Academic Journal of Applied Sciences
ISSN: 3105-0549 (Print), ISSN: 3105-0557 (Online) | Vol. 1, No. 2, 2026

Latent Cognitive Reinforcement for Anti-Backdoor Strategy

Yanwen Wang, Yanchang Liu, Xiaorou Zhang

School of Physics and Electronic Engineering, Northeast Petroleum University, Daqing, Heilongjiang 163318, China

Abstract: With the widespread deployment of deep neural networks in sensitive applications, backdoor attacks have become a serious security concern. Such attacks enable adversaries to manipulate the prediction results without affecting the normal classification of clean samples by embedding hidden triggers in the model. Aiming at the limitations of existing defense methods in terms of adaptive attacks and generalization ability, this paper proposes the Latent Cognitive Reinforcement (LCR) backdoor defense strategy. The method utilizes a multi-layer feature representation of the intermediate and output layers to extract the cognitive patterns of the model, thereby identifying and eliminating potential backdoor trigger signals. Compared with tradit ional methods such as anomaly detection, neuron pruning, or adversarial training, LCR is more generalized and robust. Experiments on two datasets, CIFAR-10 and GTSRB, and five mainstream model architectures prove that LCR achieves an average of 91.80% in detection performance (AUROC), which significantly outperforms the existing state-of-the-art defense techniques and demonstrates good cross-model and cross-attack scenario adaptability.

Keywords: Neural networks; Security; Backdoor attacks; Backdoor defense; Latent patterns.

1. Introduction

Deep Neural Networks (DNNs) play a key role in a wide range of domains, greatly advancing computer vision, autonomous systems, natural language processing, and cybersecurity. The ability of DNNs models to automatically learn to extract complex patterns from massive amounts of data has led to a wide range of real-world applications ranging from healthcare diagnostics to financial fraud detection [1, 2]. However, as DNNs are increasingly deployed in real-world security-sensitive environments, model security is a growing concern. Among the many threats, backdoor attacks have become a very dangerous attack due to their stealthy nature and serious harm. Such attacks, by implanting specific triggers in the training data or models, enable the attacker to manipulate the model outputs through inputs with triggers in the inference phase with little or no impact on the performance of the clean inputs.

Most of the existing defense strategies are based on certain idealized assumptions, such as having access to clean training data, known attack patterns, or having access to the internal structure of the model. However, in real-world scenarios, these conditions are often difficult to fulfill. Detection-based approaches [3] usually rely on analyzing output distributions, trigger backward inference, or monitoring activation functions during model inference to identify anomalous model behavior. However, such methods are usually not resistant to adaptive attacks that modify triggers to evade detection. Model pruning methods include fine-grained pruning [4] and adversarial fine-tuning [5], which aim to remove backdoor neurons by retraining the model or adjusting network parameters. While effective in some cases, they typically require access to large amounts of clean data, which is not always available and carries the risk of degrading the overall performance of the model. Data sanitization methods [6] attempt to weed out training samples that have been implanted with triggers (contaminated samples) through statistical anomaly detection, but complex poisoning strategies may subtly mix backdoor triggers with clean data, making separation extremely difficult. More critically, current defense mechanisms generally face the problem of poor generalization ability, and defenses that are effective against one class of backdoor attacks may fail against another class of backdoor attacks.

To address the above problems, this paper proposes a Latent Cognitive Reinforcement (LCR) generic defense strategy that aims to enhance the robustness of the model against backdoor attacks. By utilizing multi-layer cognitive representations to enhance model resilience against backdoor attacks, LCR does not rely on explicit attack pattern recognition or a large amount of training data, but instead mines the multi-layer feature representations of the model in the intermediate layer and the output layer to identify latent cognitive patterns, which in turn detects and eliminates backdoor triggering signals. Different from the traditional static filtering or model post-processing repair, LCR realizes the backdoor attack-awareness capability of the model by means of cognitive-level reinforcement based on features. Experiments demonstrate that this strategy not only significantly improves the accuracy of backdoor detection, but also achieves better generalization performance across attack types and model architectures while maintaining the original performance of the model.

2. Problem Statement

LCR as a backdoor attack detection method is based on the fact that the model's prediction of a backdoor image depends on a small number of pixels. The detection can be performed in training and testing phases: in the training phase, the defender can remove backdoor samples; in the testing phase, the defender can identify potential attacks and their triggers.

2.1. Threat Model

In this paper, we consider a typical data poisoning backdoor attack scenario, where an attacker attempts to implant backdoor logic in a neural network by tampering with the training data, while keeping the model to maintain normal behavior with clean inputs. This attack is characterized by its

stealthy nature and wide range of hazards, and is often difficult to detect in standard performance evaluations.

