Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CLEAR: Cross-Layer Extreme Activation Representation for Backdoor Sample Detection

Hui Li
Zhengzhou University of Light Industry
Bingyan Li
Zhengzhou University of Light Industry
Wanwei Huang
byli2001@163.com

Zhengzhou University of Light Industry
Lifeng Jin
Zhengzhou New Century Digital Technology Co., Ltd
Chunfeng Du
Zhengzhou University
Yi Ding
The Third Construction Co., Ltd of China CREC Railway Electrification Engineering Group
Shilin Liang
Zhengzhou University of Economics and Business

Research Article

Keywords: deep learning, backdoor sample detection, anomaly representation, particle swarm optimization, density-based clustering

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11193226/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

CLEAR: Cross-Layer Extreme Activation Representation for Backdoor Sample Detection

Hui Li[a], Bingyan Li[a], Wanwei Huang[a,*], Lifeng Jin[b], Chunfeng Du[c], Yi Ding[d] and Shilin Liang[e]

[a] Department of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou 450007, China

[b] Zhengzhou New Century Digital Technology Co., Ltd., Zhengzhou 450001, China

[c] School of Computer Science and Artificial Intelligence, Zhengzhou University, Zhengzhou 450007, China

[d] The Third Construction Co., Ltd of China CREC Railway Electrification Engineering Group, Zhengzhou 450052, China

[e] College of Big Data and Artificial Intelligence, Zhengzhou University of Economics and Business, Zhengzhou, 450099, China.

*Corresponding author(s). E-mail(s): byli2001@163.com;

Contributing authors: 2007037@zzuli.edu.cn; 332416020943@zzuli.edu.cn; zzxsj2025@163.com; dcf@zzu.edu.cn; lucker2568@163.com; lsl961212@163.com;

Abstract: The application of deep learning models in security-critical scenarios makes trustworthy training data essential for model reliability. However, existing backdoor sample detection methods often fail to capture subtle anomalous responses induced by stealthy triggers, particularly under low poisoning rates, complex trigger patterns, and the absence of clean validation data. To address these limitations, this paper proposes CLEAR, a Cross-Layer Extreme Activation Representation framework for backdoor sample detection. First, batch normalization (BN) statistics align feature distributions across hidden layers, while channel-wise extreme responses are extracted as backdoor anomaly representations to amplify subtle trigger-induced activation differences. Second, to avoid

relying on prior knowledge for layer-weight selection, particle swarm optimization models hidden-layer attention weights as particle positions and uses the stability of HDBSCAN clustering after Uniform Manifold Approximation and Projection (UMAP) as the fitness function, adaptively determining the optimal weight combination that maximizes the anomalous consistency of poisoned samples. Finally, density-based clustering is applied to optimally weighted features, and a predefined cluster-size threshold identifies the most stable and compact anomalous micro-cluster, enabling prior-free screening of poisoned samples without clean labels. Experiments on CIFAR-10 and Tiny-ImageNet show that CLEAR achieves average true positive rates (TPRs) of 97.54% and 99.54%, respectively, with average false positive rates (FPRs) of 0.16% and 0.03%, significantly outperforming AC, SCALE-UP, AIBD, and CCD. Moreover, models retrained on datasets purified by CLEAR maintain classification accuracy close to that of benign models while achieving substantially lower attack success rates (ASRs), demonstrating superior detection accuracy, robustness, and overall defense performance.

Keywords: deep learning; backdoor sample detection; anomaly representation; particle swarm optimization; density-based clustering.

1 Introduction

In recent years, the rapid advancement and widespread adoption of deep learning, together with its powerful feature extraction and nonlinear modeling capabilities, have enabled it to play a pivotal role in security-sensitive applications such as computer vision [1], autonomous driving [2], sentiment analysis [3], and speech recognition [4]. However, improvements in the performance of deep learning models depend heavily on large-scale, high-quality training data, while the reliability of data sources, accuracy of annotations, and integrity of sample distributions directly determine their generalization capabilities and security boundaries. In practical deployments, training data are often obtained from public datasets, third-party platforms, and multiparty collaborative collection processes [5], resulting in complex data circulation pipelines whose trustworthiness is difficult to guarantee [6]. When an adversary injects a small number of carefully crafted malicious samples into the training set, the model may learn a covert association between the trigger and the attack target, causing it to behave normally on clean inputs while producing attacker-specified erroneous outputs

under designated trigger conditions [7]. Such data-poisoning-based backdoor attacks are characterized by high stealthiness, high attack success rates, and considerable detection difficulty, thereby posing a severe threat to the reliable deployment of deep learning models in security-critical domains. Therefore, accurately identifying and removing poisoned samples from untrusted training data to enhance the security and reliability of model training has become an important research problem in deep learning security.

In response to the aforementioned risks, existing defense methods can be broadly categorized into three types based on the stage at which they operate: input-level defenses, model-level defenses, and dataset-level defenses [8]. Input-level defense primarily operates during the model inference phase, preventing backdoor activation by detecting whether input samples contain trigger patterns. For example, Liu et al. [9] proposed placing an autoencoder as an input preprocessor at the front end of the model, utilizing the difference in reconstruction error between normal and compromised samples to filter out triggers; Javaheripi et al. [10] proposed the CLEANN embedded backdoor online detection framework, which relies on dictionary learning and sparse approximation to distinguish trigger samples, and incorporates specialized hardware to enhance operational efficiency. Model-level defenses repair compromised models by analyzing neuronal responses or parameter behaviors; for example, Jiang et al. [11] proposed BDA, which identifies and prunes backdoor-related neurons based on activation frequency and an anomaly index, while Sun et al. [12] introduced CARE, a causal repair method that constructs a causal model to localize defective neurons and employs particle swarm optimization to adjust model parameters for precise backdoor remediation. Dataset-level defenses focus on eliminating poisoned samples before model training; for example, Hayase et al. [13] employed robust covariance estimation to whiten feature representations and amplify spectral discrepancies, while Li et al. [14] proposed Anti-Backdoor Learning, which distinguishes samples according to differences in training loss and sequentially applies local and global gradient ascent to remove poisoned samples and eliminate backdoor features. However, although these three categories of defenses mitigate backdoor attacks at the dataset, input, and model levels, they generally remain dependent on attack-specific prior knowledge, clean validation sets, or particular model-access conditions in complex real-world scenarios. More importantly, most existing methods make detection decisions based on features from a single hidden layer or on input‑output mappings, overlooking the cross-layer distribution

of trigger-induced anomalous responses across different network depths. When the poisoning rate is extremely low, the trigger pattern is stealthy, or the attack follows an all-to-all (A2A) or untargeted (UT) paradigm, these weak and dispersed anomalous signals are difficult to capture using single-layer features, resulting in a substantial deterioration in the detection robustness of existing methods.

To address the aforementioned limitations, this paper proposes CLEAR, a Cross-Layer Extreme Activation Representation framework for backdoor sample detection. The central idea of CLEAR is to align feature scales across different hidden layers using batch normalization (BN) statistics and preserve weak trigger-induced anomalous responses through extreme-value extraction; to employ particle swarm optimization (PSO) to adaptively search for optimal layer-wise attention weights that amplify anomalous consistency among poisoned samples; and to integrate Uniform Manifold Approximation and Projection (UMAP) with HDBSCAN density-based clustering to achieve high-precision backdoor sample detection without attack prior knowledge or additional clean data. The principal contributions of this work are summarized as follows:

(1) We reveal the “cross-layer dispersion” phenomenon of anomalous responses under complex backdoor attacks. Empirical analyses show that, under complex attack paradigms such as A2A and UT, the anomalous activations of poisoned samples are not concentrated in any specific layer but are sparsely distributed across multiple hidden layers, invalidating conventional detection assumptions based on single-layer features or output confidence. This observation provides strong empirical motivation for the development of cross-layer aggregation-based detection strategies.

(2) We design an adaptive detection framework based on cross-layer extreme-value features and particle-swarm-based weight optimization. To address heterogeneous feature scales and unknown layer importance, the framework first aligns layer-wise feature distributions using BN statistics and extracts channel-wise extreme responses to amplify subtle trigger-induced activation traces; it then models layer-attention weights as PSO search variables and adaptively optimizes them using feedback from UMAP embeddings and HDBSCAN cluster stability, yielding an optimal weighted combination that maximizes anomalous-response consistency among poisoned samples and enables robust unsupervised fusion of weak cross-layer signals.

(3) We validate the effectiveness of the proposed method through experiments across multiple datasets and attack paradigms. Extensive experiments on CIFAR-10 and Tiny-ImageNet demonstrate that CLEAR achieves true positive rates (TPRs) approaching 100% and false positive

rates (FPRs) below 1% across various representative backdoor attack scenarios, significantly outperforming mainstream methods such as AC, SCALE-UP, AIBD, and CCD under low poisoning rates and complex trigger patterns. Furthermore, models retrained after the detected poisoned samples are removed maintain accuracy comparable to that of clean models while exhibiting substantially reduced attack success rates (ASRs), demonstrating the practical defensive value of CLEAR as a front-end data-purification module.

2 Related Work

2.1 Backdoor Attacks

Data poisoning backdoor attacks are a typical threat in the field of deep learning security [15]. The core principle behind these attacks is that an attacker injects poisoned samples containing hidden triggers into the training dataset, inducing the model to learn a hidden association between the trigger and the target output; the model maintains standard classification performance when presented with normal inputs, but when an input contains the corresponding trigger, it outputs an incorrect result predetermined by the attacker [16]. Depending on whether the original labels of the samples are modified during poisoning, data-poisoning backdoor attacks can be classified into dirty-label and clean-label attacks [17].

(1) Dirty-Label Attacks

Dirty-label attacks constitute the earliest and most widely adopted data-poisoning backdoor paradigm, characterized by relabeling injected poisoned samples as a predefined target class so that the resulting label bias guides the model to learn an association between the trigger pattern and the attack target. Gu et al. [18] proposed BadNets, the first data-poisoning backdoor attack, which implants a backdoor into a classification model by placing a fixed visible pattern in an image corner and relabeling the corresponding samples as the target class. Subsequent research has continued to evolve through the optimization of trigger patterns and enhancement of attack stealthiness. Chen et al. [19] proposed the Blend attack, which mixes pattern-based triggers with clean samples at varying opacity levels to enhance visual naturalness while preserving attack effectiveness; Nguyen et al. [20] introduced WaNet, which overcomes the limitations of planar pattern-based triggers by employing a small and smooth image-warping field that only slightly modifies local image regions, substantially improving trigger imperceptibility; Liu et al. [21] developed the Trojan attack, which adopts a reverse-engineering strategy to construct triggers and poisoned samples from trigger-

associated neurons, establishing a precise attack pathway from backdoor neurons to the target output. With respect to attack objectives, in addition to the aforementioned targeted attacks that induce specified misclassifications, Xue et al. [22] proposed an untargeted backdoor attack paradigm that does not specify a fixed misclassification target and instead produces randomly distributed predictions for triggered samples, further increasing attack stealthiness and detection difficulty.

