2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Integration Detection Model for Deep Neural Network Backdoor Attacks

Chunlu Wu[1][0009-0000-1121-0861], Junjiang He[1, *][0000-0001-7040-2425], Wengang Ma[1], Ping He[2], Xiaolong Lan[1], Shixuan Ren[1], and Tao Li[1]

[1] School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China
[2] Information Technology Department at Sichuan Chuangang Gas Co., Ltd.
hejunjiang@scu.edu.cn

Abstract. Deep Neural Network (DNN) has demonstrated exceptional performance across various domains. However, with the continuous development of adversarial attack techniques, DNN faces increasingly serious security threats. Existing backdoor attack detection methods are primarily designed for specific attack scenarios and often exhibit insufficient effectiveness when confronting complex attack forms such as dynamic sensitivity optimization and randomized obfuscation. This study proposes an Integration Detection Model for Deep Neural Network Backdoor Attacks (ID-Model), aiming to build an integrated detection framework capable of addressing various backdoor attacks. The ID-Model consists of three core components: the feature extraction and analysis module, the integrated detector module, and the data processing and alert module. Experimental results demonstrate that compared to STRIP and NNCDA methods, the ID-Model integrated detection model achieves a 19% improvement in detection accuracy under Original-Net and R-Net attacks. This research provides an important theoretical foundation for DNN security defense.

Keywords: Deep Neural Network, Backdoor Attack Detection, Integrated Detection, Security Enhancement.

1 Introduction

Deep Neural Networks (DNN) have revolutionized multiple scientific and technological domains [1, 2]. From GPT-3's text generation capabilities and Switch Transformer's scalability breakthroughs to computer vision advances in autonomous driving, DNN have demonstrated exceptional versatility. Their impact extends to game theory with AlphaGo, visual generation with CogView 2.0, and biological sciences where AlphaFold2 and RoseTTAFold have solved protein structure prediction challenges with unprecedented accuracy. Despite these advancements, the widespread deployment of DNN has introduced critical security vulnerabilities, particularly through increasingly sophisticated backdoor attacks that threaten critical infrastructure and public trust in AI technologies.

Fortunately, researchers have developed various methods to detect backdoor attacks in neural networks. Gao et al. [3] introduced the STRIP algorithm, which identifies

* Corresponding authors: Junjiang He.

potential backdoor triggers through input perturbation techniques, though its effectiveness remains limited against complex trigger patterns. Building on this foundation, Doan et al. [4] developed the Februus method, which employs a two-stage preprocessing approach to neutralize backdoor triggers before they can activate. Zeng et al. [5] further enhanced detection capabilities by incorporating additional image transformation techniques into the defensive framework. Taking a different approach, Li et al. [6] proposed the Non-Transferability Detection method, which leverages the limited transferability properties of backdoor attacks. For text-based systems, Fan et al. [7] designed the InterRNN framework specifically to detect backdoor attacks in recurrent neural network text classification systems. These complementary research efforts have significantly advanced backdoor defense techniques and established crucial theoretical foundations that continue to inform current security strategies.

However, existing backdoor attack detection methods generally lack effectiveness, and their performance significantly deteriorates when facing novel attack techniques such as dynamic trigger generation and randomized obfuscation. This lack of effectiveness primarily arises from their design philosophy, which overly relies on single defense mechanisms based on specific attack features. When attackers optimize the trigger candidate set using sensitivity analysis and combine clustering with a comprehensive penalty strategy involving neuron randomization obfuscation, traditional detection methods, such as STRIP and clustering-based detection, struggle to effectively identify malicious inputs. Therefore, developing a more adaptive and flexible defense framework that can effectively respond to dynamic attack characteristics has become an urgent challenge requiring immediate attention.

To address these challenges, the study proposes an integration detection model for deep neural network backdoor attacks for DNN backdoor attacks (ID-Model). The model innovatively integrates multiple detection mechanisms and constructs a multi-layer defense system through an ensemble learning strategy, enabling effective identification of advanced backdoor attacks, including dynamic trigger generation and randomized obfuscation.

The main contributions of the study are as follows:

- The study introduces an innovative ensemble detection framework that significantly strengthens defense against advanced backdoor attacks—such as dynamically generated triggers and randomized obfuscation—through multi-dimensional feature analysis and holistic decision-making.
- The study proposes three improved detection methods for backdoor attacks: EKLFC-CD enhances the recognition of complex backdoor patterns by analyzing features from hidden network layers; SDBD detects backdoor triggers using salt-and-pepper noise interference, reducing the computational cost of real-time monitoring; WSRBD adopts dynamic weight adjustment and multi-dimensional feature analysis to counter diverse attack strategies, improving effectiveness and reliability of security defenses.
- To validate the effectiveness of the model, experiments were conducted using two benchmark attack schemes: the Original Trigger Generator Network (Original-Net), based on a static candidate set, and the improved network (R-Net), which incorporates sensitivity analysis and a comprehensive penalty mechanism. The experimental

results demonstrate that, when facing R-Net backdoor attacks, the ID-Model achieved a 19% improvement in detection accuracy compared to well-established defense methods such as STRIP and NNCAD.

2 Related Work

In recent years, researchers have proposed various innovative methods in the field of data-based backdoor attack detection. Gao et al. [3] introduced STRIP, which distinguishes between benign data and malicious data with triggers by analyzing perturbation differences through constructed input disturbances. However, research by [8] demonstrates that STRIP has limitations when facing certain types of perturbation-resistant attacks. To enhance detection effectiveness, researchers have begun exploring preprocessing-based defense strategies. Liu et al. [9] utilized pre-trained autoencoders to suppress backdoor attacks by altering trigger patterns. Inspired by this, Doan et al. [4] proposed the Februus defense method, which employs the Grad-CAM visualization tool [10] to identify and remove potential trigger regions, while maintaining model performance through GAN-based image restoration [11].

