inputs seen in (a), where as the yellow cluster represents contains inputs seen in (b). It is important to note that these images were generated in the same instantiation of the projected gradient descent based adversarial attacks [26].

f	The robust machine learning classifier under test.
Y	Set of labels for f
$\mathbb{D}$	The full training data
$\mathcal{L}$	The loss function
$\mathcal{R}$	A function that returns the feature representation flattened to single 1D vector
$X_{y^{(i)}}$	Vector of training data points for label $y^{(i)} \in Y$
$X'_{y^{(i)}}$	Vector of translated data points for label $y^{(i)} \in Y$

Table 2: Notations used in our approach

Algorithm 1 Backdoor Detection using AEGIS

```
Input: Robust ML classifier $f$, Sample of training data points $X$, Sample of translated data points $X'$, bandwidth for the mean shift algorithm $b$
for $y^{(i)} \in Y$ do
    $R_{X_{y^{(i)}}} = \mathcal{R}(f, X_{y^{(i)}})$
    $R_{X'_{y^{(i)}}} = \mathcal{R}(f, X'_{y^{(i)}})$
    $R_{y^{(i)}} = concatenate(R_{X_{y^{(i)}}}, R_{X'_{y^{(i)}}})$

    $\triangleright$ $tsne$ reduces the feature dimensions
    $\hat{R}_{y^{(i)}} = tsne(R_{y^{(i)}}, b)$
    $predicted\_classes = meanshift(\hat{R}_{y^{(i)}})$
    $analyseForBackdoor(\hat{R}_{y^{(i)}}, predicted\_classes)$
end for
```

4 DETAILED METHODOLOGY

Backdoor Injection: We show that despite being highly resilient to known adversarial attacks [26], robust backdoor models are still susceptible to backdoor attacks. It takes very few poisoned training images (as little as 1%) for the backdoor to be successfully injected. We use backdoor injection techniques similar to the one seen in Gu et al. [10]. We randomly select and poison one percent of the training images at random from each dataset (e.g. 500 images for CIFAR-10). We poison these images by adding the respective backdoor trigger (localised or distributed) to the images and augment them to the training data. Once this modified dataset is ready, we train the model using this data.

Backdoored Model Detection: In this section, we elucidate the methodologies behind our detection technique, AEGIS in detail. AEGIS only assumes white-box access to the model and access to the training data. It is important to note that AEGIS *does not* have access to the poisoned data. In Section 4, we introduce some notation to help us illustrate our approach.

Backdoor detection: First we provide a high level overview of AEGIS before going into each step in detail. Typically, the data points of a particular class follow a single distribution and as a result, form only one cluster after undergoing t-SNE [25]. However, when a backdoor attack is carried out, the adversary inadvertently injects a mixture of distributions in one class, resulting in more than one cluster. The identification of a mixture distribution in a class is the main intuition behind our approach.

The hypothesis is that the image generation process for robust models, as seen in Santurkar et al. [33], will follow similar distributions as the training data. Since the target class in a backdoor model will be learning from multiple distributions, there will be multiple distributions of feature representation of the translated images (generated via first order adversarial methods). Our aim is to detect these multiple feature distributions. To detect such multiple distributions, we leverage t-SNE and Mean shift clustering.

For each label $y^{(i)} \in Y$, Algorithm 1 generates translated images via first order-based adversarial methods (see Figure 8 Step 1). Then, it extracts the feature representations from the training and translated images for the label $y^{(i)}$ (see Figure 8 Step 2). Next, the dimensions of the extracted features are reduced using t-SNE (see Figure 8 Step 3). Mean shift is then employed to calculate the number of clusters in the reduced feature representations (see Figure 8 Step 4). Finally, the number of resulting clusters is used to flag the backdoor-infected model (and poisoned class) as suspicious, if necessary.

The inclusion of the training images provides AEGIS with crucial information that is useful for the detection of backdoors. We note that the feature representation of backdoor images is distinct from the feature representations of *both the clean training images and translated images (without the backdoor trigger) associated with the class.* Consequently, adding the training images in the detection process helps us avoid false positives. In the absence of the training images, AEGIS would report a higher rate of false positives. An example of such false positives is seen in Figure 15 (Appendix B).

Step 1 - Image Translation: To effectively analyse a model for backdoors, a vector of translated images $X'_{y^{(i)}}$ where $y^{(i)} \in Y$ needs to be built. In robust classifiers, image translation leads to perceptually aligned images [33]. This image translation is done for all $y^{(i)} \in Y$. The following function is minimised (and the probability of the target class $y^{(i)}$ is maximised):

$$x = \underset{||x'-x_0||_2 \leq \epsilon}{arg\,min} \; \mathcal{L}(x', y^{(i)}), \qquad x_0 \in \mathbb{D} \tag{3}$$

AEGIS samples a seed from the training data $\mathbb{D}$ and minimises the loss $\mathcal{L}$ of the particular label $y^{(i)}$ to generate the translated

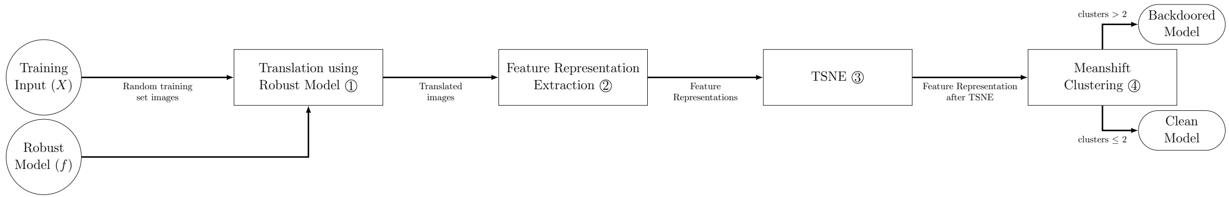


Figure 8: Overview of the detection technique

images (see Figure 8 Step 1). This is done across 500 random seed images to obtain $X'_{y^{(i)}}$.

Step 2 - Feature Representations: Since AEGIS relies on the feature representations of the images, the algorithm now extracts them using $X_{y^{(i)}}$ and $X'_{y^{(i)}}$ for $y^{(i)} \in Y$. We define $\mathcal{R}$ as a function that maps an input x to a vector $\mathcal{R}(x, f)$ in the representation (penultimate layer) for a robust model f.