(1) Attacker Capabilities

An attacker can modify a small portion of the training dataset by injecting backdoor samples, each poisoned sample contains a specific trigger. The trigger is implanted into a target class sample chosen by the attacker. The attacker cannot directly access the training process, i.e., it cannot modify the model structure and parameters, nor can it intervene in the optimization details or hyper-parameter settings of the training process. To ensure stealth, the proportion of poisoned samples needs to be controlled within 2.5%, thus ensuring that the overall categorization of clean data remains high while the backdoor is effectively implanted.

Backdoor triggers come in a variety of ways, either static (e.g., pixel patches) or adaptive triggers that can change with input characteristics. Some attacks are made by interference in the frequency domain making the trigger difficult to detect in the spatial domain. Regardless of the specific trigger design, the backdoor lies dormant during standard model evaluation, but an attacker can activate it during inference, causing the model to misclassify inputs containing embedded triggers.

(2) Defender Capabilities and Constraints

On the defense side, this paper assumes that the defender has absolute control over the entire process of training, but cannot know in advance whether there is a backdoor attack, the backdoor triggering method, or the size of the poisoning rate. Unlike existing security defense methods, this study considers a more challenging application scenario where the defender needs to effectively identify malicious backdoor samples without external confirmation data. The defender aims to ensure that the model detects and eliminates backdoor triggers without altering the predictive performance of clean samples. The technique needs to be resistant to multiple attacks, scale to large datasets, and be generalizable across different architectures.

(3) Attack process

First, the attacker randomly selects a small portion of samples from a set of training samples during the poisoning process, changes their input characteristics by embedding triggers, and sets the corresponding target labels. After that, the poisoned samples are mixed with clean samples to ensure that the attack cannot be detected. Next, in the training phase, the model learns to associate the triggers with the target categories while the association maintains a high accuracy on the clean samples. Finally, the attacker adds the same triggers to the inputs in the inference phase to guide the model to make misclassifications, which leads to the attack.

(4) Defense Objectives

The core task of defense is to accurately separate clean samples from backdoor samples without a priori labeling information. Since the backdoor exploits the feature of non-semantic correlation, it usually leaves subtle but detectable traces in the feature space. A robust defense must efficiently extract and analyze these cognitive patterns while avoiding over-correction of the model, which can degrade its performance. The goals of the LCR approach are to accurately identify latent backdoor triggers, to reduce interference with model parameters and structure, to maintain the model's high predictive performance on clean data, and to be generalizable and applicable across different datasets and models with different attack styles.

2.2. Problem Formulation

Considering a K −class image classification task, we denote the training data as $D = D_c \cup D_b$, the clean subset as $(x_i, y_i)_{i=1}^{N} \in D_c$, and the poisoned subset as $\left(x_i^{bd}, y_i^{bd}\right)_{i=1}^{M} \in D_b$. The attacker injects backdoor triggers using function $\left(x_i^{bd}, y_i^{bd}\right) = \mathrm{A}(x_i, y_i)$, which converts a clean sample into a backdoor sample. The $x \in X \subset R^{\omega \times h \times c}$ are the inputs and $y \in Y = 1, \ldots, K$ are the labels for K classes in total. The poisoning rate is defined as $\frac{|\mathcal{D}_b|}{|\mathcal{D}|} = \frac{M}{M+N}$. The defender's goal is to accurately detect samples $x \in D_b$.

Backdoor sample detection is an unsupervised binary classification task (backdoor category and clean category). We record the predicted back door sample as $\mathcal{D}_b'$ and the predicted clean sample as $\mathcal{D}_c'$. We use the L1 criterion of the learned mask M to detect the backdoor samples. Because it measures the pixel intensity of CP extracted from CD to judge whether the sample x contains a backdoor according to the mask m:

$$g(x) = \begin{cases} 1 & if \parallel m \parallel_1 \leq t, \\ 0 & if \parallel m \parallel_1 > t, \end{cases} \tag{1}$$

Where t is a threshold, $g(\cdot) = 1$ indicates that there is a backdoor sample, and $g(\cdot) = 0$ suggests that there is a clean sample. For test time detection, we assume that the defender can access a small number of confirmed clean samples D_s. Then, the defender can calculate the distribution of $x_i \in D_s$, the mean $\parallel m \parallel_1$, and the standard deviation $\sigma \parallel m \parallel_1$. The threshold can be determined as: $\mathrm{t} = \parallel m \parallel_1 - \gamma - \sigma \parallel m \parallel_1$, where γ is the hyperparameter for controlling sensitivity.

2.3. Overview of the Methodology

To solve the problems of existing defense methods relying on strong assumptions and poor adaptability to attacks, this paper proposes a feature space-based LCR strategy. The core idea is to introduce a multi-layer feature consistency mechanism with cognitive alignment constraints during the training process, so that the model can spontaneously identify the abnormal activation paths of backdoor features. LCR does not rely on clean samples and does not require explicit reconstruction of triggers, and possesses good versatility and practicability. The optimization objectives of LCR include: enhancing the robustness of the model to small perturbations; strengthening the consistency between shallow and deep feature expressions; maintaining the model's accuracy for the original classification task; and regulating the consistency between the cognitive model and the original model output distribution.