In terms of feature analysis, Chen et al. [12] proposed Feature Consistency-based Sensitivity Metric to distinguish between poisoned and clean samples. Xue et al. [13] introduced a backdoor detection method based on intentional adversarial perturbations. Peri et al. [14] designed a deep k-NN method to detect poisoned samples in feature collision and convex polytope clean-label attacks. Wei et al. [15] proposed the CCA-UD defense mechanism, which detects the presence of backdoor attacks through density-based clustering analysis of training data. Additionally, Chen et al. [16] proposed the Anti-Backdoor Model (ABM) defense algorithm, which first embeds controlled backdoors to detect poisoned data and then trains an external model via knowledge distillation to achieve defense.

Although these methods perform well in specific scenarios, their adaptability remains insufficient when confronted with backdoor attacks involving dynamic trigger generation and neuron randomization obfuscation. Particularly in scenarios where backdoor attack triggers exhibit sensitivity optimization and comprehensive penalty characteristics, the detection accuracy of existing single defense mechanisms significantly declines, failing to provide reliable security guarantees.

3 Proposed Method

As backdoor attack techniques become increasingly diverse, traditional single-defense methods lack the effectiveness required for practical applications. To address this challenge, this chapter proposes an integrated detection model for DNN backdoor attacks (ID-Model). As shown in Fig. 1, the ID-Model architecture comprises three key components: a feature extraction and analysis module, an integrated detector module, and a data processing and alerting module.

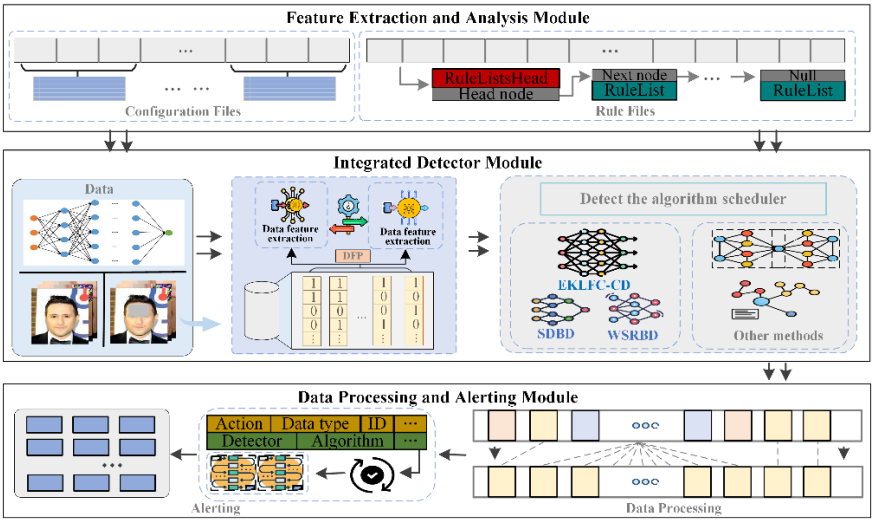

Fig. 1. Architecture Diagram of ID-Model

3.1 Feature Extraction and Analysis Module

The feature extraction and analysis module performs three core functions: configuration file parsing, data preprocessing, and rule file parsing, forming the foundation architecture of the defense model, as shown in Fig. 2. This module employs multi-level filtering algorithms and regular expressions for data validation, supporting format verification of parameters such as IPv4/IPv6 addresses and protocol types, while implementing incremental validation to improve processing efficiency. For configuration parsing, the module supports automatic conversion of multiple formats including JSON, YAML, and XML, and provides configuration hot-update capability, ensuring that new settings can be applied without system restart.

Table 1. Partial parsing strategies improve rule file processing efficiency.

Configuration Parameter	Parameter Assignment
InputDetetor	up
InputDetectorDA	k-mean
InputDetectorDataType	image
Rules	alert,2,set-timestamp,max-num:20
Rules-Info	Found TrojanNN Attack!
OutputDetector	down
rule[id 2]	InputDetector:static, triggerfile:"../trigger/squ.pgm"

Rule files are parsed into a policy tree structure based on directed acyclic graphs, where RTN nodes store basic information such as rule ID, priority, and conditional expressions, as shown in Table 1, while RON nodes contain detailed configurations including detection algorithm parameters, threshold settings, and feature vector

dimensions. The module implements a plugin-based detector architecture, supporting dynamic loading and resource-optimized scheduling, while providing an incremental rule update mechanism that supports the addition, deletion, and modification of individual rules. During the rule compilation process, conflict detection and optimization are performed to reduce runtime overhead, while supporting boolean logic combinations of conditional expressions and compound condition evaluation, providing the defense model with a highly configurable security policy framework and scalability.

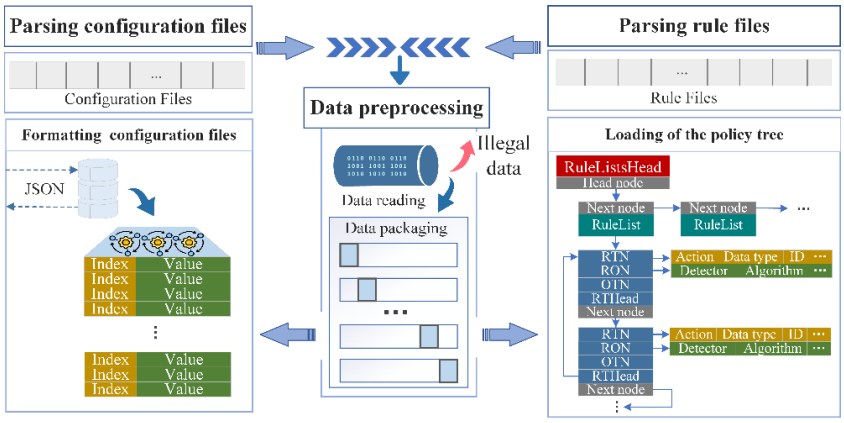

Fig. 2. Feature Extraction and Analysis Module

3.2 Integrated Detector Module

The integrated detector module structure consists of two key components: the Data Feature Platform (DFP) module and the Detection Scheduler module. The Data Feature Platform, functioning as an independent module, is responsible for data type identification and evaluation. The Detection Scheduler module includes a scheduler for feature extraction and backdoor attack detection algorithms.

