PIFC-CLD: Poison image traceback via feature clustering and euclidean norm distance for clean-label attacks in deep neural networks

Abdulruhman Abomakhleb [a,1,*], Kamarularifin Abd Jalil [a,2,*], Alya Geogiana Buja [b,3], Abdulraqeb Alhammadi [c,4]

[a] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam 40450, Selangor, Malaysia
[b] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Merlimau 77300, Melaka, Malaysia
[c] Faculty of Artificial Intelligence, Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia
[1] abd.abomakhleb@gmail.com; [2] kamarul@tmsk.uitm.edu.my; [3] geogiana@uitm.edu.my; [4] abdulraqeb.alhammadi@utm.my
* corresponding author

ARTICLE INFO

Article history
Received September 2, 2025
Revised October 13, 2025
Accepted October 14, 2025
Available online February 28, 2026

Keywords
Bullseye polytope attacks
Clean-label attacks
Adversarial attacks
Digital forensics
Euclidean norm similarity

ABSTRACT

Clean-label poisoning attacks pose a stealthy and potent threat to deep neural networks (DNNs), particularly when models rely on publicly available or outsourced training data. Among these attacks, the Bullseye Polytope method is highly transferable and capable of evading state-of-the-art defenses such as Deep k-NN. To counter this, we propose Poison Image Traceback via Feature Clustering (PIFC-CLD) - a novel forensic approach that leverages Euclidean norm distances to detect and trace clean-label attacks in DNNs. PIFC exploits the geometric consistency of feature representations to identify poisoned samples responsible for model misclassifications. Unlike traditional majority-vote-based defenses, PIFC-CLD performs clustering in feature space and detects poisoned samples based on their proximity to misclassified targets using Euclidean distance. We evaluate our approach under Bullseye Polytope attack scenarios using the CIFAR-10 dataset and WideResNet architectures. PIFC-CLD achieves 99% precision, 95% recall, and a 96% F1 score at k = 25 and ε = 0.2, demonstrating robust performance against Bullseye Polytope attacks. Furthermore, our algorithm exhibits strong resilience to parameter variations while minimizing false positives and preserving model integrity. This work bridges the gap between digital forensics and adversarial machine learning, offering a lightweight, model-agnostic, and interpretable solution for secure model training in adversarial environments.

1. Introduction

Deep neural networks (DNNs) have revolutionized numerous fields by achieving state of the art performance in tasks such as image recognition, natural language processing, and autonomous decision-making. Despite these remarkable advancements, their vulnerability to data poisoning, particularly clean-label attacks, raises critical concerns regarding their robustness and trustworthiness. As machine learning models increasingly depend on training data sourced from public or semi-public platforms, including social networks, collaborative forums, and multimedia sharing services, adversaries gain opportunities to subtly manipulate model behavior through strategic data injections.

Recent advancements in artificial intelligence have enabled the development of real-time threat-detection systems to protect critical infrastructure, including energy grids, healthcare systems, and

transportation networks. These AI-driven systems utilize machine learning algorithms to dynamically detect and respond to cyber threats, thereby enhancing both the speed and accuracy of cybersecurity defenses. However, existing solutions often lack adaptability to evolving attack vectors and fail to provide transparent and explainable decision-making processes [1].

Recent studies have extensively reviewed the use of Convolutional Neural Networks (CNNs) in image forensics, encompassing tasks such as copy-move forgery detection, image splicing, noise inconsistency analysis, and data poisoning identification. The findings highlight the increasing reliance on deep learning-based methods, driven by CNNs' superior ability to uncover subtle and complex image manipulations that often elude traditional forensic techniques. Despite these advancements, existing CNN-based forensic approaches exhibit notable limitations, particularly in their ability to generalize to previously unseen manipulation techniques and to remain robust to adaptive adversarial strategies. Notably, the review identifies data poisoning detection as a significantly underexplored area in image forensics. This gap highlights the urgent need to develop specialized, resilient forensic frameworks that accurately trace and isolate poisoned instances within deep neural network pipelines, particularly under clean-label attack conditions [2].

Recent studies have underscored the growing threat of AI poisoning, wherein adversaries deliberately manipulate training data to undermine AI systems. Such attacks compromise the integrity, reliability, and trustworthiness of AI models by either embedding malicious behaviors (poisoning). Despite increased awareness, current defenses often lack the granularity, traceability, and forensic capability needed to isolate the source and mechanism of these threats. This underscores a critical research gap: the need for robust forensic methodologies that can trace poisoned inputs within the AI pipeline, ensuring resilient and explainable AI deployments in security-critical environments [3]–[5].

Clean-label attacks generate seemingly benign samples with minimal perturbations that closely resemble legitimate data. As a result, they can bypass standard data validation procedures and induce targeted misclassifications during inference, posing a significant threat to the reliability and security of deployed systems. Clean-label attacks modify only the input data while keeping the labels unchanged, which makes them particularly difficult to detect [6]–[11]. This increasing threat highlights the urgent need for robust defense mechanisms to safeguard the security and integrity of DNNs in adversarial settings. Clean-label attacks can also be divided into triggerless and backdoor clean-label attacks. In contrast to dirty-label backdoor attacks, clean-label attacks [7], [12]–[18] prevent adversaries from altering the label of the poisoned data. A clean-label, non-trigger attack aims to misclassify one piece of unmodified test data. For instance, [19] developed the first clean-label triggerless attack in which the attacker injects poisoned data to disrupt the feature region of the targeted data. Several studies [20], [21] have demonstrated that positioning poison data on a convex polytope surrounding the target data can significantly enhance the effectiveness of clean-label attacks. These attacks are particularly potent when the victim model employs transfer learning, and the attacker has white-box access to the pre-trained model parameters. More recently, [22] proposed an attack that leverages gradient alignment between poison and target samples, effectively extending clean-label attack strategies to training scenarios initiated from scratch.

In this paper, we present a study on tracing and detecting clean-label poisoning attacks on neural networks, with a focus on Bullseye Polytope attacks [20] applied to the CIFAR-10 dataset. We propose Poison Image Traceback via Feature Clustering (PIFC-CLD), a novel algorithm designed as a reactive forensic approach to identify and isolate suspicious training samples resulting from clean-label poisoning. By leveraging feature-space clustering, the algorithm distinguishes between benign and poisoned samples based on subtle deviations introduced through clean-label poisoning. After extracting candidate samples from the same class as the misclassified target image x_t, the algorithm applies k-means clustering ($k = 2$) to partition their feature representations into two clusters. It then computes the centroid (mean feature vector) of each cluster and calculates the Euclidean distance between the misclassified target image x_t and each cluster center. The cluster with the smaller distance to x_t is inferred to contain the poisoned samples. This Euclidean-distance-based approach enables the algorithm to trace poisoned data by exploiting local inconsistencies in the feature space introduced during clean-label attacks.