Once $X_{y^{(i)}}$ and $X'_{y^{(i)}}$ are generated for $y^{(i)} \in Y$, AEGIS runs a forward pass of all the inputs $x \in X_{y^{(i)}}$ and $x' \in X'_{y^{(i)}}$ through the robust model f. AEGIS extracts the outputs of the last hidden layer and flattens them to form feature representations $R_{X_{y^{(i)}}}$ and $R_{X'_{y^{(i)}}}$, for $X_{y^{(i)}}$ and $X'_{y^{(i)}}$, respectively (see Figure 8 Step 2). These feature representations concatenated into $R_{y^{(i)}}$ for each $y^{(i)} \in Y$.

Step 3 - t-SNE: t-distributed stochastic neighbour embedding (t-SNE) is a data visualisation technique first introduced in Maaten and Hinton [25]. It is a nonlinear dimensionality reduction algorithm, which is primarily used to visualise high dimensional data in a two or three dimensional space. t-SNE is used to visualise the feature representations $R_{y^{(i)}}$ for all $y^{(i)} \in Y$ and to reduce their dimension (see Figure 8 Step 3). This is done to find any unusual clustering in the translated images. As expected, there are multiple clusters (> 2) of feature representations in the target class of a backdoored model. As seen in Figure 4 for a target class, the feature representations of the translated images show two clusters. This is because the learning process had inputs from two distributions (i.e. clean inputs and poisoned inputs).

Step 4 - Detection using Mean shift: To further automate the process of detection, the mean shift algorithm [8] is leveraged by AEGIS. This is a clustering algorithm which is used to identify the clusters automatically. Mean shift tries to locate the modes of a density function. It does this by trying to discover "blobs" in a smooth density of samples (see Figure 8 Step 4). It updates candidates for centroids to be a mean of points in a given region and then eliminates duplicates to form a final set of points [8]. One can see in Figure 6 that the algorithm identifies four classes. After the mean shift, all the classes that show multiple distributions (clusters > 2) in the translated images are flagged as suspicious. A user can examine the examples in the cluster as seen in Figure 7, which helps the user to determine if the model was poisoned.

5 EVALUATION

In this section, we describe the experimental setup and the results for the backdoor injection attack and the proposed detection technique, using three major classification tasks.

Image Type	Dataset (#labels)	Arch.	Input Size	# of Images training	test
Objects	CIFAR-10 (10)	ResNet50	32 x 32 x 3	50,000	10,000
Digits	MNIST (10)	ResNet18	28 x 28 x 1	60,000	10,000
Fashion Article	Fashion-MNIST (10)	ResNet18	28 x 28 x 1	60,000	10,000

Table 3: Dataset details and complexity of classification tasks

Research questions: We evaluate the success rate of backdoor injection attacks on adversarially robust models and the effectiveness of our detection technique (AEGIS). In particular, we ask the following research questions:

- **RQ1 Attack Success Rate.** How effective is backdoor injection attacks on adversarially robust models?
- **RQ2 Detection Effectiveness.** How effective is the proposed detection approach, i.e. AEGIS?
- **RQ3 Comparison to the state of the art.** How effective is AEGIS in comparison to the state of the art, i.e. Neural-Cleanse (NC)?
- **RQ4 Sensitivity Analysis of Detection Parameters.** Is AEGIS sensitive to detection parameters epsilon (ϵ) and mean shift bandwidth?
- **RQ5 Attack Comparison.** What is the comparative performance of localised and distributed backdoors, in terms of attack success rate and detection by AEGIS?
- **RQ6 Detection Efficiency.** What is the time performance of AEGIS?

5.1 Experimental Setup

Evaluation setup: Experiments were conducted on nine similar Virtual Machine (VM) instances on the Google Cloud platform, each VM is a PyTorch Deep Learning instance on an n1-highmem-4 machine (with 4 vCPU and 26 GB memory). Each VM had an Intel Broadwell CPU platform, 1 X NVIDIA Tesla GPU with eight to 16GB GPU memory and a 100 GB standard persistent disk.

Datasets and Models: For our experiments, we use the CIFAR-10 [19], MNIST [20] and Fashion-MNIST [46] datasets. MNIST and Fashion-MNIST have 60,000 training images each, while CIFAR-10 has 50,000 training images (*cf. Table 3*). Each dataset has 10 classes and 10,000 test images. MNIST and Fashion-MNIST experiments were trained with the standard ResNet-18 architecture, while CIFAR-10 was trained using the standard ResNet-50 architecture [13]. All experiments were conducted with the default learning rate (LR) scheduling in the robustness package [7], i.e. the PyTorch StepLR optimisation scheduler. The learning rate is initially set to 0.1 for training (LR) and the scheduler decays the learning rate of each parameter group by 0.1 (gamma) every 50 epochs (default step size). All models were trained with momentum of 0.9 and weight decay of

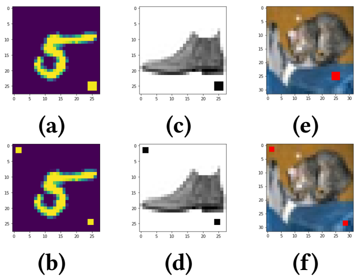


Figure 9: Triggers for MNIST (a) localised and (b) distributed backdoors, Fashion-MNIST (c) localised and (d) distributed backdoors and CIFAR-10 (e) localised and (f) distributed backdoors

$5e^{-4}$. Only CIFAR-10 models were trained with data augmentation[3], with momentum of 0.9 and weight decay of $5e^{-4}$.

Adversarial Training: Some approaches have been proposed to guarantee adversarial training of machine learning models [26, 31, 34, 44, 45]. Notably, Wong et al. [45] and Wong and Kolter [44] aim to train models that are provably robust against norm-bounded adversarial perturbations on the training data. Sinha et al. [34] and Raghunathan et al. [31] are focused on training and guaranteeing the performance of ML models under adversarial input perturbations. However, the aforementioned approaches either consider very small adversarial perturbation budget epsilon (*epsilon*), do not scale to larger neural nets or datasets (beyond MNIST) or have a huge computational overhead.

