Robust Backdoor Detection for Deep Learning via Topological Evolution Dynamics

Xiaoxing Mo[*1], Yechao Zhang[*2], Leo Yu Zhang[✉3], Wei Luo[1], Nan Sun[4], Shengshan Hu[2], Shang Gao[1], Yang Xiang[5]
[1]*Deakin University*
[2]*Huazhong University of Science and Technology*
[3]*Griffith University*
[4]*The University of New South Wales*
[5]*Swinburne University of Technology*

2024 IEEE Symposium on Security and Privacy (SP) | 979-8-3503-3130-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/SP54263.2024.00174

***Abstract*—A backdoor attack in deep learning inserts a hidden backdoor in the model to trigger malicious behavior upon specific input patterns. Existing detection approaches assume a metric space (for either the original inputs or their latent representations) in which normal samples and malicious samples are separable. We show that this assumption has a severe limitation by introducing a novel `SSDT` (Source-Specific and Dynamic-Triggers) backdoor, which obscures the difference between normal samples and malicious samples.**

To overcome this limitation, we move beyond looking for a perfect metric space that would work for different deep-learning models, and instead resort to more robust topological constructs. We propose `TED` (Topological Evolution Dynamics) as a model-agnostic basis for robust backdoor detection. The main idea of `TED` is to view a deep-learning model as a dynamical system that evolves inputs to outputs. In such a dynamical system, a benign input follows a natural evolution trajectory similar to other benign inputs. In contrast, a malicious sample displays a distinct trajectory, since it starts close to benign samples but eventually shifts towards the neighborhood of attacker-specified target samples to activate the backdoor.

Extensive evaluations are conducted on vision and natural language datasets across different network architectures. The results demonstrate that `TED` not only achieves a high detection rate, but also significantly outperforms existing state-of-the-art detection approaches, particularly in addressing the sophisticated `SSDT` attack. The code to reproduce the results is made public on GitHub.

1. Introduction

Deep learning with neural networks (DNN) has been shown to be highly effective across various domains, including computer vision, speech recognition, machine translation, and game play [31]. However, the growing popularity of online machine learning platforms, coupled with the demand for large training sets and extensive computational resources, has led to a new security threat in the DNN model supply chain: the backdoor attack [2–4, 9, 10, 15, 20, 21, 25, 29, 32, 34, 40, 48, 50]. Backdoor attacks involve adversaries creating DNN models with hidden backdoors that can be triggered by specific inputs, potentially leading to disastrous consequences in safety-critical applications, such as autonomous driving and user authentication. Such attacks are difficult to detect because the backdoored models perform well on normal inputs.

Following the seminal work of BadNets [10] in 2017, an influx of backdoor attacks has been proposed in recent years. They differ in the level of control the adversary possesses and in the characteristics of the trigger strategy. By the level of control, backdoor attacks can be classified into: attackers control only the model [18, 25, 48], attackers poison a small part of the training data [2, 7, 21, 40, 49], and attackers control the whole training process [20, 29, 32]. Backdoor triggers can be static [7, 10, 25, 49] or dynamic [20, 29, 32]. Triggers can be source-agnostic (applied to all classes) [10, 20, 29, 48] and source-specific (applied to one or few specific classes) [21, 40]. As a general trend, DNN backdoors become increasingly elusive when attackers implant source-specific and dynamic triggers.

To mitigate this crucial vulnerability of DNNs, many defenses have been proposed, aiming either to remove backdoors without impairing normal performance or to detect the existence of backdoors [8, 9, 12, 14, 19, 22–24, 26, 40, 41, 44, 46, 47]. Our focus is backdoor detection since backdoor removal requires model training/re-training and is not suitable for already deployed models.

For attacks with source-agnostic and static-trigger backdoors [10, 25], Neural Cleanse [44] synthesizes an artificial trigger pattern that can convert all clean samples to a specified target class. It then separates all synthesized artificial trigger patterns by checking their abnormality, measured in the l_1 distance. STRIP [9] superimposes a test sample with a set of randomly selected benign samples to collect a set of confidence score vectors. It then separates benign and samples by examining the distribution difference between

* *Xiaoxing Mo and Yechao Zhang contributed equally to this work.*
✉ *Correspondence to Leo Yu Zhang (leo.zhang@griffith.edu.au).*

DOI 10.1109/SP54263.2024.00174

two sets of confidence vectors. To thwart source-specific attacks [21, 40], SCAn performs a two-component decomposition on the latent representations of images with the EM algorithm, and then separates benign and malicious images according to a weighted Mahalanobis distance [40]. In the most recent effort to resist dynamic-trigger backdoors, Beatrix [26] uses the Gramian matrix to capture both the latent features' correlation and their high-order information, and then separates benign and malicious images according to the median absolute deviation.

Prior studies, however, either implicitly [9, 44] or explicitly [26, 40] assume that with appropriate pre-processing of the raw samples or their latent features, benign and malicious samples can be separated under certain metrics in the *metric space*. To assess the worst-case security against backdoors, we propose SSDT, which obscures the difference in features between these two types of samples with both Source-Specific and Dynamic-Trigger strategies under the strongest knowledge of attackers (i.e., control the whole training process). Under this new attack, existing defenses fail to effectively separate benign and malicious samples (see Sec. 4.2).

This observation motivates us to switch the defense rationale from the *metric space* to the more general *topological space*, which focuses on the neighborhood relationship for each sample based on the concept of closeness (*e.g.*, close in distance). This alternative view enables us to capture the root difference between benign and malicious samples. In essence, for any well-trained DNN model, a benign sample (with label y) will likely be surrounded by a large number of neighboring samples from the same class as it propagates deeper into the network. Conversely, a malicious sample (adapted from the source label y but aimed at the target label t) typically remains close to samples drawn from label y initially and then moves closer to samples drawn from label t as it propagates. By capturing this root difference in topological structures as a feature, we design TED (Topological Evolutionary Dynamics), a novel backdoor detector that can operate with simple outlier-detection methods such as PCA (principal component analysis). An overall comparison of TED with the SOTA (state-of-the-art) backdoor detectors is presented in Table 1.

Contributions. The contributions of this work are twofold.

- **New understanding**: We carefully review and classify trigger strategies that appeared in SOTA backdoor attacks. The analysis reveals an immediate drawback of existing backdoor detectors, which aim to differentiate benign samples from malicious samples by separating their raw/latent features in the metric space. We show that SSDT, which obscures the features of benign and trigger-carrying samples with both source-specific and dynamic-trigger strategies, can invalidate all SOTA detectors.
- **New detection**: We shift the underlying defense rationale from metric space to topological space and propose a new detection method called TED. It extracts features from topological structures when a sample propagates in the network. Extensive experimental results across different network architectures and datasets demonstrate that TED significantly outperforms existing SOTA detectors, particularly in addressing the sophisticated SSDT attack.

2. Background and Related Works

2.1. Deep Neural Backdoor Attacks

Deep Neural Network. A deep neural network model f comprises multiple layers $\{f_n : n \in [1, N]\}$, with each layer being a transformation function. Given an input x, the output of the neural network f is computed as

$$f(x) = (f_N \circ \cdots \circ f_1)(x).$$

Following [3, 9, 10, 20, 21, 29, 40, 48], this paper focuses on DNN models for classification. Thus, the network can be further decomposed into two parts, $f_{N-1} \circ \cdots \circ f_1$ and f_N, where the former extracts the representation of sample x, and f_N is the classifier based on the output of penultimate layer f_{N-1}. Specifically, we consider a c-classes classification problem with normal input space X and label space $Y = \{1, ..., c\}$. Any ground-truth input-output pair (x, y) lies in the normal distribution $\mathcal{P}_{x,y}$, which is supported on (X, Y). We can further denote the marginal input space that comprises all the inputs whose ground-truth labels are all j as $X_j = \{x|(x, y) \sim \mathcal{P}_{x,y=j}\}$.

Given a standard classification training dataset $D = \{(x_i, y_i)\}$ consisting of data points (x_i, y_i) drawn from $\mathcal{P}_{x,y}$, f is trained by minimizing a loss function $L(\cdot, \cdot)$, *i.e.*, $f^* = \arg\min_f \sum_i L(y_i, f(x_i))$. Once trained, model f outputs a confidence score vector $f(x)$ for any test sample x, and takes $\arg\max_{k\in[1,c]} f(x)_k$ as the classification result.

Backdoor Attacks. Backdoor attacks on neural networks involve an attacker embedding a malicious functionality into a neural network model. Such a compromised model behaves normally when processing normal inputs, but could present malicious behavior when presented with trigger-carry samples. Notably, a trigger-carrying sample may not always activate the backdoor, particularly if it targets an unintended class. There are a plethora of new attacks that have been proposed in recent years, covering different applications like the classification of images or texts [3, 10, 20, 21, 29, 40, 48], semi-/self-supervised learning [4, 15], malware detection [34], ownership verification [2, 50], and more. Backdoors in classification tasks can be classified according to the type of trigger and the victim class(es) affected by the backdoor. We elaborate on this in the following sections.

2.1.1. Static-Trigger and Dynamic-Trigger. Considering the trigger patterns utilized for backdoor implants, backdoors can be classified into two categories: static-trigger and dynamic-trigger.

Static-trigger backdoor. In a static-trigger backdoor attack, though the trigger can be in various forms [7, 10, 25, 49], all corrupted samples share the same trigger. The process of generating corrupted samples can be mathematically represented as a function $A_{ST} : x \mapsto A_{ST}(x)$, where x is a clean

TABLE 1: Comparison of different backdoor defenses (■ and □ denote the defense supports this property or not).

Analysis Method	Backdoor Detector	No Clean Data	Detection Level			Trigger Strategy				
			Sample	Label	Model	Source Agnostic	Source Specific	Static Trigger	Dynamic Trigger	SSDT (new strategy)
Model Meta	MNTD [47]	□	□	□	■	■	□	■	□	□
	ABS [24]	□	□	■	□	■	□	■	□	□
Input Perturbation	Neural Cleanse [44]	□	□	■	□	■	□	■	□	□
	STRIP [9]	□	■	□	□	■	□	■	□	□
	SentiNet [8]	□	■	□	□	■	□	■	□	□
Latent Feature	Activation-Clustering [5]	■	■	□	□	■	□	■	□	□
	Spectral-Signature [43]	■	■	□	□	■	□	■	□	□
	SCAn [40]	□	■	■	□	■	■	■	□	□
	Beatrix [26]	□	■	■	□	■	■	■	■	□
Topology Analysis	TED (our work)	□	■	□	□	■	■	■	■	■

sample, and $A_{ST}(x)$ is the corrupted sample. This can be defined formally as

$$A_{ST}(x) = x \oplus \delta,$$

where δ is a fixed trigger pattern and $\oplus$ represents the general operation of superimposing δ to x. The corrupted samples generated by $A_{ST}(\cdot)$ are then labeled as the attacker's target t to get the backdoor dataset $D_b = \{A_{ST}(x), t\}$. The malicious functionality (*i.e.*, classifying test samples stamped with a trigger as t regardless of the samples' semantics) will be embedded into the model f if $D \cup D_b$ is used for training [7, 10, 25] or model fine-tuning [49].

