TED++: Submanifold-Aware Backdoor Detection via Layerwise Tubular-Neighbourhood Screening

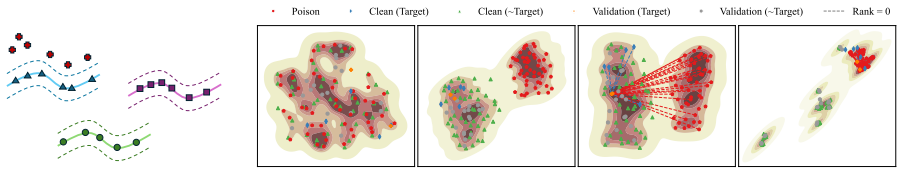
Nam Le[1], Leo Yu Zhang[2], Kewen Liao[1], Shirui Pan[2], and Wei Luo[*1]

[1]School of Information Technology, Deakin University, Australia, {s222576762, kewen.liao, wei.luo}@deakin.edu.au
[2]School of Information and Communication Technology, Griffith University, Australia, {leo.zhang, s.pan}@griffith.edu.au

October 17, 2025

Abstract

As deep neural networks power increasingly critical applications, stealthy backdoor attacks, where poisoned training inputs trigger malicious model behaviour while appearing benign, pose a severe security risk. Many existing defences are vulnerable when attackers exploit subtle distance-based anomalies or when clean examples are scarce. To meet this challenge, we introduce TED++, a submanifold-aware framework that effectively detects subtle backdoors that evade existing defences. TED++ begins by constructing a tubular neighbourhood around each class's hidden-feature manifold, estimating its local "thickness" from a handful of clean activations. It then applies Locally Adaptive Ranking (LAR) to detect any activation that drifts outside the admissible tube. By aggregating these LAR-adjusted ranks across all layers, TED++ captures how faithfully an input remains on the evolving class submanifolds. Based on such characteristic "tube-constrained" behaviour, TED++ flags inputs whose LAR-based ranking sequences deviate significantly. Extensive experiments are conducted on benchmark datasets and tasks, demonstrating that TED++ achieves state-of-the-art detection performance under both adaptive-attack and limited-data scenarios. Remarkably, even with only five held-out examples per class, TED++ still delivers near-perfect detection, achieving gains of up to 14% in AUROC over the next-best method. The code is publicly available at https://github.com/namle-w/TEDpp.



(a) (b)

Figure 1: **(a)** Conceptual model of three class submanifolds and their tubular neighbourhoods, with poisoned samples (red) lying just outside the tube yet still having nearest clean neighbours in ambient space. **(b)** UMAP projections of CIFAR-10 activations in four ResNet-18 layers under a Trojan backdoor attack. The shaded contours indicate the estimated tubular neighbourhood around the clean-class submanifold. **Phase 1:** All points—clean and poisoned—lie well inside the tube; ranks are indistinguishable. **Phase 2:** Poisoned points begin to drift off the clean manifold (outside the tube), but TED's rank-based test fluctuates and fails to separate them from clean samples. **Phase 3:** Most poisoned samples cross the tube boundary yet still attain the lowest rank 0, showing TED cannot penalise off-tube departures. **Phase 4:** Finally, clean and poisoned trajectories reconverge in the ambient space, rendering TED powerless.

1 Introduction

Deep neural networks (DNNs) are having a transformative impact on society and have been applied in numerous important domains, such as computer vision [23]. However, their increasing complexity and widespread use have made them vulnerable to significant risks. In particular, backdoor attacks pose a serious threat. In such attacks, an adversary inserts a hidden trigger into the model via poisoned samples during training, causing it to misclassify inputs containing the trigger as target classes, while still correctly classifying clean inputs. These attacks can be devastating, threatening the sustainability and security of the deep learning supply chain, especially when end users have limited control over the training process [11, 50].

According to the application stage in the DNN model lifecycle, defence methods against backdoor attacks currently include data purification [41, 21, 17], poison suppression [47, 16, 40], model-level detection [44, 49, 45] and mitigation [28, 52, 8], and input-level detection [3, 29, 9]. Among these, input-level methods have attracted significant interest because they provide the most fine-grained level of security, that is, determining whether an individual input sample is poisoned or benign/clean, while also safeguarding suspicious models that are already deployed.

Previous defences like STRIP [4], TeCo [29], and SCALE-UP [9] introduce different techniques to transform each image input and monitor how consistent the output is, thereby distinguishing poisoned and clean samples. With the

*Corresponding author: wei.luo@deakin.edu.au

rapid evolution of new backdoor attacks, these defence methods are increasingly insufficient. In response, IBD-PSC [13] builds on traditional methods by replacing input amplification with activation amplification at multiple batch normalisation layers, allowing it to monitor inconsistencies in model output, as measured by entropy. A more recent and robust advancement in this category is Topological Evolution Dynamics (TED) [32]. Rather than relying on static transformations (e.g., STRIP [4], TeCo [29], SCALE-UP [9]) or metric-based distances computed at a few individual network layers (e.g., IBD-PSC [13]), TED captures the dynamic evolution of an input as it propagates through the network. It leverages robust topological neighbour relationships and introduces a ranking-based technique, which at each layer assesses a sample's position as rank k where the sample's k-nearest neighbour in a validation set has the sample's target class. The core idea is that benign samples will maintain stable target-class neighbour rankings across all layers, whereas poisoned samples exhibit unstable rankings in the intermediate layers, even though they will ultimately be classified as the target class.

Although TED shows promise in defending against sophisticated backdoor attacks, it suffers a fundamental breakdown when its test relies on raw rank statistics (based on identifying the top k nearest neighbours) in the full ambient feature space. In high-dimensional settings, vast "empty" volumes around the low-dimensional data manifold mean that a poisoned activation can drift off-manifold yet still find itself as "nearest" neighbour to some distant validation point (See Fig 1). The well-known distance concentration phenomenon [43] further collapses the gap between nearest and farthest distances. This pathology is amplified as the validation budget shrinks. Experimental results highlight this limitation (see Fig 3 and Sec 4 for details), indicating that TED fails to detect anomalies effectively and achieves an AUROC score of less than 0.7 when the validation dataset contains fewer than 100 benign samples on CIFAR-10.

To understand why raw rank statistics break down, we traced how clean and poisoned activations evolve through the network. By visualising ResNet-18 features at each layer via UMAP—which faithfully preserves both local neighbourhoods and global structure—we see that clean examples tightly hug a low-dimensional submanifold, while poisoned points gradually drift off, only to later reconverge in ambient space (Fig 1). This drift-and-reconverge pattern explains why TED's "nearest-neighbour" rank test fails (particularly with challenging backdoor attacks such as Ada-Patch, Ada-Blend [35], and Trojan [30]): without explicit knowledge about the true manifold, off-manifold points can masquerade as neighbours once distances concentrate.

Motivated by these geometric failure modes, we introduce TED++, which exploits the critical information in the submanifold structure by constructing a thin, locally estimated tube around each class's hidden-feature manifold. By ranking activations relative to that tube boundary—rather than against all ambient-space points—TED++ achieves robust, noise-resilient rank statistics that reliably flag off-manifold (poisoned) inputs even under extreme validation-data scarcity.

We evaluate our method on benchmark datasets CIFAR-10 [18], GTSRB

[37], and TinyImageNet [19] and consistently observe improved performance over various backdoor attack scenarios, even with a few validation data points.

1. A submanifold-centric view of backdoor detection. We formalise the intuition that clean-class activations form low-dimensional submanifolds in hidden layers. This insight reveals that existing nearest-neighbour rank defences (e.g., TED [32]) ignore this geometric structure and therefore fail to flag subtle, off-submanifold anomalies. On the other hand, explicit modelling of each submanifold and its tubular neighbourhood [20] yields a robust, noise-resilient signal for backdoor defence.