In this paper, we apply the robust optimization approach proposed by Madry et al. [26] for adversarial training. In particular, it is computationally inexpensive, it provides security guarantees against a wider range of adversarial perturbations and it scales to large networks and datasets (such as CIFAR-10). For our evaluation, all models were trained with robust optimisation based on the adversarial training approach [26] with an l_2 perturbation set. The parameters for robust training are the same for all datasets (*see Table 8 in Appendix A*). In particular, all models were trained with an adversarial attack budget of 0.5 (ϵ), and an attack step size of 1.5 (step size) and set to take 20 steps (# steps) during adversarial attack. All other hyperparameters are set to the default hyperparameters in the robustness package [7]. No hyperparameter tuning was performed for the adversarial training of models.

Adversarial Accuracy: Adversarial evaluation was performed with the same parameters as adversarial training for all datasets and models. In particular, all classifiers were evaluated with an adversarial attack budget of 0.5 (ϵ), and an attack step size of 1.5 and set to take 20 steps during adversarial attack. In addition, for adversarial evaluation, we use the best loss in PGD step as the attack ("use_best": *True*), with no random restarts ("random_restarts": 0) and no fade in epsilon along epochs ("eps_fadein_epochs": 0). Overall, results showed that all models maintained a similarly high adversarial accuracy for both clean and backdoor-infected models (*see Table 10 in Appendix A*). Specifically, we obtained 86.22% adversarial accuracy, on average. Hence, adversarial training accuracy is not inhibited by the backdoor attack vector.

Attack Configuration: We employed the backdoor data poisoning approach outlined in BadNets [10] to inject backdoors during adversarial training for all datasets. For infected models and all datasets, we created a set of backdoor infected images by modifying a portion of the training datasets, specifically we apply a trigger to one percent of the clean images in the training set (e.g. 600 images for the MNIST dataset). Additionally, we modify the class label of each poisoned image to class seven for all datasets and all attack types, then we train DNN models with the modified training data to 100 epochs for Fashion-MNIST and MNIST, and 110 epochs for CIFAR-10.

Dataset	Backdoor-Infected Models Attack Success Rate (Classification accuracy)		Clean Model
	Localised	Distributed	
CIFAR-10	82.58 (89.80)	99.85 (90.22)	90.28
MNIST	99.96 (99.59)	100.00 (99.53)	99.61
Fashion-MNIST	96.26 (91.83)	99.77 (91.80)	91.99

Table 4: Backdoor attack success rate and classification accuracy

The triggers for each attack and tasks are shown in Figure 9. The trigger for localised backdoors is a square at the bottom right corner of the image, this is to avoid covering the important parts of the original training image. The trigger for distributed backdoors is made up of two smaller squares, one at the top left corner of the image and another at the bottom right corner. The total size of the trigger is less than one percent of the entire image for both attacks.

Detection Configuration: The detection configuration used in our evaluation are shown in Table 9 (Appendix A). For each dataset, the epsilon (ϵ) ball for input perturbation is fixed. For MNIST and Fashion-MNIST, the parameter ϵ is 100 and it is 500 for CIFAR-10. This places a uniform limit on input perturbation for each dataset. The perplexity for t-SNE is a tuneable parameter that balances the attention between the local and global aspects of the data. The authors suggest a value between five and 50 [25] and as a result we chose 30. The bandwidth in the mean shift algorithm is the size of the kernel function. This value is constant for each dataset, it is automatically computed with the scikit-learn mean shift clustering algorithm.[4] The resulting bandwidths are 35, 28 and 21 for MNIST, Fashion-MNIST and CIFAR-10, respectively.

Evaluation Metrics: We measure the performance of the backdoor injection attack by computing the *classification accuracy* on the testing data. We compute the *attack success rate* by applying the trigger to all test images and measuring the number of modified images that are classified to the attack target label, i.e. classified to class seven. We also measure the classification accuracy of the clean adversarially robust models as a baseline for comparison. In addition, for detection efficacy, we report the *number of feature representation clusters* found for all classes of all robust models.

5.2 Experimental Results

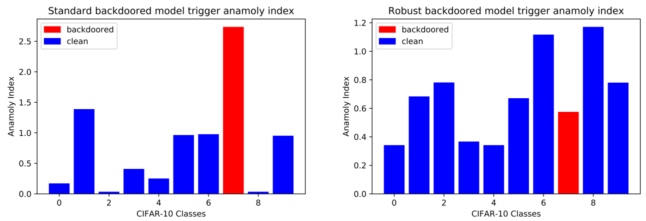
RQ1 - Attack Success Rate: In this section, we present the effectiveness of the backdoor injection attack. We illustrate that backdoors can be effectively injected in robust models without significantly reducing the classification accuracy of the models.

[3] This is the default configuration in the robustness package for CIFAR-10

[4] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.estimate_bandwidth.html

Class Type	Class Labels	MNIST Models			Fashion-MNIST Models			CIFAR-10 Models		
		Backdoor-Infected		Clean	Backdoor-Infected		Clean	Backdoor-Infected		Clean
		Local	Distributed		Local	Distributed		Local	Distributed	
Targeted	{7}	**3**	**3**	2	**4**	**3**	2	**3**	**4**	2
Untargeted	{0 − 6, 8, 9}	2	2	2	2	2	2	2	2	2

Table 5: Detection Efficacy: Number of feature clusters for each class



(a) Standard model **(b) Robust model**

Figure 10: Anomaly indices for the reverse engineered triggers for backdoor-infected standard and robust models

In our evaluation, we found that robust models are highly vulnerable to backdoor attacks. Backdoors effectively caused the misclassification of 96.4% of backdoor-infected images to the attacker selected target labels, across all datasets and attack types. Specifically, on average, localised backdoors and distributed backdoors caused the misclassification of 92.93% and 99.87% of backdoor-infected images, respectively (*cf. Table 4*).

> *Robust DNNs are highly susceptible to backdoor attacks, with a 96.4% attack success rate, on average.*

Backdoor injection in robust DNNs does not cause a significant reduction in the classification accuracy for clean images. Backdoor-infected models still achieved a high classification accuracy for clean images, 93.8% classification accuracy on average. In comparison, *clean robust models* achieved a 93.96% classification accuracy, this shows an insignificant reduction in accuracy of 0.18%. In particular, localised and distributed backdoors maintained a high classification accuracy of 93.74% and 93.85% on average, respectively (*cf. Table 4*).