Through the above design, LCR not only significantly improves the backdoor detection accuracy, but also maintains a low computational cost, which is suitable for practical deployment scenarios.

3. Latent Cognitive Reinforcement

This section describes the LCR algorithm, which provides an in-depth elaboration of LCR, including multi-layer feature analysis and multi-loss fusion mechanism, mask learning, and analysis of cognitive patterns, to effectively detect and mitigate backdoor attacks on DNNs, and to achieve accurate detection of backdoor samples with robustness enhancement.

3.1. Multi-loss Fusion Objective Function

Given a DNNs model f_θ and an input image $x \in X \subset R^{\omega \times h \times c}$ (ω, h, c are the width, height, and channel, respectively), our method learns an input mask m to extract the minimum pattern from x by solving the following optimization problem:

$$\begin{aligned} \underset{m}{\operatorname{argmin}} \| f_\theta(x) - f_\theta(x_{cp}) \|_1 + \alpha \| m \|_1 + \beta \mathrm{TV}(m) + \| f_\theta(x) - f_\theta(x_n) \|_1 \\ + \| f_\theta(x_s) - f_\theta(x_d) \|_1 + H(x_{cp}, x, \theta) + KL(x_{cp}, x, \theta) \end{aligned} \quad (2)$$

In the first term of the equation, x_{cp} is the distilled cognitive pattern, $f_\theta(x)$ is the model's output for the original input, $f_\theta(x_{cp})$ is the model's output for the cognitive pattern, and $\| f_\theta(x) - f_\theta(x_{cp}) \|_1$ represents the L1 norm of the difference in outputs, encouraging the cognitive pattern to preserve the original prediction. Where x_{cp} expression as:

$$x_{cp} = x \odot m + (1 - m) \odot \delta \quad (3)$$

Where δ is a c-dimensional random noise vector, $\odot$ is the element-wise multiplication applied to all the channels.

The m, in the second and third items, is a learnable 2D input mask that does not include the color channels, $\| m \|_1$ is the L1 norm of the mask, encouraging sparsity in the mask. $TV(m)$ denotes the total variation of the mask, promoting smoothness, α and β are two hyperparameters balancing the three terms.

\subsubsection{Noise disturbance}

(1) Loss of noise robustness

For our method to better learn to deal with random noise in the data, the learning of the difference between the predicted value and the true value is enhanced. We force the model to be insensitive to noise by comparing the difference between the model's output on the original input and the perturbed input and training the model by minimizing this difference in the third item in Equation 2. $\| f_\theta(x) - f_\theta(x_n) \|_1$ is the L1 norm between the disturbed image logits and the original logits. Where x_n is the logits of the disturbed image obtained by adding Gaussian noise with the same size as the original image to the original image. x_n is expressed as:

$$x_n = x + \delta \quad (4)$$

This step improves the robustness of the model to small perturbations or noise in the input data, allowing the model to effectively deal with complex and changing environments with better robustness, and also effectively avoiding overfitting to unknown data.

(2) Residual loss of cross-layer features

Meanwhile, in order to strengthen the cognitive patterns extracted by the model and mine the potential cognitive information, the LCR solves the feature residuals of the fifth term and transforms them into the cross-layer losses between the levels in Equation 2. Where $f_\theta(x_s)$ is the feature output of the shallow level and $f_\theta(x_d)$ is the feature output of the deep level. In deep learning models, the shallow layer (the layer closest to the input layer) tends to extract the most essential features such as edges and textures of the data, while the deep layer (the layer closest to the output layer) learns an abstract representation of the higher-level features. Cross-layer loss ensures a high degree of consistency between the shallow and deep layers by defining a loss function to measure the differences or residuals between the elemental layers, which can enable the model to better learn more robust elemental representations. On this basis, based on the mutually independent feature representations of the implicit layer and the output layer, the LCR can effectively identify and reject malicious information.

(3) Cross entropy loss

In addition, LCR uses cross-entropy to measure the difference between the model output of the cognitive pattems $f_\theta(xcp)$ and the true label y so that the model learns faster. By minimizing the cross-entropy loss, the parameters of the neural network model are optimized so that the model's classification results are as close as possible to the true label distribution, thus improving the model's performance in the face of various malicious attacks and hostile situations. The sixth term in Equation 2 is specified as:

$$H(x_{cp}, x, \theta) = L(f_\theta(x_{cp}), y) \cdot \left(1 - p_y(x, \theta)\right) \quad (5)$$

Here, $p_y(x, \theta)$ denotes the probability that the model correctly predicts the category y of the clean sample x. Based on this, this adjustment factor is introduced to enable the model to increase the value of the loss function and penalize samples subject to backdoor attacks if the model's prediction of clean samples is not highly accurate. The purpose of this approach is to enable the model to better focus its attention on the processing of existing attack samples, thus increasing the resistance to malicious code.