Data Feature Platform. The Data Feature Platform (DFP) receives pre-processed data streams from the Feature Extraction and Analysis Module (Section 3.1) through a specialized buffer interface that maintains data integrity while transferring parsed configuration parameters and rule-based validation results alongside the original data payload.

The DFP establishes bidirectional communication channels between extraction and detection modules utilizing an asynchronous handshake protocol over TCP/IP connections. Feature vectors are encapsulated within standardized JSON objects containing metadata such as timestamp and confidence metrics. An event-driven feedback mechanism enables detection results to influence extraction parameters through performance metrics propagated via acknowledgment packets. To maintain consistency, a vector clock synchronization protocol ensures proper event ordering across processing units.

The DFP effectively addresses the compatibility issues that deep neural networks face when dealing with diverse input data. The platform achieves precise data type

identification by analyzing header byte sequences of binary file streams. DFP employs an extensible plug-in architecture, enabling defense model developers to flexibly integrate custom feature processing logic into existing functional modules by inheriting core data processing interfaces.

Features undergo dimensionality reduction through Principal Component Analysis followed by z-score normalization before vectorization. The feature extraction submodule employs spectral clustering techniques to analyze neural network hidden layer features, while a bidirectional LSTM with attention mechanisms processes temporal feature evolution. The resulting feature embeddings traverse to the detection module via secure RPC channels implementing TLS 1.3 encryption, establishing an end-to-end differentiable pipeline that optimizes detection performance across diverse backdoor attack vectors.

Detection Scheduler. This optional sub-module, controlled via configuration, forwards extracted features to the detection algorithm scheduler. The scheduler dynamically assigns detection tasks based on DFP classifications, currently supporting three detection methods (Section 4) with extensible interfaces for future scalability. Upon completing detection operations, the scheduler formats results with standardized metadata tags and timestamps before passing them to the Data Processing and Alerting Module (Section 3.3), where they enter the weight-indexed FIFO queue for further processing and potential alert generation.

3.3 Data Processing and Alerting Module

The Data Processing and Alerting Module dynamically manages backdoor attack detection results through a queue mechanism in Fig. 3 (a). It employs a weight-indexed First-In-First-Out (FIFO) list structure with a default memory limit of 65536KB. The queue performs three operations on input data: Normal data is forwarded to downstream systems. Erroneous data gets intercepted to block potential risks. Anomalous data is flagged and pushed to a Buffer Queue to trigger alert workflows. When the queue reaches full capacity, the system forces discarding new entries until memory space is released. This ensures strict control over resource boundaries. Alert content is generated by the policy tree parser module according to predefined rules in configuration files. Customizable Alert Info and Plug Info fields are integrated with the base data. A Unix timestamp records the machine's system time when alerts trigger. Final outputs are structured alert messages containing detector identifiers, algorithm types, status flags, and multi-layer metadata.

The alerting system supports dual output modes. The Information Mode generates real-time alerts in standard log formats. The File Mode persists alert data as structured files. An output selector in configuration files dynamically switches between these modes. Alert messages integrate external correlation data through a cloud query interface. Custom analysis logic extends functionality via a plugin architecture. Operators conduct alert retrospectives using the log auditing mechanism. The log format (Fig. 3 (b)) enforces hierarchical nesting containing four layers: basic alerts, custom content,

plugin extensions, and timestamp fields. A hybrid enrichment strategy combines local policies with cloud-synced rules. This architecture ensures alert traceability and maintains defense system sustainability through coordinated metadata management.

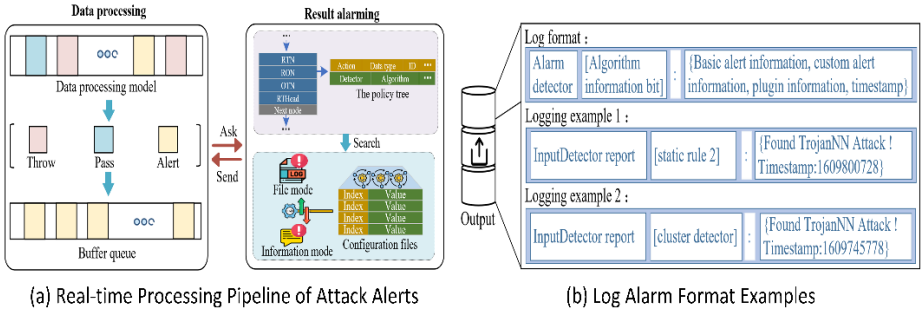

Fig. 3. Feature Extraction and Analysis Module

4 Design and Integration of EKLFC-CD, SDBD, and WSRBD in ID-Model

This section presents a detailed introduction to three innovative backdoor detection methods integrated into the detector sub-module of the ID defense model: Enhanced K-means and Latent Feature Capture Clustering Detection (EKLFC-CD), Salt-based Differential Backdoor Detection (SDBD), and Window-based Segmentation and Reconstruction Backdoor Detection (WSRBD). EKLFC-CD enhances the recognition of complex backdoor patterns by analyzing features from hidden network layers; SDBD detects backdoor triggers using salt-and-pepper noise interference, reducing the computational cost of real-time monitoring; WSRBD adopts dynamic weight adjustment and multi-dimensional feature analysis to counter diverse attack strategies, improving effectiveness and reliability of security defenses.