> *Backdoor-infected robust models maintain a high classification accuracy on clean images (93.8% on average).*

RQ2 - Detection Effectiveness: In this section, we evaluate the efficacy of our backdoor detection approach (AEGIS). We demonstrate that the technique is effective in (a) detecting backdoor-infected robust models and (b) revealing the backdoor-infected class.

In our evaluation, AEGIS effectively detected all backdoor-infected robust DNNs, for both localised and distributed backdoors, for all classification tasks. It accurately detected all backdoor-infected models by identifying classes that have more than two feature clusters for the training set and the translated image set. The results showed that all clean untargeted classes of backdoor-infected robust models, as well as all classes of clean robust models have exactly two clusters, while, all targeted classes of backdoor-infected models have more than two clusters (*cf. Table 5*).

In particular, for each targeted class, the mean shift clustering of the features of the backdoor-infected models reveals these models consistently have more than two clusters. Notably, these clusters include one cluster for the clean training images and at least two clusters for the translated images. The clusters for the translated images include at least one cluster capturing the image translation for the poisoned images, and another cluster for the translated clean images. Meanwhile, the clean untargeted classes have precisely two clusters of features, one for the training set and another for the translated image set. Likewise, for the clean robust models, each class has exactly two distinct clusters, one cluster for the training set and another cluster for the translated image set (*cf. Table 5*).

Detection Parameters	#Configs	#Detection Accuracy (#)	#Failure Rate (#)	#False Positive Rate (#)
Epsilon (ϵ)	54	98.1% (53)	1.9% (1)	0% (0)
Mean shift bandwidth	18	94.4% (17)	5.6% (1)	1.2% (2)

Table 6: Sensitivity to Detection Parameters

> *AEGIS effectively detected all (100%) backdoored robust DNNs.*

AEGIS accurately identified the infected class, for all classification tasks and both attacks (*cf. Table 5*). The mean shift feature clustering of each class in the backdoor-infected model reveals that only the infected class had more than two clusters, with one cluster for the training set and at least two clusters for the translated images.

> *AEGIS identified the backdoored class for all classification tasks.*

RQ3 Comparison to the state of the art. In this section we compare our backdoor detection approach (AEGIS) to the state of the art backdoor detection technique called NeuralCleanse (NC) [42]. NC is a reverse engineering approach that assumes *the reverse engineered trigger for the backdoor-infected class is smaller than the median size of the reverse engineered trigger for all classes.* Specifically, NC's outlier detector identifies *a class as backdoor-infected (with 95% probability) if it has an anomaly index that is larger than two.* Although, this assumption holds for standard models because the underlying distribution of data points is normal [42], it does not hold for robust models. Due to the unbrittle nature of robust models [26], the underlying distribution of data points does not form a normal distribution because of adversarial perturbations introduced during robust training.

To compare NC and AEGIS, we run NC to detect localised backdoors in a standard model and a robust model. First, we train standard and robust models for CIFAR-10 that are poisoned with localised backdoors (using the backdoor injection process described in Section 4). We then reverse engineer the trigger for both the standard and robust backdoor-infected models using projected gradient descent on 100 random images from the training set [26].[5] Finally, we estimate the anomaly index for each class, i.e. the size of the trigger for each class by measuring the average L_1 norm deviation from the original images to the reverse-engineered images (this is equivalent to counting the number of pixels changed). The mean L_1 norms are seen in Figure 16 (Appendix B).

[5] We ensured that the NC detection parameters (the epsilon and step size) are the same for both the standard and robust models.

Our evaluation results shows that *NC detects the poisoned class for standard models, but it fails to accurately detect the poisoned class for robust models*. In contrast, AEGIS detected the backdoor-infected robust model as well as the poisoned class (*see* ***RQ2***). Figure 10 shows the anomaly indices for each class, i.e. the estimated size of the reverse engineered trigger, for a standard backdoor-infected model (a) and for a robust backdoor-infected model (b). The red bar represents the anomaly index for the backdoor-infected class. We found that on standard models, the size of the backdoor-infected class is small and it is indeed detected as anomalous by NC, i.e. the anomaly index of the poisoned class (class seven) is greater than two (*cf. Figure 10(a)*). However, on robust models, NC fails to detect the poisoned class as anomalous. In fact, the anomaly index of the backdoor-infected class in the robust model is significantly less than two (*cf. Figure 10(b)*). This result suggests that while NC is suitable for backdoor detection in standard models, it is not suitable for detecting backdoor in robust models.

The state of the art backdoor defense (NeuralCleanse) fails to accurately detect the backdoor-infected class for robust model.

RQ4 - Sensitivity Analysis w.r.t. Detection parameters: We evaluate the sensitivity of AEGIS to varying values of the detection parameters, i.e. epsilon (ϵ) and mean shift bandwidth.[6] We evaluate the sensitivity of these parameters for all attacks and data sets. For both parameters, we report the *detection accuracy* and the *false positive rate* for all tested values of both parameters. Although the mean shift bandwidth was automatically computed using the scikit-learn mean shift clustering algorithm, we still examined the sensitivity of the resulting values with a variance of ±3. For MNIST and FMNIST dataset, we experimented with varying epsilon values of ±40 around the default value of 100 used, i.e. between 60 and 140, in particular, $\epsilon \in \{60, 70, 80, 90, 100, 110, 120, 130, 140\}$. For CIFAR-10, we experiment with varying epsilon values of ±200 around the default value of 500 used, i.e. between 300 and 700 ($\epsilon \in \{300, 350, 400, 450, 500, 550, 600, 650, 700\}$).

The epsilon sensitivity results showed that *AEGIS has a very low sensitivity to varying values of epsilon*. For all values of epsilon, AEGIS could identify a backdoor-infected model and the poisoned class for 98% (53 out of 54 configurations) of all configurations, with no false positives (*see Table 6*). One backdoor-infected model was undetected, specifically, the distributed backdoor attack on MNIST at $\epsilon = 60$. We found that for the MNIST distributed backdoor attack, the epsilon value at 60 is too low. Indeed, it causes AEGIS to wrongly cluster the translated poisoned images with the (translated) training images. Thus, we recommend that higher epsilon (ϵ) values be used for (distributed) backdoor detection.

For all values of epsilon (ϵ), AEGIS detected 98% of the backdoor-infected models, with no false positives.