To validate its effectiveness, we evaluate PIFC-CLD against the Bullseye Polytope attack, a highly transferable and representative clean-label poisoning method known for its success across various DNNs architectures.

This paper makes the following key contributions to the field of backdoor attack detection:

- We propose a novel PIFC-CLD algorithm for reactive defense for clean-label poisoning attacks. We evaluate it against state-of-the-art clean-label data poisoning attacks, using a WideResNet model
- We re-implement the Bullseye Polytope attack [16] and demonstrate that our proposed PIFC-CLD defense can perform a traceback and detect poison attacks in the trained victim models.
- We empirically evaluate our proposed approach on the CIFAR-10 image classification task and demonstrate its effectiveness. We achieve a 95% recall and an F1 Score of 95%, indicating consistently high detection performance

2. Related Work

Table 1 outlines various defense mechanisms to mitigate clean-label poisoning attacks in machine learning models. These defenses address the challenges of distinguishing between subtle poisoned samples and clean ones, emphasizing their methods and inherent limitations. Gao et al. [23] proposed a method that obfuscates adversarial samples to mask their triggers, thereby enhancing model robustness to clean-label attacks. However, the process is computationally expensive and can unknowingly reduce the accuracy of clean, non-adversarial data. Peri et al. [24] introduced the Deep k-NN Defense, which leverages nearest-neighbor searches in feature space to detect anomalies. While effective in many cases, this approach requires careful tuning of the feature extraction layer and the k value. It also struggles when poisoned samples closely resemble the clean data distribution. Hong et al. [25] developed a novel approach using Diffusion Denoising as a Certified Defense. This method models and removes noise from poisoned data through a diffusion process, increasing robustness. However, it is computationally demanding, less effective against subtle attacks, and requires large datasets to function optimally. Huang et al. [26] introduced MetaPoison, a meta-learning-based approach for generating and defending against poison attacks. Despite its innovative design, it is computationally intensive and requires extensive parameter tuning to achieve optimal effectiveness. This table highlights the trade-offs between defense robustness and practical applicability, emphasizing the need to balance computational efficiency with detection precision in clean-label attack defenses.

Table 1. Related work: Defenses for Clean-Label Attacks

Authors and Year	Defense Method	Weaknesses
Gao et al., 2019c [23]	Obfuscates adversarial samples to mask triggers, making models more robust to clean-label attacks.	It is computationally intensive and may reduce accuracy on clean, non-adversarial samples.
Peri et al., 2019b [24]	Deep k-NN Defense	It requires tuning the feature extraction layer and *k* value. May struggle if poisoned samples are very similar to the clean distribution
Hong et al., 2024 [25]	Denoising: models and removes noise using a diffusion process to defend against clean-label poisoning.	Computationally expensive;; assumes specific noise structure, vulnerable if the attack deviates from the assumption.
Huang et al., 2020 [26]	Meta Poison	Computationally intensive; may require extensive tuning for optimal effectiveness.

2.1. Previous Method: Deep k-NN Defense

The BP attack targets vulnerabilities in defenses, such as Deep k-NN by crafting adversarial examples that exploit the k-NN mechanism's dependence on feature-space distances [27]. Deep k-NN uses a k-nearest neighbor approach in the feature space of deep neural networks to detect outliers or adversarial

samples. The assumption is that clean samples cluster closely in the feature space, while adversarial examples appear as outliers.

The attack generates adversarial examples by solving an optimization problem that minimizes the distance between them and the target cluster in the feature space. The adversary carefully positions the adversarial instance within the "polytope" formed by the k-NN of the target class, bypassing the defense. This precision enables the adversarial example to mimic legitimate samples, thereby tricking the Deep k-NN defense into misclassifying it as part of the target class.

3. Method

3.1. PIFC-CLD: Poison Image Traceback via Feature Clustering for Clean-Label Attacks in DNNs

The Feature Clustering measures in image measurement involve grouping pixels or image regions based on their feature similarity in the feature space. Its application introduces constraints and regularization, thereby mitigating the effectiveness of Bullseye Polytope attacks.

- Distance-based Constraints: Defenses can detect abnormal perturbations introduced by adversarial examples by monitoring the Euclidean distance between samples and the centroid or k-nearest neighbors in the feature space. Adversarial examples generated by Bullseye Polytope tend to increase the Euclidean distance subtly to mimic legitimate samples. Enforcing strict thresholds on acceptable distances helps identify such anomalies.
- Normalization in Training: Regularizing the Euclidean Norm during training ensures that the features of clean samples form tight, distinct clusters in the feature space. This makes it harder for adversarial examples to find a position within the target class polytope without being detected.
- Enhanced Robustness: Defenses can be designed to penalize samples with an unusually high Euclidean Norm or deviations from expected norms. This approach forces adversarial examples to require larger perturbations to succeed, making them easier to detect.

By leveraging the Euclidean Norm, defenses can create tighter boundaries around legitimate samples in feature space, significantly reducing the attack surface for Bullseye Polytope and similar attacks.

3.2. Conceptual Framework for Detecting and Mitigating Clean-Label Poisoning Attacks in Deep Neural Networks

The framework integrates advanced Feature Clustering and Euclidean-norm distance into a DNN model to detect clean and poisoned labels in a dataset. This is critical in addressing data poisoning attacks that manipulate benign data to trigger model misclassifications, as illustrated in Fig. 1.

3.3. Input Data Preparation

The input dataset comprises two main components: benign data and poisoned data. The benign data consist of clean and accurate samples that represent the original training dataset, while the poisoned data consist of malicious samples deliberately embedded into the dataset with the intention of inducing misclassifications in the learning model. Both types of data are stored together within the dataset, as illustrated in the diagram under the label “Benign Data + Bullseye Polytope (BP)”.

3.4. Feature Extraction via DNNs

The input dataset is fed into the DNNs model, which extracts high-dimensional feature representations. The DNNs process the input through multiple layers. The Input layer accepts raw data (both benign and poisoned). Hidden layers extract meaningful features from the data to distinguish between benign and poisoned patterns. The output layer produces the classification outcome. The misclassification output identifies instances where poisoned samples lead to incorrect predictions.

3.5. Detection of Clean Labels Using the PIFC-CLD

The PIFC-CLD Algorithm uses K-means clustering on the feature vectors to divide them into two clusters and calculates the deviation between the expected, clean feature values and the observed values

from potentially poisoned samples. It also computes the Euclidean distance from the misclassified target image to each cluster center.