2. TED++: Tube-aware screening with locally adaptive ranking. Building on the manifold geometry, we introduce TED++, which (a) estimates a layerwise tubular neighbourhood (or tube) around each class's submanifold, which could be just based on a handful of clean validation examples, (b) applies Locally Adaptive Ranking **(LAR)** to assign worst-case ranks to activations lying outside these estimated tubes. TED++ delivers state-of-the-art backdoor detection performance against an extensive list of attacks.

2 Related Work

2.1 Backdoor Attacks

Backdoor attacks on neural networks involve adversaries injecting malicious triggers into models to manipulate their behaviour during inference. Generally, existing approaches can be divided into three main categories according to the attackers' capabilities: **(1)** data poison-only attacks, **(2)** training-controlled attacks, and **(3)** model-controlled attacks.

Poison-only attacks refer to scenarios where attackers have no control in the training process but can manipulate the training data by inserting triggers into a part of the dataset. A classic approach in this category is BadNets [7], which takes a small white square as the trigger and superimposes the trigger onto some randomly selected clean images, whose labels are then modified to a specific target class. Training on such poisoned datasets can make models learn a correlation between the injected trigger and the target class. Subsequent research has expanded to attacks that implant more sophisticated triggers that make poisoned samples harder to detect [38, 35], along with clean-label attacks [42, 53, 5] where the original labels of the poisoned samples are preserved to avoid detection, and physical attacks that leverage tangible objects or geometric transformations as triggers [48, 6].

Training-controlled attacks allow attackers to manipulate both the training data and the training process. These approaches often attempt to evade detection through sophisticated strategies such as introducing "noise modes" [33, 34, 32, 54] or special training regularisation techniques [2]. Another approach in this category focuses on enhancing attack effectiveness [46, 24, 55], using advanced learning frameworks that go beyond standard supervised learning to embed activations in a sophisticated manner without degrading model perfor-

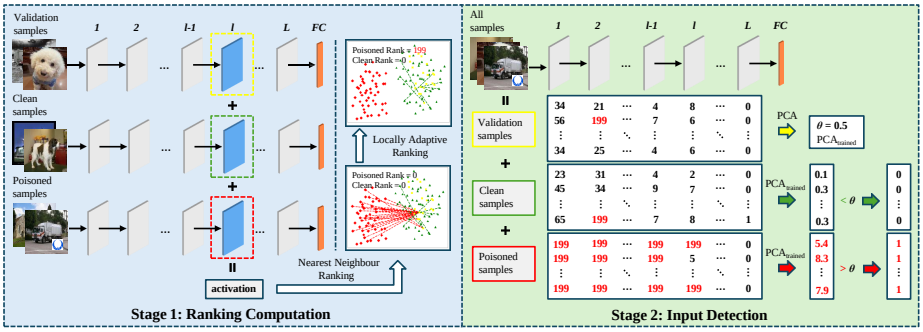


Figure 2: TED++ pipeline. **Stage 1 (Ranking Computation):** For each layer ℓ, we extract activations $h^{(\ell)}(x)$, estimate the tubular-neighbourhood radius τ_ℓ around each class submanifold using clean validation activations, and apply Locally Adaptive Ranking (LAR) by assigning worst-case ranks only to off-tube activations while retaining natural nearest-neighbour ranks for on-tube activations. **Stage 2 (Input Detection):** Each sample is represented by its L-dimensional rank trajectory $R(x) = [R_1(x), \ldots, R_L(x)]$. A PCA model trained on clean-validation trajectories learns the characteristic "tube-constrained" behaviour; at test time, samples whose reconstruction error exceeds the threshold θ are flagged as poisoned.

mance.

Model-controlled attacks represent scenarios where attackers directly manipulate model architecture or parameters rather than data. An example approach involves injecting additional malicious modules into clean models [39], while other methods explore parameter-level modifications directly to implant hidden backdoors [36]. Though recent studies have explored beneficial uses of backdoors, such as embedding watermarks or protecting intellectual property [22, 25, 26, 10, 40, 51], such applications are beyond the scope of this work.

In addition to these three capability-based categories, backdoor attacks may also be classified according to (a) the trigger type—whether it is a static [7, 1] or a dynamic [33, 32] pattern that adapts at inference time and (b) the source–target relationship, whether the trigger affects all source classes [30, 34] (source-agnostic) or only one or a few specific source classes [32, 38] (source-specific). Collectively, these various attack strategies demonstrate the growing complexity of backdoor attacks, underscoring the urgent need to develop robust backdoor detection methods capable of adapting to a wide range of adversarial settings.

2.2 Backdoor Defences

Existing backdoor defence methods can be broadly categorized based on their application stage in the model lifecycle: **(1)** data purification [41, 21, 17, 15], **(2)** poison suppression [47, 16, 40], **(3)** model-level detection [44, 49, 45], **(4)** model-

level mitigation [28, 52, 8, 56, 14], and **(5)** input-level detection [3, 29, 9]. Data purification techniques aim to remove poisoned samples from training datasets by analysing each sample's influence after model training. Poison suppression approaches alter the training process to keep models from learning backdoor triggers in poisoned samples. Model-level detection approaches often train a meta-classifier or try to replicate activation patterns to evaluate whether a model contains hidden backdoors. Model-level mitigation solutions seek to eliminate or neutralise backdoors built into trained models (see [31] for a comprehensive survey).

Input-level backdoor detection (IBD), the main focus of this paper, functions as a firewall for deployed models by detecting and preventing poisoned inputs at inference time. Compared to other techniques, input-level detection is highly computationally efficient, making it well-suited for scenarios involving third-party models or resource-constrained environments.

STRIP [4] is one of the well-known input-level backdoor detection techniques that evaluates the stability of model predictions upon the introduction of benign perturbations to inputs. This approach relies heavily on the assumption that backdoor triggers dominate predictions, ensuring consistency despite perturbations. However, because of this basic idea, STRIP is vulnerable to adaptive backdoor attacks that are specifically built to get around this kind of perturbation-based detection.

In order to detect poisoned inputs, SCALE-UP [9], another well-known IBD technique, amplifies pixel intensities and monitors prediction stability. However, SCALE-UP has natural limitations due to pixel value limits, which limit intensities to a specific range (i.e., [0,255]). Such limitations could unintentionally alter triggered inputs, decreasing the detection accuracy.

More recently, methods like IBD-PSC [13] and TED [32] have become the most effective defences against backdoor attacks. Even with few validation samples, IBD-PSC significantly improves detection accuracy and robustness by introducing a parameter-oriented scaling consistency technique. However, despite its strong performance across diverse attacks, it struggles to defend against source-specific attacks. TED is no exception. While it has also shown promise against a variety of attack categories, it suffers from several significant limitations—most notably, its reliance on global nearest ranking information and a large number of validation samples. These limitations are exposed when dealing with sophisticated attacks in practical scenarios. Additionally, TED was evaluated on only four image-based attacks, SSDT [32], IAD [33], TaCT [38], and the Composite Backdoor Attack [27] (a variant of source-specific attacks) in the previous work [32]. Despite the sophistication of these attacks, characterised by dynamic triggers and source-specific behaviour, with triggers that are conspicuous rather than invisible like WaNet [34], TED's robustness still requires further evaluation.