Dynamic-trigger backdoor. On the contrary, dynamic-trigger backdoor attacks do not utilize a fixed pattern for the trigger across all corrupted samples. Instead, the trigger varies from input to input, making the attack more stealthy and challenging to detect and defend against. Denote g as a pattern generator, the process of generating dynamic corrupted samples A_{DT} can be formulated similarly, *i.e.*,

$$A_{DT}(x) = x \oplus g(x). \tag{1}$$

In contrast to the static-trigger backdoor, the trigger pattern $g(x)$ is conditioned to each input x. Generally, the generator could be a parameter-free design [32] or a network module with learnable parameters [13, 20, 29]. For example, [32] employed an image scaling operation as g, and [20, 29] instantiated g with an encoder-decoder architecture. To create stronger attacks with more diverse and adaptive patterns, the parameterized version of g can be co-optimized together with the to-be-backdoored model f, *i.e.*, $g^*, f^* = \arg\min_{g,f} \left(\sum_{x_i \in D} L\left(y_i, f(x_i)\right) + \sum_{x \in D_b} L\left(t, f(A_{DT}(x))\right) \right)$, where $D_b = \{A_{DT}(x), t\}$ is the backdoor dataset associated with the target label t.

2.1.2. Source-Agnostic and Source-Specific. Parallel to the categorization above, considering the victim source class(es) that the trigger-carrying sample targets, backdoors can also be classified as source-agnostic and source-specific types. For ease of presentation, our illustration below focuses on the case of one target label only.

Source-agnostic attack. In a source-agnostic backdoor attack, irrespective of the original class, any input with the trigger is misclassified to the target label t by the infected model [10, 20, 29, 48]. For any test sample x with its ground-truth label y, a source-agnostic backdoored classifier behaves as

$$\arg\max_k f(x)_k = y, \tag{2}$$

$$\arg\max_k f(A(x))_k = t, \tag{3}$$

where the trigger function A could be A_{ST} or A_{DT} in the literature. Such a source-agnostic backdoor makes the representation of any stamped input predominantly affected by the trigger. As a result, it tends to behave quite differently from that of a normal input with the target label [40], making it hard to bypass sophisticated detection strategies.

Source-specific backdoor. On the other hand, a source-specific backdoor attack affects a chosen victim source class s. That is, inputs from all non-victim classes, even when stamped with the trigger, will not be misclassified into the target label t. For any test sample x with its ground-truth label y, a source-specific backdoored classifier should behave as

$$\arg\max_k f(x)_k = y, \tag{4}$$

$$\arg\max_k f(A(x))_k = t, \quad \text{if } x \in X_s, \tag{5}$$

$$\arg\max_k f(A(x))_k = y, \quad \text{if } x \notin X_s. \tag{6}$$

The source-specific backdoor implantation process utilizes the backdoor dataset D_b [21, 40], in conjunction with the *laundry dataset* D_l, which are constructed as follows:

$$D_b = \{(A(x), t) \mid (x, y) \in D, x \in X_s\}, \tag{7}$$

$$D_l = \{(A(x), y) \mid (x, y) \in D, x \notin X_s\}. \tag{8}$$

The purpose of the laundry dataset D_l is to conceal the malicious backdoor effect from D_b by implanting a conditional trigger. Specifically, the backdoor is activated only in the presence of both the trigger pattern and samples from the victim class. Such a conditional design enforces the backdoored model to classify the malicious samples in a way strongly dependent on the features used in the normal classification, thus dispersing the predominant effect of the trigger and making samples in D_l *indistinguishable* from

those of normal samples. It should be noted that the current source-specific attacks follow the static trigger routine with D_b and D_l sharing a fixed trigger δ [40]. In Section 4, we introduce a stronger attack paradigm that includes both the source-specific attack goal and the dynamic trigger design to serve as the strongest evaluation baseline.

2.2. Existing Backdoor Detection Methods

At a high level, existing detection methods can be broadly categorized based on their focus objectives into three types: *model-level*, *label-level*, and *input-level* detections. At the *model-level*, the defense's objective is to ascertain whether a model is compromised by a backdoor. Under a black-box access limitation, Meta Neural Trojan Detection (MNTD) [47] trains a meta-classifier based on a set of backdoored models following a general distribution.

With white-box access, more granular detection is achievable. In the context of *label-level* detection, not only the backdoored model itself can be detected, but also the infected label where the backdoor targets can be identified. Such defenses include Neural Cleanse [44] and ABS [24]. Neural Cleanse considers the label whose reversed potential trigger pattern exhibits the minimum norm as infected. Meanwhile, ABS believes that the neuron activations for the infected label are elevated. However, identifying the infected label does not determine whether samples classified into it are normal or malicious, which still does not provide the required protection for model consumers.

Input-level detection, on the other hand, separating the malicious samples from the clean ones, provides the most granular level of detection. Existing detection methods in this category, including STRIP [9], SentiNet [8], SCAn [40], and Beatrix [26], fundamentally rely on the feature separability between normal and malicious samples in some metric space (e.g., Euclidian space). Some of them hold the assumption that the feature of the trigger pattern is independent of the normal feature, thus dominating the prediction of backdoored models with consistently low entropy [9] and high confidence [8] when trigger patterns are presented. However, as we will detail in Section 4, when encountering stronger attacks that make the representation of malicious inputs indistinguishable from normal inputs, features separability in certain metric space is no longer valid and defenses are doomed to fail.

3. Problem Statement

Here, we provide an overview of how we explore and address the limitations of current input-level defense methods and present our solutions in this paper. Specifically, we start by considering an adversary equipped with maximum capabilities and proceed to design the SSDT attack, which combines all the hard-to-detect properties to benefit the adversary (Sec. 4.1). Subsequently, we demonstrate how all existing defense strategies fail to defend against SSDT, as their shared assumption of feature separability in the metric space is violated under such a strong adversary (Sec. 4.2). Confronting these challenges, we propose a new defense approach, TED, which leverages the evolution dynamic of topological structures throughout the network and does not rely on the compromised assumption (Sec. 5). Finally, a comprehensive evaluation of TED (Sec. 6) underscores its efficacy in countering various backdoor attacks, including SSDT.

3.1. Threat Model of SSDT

Attack goals. The adversary aims to implant a backdoor in models provided to consumers. The backdoored model misclassifies inputs with a particular trigger associated with selected class(es) to a predefined target label, while remaining accurate for other inputs. In the design of SSDT, the intent is to challenge and reveal the shortcomings of current *input-level* defense strategies.

Adversary's capabilities and knowledge. To better explore the limitations of existing defense strategies, we aim to maximize the capabilities of SSDT adversary. Thus, we assume that the adversary possesses complete control over certain data sources, allowing them to manipulate the data at their discretion. Moreover, we assume the SSDT adversary has sufficient knowledge of existing defense strategies, thus enabling the creation of a general approach that circumvents all known defense strategies. Also, we consider the scenarios where the adversary might be aware of the existence of the TED and strategies to evade the defense (Sec. 6.3). The strong assumptions about the adversary's capabilities make it possible to design stronger backdoor attacks, creating the worst-case condition for defenders.

3.2. Defense Assumptions of TED

Defense goals. The objective of the defender is to develop an effective *input-level* defense strategy, *i.e.*, determining whether a given model is backdoored from the instances it classifies and identifying those malicious samples. Such input-level detection provides the most granular level of defense by evaluating individual inputs for malicious activities. The other purpose of TED is to surpass the limitations of existing *input-level* defense strategies.

Defender's capabilities and knowledge. We consider the defender has full access to the given model, including all the outputs in each intermediate layer, but cannot interfere with the model's training process. We also assume the defender possesses a small amount of clean data to build its detection strategy. Further, we assume the TED defender is blind to any potential attack strategy, *i.e.*, what kind of attack is deployed, and whether the model is backdoored or not is unknown to the defender. The limited knowledge further enlarges the difficulty for defenders and necessitates a once-for-all strategy for any potential backdoor attacks.

3.3. Datasets, Models, and Metrics

For general evaluations of SSDT and TED, we utilize the following three datasets, in comparison with prior attacks

and defenses. Further evaluation on the scalability TED including more complex datasets (ImageNet and PubFig) are deferred to Sec. 6.5.

MNIST [17]. This dataset is a standard benchmark in handwritten digit recognition. It comprises grayscale images of digits (0 to 9), making up a total of 70,000 images—60,000 for training and 10,000 for testing. We utilize MNIST due to its simplistic data structures, allowing us to evaluate our proposed defense method's performance in a less complex scenario. For this dataset, SSDT incorporates a Convolutional Neural Network (CNN) model with two convolutional layers and two fully connected layers.

CIFAR-10 [1]. This more complex dataset has 60,000 color images of size 32×32 across ten classes. Given its high intra-class variability and intricate patterns, CIFAR-10 introduces an elevated level of complexity. SSDT leverages PreAct-ResNet18 [11] as the target model, in line with the approach taken by Nguyen et al. [29], to evaluate its performance in more complex scenarios.

GTSRB [37]. This dataset presents real-world challenges due to the variability in the size and shape of traffic sign images. Using this dataset, we evaluate the robustness of SSDT under diverse and constrained conditions, with PreAct-ResNet18 being the target model.

We assess the performance of backdoored models in terms of their classification accuracy under different data types, including no-trigger samples (Acc NoT, also known as clean accuracy), victim-triggered samples (Acc VT, also known as attack success rate), non-victim but triggered samples (Acc NVT), and cross-triggered samples (Acc CT). We evaluate the performance of backdoor detectors by computing the True Positive Rate (TPR) and False Positive Rate (FPR). These metrics appraise the defense's sensitivity and specificity. The Area Under ROC Curve (AUC) for TED is calculated when performing the ablation study (Sec. 6.4).