4.1 Enhanced K-means and Latent Feature Capture Clustering Detection

The study [17] introduces two clustering-based detection algorithms that rely on the activation values of the last hidden layer neurons. However, if an attacker pre-activates these neurons, it becomes possible to exploit the resulting anomalous activations, undermining the reliability of the algorithms. To address this limitation, the study focuses on neurons across the last three layers. When the target neurons are spaced farther apart, achieving optimal classification becomes more difficult, thus reducing the success of the attack. By monitoring the changes in these accompanying neurons, the defensive capabilities of the system can be enhanced.

The set of neurons with the highest activation values is selected from the candidate network layers including by default the last layer l_1, second-to-last layer l_2, and third-to-last layer l_3, forming the collection $N = \{N_1, N_2, N_3\}$ which corresponds to the three

default layers. Simultaneously, the set of top_m lower-layer neurons with the highest weights is selected, forming $W = \{W_1, W_2\}$, as shown in the Fig. 4. For each $N_i \in N$, $i > 1$, there exists a corresponding W_{i-1}. For any neuron *n* included in feature extraction, where $n \in N_i$, a set $n_i^{'}$ is identified within W_{i-1}, containing the top_m neurons from the (i-1)-th layer that connect to neuron n with the highest weights, as shown in Eq. 1 and Eq. 2.

$$n_i^{'} \in W_{i-1} \tag{1}$$

$$W_{i-1} = \mathrm{n}_1^{'} \cup \mathrm{n}_2^{'} \cup \cdots \cup \mathrm{n}_{l_n}^{'} \tag{2}$$

The components N_3 and W_2 exhibit structural correlation within the network model. As an example, consider $top_n = 1$ and $top_m = 3$ to calculate the relationship between the third-to-last layer N_3, the second-to-last layer N_2, and W_2. Since $top_n = 1$, the neuron with the highest activation value from the third-to-last layer is selected.

Algorithm 1 Enhanced K-Layer Feature Clustering-based Clustering Detection

Input: Training data set D_p, the highest activation value in each layer top_n, the highest weights connecting to selected neurons top_m;

Input: Candidate network layers last layer l_1, second-to-last layer l_2, third-to-last layer l_3;

Output: The result of clustering discrimination *y*;

Train DNN model K on D_p;
Initialize $N = \{\}$, $W = \{\}$; (*N* for highest activation neurons, *W* for highest weight connections);
for $s_i \in D_p$ do
    Initialize $N_1, N_2, N_3, W_1, W_2 = \{\}, \{\}, \{\}, \{\}, \{\}$;
    $N_1 \Leftarrow$ Select top_n neurons with highest activation values from last layer l_1 for input s_i;
    $N_2 \Leftarrow$ Select top_n neurons with highest activation values from second-to-last layer l_2 for input s_i;
    $N_3 \Leftarrow$ Select top_n neurons with highest activation values from third-to-last layer l_3 for input s_i;
    $W_1 \Leftarrow$ Select top_m neurons from l_2 with highest weights connecting to each neuron in N_1;
    $W_2 \Leftarrow$ Select top_m neurons from l_3 with highest weights connecting to each neuron in N_2;
    Remove neurons from N_2 that are already in W_1; (Ensure no duplicates);
    Remove neurons from N_3 that are already in W_2;
    $N \Leftarrow \{N_1, N_2, N_3\}$;
    $W \Leftarrow \{W_1, W_2\}$;
    $F \Leftarrow$ Concatenate *N* and *W* to form the feature vector for s_i;
    Append *F* to feature matrix *A*;
end for
$clusters \Leftarrow$ Apply clustering method on feature matrix *A*;
$y \Leftarrow$ Analyze clusters to detect poisoned inputs;
return *y*;

As shown in the Fig. 4, neuron number 7 in l_3 is added to N_3. In the second-to-last layer, the top $top_m = 3$ neurons with the highest weights connected to neuron number 7 are selected and adding to W_2. Additionally, in the second-to-last layer, apart from the three neurons in W_2 that connect to the third-to-last layer, the neuron with the highest activation value is selected (with $top_n = 1$). If the selected neuron is already included in W_2, then this neuron is not added to N_2.

Therefore, three different neurons will be selected in the second-to-last layer, four different neurons in the third-to-last layer, and so on, with up to thirteen different neurons potentially selected in the second-to-last layer. Thus, when $top_n = 1$ and $top_m = 3$, a maximum of 20 distinct neurons from the internal hidden layers can be extracted as potential features of the network's internal input data. In this study, the concatenated set F of N and W is considered as the region where potential activated neurons and their accompanying activated neurons that need to be examined for defense may exist. This represents the one-dimensional feature vector selected from all the features. The full algorithm for the EKLFC-CD method is presented in Algorithm 1.

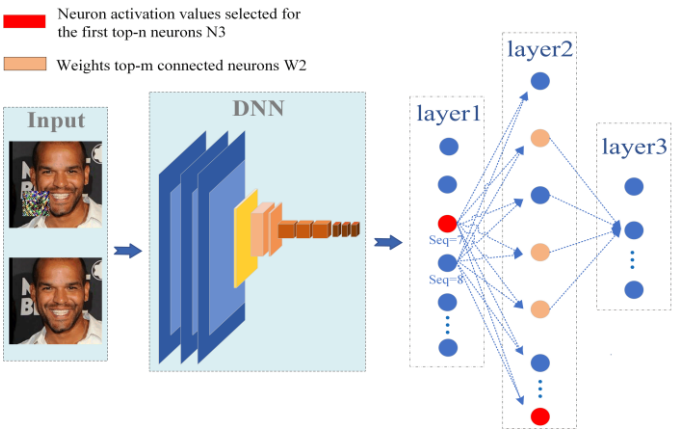

Fig. 4. Critical Neuron Identification by EKLFC-CD

4.2 Salt-based Differential Backdoor Detection

The SDBD leverages the sensitivity of backdoor triggers to common image transformations, preventing potential backdoor activation. To neutralize latent backdoor triggers, this study introduces a backdoor detection mechanism through the incorporation of perturbation noise. This method disrupts potential backdoor triggers by perturbing potential trigger points and analyzing the disparities before and after the introduction of perturbation noise within the test datasets, as shown in Fig. 5.