3 TED++: Submanifold-Aware Backdoor Detection

3.1 Preliminaries

Threat Model. This work tackles the problem of detecting input-level backdoors in a white-box setting with limited computing resources. Following [32, 13], we consider a white-box DNN $\mathcal{F} : \mathcal{X} \to [0,1]^C$ with C classes and L hidden layers (standard convolution–BN–ReLU blocks). We follow the standard poison-only threat model (See [7, 32]): A small fraction of training samples are augmented with trigger $\delta := \tau(x)$, yielding a poisoned set $\mathcal{D}_p$. Defenders have complete access to a suspicious model, but they lack the resources to remove potential backdoors. Following [32], we assume that defenders have access to a limited set of validation samples with at least two validation samples per class.

Defenders' Goals. An ideal input-level backdoor detection solution should accurately detect and remove all malicious input samples while keeping the model's inference speed intact. In short, defenders aim for two main objectives: **(1)** Effectiveness—to reliably determine if a suspicious image is malicious, and **(2)** Efficiency—to integrate seamlessly as a plug-and-play module with minimal impact on inference time.

Topological Evolution Dynamics in DNNs. Topological Evolution Dynamics (TED) [32] offers a novel, robust approach to detect backdoor attacks in deep neural networks (DNNs). Unlike traditional methods that assume separability between benign and malicious samples in a fixed metric space, TED views a DNN as a dynamical system evolving inputs into outputs. Benign inputs follow stable evolutionary trajectories within their natural class neighbourhoods across network layers, whereas malicious samples exhibit anomalous trajectories, initially close to benign classes but shifting distinctly towards attacker-specified target classes deeper in the network. TED effectively leverages these unique topological differences to reliably identify backdoored samples across various neural architectures.

3.2 Motivation and Pipeline Overview

Our method closely relates to the recently introduced SOTA defence method TED [32]. TED++'s core novelty is to (i) reframe each class's clean activations at layer ℓ as lying on a smooth submanifold with a learnable "tube" of thickness τ_l, and (ii) leverage that tube to detect off-manifold behaviour before ranking.

Ambient-Space Neighbour Ranks Overlook Off-Manifold Activations. Through the lens of submanifolds, TED can be interpreted as testing whether a sample's hidden-layer trajectory crosses one of these class submanifolds, by monitoring nearest-neighbour ranks. However, such nearest-neighbour ranks are solely based on ambient-space distances, even when malicious samples lie outside the tubular neighbourhoods (See Fig. 1). This limitation makes TED vulnerable to sophisticated backdoor attacks [16, 17], where poisoned samples may be significantly distant from benign validation samples in the feature space,

Algorithm 1 The overall algorithm of TED++

Require: Deep network f with L layers, $\{\mathcal{V}_c\}_{c=1}^{C}$ clean validation sets, neighbour-percentile β, PCA model PCA$(\cdot)$

Preprocessing:
for $\ell = 1$ to L **do**
    Compute tube radius τ_ℓ from $\{h^{(\ell)}(v) : v \in \cup_c \mathcal{V}_c\}$
Fit PCA on $\{R(v)\}_{v \in \cup_c \mathcal{V}_c}$ and set detection threshold θ
Detection:
for all $x \in X_{\text{test}}$ **do**
    for $\ell = 1$ to L **do**
        $z \leftarrow h^{(\ell)}(x)$
        $v^* \leftarrow \arg\min_{v \in \mathcal{V}_c} \|z - h^{(\ell)}(v)\|_2$
        if $\|z - h^{(\ell)}(v^*)\|_2 > \tau_\ell$ **then**
            $R_\ell(x) \leftarrow |\mathcal{V}|$ ▷ off-tube ⇒ worst rank
        else
            $R_\ell(x) \leftarrow$ rank of v^* in $\mathcal{V}_c$ by distance
    Compute error $e \leftarrow \text{PCA.recon\_error}(R(x))$
    if $e > \theta$ **then**
        Flag x as poisoned

yet still appear as nearest neighbours to some target class validation samples in intermediate layers.

Two-stage Pipeline. To address this limitation, TED++ adopts a two-stage workflow as shown in Fig 2. During the *Ranking Computation Stage* (see Sec 3.3 and Sec 3.4), we explicitly estimate a layer-wise tube radius τ_ℓ from the clean validation activations, capturing the local "thickness" of each submanifold. We then introduce Locally Adaptive Ranking (LAR): at each layer, only those activations that lie outside the healthy tube are assigned the worst possible rank, while those inside retain their natural neighbour-order. By aggregating these adjusted ranks across all layers in the *Input Detection Stage* (see Sec 3.5), we focus the detection on genuine backdoor-induced departures from the submanifold rather than benign fluctuations in manifold geometry. After collecting the ranking sequence of input x across the considered layers, TED++ leverages a PCA model trained on a benign validation set to predict a final score for input x. This score is expected to be much higher for poisoned samples and significantly lower for benign ones. This submanifold-aware screening endows TED++ with enhanced robustness to small, on-manifold variations while retaining sensitivity to true off-manifold (backdoor) perturbations. Alg 1 illustrates in detail the overall procedure of TED++.

3.3 Tubular-Neighbourhood Modelling

We denote by $h^{(\ell)}(x) \in \mathbb{R}^{d_\ell}$ the activation of input x at layer ℓ. Let $\mathcal{C} = \{1, \ldots, C\}$ be the set of classes and $\mathcal{V}_c$ a small clean validation set for class c.

We make the following assumptions and definitions:

Ideal Class Submanifolds. For each layer ℓ and class c, we imagine a (smoothly embedded) submanifold

$$M_c^{(\ell)} \subset \mathbb{R}^{d_\ell} \tag{1}$$

capturing the true geometry of class-c activations. These $M_c^{(\ell)}$ are conceptual objects under the manifold hypothesis; different classes may share the same submanifold in some layers.

Observed Validation Activations. In practice, we only have access to a finite number of clean validation samples $v \in \mathcal{V}_c$ and their associated activations at layer l, $h^{(\ell)}(v)$, to approximate the geometry of $M_c^{(\ell)}$:

$$H_c^{(\ell)} = \{h^{(\ell)}(v) \mid v \in \mathcal{V}_c\} \subset M_c^{(\ell)}. \tag{2}$$

Layer-wise Tubular Neighbourhoods. Around each submanifold $M_c^{(\ell)}$, define a tubular neighbourhood of radius τ_ℓ:

$$\mathcal{T}_c^{(\ell)}(\tau_\ell) = \{\, z \in \mathbb{R}^{d_\ell} \mid d\big(z,\, M_c^{(\ell)}\big) \le \tau_\ell \}, \tag{3}$$

where $d(z, M) = \inf_{v \in M} \|z - v\|_2$. We choose τ_ℓ so that $H_c^{(\ell)} \subset \mathcal{T}_c^{(\ell)}(\tau_\ell)$.

By computing each layer's tubular-neighbourhood radius τ_ℓ as the maximum "thickness" of benign activations, we obtain a robust anomaly detector: any input whose layer-ℓ activation falls outside this tube is flagged as suspicious.

Finally, since the ideal submanifold distance is not available in closed form, we approximate each layer–ℓ tube radius by measuring the spread of layer–ℓ activations of a class c around a test point x. Concretely, let $\mathcal{V}_c$ denote the set of all validation samples of class c. We locate the layer–ℓ $m\beta$ nearest neighbours of the test activation $h^{(\ell)}(x)$ in $\mathcal{V}_c$ and set the tubular radius τ_ℓ to the maximum of their distances:

$$\max_{1 \le c \le C} \{\, d\big(h^{(\ell)}(v), h^{(\ell)}(v')\big) \mid v, v' \in \mathrm{kNN}^{(\ell)}_{\lceil m\beta \rceil}\big(x; \mathcal{V}_c\big) \}, \tag{4}$$

where $\mathrm{kNN}^{(\ell)}_{m\beta}(x;\, \mathcal{V}_c)$ returns the test point x's $\lceil m\beta \rceil$ nearest class-c neighbours by Euclidean distance in layer–ℓ activations (with m as the number of validation samples per class and β as the neighbour percentile factor). This adaptive threshold ensures each tube's "thickness" reflects the local density of benign activations in embedding space.