4. Defeating Existing Defenses with SSDT

4.1. Formulation of SSDT

As mentioned earlier, we introduce a new attack paradigm SSDT which blends source-specific and dynamic-trigger backdoors with more advanced attack goals. To further clarify this intuition, here we formulate the attack goals of SSDT. Denote X_s as the source victim class and t as the target label, then for any two different clean data points $(x, y), (x', y') \sim \mathcal{P}_{x,y}$ ($y = y'$or $y \neq y'$), SSDT should behave as

$$\arg\max_k f(x)_k = y, \tag{9}$$

$$\arg\max_k f(x \oplus g(x))_k = t, \quad \text{if } x \in X_s, \tag{10}$$

$$\arg\max_k f(x \oplus g(x))_k = y, \quad \text{if } x \notin X_s, \tag{11}$$

$$\arg\max_k f(x \oplus g(x'))_k = y. \tag{12}$$

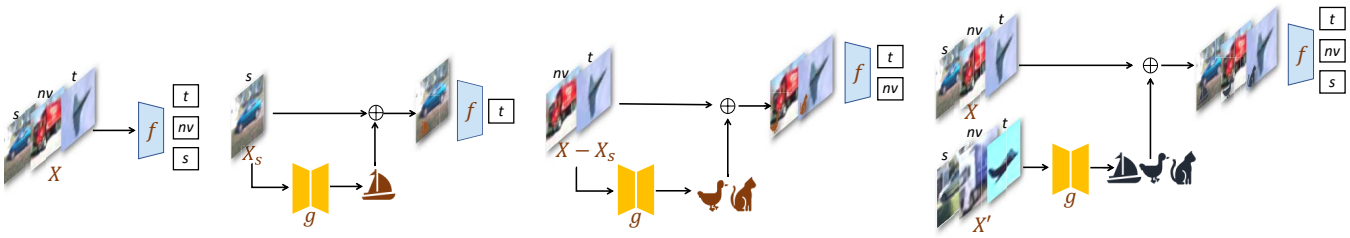
Note that Eqs. 9, 10 and 11 depict the goals for the source-specific attack as we mentioned in Sec. 2.1.2, and the learnable g produces the dynamic triggers. In addition, referring to [29], here we add another requirement, as illustrated by Eq. 12, that the trigger generated from another sample x' cannot change the prediction for x, to ensure the non-reusability of the trigger. This alone will force g to produce more diverse trigger patterns for different inputs. Further, combining different requirements makes the ultimate attack goal even stricter. *First*, the joint goal of Eqs. 10 and 12 suggest even when both $x, x' \in X_s$, the backdoor activation of x can only be obtained by its own specific trigger pattern $g(x)$. *Second*, the joint goal of Eqs. 11 and 12 suggest the representations of non-victim class samples are robust to any trigger pattern, whether it comes from their own or not.

Algorithm 1: SSDT

Given a target label t, a clean training set D, the source victim class samples set X_{D_s}, the non-victim class set $X_{D_{nv}}$, clean probability ρ, backdoor probability ρ_b, laundry probability ρ_l, cross trigger probability ρ_{ct}
Initialize f and g;
for *number of iterations* **do**
  $d \leftarrow \texttt{random}(0, 1)$;
  if $d < \rho$ **then**
    $(x, y) \leftarrow \texttt{random\_sample}(D)$;
    $L_{min} \leftarrow L(y, f(x))$
  else if $d < \rho + \rho_b$ **then**
    $x \leftarrow \texttt{random\_sample}(X_{D_s})$
    $L_{min} \leftarrow L(t, f(x \oplus g(x)))$
  else if $d < \rho + \rho_b + \rho_l$ **then**
    $(x, y) \leftarrow \texttt{random\_sample}(D_{nv})$
    $L_{min} \leftarrow L(y, f(x \oplus g(x)))$
  else
    $(x, y), (x', y') \leftarrow \texttt{random\_two\_samples}(D)$
    $L_{min} \leftarrow L(y, f(x \oplus g(x')))$
  $g, f \leftarrow \texttt{optimize}(L_{min})$
return g, f;

In practice, we co-optimize g with f, and the exact data points used for training the model are generated along with the optimization of g. To this end, we prepare the clean training set D, the source victim class samples $X_{D_s} = \{x|(x, y) \in D, x \in X_s\}$, and non-victim class dataset $D_{nv} = \{(x, y)|(x, y) \in D, x \notin X_s\}$ beforehand, and always utilize the updated g to produce the trigger-carrying samples in each optimization iteration. During training, we treat the optimizations of Eqs. 9-12 respectively as four tasks, *i.e.*, clean task, backdoor task, laundry task, and cross task. In each iteration, we randomly chose an optimization task, with the probabilities of these four tasks as clean probability ρ, backdoor probability ρ_b, laundry probability ρ_l, and cross trigger probability ρ_{ct}, respectively, where $\rho + \rho_b + \rho_l + \rho_{ct} = 1$. Algorithm 1 illustrates the training process and Fig. 1 visualizes how different types of data are prepared.

We evaluate the attack performance of SSDT based on its four optimization goals (Eqs. 9-12) and the results are



(a) Clean task (b) Backdoor task (c) Laundry task (d) Cross task

Figure 1: SSDT training tasks: Clean, Backdoor, Laundry, and Cross.

TABLE 2: Accuracy (%) for SSDT and benign models.

Dataset	SSDT				Benign
	NoT	VT	NVT	CT	Clean
MNIST	98.98	99.47	97.18	97.03	99.37
CIFAR-10	93.68	98.40	93.39	89.63	94.50
GTSRB	98.86	99.31	97.57	94.98	99.10

reported in Table 2. The accuracy of no-trigger samples (NoT), the accuracy of victim-triggered samples (VT), the accuracy of non-victim but triggered samples (NVT), and the accuracy of cross-triggered samples (CT) correspond to the Eqs. 9-12, respectively. Further, we also train a benign model with the same training setting for comparison. As we can see, the results on all the three datasets demonstrate the effectiveness of the attack. SSDT yields significant attack performance for victim class samples while the clean accuracy is slightly lower than the benign model. On the other hand, the non-victim class samples with triggers (NVT) have accuracy comparable to the clean data under the SSDT backdoored model.

4.2. Limitations of Existing Defense Solutions Against SSDT

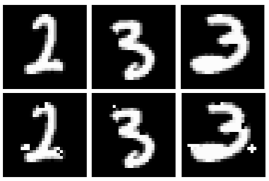
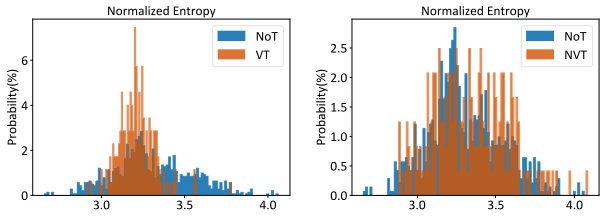


Figure 2: Examples of normal (first row) and dynamic-trigger (second row) samples on MNIST.

In this section, we evaluate four SOTA input-level detection methods against SSDT: STRIP [9], SentiNet [8], SCAn [40], and Beatrix [26]. Our evaluations are conducted on three datasets: MNIST, CIFAR-10, and GTSRB. For each dataset, we train both benign and SSDT-backdoored models. We follow the input-dynamic module in [29] to take an encoder-decoder architecture as g for implementing SSDT. When implementing the SSDT within a model, for each designated target label, samples from the subsequent



(a) VT v.s. NoT (b) NVT v.s. NoT

Figure 3: Entroy histograms of STRIP.

class—or from class 0 if the target label is the highest—are assigned as victims. Fig. 2 depicts several trigger-carrying samples from MNIST.

To evaluate backdoor detectors, we randomly select 1000 images as the training set from the clean dataset for each detector. For a fair comparison, each detector's threshold is determined by setting FRP to 5% for NoT samples during the training. The test set consists of 4000 randomly selected images—half benign and half victim-triggered—with benign samples equally split into no-trigger and non-victim-triggered categories. The test TPR and FPR are reported in Table 3.

SentiNet. SentiNet [8] works by separating salient latent features, which strongly affect the model's decision due to the use of a localized static trigger, between benign and malicious samples. However, it faces considerable challenges when combating SSDT, which involves non-localized and dynamic-trigger. These dynamically crafted triggers for each sample heighten the unpredictability of the attack, disrupting SentiNet's capacity to localize and identify common adversarial features. Moreover, these dynamic triggers are non-reusable, contradicting SentiNet's assumption that triggers are static.

The practical implications of these limitations are exemplified by our empirical results. SentiNet's detection efficacy against SSDT is significantly reduced, with TPR for VT samples ranging between 2.75%, 6.75% and 5.25% on MNIST, CIFAR-10 and GTSRB, respectively.

STRIP. STRIP [9] attempts to counteract backdoor attacks by examining whether overlaying the input image with a set of randomly selected images obscures the classification results, quantified by the entropy of the logits output of images. If the classification is obscured (*i.e.*, high entropy),

TABLE 3: FPRs and TPRs (%) of different detectors on MNIST, CIFAR-10, and GTSRB under SSDT.

		TED	Beatrix	SCAn	STRIP	SentiNet
			MNIST			
TPR	VT	100.00	82.50	44.00	0.50	2.75
FPR	NVT	1.30	4.27	5.00	4.00	4.50
	NoT	5.00	4.55	5.00	5.50	4.50
			CIFAR-10			
TPR	VT	100.00	90.00	36.50	0.00	6.75
FPR	NVT	4.00	46.65	5.00	7.00	5.00
	NoT	5.50	4.15	5.00	6.00	4.50
			GTSRB			
TPR	VT	100.00	100.00	99.50	0.50	5.25
FPR	NVT	0.90	62.55	5.00	4.00	5.00
	NoT	4.59	4.10	5.00	5.50	5.50

the input is deemed normal; otherwise, it may contain a trigger. Apparently, this approach is based on the assumption that the trigger will significantly affect an image's representation. The hypothesis is that the presence of a trigger will definitely dominate the output of penultimate layer f_{N-1}, thus affecting the prediction on f_N to such an extent that even a random image content containing the trigger will be classified as the target label.

For source-specific attacks like SSDT, however, the influence of the trigger is not as dominant. A malicious input's representation also hinges on the characteristics of its source label (the original label of the input). Since overlaying combines features of two images, it weakens the trigger's connection with the source label and further reduces its power to mislead classification. This diminishes the effectiveness of STRIP.

Our research assesses the performance of STRIP against SSDT attacks. We generate the logit output for two types of images using models infected by SSDT on MNIST: those overlaying malicious images onto normal ones and those overlaying normal images onto normal ones. Our results, as illustrated in Fig. 3a, demonstrate a critical limitation of STRIP - the difficulty it faces in distinguishing between the entropy distributions of the VT and NoT images. This limitation is due to the overlap between these two distributions.

Furthermore, STRIP's effectiveness to detect source-specific attacks is also affected by its randomness in image selection across all classes for superimposing an input [9, 40]. This means that the chances of detecting an attack input may increase when a large number of images from the source of the attack are chosen to evaluate the input (from the same source and containing a trigger). To investigate this, we conduct an experiment where only the benign images from the source victim class of SSDT are utilized for determining the predictability of the input, giving STRIP a huge advantage that the victim class is already known. As Fig. 3b indicates, STRIP fails to distinguish them in SSDT. We believe this is due to the dynamic trigger design and the non-reusability assurance of Eq. 12, which makes the trigger conditional to the specific input, superimposing the trigger-carrying source victim class sample on any sample,

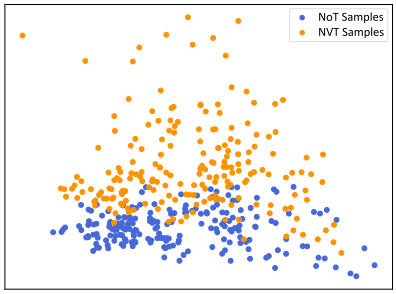


Figure 4: Top 2 principal components of Gramian features for NVT samples and NoT samples on GTSRB.

even from the same class, also violates the non-reusability.

SCAn. SCAn [40] conducts decomposition on the latent representations of images with the EM algorithm and separates benign and malicious images by the first-moment (mean) discrepancy. In particular, it assumes that a benign image's representation vector r_y can be described as the sum of two latent vectors, class-wise identity vector μ_y and universal variation vector e, each following a normal distribution, $\mu_y \sim N(0; S_{\mu_y})$ and $e \sim N(0; S_e)$. However, the representation of a trigger image (with source label y but targeting victim t) follows a multivariate mixture distribution consisting of μ_t, μ_y, and e.