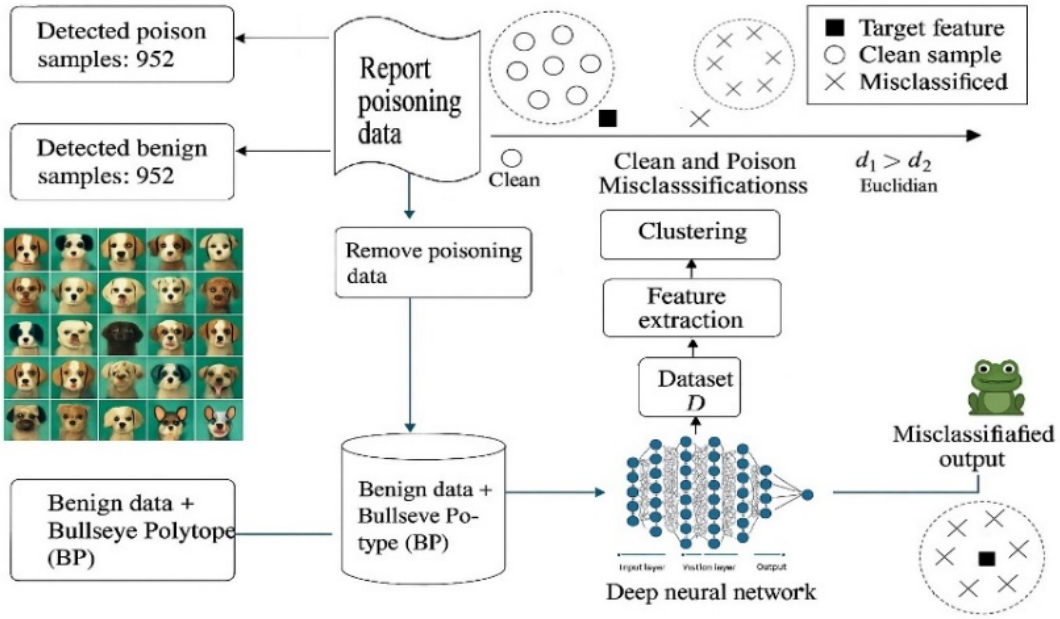

Fig. 1. Proposed Conceptual Framework for Detecting and Mitigating Clean-Label Poisoning Attacks in Deep Neural Networks

3.6. Feature Extractor Measurement

In this algorithm, Feature Distance Measurement plays a crucial role in determining which cluster of samples is likely to contain the poisoned data. The process involves comparing the Euclidean distance between the feature representation of a misclassified target image and the centroids of two clusters generated using the K-means algorithm. Each candidate's image (x_i) is transformed into a feature extractor$(\phi(x_i))$, using a pre-trained feature extractor from the penultimate layer of a deep neural network. The extracted feature extractors are grouped into two clusters (G_1) and (G_2) using K-means clustering.

In the clustering process, the centroids $(\mu_1)$and (μ_2) of the clusters are computed as the average of the feature extractor in each respective group. Mathematically, the centroid (μ_j) for cluster (G_j) Is defined as:

$$\mu_j = \frac{1}{|G_j|}\sum_{i\in G_j}\phi(x_i), \quad j \in \{1,2\} \tag{1}$$

where $(\phi(x_i))$ denotes the feature representation of the sample (x_i), and $(|G_j|)$ Is the number of samples in cluster (G_j).

3.7. Thresholding for Detection

In the context of clean-label poisoning detection, thresholding serves as a crucial mechanism for distinguishing between benign and poisoned training samples based on their relative positioning in feature space. To operationalize this, the algorithm computes the Euclidean distances between the feature representations of the misclassified target sample. $(\phi(x_t))$and the centroids of two clusters (μ_1) and (μ_2), derived via K-means clustering. These distances are formally defined as:

$$d_1 = |\phi(x_t) - \mu_1|_2, \quad d_2 = |\phi(x_t) - \mu_2|_2 \tag{2}$$

Here, (d_1) and (d_2) quantify the similarity between the target sample and each respective cluster centroid in the learned feature space. The underlying assumption is that poisoned samples are optimized to closely resemble the target's embedding during training, thereby residing closer to it in feature space.

The algorithm applies a distance-based decision rule: if $(d_1 > d_2)$, it implies that cluster (G_2) lies closer to the misclassified target, and therefore, the cluster (G_1) It is inferred to contain the poisoned samples. Conversely, if $(d_1 \leq d_2)$, then cluster (G_2) Designated as poisoned. This relative comparison serves as an implicit thresholding mechanism, enabling the separation of anomalous (i.e., poisoned) samples from clean data.

3.8. Feature Clustering and Euclidean Norm (PIFC-CLD Algorithm)

The feature clustering and Euclidean norm components of the framework apply the following mathematical operations to all data points. The process involves:

- Assigning each sample as clean or poisoned based on the computed Euclidean norm distance from cluster centroids.
- Detection of poisoned samples.
- Reported: The 'Report Poisoning Data' block logs the identified poisoned samples for analysis.
- Removed: Poisoned samples are eliminated from the training dataset to enhance model robustness

3.9. Output and Results

The system produces the following outputs:

- Misclassification Events: Instances where poisoned samples cause the DNNs to misclassify target inputs.
- Clean Data: Benign samples that remain after the poisoned samples have been removed.
- Poisoned Data Detection: A refined dataset, free from poisoning influences, ensuring data integrity for retraining and deployment.

The final output ensures that the DNNs perform classification tasks with improved accuracy and robustness. This framework effectively combines feature clustering and Euclidean norm with deep neural network feature extraction to identify and eliminate poisoned data. By removing malicious samples, the model maintains resilience against clean-label poisoning attacks.

3.10. Mathematical Modeling and Algorithm of Forensic Traceback Using Euclidean Similarity and Clustering

3.10.1. Feature Extractor Layer Definition

To formally define the forensic traceback process, we begin by modeling the data structure. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ Denote the dataset, where x_i in R^d is an input image and y_i in $\{0, \ldots, C-1\}$ is its corresponding class label. Let x_t be the misclassified target image, and let $\phi: R^d \to R^m$ denote a pretrained feature extractor , typically derived from the penultimate layer of a deep neural network.