(4) Kullback-Leibler scattering loss

Finally, in this paper, by introducing the KL scatter as the last term of the LCR objective function:

$$KL(x_{cp}, x, \theta) = KL(p(x, \theta) || p(x_{cp}, \theta)) \cdot \left(1 - p_y(x, \theta)\right) \quad (6)$$

Thereby, the difference between the two probability distributions of the original input $p(x, \theta)$ and the cognitive model $p(x_{cp}, \theta)$ is more finely measured, especially when equally weighted in case of categorization errors or other special cases. By minimizing the KL dispersion, the distributions can be effectively tuned for better results and improved model robustness during the learning and optimization process of the model.

3.2. Visualization and Analysis of Masking and Cognitive Patterns

Earlier studies have shown that the backdoor model utilizes only the most basic sparse correlations in triggering the presence of patterns and does not take into account the information of the overall image. This subsection explores the inference mechanism of the model in both the clean and backdoor sample cases through a visualization experiment. The experiment is conducted on CIFAR-10 using the VGG-16 architecture. LCR implements a backdoor attack with a poisoning rate of 1% to create the backdoor model. Subsequently, clean patterns (Cognitive Pattern, CP) are extracted from the clean training images of the backdoor model BadNets and backdoor CPs are obtained from the backdoor training images using the corresponding models. A visualization of the extracted CPs and their associated masks is given in Figure 1.

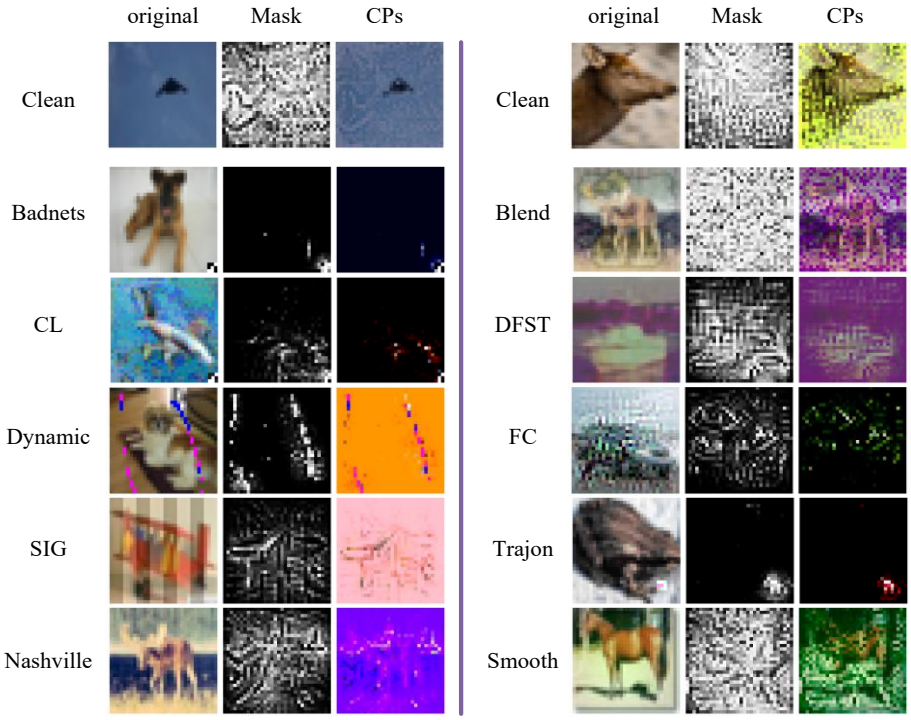

Figure 1. Visualization of masks and CPs in terms of Badnets

As shown in the first row of Fig. 1, the masks and CPs of the clean images in the backdoor model are large and semantically linked to the main object, suggesting that the model appears to utilize the actual content of the images for prediction. In summary, backdoor associations are much simpler than natural associations, regardless of which trigger model is used. Therefore, LCR uses the size of the distillation mask to effectively detect backdoor samples.

4. Experimental Results and Analysis

4.1. Experimental Setup

(1) Dataset and model architecture

In order to systematically evaluate the effectiveness of the proposed LCR method in detecting and defending against a variety of backdoor attacks, the experiments in this chapter are mainly conducted on two publicly available datasets, CIFAR10 [7] and GTSRB [8]. CIFAR10, a commonly used dataset for image categorization, contains 60,000 32×32 color images in 10 categories. According to the general convention of using CIFAR10, this dataset is divided into a training set and a test set, where the training set contains 50,000 images and the test set contains 10,000 images. GTSRB is a benchmark dataset for traffic sign recognition in Germany. This dataset contains 39209 images of different traffic signs of 32×32 in 43 categories covering various shapes, colors and backgrounds.