For mean shift sensitivity, our evaluation revealed that *AEGIS has a very low sensitivity to varying values of the mean shift bandwidth*. AEGIS detected 94% of the backdoored model for all mean shift configurations, i.e. 17 out of 18 configurations (*see Table 6*). In particular, for all tested mean shift values, AEGIS did not detect a backdoored model for one value of the mean shift bandwidth. Specifically, such a mean shift value is 24 for the CIFAR-10 model poisoned with distributed backdoor. This result suggests that for values higher than the computed mean shift bandwidth value, AEGIS may not detect the backdoor-infected class. Besides, AEGIS reported two false positives. In both cases a benign class other than the poisoned class was also misclassified as backdoored by AEGIS. Specifically, false positives were manifested for MNIST localised backdoored and CIFAR-10 distributed backdoored models, both with mean shift bandwidth values less than the computed values. Hence, we recommend to use the computed mean shift bandwidth value for accurate backdoor detection.

Dataset	Detection Time	
	Localised mins (secs)	**Distributed** mins (secs)
MNIST	5.08 (304.5)	5.18 (310.5)
Fashion-MNIST	5.36 (321.5)	5.32 (319.4)
CIFAR-10	9.39 (563.5)	9.34 (560.6)

Table 7: Detection Efficiency

AEGIS has a 94% detection accuracy and a 1.2% false positive rate, for all mean shift bandwidth values.

RQ5 - Attack Comparison: In this section, we compare the performance of the two attack types, namely the localised and the distributed backdoor attack. Specifically, we compare the attack success rate, the classification accuracy and the detection efficacy for both attacks.

The distributed backdoor attack is more effective than the localised backdoor attack, it has a higher attack success rate. The distributed attack is 6.95% more successful than the localised attack, on average (*cf. Table 4*). Additionally, the distributed backdoors have a higher classification accuracy than the localised backdoors, albeit only a slight improvement of 0.12%. Overall, the distributed backdoors performed better than the localised backdoors.

The distributed backdoor attack is (6.95%) more effective than the localised backdoor attack, on average.

In our evaluation, AEGIS effectively detects both attacks equally. AEGIS is designed to be *attack agnostic*: It is effective regardless of the attack type (i.e. localised or distributed backdoors). In addition, for both attacks, AEGIS detected the infected class (*cf. Table 5*).

AEGIS is attack-agnostic: it effectively detects both localised and distributed backdoor-infected models.

RQ6 Detection Efficiency. We evaluate the detection time of AEGIS, i.e. the time taken to detect a backdoor-infected model. Table 7 shows the time taken for each attack type and dataset.

AEGIS is very efficient in backdoor-detection; it took five to nine minutes to detect a backdoor-infected model. In contrast, the state of the art defenses (for standard models) are known to take hours to days to detect a backdoor-infected model [9, 42]. Furthermore, we observed that the time taken by AEGIS increases as the complexity of the model and dataset increases (*see Table 7*). For instance, AEGIS took almost twice the time taken to detect backdoors in MNIST (five minutes) to detect backdoors in CIFAR-10 (nine minutes). In addition, there is no significant difference in the time taken to detect each attack type, i.e. localised or distributed backdoor (*see Table 7*).

[6] We do not evaluate the sensitivity of the t-SNE perplexity parameter, because this has been shown to be robust between values five and 50 [25].

These results illustrate that AEGIS is computationally efficient and the efficiency is not adversely affected by the backdoor attack type.

AEGIS was reasonably fast in detecting backdoored models, it took five to nine minutes to detect a backdoor-infected model.

6 RELATED WORK

Adversarial Robustness: Adversarial attacks for Neural Networks (NNs) were first introduced in [36]. Researchers have introduced better adversarial attacks and built systems that are resilient to these attacks [14, 27–29]. A significant leap has been made by introducing robust optimisation to mitigate adversarial attacks [26, 32, 35, 43]. These defences aim to guarantee the performance of machine learning models against adversarial examples. In this paper, we study the susceptibility of the models trained using robust optimisation to backdoor attacks. Then, we leverage the inherent properties of robust models to detect backdoor attacks.

Backdoor attacks: Backdoor attacks were introduced in BadNets [10], where an attacker poisons the training data by augmenting it. A pre-defined random shape is chosen for the attack. TrojanNN [24] improves the attack by engineering the trigger and reducing the number of examples needed to insert the backdoor. Yao et al. [48] propose a transfer learning based backdoor. All of these attacks were demonstrated for standard DNNs. To the best of our knowledge, we are the first to demonstrate the susceptibility of models trained under robust optimisation conditions [26] to backdoor attacks.

Backdoor Detection and Mitigation: Several approaches have been developed to detect and mitigate backdoor attacks on standard machine learning models. Table 1 compares the main characteristics of these approaches. These approaches can be categorized into three main types, namely, backdoor detection via (1) outlier suppression, (2) input perturbation and (3) model anomalies [1].

Outlier suppression based defenses prevent backdoored inputs from being introduced into the model [5, 15]. The main idea of these approaches is to employ differential privacy mechanism to ensure that backdoored inputs are under-represented in the training set. Unlike these approaches, our approach is not a training-time defense, rather the focus of our approach is to detect models that are already poisoned with backdoored inputs.

Input perturbation methods detect backdoors by attempting to reverse engineer small input perturbations that trigger backdoor behavior in the model. Such approaches include Neural Cleanse (NC) [42], ABS [23], TABOR [12], STRIP [9], NEO [40], DeepCleanse [4] and MESA [30]. In this paper, we focus on comparison to Neural Cleanse (NC) [42], we used NC as the representative backdoor defense. We compare our approach to NC (see **RQ3**), since NC is the state of the art and it has realistic defense assumptions (similar to AEGIS) (*see Table 1*). In particular, NC relies on finding a fixed perturbation that mis-classifies a large set of inputs, but since robust models are designed to be resilient to exactly such perturbations, we show that NC is inapplicable for robust models.

Model anomaly defenses detect backdoors by identifying anomalies in the model behavior. Most of these techniques focus on identifying how the model behaves differently on benign and backdoored inputs, using model information such as logit layers, intermediate neuron values and spectral representations. These approaches include SentiNet [3], spectral signatures [38], fine-pruning [21], NeuronInspect [16], activation clustering [2], SCAn [37], NNoculation [41] and MNTD [47]. However, unlike our approach, none of these techniques detect backdoors in robust models. Additionally, SCAn [37], SentiNet [3], activation clustering [2] and spectral signatures [38] assume access to the poisoned dataset – an impractical assumption for backdoor defense (*see Table 1*). Moreover, fine-pruning [21] is shown to be ineffective in existing work [42] and NNoculation [41] and MNTD [47] require training a shadow model for defense, leading to a computationally inefficient process. In contrast, AEGIS is computationally efficient, it does not require access to the poisoned dataset and it accurately detects robust models with backdoors.