3.4 Locally Adaptive Ranking

Having estimated a layer-wise tube radius τ_ℓ around each class submanifold in the previous section, we now introduce a ranking scheme that explicitly uses this local thickness to detect off-tube activations. Standard nearest-neighbour ranks fail to penalise points that lie off the estimated submanifold, so we propose a **Locally Adaptive Ranking** (LAR) that assigns the worst possible rank to any

activation escaping its tube, while preserving the natural ranking for those that remain inside.

For each input x with predicted class c, we measure its alignment with the validation manifold at layer ℓ by computing the Euclidean distance between its activation $h^{(\ell)}(x)$ and the activation of every validation sample $v \in \mathcal{V}$. We then sort the full validation set in ascending order of distance:

$$v_{(1)}, v_{(2)}, \ldots, v_{(|\mathcal{V}|)} = \arg\operatorname{sort}_{v \in \mathcal{V}} \, d\big(h^{(\ell)}(x), h^{(\ell)}(v)\big), \tag{5}$$

where $v_{(k)}$ is the k-th nearest neighbour to x in activation.

Denoting by $y(v)$ the ground-truth label of v, we define the layer-wise rank of x as the index of the first neighbour whose true label matches the model's prediction:

$$R_\ell(x) \;=\; \min\big\{\, k \;\big|\; y\big(v_{(k)}\big) = c \big\}. \tag{6}$$

This rank $R_\ell(x)$ captures how quickly a same-class validation activation appears when we scan the entire set by increasing distance.

To decide whether $h^{(\ell)}(x)$ lies inside its class-c tube $\mathcal{T}_c^{(\ell)}(\tau_\ell)$, we first find the closest validation activation of class c. Specifically, let

$$v^* \;=\; \arg\min_{v \in \mathcal{V}_c} \big\| h^{(\ell)}(x) - h^{(\ell)}(v) \big\|_2 \tag{7}$$

be that nearest neighbour. We then adjust the rank by checking whether $h^{(\ell)}(x)$ lies outside the tube of radius τ_ℓ:

$$R_\ell(x) = \begin{cases} |\mathcal{V}|, & \big\| h^{(\ell)}(x) - h^{(\ell)}(v^*) \big\|_2 > \tau_\ell, \\ R_\ell(x), & \text{otherwise}, \end{cases} \tag{8}$$

where $|\mathcal{V}|$ is the total number of validation samples. By forcing off-tube activations to take the worst-case rank, LAR sharply increases sensitivity to true off-manifold (backdoor) perturbations while leaving on-tube variations unpenalised.

3.5 Trajectory Modelling and Detection

As illustrated by the failure case in Fig. 1, slight backdoor-induced perturbations can slip past layer-wise checks—each individual deviation is too small to trigger an alarm, yet together they form a coherent off-manifold path. Having established in the previous section how Locally Adaptive Ranks quantify per-layer deviations from the clean-tube manifold, we now exploit their sequential structure: TED++ models each sample's rank sequence as a trajectory in $\mathbb{R}^L$ and learns its normal subspace via fitting or training PCA for robust backdoor detection.

Building on the Topological Evolution Dynamics framework [32], TED++ refines this trajectory-based subspace approach by incorporating locally adaptive rank statistics. Concretely, each sample $x \in \mathcal{V}$ is represented by its sequence of ranks:

$$R(x) \;=\; \big[\, R_1(x),\, R_2(x),\, \ldots,\, R_L(x) \big], \tag{9}$$

Table 1: Benign accuracy (BA) and attack success rate (ASR) w.r.t. different attacks and datasets with the ResNet-18 model.

Attacks→	None		BadNets		Blend		Ada-Patch		Ada-Blend		WaNet		Trojan		IAD		TaCT		SSDT	
Datasets↓	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
CIFAR-10	0.9391	——	0.9250	1.0000	0.9358	0.9993	0.9375	0.8696	0.9293	0.8693	0.9275	0.9762	0.9349	1.0000	0.9328	0.9999	0.9415	0.9975	0.9404	0.9710
GTSRB	0.9699	——	0.9685	1.0000	0.9707	0.9990	0.9614	0.9984	0.9647	0.9924	0.9596	0.9090	0.9683	1.0000	0.9676	1.0000	0.9714	1.0000	0.9743	0.9888
TinyImageNet	0.6081	——	0.5639	0.9328	——	——	0.5836	1.0000	——	——	——	——	0.5759	1.0000	——	——	——	——	0.5948	0.9200

Table 2: Performance (AUROC, F1) of our method compared to other state-of-the-art defence methods against various attacks on CIFAR-10. TED and TED++ are provided with a small validation set consisting of 5 samples per class. We mark the best result in **boldface**, the second best result as underlined and failed cases (< 0.7) in red.

Attacks→	BadNets		Blend		Ada-Patch		Ada-Blend		WaNet		Trojan		IAD		TaCT		SSDT		*Avg.*	
Defences↓	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
SCALE-UP	0.9643	0.9151	0.6876	0.5238	0.7963	0.7377	0.7552	0.6392	0.7247	0.6137	0.9219	0.8808	0.9657	0.9292	0.6001	0.2878	0.4974	0.1183	0.7681	0.6273
STRIP	0.6408	0.2323	0.7389	0.5608	0.8212	0.6847	0.9103	0.8193	0.4564	0.1131	0.7115	0.3099	0.9868	0.9397	0.4601	0.1006	0.4916	0.0924	0.6908	0.4281
IBD-PSC	**0.9960**	**0.9563**	**0.9950**	0.9648	0.8856	0.9166	0.8588	0.7711	**0.9736**	**0.9588**	**0.9650**	0.9558	**1.0000**	**0.9726**	0.8344	0.8762	0.4853	0.0671	0.8882	0.8266
TED	0.9680	0.9388	0.9920	**0.9709**	0.8636	0.8095	0.6228	0.0370	0.9652	0.9200	0.6288	0.1111	0.8148	0.6667	0.6860	0.0385	0.9248	0.8478	0.8296	0.6936
TED++	0.9944	0.9515	0.9208	0.8283	**0.9964**	**0.9709**	**0.9308**	**0.8909**	0.9172	0.8788	0.9424	**0.9709**	0.9988	0.9259	**1.0000**	**0.9524**	**0.9980**	**0.9174**	**0.9665**	**0.9581**

where $R_\ell(x)$ is the Locally Adaptive Rank at layer ℓ. Under the *tube-constrained manifold hypothesis*, clean-input trajectories occupy a low-dimensional subspace in $\mathbb{R}^L$, exhibiting only mild, structured fluctuations. We collect these rank trajectories from all clean validation samples,

$$\mathcal{R} = \{\, R(v) \mid v \in \mathcal{V} \},\tag{10}$$

and fit a PCA model to capture their principal modes. Let $U_K \in \mathbb{R}^{L\times K}$ be the matrix whose K columns are the top K principal components of $\mathcal{R}$. We then reconstruct any trajectory $R(x)$ as

$$\hat{R}(x) \;=\; U_K\, U_K^\top\, R(x),\tag{11}$$

and define its squared reconstruction error $e(x) \;=\; \big\|R(x) - \hat{R}(x)\big\|_2^2$. At test time, we declare

$$\mathrm{Detect}(x) \;=\; \big[e(x) > \theta\big],\tag{12}$$

flagging x as "poisoned" whenever its trajectory error $e(x)$ exceeds the threshold θ (set to control false alarms on the validation set).