Assuming that the network model is expressed as a function $F(x)$, any input data x_i belonging to the input image dataset is I, and its corresponding categorization representation results in R_i, i.e., any $x_i \in I$, are available as in Eq. 3.

$$F(x_i) = R_i \tag{3}$$

The trigger data designed by the adversary is t, and the set purpose classification is R_t Then the malicious data I_t with trigger obtained by superimposing the arbitrary input data x_i over the trigger is Eq. 4.

$$I_t = (1 - \theta) \cdot I + \theta \cdot t \tag{4}$$

The malicious data I_t generates a classification result in the network model, as shown in Eq. 5.

$$F(I_t) = R_t \tag{5}$$

Pepper noise is chosen to salt the input image data thereby changing the classification performance of the malicious data. The noise density set here for adding input data to the pretzel noise is α and the final pretzel noise data obtained is Eq. 6.

$$S = GenSalt(\alpha) \tag{6}$$

The test data $I_t + S$ obtained by adding the generated pretzel noise S to the malicious data I_t, the classification result of the test data obtained in the network model is Eq. 7.

$$F(I_t + S) = R^{'} \tag{7}$$

If the difference between $R^{'}$ and R_t results in $Distance(R_t, R^{'})$, according to the result of discriminating the difference Distance is the final result of the backdoor detection of the difference in salt addition.

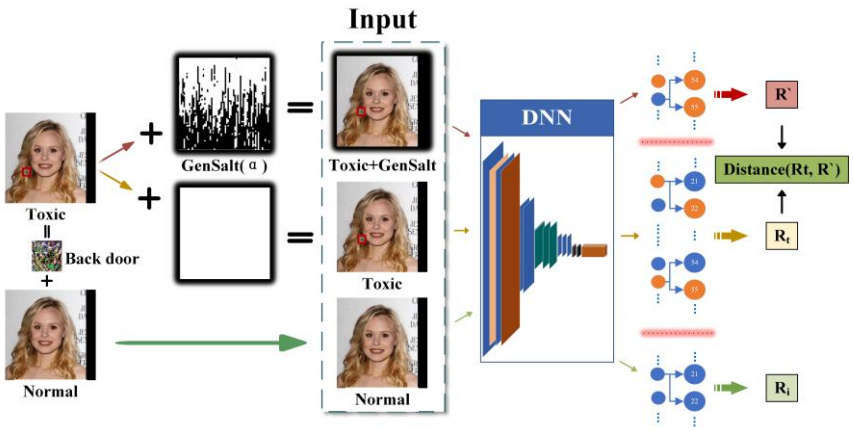

Fig. 5. SDBD Dynamic Boundary Evolution

4.3 Window-based Segmentation and Reconstruction Backdoor Detection

Liu et al. [18] introduced a DNN backdoor attack with a trigger, typically covering less than 11% of the total input image pixels. This trigger, when superimposed on input data, consistently activates adversary-designed neurons, a characteristic of DNN

backdoor attacks. Capitalizing on the smaller trigger backdoor and the stable classification expectation produced by the superposition of any data, the researchers introduced the window cut reorganization method for backdoor detection.

The DNN backdoor trigger overlays input image data, resulting in the same classification outcome. This reduces the problem of detecting malicious triggers to identifying regions within the input image that, when combined with test data, produce consistent results. For any input data x, if there exists a region m such that overlaying any test data I_{test} with $m + I_{test}$ consistently yields the same output $F(m + I_{test})$, it suggests that region m may contain a backdoor trigger t. To search for region m within x, it is noted that malicious triggers usually occupy less than 11% of the input image pixels. Therefore, a sliding window approach is employed, using a rectangular window of size k to define the search area for m. The details of the detection algorithm are provided in Algorithm 2.

Algorithm 2 Window-based Segmentation and Reconstruction Backdoor Detection

Input: An untrusted training dataset D_p, Sliding window size *k*, Step size for window sliding *s*;

Output: Malicious detection result *y*;

Initialize window parameters *k* and *s*;
for each input data $x \in D_p$ do
for each window position m do
    for each test set data $I_{test} \in T$ do
        Overlay test set data I_{test} onto the region *m* of input data *x* to obtain overlaid data $m + I_{test}$;
    end for
    if the overlaid data $m + I_{test}$ generates $== F(m + I_{test})$ then
        Mark current region *m* as a potential backdoor trigger *t*;
    end if
end for
end for
Put all marked regions *m* into a boundary detector;
return *y*;

With each sliding step of the search window set to *s*, the window advances by *s* units along the boundary. At each position, the region within the window is extracted and overlaid onto the I_{test} data from test set *T* to obtain classification results. These results are then subjected to the boundary discriminator to identify malicious input based on the superimposed discriminative outcomes.

An in-depth analysis of the three detection methods reveals their distinct advantages in different backdoor attack scenarios. The EKLFC-CD method, which extracts features from network hidden layers, provides robust security by identifying both common and anomalous backdoor patterns, making it effective during model training. The SDBD method leverages salt-and-pepper noise interference to detect backdoor triggers, excelling in real-time monitoring with low computational cost. The WSRBD method uses dynamic weight adjustment and multi-dimensional feature analysis to counter various attack strategies, offering adaptable and reliable security for network models.

5 Experimental Evaluation

The study designed two progressive experimental phases: first, it tested Original-Net and R-Net attacks across 7 network model-dataset combinations to evaluate the baseline performance of five detection methods: EKLFC-CD, SDBD, WSRBD, STRIP, and NNCAD; second, it constructed a cross-domain composite dataset containing three categories of data features and used two attack methods to generate poisoned samples as input, with a focus on verifying the effectiveness of the ID-Model integrated defense model when confronting complex attack scenarios. The experimental design comprehensively covers systematic validation ranging from basic defensive performance to cross-domain composite scenarios.

