The "Beatrix" Resurrections: Robust Backdoor Detection via Gram Matrices

Wanlun Ma[†], Derui Wang[‡], Ruoxi Sun[‡], Minhui Xue[‡], Sheng Wen[†], and Yang Xiang[†]

[†]Swinburne University of Technology, Australia
[‡]CSIRO's Data61, Australia

***Abstract*—Deep Neural Networks (DNNs) are susceptible to backdoor attacks during training. The model corrupted in this way functions normally, but when triggered by certain patterns in the input, produces a predefined target label. Existing defenses usually rely on the assumption of the universal backdoor setting in which poisoned samples share the same uniform trigger. However, recent advanced backdoor attacks show that this assumption is no longer valid in dynamic backdoors where the triggers vary from input to input, thereby defeating the existing defenses.**

In this work, we propose a novel technique, *Beatrix* (backdoor detection via Gram matrix). Beatrix utilizes Gram matrix to capture not only the feature correlations but also the appropriately high-order information of the representations. By learning class-conditional statistics from activation patterns of normal samples, Beatrix can identify poisoned samples by capturing the anomalies in activation patterns. To further improve the performance in identifying target labels, Beatrix leverages kernel-based testing without making any prior assumptions on representation distribution. We demonstrate the effectiveness of our method through theoretical justifications and extensive comparisons with state-of-the-art defensive techniques. The experimental results show that our approach achieves an F1 score of 91.1% in detecting dynamic backdoors, while the state of the art can only reach 36.9%.

I. Introduction

With an explosive growth of Machine Learning (ML) and Artificial Intelligence (AI), Deep Neural Networks (DNNs) are widely adopted in many significant real-world and security-critical scenarios, including facial recognition [77], self-driving navigation [80], and medical diagnosis [66]. Despite these surprising advances, it has been known that DNNs suffer from severe security issues, such as privacy leakage [78], adversarial attacks [30], [82] and backdoor attacks (a.k.a. Trojan attacks) [17], [32]. In particular, the backdoor attack is a technique of embedding a hidden malicious functionality into the DNN, which is activated only when a certain trigger appears. This hidden functionality is usually the misclassification of an input sample to the attacker's desired target class, given the presence of a predefined trigger. For example, a stop sign corrupted by a few pieces of tape will be recognized as a speed limit sign by the navigation system in self-driving cars, which may lead to fatal consequences [25].

BadNets [32] is one of the first works to study the threat of neural backdoors. After that, many variants of backdoor attacks have been proposed [17], [38], [53], [93]. Despite varying in mechanisms and scenarios, all these existing backdoor attacks are premised on adopting a universal (or sample-agnostic) trigger, *i.e.,* different poisoned samples carry the same trigger. This uniform backdoor trigger becomes the Achilles' heel of the backdoor attacks. Based on the fact that the trigger is fixed and universal, existing defensive techniques [14], [18], [27], [52], [89] can easily reconstruct or detect the trigger according to the same behaviors among different poisoned samples. For example, Neural Cleanse [89] utilizes an optimization scheme to synthesize potential trigger patterns that can convert all benign images of other classes to a specific class. The synthesized trigger pattern with abnormally small norm is considered as the attack pattern used by the adversary. Additionally, a defender can also perform run-time detection on each input sample. To examine a malicious input, STRIP [27] superimposes an input image to a set of randomly selected images and measures the entropy of the prediction outputs. If the predictions of the blending images are consistent (*i.e.,* low entropy of prediction outputs), this input is regarded as a malicious one. In addition, SentiNet [18] exploits the explanation technique (*e.g.,* Grad-CAM [73]) to locate a potential trigger region by finding a highly salient contiguous region of a given input.

Witness to the success of the existing defenses, one might think that the threat of backdoor attacks is mitigated or neutralized. Unfortunately, the crucial weakness of such static and sample-agnostic trigger became known to adversaries and they started exploring more advanced approaches in their attacks. In the new attack paradigms, backdoor triggers (referred to as *dynamic* [59] or *sample-specific* [48] triggers) vary from sample to sample. The success of existing defensive techniques [18], [27], [89] mostly relies on the assumption that the triggers are sample-agnostic. However, the sample-specific backdoor attacks break the fundamental assumption of the existing defensive techniques, as the dynamic backdoor introduces diverse information into the trigger pattern, which makes it harder for the defender to model the trigger. As shown in Table I, current backdoor defensive techniques mainly focus on universal backdoor attacks, leaving dynamic backdoor attacks as an unaddressed crucial threat to DNNs (see more discussions in Section II-C). Although a poisoned sample is misclassified to a target label, its intermediate representation has been shown to be different from those of the normal samples in the target class [14], [84], [86]. This observation provides an important indicator to distinguish the malicious samples from the normal ones. However, when you zoom in with

Network and Distributed System Security (NDSS) Symposium 2023
27 February - 3 March 2023, San Diego, CA, USA
ISBN 1-891562-83-5
https://dx.doi.org/10.14722/ndss.2023.23069
www.ndss-symposium.org

TABLE I: A summary of the existing defenses and our work.

Type	Approaches	Detection Target			Black-box Access	No Need of Clean Data	All-to-all Attack	Trigger Assumption		
		Input	Model	Trigger				Universal	Partial	Dynamic
I: Input Masking	STRIP [27]	●	○	○	●	○	○	●	○	○
	Februus [22]	●	○	●	○	○	●	●	○	○
	SentiNet [18]	●	○	●	○	○	●	●	○	○
II: Model Inspection	Neural Cleanse [89]	○	●	●	○	○	○	●	○	○
	ABS [52]	○	●	●	○	○	○	●	○	○
	MNTD [92]	○	●	○	●	○	●	●	○	○
III: Feature Representation	Activation-Clustering [14]	○	●	○	○	●	●	●	○	○
	Spectral-Signature [86]	○	●	○	○	●	●	●	○	○
	SPECTRE [33]	○	●	○	○	○	●	●	○	○
	SCAn [84]	●	●	○	○	○	●	●	●	○
	Beatrix (our work)	●	●	○	○	○	●	●	●	●

○: the item is not supported by the defense; ●: the item is supported by the defense.

order information, out-of-distribution (OOD) samples present more details than trojaned ones since OOD detection requires much higher-order in screening out OOD samples [41], [50], [71], [95]. This observation renders OOD detection methods to be less appealing to trojan detection as the OOD detection signal is too strong. To further demystify the reasons, OOD detection requires sufficient and uncontaminated data as *a priori* knowledge, which is not practical in backdoor detection. Moreover, although the Gram matrix based OOD detector achieves successful performance [71], our experimental results demonstrate that it lacks robustness (using fragile deviation metrics) and efficiency (computing an over-powerful Gramian, *e.g.,* 10-order Gramian as used in the work [71], see details in Section V-A) in detecting trojaned inputs.

Our work. In this paper, we show that Gramian information of dynamically trojaned data points is highly distinct from that of the benign ones. Therefore, if we carefully design the order information (*e.g.,* less than 10-order) and detection metrics, a Gram matrix could be an effective tool for backdoor detection.

Our method, ***Beatrix*** (backdoor detection via gram matrix), captures not only the feature correlations but also the order information of the intermediate representations to reveal subtle changes in the activation pattern caused by backdoor triggers. Beatrix learns robust class-conditional statistics from the activation patterns of legitimate samples to effectively and efficiently harness the Gramian information in trojan detection. In the presence of a backdoor attack, Beatrix can capture the anomalies in the activation patterns since the difference in the feature representations of poisoned samples and legitimate samples is highlighted by our detection metrics.

Contributions. Our main contributions are summarized as follows.

- We present a comprehensive analysis and insights of mainstream defenses to unveil their limitations against dynamic backdoor attacks.
- We develop and implement Beatrix, a novel approach to defend against backdoor attacks. Beatrix utilizes a statistically robust deviation measurement with Gramian information to capture the anomalies in the activation patterns induced by poisoned samples. Beatrix also leverages Regularized Maximum Mean Discrepancy to further improve the performance in identifying infected classes.
- We demonstrate the effectiveness and robustness of our proposed method through theoretical justifications and extensive comparisons with state-of-the-art defensive techniques. We show that Beatrix can effectively detect sample-specific backdoor attacks and significantly outperform the existing defenses.

II. Background

In this section, we begin by briefly introducing the concept of Gram Matrix and the advances in backdoor attacks. We then discuss the limitations of existing defenses.

A. Gram Matrix in DNNs

Gramian information is widely used in areas such as Gaussian process regression [67] and style transfer learning [28]. It computes the inner products of a set of m-dimensional vectors. The vectors, for instance, can be random variables in Gaussian process or vectorized internal activation patterns in style transfer. Formally, we suppose $A := \{a_k | a_k \in \mathbb{R}^m\}_{k=1}^n$ is a set of m-dimensional random variables, and then the gramian information between $a_i, a_j \in A$ is defined as $G_{ij} = \sum_k^m a_{ik} \cdot a_{jk}$. Thus, the Gram matrix G is an $n \times n$ symmetric matrix containing the gramian information between each pair of random variables in A. Since the off-diagonal entries of G represent the pairwise correlation between a_i and a_j, Gram matrix can be used as an covariance matrix in Gaussian process regression [67]. On the other hand, due to its effectiveness in feature learning, the Gram matrix shows remarkable performance in capturing stylistic attributes (*e.g.,* textures and patterns) in neural activations [46]. The high-order form of the Gram matrix has also been leveraged to improve OOD detectability [71]. The entries of p-th order of the Gram matrix is defined as $G_{ij}^p = \left(a_i^p a_j^{pT}\right)^{1/p}$, where p is the exponent.

B. Backdoor Attacks

Backdoor attacks are a technique of injecting some hidden malicious functionality into ML systems [26], [56], [62]. The injected backdoor is activated only when a certain trigger appears in the input. This hidden functionality usually results in misclassifying the input sample into a target class predefined by the attacker.

Universal (sample-agnostic) backdoor. Although various backdoor attacks [17], [32], [38], [93] have been proposed, the majority of them have a static trigger setting, meaning that there is only one universal trigger and any clean sample with that trigger will be misclassified to the target label [47]. Particularly, in the most common backdoor attack (*i.e.*, BadNets [32]), an adversary can form a backdoor trigger $t = (m, p)$, where m and p denote the blending mask and the trigger pattern, respectively. During the training of a DNN, a clean training sample pair such as (x, y) is randomly replaced by the poisoned pair (x_{bd}, y_{bd}) with a certain probability using the trigger embedding function $\mathcal{B}$, which is defined as

$$\begin{aligned} x_{bd} &= \mathcal{B}(x, t) && (1) \\ &= x \cdot (1 - m) + p \cdot m && (2) \end{aligned}$$

Partial (source-specific) backdoor. In the partial attack, only samples in a specific source class can activate the backdoor and be misclassified into the target class set by the trigger [84], [89]. As for samples in other classes, the trigger will not activate the backdoor. It is worth noting that all the trojaned samples still share the same uniform trigger in the source-specific backdoor attack.

Dynamic (sample-specific) backdoor. Compared to the universal and partial backdoor attacks, dynamic backdoor attacks [48], [59], [70] make triggers that vary from sample to sample and this complicates the detection of such backdoors.

To the best of our knowledge, there exist three dynamic backdoor attacks [48], [59], [70] [1]. All of them utilize a trigger generating network to launch dynamic backdoor attacks. In this work, in addition to universal backdoors, we try defending against the state-of-the-art dynamic ones, and more specifically, invisible sample-specific [48] and input-aware dynamic backdoor [59] attacks. (We do not consider [70] since its triggers are not sample-specific and the code is not released.) Both of them consider the uniqueness and exclusiveness of triggers, *i.e.*, each sample has a unique trigger which is non-reusable for any other sample. Herein, we will brief their attack paradigms, and readers can find more details from the original papers [48], [59].

Compared to the fixed and universal backdoor triggers, a *dynamic* or *sample-specific* trigger is a function of the corresponding input sample x. Suppose g is a trigger generator. It can be defined as a function mapping an input from the sample space $\mathcal{X}$ to a trigger in the trigger space $\mathcal{T}$:

$$g : x \in \mathcal{X} \to t \in \mathcal{T}. \tag{3}$$

Additionally, the dynamic triggers should be non-reusable and unique. Henceforth,

$$\arg\max_{y^*} f_{y^*}(\mathcal{B}(x, g(\hat{x}))) = y_{bd} \cdot \mathbb{1}(x = \hat{x}) + y \cdot \mathbb{1}(x \neq \hat{x}), \tag{4}$$

where y_{bd} is an backdoor target and y is the ground truth label of x. In other words, a clean sample x with the trigger generated based on another sample will not activate the hidden backdoor in the model f.