This simple test—driven by the general principle of subspace-based anomaly detection—efficiently closes the loop from our illustrative failure case in Fig 1 through the layer-wise rank modelling and into a concrete, low-overhead implementation that reliably distinguishes true backdoor perturbations from clean variations.

4 Experiments

4.1 Experiment Settings

Datasets and Models. We conduct experiments on CIFAR-10 [18] and GTSRB [37] for detailed evaluation, and use TinyImageNet [19] as an extension due to

Table 3: Performance (AUROC, F1) of our method compared to other state-of-the-art defence methods against various attacks on GTSRB. TED and TED++ are provided with a small validation set consisting of 5 samples per class. We mark the best result in **boldface**, the second best result as underlined and failed cases (< 0.7) in red.

Attacks→	BadNets		Blend		Ada-Patch		Ada-Blend		WaNet		Trojan		IAD		TaCT		SSDT		*Avg.*	
Defences↓	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
SCALE-UP	0.9025	0.8371	0.6224	0.5577	0.8898	0.8244	0.5979	0.5314	0.3007	0.1753	0.2145	0.0652	0.8968	0.8321	0.4970	0.1005	0.5102	0.0926	0.6035	0.4471
STRIP	0.9531	0.8885	0.9154	0.8426	0.9927	0.9470	0.9390	0.8829	0.4556	0.1390	0.7473	0.4804	**0.9984**	0.9497	0.4253	0.0287	0.5111	0.1200	0.7709	0.5865
IBD-PSC	**0.9638**	**0.9650**	0.9116	0.3605	0.9706	0.9422	0.8636	0.0938	0.8855	**0.9162**	**0.9581**	**0.9525**	0.9633	**0.9642**	0.4876	0.0000	0.5312	0.5303	0.8373	0.6361
TED	0.9566	0.9423	0.9320	0.5217	0.9432	0.9149	0.7396	0.6301	0.9136	0.9091	0.8912	0.5352	0.9384	0.9320	0.8440	0.7229	**0.9996**	**0.9804**	0.9065	0.8163
TED++	0.9380	0.9020	**0.9964**	**0.9608**	**1.0000**	**0.9709**	**0.9686**	**0.9412**	**0.9138**	0.8090	0.9500	0.9346	0.9764	0.9524	**0.9166**	**0.9159**	0.9576	0.8421	**0.9575**	**0.9405**

Table 4: Performance (AUROC, F1) of our method compared to other state-of-the-art defence methods against various attacks on TinyImageNet. TED and TED++ are provided with a small validation set consisting of 5 samples per class. We mark the best result in **boldface**, the second best result as underlined and failed cases (< 0.7) in red.

Defences→	IBD-PSC		TED		TED++	
Attacks↓	AUC	F1	AUC	F1	AUC	F1
BadNets	**0.9511**	**0.9569**	0.8096	0.6265	0.8822	0.8214
Ada-Patch	**0.9991**	**0.9795**	0.5848	0.0000	0.9600	0.8772
Trojan	**1.0000**	**1.0000**	0.6260	0.0370	0.9340	0.8696
SSDT	0.3350	0.0000	0.7560	0.0000	**0.7920**	**0.7921**
Average	0.8213	0.7341	0.6934	0.1659	**0.8920**	**0.8400**

computational constraints, employing the ResNet-18 [12] architecture.

Attack Baselines. We compare our method to other existing defences against nine representative backdoor attacks: BadNets [7], Blend [1], Ada-Patch, Ada-Blend [35], TaCT [38], WaNet [34], Trojan [30], IAD [33], and SSDT [32]. The first five attacks are poison-only attacks, while the remaining four are training-controlled attacks. More specifically, TaCT is a source-specific attack, IAD is a dynamic-trigger attack, and SSDT is both source-specific and dynamic-trigger. All attacks were successfully trained using the ResNet-18 [12] model, and their corresponding results are presented in Tab 1.

Defence Settings. We compare our method to state-of-the-art input-level detection defences, including TED [32], IBD-PSC [13], STRIP [4], and SCALE-UP [9]. We set $\beta = 0.5$ to determine the LAR-based adaptive threshold τ_ℓ. Defenders can access 2 to 20 benign samples per class for each dataset for different experiments.

Evaluation Metrics. We use two well-recognised metrics: AUROC, which evaluates the performance of detection methods over a range of thresholds; and the F1 score, which integrates both the precision and recall aspects of detection.

Table 5: Comparison of AUROC performance between TED and TED++ under various attack types, using different numbers of validation samples per class on CIFAR-10 and GTSRB datasets. We mark better results in **boldface** and failed cases (< 0.7) in red.

Attacks	CIFAR-10								GTSRB							
	20		10		5		2		20		10		5		2	
	TED	TED++	TED	TED++	TED	TED++	TED	TED++	TED	TED++	TED	TED++	TED	TED++	TED	TED++
BadNets	0.9548	**0.9948**	0.9724	**0.9912**	0.9680	**0.9944**	0.8380	**0.9488**	0.9604	**0.9920**	0.9546	**0.9748**	**0.9566**	0.9380	**0.9230**	0.8912
Blend	0.9772	**0.9980**	**0.9844**	0.9708	**0.9920**	0.9208	0.3676	**0.8836**	0.9864	**0.9908**	0.9620	**0.9760**	0.9320	**0.9964**	0.1400	**0.8538**
Ada-Patch	0.8360	**0.9928**	0.8080	**0.9924**	0.8636	**0.9964**	0.4548	**0.9356**	0.9352	**0.9844**	0.9180	**0.9874**	0.9432	**1.0000**	0.7200	**0.9394**
Ada-Blend	0.7600	**0.9984**	0.6356	**0.9864**	0.6228	**0.9308**	0.6796	**0.9616**	0.8980	**0.9956**	0.8556	**0.9870**	0.7396	**0.9686**	0.3450	**0.8716**
WaNet	0.8660	**0.9516**	0.7556	**0.9348**	**0.9652**	0.9172	**0.9180**	0.8872	**0.9246**	0.9202	**0.9400**	0.8990	0.9136	**0.9138**	**0.8928**	0.8384
Trojan	0.7912	**0.9996**	0.7948	**1.0000**	0.6288	**0.9424**	0.7188	**0.9660**	0.9462	**0.9874**	0.9370	**0.9810**	0.8912	**0.9500**	0.3476	**0.8894**
IAD	0.8988	**0.9956**	0.8500	**0.9944**	0.8148	**0.9988**	0.6132	**0.9812**	0.9832	**1.0000**	0.9740	**1.0000**	0.9384	**0.9764**	**0.9996**	0.9800
TaCT	0.7428	**1.0000**	0.7516	**1.0000**	0.6860	**1.0000**	0.8904	**1.0000**	0.9328	**0.9636**	0.8916	**0.9392**	0.8440	**0.9166**	0.5436	**0.8364**
SSDT	0.9996	**1.0000**	0.9716	**0.9984**	0.9248	**0.9980**	0.7536	**0.9492**	**0.9996**	0.9816	**0.9996**	0.9476	**0.9996**	0.9576	**0.9388**	0.8042
Average	0.8696	**0.9923**	0.8360	**0.9853**	0.8296	**0.9665**	0.6924	**0.9459**	0.9518	**0.9795**	0.9369	**0.9658**	0.9065	**0.9575**	0.6500	**0.8783**

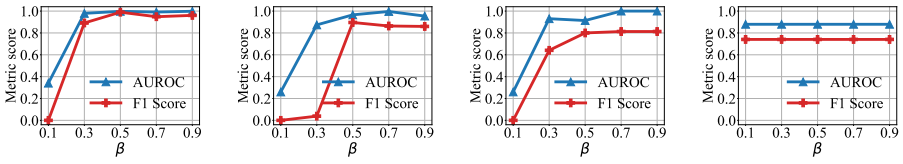
4.2 Main Results

As shown in Tab 2 and 3, TED++ successfully detects poisoned samples, as demonstrated by the AUROC and F1 Score against various attacks. This makes TED++ robust and comprehensive among current defences, particularly since TED [32] performs poorly with small validation sets and IBD-PSC [13] cannot detect source-specific attacks such as TaCT [38] and SSDT [32]. In addition, SCALE-UP [9] and STRIP [4] are not as effective as the other methods, especially when compared to TED++, despite us using only 5 validation samples per class in all experiments.