For a DNN $f_\theta(x) = g_\theta(\phi(x))$, where $(\phi(x_i))$ represents the feature extractor and $g_\theta(\cdot)$the final classification layer, the extractor is given by:

$$\phi: \mathbb{R}^d \to \mathbb{R}^m, \phi(x) = h_{L-1}(x) \quad (1)$$

where $h_{L-1}$is the activation output of the $(L-1)$-th layer

Let $\mathcal{C} \subseteq \mathcal{D}$ Represent the subset of samples in the class c, where c is the predicted (but incorrect) class label for x_t. Define the set of feature extractor as:

$$F = \{\phi(x_i) \mid (x_i, y_i) \in \mathcal{C}\} \quad (2)$$

and let $\phi(x_t)$ be the feature extractor of the target image

3.10.2. Distance Metric Definition

To identify the poisoned samples responsible for the misclassification, we compute the Euclidean distance between the feature vector of x_t and every feature vector in F.

$$d_i = |\phi(x_t) - \phi(x_i)|_2 \quad \forall x_i \in \mathcal{C} \tag{5}$$

3.10.3. Neighbor Set Construction

Let $N_k(x_t) \subset \mathcal{C}$ be the $k_nearest$ neighbors of x_t under the distance metric in Equation 5, define the index set of those samples as:

$$\mathcal{I} = \{i_1, i_2, \ldots, i_k\} \tag{6}$$

and let $D = \{d_i \mid i \in \mathcal{I}\}$ Be the distances corresponding to the k −Nearest neighbors.

3.10.4. Clustering Objective Definition

We then apply $k - means$ clustering with $k = 2$ to the feature vectors $\{\phi(x_i)\}_{i\in\mathcal{I}}$, producing two clusters G_1 and G_2 with respective centroids μ_1 and μ_2.

$$\mu_1 = \frac{1}{|G_1|}\sum_{i\in G_1}\phi(x_i),\ \mu_2 = \frac{1}{|G_2|}\sum_{i\in G_2}\phi(x_i) \tag{7}$$

3.10.5. Decision Rule (Poison Cluster Identification)

Next, compute the Euclidean distance between the target image's feature vector and each cluster centroid:

$$d_1 = |\phi(x_t) - \mu_1|_2, d_2 = |\phi(x_t) - \mu_2|_2 \tag{8}$$

We assume that the poisoned samples lie closer in feature space to x_t due to the nature of clean-label attacks (e.g., Bullseye Polytope). Therefore, the cluster with the smaller distance to $\phi(x_t)$ is considered the poisoned cluster:

$$Poison\ Cluster = \begin{cases} G_1, & if\ d_1 < d_2 \\ G_2, & otherwise \end{cases} \tag{9}$$

Let the final set of suspected poisoned sample $\mathcal{P}$ indices be:

$$\mathcal{P} = \{i \in \mathcal{I} \mid x_i \in \text{Poison Cluster}\} \tag{10}$$

This process forms the basis of the "Using Feature Clustering and Euclidean Norm for Poison Image Traceback via Feature Clustering for Clean-Label Attacks in DNNs" algorithm (PIFC-CLD). The assumption is that poison instances lie closer in feature space to the target sample due to the Bullseye Polytope optimization, which explicitly minimizes such a distance.

3.11. Algorithm

This algorithm uses clustering to identify poison images in a specific class (Fig. 2). It receives a misclassified image. x_t That was likely influenced by poison images belonging to a particular class C. The goal is to discover which images in class c they are potentially poisonous.

Initially, the algorithm initializes an empty list I to store the indices of candidate poison images (line 1), and a list. $\mathcal{F}$ To collect their corresponding feature vectors (line 2). It then iterates through each sample (x_i, y_i) In the dataset $\mathcal{D}$ (line 3). For each sample, if the label y_i Matches the target class c (line 4), the sample's index i is added to the candidate list I (line 5), and its feature representation $\phi(x_i)$ is computed and stored in $\mathcal{F}$ (line 6).

Once all candidate samples have been collected, the algorithm is applied. K_means clustering on the feature vectors in $\mathcal{F}$, dividing them into two clusters ($k = 2$) (line 9). These clusters, G_1 and G_2, are based on the indices in I (line 10). The algorithm then computes the mean feature vector (i.e., the cluster center) for each group: $\mu^1 for G^1 and \mu^2 for G^2$ (line 11).

Next, the algorithm calculates the Euclidean distance between the misclassified target image and the actual target image. x_t To each of the cluster centers (line 12). Specifically, d_1 is the distance to μ_1, and d_2 Is the distance to μ_2. It compares these distances to determine which cluster is more likely to contain the poison samples: if $d_1 > d_2$The poisoned images are assumed to lie in G_1; otherwise, in G_2 (lines 13–17). Finally, the algorithm returns the indices of the suspected poison images (line 18).

Algorithm 1: Identify Poison Images via Feature Clustering

ENSURE: Indices of suspected poison images

1: Initialize an empty list of candidate indices $\mathcal{I} \leftarrow [\ ]$
2: Initialize a list of feature vectors $\mathcal{F} \leftarrow [\ ]$
3: for all $(x_i, y_i) \in \mathcal{D}$
4: If $y_i = c$
5: Append i to $\mathcal{I}$
6: Compute and append the feature vector $f_i = \phi(x_i)$ to $\mathcal{F}$
7: Perform k-means clustering with $k = 2$ on $\mathcal{F}$
8: Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the two resulting clusters (indices referring to $\mathcal{I}$)
9: Compute cluster centers:

$$\mu_1 = \frac{1}{|\mathcal{G}1|}\sum i \in \mathcal{G}_1\, \phi(x_i),\ \mu_2 = \frac{1}{|\mathcal{G}2|}\sum i \in \mathcal{G}_2\, \phi(x_i)$$

10: Compute distances from the misclassified image to each cluster center:

$$d_1 = |\phi(x_t) - \mu_1|_2, d_2 = |\phi(x_t) - \mu_2|_2$$

11: If $d_1 > d_2$
12: **Poison images** $\leftarrow \mathcal{G}_1$
13: else
14: **Poison images** $\leftarrow \mathcal{G}_2$
Indices of poison images

Fig. 2. Algorithm to identify Poison images via feature clustering

4. Results and Discussion

4.1. Dataset and Setting

As our objectives relate to identifying the data source used to train a machine learning model, we will use a set of image datasets from different sources for the same image category. Consider CIFAR-10 as the primary dataset and use other datasets to match their subsets of categories [28]. We will build a single dataset from these datasets by balancing the ratio of contributions from each. CIFAR-10 is an image dataset with 60k 32 × 32 color images from 10 classes: 'plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', and 'truck. For convenience, we enumerate these 10 classes as classes 1 to 10. The CIFAR-10 training set contains 50k images (5k per class), and the test set includes the remaining 10k images (1k per class).

4.2. Experimental Setup and Model Configuration

We evaluate the effectiveness of our Similarity-Based Poison Traceback (PIFC-CLD) approach in clean-label backdoor attacks using the CIFAR-10 dataset. The widely adopted Wide-ResNet model serves as the backbone for classification and the embedding extractor for negative class identification. This decision is based on its demonstrated effectiveness in image classification tasks and its architectural depth, which enhances feature discrimination. According to the PyTorch Hub, Wide-ResNet models are well-suited for benchmarking robustness to clean-label attacks.