(2) Clean-Label Attacks

Clean-label attacks are an advanced poisoning paradigm designed for highly stealthy settings, in which the original labels remain unchanged and only target-class samples are embedded with stealthy triggers or subjected to minor perturbations, thereby preserving consistency between labels and visual semantics and evading label-anomaly-based detection methods. The SIG attack proposed by Barni et al. [23] is a representative example of this category, generating poisoned samples by adding subtle frequency-domain perturbations to clean target-class samples and implanting a backdoor without modifying their labels. This attack challenges the conventional assumption that backdoor attacks necessarily depend on label manipulation, leaving poisoned samples without evident anomalies under either manual inspection or label-statistical analysis, substantially improving attack stealthiness and resistance to detection, and driving data-poisoning backdoor attacks toward increasingly stealthy and adversarial forms.

As defense technologies continue to advance, backdoor attacks are evolving toward more imperceptible triggers, weaker attack signatures, and reduced reliance on prior knowledge, exposing deep neural networks to increasingly severe security threats. Therefore, developing a general defense method capable of adapting to complex trigger scenarios and accurately countering highly stealthy backdoor attacks has become an urgent requirement for securing deep neural network applications.

2.2 Backdoor Defenses

As data-poisoning backdoor attacks continue to evolve toward greater stealthiness and dynamism, a wide range of defense techniques for deep learning models have rapidly emerged. According to their operational stages and targets of intervention, existing backdoor defenses can be broadly classified into dataset-level, input-level, and model-level approaches, which counter backdoor threats at the training-data source, inference input, and internal model levels, respectively [24].

(1) Input-Level Defense Methods

Input-level defenses primarily operate during model inference and seek to detect, filter, or process input samples, thereby preventing trigger-embedded samples from entering the inference pipeline, avoiding maliciously induced decisions, and preserving the accuracy of model outputs. Gao et al. [25] proposed an image-superposition-based backdoor detection approach that overlays multiple image patterns onto a target sample and identifies backdoor inputs by analyzing differences in the randomness of model predictions on the resulting composite samples. Chou et al. [26] proposed SentiNet, which employs Grad-CAM interpretability and selective-search segmentation to localize influential image regions, selects distinctive regions as potential triggers, transfers these regions to clean samples, and determines trigger presence from the resulting classification outputs and confidence scores. Li et al. [27] designed a defense strategy based on geometric transformations such as scaling and flipping, which spatially perturbs input samples to disrupt the inherent feature consistency of backdoor triggers and thereby suppress backdoor activation.

Although such defenses entail relatively low deployment costs and provide some protection against visible triggers, they cannot accommodate the full range of stealthy triggers and are less effective against attacks lacking salient trigger characteristics, limiting their applicability to complex attack scenarios.

(2) Model-Level Defense Methods

Model-level defenses are applied after model training and intercept or eliminate backdoors through parameter analysis, structural adjustment, or anomaly correction. Liu et al. [28] proposed Fine-Pruning, a model-pruning defense that identifies backdoor-associated neurons with anomalous weight characteristics and removes them through pruning, thereby weakening or eliminating the effects of backdoor attacks. Kolouri et al. [29] optimized universal litmus patterns and constructed a meta-classifier that determines whether a target model contains a backdoor based on its responses to the optimized patterns. Huang et al. [30] employed model interpretability analysis to compare the sparsity, smoothness, and stability of saliency heatmaps generated by benign and backdoored models, and then used the three resulting discriminative features to identify backdoored models through an outlier-detection meta-classifier. Xiang et al. [31] proposed a backdoor detection method based on expected transfer probability and mathematically derived a fixed decision threshold for binary classification settings.

Although these methods act directly on the model and can internally suppress backdoor propagation, they involve complex and costly deployment procedures, may degrade normal classification performance, and adapt poorly to complex attack scenarios.

(3) Dataset-Level Defense Methods

Dataset-level defenses operate during model training and safeguard the security and reliability of the training process by detecting and classifying training samples to accurately localize and remove malicious poisoned samples. Tran et al. [32] proposed Spectral Signatures, which leverages feature representations learned by deep neural networks and applies singular value decomposition (SVD) to amplify signals associated with backdoor samples, thereby detecting and removing them from the training set. Chan et al. [33] further introduced an SVD-based method robust to gradient noise, which extracts poisoning signals to identify the target class and neutralize backdoor effects. Chen et al. [34] proposed Activation Clustering (AC), which distinguishes clean samples from backdoor-poisoned samples within each label class by analyzing the clustering distribution of final hidden-layer activations together with differences in model behavior. Zeng et al. [35] proposed a frequency-domain analysis method that distinguishes clean images from malicious trigger-bearing images in the training set by exploiting differences in their spectral components after the discrete cosine transform.

In recent years, researchers have further explored enhanced detection approaches for low poisoning rates and complex trigger patterns. Guo et al. [36] proposed SCALE-UP, which identifies backdoor samples in a label-only black-box setting by scaling the pixel values of a test input and analyzing prediction consistency throughout the scaling process. Yin et al. [37] proposed AIBD, which isolates poisoned samples using differences in adversarial-example generation and a progressive Top-q strategy; however, its reliance on distinct adversarial responses between clean and poisoned samples can reduce detection accuracy when their behavioral characteristics overlap under complex attacks. Zhuang et al. [38] proposed CCD, which distinguishes poisoned samples from clean samples by analyzing confidence consistency during the early stages of training; however, its reliance on a single training dynamic makes it prone to missed detections and false positives when low poisoning rates or complex attacks cause confidence features to overlap.

Although the aforementioned methods achieve a certain level of effectiveness under conventional attack settings, they still exhibit several limitations: first, most rely exclusively on

activation features from a single hidden layer, typically the penultimate layer, and therefore neglect cross-layer response differences induced by backdoor triggers across network depths, as observed for A2A and UT attacks in the Introduction, causing weak and dispersed signals to be overlooked; second, some require attack priors, clean reference samples, fixed thresholds, or a predefined number of clusters, leading to a sharp decline in generalization when poisoning rates vary or trigger patterns are unknown; third, methods such as Spectral Signatures rely on SVD to extract global principal components, allowing anomalous signals to be overwhelmed by benign features at low poisoning rates, whereas AC assumes a predefined clustering structure and cluster-size distribution and therefore struggles to reliably uncover the intrinsic homogeneous anomaly patterns shared by poisoned samples.

In summary, input-level and model-level defenses focus on inference-time interception and post-training model repair, respectively, and therefore cannot prevent the model from learning poisoned features during training. By contrast, dataset-level defenses detect and purify training samples to block backdoor implantation at its source, providing a more direct defense with controllable deployment costs. However, existing methods remain deficient in cross-layer feature utilization, parameter adaptivity, and robustness against complex and stealthy attacks. Therefore, targeting source-level dataset purification, this paper proposes a backdoor sample detection method that integrates cross-layer extreme-value representations with adaptive particle-swarm-based weight optimization to overcome the aforementioned limitations.

3 Threat Model and Problem Definition

3.1 Threat Model

Attacker Model: This study assumes that the attacker can manipulate a subset of the training data and inject a certain number of malicious samples into the original training set during data collection or collaboration. The attacker can freely design the trigger pattern, such as a visible pattern, frequency-domain perturbation, or spatial warping, and may either modify the labels of poisoned samples for dirty-label attacks or preserve their original labels for clean-label attacks, thereby constructing three representative attack paradigms: All-to-One (A2O), All-to-All (A2A), and Untargeted (UT). However, the attacker cannot interfere with the defender's subsequent model-training process or modify the model architecture, training hyperparameters, or data-preprocessing strategy.

Defender Model: This study assumes that the defender is the model-training party and has access only to an untrusted training dataset that may have been contaminated. The defender has no prior knowledge of whether a backdoor exists, the specific trigger form, the attack target label, or the locations of poisoned samples, and does not rely on an additional clean validation set. The defender can train the model according to the standard procedure and use the hidden-layer feature responses of training samples to detect, screen, and purify the dataset without prespecifying the attack type or poisoning rate.

3.2 Problem Definition

The training dataset is regarded as an untrusted sample set that may have been contaminated, as defined in Eq.(1):

$$D = \{(x_n, y_n)\}_{n=1}^{N} \tag{1}$$

Here, x_n denotes the n input sample, y_n denotes its corresponding class label, and N is the total number of training samples. Because the training data may originate from public datasets, third-party platforms, or multiparty collaborative collection processes, an attacker may inject a small number of trigger-bearing poisoned samples into the dataset. Accordingly, the training set D can be partitioned into a benign sample set D_{clean} and a poisoned sample set D_{poison}, as shown in Eq.(2):

$$D = D_{clean} \cup D_{poison},\ D_{clean} \cap D_{poison} = \varnothing \tag{2}$$

Here, D_{clean} denotes the set of unmodified benign samples, whereas D_{poison} denotes the set of backdoor samples constructed by the attacker through trigger injection and label manipulation. In practical defense scenarios, the defender has no knowledge of the trigger type, attack target label, poisoning rate, or locations of poisoned samples and does not rely on an additional clean validation set; therefore, potential poisoned samples must be identified without attack prior knowledge.

The objective of this study can be formalized as follows: given an untrusted training set D and a classification model F, a poisoned-sample detection mechanism is constructed. The mechanism extracts cross-layer extreme-value anomaly features from the hidden-layer responses of training samples and combines layer-weight optimization with density-based clustering to enhance the separability between poisoned and benign samples in the feature space, ultimately producing the

detected poisoned-sample set as defined in Eq.(3):

$$D_{poison} \acute{I} D \tag{3}$$

4 Cross-Layer Extreme Activation Representation for Backdoor Sample Detection

4.1 Overall Architecture of the CLEAR Method

To address the high stealthiness of data-poisoning backdoor attacks under low poisoning rates, complex trigger patterns, and the absence of a clean validation set, as well as the insufficient robustness of existing detectors, we propose a cross-layer extreme-value method for backdoor sample detection based on particle swarm weight optimization. The proposed method uses anomalous activation responses induced by backdoor triggers in hidden layers as discriminative evidence and integrates cross-layer feature alignment, extreme-value anomaly representation, adaptive layer-weight optimization, and density-based clustering to identify and remove poisoned samples from an untrusted training set without attack prior knowledge or additional clean data. The overall framework of CLEAR is illustrated in Figure 1.