After extensive experiments on CIFAR-10 [18] and GTSRB [37], we have demonstrated that TED++ is the most effective input-level backdoor defence, as it does not fail against any of the selected attacks. We present additional experiments on a large-scale dataset, involving the top three input-level defences to date against four representative attacks in different categories: BadNets [7], Ada-Patch [35], Trojan [30], and SSDT [32]. The experiments are conducted on TinyImageNet [19], which contains 200 classes. Due to the large number of classes, we aim to make the experiment as realistic as possible by selecting only five samples per class for the validation set for TED and TED++. The results, presented in Tab 4, are consistent with previous findings on CIFAR-10 and GTSRB. IBD-PSC performs competitively across all attacks except SSDT, while TED++ maintains strong performance in all attacks, even against SSDT. On the other hand, TED shows the weakest performance due to the validation set limitation.

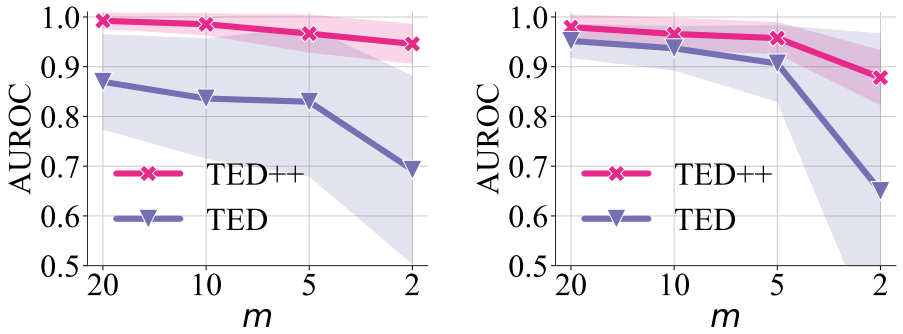
4.3 Ablation Study

Impact of the Number of Validation Samples m. For TED++ to operate properly, defenders must be provided with a validation dataset containing at least two samples per class. We investigate the effect of varying the number m of validation samples per class on the performance of TED++. In particular, we tested TED++ and TED [32] against nine attacks by employing different values of m ranging from 20 samples per class to 2 samples per class on both



(a) $m = 20$ (b) $m = 10$ (c) $m = 5$ (d) $m = 2$

Figure 3: Performance of TED++ against the Blend attack using various combinations of validation sample counts m and the neighbour percentile factor β on CIFAR-10.



(a) CIFAR-10 (b) GTSRB

Figure 4: Comparison of AUROC scores between TED and TED++ as the number of validation samples decreases on CIFAR-10 and GTSRB.

CIFAR-10 [18] and GTSRB [37] in Tab 5. The results indicate that TED++ sustains consistent performance even when m is reduced to its minimum value, whereas the performance of TED deteriorates rapidly, which is illustrated in Fig 4. This demonstrates that when validation samples are limited, the specific value of m has a relatively minor influence on the effectiveness of TED++.

Impact of the Neighbour Percentile Factor β. TED++ uses the factor β to determine the number of nearest validation neighbours that establish the threshold for the ranking computation. Based on our observation, each clean sample has a nearest neighbour at a distance comparable to those among other clean samples. In contrast, poisoned samples tend to have a greater distance to their neighbours compared to clean samples. In our algorithm, we assume that if the distance from a sample to its neighbour exceeds the maximum distance between that neighbour and its $\lceil m\beta \rceil$ nearest neighbours, the suspicious sample will be assigned a high rank, with a selected value being $\beta = 0.5$. To validate this assumption, we performed experiments, illustrated in Fig 3, which include line plots for various β values across different numbers of validation samples per class, specifically for TED++ against the Blend [1] attack on CIFAR-10. The

Algorithm 2 Nearest-Neighbour Label Flipping

Require: A C-class network f; validation sets $\{\mathcal{V}_i\}_{i=1}^{C}$ with $\mathcal{C}_{\text{val}} = \{\, i \mid |\mathcal{V}_i| \geq 2 \,\}$; test example x with confidence vector $\mathbf{p} = f(x) \in [0,1]^C$.

procedure NNLF(x)
  $\hat{y} \leftarrow \arg\max_k \mathbf{p}_k$ ▷ initial predicted label
  if $\hat{y} \notin \mathcal{C}_{\text{val}}$ **then** ▷ label absent in validation set
    $\mathbf{p}_{\text{mask}} \leftarrow \mathbf{p}$ with $\mathbf{p}_k \leftarrow 0 \;\forall\, k \notin \mathcal{C}_{\text{val}}$
    $\hat{y} \leftarrow \arg\max_{k \in \mathcal{C}_{\text{val}}} \mathbf{p}_{\text{mask},k}$ ▷ flip to highest-confidence valid class
  return $\hat{y}$ ▷ final predicted label

plots show that $0.9 \geq \beta \geq 0.5$ is optimal because increasing β beyond this range can cause TED++ to fail if there are outliers in the validation set, while lowering it can make the threshold overly sensitive. This finding is further supported by the results presented in Tab 2 and Tab 3.

Impact of Missing Validation Classes ρ. TED++ uses $\rho \in [0, 1)$ to quantify the fraction of validation classes missing: $\rho = 0$ indicates none are missing, while larger ρ values denote progressively more absent classes. Although TED++ is a robust method performing well against existing backdoor attacks with only a minimal validation dataset, it has a strict requirement: defenders must have at least two validation samples per class for the defence to function properly. This can be particularly challenging, even when defenders have access to a large validation set, as not every class is guaranteed to have two or more samples. The question is whether TED++ can still work when the validation set contains only a subset of the total classes (i.e., when $\rho > 0$). In our exploration of how image inputs pass through the network, we observe that the benign samples will be close to each other in the embedding space across different classes, and the distance will only really differ when the input is poisoned. To address this limitation, we apply a technique called **Nearest-Neighbour Label Flipping**. Specifically, when an input is predicted with a label not present in the validation set, we extract its confidence scores for each label and exclude all labels absent from the validation set. We then flip the input's predicted label to the remaining class with the highest confidence score. We conduct experiments presented in Tab 6 to evaluate TED++ against diverse attacks on CIFAR-10, GTSRB, and TinyImageNet under varying ρ values. Results indicate that even when $\rho = 0.4$, TED++ maintains robust performance. Specifically, the average AUROC scores slightly drop from 0.9665 to 0.8801 on CIFAR-10, from 0.9575 to 0.8694 on GTSRB, and notably increase from 0.8096 to 0.8753 on TinyImageNet as ρ increases from 0 to 0.4. The performance improvement observed on TinyImageNet can be attributed to its large number of classes; reducing the number of classes decreases noise in ranking computations, consequently increasing accuracy. The Nearest-Neighbour Label Flipping procedure is illustrated in Alg 2.