TABLE II: Limitations against Dynamic Backdoors

Defense	Limitation
Type-I	Perturbation-resistant assumption of triggers
Type-II	Can only reconstruct sample-agnostic triggers
Type-III	Strong assumption on the distribution of feature representations

Both dynamic backdoor attacks and adversarial attacks aim to make models misbehave and share many similarities. Still, they have certain differences. Given a classifier f, an adversary injects dynamic backdoor by jointly training a trigger generation function g with f on a clean distribution p_{data} and a poisoning distribution p_{bd}. The overall training objective is

$$\max_{g,f} \Pr_{(x,y_{bd})\sim p_{bd}}[f(x+g(x)) = y_{bd}] + \max_{f} \Pr_{(x,y)\sim p_{data}}[f(x) = y]. \tag{5}$$

The training objective may recall a similar objective in targeted evasive (adversarial) attacks in which an adversary $\hat{g} : \max_{\hat{g}} \Pr[f(x + \hat{g}(x)) = y_t]$. However, it is easily found that the dynamic backdoor attacker has the capability of training g on the training dataset of f while the adversarial attacker optimizes $\hat{g}$ either in a per-sample manner or over an external dataset whose distribution is similar to that of the training dataset. As a ramification, $f_{y_{bd}}(x + g(x)) \gg f_{y^* : y^* \neq y_{bd}}(x + g(x))$, which means f tends to be overfitted to the data distribution modified by $g(x)$. Such attack was found hard to be detected by adversarial detection methods due to the statistics of f has been changed and the high confidence in the misclassification of $x + g(x)$ [8]. In addition, further modifying the triggers during the inference stage harms the stealthiness of the dynamic backdoor attacks and requires more capability from the attacker's side, which breaks the threat model of the dynamic backdoor. Such modification generates a separate adversarial attack rather than being a part of the dynamic backdoor attack, which falls out of the scope of this paper.

C. Existing Defenses

As summarized in Table II, Type-I defenses assume that the backdoor trigger is resistant to perturbations. Thus, the trigger regions or trigger-carrying images can cause the same misclassification when overlaid on other clean images [18], [22], [27]. However, this assumption is violated in dynamic backdoors where the trigger is only activated for a specific sample. Moreover, Type-II defenses try to reconstruct a universal trigger that can convert any clean sample to the same target class [52], [89]. Unfortunately, this reconstructed trigger is only valid for sample-agnostic backdoor. In contrast, the dynamic backdoor triggers vary from sample to sample, rendering the reconstructed universal trigger to be totally different from the actual dynamic triggers. In addition, Type-III defenses attempt to distinguish the difference between the representations of clean samples and those of trojaned samples. However, they either use trivial clustering techniques [14], [86] or model the representations with Gaussian distributions [33], [84], which cannot resist dynamic backdoor attacks. The detailed results can be found in our experimental analysis in Section V-B.

A case study of SCAn. Although SCAn [84] reveals the drawbacks of current defenses relying on the sample-agnostic backdoor assumption, *it only considers the partial backdoor but leaves the sample-specific attack as an open problem.* We argue that there are three limitations of SCAn, which

[1] We include [48] here because it is a sample-specific backdoor attack which is very similar to the input-aware dynamic attack of [59], though the authors do not use the term *dynamic* in their paper [48] explicitly.

indicates SCAn cannot be extended to defending against the sample-specific backdoor. Firstly, SCAn assumes that the representations of normal and trojaned samples can be distinguished by the first moment (mean) discrepancy (*Two-component decomposition assumption* in SCAn). However, in the dynamic backdoor attack, the first moment information becomes less discriminative. Secondly, SCAn models the feature distribution by a Gaussian distribution under the Linear Discriminant Analysis (LDA) [57] assumption, *i.e.*, different mean values but same covariance for the distributions of clean and trojaned feature representations (*Universal Variance assumption* in SCAn). However, our normality test shown in Figure 1 resonates with the observation of previous work [95] that the feature space of DNNs does not necessarily conform with a Gaussian distribution. Thirdly, as demonstrated by our theoretical analysis in Appendix B, the essence of SCAn is to compute the weighted Mahalanobis distance between the representations of clean samples and those of trojaned samples, indicating that the effectiveness of SCAn is also dependent on the weighted value (the ratio of trojaned samples). According to their estimate [84], SCAn needs to discern roughly 50 trojaned images before it can reliably detect further attacks. This is a severe drawback for security-critical applications. For example, the adversary may have bypassed an authentication system dozens of times before being caught.

III. Overview and Motivation

To overcome the limitations of existing defenses, we propose a novel backdoor detection approach that covers the sample-specific backdoor attack. We first introduce the threat model in our work and then present our key observations and ideas. Finally, we analyze the superiority of Gram matrix.

A. Threat Model

We consider the standard threat model which is consistent with that of the most recent backdoor attack and defense studies [32], [84], [89].

Adversary. Similar to most backdoor poisoning settings, the goal of the adversary is to deliberately inject one or more backdoors into the target model. The compromised model performs well on clean samples, whereas it misclassifies attack samples (trigger-carrying samples) to the predefined target label. We assume that the adversary can access the training set of the model, and is capable of poisoning the training data without major constraints, but has no direct access to the model. This scenario allows us to study the attack under worst-case conditions from the defender's point of view.

Defender. The goal of the defender is to perform input-level detection to determine whether an input will trigger a malicious behavior in a untrusted model in an online setting such as the Machine-Learning-as-a-Service scenario. Furthermore, the defender aims to tell the infected classes of a backdoored model based on the instances it classifies. We assume that the defender has white-box access to the target model, including the feature representation in the intermediate layers. Additionally, the defender needs a small set of clean data to help it with the detection, which was also a requirement in the previous works [18], [27], [84], [92].

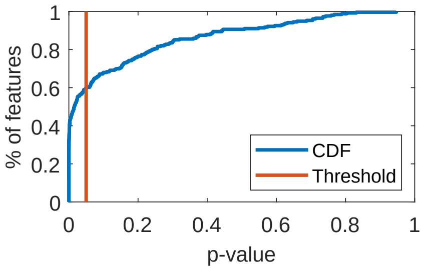


Fig. 1: Normality Test by Shapiro-Wilk test. We can find that about 60% features do **NOT** follow a normal distribution under a 95% confidence score. This demonstrates that the Gaussian distribution assumption in [33], [84] is untenable in more advanced attacks such as the dynamic backdoor attack.

B. Intuition and Key Idea

Intuition. A key observation is that the clean samples of a certain class and the trojaned (trigger-carrying) samples targeted at that class are disjoint in the pixel space. Consequently, even though a trojaned sample is misclassified into the target label, its intermediate representation is somehow different from those of normal samples of the target class. The anomaly triggered by the trojaned samples can be characterized by inconsistencies between the intermediate feature representations and their predicted labels. This observation provides a basis for investigating the problem of characterizing trojaned samples from the perspective of OOD detection.

A predictive uncertainty study [7] reveals that DNNs perform well on the samples drawn from the distribution seen in the training phase, but tend to behave unexpectedly when they encounter OOD samples that lie far from the training distribution. Analogously, trojaned samples can be thought of as OOD samples drawn from a distinct distribution in contrast to the distribution of clean samples. Therefore, we believe there is a link between the detection of trojaned samples and the OOD detection [41], [50], [71], [95].

However, we notice there exist differences between OOD detection and backdoor detection. First, the feature representations of OOD samples can be effectively modeled by Gaussian distributions. This assumption shows superior performance on OOD detection tasks [41], [95]. On the contrary, Gaussian distributions are less capable of modeling the features of trojaned samples, due to the complexity and diversity of backdoor triggers. This dilemma is further emphasized by dynamic backdoors. As our experimental results show in Section V-B, backdoor detection methods that use Gaussian distribution, *e.g.*, SCAn [84], achieve suboptimal performance in the identification of dynamic backdoors. A normality test in Figure 1 also exposes the problem. Second, less attention has been paid to adversarial robustness in OOD detection methods. *A priori* knowledge of in-distribution/clean samples is a key piece of information for many OOD detectors [41], [71], [95] and backdoor detection methods [27], [89]. The OOD detection task assumes that a set of clean samples used for training the detector can be well curated. However, this assumption is challenged in the scenario of backdoor detection since poisoned samples may carry invisible triggers [48], [76] which can hardly be filtered even by manual inspection. The lack of adversarial robustness restricts the deployment of OOD detectors in detecting trojaned inputs (see V-A in more

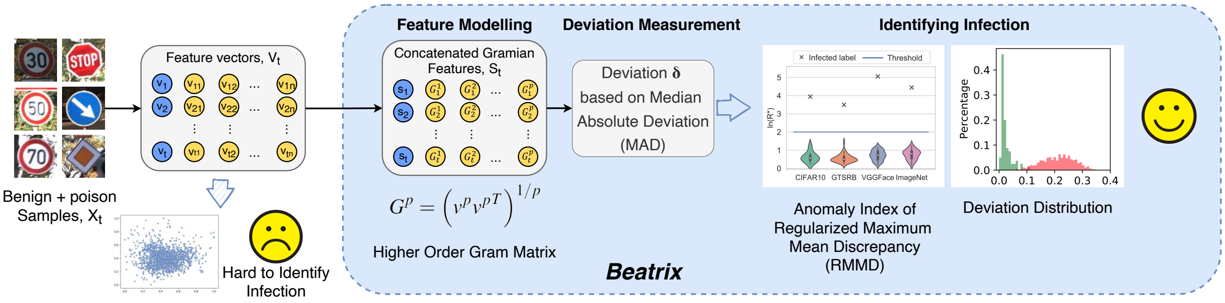

Benign + poison Samples, X_t
Feature vectors, V_t
v_1 v_{11} v_{12} … v_{1n}
v_2 v_{21} v_{22} … v_{2n}
v_t v_{t1} v_{t2} … v_{tn}
Hard to Identify Infection
Feature Modelling
Concatenated Gramian Features, S_t
s_1 G_1^1 G_1^2 … G_1^p
s_2 G_2^1 G_2^2 … G_2^p
s_t G_t^1 G_t^2 … G_t^p
$G^p = \left(v^p v^{pT}\right)^{1/p}$
Higher Order Gram Matrix
Deviation Measurement
Deviation δ based on Median Absolute Deviation (MAD)
Identifying Infection
× Infected label
Threshold
ln(R*)
0
1
2
3
4
5
CIFAR10
GTSRB
VGGFace
ImageNet
Anomaly Index of Regularized Maximum Mean Discrepancy (RMMD)
Percentage
0.0
0.1
0.2
0.3
0.4
Deviation Distribution
Beatrix

Fig. 2: An overview of Beatrix.

detail). Finally, OOD detectors require sufficient in-distribution data, or even OOD data [41], [50], [95]. This requirement ensures that the parameters of the detection models can be effectively estimated. However, there are usually limited clean data available to the backdoor defender. Thus, the requirements for statistical robustness under limited data are different in OOD and backdoor detection methods.

The key idea. From our observation, backdoor defense can be viewed as the problem of detecting OOD. We argue that the problem of finding a robust detector for neural backdoors can be connected to feature modeling methods used in areas such as Gaussian process regression, style transfer, and OOD detection [28], [67], [71]. In this paper, we find that although the trojaned and clean samples are deeply fused in the original feature space, they are distinguishable in the Gramian feature space, indicating Gram matrix is an effective tool for feature modeling. The Gram matrix derives from the inner products of feature maps across different channels. Thus, Gram matrices not only consider features in each individual channel but also incorporate the feature correlations across channels [46], [71].

We note that the previous representation-based backdoor detection methods either use trivial clustering techniques [14], [86] or Gaussian models [33], [84] in the detection. These methods ignore the high-order information and only consider the first moment (mean) discrepancy between clean and trojaned samples. However, the simplification reduces the discriminative power of the methods against more complex attacks such as dynamic backdoors. To tackle the problem, we turn to high-order statistics of the feature representations since they are more sensitive to the changes in the feature space, due to its higher-power format. Our method employs not only first-order moments but also high-order moments for feature modeling. We utilize the Gram matrix and its appropriately high-order forms to capture not only the feature correlations but also appropriately high-order information to detect trojaned samples. In addition, considering the adversarial robustness and statistical robustness in backdoor detection, we do not use the multivariate Gaussian to model the trojaned samples in the deviation measurement as Gaussian models only perform well with sufficient data [64]. Instead, we utilize Median Absolute Deviation (MAD), a more robust estimation of statistical dispersion, to measure the deviation of trojaned samples. Finally, when the training dataset of a given class is contaminated, the set of examples can be viewed as a mixture of two subgroups [14], [33], [84], [86]. In contrast to previous works [33], [84] assuming that the two subgroups follow Gaussian distributions with two different means but the same covariance, we employ Regularized Maximum Mean Discrepancy (RMMD) [20] to enhance the adversarial robustness of our method. RMMD is a Kernel-based two-sample testing technique which does not have any assumption on the distributions. RMMD performs a hypothesis test on whether the feature representations in a given class are drawn from a mixture group (*i.e.*, contaminated class) or a single group (*i.e.*, uncontaminated class).