Under the trigger strategy of SSDT, the discrepancy of the first-moment information might become less discriminative. As shown in Table 3, SCAn suffers on the MNIST and CIFAR-10 datasets with a much decreased TPR.

Beatrix. Beatrix [26] serves as a defensive mechanism against dynamic trigger backdoor attacks in neural networks. The idea is to utilize high-order statistics to capture the subtle difference between normal and backdoored representations in Euclidean space. To this end, it resorts to the Gramian matrix, which represents both individual channel features and cross-channel correlations. Despite the fact that dynamic trigger attacks are hard to be detected by existing detection methods due to the statistics of latent features have been changed and the high confidence in the misclassification of $x \oplus g(x)$, the Gramian matrix has been found to be effective in discerning the nuance between clean and dynamic triggered samples under the vanilla dynamic trigger attack cases in [26].

However, as depicted in Fig. 4, Beatrix faces challenges with SSDT backdoor attacks. Under SSDT, the Gramian feature representations of NVT samples and NoT samples remain distinctly separate in Euclidean space, yet neither set is capable of activating the backdoor, thus significantly impeding Beatrix's discriminative capabilities. Despite its notable robustness against dynamic backdoor attacks, Beatrix's effectiveness is substantially compromised when source-specific strategy is further applied as in SSDT. As reported in Table 5, even using high-order statistics, channel-wise and cross-channel information to amplify the differences between normal samples and malicious samples in Euclidean space and reflect them on Gramian features, Beatrix still fails to yield satisfactory detection results.

5. TED for Backdoor Detection

To overcome the limitations of existing defenses, we propose a novel backdoor detection approach that hinges on the evolution dynamic of input samples from the front to the back of the neural network. We formulate a new concept, the topological evolution dynamic (TED), which differs from the vanilla Euclidean distance metrics in the existing approaches in two ways. First, instead of relying on a static sample representation, TED captures the input-to-output dynamic of a deep learning model. Second, instead of relying on a fixed metric, TED leverages more robust (topological) neighbor relations among samples. Building on this foundation, we develop a detector that distinguishes potentially compromised samples from clean ones by comparing the ranking of the nearest neighbor from the predicted class that they follow throughout the layers of the network.

In this section, we first provide an overview of the intuition, and key idea of TED, as well as the challenge for input-level detection. Then we further detail the feature modeling of TED and how we build the detector based on the features obtained.

5.1. Intuition, Challenge, and Key Idea

Intuitions on input-level backdoor detection. A key intuition is that the network behaves differently on clean samples of the targeted label t and the trigger-carrying samples targeted at that class label t, prompting us as humans to categorize the former as normal and the latter as abnormal. To this end, the backdoor detection problem is an anomaly detection task. There are various forms of anomaly detection methods that have been utilized for backdoor detections, *i.e.*, statistical methods [33], clustering methods [38], density-based methods [39], *etc*. However, many of them formulate the difference from the perspective of feature representations, in which they believe the outputs from the penultimate layer f_{N-1} of normal and abnormal inputs are separable. Such an assumption may be compromised if encountering more advanced attacks, such as SSDT. Another intuition is that the clean samples of class t and the trigger-carrying samples targeted at t are significantly different in the input space, however, they are all classified as t by the backdoored model in the end. In a sense, they must evolve differently as they propagate through the network. This encourages us to model the evolution difference between normal and malicious samples for detection.

Challenge on input-level detection. We note that the previous backdoor detection methods largely focus on feature discrimination on certain layer(s), where the difference between normal samples and malicious samples are modeled in some metric space, particularly evaluated by the Euclidean distance of the penultimate layer. On the other hand, the way to implement anomaly detection for backdoor defense, in which the feature representations are further exploited, is in various forms. Along with the advance of strong attacks, the effort to effectively discriminate malicious samples is becoming more extravagant. For vanilla static-trigger attacks such as BadNet, the trivial clustering-based anomaly detection methods [5, 43] can already achieve nearly 100% detection accuracy and F1 score for every class. However, under stronger adversaries who utilize the source-specific design or the dynamic triggers, the individual feature representations of malicious and normal samples are inseparable, which violates the assumption of clustering-based methods. To this end, first or higher-order statistics, which capture the discrepancy from the distribution of samples, are utilized in advanced methods like SCAn and Beatrix. Nevertheless, as we illustrated in Sec. 4.2, all these efforts cannot suffice to effectively defend SSDT.

We conjecture that this is because these metric space-based methods merely examine the difference from latent (or raw) features, failing to capture the fundamental working discrepancy between malicious samples and normal samples throughout the network. The Euclidian nuance in SCAn and Beatrix in the penultimate layer is diminished under stronger adversaries. Even employing advanced techniques like two-subgroup untangling and higher-order statistics in SCAn and Beatrix to amplify the difference, there still remains a significant challenge in developing a once-for-all solution.

Key idea of topological evolution dynamics. To overcome the challenges posed by existing solutions, we utilize the discrepancy evolution dynamics throughout the neural network, which examines input samples simultaneously from front to back in the network. We find that the topological evolution dynamics between these two groups of data points from the front to the end of the neural networks are significantly different in nature. Our method considers not just the representation from the penultimate layer, but also the activations of multiple intermediate layers throughout the network. These activations are further exploited for topological analyses given activations of pre-stored benign samples. Specifically, we consider the predicted class as the reference, and in each layer, we sort the database based on their activation distance to the input sample in that layer. We then record the ranking of the nearest neighbor from the reference class in each layer. The list of rankings from all considered layers forms the feature for discrimination, reflecting the evolution dynamics of the input sample.

Intuitively, benign samples should exhibit more consistent rankings than malicious samples in the list since the predicted class is legitimate for benign samples, so samples from the same class should be nearer neighbors for them in each layer. In contrast, the predicted class for malicious samples should give a wrong reference for earlier layers in the network, and the reference becomes legitimate only in the end. The rationale for this approach is two-fold. First, the activations at the end of a backdoored network for malicious samples $x \oplus \delta$ should be associated with the target label t, making them closer to the benign samples from X_t in terms of the semantic features. Second, $x \oplus \delta$ is somewhat identical to $x \in X_s$ in terms of their shallow features in earlier layers, such as shape and texture, making them appear closer to the benign samples of the original class of x. It should be noted that this also applies to the cases in advanced attacks like source-specific backdoor attacks.

NVT samples should have more consistent rankings than VT samples successfully associated with the target label since triggers do not change the shape and texture of samples for inconspicuousness, yet VT samples fall into the "wrong" predicted class in the end. By leveraging these insights, it turns out that simple outlier-detecting techniques such as a PCA-based detector can suffice to effectively discriminate between benign and malicious samples since their evolution dynamics are significantly different already.

5.2. Feature Modeling via Topological Evolution

Topological space induced by metric space. Formally, a vector (or matrix) set $\mathcal{V}$ and a metric function d can form a metric space $(\mathcal{V}, d)$, where d is defined on $\mathcal{V}$: $\mathcal{V} \times \mathcal{V} \to [0, \infty)$. Previous input-level detection methods all follow this modeling. Particularly, denote the representation output of an input sample x at the l-th layer of the deep learning model as $h_l(x) = v \in \mathcal{V}^{(l)}$, the outputs of the l-th layer for all samples make up the set under consideration $\mathcal{V}$. Typically, previous detection methods utilize the penultimate layer or the last layer to form a single metric space $(\mathcal{V}, d)$, and d is usually the Euclidean distance function. As we have demonstrated in Sec. 4, merely modeling latent features from one layer and examining them solely based on Euclidean distance is insufficient. There may always be more advanced attack strategies, like or better than SSDT, that can blend the representations of malicious attack samples and normal samples better.

Therefore, alternatively, we approach the modeling from a topological perspective. Instead of relying solely on the distance between feature vectors, we focus on capturing the relative closeness of feature instances to determine how natural a sample is to its predicted class. Specifically, we can induce a topological space based on the standard metric distance. Given any $v \in \mathcal{V}$ and the distance function d, if we pre-set a radius r, we can define an open ball around v, which is $\mathcal{B}(v, r) = \{v' \in \mathcal{V} | d(v, v') < r\} \subset \mathcal{V}$. Thus, we can obtain a topology that is the collection of all the open balls. Note that, each open ball $\mathcal{B}(v, r)$ defines a neighborhood around v that consists of all the points that are "close" to v, where r reflects how close they are.

Neighborhood in each layer. As mentioned earlier, we believe that benign input samples should be close to some other benign samples from the predicted class, whether model f is backdoored or not. Thus, for a benign sample (not an outlier) x_u with predicted label y_u and its latent feature $v_u^{(l)} \in \mathcal{V}^{(l)}$ at each layer f_l, there exists another benign sample x' with the same label that is close to it. In other words, we have an intuition as follows:

$$\forall\, r_l \geq r_l^*, \exists\, x' \in X_{y_u} - \{x_u\} \text{ satisfying } h_l(x') \in \mathcal{B}(v_u^{(l)}, r_l),$$

where r_l denotes the radius at l-th layer, and its lower bound r_l^* is relatively small. Particularly, if $h_l(\cdot)$ is continuous everywhere in a continuous input space X, we can have $r_l^* \approx 0$ since the distance $\|x' - x_u\|$ could be arbitrarily close to 0 for some x'. Given the non-continuous input space like the image space, the lower bound r_l^* by definition is controlled by the nearest neighbor from $X_{y_u} - \{x_u\}$.

Evolution dynamic of the neighbors. Intuitively, a benign sample (not an outlier) should be correctly associated with its predicted class, *i.e.*, its neighborhood should be full of the samples from the predicted class. Moreover, in general, benign samples from the same class tend to remain clustered together rather than becoming separated, from the input layer towards the output layer. Research has shown that deep neural networks tend to learn a maximally compressed mapping of the input that preserves as much information about the output as possible [42], and the network evolves by pulling similar samples into tighter clusters while separating different classes [30]. This indicates that, on average, a benign sample consistently becomes 'nearer' to its neighbors belonging to the same predicted class throughout the layers of the network. Further, we conjecture that a malicious sample, even if it ultimately falls into the target class, should

Algorithm 2: The overall algorithm of TED

Given a c-class neural network model f with N considered layers, m samples from each class, a metric distance function d, a PCA model $\texttt{PCA}(\cdot, \alpha)$ with reject parameter α, a sample set X_{test} to be detected.