Five architectures were used for the experiments including VGG16, ResNet-18, PreActivationResNet101, MobileNetV2 and GoogLeNet.

(2) Attack configuration

In this paper, 10 advanced backdoor attacks are used, including BadNets [9], Blend [10], CL [11], DFST [12], Dynamic [13], FC [14], SIG [15], Trojan [16], Nashivell [17] and Smooth [18]. The implementation details of the backdoor attacks used in the experiments are summarized in Table 1.

Table 1. Configuration and implementation details of the backdoor attack

Attack	Trigger Size	Trigger Mode	Trigger Type	Strategy
BadNets	Patch	Grid	Class-wise	Pixel modification
Blend	Full image	Hello Kitty	Class-wise	Blend in
CL	Full image	Patch + Adv Noise	Class-wise	Optimizations
DFST	Full image	Style transfer	Sample-specific	GAN
Dynamic	Few pixels	Mask generator	Sample-specific	Mask Generator
FC	Full image	Optimized Noise	Sample-specific	Optimization
SIG	Full image	Periodical signal	Class-wise	Blend in
Nashville	Full image	Style transfer	Sample-specific	Instagram filter
Trojan	Patch	Watermark	Class-wise	Pixel modification
Smooth	Full image	Style transfer	Sample-specific	Smooth frequency space

A categorization pattern is when the same pattern is applied to all images of a certain category, while a sample-specific pattern is when each sample has a specific pattern. LCR has verified that all attacks are valid on the test set. The visualization of the backdoor samples is shown in Section 4.2.

(3) Detection Configuration

This paper compares LCR with five state-of-the-art backdoor sample detection methods: AC [19], ABL [20], Frequency [18], SS [21], STRIP [22] and CD [23]. For test-time detection, the model's predictions were used as labels for AC and SS, and ABL was excluded from this experiment because it could not be applied at test time. Detection

Configuration ABL is a hybrid approach combining backdoor sample detection and mitigation, and was used only for sample detection in this study, called backdoor sample isolation. In detection, the average of the sample-specific training losses during the first 20 cycles of model training is used as the confidence score. The AC, SS, and STRIP methods focus on detecting toxic samples with their original settings. The SS method uses the singularity vector of the upper-right singular value decomposition (SVD) as the confidence score, whereas STRIP uses all the training data as the sample pool. In the frequency defense, the image is transformed to frequency space by the discrete cosine transform and the detector is trained on the GTSRB using a variety of triggers without assumptions about the type or location of the triggers. Frequency defense methods were excluded from the evaluation of the GTSRB dataset. For testing, no baseline methods other than STRIP can be directly applied. Model predictions were used instead because AC and SS require real labeling Y. The ABL method was excluded from testing because it requires training statistics and has no intuitive solution.

(4) LCR implementation details

For the CIFAR10 experiment, 60 periods were trained using the SGD optimizer with weights decaying 5 × 104 and an initial learning rate of 0.1 decaying 0.1 at the 45th period. for GTSRB, the same settings as for CIFAR10 were used, except for 40 periods and decay at the 35th period.

In the LCR, β was set to 10. for models trained on CIFAR10 and GTSRB, α was set to 0.01/0.001. in this paper, the input mask was learned using the Adam optimizer, with an initial learning rate of 0.1, β_1=0.1, β_2=0.1 and the training steps were set to 100, and the mask was taken through the hyperbolic tangent scaling to the range [0, 1].

(5) Evaluation Metrics

In this paper, AUROC is used to evaluate the performance of the model under different thresholds. The higher the value of AUROC the higher the correctness rate and the closer to 1.0 the truer the detection method is.

4.2. Masking and Cognitive Pattern Visualization

In order to be able to distinguish the ten backdoor attack methods more clearly and visually compare the image feature information represented by the masks and CPs, in this paper, we have done visualization experiments for all the attacks, and extracted the masks and CPs from the model VGG-16 using the LCR implementation with a poisoning rate of 1%. for the ten baseline backdoor attack methods, we have shown the inputs of the CIFAR dataset trained on the model VGG16 for each of them. images, learned masks, and CPs are visualized and a comparison between clean and backdoor samples is included. Each row of images in Fig. 2 contains the input image (first column), the learning mask (second column), and the CP (third column), which are randomly selected clean and backdoor samples from each corresponding experiment evaluated in Section 3.3.