4.3. Model Parameters and Hyperparameters

The Wide-ResNet-28-10 model used in our experiments has 11,181,642 trainable parameters in total. We trained the model using Stochastic Gradient Descent (SGD) with a momentum factor of 0.9

with Adam optimizer (lr=0.001, batch size=128, dropout_rate=0.3, weight_decay= 1×10^{-4}, beta = 0.9, seeds (42, 123, 999)), early stopping, and repeated runs under different random seeds to ensure reproducibility.

4.4. Clean-Label Attack Setup Using Bullseye Polytope

We implement the Bullseye Polytope attack [20] following the original papers and evaluate the effectiveness of PIFC-CLD on CIFAR-10. To simulate clean-label attacks, we injected poisoned samples into the training dataset at two poison ratios: 10% and 20%.In our experiments, target images were selected from the CIFAR-10 test set as correctly classified samples of a given source class (image_class), with a fixed target class (target_image_class). Each experiment used up to 30 target images per class pair (number_of_instances = 30).

4.5. Influence of Parameter Factors on Reactive Defense Behavior

Reactive defense mechanisms, particularly those based on local feature space analysis, such as Deep k-NN, are sensitive to key parameter configurations, including the number of poisoned samples and the neighborhood size (k). Experimental evaluations demonstrate the following insights.

- Number of Poisoned Samples: As the number of poisoned instances increases, their collective influence in feature space can overpower most benign instances. This reduces the effectiveness of defenses that rely on neighborhood label consistency. For example, when the number of poisoned samples is sufficiently high to match or exceed the number of clean samples in a local feature neighborhood, the detection performance deteriorates significantly.
- Neighborhood Size (k): The parameter k determines the number of nearest neighbors considered for conformity evaluation. Choosing a k that is too small may lead to unstable classification due to local noise, while a huge k may dilute the effect of local poison-induced anomalies

4.6. Performance Metrics

4.6.1. Effectiveness and Efficiency of Reactive Defense

We evaluated the effectiveness and efficiency of reactive defense based on similarity in four aspects:

4.6.1.1. Evaluation Metrics

- Accuracy (AC). Accuracy measures the proportion of correct predictions among the total number of input samples. In machine learning, accuracy can be calculated using the counts in the confusion matrix, which include true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The confusion matrix summarizes the predictions made by a classification model compared to the actual outcome.

$$\text{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+FP+TN+FN}} \tag{11}$$

- Precision and Recall in DL Security. These metrics are commonly used to evaluate a model's performance in detecting true positives (e.g., identifying actual attacks) versus false positives.
- Precision measures the proportion of correctly identified attacks among all instances that the model labeled as attacks.
- Recall measures the model's ability to identify all actual attacks in the dataset.

  These metrics are especially critical in deep learning (DL) security contexts, such as intrusion detection systems, where false positives can be disruptive, and missed detections can lead to critical vulnerabilities.

$$\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \tag{12}$$

$$\text{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \tag{13}$$

- F1-Score: The F1-Score provides a harmonic mean of precision and recall, offering a balance between the two in situations where both false positives and false negatives carry significant cost.

$$F1\text{-}Score = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (14)$$

4.7. Experiment Analysis

4.7.1. Accuracy Evaluation of PIFC-CLD

The accuracy observed in the PIFC-CLD results can be attributed to the algorithm's sensitivity to two key hyperparameters: the number of neighbors. **k_neighbors** and the perturbation constraint **(€).**

4.7.1.1. Effect of Increasing $k_{neighbors}$

The $k_{neighbors}$ the parameter determines how many samples from the same class are selected as potential candidates for crafting or tracing poison images. As this value increases, the following effects are observed.

- Initial Benefit ($(k = 20\text{--}30)$): The algorithm has sufficient local context to identify subtle discrepancies between benign and poisoned samples. Euclidean similarity can effectively detect the spatial deviations in feature space caused by poisoning.
- Accuracy Decline (k > 30): When the neighborhood size becomes too large, the similarity signal becomes diluted. This occurs because:
- Clean samples with naturally occurring intra-class variance begin to dominate the local neighborhood.
- Poisoned samples, typically a minority, become statistical outliers, reducing their influence on the cluster centroid or feature mean.

This reduces the contrast in Euclidean distances between clean and poisoned samples, making it more challenging to detect poisoning reliably.

4.7.1.2. Effect of Increasing €

The € Parameter controls the magnitude of allowed perturbations when generating poisoned samples or crafting neighborhood representations.

- **Low Epsilon** ($(\epsilon \leq 0.2)$): The algorithm operates under a tight constraint, limiting deviations from the original sample position. This preserves high feature coherence, enabling PIFC-CLD to detect poisoned samples through small yet consistent feature shifts.
- **High Epsilon** ($(\epsilon > 0.2)$): The poisoned samples become more dispersed in feature space.

Their features exhibit greater overlap with benign samples, thereby weakening the decision boundaries. As a result, the Euclidean distances between poisoned and clean samples may fall within the range of natural intra-class variation, increasing the likelihood of false negatives or missed detections. Additionally, the elevated variance may lead to false positives, as specific benign samples may appear anomalous despite being legitimate.

4.7.1.3. Interaction Between $k_{neighbors}$ and ϵ

There is a non-linear interaction between these two hyperparameters. For instance. A large $k\_neighbors$ value combined with a high (ϵ) Amplifies the problem by:

- Providing weak or irrelevant local context (too many loosely related samples), and introducing noise in feature distributions (from dispersed poisoned examples).
- This interaction significantly contributes to the decline in detection accuracy, particularly when both parameters exceed their optimal operational ranges. Accuracy performance of PIFC-CLD across varying k and ϵ configurations as show in Fig. 3.

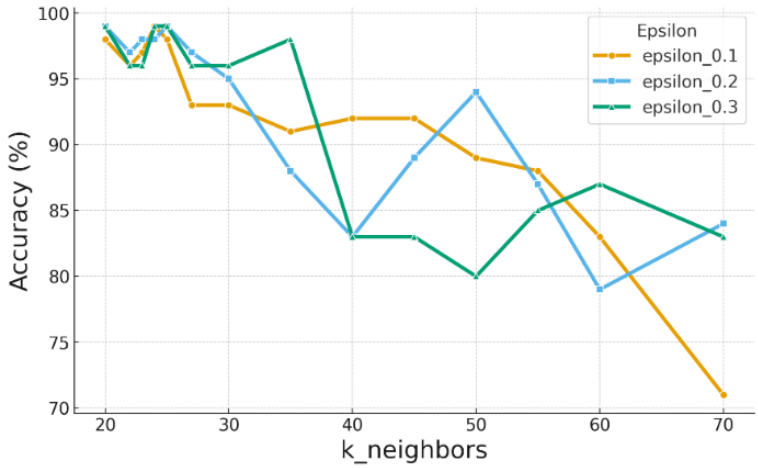

Fig. 3. Accuracy performance of PIFC-CLD across varying k and ϵ configurations

Before applying the PIFC-CLD defense, the clean test accuracy of the poisoned model under the Bullseye Polytope attack was approximately 36.7%, indicating severe performance degradation caused by successful poisoning. After identifying and removing the detected poison samples using our Euclidean-norm–based clustering defense, the retrained model achieved an accuracy of 98.2%, effectively returning to near-clean performance levels. These results demonstrate that PIFC-CLD not only detects poisoned samples with high precision but also substantially recovers the model's predictive integrity, thereby validating its practical utility for post-training forensic cleansing.