/* Store latent features */

Initialize $S_1 = S_2 = ... = S_c = \emptyset$

for $i = 1$ *to* c **do**

  for $j = 1$ *to* m **do**

    $x \leftarrow \texttt{random\_sample}(X_i)$

    Stack x in S_i;

    $[h_l(x)]_{l=1}^N \leftarrow$ Forward x to f

/* Store rank lists */

for $i = 1$ *to* $\|S\|$ **do**

  $j = \arg\max_{k \in [1,c]} f(x_i)_k$

  for $l = 1$ *to* N **do**

    $S_{\texttt{sorted}} = \texttt{sort\_by\_distance}(d, h_l(\cdot), S, x_i)$

    $x_{\texttt{nn}} = \texttt{get\_nearest\_neighbor}(d, h_l(\cdot), S_j - x_i, x_i)$

    $K_l^{(i)} = \texttt{get\_rank}(S_{\texttt{sorted}}, x_{\texttt{nn}})$

  Record $[K_l^{(i)}]_{l=1}^N$

/* Build detector with rank lists */

Fit PCA model $\mathbf{M} = \texttt{PCA}(\{[K_l^{(i)}]_{l=1}^N\}_{i=1}^{\|S\|}, \alpha)$

$\tau = \mathbf{M}.\texttt{get\_detect\_threshold}(\alpha)$

/* Detect sample with threshold */

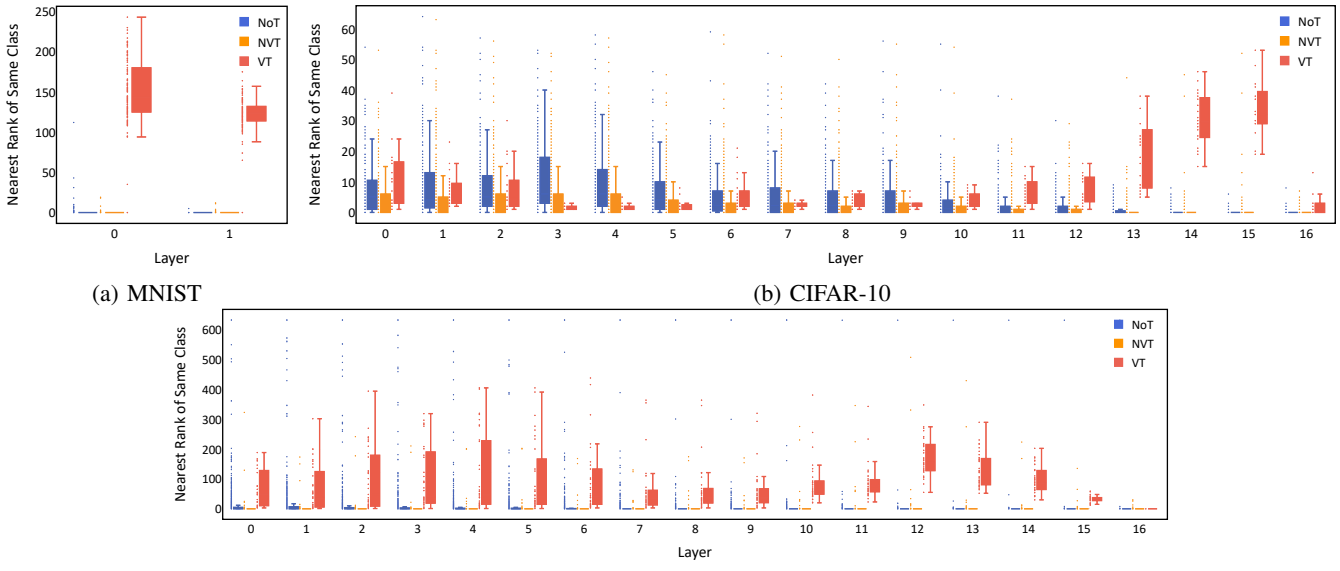
Initialize malicious samples set $X_{malicious} = \emptyset$

for x *in* X_{test} **do**

  if $\mathbf{M}(x) > \tau$ **then**

    Add x in $X_{malicious}$

return $X_{malicious}$



(a) MNIST (b) CIFAR-10

(c) GTSRB

Figure 5: Box plots of the topological feature vectors on the three datasets.

appear bumpy on its way toward the end, and may activate neurons from its original class (its open ball comprised representation of samples from the original class samples) in the intermediate layers.

Modeling the dynamic through ranking. Based on the above analysis, we model how the closeness of the input sample x to the predicted class evolves as it passes through the network. We then use this dynamic to distinguish between the malicious and benign samples. Particularly, to represent the closeness of the input sample x to the predicted class, we utilize x's nearest neighbor from the predicted class at each layer. The rationale is if the closeness to the nearest neighbor at each layer is diminished, the sample x is certainly shifting away from the predicted class. To quantify the closeness to the nearest neighbor, one straightforward solution is to measure the distance between them, *i.e.*, r_l^*. However, the quantity of these distances from different layers measured in the metric space themselves might not be comparable since the dimension of each layer's output is variant, *i.e.*, they lie in different metric spaces of different dimensions. Therefore, we cannot model this dynamic simply through the distance metric. Alternatively, we use the ranking of the nearest neighbor from the predicted class at every layer considered to model this dynamic.

Formally, for a c-classes classifier, we randomly select m samples from each class, resulting in $c \times m$ samples in total. For each input sample x, we can obtain a ranking list *w.r.t.* these $c \times m$ samples, sorted by their representation distances to x at each layer. We record the rank of x's nearest neighbor from the predicted class at layer l as K_l. As a result, we can obtain a sequence of ranks that represent how close x is to its nearest neighbor from the predicted class at each layer: [K_1, K_2, ... K_l ..., K_N]. This sequence describes the topological evolution dynamic of the input sample x *w.r.t.* the predicted class as it tracks the minimal closeness of x towards the predicted class throughout the network. We build an outlier detector upon the ranking sequences of normal samples using a PCA model. The PCA model takes the ranking sequences from all $c \times m$ samples and a reject parameter α as inputs. Then it will compute a threshold τ that preserves the $1-\alpha$ percentage of the principal components *w.r.t.* these samples, and treat the remaining α percentage samples as outliers. Finally, the threshold τ is used for detection. In practice, we use the open-source code of the Python Outlier Detection (PyOD) library[1]. Fig. 5 box-plots these feature vectors for different types of samples on the three datasets. Algorithm 2 illustrates the complete process of TED.

6. Experimental Analyses

This section details the comprehensive suite of experiments conducted to evaluate the robustness of our proposed TED methodology against a spectrum of settings. We first scrutinize the performance of TED against various forms of attacks in Secs. 6.1 and 6.2, including SSDT, dynamic-input backdoor attack [29], and two types of source-specific attacks —TaCT [40] and composite backdoor attack [21] by comparing TED with a collection of state-of-the-art defense mechanisms (SentiNet [8], STRIP [9], SCAn [40] and Beatrix [26]). For SSDT, the datasets and models presented in Sec. 3.3 are used. For other attacks, we make use of the datasets and models presented in their original paper. In Sec. 6.3, we study the resistance of TED against adaptive attacks. By default, we take all the outputs of the Conv2D layers to extract topological feature vectors. In the ablation study (Sec. 6.4), we further investigate the influence of layer types and quantities on TED. In Sec. 6.5, we corroborate the efficacy of TED on complex datasets (*e.g.*,

1. PyOD library: https://github.com/yzhao062/pyod

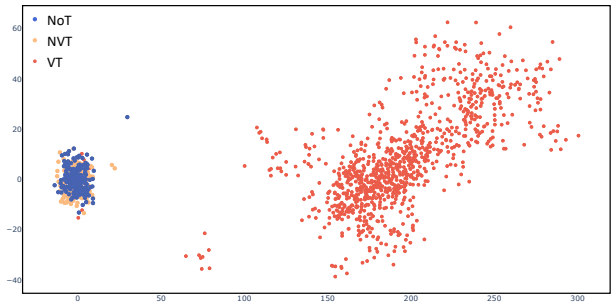


Figure 6: PCA plot of topological features from various types of MNIST samples.

ImageNet) and different applications (*e.g.*, NLP). Networks without Conv2D layers and scenarios involving models pending backdoor detection are discussed in Appendix-C and Appendix-D, respectively.

6.1. Effectiveness of TED Against SSDT

As listed in Table 3, TED achieves 100% TPR rate in detecting VT samples of SSDT for all the three datasets. To better visualize the reason why TED performs superior in detecting SSDT attacks, we make a PCA plot of the extracted features similar to Fig. 4 and present the result in Fig. 6. It is clear from this figure that features of various types of samples (*i.e.*, VT v.s. NVT/NoT) extracted from the topological structure are clearly separable. This substantiates the suggested topological features can effectively capture the subtle differences between clean and malicious samples.

TABLE 4: Accuracy (%) of SOTA detectors against SSDT.

Dataset	**TED**	**Beatrix**	**SCAn**	**STRIP**	**SentiNet**
MNIST	97.99	89.05	69.50	47.88	49.13
CIFAR-10	97.63	82.30	65.75	46.75	51.00
GTSRB	98.63	83.34	97.25	48.63	50.00

TABLE 5: Precision (%) of SOTA detectors against SSDT.

Dataset	**TED**	**Beatrix**	**SCAn**	**STRIP**	**SentiNet**
MNIST	96.90	94.93	89.80	9.52	37.93
CIFAR-10	95.47	77.99	87.95	0	58.70
GTSRB	98.63	75.00	95.22	0	50.00

We further measure the accuracy and precision of TED and other SOTA detectors and report the results in Tables 4 and 5. From these tables, both STRIP and SentiNet exhibit about 50% detection accuracy (*i.e.*, random guess), failing to detect SSDT backdoor totally. Moreover, SCAn is inferior to Beatrix and TED, since it is designed for resisting source-specific but trigger-static backdoors, and SSDT violates its trigger strategy. Beatrix suffers from the issue of high false positives (low precision as shown in Table 5) under SSDT, making its detection accuracy 13% lower than TED on average.

6.2. Evaluating TED Against Diverse Attacks

Noted that SCAn is customized to detect a source-specific backdoor called TaCT (targeted contamination attack) [40], and Beatrix is tailored to defeat a trigger-dynamic backdoor [29]. We assess TED against these attacks and their variants. The results are reported on CIFAR-10 and GTSRB as customized detectors demonstrate 100% TRP and 0% FPR on MNIST. For the same reason, we do not include the result of source-agnostic or static-trigger attacks [7, 10, 25, 49].

Dynamic-input backdoor attack [29]. This attack works under the same threat model as SSDT. It co-optimizes an encoder-decoder network g together with the to-be-backdoored model f to generate dynamic trigger-carrying samples (Eq. 1). Beatrix enlarges the subtle difference between normal and dynamically-triggered samples in Euclidean space by using the Gramian matrix to record their latent features' correlation and their high-order statistics.

We use the code provided by [29] and report the detection results in Table 6. From the results, it is clear that TED is comparable to Beatrix in resisting traditional trigger-dynamic backdoor attacks. As shown in Fig. 11 of Appendix-A, the box plot of the topological feature vectors under this attack [29] also supports this conclusion.

TABLE 6: Detection performance against the attack in [29].