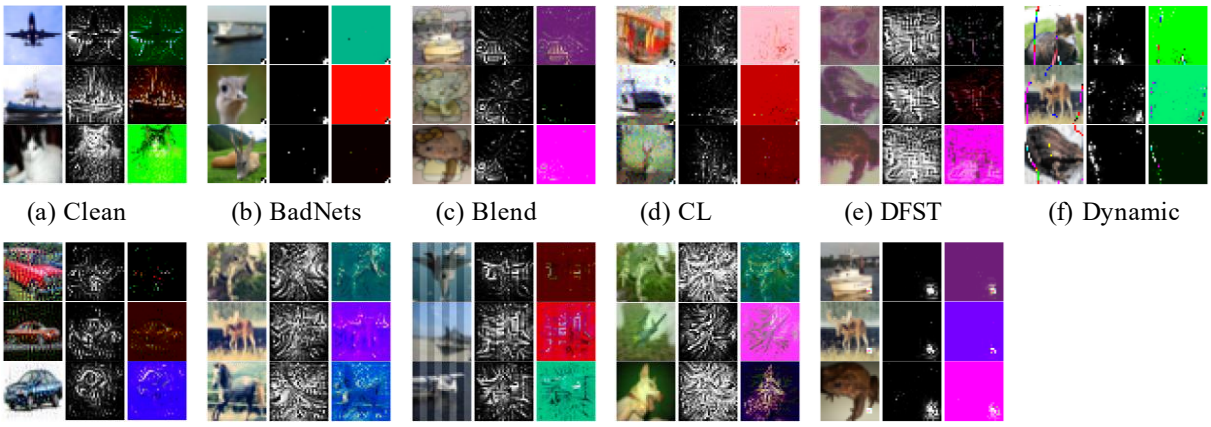

Figure 2. Learned masks and CPs for the input images corresponding to various attack methods on CIFAR10

The difference between the backdoor samples and the clean samples for the different attacks can be seen comparatively in the images visualized for these samples, as well as compared with the samples for the other backdoor attacks, which are also different. In contrast, for the three types of attacks that use patches or scattered pixels as triggers (BadNets, Trojan, and Dynamic), the CPs of the backdoor images reflect their respective triggering modes, while the masks emphasize the key parts of the triggers. In stark contrast, the remaining seven attacks are triggered using full images, but they have a very sparse number of CPs and lack semantic associations. This suggests that the model indeed relies on backdoor features to predict category labels.

For all backdoor samples, their masks and CPs clearly ignore the feature information of the samples compared to the clean samples, leaving only the backdoor trigger critical part. Although some attacks retain some sample features, their masks and CPs do not show them completely and are mixed with trigger features. The comparison between the clean and backdoor samples shows that the number of features in the mask and CP is very sparse and lacks semantic associations.

4.3. Comparative Analysis of Defense Performance

In this study, we conduct a defense study of backdoor attacks on two datasets, CIFAR10 and GTSRB, to compare the defense performance of AC, ABL, Frequency, SS, STRIP, and CD algorithms on ten types of backdoor attacks, namely BadNets, Blend, CL, DFST, Dynamic, FC, SIG, Trojan, Nashivell, and Smooth. defense performance on backdoor attacks. Table 3 summarizes the average AUROC results of this study with the detection performance of all defense methods on the five model architectures of VGG16, RN-18, PARN-101, MobileV2 and GoogLeNet. The performance was evaluated experimentally on the training and test sets with a poisoning rate of 1%. The average results in Table 2

show that the LCR approach significantly outperforms the baseline defense technique in almost every backdoor attack scenario across 5 model architectures and 2 datasets. The best results are shown in bold. The data also clearly shows that the various models exhibit different defense responses.

Table 2. Average AUROC (%) of defense for 5 architectures on training/test set

Dataset	Attack	ABL	AC	Frequency	SS	STRIP	CD	Ours
CIFAR10	BadNets	65.19/-	51.86/60.22	95.44/95.56	68.03/65.67	97.62/96.92	94.46/94.01	99.76/98.99
	Blend	51.86/-	46.46/51.63	87.96/46.57	47.77/46.57	68.53/64.10	73.80/71.42	91.88/84.45
	CL	71.34/-	42.65/29.21	95.42/94.88	44.09/39.30	97.75/95.03	98.43/76.82	99.41/84.84
	DFST	67.08/-	36.19/40.83	54.88/56.93	37.11/24.69	54.51/53.14	58.80/56.40	63.28/68.39
	Dynamic	63.30/-	54.29/47.74	73.62/74.38	56.22/50.29	86.58/81.28	98.72/97.16	99.38/97.65
	FC	64.34/-	49.35/82.58	59.78/58.11	45.35/66.17	78.79/75.90	98.08/95.37	98.46/97.26
	SIG	83.40/-	53.90/37.35	91.37/80.97	52.65/40.28	78.77/51.46	89.02/75.68	92.85/77.37
	Smooth	39.58/-	50.54/31.26	46.61/43.62	55.34/44.78	57.82/56.66	73.44/69.41	82.89/79.50
	Nashville	28.15/-	35.23/32.52	33.84/38.20	41.45/38.44	46.26/46.11	89.66/86.02	84.48/84.78
	Trojan	61.90/-	47.71/55.58	94.16/93.80	63.97/60.60	89.26/90.20	92.81/92.38	98.36/95.87
	Average	59.61/-	45.25/47.19	73.03/71.93	51.20/47.68	75.59/71.08	86.72/81.47	90.15/85.79
GTSRB	BadNets	31.93/-	82.93/61.27		64.72/77.69	60.68/62.72	89.78/90.66	99.04/98.93
Average		57.10	20.10/58.00	75.72/71.93	52.43/65.25	74.23/69.78	87.00/86.07	91.80/87.93