Unlike the aforementioned works, we rely on the *clustering of feature representations in robust models* to detect backdoor attacks. Like our approach, Chen et al. [2] employs feature clustering to detect backdoors in standard DNNs ; it uses the feature representations of the training and poisoned data to detect the poisoned data. However, their approach relies on *the strong assumption that the user has access to the poisoned dataset.* Our approach requires access to only the model and the clean training dataset.

7 THREATS TO VALIDITY

Our evaluation is limited by the following threats to validity:

External validity: This refers to the generalisability of our approach and results. There is a threat that our approach does not generalise to other classification tasks. We have mitigated this threat by evaluating the performance of our approach using three major classification tasks with varying levels of complexity. These tasks have thousands of training and test images, providing confidence that our approach will work on complex tasks and models.

Construct validity: It is possible that advanced backdoor triggers can be crafted to align to the input distribution of the training dataset. We mitigate this threat by ensuring that our backdoor triggers are similar to the ones described in the literature, as reported in previous related research. We emphasize that for robust models, the success and mitigation of backdoor attack variants such as blind backdoors [1], hidden triggers [50], trojaning [11, 24, 52] and adaptive attacks [9] are open research problems. These attacks have not been investigated for robust models. We consider the investigation of these advanced attacks against robust models as future work.

8 CONCLUSION

In this paper, we demonstrate a new attack vector for robust ML models, namely backdoor attacks. We show that robust models are susceptible to backdoors. Then, we leverage the inherent properties of robust ML models to detect this attack. AEGIS accurately detects backdoor-infected models and the poisoned class, without any access to the poisoned data. Our work reveals a major strength of robust optimisation in exposing backdoors. Our code and experimental data are available for replication:

https://github.com/sakshiudeshi/Expose-Robust-Backdoors

REFERENCES

[1] Eugene Bagdasaryan and Vitaly Shmatikov. 2020. Blind Backdoors in Deep Learning Models. *arXiv preprint arXiv:2005.03823* (2020).

[2] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. 2019. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. In *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019.*

[3] Edward Chou, Florian Tramèr, Giancarlo Pellegrino, and Dan Boneh. 2018. Sentinet: Detecting physical attacks against deep learning systems. *arXiv preprint arXiv:1812.00292* (2018).

[4] Bao Gia Doan, Ehsan Abbasnejad, and Damith Ranasinghe. 2019. DeepCleanse: A Black-box Input Sanitization Framework Against Backdoor Attacks on Deep-Neural Networks. *arXiv preprint arXiv:1908.03369* (2019).

[5] Min Du, Ruoxi Jia, and Dawn Song. 2019. Robust Anomaly Detection and Backdoor Attack Detection Via Differential Privacy. *arXiv preprint arXiv:1911.07116* (2019).

[6] Min Du, Ruoxi Jia, and Dawn Song. 2019. Robust Anomaly Detection and Backdoor Attack Detection Via Differential Privacy. *CoRR* abs/1911.07116 (2019). arXiv:1911.07116 http://arxiv.org/abs/1911.07116

[7] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, and Dimitris Tsipras. 2019. Robustness (Python Library). https://github.com/MadryLab/robustness

[8] Keinosuke Fukunaga and Larry D. Hostetler. 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory* 21, 1 (1975), 32–40.

[9] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. 2019. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference.* 113–125.

[10] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *CoRR* abs/1708.06733 (2017). http://arxiv.org/abs/1708.06733

[11] Chuan Guo, Ruihan Wu, and Kilian Q Weinberger. 2020. TrojanNet: Embedding Hidden Trojan Horse Models in Neural Networks. *arXiv preprint arXiv:2002.10078* (2020).

[12] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. 2019. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763* (2019).

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

[14] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. 2017. Adversarial Example Defense: Ensembles of Weak Defenses are not Strong. In *11th USENIX Workshop on Offensive Technologies, WOOT 2017, Vancouver, BC, Canada, August 14-15, 2017.*

[15] Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. 2020. On the Effectiveness of Mitigating Data Poisoning Attacks with Gradient Shaping. *arXiv preprint arXiv:2002.11497* (2020).

[16] Xijie Huang, Moustafa Alzantot, and Mani Srivastava. 2019. NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations. *arXiv preprint arXiv:1911.07399* (2019).

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-To-Image Translation With Conditional Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[18] Simran Kaur, Jeremy Cohen, and Zachary C. Lipton. 2019. Are Perceptually-Aligned Gradients a General Property of Robust Classifiers? *CoRR* abs/1910.08640 (2019). http://arxiv.org/abs/1910.08640

[19] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[20] Yann LeCun, Corinna Cortes, and Christopher JC Burges. 1998. The MNIST database of handwritten digits, 1998. *URL http://yann. lecun. com/exdb/mnist* 10 (1998), 34.

[21] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. In *Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings.* 273–294.

[22] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised Image-to-Image Translation Networks. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 700–708. http://papers.nips.cc/paper/6672-unsupervised-image-to-image-translation-networks.pdf

[23] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. 2019. ABS: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security.* 1265–1282.

[24] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning Attack on Neural Networks. In *25nd Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-221, 2018.* The Internet Society.

[25] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.*

[27] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017.* 506–519.

[28] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016. The Limitations of Deep Learning in Adversarial Settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016.* 372–387.

[29] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016.* 582–597.

[30] Ximing Qiao, Yukun Yang, and Hai Li. 2019. Defending Neural Backdoors via Generative Distribution Modeling. In *Advances in Neural Information Processing Systems.* 14004–14013.

[31] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344* (2018).

[32] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified Defenses against Adversarial Examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.* https://openreview.net/forum?id=Bys4ob-Rb

[33] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Image Synthesis with a Single (Robust) Classifier. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada.* 1260–1271.

[34] Aman Sinha, Hongseok Namkoong, and John Duchi. 2017. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571* (2017).