5.1 Evaluation Criteria

Drawing from the evaluation of detection results in anomaly detection, relevant evaluation metrics for DNN backdoor attack detection can be introduced. The performance evaluation of DNN backdoor detection employs three primary metrics: Accuracy (ACC), False Acceptance Rate (FAR), and False Rejection Rate (FRR). These metrics are formulated as: $ACC = \frac{TP+TN}{T+N} \cdot 100\%$, where ACC represents the ratio of correctly classified instances. $FAR = \frac{FP}{N} \cdot 100\%$, where FAR quantifies the proportion of erroneously accepted backdoored samples. $FRR = \frac{FN}{P} \cdot 100\%$, where FRR measures the proportion of incorrectly rejected benign samples. Achieving optimal detection performance involves maximizing accuracy while simultaneously minimizing false acceptance and false rejection rates.

5.2 Experimental Data and Experimental Models

Network Models. The network models employed are popular open-source models in computer vision: VGGNet16 [19], ResNet [20], and AlexNet [21].

Datasets. The datasets include the VGG-Face dataset [22], CIFAR-10 [23], and MNIST [24]. 20% of each dataset is allocated as the test set.

Main Parameters. Prior to model training, basic image preprocessing steps, including denoising and normalization, are performed. For CIFAR-10 and MNIST, the experiments are conducted for 10 iterations, with 100 data points per iteration, and a learning rate of 0.01. The VGG-Face dataset involves 20 iterations, with 250 data points per iteration. The learning rate starts at 0.01 and is reduced to 0.005 after 10 iterations. Training is terminated when a significant drop in model performance is observed. If the validation loss increases by more than α=0.05 for three consecutive iterations, training stops, as shown in Eq. 8.

$$L_{val}(n) > L_{min} \cdot (1 + \alpha) \tag{8}$$

Backdoor Attacks. Backdoor attacks use two different trigger generation networks: the original trigger generation network based on a static candidate set (Original-Net) and the improved network incorporating sensitivity analysis and a comprehensive penalty mechanism (R-Net) [8]. Original-Net uses a fixed, manually selected trigger candidate set and optimizes the trigger by minimizing the mean squared error loss function. In contrast, R-Net introduces a dynamic sensitivity analysis method to select the candidate set and employs a comprehensive penalty mechanism that combines clustering obfuscation with randomized neuron activation. These two methods represent the traditional and improved backdoor attack paradigms, providing comprehensive attack benchmarks for this study.

5.3 Experimental Analysis

The experiment analyzes the performance of five detection methods across various network model-dataset combinations and verifies the effectiveness of the proposed ID-Model through composite datasets, demonstrating its effectiveness when confronting complex backdoor attack scenarios.

Performance Comparison Experiment. As shown in Table 2, when facing traditional Original-Net attacks, all five detection methods perform well, with accuracy generally exceeding 90%. Specifically, EKLFC-CD achieves extremely high accuracy reaching up to 99.2% across most network model-dataset combinations while maintaining low false acceptance rates with an average of 0.84% and false rejection rates with an average of 1.19%. In contrast, conventional STRIP and NNCAD methods show relatively weaker performance in certain combinations, particularly in the VGGNet16-MNIST combination where NNCAD achieves only 93.2% accuracy.

When confronting advanced R-Net attacks featuring dynamic trigger generation and randomized obfuscation characteristics, traditional detection methods exhibit significant performance degradation. Data indicates that the accuracy of the STRIP method drops dramatically from an average of 97.19% to 78.50%, showing an average decrease of 18.69 percentage points, while the NNCAD method declines even more precipitously from an average of 96.37% to 73.11%, representing an average decrease of 23.26 percentage points. Particularly noteworthy is that when facing R-Net attacks, NNCAD's FAR value surges to an average of 26.49%, meaning that over a quarter of malicious samples are misclassified as benign.

The data analysis clearly demonstrates that the three methods proposed in this study exhibit significant performance advantages when addressing advanced backdoor attacks with dynamic trigger generation and randomized obfuscation characteristics. Compared to conventional methods, these approaches successfully overcome effectiveness deficiencies, providing more reliable solutions for deep neural network security protection. The EKLFC-CD method is particularly noteworthy, maintaining

performance close to its original capabilities even in the most challenging attack scenarios, demonstrating its tremendous potential in practical applications.

Table 2. Performance comparison of different defense methods across network model-dataset combinations. (M-D denotes network model type M and dataset type D, specifically: VGG-Net16-CIFAR-10 (V-C), VGGNet16-VGGFace-10 (V-V), VGGNet16-MNIST (V-M), ResNet-CIFAR-10 (R-C), ResNet-MNIST (R-M), AlexNet-CIFAR-10 (A-C), AlexNet-MNIST (A-M). Original-Net is abbreviated as O-Net.)