4.7.2. Evaluating Precision in PIFC-CLD

The precision evaluation results demonstrate the robustness of the proposed PIFC-CLD method for identifying poisoned samples in clean-label attacks. Across various combinations of k-nearest neighbors and perturbation bounds (ϵ), PIFC-CLD consistently achieves exceptionally high precision, with most configurations reporting values between 98% and 99% (Fig. 4). This indicates that PIFC-CLD reliably distinguishes between poisoned and clean samples without excessively flagging benign inputs.

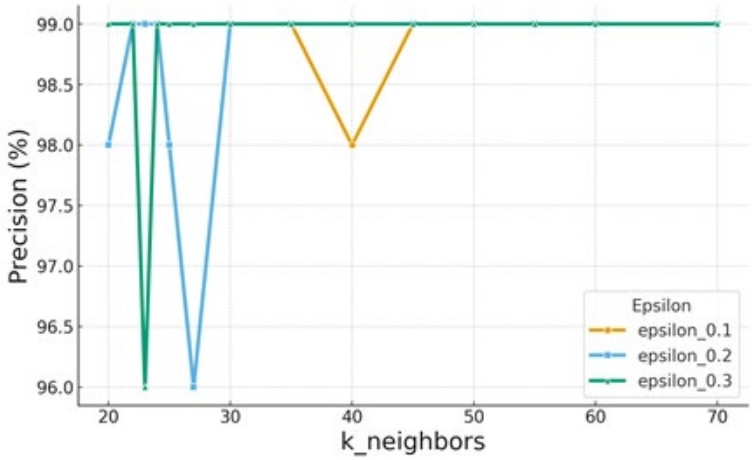

Fig. 4. Precision performance of PIFC-CLD across varying k and ϵ configurations

This level of detection precision directly addresses a notable shortcoming in alternative defense mechanisms, such as Deep k-Nearest Neighbors (Deep k-NN). Although Deep k-NN has shown some effectiveness in mitigating clean-label attacks by expanding the neighborhood size, but its precision often remains critically low, as reported in prior studies (e.g., around 20%). The root of this limitation lies in the defense's reliance on majority voting: if poisoned samples constitute a minority, the method misclassifies them as benign. Conversely, if the poisoned class is numerically dominant, the detection mechanism becomes overwhelmed, resulting in undetected poisoned data.

Moreover, Deep K-NN fails in more complex scenarios where the target object is not associated with any existing training class, such as when adversaries introduce novel or out-of-distribution classes. In such cases, poisoned samples lack sufficiently representative neighbors, rendering majority-based techniques ineffective. To compensate, Deep k-NN requires a substantial increase in k to expand its context. However, this expansion often incorporates excessive clean data from adjacent classes, inflating the number of false positives and degrading test-time performance. In contrast, PIFC-CLD utilizes geometric analysis in the learned feature space, employing Euclidean distance. Rather than relying on class-based label distributions within a fixed neighborhood, PIFC-CLD evaluates the similarity between a sample's feature representation and known clean versus poisoned clusters. This enables high-fidelity detection even when poisoned samples form tight sub-clusters or when class membership is ambiguous.

Notably, our findings corroborate prior research regarding the optimal operational window for k: when k is normalized to a value between 1 and 2 relative to class size, PIFC-CLD achieves maximal precision without overfitting or introducing false positives. Performance remains stable across this range (e.g., $k = 23\ to\ k = 35$), highlighting PIFC-CLD 's resilience and flexibility.

The PIFC-CLD method advances the state-of-the-art in forensic detection of poisoned data by providing a model-agnostic, distance-based detection approach that does not rely on class labels. This enables the method to remain effective even when labels are unreliable or manipulated. In addition, the approach maintains high precision in sparse or adversarially crafted regions of the feature space, where poisoned samples are typically designed to hide. Compared to large-k Deep k-NN configurations, PIFC-CLD also reduces false positives, leading to more reliable identification of malicious data while preserving legitimate samples.

4.7.3. Recall Performance and Forensic Robustness of PIFC-CLD

The proposed PIFC-CLD method consistently demonstrates high recall across varying neighborhood sizes. ($k\_neighbors$) and perturbation budgets (ϵ). As illustrated in Fig. 5, PIFC-CLD achieves near-perfect recall ($\geq 99\%$) for small and moderate values of $k\_neighbors$ (20–25), and across all tested (ϵ) Levels (0.1 to 0.3). This indicates that the method can successfully detect nearly all poisoned samples, particularly under realistic attack settings where the perturbation strength is not excessive.

Recall begins to decline as the neighborhood size increases beyond ($k = 30$). This degradation becomes more pronounced at ($k = 40$), particularly for ($\epsilon = 0.3$), where recall drops to 83%. This decline aligns with the forensic principle that over-generalization dilutes discriminative signals in the feature space. As the neighborhood expands, clean and poisoned data increasingly overlap in high-dimensional embeddings, making them harder to separate. Consequently, PIFC-CLD like any reactive defense must balance between granularity and generality in anomaly detection within the feature space.

This performance is in contrast to the Deep k-NN defense proposed in prior work [19], which, while effective in certain conditions, suffers from several critical limitations that PIFC-CLD addresses.