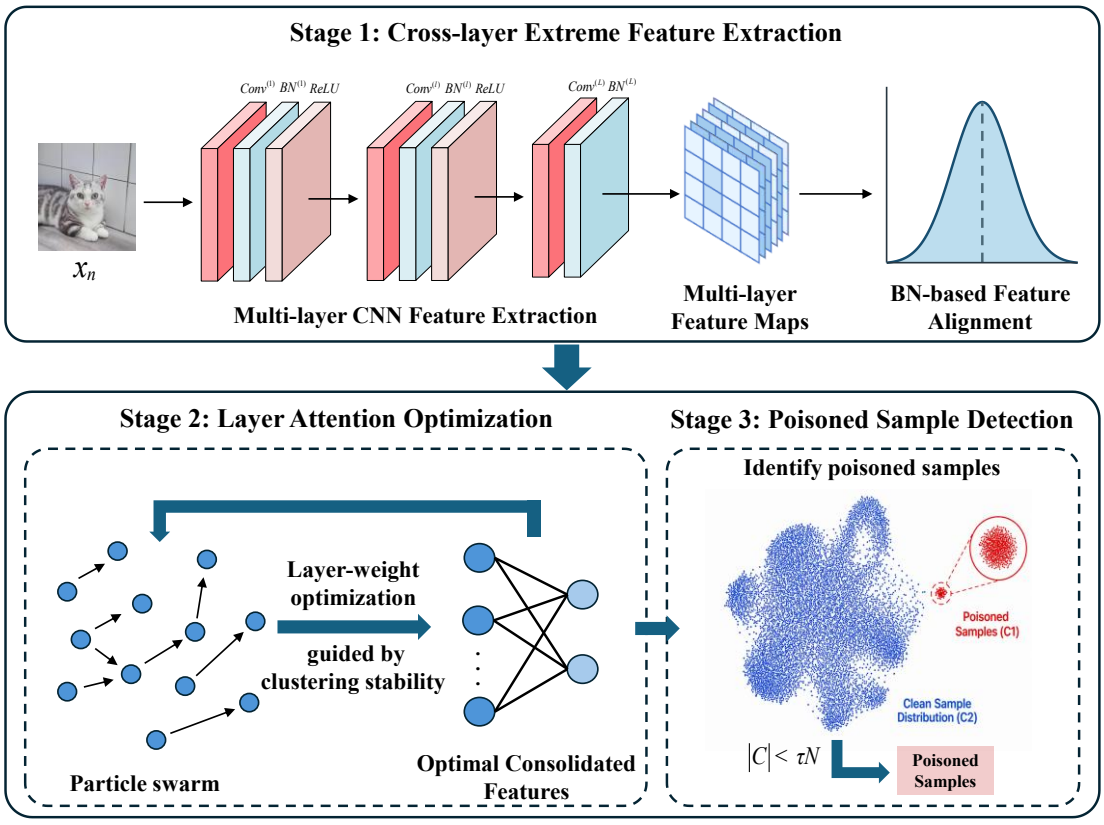


Figure 1 Overall Framework of the CLEAR Method

The CLEAR method mainly comprises the following three stages:（1）Cross-Layer Normalized

Extreme-Value Feature Extraction. By using BN layer statistics to align the distributions of feature maps across different hidden layers, we ensure that cross-layer features are on a unified scale. We then extract extreme responses from the feature maps of each channel to construct cross-layer latent representations that reflect abnormal activations in the trigger, thereby preserving weak but discriminative anomalous signals in backdoor samples.（2）Particle-Swarm-Optimization-Based Adaptive Layer-Weight Search. The attention weights assigned to different hidden layers are modeled as particle coordinates in the PSO search space, while the stability of HDBSCAN clusters obtained after UMAP-based low-dimensional projection is used as fitness feedback to guide the adaptive search for optimal layer weights. This procedure increases the contribution of hidden-layer features that are more sensitive to backdoor-related anomalous responses while reducing interference from irrelevant semantic noise.（3）Density-Clustering-Based Anomalous Microcluster Detection. Based on the optimized layer weights, the weighted latent representations of all training samples are reconstructed, followed by another round of UMAP dimensionality reduction and HDBSCAN density-based clustering. Because poisoned samples carrying the same trigger exhibit stronger anomalous consistency in the optimized feature space, they tend to form small but highly stable anomalous microclusters; accordingly, CLEAR identifies the microcluster with the highest clustering stability among those satisfying the size constraint as the poisoned-sample set.

4.2 Cross-Layer Normalized Extreme-Value Feature Extraction

To effectively suppress redundant background noise and amplify weak anomalous activations induced by backdoors, CLEAR introduces a cross-layer latent representation extraction mechanism. To address inconsistencies in numerical scales across layers and interference from redundant high-dimensional features, the proposed mechanism sequentially performs feature-distribution alignment and anomalous extreme-value extraction.

(1) Feature Alignment:

To eliminate internal covariate shift among activations from different layers and enable effective cross-layer fusion on a common scale, CLEAR employs BN layers to align the distributions of convolutional-layer outputs. Let the training dataset contain N samples, where x_n denotes the n- th training sample. Suppose that L candidate hidden layers are selected from the deep network for cross-layer feature fusion, where $l = 1,2,\mathrm{K},L$ denotes the hidden-layer index. The model F can be represented in a cascaded form, as shown in Eq. (4):

$$F = FC \circ f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)} \tag{4}$$

Here, FC denotes the final classification layer, $\circ$ denotes function composition in which the output of one layer serves as the input to the next, and $f^{(l)}$ denotes the l-th hidden layer. For sample x_n, let $A_n^{(l)}$ denote its raw activation map at the l-th convolutional layer, where $A_n^{(l)} \in \mathbb{R}^{C_l \times H_l \times W_l}$, and C_l, H_l and W_l denote the number of channels, height, and width, respectively.

CLEAR aligns the activation maps to a common scale using the running mean $\mu^{(l)} \in \mathbb{R}^{C_l}$ and variance $\left(\sigma^{(l)}\right)^2 \in \mathbb{R}^{C_l}$ of the corresponding BN layer, and the normalized response of the raw activation $a_{n,c,r,s}^{(l)}$ for sample x_n at channel c and spatial location (r,s) of layer l is given in Eq. (5):

$$\hat{a}_{n,c,r,s}^{(l)} = \frac{a_{n,c,r,s}^{(l)} - \mu_c^{(l)}}{\sqrt{(\sigma_c^{(l)})^2 + \delta}} \tag{5}$$

Here, δ is a smoothing constant introduced to prevent division by zero. This normalization maps activations from different hidden layers to statistically comparable, approximately zero-mean and unit-variance spaces, providing a consistent basis for subsequent layer-weight optimization.

(2) Extreme-Value Anomaly Extraction:

CLEAR extracts channel-wise extrema from the aligned feature maps along the spatial dimensions and uses them to represent the anomalous response of each channel. Specifically, CLEAR focuses on anomalous suppression responses induced by triggers in certain channels. Empirical observations indicate that backdoor triggers often induce suppression responses in certain channels that are substantially lower than those of benign samples, namely negative anomalies; therefore, the global minimum of each channel is selected as its anomalous feature. If a trigger induces positive anomalies, the corresponding channel-wise maximum can be used symmetrically.

For the aligned feature map $\hat{A}_n^{(l)}$ of the n-th sample at the l-th hidden layer, the representative anomalous feature of channel c is computed according to Eq. (6):

$$v_{n,l,c} = \min_{1 \le r \le H_l, 1 \le s \le W_l} (\hat{a}_{n,c,r,s}^{(l)}) \tag{6}$$

Here, $v_{n,l,c}$ denotes the extreme-value anomalous feature extracted from channel c of the l- th hidden layer for the n- th sample, $\hat{a}_{n,c,r,s}^{(l)}$ denotes the BN-aligned activation response, and r and s denote the spatial indices of the channel feature map.

Concatenating the extreme-value features of all channels in the l- th layer in channel order yields the anomalous feature vector of sample x_n at that layer, as shown in Eq. (7):

$$v_{n,l} = \{v_{n,l,1}, v_{n.l,2}, \frac{1}{4}, v_{n,l,C_l}\} \hat{I} R^{d_l} \tag{7}$$

Here, $v_{n,l}$ denotes the extreme-value anomaly feature vector of the n- th sample at the l- th hidden layer, and d_l denotes the resulting feature dimension after extreme-value extraction. Because CLEAR extracts the global minimum over the $H_l \acute{} W_l$ spatial dimensions of each channel and retains only one scalar per channel, the resulting feature dimension d_l equals the number of channels C_l.

Subsequently, the anomalous feature vectors of sample x_n from all L candidate hidden layers are concatenated to obtain an unweighted cross-layer latent representation, as shown in Eq. (8):

$$v_n = \text{Concat}(v_{n,1}, v_{n,2}, \frac{1}{4}, v_{n,L}) \hat{I} R^{d_{\text{total}}} \tag{8}$$

Here, v_n denotes the unweighted cross-layer latent representation of the n- th sample, $\text{Concat}(g)$ denotes vector concatenation, and d_{total} denotes the total feature dimension after cross-layer concatenation.

4.3 Adaptive Search for Layer Weights Based on Particle Swarm Optimization

Particle swarm optimization (PSO) does not require gradient information and performs a collaborative search based on the individual and collective experience of particles, enabling it to preserve global exploration while progressively converging toward a favorable weight configuration; therefore, it is suitable for adaptive layer-weight optimization without a clean validation set or attack prior knowledge. Accordingly, CLEAR encodes the layer-attention vector as a particle position in a continuous search space and optimizes it using cluster-stability feedback, thereby identifying the layer-weight configuration that most strongly accentuates poison-induced anomalies.

(1) Particle Representation and Initialization

The attention weights of the hidden layers are modeled as particle coordinates in an L

dimensional continuous search space. Let the swarm size be P, and let the position vector of the p- th particle at iteration t represent a candidate configuration of layer-attention weights, as defined in Eq. (9):

$$w_p^{(t)} = \left[w_{p,1}^{(t)}, w_{p,2}^{(t)}, \cdots, w_{p,L}^{(t)} \right], w_{p,l}^{(t)} \in [0,1], \sum_{l=1}^{L} w_{p,l}^{(t)} = 1 \tag{9}$$

Here, $w_{p,l}^{(t)}$ denotes the attention weight assigned to the l- th hidden layer by the p- th particle at iteration t. These weights are layer-level parameters shared across all samples.