C. Theoretical Analysis of Gram Matrix

Let $x^p \sim P_{c,p}$ and $y^p \sim P_{t,p}$ be p-th order representation vectors sampled from the clean distribution $P_{c,p}$ and the trojaned distribution $P_{t,p}$, respectively. Beatrix models the feature representations without any assumption on $P_{c,p}$ and $P_{t,p}$. Instead, Beatrix relies on Gram matrix to extract discriminative information of $P_{c,p}$ and $P_{t,p}$ from their statistical moments [13]. Let u_1^p and S_1^p be the mean vector and the covariance matrix of x^p, respectively. The second raw moment of x^p over $P_{c,p}$ is:

$$\begin{aligned} E(x^p x^{pT}) &= E(x^p)E(x^p)^T + E[(x^p - u_1^p)(x^p - u_1^p)^T] \\ &= u_1^p u_1^{pT} + S_1^p. \end{aligned} \tag{6}$$

The Gramian feature of x^p is defined as $G_{x^p} = x^p x^{pT}$. Then, $E(G_{x^p}) = E(x^p x^{pT}) = u_1^p u_1^{pT} + S_1^p$. Similarly, let u_2^p and S_2^p be the mean vector and the covariance of y^p, we have $E(G_{y^p}) = u_2^p u_2^{pT} + S_2^p$ over $y^p \sim P_t$. Therefore, the expected discriminative information captured by the Gram matrices over different exponents can be represented from the prospective of statistical moments:

$$\begin{aligned} M(P_c, P_t) &= E_{p\in\mathbb{Z}^+}[E(G_{x^p}) - E(G_{y^p})] \\ &= E_{p\in\mathbb{Z}^+}[u_1^p u_1^{pT} - u_2^p u_2^{pT} + S_1^p - S_2^p], \end{aligned} \tag{7}$$

where P_c is a collection of distributions of clean representations with elements of different powers and P_t is that of trojaned representations. Equation 7 shows that Gramian features not only capture the first moment discrepancy (*i.e.*, $u_1^p u_1^{pT} - u_2^p u_2^{pT}$) like SCAn (when $p = 1$, see Equation 21), but also the second moment discrepancy (*i.e.*, $S_1^p - S_2^p$). Compared to previous methods such as SCAn, a trojaned y can still be distinguished from clean representations when the clean and trojaned distributions have the same mean. Moreover, by considering various p values, the second moment discrepancy can capture more information about high-order features and better models $M(P_c, P_t)$.

IV. Design of Beatrix

In this section, we provide the details of our approach to detecting backdoor attacks. The framework of Beatrix is illustrated in Figure 2.

A. Feature Modeling via Gram Matrices

Formally, let h be the sub-model up to the l-th layer of a DNN model. Then, the feature representation of the input sample x at the l-th layer of the DNN model is defined as $h(x) = v \in \mathbb{R}^{n \times m}$, where n is the number of channels at layer l and m is the height times the width of each feature map. The features correlations between channels can be expressed by

$$G = vv^T, \tag{8}$$

where $G \in \mathbb{R}^{n \times n}$ denotes the Gram matrix of the feature maps in an inner product space. In order to capture more prominent activations in feature maps, we also use the high-order Gram matrix

$$G^p = \left(v^p v^{pT}\right)^{1/p}, \tag{9}$$

where v^p denotes the p-th power of the feature representation v, and G^p is the p-th order Gram matrix of v.

The off-diagonal entries of G^p represent the pairwise correlation between feature maps at the l-th layer while the entries at the diagonal only relate to a single feature map. Since the matrix G^p is symmetric, we only need the upper (or lower) triangular part of it. In particular, the vectorized triangular matrix which contains the entries on and above (or below) the main diagonal, can form a $\frac{1}{2}n(n+1)$-dimensional vector like $\vec{G^p}$.

We can compute $\vec{G^p}$ for each order $p \in \{1, ..., P\}$, where P is a hyperparameter representing the bound of the order. By concatenating all the output vectors $\vec{G^p}$, we can derive a new representation vector $s = [\vec{G^1},\ \vec{G^2},\ ...,\ \vec{G^P}] \in \mathbb{R}^{\frac{1}{2}n(n+1)P}$ for the input sample x.

Let $\mathcal{X}_t$ denote a set of clean samples in class t. $\mathcal{X}_t$ has a feature presentation set $\mathcal{V}_t = \{v_i := h(x_i),\ x_i \in \mathcal{X}_t\}$ and a concatenated Gramian feature set $\mathcal{S}_t = \{s_i,\ i \in \{1, 2, ..., |\mathcal{X}_t|\}\}$. The task of backdoor detection can be formulated as an outlier detection problem: given the feature $\hat{s}$ of an input sample $\hat{x}$ and its predicted label $\hat{y}_t$ by the target model f, we try to determine whether $\hat{s}$ is an outlier, with respect to $\mathcal{S}_t$ based on the statistical properties of $\mathcal{S}_t$.

B. Deviation Measurement

A natural choice of computing the deviation of a point $\hat{s}$ is to build a multivariate Gaussian model of $\mathcal{S}_t$. However, the problem is non-trivial because of *i)* the large dimensionality of the feature vector s and *ii)* the limited number of clean samples for estimating Gaussian parameters (especially, the covariance matrix). Therefore, building a multivariate Gaussian model for high dimensional variables is not statistically robust when there are limited data samples available. Additionally, since the feature modeling with Gram matrices has already considered the feature correlations, we can just simplify the problem and model each element in $\mathcal{S}_t$ independently. Thus, high dimensional estimation can be simplified into one dimensional estimation for each independent element.

In the simplified case, one may still consider using the Gaussian model but its univariate version to estimate mean and standard deviation of the features. However, recall that there is limited clean data available for the defender, the estimation results (*i.e.*, mean and standard deviation) are easier to be affected by the outliers as Gaussian models only perform well with sufficient data [64]. More importantly, the individual elements of s may not follow a Gaussian distribution strictly.

Instead of using a univariate Gaussian model, we propose to utilize Median Absolute Deviation (MAD) which is known to be more resilient to outliers in a dataset than the standard deviation $\hat{\sigma}$ [42]. The absolute deviations between all data points and their medians are gained before MAD is employed as the median of these absolute deviations.

Given the set of concatenated Gramian features $\mathcal{S}_t$ of all clean samples $\mathcal{X}_t$ in class t, we can compute the median and the MAD with respect to each concatenated Gramian vector:

$$\tilde{s}_j = median(\{s_{ij},\ \forall i \in \{1, 2, ..., |\mathcal{X}_t|\}\}), \tag{10}$$

$$MAD_j = median(\{|s_{ij} - \tilde{s}_j|,\ \forall i \in \{1, 2, ..., |\mathcal{X}_t|\}\}). \tag{11}$$

Then the deviation of the observed j-th value $\hat{s}_j$ in the candidate feature point $\hat{s}$ is defined as:

$$\delta_j = \delta(\hat{s}_j) \tag{12}$$

$$= \begin{cases} 0 & ,\ if \quad min \leq \hat{s}_j \leq max, \\ \dfrac{min - \hat{s}_j}{min}, & if \quad \hat{s}_j \leq min, \\ \dfrac{\hat{s}_j - max}{max}, & if \quad max \leq \hat{s}_j, \end{cases} \tag{13}$$

where $min = \tilde{s}_j - k \cdot MAD$, $max = \tilde{s}_j + k \cdot MAD$, and k is a predefined scale factor that is set to 10 in our case.

Then the deviation of the candidate feature point $\hat{s}$ is the sum of the deviation values over all entries in $\hat{s}$:

$$\delta = \frac{2}{n(n+1)P} \sum_{j=1}^{\frac{1}{2}n(n+1)P} \delta_j. \tag{14}$$

Threshold determination. The detection boundary of Beatrix is estimated by benign inputs. Due to the limited number of clean data available to the defender, we employ bootstrapping to compute the deviations of benign inputs [23]. Specifically, we randomly draw $\frac{1}{T}$ samples from the clean dataset as testing samples. The remaining samples are used as training samples to estimate the min/max values. The procedure is repeated for T iterations to obtain the deviations of benign samples. The detection boundary can be determined by the defender when choosing different percentiles like STRIP [27]. For example, the defender can choose 95% as the detection boundary. This means that 95% of the deviations of benign samples are less than this detection boundary.

C. Identifying Infected Labels

The performance of Beatrix can be further improved through a local refinement of the detection results to reduce false positives. Since the detection threshold in Section IV-B is predefined, there could be false positive in the detection results. In an offline setting, Beatrix accumulates the historical detection results for a second statistical analysis to ablate false trojaned targets given by the threshold thereof. In the presence of a backdoor attack, the feature representations of samples in the infected class can be considered as a mixture of two subgroups [14], [33], [84], [86]. However, previous works [33], [84] assume that these two subgroups follow Gaussian distributions with two different means but the same covariance. Therefore, they perform a hypothesis testing to determine whether these two distributions are significantly different. However, as we discussed in Section II-C, the Gaussian assumption is not tenable in more complex scenarios, such as dynamic backdoor attacks.

Therefore, we resort to the Kernel-based two-sample testing which addresses whether two sets of samples are identically distributed without assumption on their distributions [19], [20], [31], [88]. A popular test statistic for this problem is the Maximum Mean Discrepancy (MMD) [31], which is defined based on a positive definite kernel function k [72]. Kernel methods provide the embedding of a distribution in a reproducing kernel Hilbert space (RKHS). For MMD with a linear kernel, $k(x, x') = \langle x, x' \rangle$, it measures the distribution distance under their first moment discrepancy [58]. In practice, a common option is to use the Gaussian kernel $k(x, x') = exp(-\beta \|x - x'\|_2^2)$, which contains infinite order of moments by looking at its Taylor series [49].

In this work, we use an extension of MMD metric, termed Regularized MMD (RMMD), which incorporates two penalty terms to achieve better performance when the two sample sets are small and imbalanced [20]:

$$\begin{aligned} RMMD(P,Q) &= MMD(P,Q)^2 - \lambda_P \|\mu_P\|_{\mathcal{H}}^2 - \lambda_Q \|\mu_Q\|_{\mathcal{H}}^2 \quad (15) \\ &= \|\mu_P - \mu_Q\|_{\mathcal{H}}^2 - \lambda_P \|\mu_P\|_{\mathcal{H}}^2 - \lambda_Q \|\mu_Q\|_{\mathcal{H}}^2, \quad (16) \end{aligned}$$

where P and Q denote the representation distributions represented by the samples in the two subgroups obtained from the deviation measurement.

According to the work [20], this test statistic follows an asymptotic normal distribution based on theorems in [36], [74]. Similar to Neural Cleanse [89] and SCAn [84], we can leverage the Median Absolute Deviation to identify the infected label(s) with abnormally large values of RMMD statistic instead of directly computing the p-value of this test. Specifically, we denote R_t as the RMMD statistic of class t, and then anomaly index R_t^* is defined as:

$$R_t^* = |R_t - \tilde{R}| / (MAD(\tilde{R}) * \eta), \quad (17)$$

$$\text{where} \quad \tilde{R} = median(\{R_t : t \in \mathcal{L}\}), \quad (18)$$

$$MAD(\tilde{R}) = median(\{R_t - \tilde{R}| : t \in \mathcal{L}\}). \quad (19)$$

As R_t follows an asymptotic normal distribution, we apply a constant factor $\eta = 1.4826$ to the anomaly index. We identify any label with an anomaly index $R_t^* \geq e^2$ as an infected label with the confidence probability $\geq (1 - 10^{-9})$ [84].

V. Evaluation

In this section, we first evaluate the effectiveness of our proposed method against the dynamic backdoor and then compare Beatrix with state-of-the-art defensive techniques. Finally, we also demonstrate the robustness of Beatrix against other attacks. The datasets and model structures used in our experiments are summarized in Table III. We provide detailed introduction of the experiment setup in Appendix C.

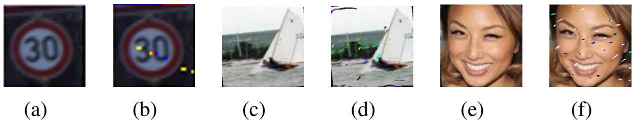


Fig. 3: Examples of poisoned samples under the input-aware dynamic backdoor. (a), (c) and (e) are clean images, (b), (d) and (f) are poisoned images.

A. Effectiveness Against Dynamic Backdoor

Attack configuration. We implement the input-aware dynamic backdoor attack [59] using the code released by the authors [3]. Figure 3 illustrates several examples of poisoned samples. We conduct the common single-target attack, *i.e.*, the target label is the same for all trojaned samples. We set both the backdoor probability (trojaned samples with their paired triggers) and the cross-trigger probability (trojaned samples with inconsistent triggers) as 0.1. For all the four datasets, the backdoor attack success rates (ASR) are almost 100% while still achieving a comparable performance on clean data as the benign models do, as shown in Table IV. It is worth noting that the cross-trigger accuracy (the accuracy of classifying images containing dynamic triggers deliberately generated for other images) is over 80% on all the four datasets, and this shows the nonreusability and uniqueness of the triggers on mismatched clean images. The evaluation results on the invisible sample-specific backdoor attack [48] are shown in Section V-C.

Effectiveness on various datasets. From each dataset, we randomly select 30 images per class as a clean dataset for the defender. The clean dataset accounts for no more than 6% of the whole dataset. The bound of the order of Gram matrix is set as 9. As is shown later, the detection effectiveness is stable when $P \geq 4$. The experimental results on the four datasets show that our defensive technique is very effective in detecting the dynamic backdoor attack. Figure 4 illustrates the logarithmic anomaly index values $ln(R^*)$ of infected and uninfected labels. All infected labels have much larger anomaly index values, compared to uninfected labels. This demonstrates that our defensive technique can effectively detect target classes in infected models on various datasets and model architectures. Moreover, Figure 5 illustrates that the deviations of the trojaned samples are larger than those of the benign ones. This demonstrates that our method can also effectively distinguish benign from polluted samples.