Table 2 compares the detection performance (AUROC%) of the LCR defense approach with a variety of other defense schemes for different datasets and attack types. From the average results, the LCR approach shows significant advantages. On the CIFAR10 dataset, LCR achieves an average AUROC of 90.15% on the training set, which is significantly higher than other methods. Optimal performance is achieved in 9 out of 10 attack types, especially in the BadNets attack which reaches an excellent performance of 99.76%/98.99%. For the combined performance across datasets, on the GTSRB dataset, LCR achieves a high AUROC of 99.04%/98.93% for detecting BadNets attacks, and the overall average AUROC reaches 91.80%/87.93%, which is comprehensively better than the other compared methods.

Among the five baselines, SS performs poorly because it needs to detect backdoor labels. However, in this case, the defender has no knowledge of backdoor labels. For SS to work, it may need to be paired with a backdoor label detection method. In addition, AC is not very good, although it can sometimes repel different types of attacks well. both AC and SS are deep feature-based detection techniques. In addition, the stability aspect is compared with other methods that fluctuate a lot (e.g., the AC method performs only 29.21% of the test set in CL attacks), LCR maintains a stable and efficient detection capability in all attack types.

These data fully prove that the LCR method has significant advantages against a wide range of backdoor attacks, which can maintain high detection accuracy and stable performance across different datasets and attack types, providing a reliable solution for backdoor defense.

4.4. Adaptation at Different Poisoning Rates

In order to validate the effectiveness of the proposed method in this study for backdoor attack defense when the poisoning rate varies, the performance is evaluated experimentally on the training set and the test set with the poisoning rate of 2.5% while keeping the same environment and setting the same parameters. Also experiments are conducted to validate the backdoor attack by cleaning accuracy (CA) and attack success rate (ASR), CA is evaluated on clean test set. And AUROC is evaluated after defense of LCR's approach. In this subsection, only experiments are carried out on the effectiveness of LCR defense for all attacks on CIFAR10 dataset on VGG16 architecture and the results are presented in Table 3.

Table 3. Average AUROC (%) for defense of CIFAR10 training/test set on VGG16

Attack	CA	ASR	Ours
BadNets	80.79	100.00	99.74/99.48
Blend	81.98	92.80	92.33/89.76
CL	81.96	91.60	99.97/95.39
DFST	81.31	100.00	96.42/96.30
Dynamic	82.62	98.40	98.80/98.39
FC	89.46	98.00	99.27/97.35
SIG	89.37	97.20	96.80/92.96
Smooth	89.78	94.40	88.01/83.32
Nashville	89.36	97.20	91.64/87.25
Trojan	81.05	95.60	97.47/96.02
Average	84.77	96.52	96.05/93.62

The average CA value for all attacks is 84.77% and the ASR values are generally high, indicating that the model performs more consistently on clean data and is very effective when undefended.

The method proposed in this study performs well on most of the attacks, with AUROC values close to or above 90% on both the training and test sets. The defense against Blend and Smooth attacks is slightly weaker, probably due to the characteristics of these attacks that make them more difficult to defend. The average AUROC is 96.05% (training set) and 93.62% (test set), which indicates that the defense method is effective in distinguishing backdoor attacks from normal samples at different poisoning rates. In conclusion, the present defense method performs well in most backdoor attacks, especially in maintaining high cleaning accuracy while significantly reducing the success rate of the attack, which validates its effectiveness.

5. Conclusion

In this study, a backdoor defense method based on potential cognitive reinforcement is proposed to extract cognitive patterns of the model to identify and suppress backdoor

triggers by using multi-layer feature representations of the intermediate and output layers. Compared with existing methods, the latent cognitive reinforcement defense strategy does not require a priori knowledge or clean data, and possesses greater generality and robustness. Under two datasets, CIFAR-10 and GTSRB, five network architectures, and ten backdoor attacks, LCR significantly outperforms existing defense techniques in terms of average AUROC and stability. The results show that LCR is an efficient and scalable defense mechanism for the security of practical deep learning models.