At the initial stage of the algorithm, namely $t = 0$, CLEAR initializes all particle positions without bias using a uniform random distribution to ensure sufficient diversity for global exploration throughout the high-dimensional attention space. The l- th weight of particle p is independently sampled and subsequently normalized, as shown in Eqs. (10) and (11):

$$\hat{w}_{p,l}^{(0)} \sim U(0,1), l = 1,2,\cdots,L \tag{10}$$

$$w_{p,l}^{(0)} = \frac{\hat{w}_{p,l}^{(0)}}{\sum_{j=1}^{L} \hat{w}_{p,j}^{(0)}} \tag{11}$$

Here, $U(0,1)$ denotes the standard uniform distribution over the interval (0,1), and the initial velocity $u_p^{(0)}$ is set to zero. Because the initial particle positions are generated through uniform random sampling, the swarm retains adequate search diversity during the initial stage even when the initial velocities are zero.

(2) Weighted Representation Construction

Based on the position $w_p^{(t)}$ of particle p at iteration t, the extreme-value anomaly features extracted from different hidden layers for sample n are dynamically weighted and fused to obtain the weighted high-dimensional latent representation $\hat{v}_n^{(p,t)} \in \mathbb{R}^{d_{total}}$, as shown in Eq. (12):

$$\hat{v}_n^{(p,t)} = Concat(w_{p,1}^{(t)} \times v_{n,1}, w_{p,2}^{(t)} \times v_{n,2}, \cdots, w_{p,L}^{(t)} \times v_{n,L}) \tag{12}$$

For the complete training set, the weighted feature matrix corresponding to particle p at iteration t can be further constructed as shown in Eq. (13):

$$\hat{V}^{(p,t)} = \left[\hat{v}_1^{(p,t)}; \hat{v}_2^{(p,t)}; \cdots; \hat{v}_N^{(p,t)} \right] \in \mathbb{R}^{N \times d_{\text{total}}} \tag{13}$$

(3) Iterative Weight Optimization

At every iteration, fitness feedback is used to update the personal-best and global-best particle positions. The personal-best position of particle p and the corresponding swarm-wide global-best position are defined in Eqs. (14) and (15) , respectively:

$$\text{pbest}_p = \arg \max_{w_p^{(s)},\, 0 \le s \le t} F_p^{(s)} \tag{14}$$

$$\text{gbest} = \arg \max_{\text{pbest}_p,\, 1 \le p \le P} F_p \tag{15}$$

Here, pbest_p records the best position visited by particle p, whereas gbest denotes the particle position with the highest fitness in the current swarm. Specifically, after the fitness $F_p^{(t)}$ of particle p is evaluated at iteration t, the current weight position replaces pbest_p if $F_p^{(t)} > F(\text{pbest}_p)$; otherwise, the previous personal-best position is retained. Once all personal bests have been updated, the highest-fitness personal-best position is designated as the current gbest.

Each particle subsequently updates its velocity and position according to its personal historical best and the swarm-wide global best, as shown in Eq. (16):

$$\begin{aligned} u_p^{(t+1)} &= \omega^{(t)} u_p^{(t)} + c_1 r_1 \otimes \left(\text{pbest}_p - w_p^{(t)}\right) + c_2 r_2 \otimes \left(\text{gbest} - w_p^{(t)}\right) \\ \hat{w}_p^{(t+1)} &= w_p^{(t)} + u_p^{(t+1)} \end{aligned} \tag{16}$$

Here, $u_p^{(t)}$ denotes the velocity vector of particle p at iteration t, $\omega^{(t)}$ is the inertia weight, c_1 and c_2 are the cognitive and social learning factors, respectively, and $r_1, r_2 \sim U(0,1)$ are random variables. CLEAR employs a linearly decaying inertia weight to encourage global exploration early in the search and facilitate local convergence in later iterations, as defined in Eq. (17):

$$\omega^{(t)} = (\omega_{\max} - \omega_{\min}) \left(\frac{T - t}{T}\right) + \omega_{\min} \tag{17}$$

Here, T is the iteration budget, and $\omega_{\max}$ and $\omega_{\min}$ specify the upper and lower bounds of the inertia weight.

Because the layer-attention weights must be nonnegative and sum to one, CLEAR applies nonnegative rectification and normalization after each particle-position update, as shown in Eq. (18):

$$w_{p,l}^{(t+1)} = \frac{\max\left(\hat{w}_{p,l}^{(t+1)}, 0\right)}{\sum_{j=1}^{L} \max\left(\hat{w}_{p,j}^{(t+1)}, 0\right) + \delta} \tag{18}$$

Here, l indexes the layer being normalized, j is the summation index over all L layers, and δ ensures numerical stability.

After the predefined maximum number of iterations T, the particle-swarm optimization process converges. The final global-best particle position is then adopted as the optimal distribution of layer-attention weights, as shown in Eq. (19):

$$W^{*} = \text{gbest}^{(T)} = \left[w_1^{*}, w_2^{*}, \ldots, w_L^{*}\right] \tag{19}$$

Here, w_l^{*} is the optimized attention weight assigned to hidden layer l. These weights reflect the contributions of different hidden layers to backdoor-related anomalous responses, with a larger weight indicating that the corresponding layer is more important for amplifying anomalous consistency among poisoned samples.

4.4 Density-Based Anomalous Microcluster Detection

(1) Low-Dimensional Density Modeling

To provide PSO with a well-defined optimization direction, CLEAR incorporates the results of dimensionality reduction and density-based clustering into the fitness calculation process. For the weighted feature matrix $\hat{V}^{(p,t)}$ associated with each particle, UMAP is first employed to map it into a low-dimensional manifold space, as shown in Eq. (20):

$$Z^{(p,t)} = \text{UMAP}\left(\hat{V}^{(p,t)}\right) \in \mathrm{R}^{N \times q} \tag{20}$$

Here, q denotes the embedding dimension, which is typically set to 2. HDBSCAN is then applied to $Z^{(p,t)}$ to identify locally dense and highly stable anomalous microclusters.

HDBSCAN characterizes local density relationships among samples using the mutual reachability distance. The mutual reachability distance between samples a and b is formulated in Eq. (21):

$$d_{mreach}(a,b) = \max\left\{core_k(a), core_k(b), d(a,b)\right\} \tag{21}$$

Here, $core_k(a)$ denotes the minimum radius of a neighborhood centered at sample a that contains at least k samples, including aaa itself. Equivalently, the distance from a to its k- th nearest neighbor，whereas $d(a,b)$ is the Euclidean distance between a and b. Based on the mutual reachability distance, HDBSCAN constructs a minimum spanning tree and generates a hierarchical clustering structure, from which the stability score of each candidate cluster is computed according to variations in density. A higher cluster-stability score indicates that the cluster

is less likely to split as the density hierarchy changes and therefore exhibits stronger topological cohesion.

The collection of clusters returned by HDBSCAN for $Z^{(p,t)}$ is defined in Eq. (22):

$$\mathrm{C}^{(p,t)} = \left\{C_1^{(p,t)}, C_2^{(p,t)}, \frac{1}{4}, C_K^{(p,t)}\right\} \tag{22}$$

Here, $C_k^{(p,t)}$ denotes the k- th cluster obtained for particle p at iteration t.

(2) Fitness Calculation

To exclude the influence of the large dominant benign cluster, CLEAR imposes an anomalous-microcluster size constraint and selects candidate microclusters whose sample proportions are below the threshold τ, as shown in Eq. (23):

$$\mathrm{C}_{micro}^{(p,t)} = \left\{C_k^{(p,t)} \middle| \left|C_k^{(p,t)}\right| < \tau N\right\} \tag{23}$$

Here, $\tau \hat{\mathrm{I}} [0.05, 0.2]$ specifies the maximum admissible proportion of samples in an anomalous microcluster, while N denotes the total number of training samples. The fitness of particle p at iteration t is defined as the stability score of the most stable candidate anomalous microcluster, as shown in Eq. (24):

$$F_p^{(t)} = \max_{C_k^{(p,t)} \hat{\mathrm{I}}\, \mathrm{C}_{micro}^{(p,t)}} \mathrm{S}(C_k^{(p,t)}) \tag{24}$$

Here, $\mathrm{S}(C_k^{(p,t)})$ denotes the stability score assigned by HDBSCAN to the k- th candidate microcluster. If no microcluster satisfies the size constraint, the corresponding particle is assigned zero fitness. A larger fitness value $F_p^{(t)}$ indicates that the layer weights encoded by the particle more effectively amplify the anomalous consistency of backdoor samples, enabling them to form a more stable topologically collapsed microcluster in the low-dimensional space.

(3) Poisoned-Sample Detection

The optimized layer-attention weights W^* are applied to the cross-layer extreme-value features of the entire training set to construct the final globally weighted representations. The final weighted representation of sample n, denoted by $\hat{v}_n^* \hat{\mathrm{I}}\, \mathrm{R}^{d_{total}}$, is formulated in Eq. (25):

$$\hat{v}_n^* = Concat(w_1^* \times v_{n,1}, w_2^* \times v_{n,2}, \frac{1}{4}, w_L^* \times v_{n,L}) \tag{25}$$

Furthermore, the final weighted feature matrix corresponding to all training samples is constructed as shown in Eq. (26):

$$\hat{V}^{*}=\left[\hat{v}_{1}^{*};\hat{v}_{2}^{*};\cdots;\hat{v}_{N}^{*}\right]\in R^{N'\times d_{\text{data}}} \tag{26}$$

Finally, UMAP and HDBSCAN are reapplied to $\hat{V}^{*}$, and the most stable anomalous microcluster satisfying the size constraint is designated as the poisoned subset.

Table 1 summarizes the complete CLEAR detection procedure. The algorithm takes an untrusted training set, the model under examination, and the selected hidden layers as inputs and mainly comprises three stages: cross-layer extreme-value feature construction, layer-weight optimization, and anomalous-microcluster identification. CLEAR aligns candidate-layer activations using BN statistics and extracts channel-wise extrema to construct cross-layer representations; it then optimizes the particle-encoded layer weights according to the stability of HDBSCAN microclusters in the UMAP embedding space; finally, the most stable size-constrained anomalous microcluster obtained using the optimized weights is identified as the poisoned subset.