Clean data for deviation measurement. To achieve a high accuracy in discriminating the mixed representations of benign and poisoned samples, a small set of clean samples is required for estimating the threshold in advance [18], [27], [84], [92]. The above experiments show that our method can effectively detect the dynamic attack with 30 clean images per class. Our study further demonstrates that our method can perform effectively with less clean data and even with the contaminated data. As shown in Figure 6, we can find that even with only 8 clean images, Beatrix can still accurately identify the infected class. Moreover, we also test the robustness of Beatrix when the clean data is moderately contaminated. Figure 7 shows that Beatrix is still effective when no more than 16% (or 5 images) of the clean images per class are contaminated with poisoned ones. We also compare Beatrix with the OOD detection method [71] under this data contaminating scenario.

TABLE III: Detailed information about dataset, model architecture and clean accuracy.

Dataset	# of Classes	# of Training Images	# of Testing Images	Input size	Model Architecture	Top-1 accuracy
CIFAR10	10	50000	10000	32 × 32 × 3	PreActResNet18	94.5%
GTSRB	43	39209	12630	32 × 32 × 3	PreActResNet18	99.1%
VGGFace	100	38644	9661	224 × 224 × 3	VGG16	90.1%
ImageNet	100	50000	10000	224 × 224 × 3	ResNet101	83.8%

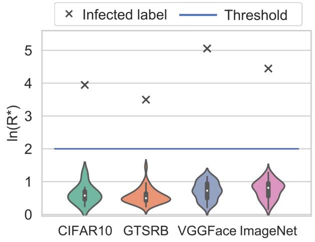


Fig. 4: The logarithmic anomaly index of infected labels on the four datasets.

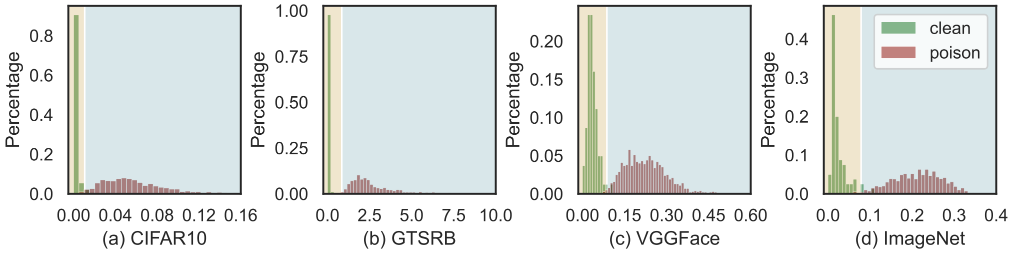


Fig. 5: Deviation distribution of benign and trojaned samples. The trojaned sample shows a much larger deviation than benign samples. The color boundary in the background indicates the decision threshold (same for the figures in the following sections).

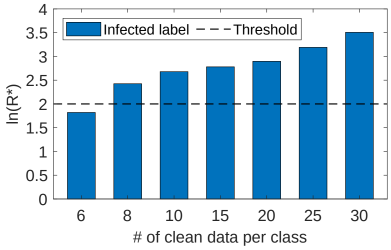


Fig. 6: The logarithmic anomaly index of infected labels when using different number of clean data.

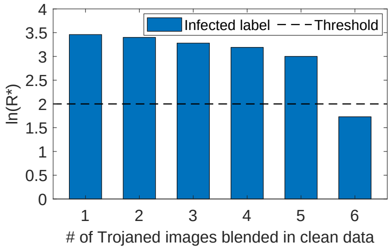


Fig. 7: The logarithmic anomaly index of infected labels when clean data is contaminated.

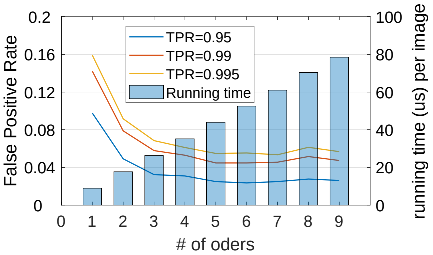


Fig. 8: False positive rate of benign images when incorporating different bound on the order of Gram matrix.

TABLE IV: Attack success rate, cross-trigger accuracy and classification accuracy of infected models.

Dataset	Infected Model			Benign Model
	Attack Success Rate	Cross-trigger Accuracy	Clean Accuracy	Clean Accuracy
CIFAR10	99.4%	88.6%	93.9%	94.5%
GTSRB	99.7%	96.1%	99.2%	99.1%
VGGFace	98.5%	82.5%	89.8%	90.1%
ImageNet	99.5%	81.3%	83.5%	83.8%

Figure 9 shows that Beatrix is much more robust than the OOD detector when the clean data is contaminated by trojaned samples. As a result, the OOD detection method [71] cannot be directly applied to the backdoor detection.

The order of Gram matrix. In the above experiments, we consider Gram matrices from the first to the ninth order. However, incorporating high-order information induces much more computational overhead. Particularly, this overhead is of vital importance for the online scenario. Thus, it is crucial to choose an appropriate set of orders to achieve a better trade-off between detection effectiveness and computational complexity. Specifically, we construct the detector with different values of P and evaluate them in an online setting. Figure 8 illustrates that the testing time for an input sample increases from $8.9{\times}10^{-6}$s to $78.5{\times}10^{-6}$s when the order bound increases from 1 to 9. Additionally, we find that the detection capability (false positive rate) of our method is stabilized when $P \geq 4$. We note that the OOD detector [71] needs much higher-order Gram matrices (*i.e.*, $P = 10$) to discriminate between in-distribution and out-of-distribution datasets, especially when the two datasets are similar-looking. This experimental result shows that, considering the computational efficiency, it is sufficient to employ up to the third or the fourth order information to capture discriminative characteristics of benign and malicious inputs. Therefore, for the remaining experiments, we set P as 4 (see efficiency comparison in Section V-B).

Defending against all-to-all attack. Here, we consider another type of adversary who launches all-to-all attacks [59]. Specifically, for a c-way classifier, the trojaned samples originally in the i-th class are misclassified into the ((i+1) mod c)-th class. Since samples from all classes are infected by the all-to-all attack, the anomaly index values R_t^* are no longer effective. However, as shown in Figure 10, most of the RMMD statistics R_t of different infected labels in the all-to-all infected models are much larger than those of uninfected labels in the single-target attack. This demonstrates that Beatrix can still effectively defend against all-to-all attack relying on the RMMD statistics. In addition, we note that the all-to-all attack is the worst case of multi-target attacks that all labels are infected. However, the more labels that are infected, the less stealthy and lower performance the backdoor attack shows, especially for large datasets. As discussed in the recent research [84], when half of the labels are infected on the 1000-class ImageNet model, its clean accuracy and attack success rate drops 5% and 41%, respectively. Additionally, we consider the all-to-all attack with dynamic backdoors, which is much stronger than the all-to-all attack with universal backdoors [92].

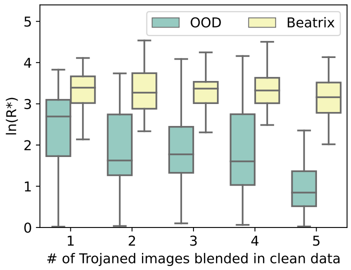


Fig. 9: Robustness comparison between Beatrix and the OOD detection method [71].

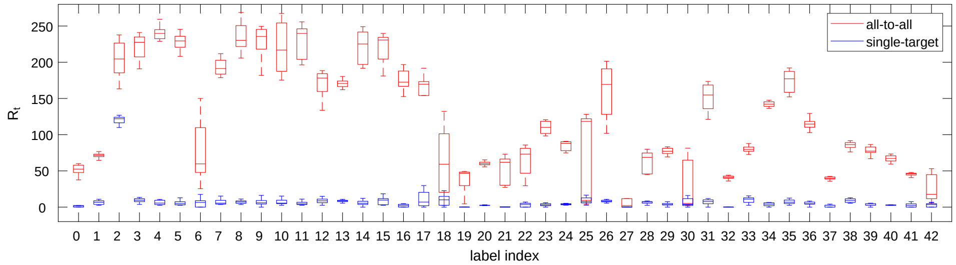


Fig. 10: RMMD statistics R_t of different labels in all-to-all infected models (red) and in single-target (label 2 is target) infected models (blue). Although the anomaly index R_t^* may not be effective for the all-to-all attack, their RMMD statistics R_t are much larger than those of uninfected labels in the single-target attack.

TABLE V: Defense performance against dynamic backdoors on GTSRB and CIFAR10.

Scenario	**Method**	**GTSRB**			**CIFAR10**		
		REC(%)	**PRE**(%)	**F1** (%)	**REC**(%)	**PRE**(%)	**F1**(%)
Offline	NC	18.6	9.3	12.4	40.0	25.0	30.8
	ABS	27.0	41.5	32.7	37.0	49.3	42.3
	MNTD	55.8	91.8	69.4	57.4	63.8	60.4
	AC	74.4	6.0	11.1	80.0	28.26	42.1
	SCAn	44.2	31.7	36.9	40.0	57.1	47.1
	Beatrix	**95.3**	**87.2**	**91.1**	**100.0**	**83.3**	**90.9**
Online	STRIP	23.0	41.9	29.7	20.9	43.9	28.3
	SentiNet	0.00	0.00	0.00	0.00	0.00	0.00
	SCAn	83.3	68.7	75.3	28.6	39.1	33.0
	Beatrix	**99.8**	**99.8**	**99.8**	**99.0**	**95.4**	**97.2**

TABLE VI: Defense performance against universal backdoors on GTSRB and CIFAR10.

Scenario	**Method**	**GTSRB**			**CIFAR10**		
		REC(%)	**PRE**(%)	**F1**(%)	**REC**(%)	**PRE**(%)	**F1**(%)
Offline	NC	97.1	61.6	75.4	92.5	51.4	66.7
	ABS	95.3	81.2	87.7	90.0	48.6	63.2
	MNTD	90.8	81.5	85.8	77.2	77.4	77.3
	AC	96.5	88.8	92.5	87.5	70.6	79.1
	SCAn	95.9	**96.5**	96.2	92.5	90.2	91.4
	Beatrix	**97.7**	96.0	**96.8**	**95.0**	**90.5**	**92.7**
Online	STRIP	86.7	97.9	91.9	87.8	96.1	91.7
	SentiNet	91.5	96.2	93.8	90.3	96.9	93.5
	SCAn	88.0	95.1	91.4	91.0	96.2	93.5
	Beatrix	**99.8**	**96.4**	**98.1**	**97.2**	**97.1**	**97.2**

B. Comparison

In this subsection, we compare our method with several state-of-the-art methods under two scenarios that are used in [84]. In the offline protection setting, a dataset containing benign and malicious samples are processed at once, and the defender is supposed to determine whether a data class is benign or infected. On the other hand, in the online setting, samples are processed one by one, and the defender is supposed to determine whether a sample is legitimate or malicious.

In the offline setting, we launch the dynamic backdoor attack [59] to generate 43 infected models with respect to the 43 classes in GTSRB and 10 infected models for CIFAR10. Additionally, we launch the conventional backdoor attack [32] to generate 172 (43×4) infected models for GTSRB and 40 (10×4) infected models for CIFAR10 with four different static triggers (Figure 19) which are also used in [18], [27], [84], [89]. For the online setting, we randomly select 4,000 samples from each dataset as testing samples, half of which carry dynamic (or static) triggers.

1) Offline defense: In the offline setting, we consider four competitive methods, namely Neural Cleanse (NC) [89], ABS [52], MNTD [92], Activation Clustering (AC) [14], and SCAn [84]. We re-implemented AC according to the paper [14] since the source code is not publicly available. We also re-implemented SCAn using TensorFlow Probability (TFP) based on the original MATLAB version [5]. Moreover, we used the PyTorch version of NC [3], ABS [1] and MNTD [4]. The comparison results are presented in Tables V and VI. The results indicate that our proposed method largely outperforms all four benchmark methods against dynamic backdoors while slightly outperforming them against universal backdoors.

NC leverages an optimization-based reverse engineering approach to find a trigger pattern that causes any benign input from other classes to be misclassified into a target label. However, in dynamic backdoor attacks, triggers are unique and non-reusable instead of being static and universal. As a result, the reverse-engineered triggers obtained by NC are visually and functionally different from the actual dynamic triggers. As for the 43 trojaned models infected by the dynamic backdoor on GTSRB, there are $43 \times 1 = 43$ poisoned classes (positives) and $43 \times 42 = 1806$ benign classes (negatives). The experimental results show that NC is not effective against dynamic backdoor attacks, achieving 18.6% (8/43) recall, and 9.3% (8/86) precision and 12.4% F1-score in GSTRB.

AC utilizes a two-class clustering method to separate the benign and malicious samples based on their feature vectors (activations). Specifically, AC performs dimensionality reduction using Independent Component Analysis (ICA), and then clusters them using 2-means. A high silhouette score [68] of the clustering results indicates the class is infected because the two clusters obtained by 2-means do fit the data well. However, under the dynamic backdoor attack, the feature presentations become less distinguishable so that AC cannot effectively separate the activations of benign and trigger-carrying samples. Therefore, AC achieves 92.5% F1-score against universal backdoor attacks whereas it yields only 11.1% F1-score against dynamic backdoor attacks on GTSRB.