	Dataset			
	CIFAR-10		GTSRB	
	TED	**Beatrix**	**TED**	**Beatrix**
TPR (%)	91.60	99.00	98.00	99.80
Precision (%)	99.34	95.40	100.00	99.80
F1 (%)	95.31	97.20	98.90	99.80

Targeted contamination attack [40]. TaCT is the first to explicitly use the source-specific trigger strategy to disperse the predominant effect of the trigger, while such an effect is the key enabler for detectors like Neural Cleanse [44], SentiNet [8] and STRIP [9]. In particular, TaCT is implemented by stamping the same static trigger (*e.g.*, a small square) on the samples from the victim source class to prepare the backdoor dataset D_b (as in Eq. 7) and on the samples from the non-victim class to prepare the laundry dataset D_l (as in Eq. 8). And SCAn is customized to address this challenge by observing that, after conducting decomposition on benign and malicious samples, their features are separable in terms of the first moment.

We follow the same settings used in TaCT to launch this attack, and the detection performance of SCAn and TED are listed in Table 7. From this table, it is evident that TED demonstrates comparable performance to SCAn on TPR and FPR. We further note that as mentioned in [26, 40], SCAn requires to discern about 50 VT samples before it can reliably detect further samples, meaning that in the online setting, SCan would likely miss the first dozens of VT instances. TED does not suffer from this cold-start problem.

TABLE 7: Detection False Positive Rate (%) against TaCT.

	Dataset			
	CIFAR-10		GTSRB	
	TED	**SCAn**	**TED**	**SCAn**
95% TPR	0.75	0.47	0	0.32
99% TPR	2.00	0.48	0.90	1.10

Fig. 10 in Appendix-A further depicts the topological feature vectors under TaCT.

Composite backdoor attack [21]. It is a variant of the source-specific attack. It differs from TaCT in how the backdoor dataset D_b (Eq. 7) and the laundry dataset D_l (Eq. 8) are prepared. When preparing D_b, a small area (*i.e.*, a cutout) of the sample from X_s is stamped on itself; when preparing D_l, a cutout of the sample from non-victim classes is also stamped on itself (the *mixer* function in [21]). That is said, the trigger patterns for D_b and D_l are different.

We follow the exact settings employed in [21] and conduct experiments on CIFAR-10. This attack achieves 85.62% accuracy on benign samples and 84.25% attack success rate. TED achieves TPR of 94.73% at FPR of 5%, along with an AUC score of 0.9849, compared to Beatrix with 92.11% TPR at 5% FPR and 0.9688 AUC score.

6.3. Adaptive Attacks on TED

A crucial aspect of evaluating the robustness of a backdoor detector is its resistance to adaptive attackers, who are aware of the defense mechanism and aim to bypass detection. Indeed, under the trigger-dynamic strategy (*but not source-specific*), Beatrix [26] investigates adaptive attackers who try to minimize the difference in Gramian features between benign samples and malicious samples when embedding a backdoor. Similarly, under the source-specific strategy (*but not trigger-dynamic*), SCAn [40] studies how to optimize a better trigger that can bypass detection. From this sense, the designed SSDT attack is just an adaptive attack that breaks the assumption of the latent separability in the metric space, used by [26, 40] and many more. So, the question is: **will TED make a real difference when resisting adaptive attacks?**

To answer that question, we first follow the same wisdom from earlier works to design adaptive attacks. That is, the adaptive attacker aims to optimize for a better attack by designing an appropriate loss function other than the original backdoor learning loss $\mathcal{L}_o$. Given the non-invertible and non-differentiable nature of our topological feature extraction method, the attacker cannot directly optimize the topological features for malicious samples as before. Alternatively, the attacker studies loss functions that can allow malicious samples to mimic the topological features of clean samples.

In this regard, we formulate the following three heuristic loss designs:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_o + \lambda_1 \mathcal{L}_1,$$
$$\mathcal{L}_{\text{total}} = \mathcal{L}_o + \lambda_2 \mathcal{L}_2,$$
$$\mathcal{L}_{\text{total}} = \mathcal{L}_o + \lambda_3 \mathcal{L}_3.$$

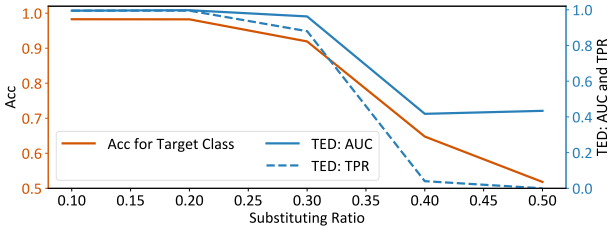


Figure 7: Accuracy, AUC, and TPR v.s. substituting ratio.

Here, $\mathcal{L}_1$ is inspired by the distance metric learning for large margin nearest neighbor classification [45], which is designed to minimize the Euclidean distance between poisoned activations within the same class while maximizing it between different classes, thus mimicking the topological feature of a benign sample. $\mathcal{L}_2$ is inspired by the K-means algorithm [27], which is designed to reduce the distance from a given point (the poisoned activation) to the geometric centroid of the target class. And $\mathcal{L}_3$ is inspired by the latest advancement of NLP backdoor [18]. In this case, the trigger-carrying samples are forced to match the clean samples from the target class only at the first few layers, with the aim of making topological features associated with deeper layers appear normal. The formal expressions of these three loss functions are given in Appendix-B.

We implement the above three adaptive attacks on MNIST with a poison rate 2% and $\lambda_i = 1$ ($i \in [1, 3]$) using ResNet-18[2]. After successfully reducing the loss $\mathcal{L}_{\text{total}}$ after 10,000 iterations, the model's accuracy and attack success rate remain high, sustaining levels above 99%. Surprisingly, TED demonstrates remarkable resilience against these three adaptive attacks, achieving an AUC of 0.99 and a TPR of 100%. We attribute this to that neither of the three loss functions $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ can truly break the feature separability in the topological space since it captures the very nature of benign and malicious samples. As an example, we empirically observe, after applying $\mathcal{L}_3$, the topological features exhibit distinctive dynamics in the middle layers.

Given the complexities associated with manipulating the loss function, we turn to the strategy of manipulating the training data. In this approach, a certain percentage (*i.e.*, the substitution ratio) of the clean samples from the victim class is re-labeled as the target. This equals replacing the training data with specific dirty/noisy labels, which has been extensively studied in the ML community [36]. The rationale of this adaptive attack is that samples with dirty labels will gather around the trigger-carrying samples, creating an innocent-looking topological feature as it propagates.

We conduct the experiment using the same setting as above, and the results are depicted in Fig. 7. Despite the substitution, the model maintains a stable attack success rate of 99.5%, even as the substitution ratio increased. The rise in the substitution ratio directly correlates with a decline in the accuracy of benign samples from the target class, caused by the dirty labelling mechanism. In the worst case, when

2. We exclude the usage of shallow models since the rationale of $\mathcal{L}_3$ is incompatible with shallow models.

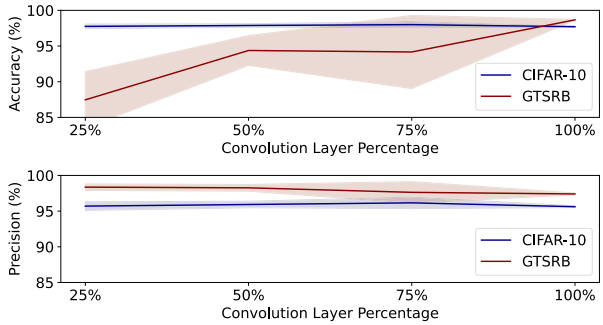


Figure 8: Accuracy (top) and precision (bottom) with various percentages of convolution layers in TED.

50% of the samples from the target class are mislabeled, the TPR of TED is only 50% (random guessing). But the model accuracy for this target class is only 50%, which is also useless. Another observation is that TED maintains 100% TPR until the substitution ratio exceeds 20%. We argue this does not pose a real threat to TED since deliberately dirty-labeling over 20% of one class in the training data can be easily spotted by many off-the-shelf methods [36].

6.4. Ablation Studies of TED

At the heart of TED implementation is the layer-wise embedding, from which we extract the topological feature vector. As mentioned at the beginning of Sec. 6, we take all embeddings output by the Conv2D layers by default. Here, we further conduct a series of ablation studies to explore the properties of TED. All the experiments are implemented under CIFAR-10 and GTSRB, with neural architecture varying from shallow models (*e.g.*, 4-layer CNN) to deeper ones (*e.g.*, PreAct-ResNet18), attacked by SSDT by default.

Layer-reduced variant of TED. Though it is clear that TED will be more efficient than other online detectors like Beatrix [26], SCAn [40], and STRIP [9] due to its simplicity in feature extraction, it is always beneficial to squeeze efficiency in real applications. For this purpose, we conduct a test by randomly sampling some Conv2D layers from PreAct-ResNet18 (measured in percentage), and depict the accuracy and precision of the layer-reduced version TED in Fig. 8. It is observed that different layer-reduced versions maintain high accuracy (mostly over 90%) and precision (over 95%), and it might be possible that the efficiency of TED can be further improved.

Effect of layers other than Conv2D. We also add the ReLU and Linear layers for building the topological features, and the detection AUC of TED is reported in Table 8. For deep networks like PreAct-ResNet18, the detection AUCs are similar to each other under different settings, regardless of the attack method (*i.e.*, SSDT or Composite backdoor [21]). For shallow networks like 4-layer CNN, adding more layers does boost the detection performance. We attribute this to the fact that the Conv2D operator in shallow networks is not trained to have strong feature extraction capability [28], making the topological features from these embedding spaces less discriminative. As such, we should use the full layers for building TED if the network is shallow. We delay the discussion of detecting SSDT backdoor for network without Conv2D layers to Appendix-C.

TABLE 8: AUC of TED on CIFAR-10 with various layer combinations.

Attack	Model	C	C & R	C & R & L
SSDT	PreAct-ResNet18	.9953	.9953	.9954
Composite [21]	PreAct-ResNet18	.9494	.9766	.9852
Composite [21]	4-layer CNN	.7422	.9466	.9849

Notes: C for Conv2D, R for ReLU, and L for Linear.

6.5. Further Evaluation on Other Tasks, Datasets, Models, and Attacks

We further investigate TED's performance on more complex datasets and tasks, including advanced backdoor attacks in NLP tasks [48].

First, we test TED on the PubFig dataset and a subset of the ImageNet dataset with VGG16 as the target model [35]. The PubFig dataset presents a challenging scenario with 83 classes of 64×64 cropped facial images, emulating a facial recognition environment [16]. The subset of ImageNet includes 100 randomly selected classes, each with 500 images, following the experimental settings in [20, 26]. On VT samples, SSDT can achieve 92.5% accuracy in PubFig and 97.5% in ImageNet, with over 99% accuracy on NoT samples in both datasets.

We extract topological features from the output of the Conv2D, ReLU, and Linear layers, using 200 inputs per label as the training set. TED shows promising results with a 0.9296 AUC on PubFig and 0.9972 on ImageNet. Additionally, in terms of the exploration of generalizability, when TED is trained on a limited subset with 200 images from 10 ImageNet classes, it still retains efficacy when tested on 10 entirely different classes, achieving an AUC of 0.9816, TPR of 93.5%, and FPR of 3.5%. Similarly, we box-plot the topological feature vectors for the NoT, NVT, and VT samples on these two datasets, and the result is shown in Fig. 12 of Appendix-A.