Table 1 Overall Detection Procedure of CLEAR

Input: untrusted training set $D=\{x_n\}_{n=1}^{N}$; model F; selected hidden layers $l=1,\ldots,L$, BN statistics $\mu^{(l)},\sigma^{(l)}$; particle number P; maximum iteration T; threshold τ.
Output: Detected backdoor set D_{poison}.
(1) **for** $n=1$ to N **do**
(2) **for** $l=1$ to L **do**
(3) extract $A_n^{(l)}$ from the l-th hidden layer, and obtain $\hat{A}_n^{(l)}$ by BN-based feature alignment with $\mu^{(l)},\sigma^{(l)}$, and α;
(4) compute channel-wise minimum responses from $\hat{A}_n^{(l)}$ to from $v_{n,l,c}$;
(5) **end for**
(6) construct $v_n=\text{Concat}(v_{n,1},v_{n,2},\cdots,v_{n,L})\in R^{d_{\text{total}}}$;
(7) **end for**
(8) initialize P particles $w_p^{(0)}=(w_{p,1}^{(0)},\ldots,w_{p,L}^{(0)})$ and velocities $u_p^{(0)}=0$;
(9) **for** $t=0$ to $T-1$ **do**
(10) **for** $p=1$ to P **do**
(11) construct $\hat{v}_n^{(p,t)}=Concat(w_{p,1}^{(t)}\times v_{n,1},\cdots,w_{p,L}^{(t)}\times v_{n,L})$, and obtain $\hat{V}^{(p,t)}$;
(12) obtain $Z^{(p,t)}=\text{UMAP}(\hat{V}^{(p,t)})$, and $C^{(p,t)}=\text{HDBSCAN}(Z^{(p,t)})$;
(13) select $C_{micro}^{(p,t)}=\{C_k^{(p,t)}\mid\lvert C_k^{(p,t)}\rvert<\tau N\}$;
(14) compute $F_p^{(t)}=\max_{C_k^{(p,t)}\in C_{micro}^{(p,t)}}S(C_k^{(p,t)})$, and update pbest_p;
(15) **end for**
(16) update gbest, $u_p^{(t+1)}$, and normalized $w_p^{(t+1)}$ by PSO;
(17) **end for**
(18) set $W^{*}=$ gbest, rebuild $\hat{V}^{*}$, and return the samples in the most stable size-constrained cluster after UMAP and HDBSCAN as D_{poison}.

5 Experiment

5.1 Experimental Settings

This section details the experimental protocol, including the implementation environment and hyperparameters, datasets and evaluation metrics, attack configurations, and baseline defenses.

5.1.1 Experimental Environment and Parameters

The experiments in this section were conducted on the Ubuntu operating system, with the algorithms implemented in Python 3.8 and PyTorch 1.10.1, while model training and testing were performed on an NVIDIA GeForce RTX 4090 GPU with 24 GB of memory.

Because the experiments involve key procedures such as particle swarm optimization and layer-attention weight updating, the relevant hyperparameters of CLEAR are uniformly configured to ensure experimental fairness and result reproducibility, as listed in Table 2.

Table 2 Hyperparameter Settings

Hyperparameter	Parameter Symbol	Setting
Particle swarm size	P	30
Maximum number of iterations	$T_{\max}$	50
Anomalous-microcluster size threshold	τ	0.15
Maximum inertia weight	$\omega_{\max}$	0.9
Minimum inertia weight	$\omega_{\min}$	0.4
Individual learning factor	c_1	2.0
Social learning factor	c_2	2.0

(1) The particle swarm size P is fixed at 30 to preserve population diversity in the low-dimensional layer-weight search space, thereby sustaining global exploration and mitigating premature convergence to local optima; (2) The maximum number of iterations $T_{\max}$ is set to 50, allowing the linearly decreasing inertia weight to guide the swarm from global exploration toward stable local convergence, with subsequent experiments confirming that additional iterations yield no appreciable performance gains; (3) The anomalous-microcluster size threshold τ is set to 0.15, a value within the recommended range of 0.05–0.2 that retains candidate clusters associated with typical poisoning rates of 5%–20% while excluding large benign-dominated clusters; (4) The inertia weight decreases linearly from $\omega_{\max} = 0.9$ to $\omega_{\min} = 0.4$, promoting broad exploration during

early iterations and increasingly refined exploitation during later iterations to achieve an effective balance between search diversity and convergence efficiency; (5) The individual and social learning factor are both set to $c_1 = c_2 = 2.0$, balancing the influence of the personal and global best solutions to ensure stable and convergent layer-weight updates.

5.1.2 Datasets and Evaluation Metrics

To evaluate the effectiveness of the proposed method under different scenarios and backdoor attack types, two representative image-classification datasets are employed: CIFAR-10 [39] and Tiny-ImageNet [40]. Both are general-purpose image-classification benchmarks widely used in computer vision and deep-learning research, offering broad category coverage, standardized data partitions, and readily reproducible experimental results, thereby enabling evaluation of the generalizability and robustness of the proposed method across different dataset scales and task complexities. The detailed characteristics of the datasets are presented in Table 3.

Table 3 Details of the Datasets Used in the Experiments

Dataset	Classes	Input Size	Training Samples	Test Samples
CIFAR-10	10	32×32×3	50000	10000
Tiny-ImageNet	200	64×64×3	100000	10000

To comprehensively evaluate the ability of the proposed method to detect backdoor-poisoned samples and the security performance of the sanitized model, CLEAR is evaluated from the perspectives of sample-level detection and model-level defense. Sample-level detection performance is measured using the true positive rate (TPR) and false positive rate (FPR), whereas model-level defense performance is assessed using benign accuracy (BA) and attack success rate (ASR).

(1) The TPR represents the proportion of correctly identified poisoned samples among all ground-truth poisoned samples. TPR is calculated as shown in Eq. (27):

$$\mathrm{TPR} = \frac{\text{Number of correctly identified poisoned samples}}{\text{Total number of poisoned samples}} \tag{27}$$

(2) The FPR represents the proportion of benign samples incorrectly classified as poisoned among all ground-truth benign samples. FPR is calculated as shown in Eq. (28):

$$\mathrm{FPR} = \frac{\text{Number of benign samples incorrectly identified as poisoned}}{\text{Total number of benign samples}} \tag{28}$$

(3) BA denotes the classification accuracy of the model on clean test samples and is used to

measure the ability of the sanitized model to recognize normal samples. BA is calculated as shown in Eq. (29):

$$\mathrm{BA} = \frac{\text{Number of correctly classified clean test samples}}{\text{Total number of clean test samples}} \tag{29}$$

(4) The ASR represents the proportion of triggered test samples that are successfully induced into a target or incorrect class and is used to measure the residual effectiveness of the backdoor attack. ASR is calculated as shown in Eq. (30):

$$\mathrm{ASR} = \frac{\text{Number of successfully attacked triggered test samples}}{\text{Total number of triggered test samples}} \tag{30}$$

In summary, TPR and FPR are primarily used to evaluate the ability of CLEAR to detect poisoned samples in the training set, with a higher TPR and lower FPR indicating better detection performance, whereas BA and ASR are mainly used to evaluate the security and utility of the model retrained on the sanitized dataset, with a higher BA and lower ASR indicating better defense performance.

5.1.3 Attack and Defense Settings

To comprehensively evaluate the detection performance of CLEAR, four representative backdoor attack methods are selected: BadNets, Blend, Trojan, and WaNet. Each attack is implemented according to its original specification, with $y_t = 0$ and a poisoning rate of 10%, to ensure a faithful and consistent evaluation. To assess generalizability across different attack objectives, we consider the A2O, A2A, and UT attack paradigms: (1) In the A2O setting, poisoned samples from all source classes are embedded with a common trigger and mapped to a single attacker-specified target label. (2) In the A2A setting, a one-to-one mapping is defined between source and target classes such that poisoned samples from class i are reassigned to a predetermined class $i¢$. (3) Unlike A2O and A2A, which explicitly direct triggered samples toward specified target labels, an untargeted attack does not specify a fixed target class.

Four representative defense baselines are selected for comparison: (1) AC[33]: performs binary clustering using last-layer activation features and identifies poisoned samples according to differences in cluster size; (2)SCALE-UP[38]: amplifies weak anomalous signals through robust covariance estimation and feature whitening; (3) AIBD[39]: performs outlier mining on intermediate-layer activations using an activation-based isolation forest without requiring label information; and (4) CCD[40]: separates poisoned and benign clusters in the representation space using a contrastive

clustering loss and encourages intracluster compactness and intercluster separation under unsupervised conditions. These methods cover different technical routes, including unsupervised activation analysis, black-box input detection, adversarial heuristic defense, and lightweight training-stage detection, enabling a comprehensive comparison with CLEAR across multiple defense paradigms and ensuring the comprehensiveness, balance, and credibility of the experimental evaluation.

5.2 Experimental Results and Analysis

This section analyzes the experimental results, including comparisons among defense methods and validation of the effectiveness of models trained on sanitized datasets. By comparing different methods in terms of detection performance and defense effectiveness, the effectiveness and robustness of CLEAR under complex backdoor attacks are evaluated.

5.2.1 Comparative Experiments on Defense Methods

Tables 4 and 5 report the TPR and FPR of the evaluated detection methods on CIFAR-10 and Tiny-ImageNet, respectively. Overall, compared with four mainstream backdoor defense methods, namely AC, SCALE-UP, AIBD, and CCD, CLEAR achieves superior detection performance in almost all attack scenarios, as analyzed below:

On the CIFAR-10 dataset, CLEAR achieves a TPR close to 100% and an FPR below 1% for the vast majority of attack types and attack modes. In contrast, the AC method fails almost completely in A2A and UT modes, with TPR generally approaching 0, while SCALE-UP experiences a sharp decline in detection performance under dynamic trigger attacks such as WaNet, and its FPR consistently remains above 20%. AIBD maintains a high detection rate but has a significantly higher FPR than CLEAR, while CCD performs well under simple attacks but has limited detection capabilities in complex modes such as A2A and UT.

On the more complex Tiny-ImageNet dataset, CLEAR achieves an average TPR of 99.54% and an average FPR of only 0.03% across ten attack configurations, whereas AC, SCALE-UP, AIBD, and CCD obtain average TPRs of 11.96%, 28.36%, 39.21%, and 41.23% and average FPRs of 0.22%, 6.72%, 1.93%, and 2.62%, respectively, demonstrating the strong adaptability of CLEAR to complex data distributions and high-dimensional feature spaces, particularly under challenging A2A and UT attacks.