[35] Aman Sinha, Hongseok Namkoong, and John C. Duchi. 2018. Certifying Some Distributional Robustness with Principled Adversarial Training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.* https://openreview.net/forum?id=Hk6kPgZA-

[36] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings.*

[37] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. 2019. Demon in the Variant: Statistical Analysis of DNNs for Robust Backdoor Contamination Detection. *arXiv preprint arXiv:1908.00686* (2019).

[38] Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral Signatures in Backdoor Attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.* 8011–8021.

[39] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness May Be at Odds with Accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.*

[40] Sakshi Udeshi, Shanshan Peng, Gerald Woo, Lionell Loh, Louth Rawshan, and Sudipta Chattopadhyay. 2019. Model Agnostic Defence against Backdoor Attacks in Machine Learning. *CoRR* abs/1908.02203 (2019). http://arxiv.org/abs/1908.02203

[41] Akshaj Kumar Veldanda, Kang Liu, Benjamin Tan, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, Brendan Dolan-Gavitt, and Siddharth Garg. 2020. NNoculation: Broad spectrum and targeted treatment of backdoored DNNs. *arXiv preprint arXiv:2002.08313* (2020).

[42] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy, SP 2019, Proceedings, 20-22 May 2019, San Francisco, California, USA.*

[43] Eric Wong and J. Zico Kolter. 2018. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018.* 5283–5292. http://proceedings.mlr.press/v80/wong18a.html

[44] Eric Wong and Zico Kolter. 2018. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In *International Conference on Machine Learning.* 5286–5295.

[45] Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. 2018. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*. 8400–8409.

[46] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:cs.LG/cs.LG/1708.07747

[47] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. 2019. Detecting AI Trojans Using Meta Neural Analysis. *arXiv preprint arXiv:1910.03137* (2019).

[48] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. 2019. Latent Backdoor Attacks on Deep Neural Networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*. 2041–2055. https://doi.org/10.1145/3319535.3354209

[49] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. DualGAN: Unsupervised Dual Learning for Image-To-Image Translation. In *The IEEE International Conference on Computer Vision (ICCV)*.

[50] Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David Miller. 2020. Backdoor Embedding in Convolutional Neural Network Models via Invisible Perturbation. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*. 97–108.

[51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In *The IEEE International Conference on Computer Vision (ICCV)*.

[52] Minhui Zou, Yang Shi, Chengliang Wang, Fangyu Li, WenZhan Song, and Yu Wang. 2018. Potrojan: powerful neural-level trojan designs in deep learning models. *arXiv preprint arXiv:1802.03043* (2018).

A ADDITIONAL TABLES

Dataset	Epochs	LR	Batch Size	LR Schedule
CIFAR-10	110	0.1	128	Drop by 10 at epochs ∈ [50, 100]
MNIST	100	0.1	128	Drop by 10 at epochs ∈ [50, 100]
Fashion-MNIST	100	0.1	128	Drop by 10 at epochs ∈ [50, 100]

Table 8: Standard hyperparameters used for model training.

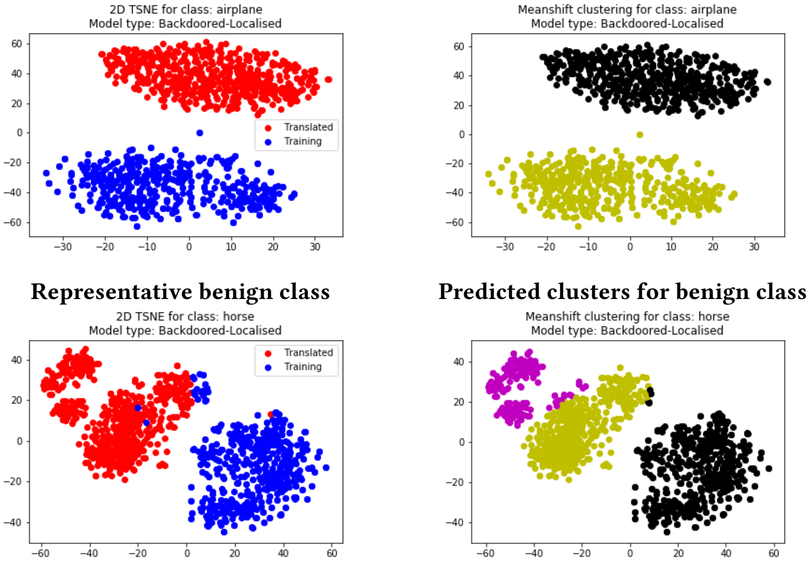
Detection Parameters	All Models		
	MNIST	Fashion-MNIST	CIFAR-10
Epsilon (ϵ)	100	100	500
t-SNE Perplexity	30	30	30
Mean shift Bandwidth	35	28	21

Table 9: Backdoor Detection Parameters

Dataset	Adversarial Accuracy		
	Backdoor-Infected Models		Clean Model
	Localised	Distributed	
CIFAR-10	68.26	68.17	68.64
MNIST	99.51	99.49	99.55
Fashion-MNIST	90.78	90.66	90.91

Table 10: Adversarial accuracy of all robust classifiers

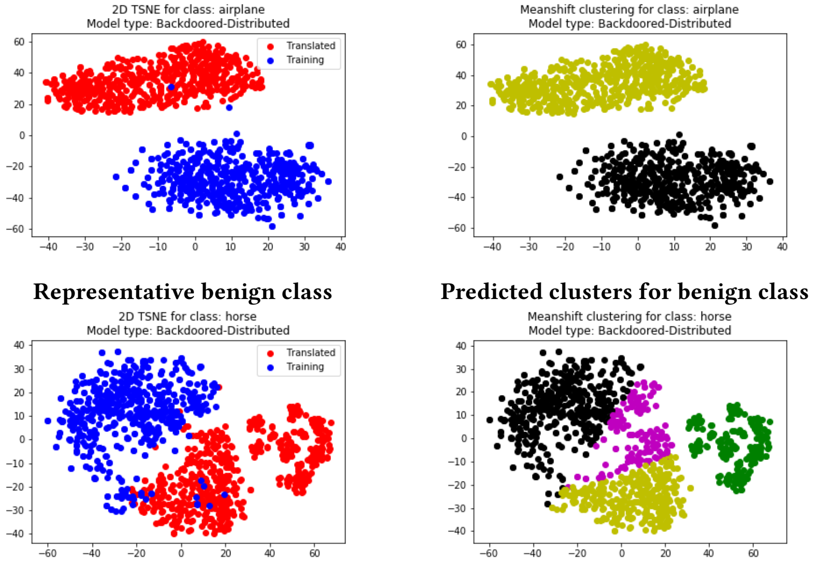
B ADDITIONAL FIGURES



Representative benign class **Predicted clusters for benign class**

Target class **Predicted clusters for backdoored class)**

Figure 11: Feature representation clusters for backdoored CIFAR models (Localised) with target class *Horse* (7). This figure shows class *0* and *7*. The left column shows the feature representations of the translated and the training images, whereas the right column shows the result of the Mean shift clustering on the corresponding points where different colours represent different classes.