Attacks	M-D	STRIP			NNCAD			EKLFC-CD		
		ACC	FAR	FRR	ACC	FAR	FRR	ACC	FAR	FRR
O-Net	V-C	97.6%	2.1%	6.3%	96.2%	1.2%	3.2%	**98.2%**	**0.7%**	**1.2%**
	V-V	96.3%	**0.2%**	7.2%	94.3%	1.7%	1.2%	96.3	0.7%	2.6%
	V-M	**99.2%**	**0.2%**	2.6%	93.2%	**0.4%**	3.6%	91.2%	1.0%	1.2%
	R-C	98.0%	**0.7%**	1.0%	98.0%	1.8%	0.8%	**99.0%**	1.2%	**0.7%**
	R-M	97.7%	0.5%	5.6%	96.7%	0.5%	2.7%	**98.7%**	**0.3%**	1.5%
	A-C	**99.3%**	**0.2%**	2.7%	98.0%	1.2%	1.5%	97.1%	0.9%	**0.7%**
	A-M	96.2%	0.7%	7.5%	96.2%	1.5%	7.5%	**99.2%**	1.1%	**0.4%**
R-Net	V-C	74.6%	7.5%	13.3%	71.5%	28.3%	4.3%	97.2%	**2.0%**	1.2%
	V-V	82.5%	6.5%	17.3%	72.7%	27.3%	5.6%	97.2%	1.5%	2.0%
	V-M	77.6%	9.0%	13.2%	72.3%	35.2%	4.5%	**99.2%**	**0.7%**	2.0%
	R-C	72.3%	10.5%	14.7%	71.3%	27.7%	2.2%	**96.7%**	1.3%	2.8%
	R-M	87.6%	5.2%	11.8%	69.6%	35.8%	7.8%	**99.2%**	**0.4%**	**0.4%**
	A-C	82.6%	4.6%	16.3%	82.1%	16.6%	2.3%	**98.9%**	0.9%	1.6%
	A-M	72.3%	8.6%	14.5%	72.3%	14.5%	3.7%	**98.2%**	1.2%	**1.6%**

Attacks	M-D	SDBD			WSRBD		
		ACC	FAR	FRR	ACC	FAR	FRR
O-Net	V-C	91.7%	3.2%	2.0%	96.3%	0.8%	1.3%
	V-V	89.2%	3.8%	4.8%	**98.7%**	0.4%	**0.7%**
	V-M	93.5%	5.2%	3.7%	97.2%	0.9%	2.2%
	R-C	91.7%	1.2%	1.3%	94.9%	1.0%	1.4%
	R-M	95.5%	1.9%	1.9%	95.7%	0.4%	**0.7%**
	A-C	98.1%	0.9%	0.8%	97.5%	0.4%	**0.7%**
	A-M	96.9%	2.2%	3.4%	95.4%	**0.5%**	2.1%
R-Net	V-C	**97.6%**	2.6%	**0.4%**	94.6%	1.2%	2.3%
	V-V	91.5%	2.2%	2.2%	**97.6%**	**0.6%**	**0.5%**
	V-M	93.4%	1.8%	3.1%	96.5%	0.6%	**1.2%**
	R-C	93.6%	1.9%	**0.9%**	95.3%	**0.5%**	3.1%
	R-M	95.6%	2.8%	2.1%	97.4%	0.7%	1.8%
	A-C	94.6%	1.1%	2.2%	98.1%	**0.2%**	**0.9%**
	A-M	94.2%	2.7%	3.5%	95.8%	**1.0%**	3.5%

ID-Model Effectiveness Verification. The experiment compared the defensive performance of ID-Model, STRIP, and NNCDA against dual-trigger backdoor attacks with each maintaining a 10% poisoning rate across three network architectures including

Vggnet16, Resnet, and Alexnet, displaying metrics for accuracy, false acceptance rate, and false rejection rate.

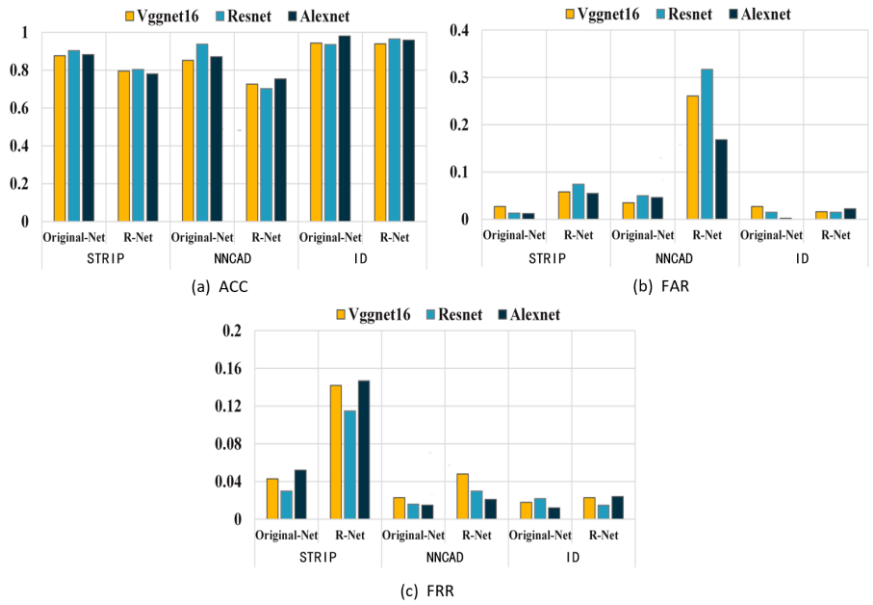

Fig. 6. Performance Comparison of ID-Model with Traditional Detection Methods Across Different Network Architectures.

The accuracy metric in Fig. 6 (a) demonstrates the exceptional performance stability of the ID model. Although existing methods can maintain high accuracy, above 80%, when defending against the Original-Net attack, their performance drops sharply when facing the R-Net attack with dynamic trigger generation and random obfuscation. STRIP's accuracy is 19%, and NNCAD experiences an even greater decline, with an accuracy of 23%. In contrast, regardless of the attack type, the ID model consistently maintains an accuracy of over 94% across all network architectures, showcasing its outstanding effectiveness against advanced attack strategies.

In Fig. 6 (b) and Fig. 6 (c), regarding the R-Net attack, NNCAD exhibits an alarming false acceptance rate exceeding 25%, indicating that more than a quarter of malicious samples can evade detection. Meanwhile, STRIP's FAR slightly increases, but its rejection rate rises sharply to over 14%, meaning many legitimate samples are incorrectly labeled as malicious. The ID model consistently maintains a low false acceptance rate and rejection rate across all test configurations, demonstrating balanced and reliable performance. This balanced performance is crucial for real-world deployment scenarios, as both security with low false acceptance rate and usability with low false rejection rate are key considerations. The experimental results convincingly demonstrate that the proposed integrated approach, by leveraging complementary techniques, builds a more comprehensive and resilient defense system, providing an effective solution to counter complex backdoor attacks.