Table 4 Detection Performance of the ResNet-18 Model on CIFAR-10 (%)

Dataset	Attack Type	Defense⟶	AC		SCALE-UP		AIBD		CCD		CLEAR (ours)	
		Attack↓	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
		No Attack	—	7.29	—	31.44	—	9.41	—	3.26	—	**0.05**
CIFAR-10	A2O	BadNets	99.24	**0.00**	**100.00**	19.98	**100.00**	2.14	**100.00**	**0.00**	**100.00**	**0.00**
		Blend	0.00	6.08	99.49	26.33	99.26	1.12	99.98	0.06	**100.00**	**0.03**
		Trojan	99.46	0.06	**100.00**	13.70	**100.00**	0.39	99.96	**0.01**	**100.00**	**0.01**
		WaNet	90.36	**0.00**	31.91	27.64	**99.34**	1.33	96.46	0.02	98.11	**0.00**
	A2A	BadNets	0.04	10.02	20.76	18.98	6.97	5.73	9.34	1.08	**100.00**	**0.00**
		Blend	0.00	10.02	7.52	23.19	18.22	2.61	19.36	3.42	**99.43**	**0.00**
		Trojan	0.00	9.96	28.26	12.03	10.51	0.74	23.39	1.71	**99.92**	**0.06**
		WaNet	0.00	9.96	11.55	21.27	13.85	0.59	12.62	10.83	**95.66**	**0.00**
	UT	BadNets	0.66	9.94	10.07	25.48	4.62	7.75	0.13	0.08	**99.91**	**0.02**
		Blend	0.00	10.03	22.34	31.64	12.93	14.26	12.55	20.04	**100.00**	**0.45**
		Trojan	0.04	10.02	13.19	9.81	8.61	3.20	1.67	3.19	**96.02**	**0.30**
		WaNet	0.30	9.90	9.36	28.05	8.80	3.85	8.50	8.17	**81.46**	**1.04**

Table 5 Detection Performance of the ResNet-18 Model on Tiny-ImageNet (%)

Dataset	Attack Type	Defense⟶	AC		SCALE-UP		AIBD		CCD		CLEAR (ours)	
		Attack↓	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
		No Attack	—	0.56	—	4.59	—	12.50	—	0.06	—	**0.05**
Tiny-ImageNet	A2O	BadNets	0.00	0.17	94.72	28.02	98.45	3.56	**100.00**	0.86	**100.00**	**0.00**
		Blend	46.63	**0.00**	91.66	5.00	96.12	2.85	97.15	1.12	**100.00**	**0.03**
		Trojan	0.00	0.02	82.89	5.94	**99.99**	1.27	97.80	0.95	99.98	**0.01**
		WaNet	62.16	**0.00**	0.00	5.21	95.88	3.10	**96.42**	1.85	96.11	**0.00**
	A2A	BadNets	0.07	0.51	2.58	3.64	1.02	5.63	4.15	2.30	**100.00**	**0.00**
		Blend	0.00	0.47	0.33	3.20	0.01	0.51	5.22	3.16	**99.43**	**0.00**
		Trojan	0.00	0.66	1.45	3.92	0.07	0.84	3.85	1.98	**99.92**	**0.06**
	UT	BadNets	0.38	0.40	3.98	4.82	0.49	0.03	2.40	4.51	**99.91**	**0.02**
		Blend	0.68	**0.00**	2.16	2.83	0.00	0.46	3.11	5.60	**100.00**	0.15
		Trojan	9.64	**0.00**	3.87	4.66	0.06	1.02	2.15	3.83	**100.00**	**0.00**

5.2.2 Analysis of Model Effectiveness after Detection

To further verify the practical effectiveness of CLEAR in improving secure model retraining after poisoned-sample detection, models are retrained from scratch on the sanitized datasets, and their BA and ASR are comprehensively evaluated on CIFAR-10 and Tiny-ImageNet. The results presented in Tables 6 and 7 demonstrate that models retrained on datasets sanitized by CLEAR maintain benign accuracy comparable to that of the original clean models while reducing the attack success rate to an extremely low level close to zero.

On CIFAR-10, under the A2O, A2A, and UT attack modes involving BadNets, Blend, Trojan,

and WaNet, the benign accuracy of the models retrained after CLEAR sanitization remains stable between 93% and 95%, which is comparable to that of the original clean model. CLEAR reduces ASR to below 10% in most attack scenarios, substantially outperforming the comparison methods; however, the ASR remains relatively high at 42.71% for the Blend attack under the UT setting, indicating that this attack leaves a stronger residual backdoor effect that warrants further investigation. On the larger and more complex Tiny-ImageNet dataset, CLEAR continues to deliver reliable cleansing and retraining results. Its positive accuracy is comparable to that of the original clean model, with attack success rates mostly kept below 3% and often approaching 0, whereas other comparison methods exhibit significant performance degradation or defense failure. This fully validates CLEAR's robustness and practicality in large-scale, high-complexity scenarios.

In summary, CLEAR not only accurately identifies and removes backdoor samples from the training set but also, by leveraging the high-quality, cleaned dataset, ensures that the model retains its original performance after being retrained from scratch and significantly suppresses the backdoor effect in most attack scenarios. It thus provides a comprehensive and effective solution for the secure training and reliable deployment of deep learning models using untrusted data sources.

Table 6 Effectiveness of Secure Training from Scratch on the Sanitized CIFAR-10 Dataset

Dataset	Attack Type	Defense⟶	Benign Model		No Defense		AC		SCALE-UP		AIBD		CCD		CLEAR (ours)	
		Attack↓	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
CIFAR-10	A2O	BadNets	94.16	0.61	93.57	100.00	93.53	0.79	91.17	0.18	92.44	0.43	92.08	**0.16**	**94.06**	0.58
		Blend	94.16	0.38	93.43	100.00	93.42	0.01	87.74	90.74	90.05	1.28	91.57	1.34	**94.25**	**0.21**
		Trojan	94.16	1.61	92.99	100.00	**93.46**	29.38	89.86	**0.51**	92.17	0.69	90.86	1.50	93.10	2.00
		WaNet	94.16	0.71	93.05	99.11	92.61	1.04	88.43	97.33	93.39	**0.75**	90.52	3.31	**95.22**	0.83
	A2A	BadNets	94.16	0.48	93.04	86.32	82.73	67.45	88.29	64.39	91.37	69.63	91.95	60.34	**93.66**	**0.11**
		Blend	94.16	8.09	92.74	86.46	82.89	90.42	89.76	86.02	**93.77**	74.57	91.26	83.23	92.90	**9.03**
		Trojan	94.16	1.36	93.82	91.87	82.47	91.39	90.57	86.28	90.61	80.03	93.31	89.72	**93.42**	**2.02**
		WaNet	94.16	0.71	92.95	87.66	83.32	85.91	81.76	83.30	90.45	88.16	**92.20**	80.90	92.08	**0.98**
	UT	BadNets	94.16	0.50	92.01	93.17	80.99	94.03	85.04	94.20	89.63	91.58	91.75	90.60	**94.93**	**0.94**
		Blend	94.16	69.55	93.38	81.03	82.87	86.12	87.99	87.79	87.82	88.78	91.59	89.72	**93.28**	**42.71**
		Trojan	94.16	5.81	91.76	91.97	83.10	90.11	90.13	87.31	93.02	88.39	90.04	84.99	**94.10**	**7.60**
		WaNet	94.16	2.47	92.45	78.52	82.58	74.29	82.36	80.03	90.33	79.71	**92.37**	73.61	91.82	**1.77**

Table 7 Effectiveness of Secure Training from Scratch on the Sanitized Tiny-ImageNet Dataset

Dataset	Attack Type	Defense⟶	Benign Model		No Defense		AC		SCALE-UP		AIBD		CCD		CLEAR (ours)	
		Attack↓	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
Tiny-	A2O	BadNets	67.75	0.01	61.19	97.36	42.12	98.00	65.09	2.33	66.62	0.07	**67.46**	1.46	67.42	**0.01**

ImageNet		Blend	67.75	0.07	66.82	100.00	67.35	83.28	**70.86**	100.00	70.27	1.41	68.33	0.75	68.28	**0.27**
		Trojan	67.75	0.00	67.03	100.00	65.29	100.00	70.25	99.02	70.03	**0.00**	66.47	3.54	**71.30**	**0.00**
		WaNet	67.75	0.03	61.96	100.00	**70.12**	99.05	63.34	99.13	64.56	3.59	65.51	1.36	69.15	**0.14**
	A2A	BadNets	67.75	1.15	68.22	19.88	68.44	17.38	64.36	15.26	64.73	9.83	**69.72**	10.59	66.82	**2.09**
		Blend	67.75	0.77	70.30	46.63	70.58	37.74	65.11	33.21	68.36	29.96	64.89	39.46	**70.70**	**0.63**
		Trojan	67.75	0.65	67.81	31.29	68.93	18.85	**69.22**	15.58	62.90	30.04	66.07	22.63	69.02	**0.77**
	UT	BadNets	67.75	2.77	70.02	93.04	65.99	89.54	61.78	91.86	67.94	97.72	68.88	90.03	**70.22**	**2.59**
		Blend	67.75	58.38	69.32	99.87	**70.35**	99.60	69.46	97.55	68.23	99.06	68.35	98.10	69.86	52.30

5.2.3 Analysis of the Effect of Model Architecture

To further verify the cross-architecture generalizability and robustness of the proposed method and prevent the experimental results from being affected by a particular model structure, additional experiments are conducted using different model architectures. VGG-19 and MobileNet v2, which differ substantially in architecture, are selected as the test models, and all experiments are conducted on CIFAR-10. As shown in Table 8, CLEAR maintains stable detection performance under both architectures, with TPR approaching 100% and FPR remaining at a low level. However, the relatively shallow structure of VGG-19 may reduce feature discriminability, causing some benign samples to be incorrectly classified under untargeted attacks, although the overall TPR remains at 100%, and the FPR can be reduced by appropriately adjusting the anomalous-microcluster threshold. These results demonstrate that CLEAR does not depend on a particular network structure and can effectively capture anomalous features of backdoor samples across different model architectures, indicating favorable cross-architecture robustness and broad applicability.

Table 8 Detection Performance under Different Model Architectures

Attack Mode↓	Models→ Attacks↓	VGG-19		MobileNet_v2		ResNet-18	
		TPR (%)	FPR (%)	TPR (%)	FPR (%)	TPR (%)	FPR (%)
A2O	BadNets	100.00	0.00	100.00	0.00	100.00	0.00
	Blend	100.00	0.04	100.00	0.03	100.00	0.03
	Trojan	99.98	0.00	99.96	0.01	100.00	0.01
A2A	BadNets	100.00	0.01	100.00	0.00	100.00	0.00
UT	BadNets	100.00	9.89	100.00	0.00	99.91	0.02

5.2.4 Parameter Sensitivity Analysis

To investigate the robustness of the CLEAR defense framework to its core parameters, hyperparameter sensitivity experiments are conducted on CIFAR-10 and Tiny-ImageNet. The

experiments examine the effects of the particle swarm size P、the maximum number of iterations $T_{\max}$, and the prior anomalous-microcluster size threshold τ on the final defense performance. The experimental results are shown in Figure 2.