5 Discussion and Future Work

Discussion. Fundamentally, TED++ uses locally adaptive ranking and topological evolution dynamics to identify backdoor samples, based on the assumption that the trajectories of poisoned inputs deviate significantly from those of benign samples of the target class. TED++ has consistently demonstrated outstanding performance across a wide range of circumstances during our evaluation. We tested nine representative backdoor attacks on three popular benchmark datasets and achieved outstanding results with AUROC scores above 0.95 in almost all cases. Furthermore, even in challenging circumstances like class imbalance and data scarcity, TED++ maintained exceptional effectiveness, achieving AUROC scores mostly better than 0.8.

Given the robustness of the trajectory-based detection mechanism, we find TED++ detects a broad range of backdoor attacks. As long as an input is poisoned, its trajectory through the neural network inherently differs from that of benign inputs. To date, creating an attack with perfect inseparability in the latent feature space has proven infeasible. Previously, [35] revisited the assumption of latent separability for backdoor defences and concluded that this assumption can fail by designing Ada-Blend and Ada-Patch attacks. However, in our experiments, these attacks are still easily detected by TED++, demonstrating that this conclusion is not entirely accurate: most layers of the model clearly showcase separability between poisoned and clean samples in the embedding space. However, we acknowledge a practical limitation: when employing extremely deep neural networks with highly complex embedding spaces, the computational cost of rank-based metrics can become high.

Several critical limitations present in the original TED [32] method have been effectively addressed in our work. Specifically, TED's reliance on a large validation dataset and its expensive computational cost have been significantly alleviated in TED++. Although TED++ still has a comparable computational complexity per inference compared to TED, its dramatically reduced reliance on validation data greatly accelerates the inference process. Fig 5 shows the per-sample inference time of all SOTA defences using a 35-layer ResNet-18 model on the CIFAR-10 [18] dataset, demonstrating that TED++ achieves inference times comparable to the previous methods while delivering the highest overall performance. These results highlight TED++'s practical suitability for real-time applications.

In comparison with current state-of-the-art input-level backdoor defences, TED++ outperforms all competitors, standing on par only with IBD-PSC [13]. However, unlike IBD-PSC, TED++ effectively detects source-specific attacks, which remain a critical limitation of IBD-PSC. Hence, we consider TED++ to be the most comprehensive input-level defence solution available to date.

Future Work. In this work, we successfully addressed the limitations identified in the previous work [32] by enhancing its capability to detect a diverse set of backdoor attacks. Our experiments position TED++ and IBD-PSC as the two leading input-level defence mechanisms, each with distinct strengths and limitations. While TED++ may encounter computational constraints with very

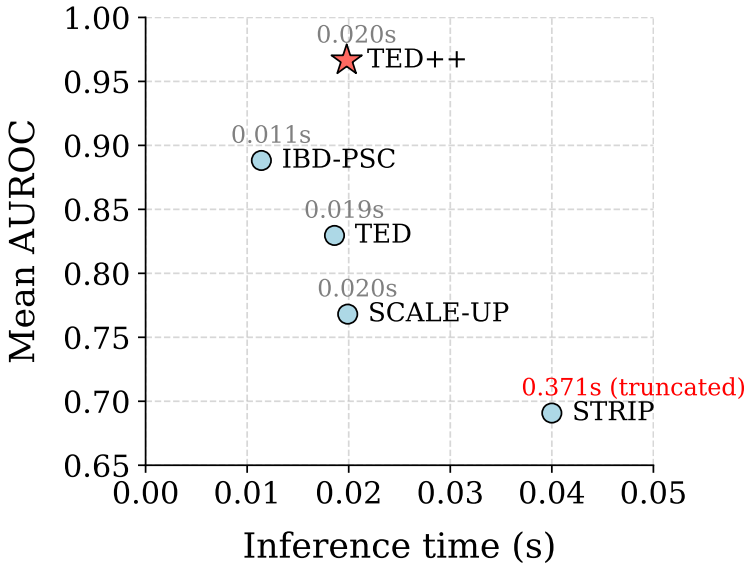


Figure 5: Inference times of state-of-the-art input-level defences on the CIFAR-10 dataset. We provide TED and TED++ with 5 validation samples per class, and apply default settings for the other defences.

deep networks or intricate embedding spaces, IBD-PSC remains vulnerable to source-specific attacks.

This observation opens a promising direction for future research: designing a next-generation input-level backdoor defence that combines the detection robustness of TED++ with improved computational efficiency. Achieving this goal may require exploring innovative approaches to deep neural network architectures or embedding strategies, which aim to highlight the inherent differences between poisoned and benign inputs without incurring significant computational costs, therefore providing a more scalable and widely applicable solution.

6 Conclusion

In this paper, we propose TED++, a robust backdoor defence that outperforms existing methods by effectively detecting backdoor attacks using a minimal validation dataset, a capability that previous defences lack. TED++ was developed based on insights into how poisoned and clean samples propagate through neural networks and interact within the hidden feature space. We observed distinct behavioural patterns of poisoned samples compared to clean ones across most network layers. Recognising the limitations of current defence strategies, we introduced the locally adaptive ranking method in TED++, significantly enhanc-

ing detection performance. Specifically, TED++ achieved notable improvements on CIFAR-10 [18], with increases of 16.50% in AUROC over TED [32], and 8.12% in AUROC over IBD-PSC [13]. On the GTSRB [37] dataset, TED++ improved by an average AUROC of 5.63% compared to TED, and by 14.36% to IBD-PSC. Moreover, TED++ successfully defended against all tested attacks on TinyImageNet [19], whereas both IBD-PSC and TED failed. Beyond performance, TED++ addresses critical practical limitations, including reducing computational costs and managing scenarios with insufficient validation classes, ensuring its applicability under diverse conditions. The comprehensive experimental results demonstrate that TED++ is currently the most robust and effective input-level backdoor defence method available.

References

[1] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *arXiv*, 2017.

[2] Qiuyu Duan, Zhongyun Hua, Qing Liao, Yushu Zhang, and Leo Yu Zhang. Conditional backdoor attack via JPEG compression. In *AAAI*, volume 38, 2024.

[3] Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[4] 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 *ACSAC*, pages 113–125, 2019.

[5] Yinghua Gao, Yiming Li, Linghui Zhu, Dongxian Wu, Yong Jiang, and Shu-Tao Xia. Not all samples are born equal: Towards effective clean-label backdoor attacks. *Pattern Recognition*, 2023.

[6] Xueluan Gong, Ziyao Wang, Yanjiao Chen, Meng Xue, Qian Wang, and Chao Shen. Kaleidoscope: Physical backdoor attacks against deep neural networks with RGB filters. *IEEE Transactions on Dependable and Secure Computing*, 2023.

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

[8] Junfeng Guo, Ang Li, Lixu Wang, and Cong Liu. PolicyCleanse: Backdoor detection and mitigation in reinforcement learning. In *ICCV*, 2023.

[9] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. SCALE-UP: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. In *ICLR*, 2023.

[10] Junfeng Guo, Yiming Li, Lixu Wang, Shu-Tao Xia, Heng Huang, Cong Liu, and Bo Li. Domain watermark: Effective and harmless dataset copyright protection is closed at hand. In *NeurIPS*, 2023.

[11] Wenbo Guo, Lun Wang, Yan Xu, Xinyu Xing, Min Du, and Dawn Song. Towards inspecting and eliminating trojan backdoors in deep neural networks. In *ICDM*, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Linshan Hou, Ruili Feng, Zhongyun Hua, Wei Luo, Leo Yu Zhang, and Yiming Li. IBD-PSC: Input-level backdoor detection via parameter-oriented scaling consistency. In *ICML*, 2024.

[14] Linshan Hou, Zhongyun Hua, Wei Luo, and Leo Yu Zhang. Fixguard: Repairing backdoored models via class-wise trigger recovery and unlearning. *IEEE Signal Processing Letters*, 2025.