SCAn models the representation distribution by a Gaussian distribution so that it uses a set of clean samples to estimate the covariance matrix. Then SCAn leverages Linear Discriminant Analysis (LDA) to separate the feature representations into two subgroups. A high statistic value from the likelihood-ratio test indicates the class is infected. As we discussed in Section II-C, the mean discrepancy is ineffective and the universal covariance assumption is not true in the dynamic backdoor attack. As a result, SCAn is ineffective against the dynamic backdoor attack because of its intrinsic limitations. Its F1-score against universal attacks on GSTRB is 96.2%, but it drops to 36.9% when it encounters dynamic attacks. Figure 11 provides a more in-depth analysis from the representation space perspective on CIFAR10. It shows that SCAn cannot raise the alarm when the feature representations of clean and poisoned samples are deeply fused in the dynamic backdoor. In contrast, Beatrix remains effective against the dynamic backdoor since Gram matrix is capable of capturing the subtle differences between clean and poisoned representations, as shown in Figure 11(c).

ABS can only determine whether a model is infected or not. Therefore, we trained 100 clean models and 100 infected models with random initialization for the evaluations in universal and dynamic attacks. ABS assumes that each target label is associated with only one trigger and the trigger subverts all benign samples to the target label. This assumption is broken by the dynamic backdoor attack, in which each trojaned sample has its own unique trigger. Additionally, based upon the universal trigger assumption, ABS assumes there is only one compromised neuron activated at a time by the trigger. Put differently, the changes in the activation of the intermediate layer only depend on this single neuron when encountering a trigger-carrying sample. However, due to the uniqueness of dynamic triggers, the abnormal changes in the activation pattern are dispersed across multiple neurons. Therefore, their reverse engineered trigger cannot reflect the malfunction in the model infected by dynamic backdoor attacks, and consequently, it yields only 32.7% and 42.3% F1-score on GSTRB and CIFAR10 against dynamic backdoor attacks.

MNTD, similar to ABS, can only flag a model as either trojaned or benign. Therefore, we evaluate MNTD with 100 clean/infected models used in the evaluation of ABS. MNTD uses jumbo learning to generate thousands of shadow models and then train a meta-classifier to learn the output differences of trojaned between clean models. MNTD fails to detect infected models with dynamic backdoor as the representations are already deep fused in the middle activation layers (see Figure 11(b)). Therefore, MNTD achieves 85.8% F1-score against universal backdoor attacks on GTSRB whereas it drops to 69.4% when it encounters dynamic attacks.

2) Online defense: In the online setting, we consider three existing defenses, STRIP [27], SentiNet [18], and SCAn [84]. We re-implemented SentiNet according to the paper, and used the Pytorch version of STRIP [3]. SCAn is configured for online defense following the paper. The experimental results are shown in Tables V and VI.

STRIP works by superimposing a set of randomly selected clean images to an input image and measuring the entropy of the prediction outputs. In the conventional backdoor attack, a trojaned image with a static trigger is resistant to this perturbation, leading to a much lower entropy of the outputs compared to that of a benign input. The effectiveness of STRIP relies on the dominant impact of the trigger [84]. However, this assumption no longer holds in the dynamic backdoor attack where images with mismatching triggers will deactivate the backdoors [48], [59], and subsequently the F1-score of STRIP drops from 91.9% and 91.7% against universal backdoors to only 29.7% and 28.3% against dynamic backdoors.

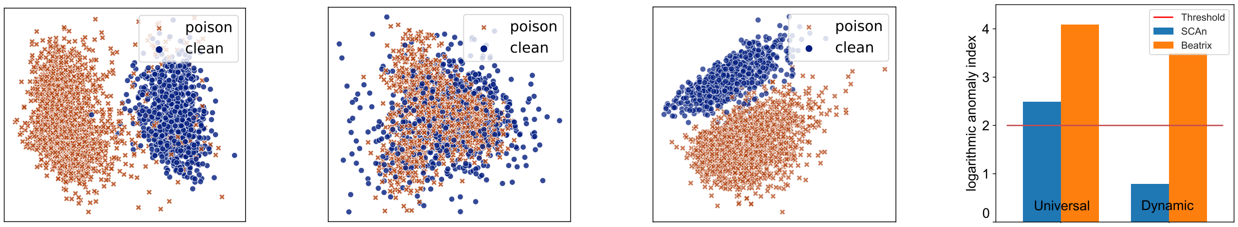
SentiNet utilizes the model interpretability technique [73] to locate highly salient contiguous region (*i.e.*, a potential trigger-region) of a given input. The extracted salient region is then overlaid on a set of clean images whose classification results are used to distinguish normal images from malicious images, since the trigger region is much more likely than normal region to subvert the clean images to the target label. What has been assumed here is that the attack is *localized* (*i.e.*, the trigger-region is constrained to a small contiguous part) and *universal* (*i.e.*, the attack is sample-agnostic). Both assumptions are broken by the dynamic backdoor attack where triggers are sample-specific and are distributed over different and disjointed portions of an image. Thus, the recall, precision and F1-score of SentiNet are 0%, indicating that it is no longer effective against this advanced attack.

TABLE VII: Efficiency comparison.

Method	STRIP	SentiNet	SCAn	Beatrix
Time (s)	0.04	0.11	13.58	35.14×10^{-6}

SCAn builds a composition model (covariance estimation) as well as an untangling model (mean estimation for each class) on a set of clean samples in an offline manner. Therefore, for each input sample, SCAn needs to update the untangling model for the image class. The incoming sample is tagged as malicious if it results in a class anomaly index larger than the threshold (e^2) and falls into the smaller subgroup. As we discussed above, SCAn models the feature representations under the LDA assumption (see Section II-C), which is violated in the dynamic backdoor attack. Consequently, the error in the composition model leads to the ineffectiveness of the untangling model in distinguishing representations of benign inputs from those of malicious inputs. As a result, the F1-score of SCAn drops from 93.5% (in universal attacks) to 33.5% (in dynamic attacks). Additionally, due to *its dependency on the accumulation of adversarial inputs* [84], SCAn shows a suboptimal performance against universal attacks in the online setting while it achieves similar performance in the offline setting compared to Beatrix.

Efficiency comparison. As the overhead is important for the online scenario, we also compare the efficiency of Beatrix with those of other online defenses on the GTSRB dataset. The experiment is conducted on one NVIDIA GeForce RTX 3090 GPU. The running time is averaged over 1,000 testing samples. As shown in Table VII, our approach is much faster than the three baseline methods. SCAn consumes the longest time (13.58s) compared to other three methods since it needs to update the untangling model when there is an incoming sample. SentiNet pastes the potential trigger region of the input image on clean and noise images while STRIP directly superimposes each input on clean images. Thus, SentiNet (0.11s) is slightly slower than STRIP (0.04s). The main computation overhead of Beatrix is building the deviation measurement model with clean samples, but this measurement model can be obtained through an offline training. Therefore, Beatrix only needs a forward pass to get the intermediate presentation for each sample and computes Gramian features of different orders, which can be obtained simultaneously when the user (defender) trains or tests her/his model with a dataset. Therefore, Beatrix takes only 35.14 $\times10^{-6}$s (for $P = 4$). Furthermore, as demonstrated in Section V-A, the defender can choose a different number of orders to trade off efficiency and effectiveness.



(a) Universal backdoor (b) Dynamic backdoor (c) Gramian of dynamic backdoor (d)

Fig. 11: A case study of SCAn and Beatrix on CIFAR10 with 0-th label being the trojan target. (a) and (b) illustrate the target class' representations projected onto their first two principle components under the universal and dynamic backdoor attack, respectively. (c) shows the projection of the Gramian features s of representations in (b). (d) shows the logarithmic anomaly index returned by SCAn and Beatrix.

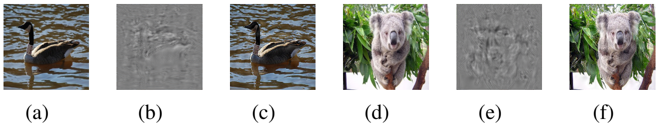


Fig. 12: Examples of poisoned samples under ISSBA. (a) and (d) are clean images, (b) and (e) are sample-specific triggers, and (c) and (f) are poisoned images.

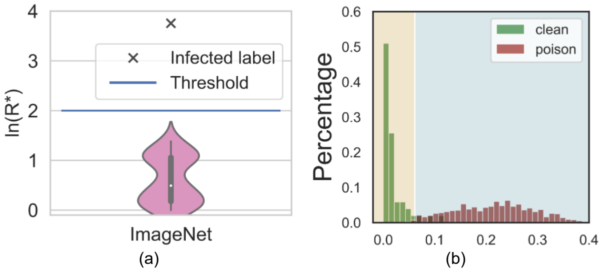


Fig. 13: (a) The logarithmic anomaly index of infected and uninfected labels under ISSBA. (b) Deviation distribution of benign and trojaned samples in the infected class under ISSBA.

C. Robustness Against Other Attacks

Invisible sample-specific backdoor attack. Motivated by the advances in DNN-based image steganography [11], [83], Li *et al.* proposed an invisible sample-specific backdoor attack (ISSBA), where triggers are generated by a pre-trained encoder network [48]. The generated triggers are invisible additive noise containing the information of a representative string of the target label. The attacker can flexibly design the string as the name and the index of the target class or even a random character. The encoder network embeds the string into a clean image to obtain a poisoned image. Thus, the poisoned image generator (encoder) is conditioned on the input images, indicating the backdoor triggers vary from input to input. A DNN classifier trained on the poisoned images will misclassify images trojaned by the same encoder into the classes indicated by the embedded strings. Some examples of the poisoned ImageNet samples and their corresponding triggers are shown in Figure 12.

We evaluate Beatrix on this attack using the infected model and the dataset shared by the authors [48]. Since they only release their implementation on the ImageNet dataset, we use the released infected model trained on a ImageNet subset which contains 200 classes with the 0-th label (goldfish) being the trojan target. As shown in Figure 13, the anomaly index of the infected label (label 0) is much larger than the uninfected labels (labels 1-199), indicating that Beatrix can effectively defend against this attack. More detailed comparison with other detection methods can be found in Appendix D.

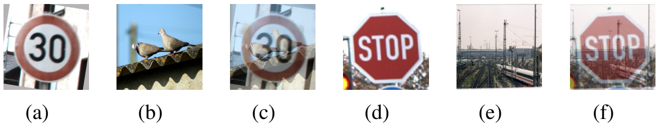


Fig. 14: Examples of poisoned samples under *Refool*. (a) and (d) are clean images, (b) and (e) are reflection patterns, and (c) and (f) are poisoned images.

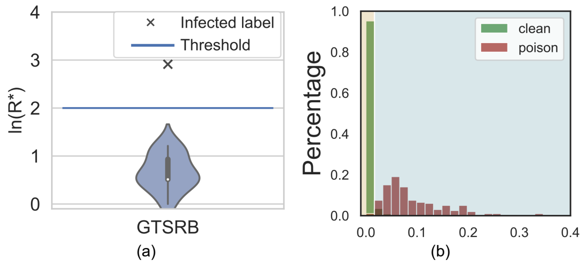


Fig. 15: (a) The logarithmic anomaly index of infected and uninfected labels under *Refool*. (b) Deviation distribution of benign and trojaned samples in the infected class under *Refool*.

Reflection backdoor attack. Liu *et al.* proposed *reflection backdoor* (*Refool*) using a special backdoor pattern based on a nature phenomenon — reflection [55]. Reflection occurs wherever there are glasses or smooth surfaces. *Refool* generates poisoned images by adding reflections to clean images based on mathematical models of physical reflection scenarios. Different from conventional backdoor attacks that rely on a fixed trigger pattern, *Refool* can utilize various reflections as the trigger pattern, making it stealthier than other attacks. Additionally, *Refool* blends the clean image with a triggering reflection pattern, so that the trigger is complex and spans all over the image. Some examples of clean images and their poisoned counterparts with reflection triggers are illustrated in Figure 14.

We evaluate Beatrix against this attack using the code and datasets shared by the authors [55]. The released datasets include three traffic sign datasets: GTSRB, BelgiumTSC [85] and CTSRD [37]. In this experiment, we use the GTSRB dataset and randomly choose the reflection images from PascalVOC [24] following the original implementation. The target class is the speed limit sign of 30 km/h. Our detection results are demonstrated in Figure 15. It shows that the anomaly

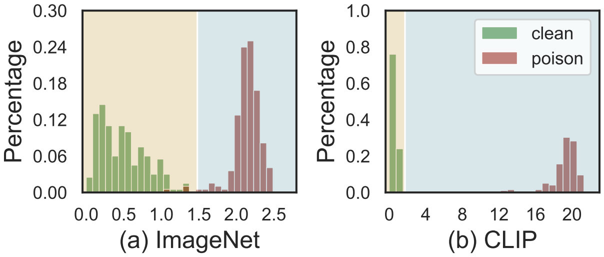


Fig. 16: Deviation distribution of benign and trojaned samples in the infected class of (a) Imagnet encoder and (b) CLIP encoder under BadEncoder attack.

index of the infected label is larger than the threshold, and those of the uninfected labels are all below the threshold. For online detection, Beatrix can effectively distinguish clean images from trigger-carrying ones with 99.99% TPR @ 5% FPR and 98.50% TPR @ 1% FPR, as shown in Figure 15(b).