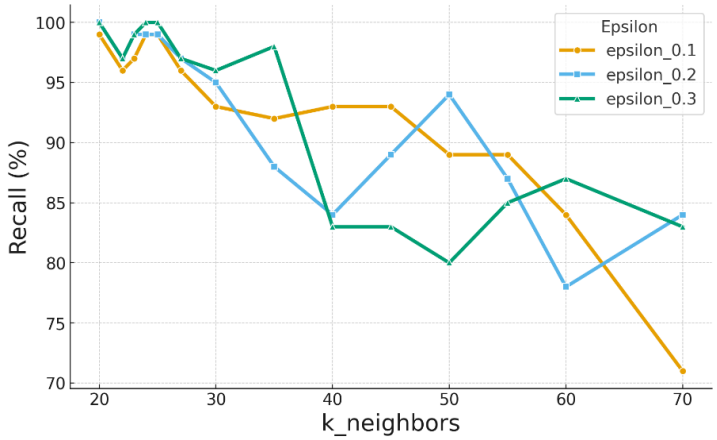

Fig. 5. Recall performance of PIFC-CLD across varying k and ϵ configurations

- Recall vs. False Positive Trade-off. Deep k-NN increases its neighborhood size to suppress poisoned majorities. However, this leads to a significant drop in precision often below 20% as it mistakenly flags many benign samples as poisoned. In contrast, PIFC-CLD maintains high precision and recall by analyzing geometric consistency via Euclidean norms rather than relying on label frequency.
- Sensitivity to Class Imbalance and Unknown Targets. Deep k-NN struggles when poison clusters are dense and compact, especially when overlapping with small target classes or out-of-distribution (OOD) samples. PIFC-CLD mitigates this by using cluster centroid distances, ensuring that numerically minor but anomalous poison groups are detected.
- Model Agnostic and Transparent Traceback. While Poison Forensics frameworks promote traceability, they often lack mechanisms for precise sample attribution under clean-label constraints. PIFC-CLD fills this gap by applying Euclidean-based clustering directly to feature representations, making the process interpretable, modular, and classifier-independent.
- Adaptability Across (ϵ) Levels. As (ϵ) Increases simulating stronger attacker capabilities, PIFC remains resilient, maintaining over 95% recall up to ($k = 35$), and gracefully degrades beyond. This highlights PIFC's practicality in real-world scenarios where poison strength varies, and robustness across (ϵ) Thresholds are essential.

4.7.4. F1 Score Evaluation of PIFC-CLD

F1 Score Evaluation of PIFC-CLD. The F1 score, a harmonic mean of precision and recall, serves as a balanced indicator of detection performance, particularly under class imbalance, a common scenario in clean-label poisoning. As shown in Fig. 6, the proposed PIFC method consistently achieves high F1 scores across various hyperparameter configurations. Specifically, for k_neighbors values between 20 and 30, and across all evaluated perturbation budgets ($\epsilon = 0.1, 0.2, 0.3$), PIFC maintains F1 scores between 97% and 99%. This confirms its robustness in accurately identifying poisoned samples without significantly compromising the integrity of clean data.

As the neighborhood size increases ($(k \geq 35)$), PIFC-CLD still retains competitive F1 performance, particularly under low and moderate perturbation budgets. Notably, at ($k = 35$) The method achieves an F1 score of 99% for ($\epsilon = 0.3$), underscoring its ability to detect adversarial perturbations even under more severe attack scenarios. However, a gradual decline is observed beyond this point: at ($k = 60$) and ($k = 70$), the F1 score drops to the 88%–90% range. This performance degradation reflects the increasing influence of false positives, caused by the blurring of cluster boundaries in high-dimensional feature spaces. As neighborhood size grows, the scope of analysis expands, unintentionally encompassing benign samples with distributions similar to those of poisoned data.

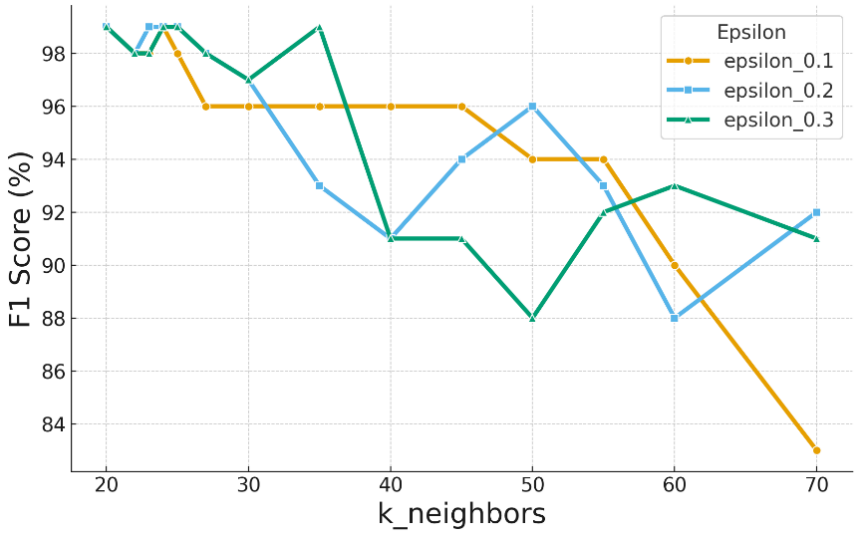

Fig. 6. F1 Score performance of PIFC-CLD across varying k and ϵ configurations

4.7.5. Comparative Limitations in Prior Work

As highlighted in Huang et al. [19], Deep -NN defenses attempt to mitigate poisoning by enlarging the neighbor set. However, this introduces a critical trade-off: small values may fail to capture tightly grouped poison clusters, while large. Values often encompass benign regions, sharply reducing precision, and consequently, the F1 score. In clean-label attack scenarios, where poisoned samples are nearly indistinguishable from clean ones in both appearance and semantics, Deep -NN models have been reported to yield precision values as low as 20%. PIFC-CLD circumvents this limitation by leveraging Euclidean feature similarity and centroid-based distance metrics, ensuring high precision and recall without the need for vast neighborhoods

PIFC maintains a strong F1 score ($\geq 91\%$) under ($\epsilon = 0.3$) across most configurations, outperforming prior defense strategies that often degrade significantly as adversarial strength increases. This demonstrates PIFC-CLD 's resilience in forensic settings where perturbation magnitudes and poisoning ratios may vary unpredictably. Its iterative, centroid-aware design supports effective detection without incurring the high false positive rates commonly associated with large-k Neighborhood-based methods.

4.8. Discussion

While various defenses, such as Deep k-NN, spectral activation clustering, and influence-function traceback, have been proposed to mitigate data poisoning, their computational demands and interpretability vary significantly. To contextualize PIFC-CLD within this landscape, Table 2 compares it with representative methods in terms of their underlying assumptions, required access, computational complexity, and traceability performance. PIFC fundamentally differs in that it is a post-hoc, model-agnostic, and computationally lightweight approach. It requires only a single pass to extract feature embeddings, followed by k-means clustering (k = 2) within the target class. Unlike gradient- or Hessian-based approaches, PIFC provides a fast and interpretable traceback mechanism to identify poisoned samples after training, without modifying the learning process.