[15] Linshan Hou, Wei Luo, Zhongyun Hua, Songhua Chen, Leo Yu Zhang, and Yiming Li. Flare: Toward universal dataset purification against backdoor attacks. *IEEE Transactions on Information Forensics and Security*, 20:6459–6473, 2025.

[16] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *ICLR*, 2022.

[17] Najeeb Moharram Jebreel, Josep Domingo-Ferrer, and Yiming Li. Defending against backdoor attacks by layer-wise feature analysis. In *SIGKDD*, 2023.

[18] Alex Krizhevsky. The cifar-10 dataset. `https://www.cs.toronto.edu/~kriz/cifar.html`, 2009.

[19] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. *CS231n*, 7(7):3, 2015.

[20] John M Lee. *Introduction to smooth manifolds*. Springer, 2012.

[21] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021.

[22] Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. In *NeurIPS*, 2022.

[23] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[24] Yiming Li, Tongqing Zhai, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor attack in the physical world. In *ICLR Workshop*, 2021.

[25] Yiming Li, Linghui Zhu, Xiaojun Jia, Yong Jiang, Shu-Tao Xia, and Xiaochun Cao. Defending against model stealing via verifying embedded external features. In *AAAI*, 2022.

[26] Yiming Li, Mingyan Zhu, Xue Yang, Yong Jiang, Tao Wei, and Shu-Tao Xia. Black-box dataset ownership verification via backdoor watermarking. *IEEE Transactions on Information Forensics and Security*, 2023.

[27] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *CCS*, pages 113–131, 2020.

[28] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

[29] 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 *CVPR*, 2023.

[30] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*, 2018.

[31] Xiaoxing Mo, Nan Sun, Leo Yu Zhang, Wei Luo, Shang Gao, and Yong Xiang. Arms race in deep learning: A survey of backdoor defenses and adaptive attacks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 308–325. Springer Nature Singapore Singapore, 2025.

[32] Xiaoxing Mo, Yechao Zhang, Leo Yu Zhang, Wei Luo, Nan Sun, Shengshan Hu, Shang Gao, and Yang Xiang. Robust backdoor detection for deep learning via topological evolution dynamics. In *IEEE S&P*, 2024.

[33] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *NeurIPS*, 2020.

[34] Tuan Anh Nguyen and Anh Tuan Tran. WaNet – Imperceptible Warping-based Backdoor Attack. In *ICLR*, 2021.

[35] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the Assumption of Latent Separability for Backdoor Defenses. In *ICLR*, 2023.

[36] Xiangyu Qi, Tinghao Xie, Ruizhe Pan, Jifeng Zhu, Yong Yang, and Kai Bu. Towards practical deployment-stage backdoor attack on deep neural networks. In *CVPR*, 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] 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*, 2021.

[39] Ruixiang Tang, Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. An embarrassingly simple approach for trojan attack in deep neural networks. In *SIGKDD*, 2020.

[40] Ruixiang Tang, Jiayi Yuan, Yiming Li, Zirui Liu, Rui Chen, and Xia Hu. Setting the trap: Capturing and defeating backdoor threats in plms through honeypots. In *NeurIPS*, 2023.

[41] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, 2018.

[42] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv*, 2019.

[43] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 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 *IEEE S&P*, 2019.

[45] Hang Wang, Zhen Xiang, David J Miller, and George Kesidis. MM-BD: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *IEEE S&P*, 2024.

[46] Z. Wang, J. Zhai, and S. Ma. BppAttack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *CVPR*, 2022.

[47] Zhenting Wang, Hailun Ding, Juan Zhai, and Shiqing Ma. Training with more confidence: Mitigating injected and natural backdoors during training. In *NeurIPS*, 2022.

[48] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. Backdoor attacks against deep learning systems in the physical world. In *CVPR*, 2021.

[49] Zhen Xiang, Zidi Xiong, and Bo Li. UMD: Unsupervised model detection for x2x backdoor attacks. In *ICML*, 2023.

[50] Honghui Xu, Zhipeng Cai, Zuobin Xiong, and Wei Li. Backdoor attack on 3D grey image segmentation. In *ICDM*, 2023.

[51] Mengxi Ya, Yiming Li, Tao Dai, Bin Wang, Yong Jiang, and Shu-Tao Xia. Towards faithful xai evaluation via generalization-limited backdoor watermark. In *ICLR*, 2024.

[52] Yi Zeng, Si Chen, Won Park, Z Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *ICLR*, 2022.

[53] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. Narcissus: A practical clean-label backdoor attack with limited information. In *CCS*, 2023.

[54] Hangtao Zhang, Shengshan Hu, Yichen Wang, Leo Yu Zhang, Ziqi Zhou, Xianlong Wang, Yanjun Zhang, and Chao Chen. Detector collapse: Backdooring object detection to catastrophic overload or blindness. In *IJCAI*, 2024.

[55] Jie Zhang, Chen Dongdong, Qidong Huang, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Poison ink: Robust and invisible backdoor attack. *IEEE Transactions on Image Processing*, 2022.

[56] Yechao Zhang, Yuxuan Zhou, Tianyu Li, Minghui Li, Shengshan Hu, Wei Luo, and Leo Yu Zhang. Secure transfer learning: Training clean model against backdoor in pre-trained encoder and downstream dataset. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 1–19. IEEE, 2025.

Table 6: AUROC performance of TED++ under various attack types, with 5 validation samples per class on CIFAR-10 and GTSRB and 2 validation samples per class on TinyImageNet, when $\rho = 0.4$ increases from 0%–40%. We mark failed cases (< 0.7) in red. When $\rho = 0.4$, the validation set contains 30 samples for CIFAR-10, 129 samples for GTSRB, and 240 samples for TinyImageNet.

Datasets	**Attacks**	0%	10%	20%	30%	40%
CIFAR-10	BadNets	0.9944	0.9442	0.9588	0.9084	0.8656
	Blend	0.9208	0.8668	0.8960	0.8788	0.7772
	Ada-Patch	0.9964	0.9804	0.9952	0.9896	0.9888
	Ada-Blend	0.9308	0.8904	0.8872	0.9168	0.8112
	WaNet	0.9172	0.9060	0.9192	0.8428	0.7868
	Trojan	0.9424	0.9257	0.9329	0.8935	0.9056
	IAD	0.9988	0.9924	0.9632	0.9848	0.9516
	TaCT	1.0000	0.9618	0.9618	0.9430	0.9412
	SSDT	0.9980	0.9788	0.9644	0.9260	0.8932
	Average	0.9665	0.9385	0.9421	0.9204	0.8801
GTSRB	BadNets	0.9380	0.8658	0.8156	0.8548	0.8214
	Blend	0.9964	0.9250	0.8752	0.8868	0.9304
	Ada-Patch	1.0000	0.9800	0.8950	0.9828	0.9288
	Ada-Blend	0.9686	0.9410	0.9068	0.8954	0.8694
	WaNet	0.9138	0.8684	0.7602	0.8247	0.8575
	Trojan	0.9500	0.9002	0.8268	0.7896	0.7844
	IAD	0.9764	1.0000	0.9888	0.9886	1.0000
	TaCT	0.9166	0.9102	0.8420	0.8304	0.8032
	SSDT	0.9576	0.9248	0.8772	0.8472	0.8296
	Average	0.9575	0.9239	0.8653	0.8778	0.8694
TinyImageNet	BadNets	0.8558	0.9096	0.8780	0.9012	0.8708
	Ada-Patch	0.9000	0.9088	0.9436	0.9316	0.9340
	Trojan	0.7224	0.8228	0.8824	0.8676	0.8404
	SSDT	0.7600	0.8000	0.8400	0.8640	0.8560
	Average	0.8096	0.8603	0.8860	0.8911	0.8753