BadEncoder attack. Another recently introduced backdoor attack is BadEncoder [38], which has been proposed for self-supervised learning pipelines. Self-supervised learning aims to pre-train an image encoder using a large amount of unlabeled data. Thus, the pre-trained encoder can be used as a feature extractor to build downstream classifiers in many different tasks. BadEncoder aims to compromise the self-supervised learning pipeline by injecting backdoors into a pre-trained image encoder such that the downstream classifier built upon this trojaned encoder will inherit the backdoor behavior. To craft a trojaned image encoder, BadEncoder fine-tunes a clean image encoder with two additional loss terms named effectiveness loss and utility loss. The effectiveness loss measures the similarity between feature vectors of reference inputs (*i.e., clean inputs in the target class*) and those of trigger-carrying inputs produced by the trojaned encoder. To maintain utility as well as stealthiness, BadEncoder applies the utility loss to encourage the trojaned encoder and the clean encoder produces similar outputs given the same clean inputs. In this way, a downstream classifier built upon the trojaned encoder will behave normally on clean inputs but misclassify trigger-carrying inputs into the target class since their feature representations are similar to those of clean target inputs.

We put Beatrix into test against BadEncoder using the infected models and datasets published along with the paper that introduced the attack itself [38]. In the experiments, we consider two real-world image encoders: ImageNet encoder originally released by Google [15] and CLIP encoder by OpenAI [65]. The target downstream dataset is GTSRB, and the target class is the 12th-label (*i.e.*, the priority road sign). Instead of building a downstream classifier, we directly evaluate Beatrix on distinguishing clean and poison inputs based on their feature vectors produced by the backdoor encoder. As shown in Figure 16, Beatrix can effectively defend against BadEncoder. Specifically, Beatrix achieves 99.8% TPR @ 1% FPR on the infected ImageNet encoder, and 100% TPR @ 1% FPR on the infected CLIP encoder.

D. Beyond the Image Domain

Akin to previous works, we mainly focus on the image classification tasks. There are backdoor attacks in other domains, such as natural language processing (NLP) [9], [16], [44], acoustics signal processing [94] and malware detection [43], [75]. Here, we extend our approach to mitigate the threats posed by backdoor attacks in speech recognition and text classification domains.

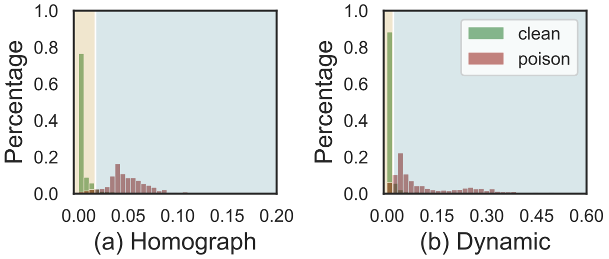


Fig. 17: Deviation distribution of benign and trojaned samples in the infected class under (a) Homograph Backdoor Attack and (b) Dynamic Sentence Backdoor Attack.

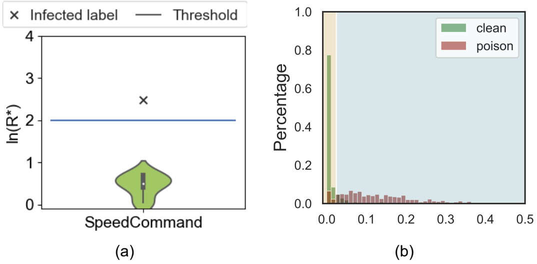


Fig. 18: (a) The logarithmic anomaly index of infected and uninfected labels of a speech recognition backdoor model. (b) Deviation distribution of benign and trojaned samples in the infected class under the speech recognition backdoor attack.

For the NLP task, we evaluate Beatrix on the homograph backdoor attack and dynamic sentence backdoor attack proposed by Li *et al.* [44]. The homograph backdoor attack inserts triggers by replacing several characters of the clean sequences with their homograph equivalent. Given an original sentence, the dynamic sentence backdoor attack uses pre-trained language models to generate a suffix sentence to act as the trigger. We use the code and dataset shared by the authors to train poisoned BERT model on the toxic comment classification dataset [39]. To balance the number of positive (*i.e.*, toxic) and negative (*i.e.*, non-toxic) samples, it draws 16225 negative samples from the negative texts so the final dataset consists of 32450 samples. The dataset is then split to give 29205 (90% of the dataset) in the training set and 3245 (10%) in the test set. Since this is a binary classification task, we directly evaluate the online defense performance of Beatrix. As shown in Figure 17, Beatrix achieves 89.8% TPR @ 5% FPR on homograph attack and 75.2% TPR @ 5% FPR on dynamic sentence attack.

For the speech task, we use the speech recognition backdoor implementation provided in the work [92] on the SpeechCommand dataset [90]. The original dataset consists of 65,000 one-second audio files of 35 classes. Following the previous work [92], we use the files of 10 classes (*i.e.*, “yes”, “no”, “up”, “down”, “left”, “right”, “on”, “off”, “stop”, “go”), which gives 30,769 training samples and 4,074 testing samples. It first extracts the mel-spectrogram of each file and then trains an LSTM model over the mel-spectrograms. The backdoor trigger is a consecutive noise signal whose length is 0.05 seconds. Our detection results shown in Figure 18 demonstrate that Beatrix

TABLE VIII: Adaptive attack.

	λ	0.05	0.1	0.5	1	5
GTSRB	CA	98.5%	98.1%	93.8%	79.7%	4.7%
	ASR	97.5%	96.8%	95.2%	89.1%	-
	5% FPR	99.8%	98.6%	97.4%	92.4%	-
	1% FPR	98.8%	96.6%	93.6%	81.0%	-
CIFAR10	CA	92.7%	91.8%	89.8%	69.5%	10.0%
	ASR	98.1%	96.2%	94.6%	85.1%	-
	5% FPR	97.6%	94.3%	83.3%	40.5%	-
	1% FPR	87.2%	76.6%	51.7%	14.0%	-

can detect the infected label effectively and distinguish clean signal from trigger-carrying ones with 77.5% TPR @ 5% FPR.

E. Adaptive Attack

We study an adaptive adversary who targets at the deviation measurement of Beatrix. We consider a strong white-box adversary who controls the training process of the victim model. The objective of the adversary is to force the activation patterns of poisoned images to resemble those of clean images, so that Beatrix cannot separate the benign and malicious inputs. To achieve this goal, we design an adaptive loss to minimize the distance between poisoned and clean images of a target class, in the representation space based on multiple high-order Gram matrices. In the experiment, we set the order upper bound P as 4 in the adaptive loss. Therefore, the loss function of the adaptive attack can be denoted as follows:

$$L = L_o + \lambda L_a,$$
$$L_a = \mathbb{E}_{x \in X_{/y_t}, x_t \in X_{y_t}} \left[\sum_{p=1}^{P} \| G^p(\mathcal{B}(x, g(x))) - G^p(x_t) \|^2 \right],$$

where L_o is the original dynamic backdoor loss for the victim model f [59], and L_a denotes the adaptive loss. G^p denotes the p-th order Gram matrix of the internal activation from an input image. X_{y_t} and $X_{/y_t}$ denotes target and non-target training data, respectively. λ is a hyperparameter balancing the model performance and the adaptive strength.

We perform this attack on the GTSRB and CIFAR10 datasets. We evaluate the online performance of Beatrix to determine whether an input image is benign or malicious during the inference phase. In the experiment, we randomly select 500 clean images from the target class and 500 poisoned images from other classes. The results are shown in Table VIII. We can find that the true positive rate of Beatrix slightly decreases when the hyper-parameter λ increases from 0.05 to 0.5. In particular, when λ increases to 1, Beatrix yields only 14.0% TPR @ 1% FPR on CIFAR10 dataset, indicating that Beatrix is no longer effective. However, the model performance (*i.e.*, clean accuracy on benign inputs and attack success rate on malicious ones) also decreases significantly.

VI. Related Work

Backdoor attacks. With the increasing use of pre-trained DNNs in security-sensitive domains, backdoor attacks have recently been recognized as a new threat. Current attacks inject backdoors to DNNs by either poisoning the training dataset or directly manipulating the model parameters. BadNets was first introduced as a poisoning-based backdoor attack in which the adversary has full control over the model and training data [32]. Subsequently, a series of attempts were made to launch attacks with fewer poisoned samples and limited access to training data [17], [53]. On the other end, to make the poisoning data stealthier, there are attacks that inject poisoned samples without altering their labels [76], [87]. Recently, backdoors activated by semantic triggers with diverse patterns were studied to further conceal the attacks [12], [45], [55], [69]. The threat of neural backdoors in emerging machine learning paradigms has also been investigated [38], [76], [91], [93]. Interestingly, model replacement has been identified as an attack surface in federated learning [10]. Although various backdoor attacks are proposed, most of them employ static triggers which can be easily detected by existing defensive techniques. On the contrary, recently proposed dynamic backdoor attacks craft input-aware triggers, which hardens the detection of such backdoors [48], [59], [70].

Defenses against backdoor attacks. Current defensive techniques can be broadly categorized into two branches. The first line of works conducts a model analysis to identify backdoors. Otherwise, defenders can perform run-time detection against triggered backdoors. A common wisdom in model analysis is to discover anomalies in the learned representations [14], [86]. Moreover, other methods are reverse-engineering possible triggers and unlearning inserted backdoors [52], [54], [89]. However, these methods are reported as ineffective against large triggers and source-label-specific attacks [52], [84]. There is also a method that adopts a meta classifier trained on a set of clean and trojaned models to identify compromised models [92]. A defender can also implement run-time detection methods when the DNN has been deployed. STRIP was proposed to detect backdoors controlled by dominant triggers [27]. SentiNet adopts Grad-Cam to detect potential backdoor locations and malicious inputs [18], [73]. A similar idea is also explored and extended in Februus [22]. However, the aforementioned methods are vulnerable to adaptive attacks such as the source-label-specific backdoor. Lately, SCAn proposed a statistical method to defend against this attack [84]; however, it is less desirable against attacks with dynamic triggers.

VII. Conclusion

In this paper, we demonstrated that the existing defensive techniques heavily rely on the premise of the universal backdoor trigger, in which poisoned samples share the same trigger and thus show the same abnormal behavior. Once this prerequisite is violated, they can no longer effectively detect the advanced backdoor attacks like dynamic backdoors, where the adversary injects sample-specific triggers to each input. Based on the observation that Gramian information of dynamically trojaned data points is highly distinct from that of the benign ones, we developed Beatrix to capture not only the feature correlations but also the appropriately high-order information of the representations of benign and malicious samples, and utilized the Kernel-based two-sample testing to identify the infect labels. Experimental results show the effectiveness and robustness of our proposed approach. Beatrix can successfully defend against backdoor attacks for not only the conventional ones but also the advanced attacks, such as dynamic backdoors which can defeat the aforementioned defensive techniques.

References

[1] *ABS*, https://github.com/naiyeleo/ABS.git.

[2] *Composite Backdoor Attack*, https://github.com/TemporaryAcc0unt/composite-attack.

[3] *Input-Aware Dynamic Backdoor Attack*, https://github.com/VinAIResearch/input-aware-backdoor-attack-release.

[4] *MNTD*, https://github.com/AI-secure/Meta-Nerual-Trojan-Detection.

[5] *SCAn*, https://github.com/TDteach/backdoor.git.

[6] *WANET*, https://github.com/VinAIResearch/Warping-based_Backdoor_Attack-release.

[7] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.

[8] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *ICLR*, 2018.

[9] A. Azizi, I. A. Tahmid, A. Waheed, N. Mangaokar, J. Pu, M. Javed, C. K. Reddy, and B. Viswanath, "T-miner: A generative approach to defend against trojan attacks on dnn-based text classification," in *USENIX Security*, 2021.

[10] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *AISTATS*, 2020.

[11] S. Baluja, "Hiding images in plain sight: Deep steganography," in *NeurIPS*, 2017.

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

[13] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[14] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," in *SafeAI@AAAI*, 2019.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[16] X. Chen, A. Salem, D. Chen, M. Backes, S. Ma, Q. Shen, Z. Wu, and Y. Zhang, "Badnl: Backdoor attacks against nlp models with semantic-preserving improvements," in *ACSAC*, 2021.

[17] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[18] E. Chou, F. Tramer, and G. Pellegrino, "SentiNet: Detecting localized universal attacks against deep learning systems," in *IEEE Symposium on Security and Privacy Workshops (SPW)*. IEEE, 2020, pp. 48–54.

[19] K. P. Chwialkowski, A. Ramdas, D. Sejdinovic, and A. Gretton, "Fast two-sample testing with analytic representations of probability measures," in *NeurIPS*, 2015.

[20] S. Danafar, P. Rancoita, T. Glasmachers, K. Whittingstall, and J. Schmidhuber, "Testing hypotheses by regularized maximum mean discrepancy," *arXiv preprint arXiv:1305.0423*, 2013.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *CVPR*, 2009.