Next, we go beyond the easier cases of feedforward networks with a clear concept of layers, and evaluate TED on more complex model architectures. In particular, we assess the performance of TED against backdoor attacks on BERT-based models in NLP applications.

Due to the architectural difference of the transformer, we register forward hooks to record activations at the dense layer, self-attention layer, and word embedding layer. And then we extract the topological feature vectors as usual (see Sec. 5.2) with one exception: instead of recording the ranking of the nearest neighbor from the predicted class in each hook, we record the rankings of the k-nearest neighbors. We define the normalized version of k as the radius $r = \frac{k}{c \cdot m} * 100\%$. As before, c is the number of classes in the classification task, and m is the number of benign samples in each class.

TABLE 9: Performance of TED on toxicity detection over Twitter data with various radius r.

Attack	Metric	Radius r				AVG
		0.50%	1.00%	1.50%	2.00%	
EP	TPR (%)	92.50	92.00	92.00	92.00	92.12
	FPR (%)	20.50	19.00	21.50	23.00	21.00
	AUC	.9309	.9267	.9296	.9280	.9288
DFEP	TPR (%)	95.00	95.00	94.00	94.00	94.50
	FPR (%)	21.00	18.00	19.00	18.00	19.00
	AUC	.9565	.9528	.9545	.9527	.9541

TABLE 10: Performance comparison between TED, DAN, and STRIP for toxicity detection.

Attack	Metric	**TED**	**DAN**	**STRIP**
EP	TPR (%)	92.12	93.42	94.99
	FPR (%)	21.00	30.09	66.11
DFEP	TPR (%)	94.50	93.54	94.99
	FPR (%)	19.00	21.39	48.89

We then evaluate TED against the embedding poisoning (EP) and data-free embedding poisoning (DFEP) backdoor attacks proposed recently in [48]. Instead of using conventional backdoor attacking methods, this kind of attacks modify a single-word embedding vector, either with data (*i.e.*, EP method) or without data (*i.e.*, DFEP method), to implant backdoor. In EP, malicious samples are constructed following the method presented in BadNets [10] but updating only the word embedding weight for the trigger word during backpropagation. In DFEP, poisoning is conducted in a smaller sentence space derived from a general text corpus containing all human-written natural sentences. In our experiments, we sample sentences from the WikiText-103 corpus to create fixed-length fake samples and randomly insert the trigger word into these samples, forming a fake poisoned dataset. We use the code shared by the authors to train backdoored BERT models for toxic input classification on Twitter data.

We implement TED with the above backdoored BERT models, and compare the detection performance of TED with other detectors, including DAN [6] and STRIP [9]. Table 9 reports the performances of TED with various r, demonstrating an average AUC of 0.9288 and 0.9541 against EP and DFEP attacks, respectively. Moreover, as shown in Table 10, TED performs significantly better than DAN [6] and STRIP [9] in terms of FPR while maintaining the same level of TPR.

7. Discussion, Limitations, and Future Directions

Limitations. Our TED is meant to detect backdoor samples for an infected model in a blind scenario, where the attack pattern and even whether the model is infected or not are unknown. To this end, we approach the problem through an anomaly detection manner. In particular, TED builds an outlier detector upon a clean sample set. Within this set, a small percentage (α) of the normal samples are identified as outliers by automatically computing an anomaly score threshold τ, and then τ is used for detecting backdoor samples. As a result, the subsequent detector will also likely cause some false-positive cases by treating some benign samples as outliers. Especially when the model is not backdoored (uninfected case), it will also reject about α percentage of the normal inputs.

To understand such a security-utility trade-off, we conducted experiments to investigate how many percentages of normal inputs are required to be rejected under the uninfected case to keep an acceptable TPR for the infected case. The results (see Fig. 9 in Appendix-D) show when setting the parameter α ranging from 1% to 5%, the TPRs in the infected cases under all three datasets are all above 92.5%. Furthermore, other than setting the hard parameter α, we adopt the Z-score-based method, a more advanced approach to determine a robust reject threshold regardless of whether the model is infected or not. The threshold τ is computed by setting to reject not strictly α percentage of outliers, but the samples that exceed four standard deviations ($4 \times \sigma$) from the mean of the feature distribution under Gaussian modeling. This method results in a TPR of 92.4% on the infected model. Notably, the uninfected model exhibits an FPR of only 0.7%, which is fairly acceptable.

Fundamentally, TED leverages the topological evolution dynamic to build the detector, whose working principle relies on the assumption that backdoor samples should have a drastically different trajectory relative to the benign samples of target label t. In our evaluation, this assumption seems to hold in general, even under some very sophisticated attacks. In particular, with our proposed SSDT attack that combines the hard-to-detect properties of existing attack approaches, the backdoor samples still show expected irregularities in their trajectory. Furthermore, despite designing multiple adaptive attacks to challenge this assumption, none have succeeded. However, we acknowledge that future research may reveal novel attacks that invalidate this assumption if they successfully render the activations of backdoor samples fall within the range of benign samples of the target label t across the whole network.

Future work. In this work, we propose to utilize the evolution dynamic for backdoor sample detection. Specifically, TED captures the evolution dynamics by analyzing the rank sequence of the nearest samples belonging to the predicted class. This approach has demonstrated remarkable performance in countering existing attack methods thus far. However, we believe that further advancements can be made by improving the modeling of the evolution dynamics. Merely relying on the rank of the nearest sample from the predicted class as a measure of "closeness" is a straightforward approach but may not be optimal. Therefore, future research could explore alternative methods to assess the proximity of an input sample to the distribution of samples from the target class at each layer. By examining the "closeness" in this manner, we anticipate that the performance of our approach could be further enhanced.

8. Conclusion

Backdoor attacks present a critical security threat to the DNN model supply chain, and many defensive mechanisms have been proposed to address this threat. Our work identified that existing backdoor detectors' success relies on the assumption of trigger strategy such that the latent representations of benign and malicious samples can be separated in the metric space. Experimental results confirmed that SOTA backdoor detectors built on this assumption are ineffective against our proposed SSDT, a sophisticated backdoor attack that blends both the source-specific and the dynamic-trigger strategies. To overcome this limitation, we turned our attention to the analysis in the topological space and proposed TED, capitalizing on the inherent differences in the evolutionary dynamics of topological structures between benign and malicious samples. The effectiveness of TED is corroborated through extensive experimental results on CV and NLP tasks. By adopting a novel perspective that considers the topological structure, this work represents a significant step forward in understanding and mitigating backdoors in deep learning.

Acknowledgement

Shengshan's research is supported in part by the National Natural Science Foundation of China under Grant No. 62372196, 62202186 and U20A20177.

References

[1] The cifar-10 dataset. https://www.cs.toronto.edu/~kriz/cifar.html.

[2] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *USENIX Security*, pages 1615–1631, 2018.

[3] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. In *Usenix Security*, 2021.

[4] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. *arXiv preprint arXiv:2106.09667*, 2021.

[5] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

[6] Sishuo Chen, Wenkai Yang, Zhiyuan Zhang, Xiaohan Bi, and Xu Sun. Expose backdoors on the way: A feature-based efficient defense against textual backdoor attacks. *arXiv preprint arXiv:2210.07907*, 2022.

[7] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[8] Edward Chou, Florian Tramèr, Giancarlo Pellegrino, and Dan Boneh. Sentinet: Detecting physical attacks against deep learning systems. *arXiv preprint arXiv:1812.00292*, 4, 2018.

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

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

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *14th European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[12] Shengshan Hu, Wei Liu, Minghui Li, Yechao Zhang, Xiaogeng Liu, Xianlong Wang, Leo Yu Zhang, and Junhui Hou. Pointcrt: Detecting backdoor in 3d point cloud via corruption robustness. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 666–675, 2023.

[13] Shengshan Hu, Ziqi Zhou, Yechao Zhang, Leo Yu Zhang, Yifeng Zheng, Yuanyuan He, and Hai Jin. Badhash: Invisible backdoor attacks against deep hashing with clean label. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 678–686, 2022.

[14] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *International Conference on Learning Representations*, 2022.

[15] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2043–2059. IEEE, 2022.

[16] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th international conference on computer vision*, pages 365–372. IEEE, 2009.

[17] Yann LeCun, Lawrence D Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276):2, 1995.

[18] Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. Backdoor attacks on pre-trained models by layerwise weight poisoning. *arXiv preprint arXiv:2108.13888*, 2021.

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

[20] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16463–16472, 2021.

[21] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 113–131, 2020.

[22] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings 21*, pages 273–294. Springer, 2018.

[23] Xiaogeng Liu, Minghui Li, Haoyu Wang, Shengshan Hu, Dengpan Ye, Hai Jin, Libing Wu, and Chaowei Xiao. Detecting backdoors during the inference stage based on corruption robustness consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16363–16372, 2023.

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

[25] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*, 2018.

[26] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "beatrix" resurrections: Robust backdoor detection via gram matrices. In *NDSS*, 2023.

[27] J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.

[28] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. When and why are deep networks better than shallow ones? In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[29] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

[30] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[31] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, 51(5):1–36, 2018.

[32] Erwin Quiring and Konrad Rieck. Backdooring and poisoning neural networks with image-scaling attacks. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 41–47. IEEE, 2020.

[33] Peter J Rousseeuw and Mia Hubert. Anomaly detection by robust statistics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(2):e1236, 2018.

[34] Giorgio Severi, Jim Meyer, Scott E Coull, and Alina Oprea. Explanation-guided backdoor poisoning attacks against malware classifiers. In *USENIX Security Symposium*, pages 1487–1504, 2021.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[36] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

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

[38] Iwan Syarif, Adam Prugel-Bennett, and Gary Wills. Unsupervised clustering approach for network anomaly detection. In *Networked Digital Technologies: 4th International Conference, NDT 2012, Dubai, UAE, April 24-26, 2012. Proceedings, Part I 4*, pages 135–145. Springer, 2012.

[39] Bo Tang and Haibo He. A local density-based approach for outlier detection. *Neurocomputing*, 241:171–180, 2017.

[40] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection. In *USENIX Security Symposium*, pages 1541–1558, 2021.

[41] Guanhong Tao, Yingqi Liu, Guangyu Shen, Qiuling Xu, Shengwei An, Zhuo Zhang, and Xiangyu Zhang. Model orthogonalization: Class distance hardening in neural networks for better security. In *2022 IEEE Symposium on Security and Privacy (SP). IEEE*, volume 3, 2022.

[42] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

[43] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in Neural Information Processing Systems*, 31, 2018.

[44] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

[45] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

[46] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021.

[47] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 103–120. IEEE, 2021.

[48] Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In *ACL*, pages 2048–2058, 2021.

[49] Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16473–16481, 2021.

[50] Qi Zhong, Leo Yu Zhang, Shengshan Hu, Longxiang Gao, Jun Zhang, and Yong Xiang. Attention distraction: Watermark removal through continual learning with selective forgetting. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022.