Representative benign class **Predicted clusters for benign class**

Target class **Predicted clusters for backdoored class)**

Figure 12: Feature representation clusters for backdoored CIFAR models (Distributed) with target class *Horse* (7). This figure shows class *0* and *7*. The left column shows the feature representations of the translated and the training images, whereas the right column shows the result of the Mean shift clustering on the corresponding points where different colours represent different classes.

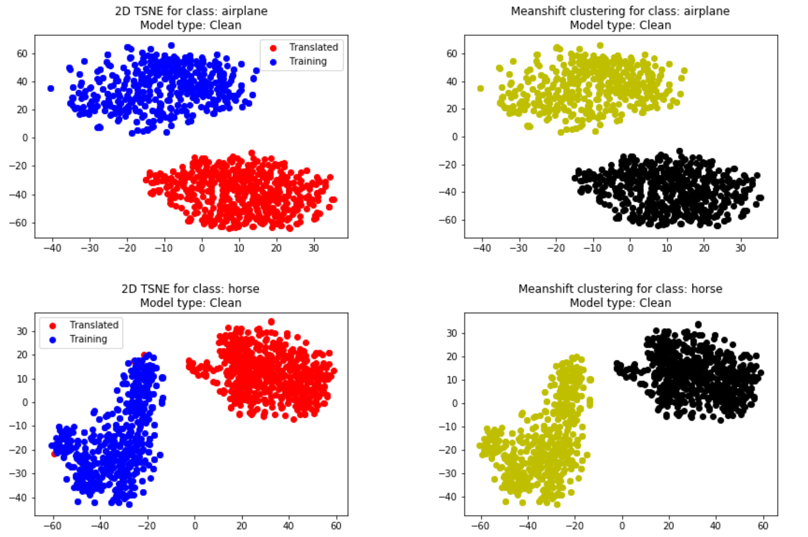


Figure 13: Feature representation clusters for clean CIFAR10 models. This figure shows class *0* and *7*. The left column shows the feature representations of the translated and the training images, whereas the right column shows the result of the Mean shift clustering on the corresponding points where different colours represent different classes.

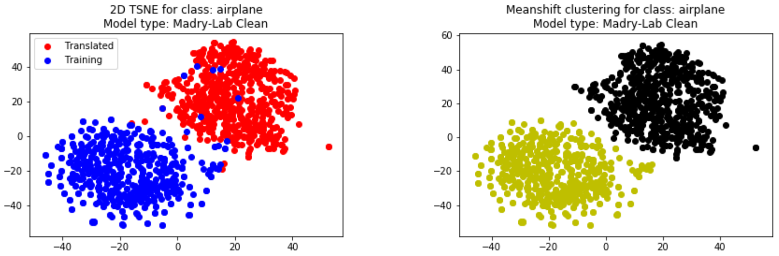


Figure 14: Feature representation clusters for clean CIFAR10 models from Madry-Lab. This figure shows class *0*. The left column shows the feature representations of the translated and the training images, whereas the right column shows the result of the Mean shift clustering on the corresponding points where different colours represent different classes. It is important to note that the translated images and training set images form separate clusters.

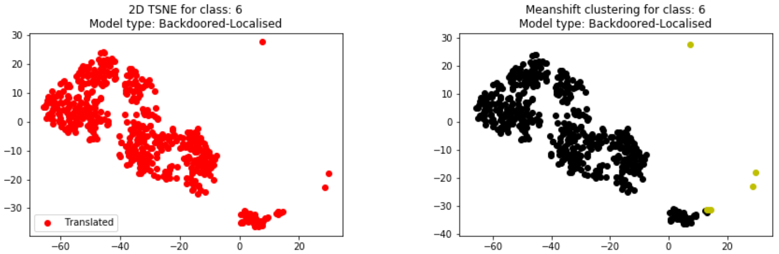
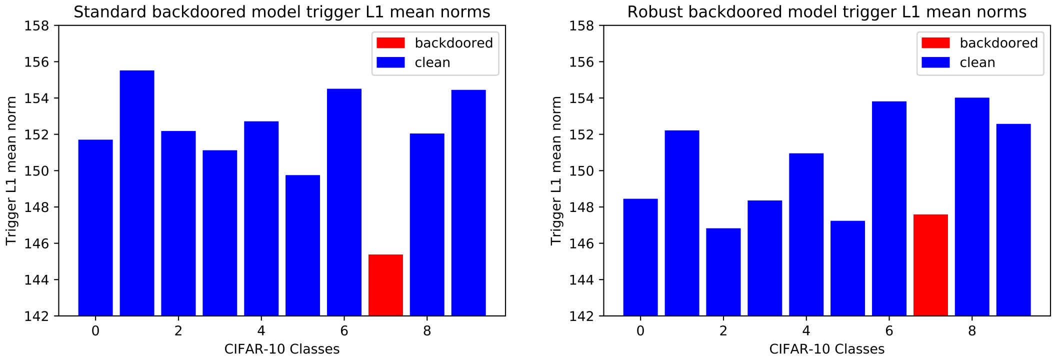


Figure 15: Representative false positives. These kinds of false positives occur when AEGIS only considers the translated images in the detection for backdoors. This figure shows class *6* of a robust MNIST model poisoned with a localised backdoor. The left column shows the feature representations of the translated and the training images, whereas the right column shows the result of the Mean shift clustering on the corresponding points where different colours represent different classes.



(a) Standard model

(b) Robust model

Figure 16: L1 norms (mean) of the reverse engineered triggers for backdoor-infected standard and robust models