[22] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defense against trojan attacks on deep neural network systems," in *ACSAC*, 2020.

[23] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC Press, 1994.

[24] M. Everingham and J. Winn, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) development kit," *Pattern Analysis, Statistical Modelling and Computational Learning*, vol. 8, p. 5, 2011.

[25] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *CVPR*, 2018.

[26] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal, and H. Kim, "Backdoor attacks and countermeasures on deep learning: A comprehensive review," *arXiv preprint arXiv:2007.10760*, 2020.

[27] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," in *ACSAC*, 2019.

[28] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *CVPR*, 2016.

[29] X. Gong, Y. Chen, Q. Wang, H. Huang, L. Meng, C. Shen, and Q. Zhang, "Defense-resistant backdoor attacks against deep neural networks in outsourced cloud environment," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2617–2631, 2021.

[30] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *ICLR*, 2015.

[31] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[32] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[33] J. Hayase, W. Kong, R. Somani, and S. Oh, "SPECTRE: Defending against backdoor attacks using robust statistics," *arXiv preprint arXiv:2104.11315*, 2021.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[35] ——, "Identity mappings in deep residual networks," in *ECCV*, 2016.

[36] W. Hoeffding, "A class of statistics with asymptotically normal distributions," *Annals of Mathematical Statistics*, vol. 19, no. 3, pp. 293–325, 1948.

[37] L. Huang, *Chinese Traffic Sign Database*, Beijing Jiaotong University, http://www.nlpr.ia.ac.cn/pal/trafficdata/recognition.html.

[38] J. Jia, Y. Liu, and N. Z. Gong, "BadEncoder: Backdoor attacks to pre-trained encoders in self-supervised learning," in *IEEE S&P*, 2022.

[39] Kaggle, *Toxic Comment Classification Challenge Database*, https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/.

[40] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Technical Report, Citeseer*, 2009.

[41] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *NeurIPS*, 2018.

[42] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of experimental social psychology*, vol. 49, no. 4, pp. 764–766, 2013.

[43] C. Li, X. Chen, D. Wang, S. Wen, M. E. Ahmed, S. Camtepe, and Y. Xiang, "Backdoor attack on machine learning based android malware detectors," *IEEE TDSC*, 2021.

[44] S. Li, H. Liu, T. Dong, B. Z. H. Zhao, M. Xue, H. Zhu, and J. Lu, "Hidden backdoors in human-centric language models," in *CCS*, 2021.

[45] S. Li, M. Xue, B. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE TDSC*, 2021.

[46] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," in *IJCAI*, 2017.

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

[48] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *ICCV*, 2021.

[49] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *ICML*, 2015.

[50] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *ICLR*, 2018.

[51] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Composite backdoor attack for deep neural network by mixing existing benign features," in *CCS*, 2020.

[52] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "ABS: Scanning neural networks for back-doors by artificial brain stimulation," in *CCS*, 2019.

[53] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *NDSS*, 2018.

[54] Y. Liu, G. Shen, G. Tao, Z. Wang, S. Ma, and X. Zhang, "EX-RAY: Distinguishing injected backdoor from natural features in neural networks by examining differential feature symmetry," *arXiv preprint arXiv:2103.08820*, 2021.

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

[56] Y. Liu, A. Mondal, A. Chakraborty, M. Zuzak, N. Jacobsen, D. Xing, and A. Srivastava, "A survey on neural trojans," in *ISQED*, 2020.

[57] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Proceedings of the IEEE Signal Processing Society Workshop*. IEEE, 1999, pp. 41–48.

[58] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf, "Kernel mean embedding of distributions: A review and beyond," *arXiv preprint arXiv:1605.09522*, 2016.

[59] A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," in *NeurIPS*, 2020.

[60] ——, "Wanet–imperceptible warping-based backdoor attack," in *ICLR*, 2021.

[61] R. Pang, H. Shen, X. Zhang, S. Ji, Y. Vorobeychik, X. Luo, A. Liu, and T. Wang, "A tale of evil twins: Adversarial inputs versus poisoned models," in *CCS*, 2020.

[62] R. Pang, Z. Zhang, X. Gao, Z. Xi, S. Ji, P. Cheng, and T. Wang, "TROJANZOO: Towards unified, holistic, and practical evaluation of neural backdoors," in *EuroS&P*, 2022.

[63] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC*, 2015.

[64] J. V. Psutka and J. Psutka, "Sample size for maximum likelihood estimates of gaussian model," in *CAIP*, 2015.

[65] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[66] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun *et al.*, "Scalable and accurate deep learning with electronic health records," *NPJ Digital Medicine*, vol. 1, no. 1, pp. 1–10, 2018.

[67] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced Lectures on Machine Learning: ML Summer Schools 2003*, 2003, pp. 63–71.

[68] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[69] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," *AAAI*, 2020.

[70] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," in *EuroS&P*, 2022.

[71] C. S. Sastry and S. Oore, "Detecting out-of-distribution examples with Gram matrices," in *ICML*, 2020.

[72] B. Schölkopf, A. J. Smola, F. Bach *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

[73] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017.

[74] R. J. Serfling, *Approximation theorems of mathematical statistics*. John Wiley & Sons, 2009, vol. 162.

[75] G. Severi, J. Meyer, S. Coull, and A. Oprea, "{Explanation-Guided} backdoor poisoning attacks against malware classifiers," in *USENIX Security*, 2021.

[76] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *NeurIPS*, 2018.

[77] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *CCS*, 2016.

[78] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE S&P*, 2017.

[79] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015.

[80] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "DARTS: Deceiving autonomous cars with toxic signs," *arXiv preprint arXiv:1802.06430*, 2018.

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

[82] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *ICLR*, 2014.

[83] M. Tancik, B. Mildenhall, and R. Ng, "StegaStamp: Invisible hyperlinks in physical photographs," in *CVPR*, 2020.

[84] D. Tang, X. Wang, H. Tang, and K. Zhang, "Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection," in *USENIX Security*, 2021.

[85] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3d localisation," *Machine vision and applications*, vol. 25, no. 3, pp. 633–647, 2014.

[86] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *NeurIPS*, 2018.

[87] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.

[88] G. Varoquaux *et al.*, "Comparing distributions: ℓ_1 geometry improves kernel two-sample testing," in *NeurIPS*, 2019.

[89] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural Cleanse: Identifying and mitigating backdoor attacks in neural networks," in *IEEE S&P*, 2019.

[90] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[91] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "DBA: Distributed backdoor attacks against federated learning," in *ICLR*, 2019.

[92] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI trojans using meta neural analysis," in *IEEE S&P*, 2021.

[93] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent backdoor attacks on deep neural networks," in *CCS*, 2019.

[94] T. Zhai, Y. Li, Z. Zhang, B. Wu, Y. Jiang, and S.-T. Xia, "Backdoor attack against speaker verification," in *ICASSP*, 2021.

[95] E. Zisselman and A. Tamar, "Deep residual flow for out of distribution detection," in *CVPR*, 2020.

Appendix

A. Detailed Discussion of Existing Backdoor Detection

Defending against universal backdoor. In universal backdoor attacks where different trojaned samples share a same trigger, the trojaned model is overfitted to the backdoor trigger [47]. Thus, the misclassification of a trojaned image is predominantly depended on the trigger regardless of the image content [48], [59], [84]. This rudimentary setting is the Achilles heel of the existing attacks such that the overfitted trigger pattern gives rise to a series of detection methods [14], [18], [27], [52], [89]. Based on the assumption that the backdoor trigger is sample-agnostic, existing defensive techniques can easily estimate the universal trigger [52], [89] or detect the trojaned samples according to their common abnormal behavior [14], [18], [27].

The success of all these defensive techniques relies on the premise of universal backdoors where the same trigger subvert all benign inputs. Once this assumption is violated, their effectiveness will be heavily vitiated.

Defending against partial backdoor. Firstly highlighted in [89], the partial backdoor was considered as a powerful and stealthy attack since the trigger only convert samples in a specific class but has no impact on those in other classes. Although several works [22], [52], [89] attempt to defend against the partial backdoor attacks, their main assumptions still focus on the universal backdoor attacks. Realizing the vulnerability of the existing backdoor defenses that focus on sample-agnostic (a.k.a. source-agnostic) backdoors, Tang *et al.* [84] studied the source-specific backdoor attack and showed that the trojaned and benign samples are clearly distinguishable in the feature space under the universal backdoor, but deeply tangling with each other under the partial backdoor. Therefore, They proposed Statistical Contamination Analysis (SCAn) to detect the source-specific backdoor by modeling the distributions of benign and malicious samples' representations.

B. Theoretical Analysis of SCAn

SCAn models the feature distribution by a Gaussian distribution under the Linear Discriminant Analysis (LDA) assumption, *i.e.*, different mean values but same covariance for the distributions of clean and trojaned feature representations. Formally, let the identity vectors (mean values) of clean and poisoned data be u_1 and u_2 respectively. And the covariance is denoted as S_ϵ. Then, the representation of a clean sample is $r_i^{clean} = u_1 + e_i$, where $e_i \sim \mathcal{N}(0, S_\epsilon)$ and r_i^{clean} follows a Gaussian distribution $\mathcal{N}(u_1, S_\epsilon)$. Similarly, the representation of a poisoned sample can be denoted as $r_j^{poison} = u_2 + e_j$, where $e_j \sim \mathcal{N}(0, S_\epsilon)$ and $r_i^{poison} \sim \mathcal{N}(u_2, S_\epsilon)$.

Let n_1 denote the number of clean samples and n_2 denotes the number of poison samples in the target (*i.e.*, infected) class. Then, the mean value of the representations of all samples (clean and poison samples) in this class is:

$$\begin{aligned}
u_0 &= \frac{1}{N}\left(\sum^{n_1} r_i^{clean} + \sum^{n_2} r_j^{poison}\right) \\
&= \frac{1}{N}\left(\sum^{n_1}(u_1+e_i) + \sum^{n_2}(u_2+e_j)\right) \\
&= \frac{1}{N}\left(\sum^{n_1}u_1 + \sum^{n_1}e_i + \sum^{n_2}u_2 + \sum^{n_2}e_j\right)
\end{aligned}$$

Recall that e_i and $e_j \sim \mathcal{N}(0, S_\epsilon)$,

$$u_0 = \frac{n_1}{N}u_1 + \frac{n_2}{N}u_2$$

SCAn formulates the task of backdoor detection as a likelihood-ratio test problem over the feature representations of all samples (*i.e.*, $R = R^{clean} \cup R^{poison}$) based on two hypotheses:

Null Hypothesis $\boldsymbol{H}_0$: R is drawn from a single normal distribution.

Alternative Hypothesis $\boldsymbol{H}_1$: R is drawn from a mixture of two normal distributions.

Therefore, the likelihood ratio under $\boldsymbol{H}_1$ hypothesis (*i.e.*, the class is infected) is defined as [84]:

$$\mathcal{L}=\sum_k^N \left[(r_k-u_0)S_\epsilon^{-1}(r_k-u_0)^T-(r_k-u_m)S_\epsilon^{-1}(r_k-u_m)^T\right]$$

where $m \in \{1, 2\}$ is the label of the representation r_k.

$$\begin{aligned}
\mathcal{L}=&\sum_i^{n1}\left[(r_i-u_0)S_\epsilon^{-1}(r_i-u_0)^T-(r_i-u_1)S_\epsilon^{-1}(r_i-u_1)^T\right] \\
&+\sum_j^{n2}\left[(r_j-u_0)S_\epsilon^{-1}(r_j-u_0)^T-(r_j-u_2)S_\epsilon^{-1}(r_j-u_2)^T\right]
\end{aligned} \tag{20}$$

When r_k is clean sample (the first term in (20)):

$$\begin{aligned}
&\sum_i^{n1}\left[(r_i-u_0)S_\epsilon^{-1}(r_i-u_0)^T-(r_i-u_1)S_\epsilon^{-1}(r_i-u_1)^T\right] \\
=&\sum_i^{n1}[(u_1+e_i-u_0)S_\epsilon^{-1}(u_1+e_i-u_0)^T \\
&-(u_1+e_i-u_1)S_\epsilon^{-1}(u_1+e_i-u_1)^T] \\
=&\sum_i^{n1}\left[(u_1+e_i-u_0)S_\epsilon^{-1}(u_1+e_i-u_0)^T-e_iS_\epsilon^{-1}e_i^T\right] \\
=&\sum_i^{n1}[(u_1+e_i-\frac{n_1}{N}u_1+\frac{n_2}{N}u_2)S_\epsilon^{-1}(u_1+e_i-\frac{n_1}{N}u_1+\frac{n_2}{N}u_2)^T \\
&-e_iS_\epsilon^{-1}e_i^T] \\
=&\sum_i^{n1}\left[(\frac{n_2}{N}(u_1-u_2)+e_i)S_\epsilon^{-1}(\frac{n_2}{N}(u_1-u_2)+e_i)^T-e_iS_\epsilon^{-1}e_i^T\right] \\
=&\sum_i^{n1}[(\frac{n_2}{N})^2(u_1-u_2)S_\epsilon^{-1}(u_1-u_2)^T \\
&+2\frac{n_2}{N}(u_1-u_2)S_\epsilon^{-1}e_j^T+e_iS_\epsilon^{-1}e_i^T-e_iS_\epsilon^{-1}e_i^T] \\
=&\sum_i^{n1}\left[(\frac{n_2}{N})^2(u_1-u_2)S_\epsilon^{-1}(u_1-u_2)^T+2\frac{n_2}{N}(u_1-u_2)S_\epsilon^{-1}e_j^T\right]
\end{aligned}$$