Appendix A.
Detailed Settings and More Experimental Results

Settings: For the implementation of SSDT, we utilize three datasets MNIST, CIFAR-10, and GTSRB. For MNIST, we employ a 2-layer CNN as shown in Table 11. For CIFAR-10 and GTSRB, we use Pre-activation ResNet18 [11]. In Algorithm 1, the sampling rates $\rho, \rho_b, \rho_l, \rho_{ct}$ associated with no-trigger samples, victim-triggered samples, non-victim but triggered samples, and cross-triggered samples are respec-

TABLE 11: Network Architecture for SSDT on MNIST

Layer	Output Size	Kernel Size	Stride	Activation
Conv2d-1	$32 \times 24 \times 24$	5×5	1	ReLU
Dropout-3	$32 \times 24 \times 24$	-	-	-
MaxPool2d-4	$32 \times 12 \times 12$	2×2	2	-
Conv2d-5	$64 \times 8 \times 8$	5×5	1	ReLU
Dropout-7	$64 \times 8 \times 8$	-	-	-
MaxPool2d-5	$64 \times 4 \times 4$	2×2	2	-
Linear-6	512	-	-	ReLU
Dropout-8	512	-	-	-
Linear-9	10	-	-	-

tively set as $\rho = \frac{c}{c+2}$, $\rho_b = \frac{1}{c+2}$ and $\rho_l + \rho_{ct} = \frac{1}{c+2}$, where c is the number of classes.

For the implementation of TED, we randomly sample 20 clean images for each class from the test sets of MNIST and CIFAR-10 (both with 10 different classes), respectively, and we randomly sample 1,000 clean images in total from the test set of GTSRB (43 different classes). The contamination rate (*i.e.*, FPR) parameter α in Algorithm 2 is set to 5% unless otherwise specified.

More results: Fig. 11 box-plots the topological feature vectors under a dynamic-trigger backdoor attack proposed in [29]. Similarly, Fig. 10 box-plots the topological feature vectors under the source-specific backdoor attack TaCT proposed in [40]. From both figures, it is easy to see that the feature vectors between NoT samples and VT samples are clearly separable. Under complex dataset scenarios, as depicted in Fig. 12, VT samples exhibit more variability on their way towards the predicted class, whereas the NoT and NVT samples consistently approach the predicted class both the PubFig and ImageNet datasets.

Appendix B. Details of Adaptive Attacks

Here, we present the details of the three loss functions for adaptive attacks. These functions serve the purpose of mimicking the topological features of benign samples for adaptive attacks of TED.

As mentioned above, $\mathcal{L}_1$ employs the large margin nearest neighbor classification method [45] to minimize the Euclidean distance between poisoned activations within the same class while maximizing it between different classes, hence mimicking the topological feature of a benign sample. In particular, $\mathcal{L}_1$ is defined as

$$\mathcal{L}_1 = \frac{1}{\#D_b} \sum_{i=1}^{\#D_b} \max\left(0, \min_i(\{d_{T,i}\}) - \min_i(\{d_{O,i}\})\right),$$

where $\#D_b$ is the number of poisoned samples, $d_{T,i}$ and $d_{O,i}$ represent the pairwise Euclidean distances between the i-th poisoned activation and other activations of the target class and other classes, respectively.

$\mathcal{L}_2$ employs the idea of the K-means algorithm [27] to reduce the distance from a given point (the poisoned activation) to the geometric centroid of the target class, which is defined as

$$\mathcal{L}_2 = \frac{1}{\#D_b} \sum_{i=1}^{\#D_b} ||f_{N-1} \circ \cdots \circ f_1(x_i) - c_t||_2,$$

where c_t represents the centroid of the target class and $f_{N-1} \circ \cdots \circ f_1(x_i)$ represents the i-th poisoned activation.

$\mathcal{L}_3$ employs the shallow-layer weight poisoning method in [18] to make topological features associated with deeper layers appear normal. In particular, $\mathcal{L}_3$ is defined as

$$\mathcal{L}_3 = \frac{1}{\#D_b * \#\bar{D}} \sum_{i=1}^{\#D_b} \sum_{j=1}^{\#\bar{D}} ||f_k \circ \cdots \circ f_1(x_i) - f_k \circ \cdots \circ f_1(x_j^t)||_2,$$

where $\bar{D}$ is a small number of samples from the target class, and $f_k \circ \cdots \circ f_1$ are the first k layers of f.

TABLE 12: Detection performance of ShallowNet on MNIST (TPR for VT: 99.50%, FPR for NVT: 2.50%, FPR for NoT: 0.00%).

Metric	Value	Metric	Value
Precision	98.76%	Accuracy	99.13%

Appendix C. Detecting Backdoor in Shallow Networks Without Convolution

We conduct experiments using a backdoored ShallowNet on MNIST. The ShallowNet consists of two fully connected layers. The first layer takes input features of size 1,024 and maps them to an intermediate representation with 128 units. The second layer takes this 128-unit representation and maps it to the final output, which consists of 10 classes, corresponding to the 10 digits in MNIST. The TaCT backdoor method [40] is used, and the backdoored ShallowNet achieves 97.6% accuracy on NoT samples and 100% on VT samples. As discussed in Sec. 6.4, we take the two Linear layers to extract the topological features for TED. As listed in Table 12, we observe that even with only two Linear layers, TED is still able to perform well. The precision of TED is 98.76% and the accuracy is 99.13%.

Appendix D. Detection threshold τ selection.

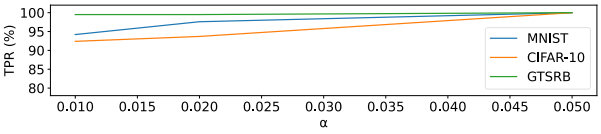
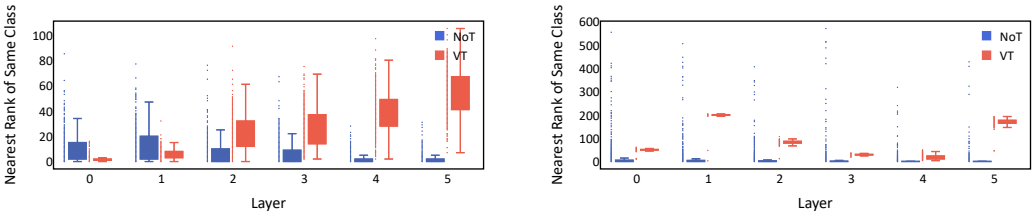


Figure 9: Ablation on the parameter α, with plots of TPR of each α on three datasets.

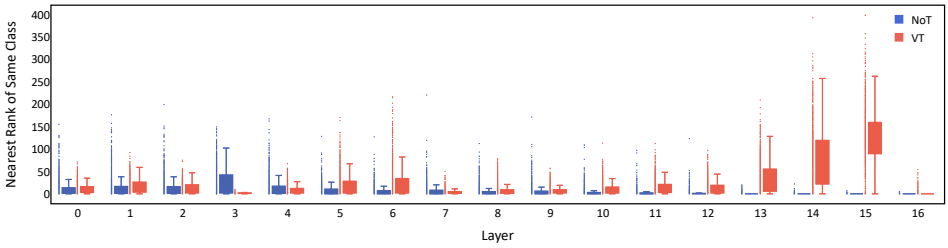
A Z-score-based outlier detection rule is applied to both a benign model and a SSDT-backdoored model on CIFAR-10. Inputs are classified as "positive detection" if the TED anomaly score exceeds four standard deviations ($4\times\sigma$) from the feature distribution mean, setting the threshold τ. This method results in a TPR of 92.4% on the backdoored model, indicating the successful detection of most poisoned inputs with minimal false positives. Notably, the benign model exhibits an FPR of only 0.7%, an acceptable rate demonstrating the method's precision.

The ablation study on the reject parameter α, ranging from 1% to 5%, reveals that all TPRs across the three datasets exceed 92.5%, as illustrated in Fig. 9.

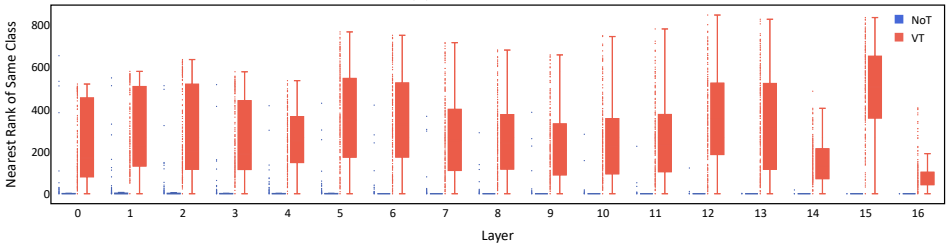


(a) CIFAR-10 (b) GTSRB

Figure 10: Box plot of topological feature vectors under TaCT [40].

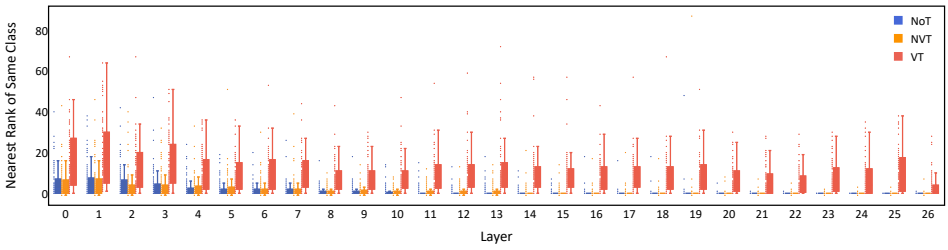


(a) CIFAR-10

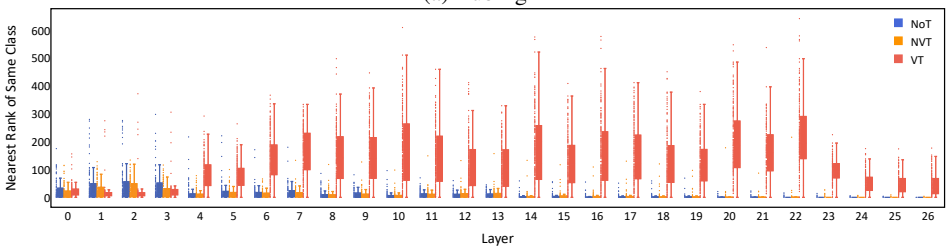


(b) GTSRB

Figure 11: Box plot of topological feature vectors under the dynamic trigger attack [29].



(a) PubFig



(b) ImageNet

Figure 12: Box plot of topological feature vectors on Pubfig and ImageNet under SSDT.

Appendix E.
Meta-review

E.1. Summary

This paper studies approaches for detecting backdoor inputs to models with poisoned training data. It proposes two novel methods: first, a source-specific dynamic trigger backdoor attack that outperforms the existing state-of-the-art input-level defenses; second, a method that leverages the topological evolution of an input through progressive network layers to distinguish backdoor trigger inputs from benign ones.

E.2. Scientific Contributions

- Addresses a Long-Known Issue
- Provides a Valuable Step Forward in an Established Field

E.3. Reasons for Acceptance

1) The proposed approach differs from prior ones in novel and interesting ways, and is effective against attacks for which existing defenses fall short.
2) The paper gives a detailed analysis of existing backdoor defenses and their limitations.
3) The experimental evaluation is thorough, considering multiple attack types, adaptive attacks, and various ablations on model type and size.