References

[1] HANIF A, SHAMSHAD F, AWAIS M, et al. BAPLe: Backdoor Attacks on Medical Foundational Models using Prompt Learning [EB/OL]. arXiv, 2024 [2025-05-05]. http://arxiv.org/abs/2408.07440.

[2] NASERI M, HAN Y, CRISTOFARO E D. BadVFL: Backdoor Attacks in Vertical Federated Learning [EB/OL]. arXiv, 2023 [2025-05-05]. http://arxiv.org/abs/2304.08847.

[3] WANG B, YAO Y, SHAN S, et al. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks [C/OL]//2019 IEEE Symposium on Security and Privacy (SP). San Francisco, CA, USA: IEEE, 2019: 707-723 [2025-02-26]. https://ieeexplore.ieee.org/document/8835365/.

[4] LIU K, DOLAN-GAVITT B, GARG S. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks [EB/OL]. arXiv, 2018 [2025-02-26]. http://arxiv.org/abs/1805.12185.

[5] JEDDI A, SHAFIEE M J, WONG A. A Simple Fine-tuning Is All You Need: Towards Robust Deep Learning Via Adversarial Fine-tuning [EB/OL]. arXiv, 2020 [2025-04-06]. http://arxiv.org/abs/2012.13628.

[6] KOH P W, STEINHARDT J, LIANG P. Stronger data poisoning attacks break data sanitization defenses [J]. Machine Learning, 2022, 111(1): 1-47.

[7] KRIZHEVSKY A. Learning Multiple Layers of Features from Tiny Images [J].

[8] HOUBEN S, STALLKAMP J, SALMEN J, et al. Detection of traffic signs in real-world images: The German traffic sign detection benchmark [C/OL]//The 2013 International Joint Conference on Neural Networks (IJCNN). Dallas, TX, USA: IEEE, 2013: 1-8 [2025-02-26]. http://ieeexplore.ieee.org/document/6706807/.

[9] GU T, DOLAN-GAVITT B, GARG S. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain [J/OL]. arXiv, 2017 [2023-11-19]. http://doc.paperpass.com/foreign/rgArti2017137662331.html.

[10] CHEN X, LIU C, LI B, et al. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning [J/OL]. 2017 [2023-11-19]. https://arxiv.org/abs/1712.05526.

[11] TURNER A, TSIPRAS D, MĄDRY A. Clean-Label Backdoor Attacks [J].

[12] CHENG S, LIU Y, MA S, et al. Deep Feature Space Trojan Attack of Neural Networks by Controlled Detoxification [EB/OL]. arXiv, 2021 [2025-02-26]. http://arxiv.org/abs/2012.11212.

[13] NGUYEN T A, TRAN T A. Input-Aware Dynamic Backdoor Attack [J].

[14] SHAFAHI A, HUANG W R, NAJIBI M, et al. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks [J].

[15] BARNI M, KALLAS K, TONDI B. A new Backdoor Attack in CNNs by training set corruption without label poisoning [EB/OL]. arXiv, 2019 [2025-02-26]. http://arxiv.org/abs/1902.11237.

[16] LIU Y, MA S, AAFER Y, et al. Trojaning Attack on Neural Networks [C/OL]//Proceedings 2018 Network and Distributed System Security Symposium. San Diego, CA: Internet Society, 2018 [2025-02-26]. https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_03A-5_Liu_paper.pdf.

[17] LIU Y, LEE W C, TAO G, et al. ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation [C/OL]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London United Kingdom: ACM, 2019: 1265-1282 [2025-02-26]. https://dl.acm.org/doi/10.1145/3319535.3363216.

[18] ZENG Y, PARK W, MAO Z M, et al. Rethinking the Backdoor Attacks' Triggers: A Frequency Perspective [C/OL]//2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada: IEEE, 2021: 16453-16461 [2025-02-26]. https://ieeexplore.ieee.org/document/9710118/.

[19] CHEN B, CARVALHO W, BARACALDO N, et al. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering [EB/OL]. arXiv, 2018 [2025-02-26]. http://arxiv.org/abs/1811.03728.

[20] LI Y, LYU X. Anti-Backdoor Learning: Training Clean Models on Poisoned Data [J].

[21] TRAN B, LI J, MA A. Spectral Signatures in Backdoor Attacks [J].

[22] GAO Y, XU C, WANG D, et al. STRIP: A Defence Against Trojan Attacks on Deep Neural Networks [EB/OL]. arXiv, 2020[2025-02-26]. http://arxiv.org/abs/1902.06531.

[23] HUANG H, MA X, ERFANI S, et al. DISTILLING COGNITIVE BACKDOOR PATTERNS WITHIN AN IMAGE [J]. 2023.