The results demonstrate that the optimization parameters P and $T_{\max}$ jointly determine the convergence quality and computational efficiency of the algorithm in the high-dimensional attention space：When either P or $T_{\max}$ is excessively small, the algorithm is prone to becoming trapped in a local optimum or exhibiting fluctuations in detection performance because of incomplete convergence; When the parameters reach $P = 30$ and $T_{\max} = 50$, the swarm-intelligence-based optimization process converges rapidly and accurately identifies the optimal feature-projection direction. Under this configuration, TPR reaches its peak and stabilizes, while an optimal trade-off between detection performance and computational cost is achieved, with further increases in these parameters providing only marginal benefits.

Regarding the detection threshold, because the optimized layer-attention weights strongly amplify backdoor-related features, TPR avoids an abrupt decline and remains above 95% as long as the value of τ accommodates the actual poisoning rate; Meanwhile, when τ varies within the broad tolerance range of 0.10–0.20, FPR consistently remains below 1%. In summary, CLEAR does not depend on stringent parameter tuning and exhibits excellent and stable defense performance under both the default configuration and reasonable parameter variations.

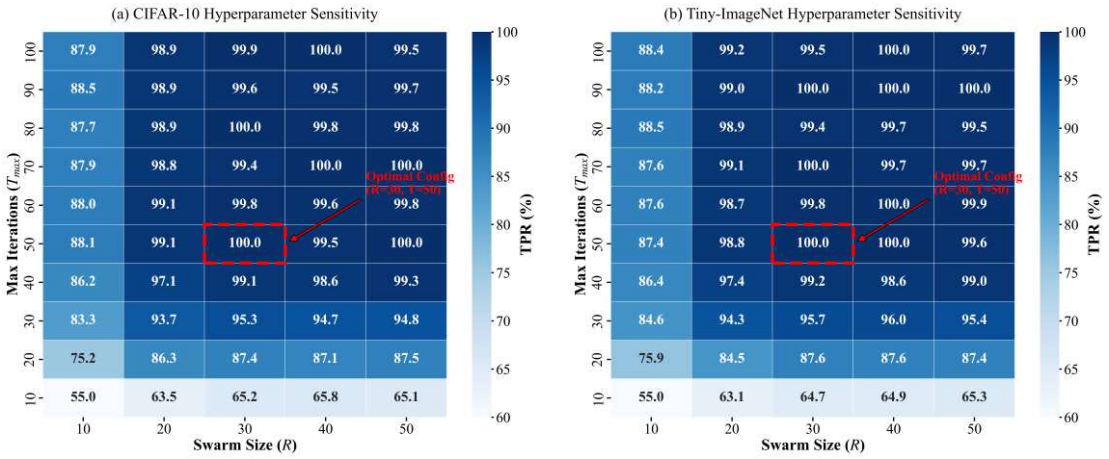


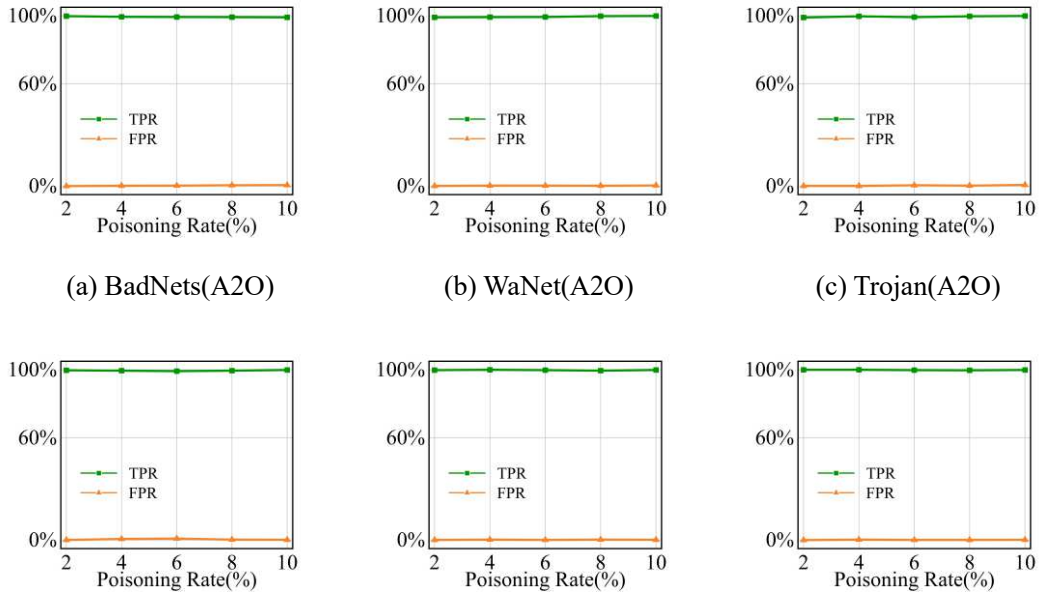
Figure 2 Effects of Core Parameters on Detection Performance

5.2.5 Analysis of the Effect of Poisoning Rate

Under extremely low poisoning rates or weak trigger intensities, the backdoor effect becomes less prominent in the model's internal representations, weakening the feature separability between

poisoned and benign samples and consequently destabilizing existing poisoned-sample detection methods. To evaluate the defense robustness of CLEAR under low-poisoning-rate conditions, six representative attack configurations—BadNets (A2O), WaNet (A2O), Trojan (A2O), BadNets (A2A), WaNet (A2A), and BadNets (UT)—are evaluated on CIFAR-10, with the poisoning rate progressively increased from 2% to 10%. Meanwhile, to avoid incorrectly interpreting unsuccessful attacks as successful defenses, detection is evaluated only when the attack success rate remains sufficiently high at each poisoning rate, thereby ensuring the validity of the low-poisoning-rate experiments.

As shown in Figure 3, CLEAR maintains stable and effective detection performance across all poisoning rates, with TPR approaching 100% and FPR approaching 0%. These results demonstrate that CLEAR effectively mitigates feature confusion caused by low poisoning rates and exhibits excellent robustness against adaptive stealthy attacks and challenging extreme scenarios.



(a) BadNets(A2O) (b) WaNet(A2O) (c) Trojan(A2O)

(d) BadNets(A2A) (e) WaNet(A2A) (f) BadNets(UT)

Figure 3 Effect of Different Poisoning Rates on Detection

5.3 Ablation Experiments

To verify the necessity and effectiveness of the core modules in CLEAR, ablation experiments are conducted by individually removing or replacing key components. Three types of ablation experiments are designed: (1) layer-weight optimization ablation (w/o PSO), in which the dynamically optimized PSO weights are replaced with static uniform weights; (2) anomalous-

feature extraction ablation (w/o Extremum), in which extreme-value extraction is replaced with conventional global average pooling (GAP); and (3) clustering and fitness-evaluation ablation (w/o HDBSCAN), in which topology-stability-based HDBSCAN is replaced with K-Means. For ease of analysis, this section primarily presents ablation results on CIFAR-10 under four A2O backdoor attacks—BadNets, Blend, Trojan, and WaNet—and uses TPR and FPR as evaluation metrics to quantify the contribution of each module to defense performance. The experimental results are shown in Figure 4.

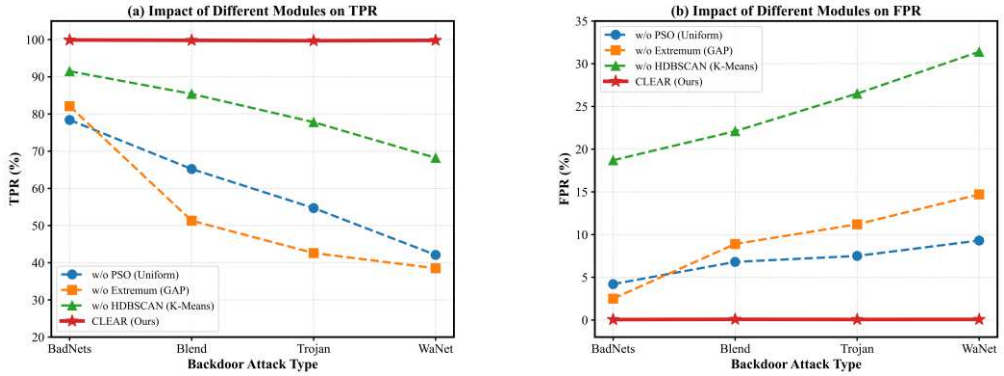


Figure 4. Results of the Ablation Experiments

(1) Ablation of layer-weight optimization: Different hidden layers of a deep neural network exhibit substantial differences in their sensitivity to trigger patterns. To verify the necessity of PSO-based optimization, the particle swarm optimization process is removed, and the features extracted from all hidden layers are directly assigned equal weights. The results show that after removing PSO-based weight optimization, the TPRs under Trojan and WaNet decrease from 99.9% and 95.7% for the complete model to 54.7% and 42.1%, respectively, while the corresponding FPRs increase to 7.5% and 9.3%. This degradation occurs because uniform weights fail to emphasize the key layers exhibiting strong backdoor responses and introduce irrelevant semantic information from other layers, causing weak backdoor signals to be diluted during cross-layer fusion. These results demonstrate that adaptive layer-weight optimization effectively strengthens anomalous responses in key layers and is essential for improving the detection of complex and stealthy attacks.

(2) Ablation of the extreme-value feature extraction mechanism: To verify the ability of extreme-value responses to represent backdoor anomalies, extreme-value extraction is replaced with global average pooling for spatial feature aggregation. The results show that after replacing extreme-value extraction with global average pooling, the ablated variant still achieves a TPR of 82.1%

against BadNets, whose trigger features are conspicuous; however, under stealthy attacks such as Blend and WaNet, whose triggers are highly integrated with the image background, the TPR decreases to 51.3% and 38.5%, respectively, while the FPR exceeds 10% in both cases. These results indicate that global average pooling averages the spatial responses and weakens local extreme activations induced by triggers, whereas extreme-value extraction better preserves salient trigger-induced responses and is particularly effective in capturing weak anomalous signals generated by stealthy triggers.

(3) Ablation of the density-based clustering mechanism: To verify the role of HDBSCAN in anomalous-microcluster identification, HDBSCAN is replaced with K-Means, and the silhouette coefficient is used as the fitness measure. The results show that the FPR increases sharply to 26.5% and 31.4% under Trojan and WaNet, respectively, and although the TPR remains between 70% and 80%, a large number of benign samples are misclassified. This degradation occurs because the optimized feature space exhibits a non-convex manifold structure and a substantial size imbalance between the poisoned microcluster and the dominant benign cluster, with a poisoning rate of only 10%, whereas the spherical partitioning assumption of K-Means forces benign samples near the distribution boundary into the anomalous cluster. By contrast, HDBSCAN does not require a predefined number of clusters and can identify arbitrarily shaped anomalous microclusters based on density reachability and cluster stability, making it more suitable for complex and imbalanced feature distributions.