6 Experimental Evaluation

This study proposes an Integrated Detection Model for Deep Neural Network Backdoor Attacks (ID-Model), which integrates multiple detection methods to construct a multi-level defense system that demonstrates excellent effectiveness when facing complex attack forms such as dynamic sensitivity optimization and randomized obfuscation. The research's main contributions are manifested in three aspects: first, it innovatively designs and implements three complementary detection methods (EKLFC-CD, SDBD, WSRBD); second, it constructs an extensible integrated detection framework that employs a hierarchical decision mechanism to achieve end-to-end protection from feature extraction to threat warning; finally, experimental results show that ID-Model achieves detection accuracies of 94.3% for Original-Net attacks and over 90% for R-Net attacks, representing an average improvement of 19% compared to existing optimal methods. Future work will improve efficiency, boost detection sensitivity, and enhance defense against emerging backdoor attacks, especially in black-box settings, while broadening evaluation against new SOTA defenses and adaptive attacks.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (No.U24A20239, No.62032002, and No. 62402300); in part by the Sichuan Provincial Science and Technology Department regional innovation cooperation key project (Grant No.2025YFHZ0265); in part by the Youth Science Foundation of Sichuan,(No.2025ZNSFSC1474); in part by the China Postdoctoral Science Foundation (No.2024M752211); in part by Sichuan Province Science and Technology Innovation Seedling Project (MZGC20240056); in part by the key laboratory of data protection and intelligent management ministry of education (SCUSACXYD202301).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Gustafsson, F.K., Danelljan, M., Schön, T.B.: Evaluating scalable bayesian deep learning methods for robust computer vision. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), pp. 318–319 (2020)
2. Wei, J., Fan, M., Jiao, W., Jin, W., Liu, T.: BDMMT: Backdoor sample detection for language models through model mutation testing. IEEE Trans. Inf. Forensic Secur. (2024)
3. Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: STRIP: A defence against trojan attacks on deep neural networks. In: Proceedings of the 35th Annual Computer Security Applications Conference, pp. 113-125 (2019)
4. Doan, B.G., Abbasnejad, E., Ranasinghe, D.C.: Februus: Input purification defense against trojan attacks on deep neural network systems. In: Proceedings of the 36th Annual Computer Security Applications Conference, pp. 897-912 (2020)
5. Qiu, H., Zeng, Y., Guo, S., Zhang, T., Qiu, M., Thuraisingham, B.: DeepSweep: An evaluation framework for mitigating DNN backdoor attacks using data augmentation. In: ACM Asia Conf. Comput. Commun. Secur. (ASIACCS), pp. 363-377 (2021)

6. Li, Y., Ma, H., Zhang, Z., Gao, Y., Abuadbba, A., Xue, M., Fu, A., Zheng, Y., Al-Sarawi, S.F., Abbott, D.: NTD: Non-transferability enabled deep learning backdoor detection. IEEE Trans. Inf. Forensic Secur., vol. 19, pp. 104-119 (2023)
7. Fan, M., Si, Z., Xie, X., Liu, Y., Liu, T.: Text backdoor detection using an interpretable RNN abstract model. IEEE Trans. Inf. Forensic Secur., vol. 16, pp. 4117-4132 (2021)
8. Ren, S., Wang, M., Zhao, H.: An improved backdoor attack method for deep neural networks. Inf. Net. Sec., vol. 21, no. 05, pp. 82-89 (2021)
9. Liu, Y., Xie, Y., Srivastava, A.: Neural trojans. In: IEEE Int. Conf. Comput. Des. (ICCD), pp. 45-48 (2017)
10. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vis., vol. 128, pp. 336-359 (2020)
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM, vol. 63, no. 11, pp. 139-144 (2020)
12. Chen, W., Wu, B., Wang, H.: Effective backdoor defense by exploiting sensitivity of poisoned samples. In: Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 35, pp. 9727-9737 (2022)
13. Xue, M., Wu, Y., Wu, Z., Zhang, Y., Wang, J., Liu, W.: Detecting backdoor in deep neural networks via intentional adversarial perturbations. Inf. Sci., vol. 634, pp. 564-577 (2023)
14. Peri, N., Gupta, N., Huang, W.R., Fowl, L., Zhu, C., Feizi, S., Goldstein, T., Dickerson, J.P.: Deep k-NN defense against clean-label data poisoning attacks. In: Comput. Vis. ECCV Workshops, pp. 55-70 (2020)
15. Guo, W., Tondi, B., Barni, M.: Universal detection of backdoor attacks via density-based clustering and centroids analysis. IEEE Trans. Inf. Forensic Secur., vol. 19, pp. 970-984 (2023)
16. Chen, C., Hong, H., Xiang, T., Xie, M.: Anti-backdoor model: A novel algorithm to remove backdoors in a non-invasive way. IEEE Trans. Inf. Forensic Secur. (2024)
17. Tan, T.J.L., Shokri, R.: Bypassing backdoor detection algorithms in deep learning. In: IEEE Eur. Symp. Secur. Privacy (EuroS&P), pp. 175-183 (2020)
18. Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks. In: 25th Annual Network and Distributed System Security Symposium (NDSS 2018), Internet Society (2018)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (ICLR 2015), Computational and Biological Learning Society, pp. 1-14 (2015)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778 (2016)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems (NIPS), vol. 25 (2012)
22. Parkhi, O., Vedaldi, A., Zisserman, A.: Deep face recognition. In: BMVC 2015 - Proceedings of the British Machine Vision Conference 2015, British Machine Vision Association, pp. 1-12 (2015)
23. Krizhevsky, A., Hinton, G.: Learning Multiple Layers of Features from Tiny Images. Technical Report, University of Toronto (2009)
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proc. IEEE 86(11), 2278-2324 (1998)