Recall that $e_i \sim \mathcal{N}(0, S_\epsilon)$,

$$=\sum_i^{n1}\left[(\frac{n_2}{N})^2(u_1-u_2)S_\epsilon^{-1}(u_1-u_2)^T\right]$$

Similarly, when r_k is poison sample (the second term in (20)):

$$\begin{aligned}
&\sum_j^{n2}\left[(r_j-u_0)S_\epsilon^{-1}(r_j-u_0)^T-(r_j-u_2)S_\epsilon^{-1}(r_j-u_j)^T\right] \\
=&\sum_j^{n2}\left[(\frac{n_1}{N})^2(u_1-u_2)S_\epsilon^{-1}(u_1-u_2)^T\right]
\end{aligned}$$

Therefore, the likelihood ratio of the infected class is

$$\begin{aligned}
\mathcal{L}=&\sum_i^{n1}\left[(\frac{n_2}{N})^2(u_1-u_2)S_\epsilon^{-1}(u_1-u_2)^T\right] \\
&+\sum_j^{n2}\left[(\frac{n_1}{N})^2(u_1-u_2)S_\epsilon^{-1}(u_1-u_2)^T\right] \\
=&\left[n_1(\frac{n_2}{N})^2+n_2(\frac{n_1}{N})^2\right](u_1-u_2)S_\epsilon^{-1}(u_1-u_2)^T \\
=&\left[\frac{n_1n_2(n_1+n_2)}{N^2}\right](u_1-u_2)S_\epsilon^{-1}(u_1-u_2)^T \\
=&\left(\frac{n_1n_2}{N}\right)(u_1-u_2)S_\epsilon^{-1}(u_1-u_2)^T
\end{aligned} \tag{21}$$

C. Experiment Setup

Datasets. To evaluate the the performance of our proposed method, we take four datasets which are commonly used in backdoor-related works [48], [62], [84], [89]:

- CIFAR10 [40]. It consists of 50,000 colored training images of size 32×32 and 10,000 testing images which are equally distributed on 10 classes;

- GTSRB [81]. It consists of 39,209 colored training images and 12,630 testing images of 43 different traffic signs. The image sizes range from 29 × 30 to 144 × 48. However, we resize them all to be 32×32.

- VGGFace [63]. In the original dataset, there are 2,622 identities and each identity has 1,000 face images in 224×224 pixels. However, about half of the links are no longer available. Following the previous work [48], we select the top 100 identities with the largest number of images. This way, we obtain 100 identities with 48,305 images. We randomly split them into training and testing samples with a ratio of 8:2.

- ImageNet [21]. This dataset is related to the object classification task. Similar to [48], [61], [62], we randomly select a subset containing 100 classes. Out of their samples, 50,000 224×224 sized images are picked for training (500 images per class) and 10,000 images are put aside for testing (100 images per class).

Models. To build the classifier on CIFAR-10 and GTSRB, we use Pre-activation ResNet18 [35], following the original dynamic backdoor [59]. Additionally, we use VGG16 [79] for VGGFace and ResNet101 [34] for ImageNet datasets, respectively. Their top-1 accuracy on the corresponding testset is summarized in Table III.

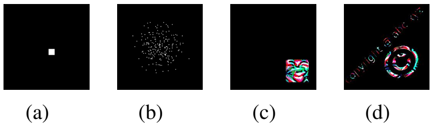


Fig. 19: Four static triggers used in our experiments.

D. Defense Performance Against ISSBA

In Section V-C, we demonstrate the robustness of Beatrix against the invisible sample-specific backdoor attack (ISSBA). In this section, we provide a comprehensive comparison with the state-of-the-art defense approaches on ISSBA. In the offline setting, we generate 20 infected models with respect to the target label from 0 to 19. For online setting, we randomly select 2000 samples from each dataset as testing samples, half of which carry dynamic (or static) triggers. Similar to results on the input-aware dynamic backdoor attack, the experimental results on ISSBA demonstrate that Beatrix can significantly outperform the existing defenses, as shown in Table. IX.

E. Ablation Study

We carried out a variety of ablation studies to demonstrate the effectiveness of using MAD and RMMD as the detection metrics in our approach. Herein, Beatrix-UG uses a univariate Gaussian model in the deviation measurement, and Beatrix-G assumes that the benign and malicious samples follow Gaussian distributions. In Beatrix-UG, we use the variance of Gramian features instead of MAD. That is

$$\bar{s}_j = mean(\{s_{ij},\ \forall i \in \{1, 2, ..., |\mathcal{X}_t|\}\}), \tag{22}$$

$$variance = mean(\{(s_{ij} - \bar{s}_j)^2,\ \forall i \in \{1, 2, ..., |\mathcal{X}_t|\}\}). \tag{23}$$

TABLE IX: Defense performance against ISSBA on ImageNet.

Scenario	Method	ImageNet		
		REC(%)	**PRE**(%)	**F1** (%)
Offline	NC	35.0	15.9	21.9
	ABS	40.0	47.1	43.2
	AC	90.0	35.3	50.7
	SCAn	40.0	72.7	51.6
	Beatrix	**85.0**	**100.0**	**91.9**
Online	STRIP	17.5	41.9	24.6
	SentiNet	0.00	0.00	0.00
	SCAn	45.8	79.2	58.0
	Beatrix	**98.5**	**99.0**	**98.7**

TABLE X: Ablation study. The defense performance against dynamic backdoors on GTSRB and CIFAR10.

Scenario	Method	GTSRB			CIFAR10		
		REC(%)	**PRE**(%)	**F1** (%)	**REC**(%)	**PRE**(%)	**F1**(%)
Offline	Beatrix-UG	**97.7**	77.8	86.6	50.0	83.3	62.5
	Beatrix-G	76.7	73.3	75.0	10.0	33.3	15.4
	Beatrix	95.3	**87.2**	**91.1**	**100.0**	**83.3**	**90.9**
Online	Beatrix-UG	99.5	93.9	96.6	95.0	79.0	86.0
	Beatrix	**99.8**	**99.8**	**99.8**	**99.0**	**95.4**	**97.2**

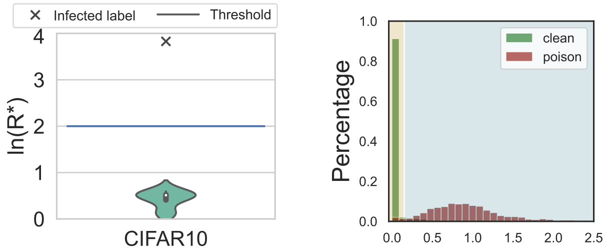
In Beatrix-G, we alter RMMD metric to mahalanobis distance metric which assumes the data distribution is characterized by a mean and the covariance matrix (*i.e.*, multivariate Gaussian distribution assumption in SCAn). Since this change only affects the identification of infected labels, we only evaluate its performance in the offline setting. As shown in Table X, for both online and offline cases, our approach with MAD and RMMD outperforms the baselines.

F. Defense Performance Against Composite Backdoor [51]

Instead of injecting new features that do not belong to any output label, Lin *et al.* proposed *composite attack* that uses composition of existing benign features/objects as the trigger. It leverages a mixer to generate poisonous samples. The poisoned model causes targeted misclassification when the trigger composition is present. We evaluate Beatrix against this attack on the object recognition task using the code shared by the authors [2]. Following the original implementation, we poison the model to predict *mixer(airplane, automobile)* to *bird* in CIFAR10 dataset. Figure 20 demonstrates the effectiveness of Beatrix. It shows that the anomaly index of the infected label (bird) is larger than the threshold, and those of the uninfected labels are all below the threshold. For online detection, Beatrix can effectively distinguish clean images from poisoned ones with 92.11% TPR @ 5% FPR and 96.88% AUC score.

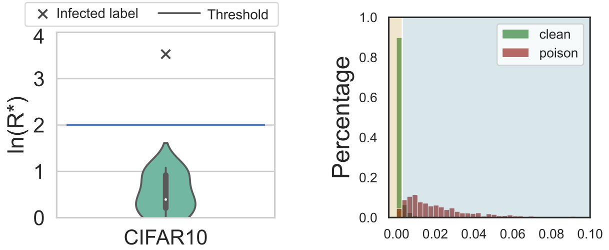
G. Defense Performance Against WaNet [60]

To improve the stealthiness of backdoor attacks, Nguyen *et al.* proposed WaNet based on image warping. WaNet adopts the same elastic transformation in generating backdoor images, making the modification unnoticeable for human eyes. We put Beatrix into test against WaNet on CIFAR10 using the published code [6]. As shown in Figure 21, Beatrix can effectively identify the infected label (label 0) of the poisoned



(a) (b)

Fig. 20: (a) The logarithmic anomaly index of infected and uninfected labels under the composite attack. (b) Deviation distribution of benign and trojaned samples in the infected class under the composite attack.



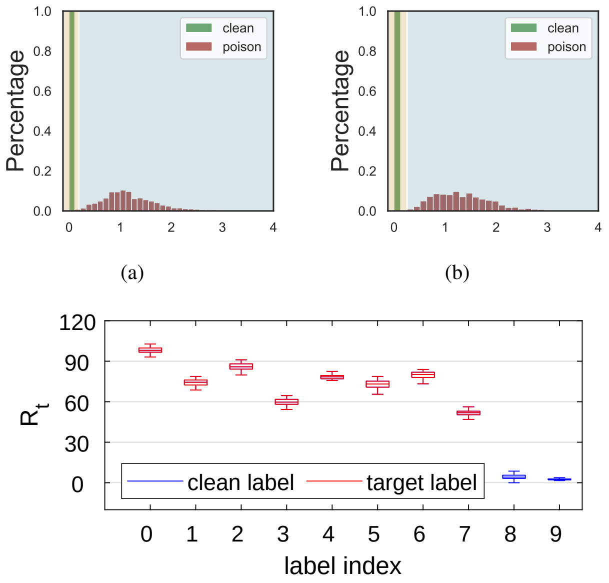
(a) (b)

Fig. 21: (a) The logarithmic anomaly index of infected and uninfected labels under WANET. (b) Deviation distribution of benign and trojaned samples in the infected class under WANET.

model, and achieve 89.90% TPR @ 5% FPR and 97.94% AUC score in online detection.

H. DEFENSE PERFORMANCE AGAINST MULTI-TRIGGER ATTACK [29]

In order to increase the triggers diversity to avoid being detected, Gong *et al.* proposed ROBNET using a multi-location patching method. And they extend the attack space by designing multi-trigger backdoor attacks that can produce different triggers targeting the same or different misclassification label(s). We re-implemented this attack according to the paper [29] since the source code is not publicly available. We evaluate Beatrix against ROBNET (*i.e.*, the multi-trigger same-label attack and multi-trigger multi-label attack) on the CIFAR10 dataset. For both types of attacks, we set the number of triggers as 8 which is the maximum number of triggers used in the original paper. Beatrix can effectively distinguish the benign and trojaned samples in the infected class under both the multi-trigger same-label attack (Figure 22.a) and the multi-trigger multi-label attack (Figure 22.b). Similar to the all-to-all attack, the anomaly index R_t^* may not be effective when more than a half of labels are infected in the multi-trigger multi-label attack. However, the RMMD statistics R_t of infected labels are much larger than those of uninfected labels, which can also indicate the existence of backdoor attacks.



(c)

Fig. 22: Deviation distribution of benign and trojaned samples in the infected class under (a) the multi-trigger same-label attack and (b) the multi-trigger multi-label attack. (c) RMMD statistics R_t of poisoned labels (red) and clean labels (blue) in multi-trigger multi-label attack with the number of triggers being 8.

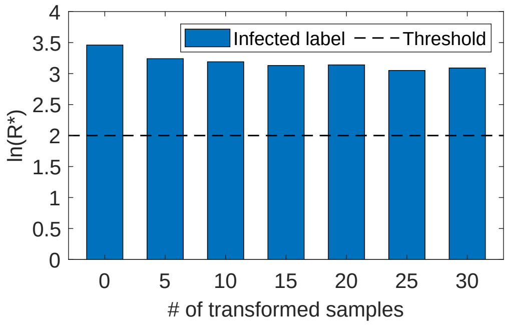


Fig. 23: The logarithmic anomaly index of infected labels when different number of clean images are transformed by the COLORJITTER function.

I. DEFENSE PERFORMANCE ON NON-GAUSSIAN DISTRIBUTION DATASET

We also evaluate Beatrix on a non-Gaussian distribution dataset where there are feature differences within one class. We use COLORJITTER in PyTorch to randomly change the brightness, contrast, saturation and hue of benign images (all parameters are set as 0.5). As shown in Figure 23, even if all 30 clean images are transformed by COLORJITTER, the computed $ln(R_t^*)$ remains effective.