In summary, the three ablation experiments verify the necessity of PSO-based layer-weight optimization, extreme-value feature extraction, and HDBSCAN density-based clustering, while demonstrating that their combined operation strengthens weak backdoor signals, improves sample separability, and reduces the misclassification of benign samples.

6 Conclusion

To address the reliance of existing backdoor sample detection methods on additional clean data, their difficulty in capturing weak anomalies, and their high false-detection rates under complex attacks, this paper proposes CLEAR, a cross-layer extreme-value backdoor sample detection method based on adaptive weight optimization. The proposed method employs BN statistics to align the feature distributions of different hidden layers, preserves trigger-induced local anomalies through channel-wise extreme-value responses, and combines particle swarm search, UMAP

manifold projection, and HDBSCAN clustering to adaptively determine cross-layer feature weights and identify anomalous microclusters, thereby detecting poisoned samples without a clean validation set or prior knowledge of the attack. Experimental results on CIFAR-10 and Tiny-ImageNet demonstrate that CLEAR provides stable detection performance against BadNets, Blend, Trojan, and WaNet under the A2O, A2A, and UT attack modes, achieving average TPRs of 97.54% and 99.54% and average FPRs of only 0.16% and 0.03%, respectively. Models retrained on the sanitized datasets maintain classification accuracy comparable to that of benign models while effectively reducing the attack success rate. Compared with AC, SCALE-UP, AIBD, and CCD, CLEAR exhibits higher detection accuracy and stronger defense robustness under complex attack scenarios.

Future work will further investigate the applicability of CLEAR to tasks involving text and speech and validate its generalizability across different network architectures, including DenseNet and Transformer. In addition, its detection performance under more complex scenarios, including lower poisoning rates, multiple triggers, multiple targets, and clean-label attacks, will be investigated to further improve the applicability and stability of the method.

References

[1] Khadem N, Nashir A, Rahmatyar S. The role of deep learning in advancing computer vision applications: a comprehensive systematic review[J]. Journal of Advanced Computer Knowledge and Algorithms, 2026, 3(1): 1-8.

[2] Zhao J, Wu Y, Deng R, et al. A survey of autonomous driving from a deep learning perspective[J]. ACM Computing Surveys, 2025, 57(10): 1-60.

[3] Shou Y, Meng T, Ai W, et al. A comprehensive survey on multi-modal conversational emotion recognition with deep learning[J]. ACM Transactions on Information Systems, 2026, 44(2): 1-48.

[4] Ahlawat H, Aggarwal N, Gupta D. Automatic speech recognition: a survey of deep learning techniques and approaches[J]. International Journal of Cognitive Computing in Engineering, 2025, 6: 201-237.

[5] Shen L, Sun Y, Yu Z, et al. On efficient training of large-scale deep learning models[J]. ACM Computing Surveys, 2024, 57(3): 1-36.

[6] He Y, Meng G, Chen K, et al. Towards security threats of deep learning systems: a survey[J]. IEEE Transactions on Software Engineering, 2020, 48(5): 1743-1770.

[7] Zhu Z X, Wang R, Zou C, et al. The victim and the beneficiary: exploiting a poisoned model to train a clean model on poisoned data[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2023: 155-164.

[8] Hanif M A, Chattopadhyay N, Ouni B, et al. Survey on backdoor attacks on deep learning: current trends, categorization, applications, research challenges, and future prospects[J]. IEEE Access, 2025, 13: 93190-93221.

[9] Liu Y T, Xie Y, Srivastava A. Neural trojans[C]//2017 IEEE International Confere nce on Computer Design. Piscataway: IEEE, 2017: 45-48.

[10] Javaheripi M, Samragh M, Fields G, et al. CleaNN: Accelerated Trojan Shield for Embedded Neural Networks[C]. The 39th Inter national Conference on Computer -Aided Design, 2020: 1-9.

[11] Jiang W, Wen X Y, Zhan J Y, et al. Interpretability-guided defense against backd oor attacks to deep neural networks[J]. IEEE Transactions on Computer-Aided De sign of Integrated Circuits and Systems, 2022, 41(8): 2611-2624.

[12] Sun B, Sun J, Pham L H, et al. Causality-Based Neural Network Repair[C]. 2022 IEEE/ACM 44th International Conference on Software Engineering, 2022: 338-3 49.

[13] Hayase J, Kong W H, Somani R, et al. SPECTRE: Defending Against Backdoor Attacks Using Robust Statistics[EB/OL]. arXiv preprint arXiv:2104.11315, 2021.

[14] Li Y G, Lyu X X, Koren N, et al. Anti-Backdoor Learning: Train ing Clean Mo dels on Poisoned Data[EB/OL]. 2021: arXiv: 2110. 11571.

[15] GOLDBLUM M, TSIPRAS D, XIE C, et al. Dataset security for machine learnin g: Data poisoning, backdoor attacks, and defenses[J]. IEEE Transactions on Patter n Analysis and Machine Intelligence, 2023, 45(2): 1563-1580.

[16] Luo T, Peng H, Fu A, et al. Just a little human intelligence feedback! Unsupervis ed learning assisted supervised learning data poisoning based backdoor removal[J]. Computer Communications, 2025, 233: 108052.

[17] LI Y, JIANG Y, LI Z, et al. Backdoor learning: A survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(1): 5-22.

[18] Gu T, Liu K, Dolan-Gavitt B, et al. Badnets: evaluating backdooring attacks on deep neural networks[J]. IEEE Access, 2019, 7: 47230-47244.

[19] Chen X, Liu C, Li B, et al. Targeted backdoor attacks on deep learning systems using data poisoning[EB/OL]. arXiv preprint arXiv:1712.05526, 2017.

[20] Nguyen A, Tran A. Wanet--imperceptible warping-based backdoor attack[J]. arXiv preprint arXiv:2102.10369, 2021.

[21] Liu Y Q, Ma S Q, Aafer Y, et al. Trojaning attack on neural networks[C]//Proceedings of the 25th Annual Network and Distributed System Security Symposium. San Diego: Internet Society, 2018: 1-18.

[22] Xue M F, Wu Y H, Ni S F, et al. Untargeted backdoor attack against deep neural networks with imperceptible trigger[J]. IEEE Transactions on Industrial Informatics, 2024, 20(3): 5004-5013.

[23] Barni M, Kallas K, Tondi B. A new backdoor attack in cnns by training set corruption without label poisoning[C]//2019 IEEE International Conference on Image Processing. Piscataway: IEEE, 2019: 101-105.

[24] BAI Y, XING G, WU H, et al. Backdoor attack and defense on deep learning: A survey[J]. IEEE Transactions on Computational Social Systems, 2025, 12(1): 404-434.

[25] Gao Y, Xu C, Wang D, et al. Strip: A defence against trojan attacks on deep neural networks[C]//Proceedings of the 35th Annual Computer Security Applications Conference, 2019: 113-125.

[26] Chou E, Tramer F, Pellegrino G. SentiNet: Detecting localized universal attacks against deep learning[C]//2020 IEEE Security and Privacy Workshops. Piscataway: IEEE, 2020: 48-54.

[27] Li Y, Zhai T, Jiang Y, et al. Backdoor attack in the physical world[EB/OL]. arXiv preprint arXiv:2104.02361, 2021.

[28] Liu K, Dolan-Gavitt B, Garg S. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks[C]//Research in Attacks, Intrusions, and Defenses. Cham: Springer, 2018: 273-294.

[29] Kolouri S, Saha A, Pirsiavash H, et al. Universal litmus patterns: Revealing backdoor attacks in cnns[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2020: 301-310.

[30] Huang X, Alzantot M, Srivastava M. Neuroninspect: Detecting backdoors in neural networks via output explanations[J]. arXiv preprint arXiv:1911.07399, 2019.

[31] Xiang Z, Miller D J, Kesidis G. Post-training detection of backdoor attacks for two-class and multi-attack scenarios[J]. arXiv preprint arXiv:2201.08474, 2022.

[32] Tran B, Li J, Madry A. Spectral signatures in backdoor attacks[J]. Advances in neural information processing systems, 2018, 31.

[33] Chan A, Ong Y S. Poison as a cure: Detecting & neutralizing variable-sized backdoor attacks in deep neural networks[J]. arXiv preprint arXiv:1911.08040, 2019.

[34] Chen B, Carvalho W, Baracaldo N, et al. Detecting backdoor attacks on deep neural networks by activation clustering[EB/OL]. arXiv preprint arXiv:1811.03728, 2018.

[35] Zeng Y, Park W, Mao Z M, et al. Rethinking the Backdoor Attacks' Triggers: A Frequency Perspective[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2021: 16453-16461.

[36] Guo J, Li Y, Chen X, et al. SCALE-UP: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency[EB/OL]. arXiv preprint arXiv:2302.03251, 2023.

[37] Yin J, Wang W, Lin W, et al. Adversarial-inspired backdoor defense via bridging backdoor and adversarial attacks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. AAAI Press, 2025: 9508-9516.

[38] Zhuang Z, Ding Y, Zheng S, et al. Cracking the Code of Backdoor Attacks with Confidence Consistency[J]. IEEE Transactions on Information Forensics and Security, 2026.

[39] Meyer L, Dijkstra J, Tebeck S, et al. Deep neural network inference on an integrated, reconfigurable photonic tensor processor[J]. Nature Communications, 2026, 17(1): 3396.

[40] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2016: 770-778.

Funding

This work was supported by the Henan Province Key R&D Project (Grant No. 251111210300), the Science and Technology Research Project of Henan Province (Grant No. 262102210231), Shenzhen Municipal Science and Technology Innovation Commission Stabilization Support Program (No. 20231128083944001), and the Endogenous Security Cloud Network Convergence R&D Center (Grant No. 602431011PQ1).

Author contributions

Conceptualization: H.L. and B.L.; Methodology: W.H.; Software: L.J. and S.L.; Validation: B.L., L.J. and C.D.; Investigation: H.L. and B.L.; Resources: H.L. and W.H.; Data curation: C.D. and S.L.; Writing–original draft: B.L. and W.H.; Writing–review & editing: H.L., L.J., C.D. and Y.D.; Visualization: Y.D. and S.L.; Supervision: W.H.; Project administration: L.J. and C.D.; Funding acquisition: H.L. All authors have read and approved the final manuscript.

Competing interests

The authors declare no competing financial interests.

Data availability

Data will be made available on request.

Ethics, Consent to Participate, and Consent to Publish declarations

Ethics, Consent to Participate, and Consent to Publish declarations: not applicable.

Clinical Trial Number

This study is a computational experiment based on public datasets, which does not involve human clinical trials. Therefore, clinical trial number is not applicable.