Table 2. Comparative characteristics of PIFC and existing defense families

Defense Type	Core Assumption	Primary Signal Used	Access Required	Computational Cost	Traceability Level
PIFC-CLD	Poisons form a sub-cluster near the misclassified target in feature space	Penultimate-layer embeddings + k-means (k=2)	Embeddings only; no gradients	Low	Per-sample (index-level)
Deep k-NN [24]	Clean neighbors dominate the local space	Neighbor labels/distances	Embeddings; repeated neighbor queries	Medium	Local (neighbor set)
Spectral / Activation Clustering [29]	Global spectral shift reveals anomaly	Activation matrices/spectra	All activations	High	Cluster-level
Influence Functions [30]	Poisoned samples change loss curvature	Gradients / Hessian–vector products	Full training gradients	Very High	Per-sample
Certified [25]	Smoothing or certified radii guarantee robustness	Certified robustness metrics/diffusion	Model access + generative smoothing	Very High	Aggregate (non-traceable)

5. Conclusion

This paper presented PIFC-CLD , a forensic detection algorithm for identifying clean-label poisoned data that causes misclassification in deep neural networks. Through geometric analysis in feature space, PIFC efficiently traces the source of the misclassification by clustering candidate training samples and

measuring their Euclidean distances from the misclassified target's feature representation. Compared to traditional defenses such as Deep k-NN, PIFC-CLD achieves superior performance in both precision and recall. Although Deep k-NN can be robust under specific configurations, it offen suffers from low detection precision and an increased number of false positives as the neighborhood size (k) grows. PIFC-CLD addresses these limitations through centroid-based feature analysis, eliminating reliance on class labels or majority voting schemes. Our experimental evaluation on CIFAR-10 under Bullseye Polytope attacks demonstrates that PIFC achieves up to 99% precision and 96% F1 scores across multiple settings, exhibiting robustness to variations in poisoning intensity and hyperparameters, such as k_neighbors and epsilon. Notably, PIFC-CLD remains effective without the computational overhead associated with backpropagation tracking or meta-learning strategies. In summary, PIFC-CLD provides a robust foundation for post-hoc forensic analysis in DNNs, enabling secure AI deployments in adversarial environments. Future work includes extending this method to federated learning systems, integrating it with trigger visualization techniques, and developing theoretical guarantees for detection bounds. Although the current evaluation was limited to WideResNet and the CIFAR-10 dataset for benchmarking, subsequent work will explore PIFC's scalability on larger, more complex datasets such as ImageNet and CelebA. In addition, we plan to evaluate its performance across a wider range of architectures, including ResNet variants, DenseNets, and Vision Transformers, as well as in distributed settings like federated learning. These future directions will further demonstrate the generality of PIFC-CLD and establish its robustness across diverse models and data regimes beyond the scope of this initial study.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. None of the authors has received any funding or grants from any institution or funding body for this research.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] N. Z. Khalaf, I. I. Al Barazanchi, A. D. Radhi, S. Parihar, P. Shah, and R. Sekhar, "Development of real-time threat detection systems with AI-driven cybersecurity in critical infrastructure," *Mesopotamian J. CyberSecurity*, vol. 5, no. 2, pp. 501–513, Jun. 2025. [Online]. Available at: https://journals.mesopotamian.press/index.php/CyberSecurity/article/view/828.

[2] M. R. Subhi, S. Yussof, L. A. B. Burhanuddin, and F. L. Khaleel, "CNNs in Image Forensics: A Systematic Literature Review of Copy-Move, Splicing, Noise Detection, and Data Poisoning Detection Methods," *Mesopotamian J. CyberSecurity*, vol. 5, no. 2, pp. 636–656, Jul. 2025. [Online]. Available at: https://mesopotamian.press/journals/index.php/CyberSecurity/article/view/845.

[3] M. Alanezi and R. M. A. AL-Azzawi, "AI-Powered Cyber Threats: A Systematic Review," *Mesopotamian J. CyberSecurity*, vol. 4, no. 3, pp. 166–188, Dec. 2024, doi: 10.58496/MJCS/2024/021.

[4] A. Abomakhelb, K. A. Jalil, A. G. Buja, A. Alhammadi, and A. M. Alenezi, "A Comprehensive Review of Adversarial Attacks and Defense Strategies in Deep Neural Networks," *Technologies*, vol. 13, no. 5, p. 202, May 2025, doi: 10.3390/technologies13050202.

[5] T. T. Nguyen *et al.*, "Manipulating Recommender Systems: A Survey of Poisoning Attacks and Countermeasures," *ACM Comput. Surv.*, vol. 57, no. 1, pp. 1–39, Jan. 2025, doi: 10.1145/3677328.

[6] A. Shafahi *et al.*, "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks," in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2018, pp. 6103–6113. [Online]. Available at: https://proceedings.neurips.cc/paper_files/paper/2018/hash/.

[7] A. Turner MIT, D. Tsipras MIT, and A. Mądry MIT, "Clean-Label Backdoor Attacks," Massachusetts Institute of Technology, 2019. [Online]. Available at: https://share.google/dnYZ1kGfkNzvrYezB.

[8] R. Bommasani *et al.*, "On the Opportunities and Risks of Foundation Models," *arxiv Mach. Learn.*, pp. 1–214, 2021, [Online]. Available at: https://arxiv.org/abs/2108.07258.

[9] B. Zhao and Y. Lao, "CLPA: Clean-Label Poisoning Availability Attacks Using Generative Adversarial Nets," *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 8, pp. 9162–9170, Jun. 2022, doi: 10.1609/aaai.v36i8.20902.

[10] A. Gupta and A. Krishna, "Adversarial Clean Label Backdoor Attacks and Defenses on Text Classification Systems," in *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, 2023, pp. 1–12, doi: 10.18653/v1/2023.repl4nlp-1.1.

[11] H. L. Xinyuan, S. Joshi, T. Thebaud, J. Villalba, N. Dehak, and S. Khudanpur, "Clean Label Attacks against SLU Systems," *arxiv Artif. Intell.*, pp. 1–8, Sep. 2024. [Online]. Available at: https://arxiv.org/pdf/2409.08985v1.

[12] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden Trigger Backdoor Attacks," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 07, pp. 11957–11965, Apr. 2020, doi: 10.1609/aaai.v34i07.6871.

[13] G. Severi, J. Meyer, S. Coull, and A. Oprea, "Explanation-Guided Backdoor Poisoning Attacks Against Malware Classifiers," *in 30th USENIX Security Symposium, 2021,* pp. 1487–1504. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/severi.

[14] S. Zhao, X. Xu, L. Xiao, J. Wen, and L. A. Tuan, "Clean-label backdoor attack and defense: An examination of language model vulnerability," *Expert Syst. Appl.*, vol. 265, p. 125856, Mar. 2025, doi: 10.1016/j.eswa.2024.125856.

[15] H. I. Kure, P. Sarkar, A. B. Ndanusa, and A. O. Nwajana, "Detecting and Preventing Data Poisoning Attacks on AI Models," *arxiv Artif. Intell.*, pp. 4–8, Mar. 2025,. [Online]. Available at: https://arxiv.org/pdf/2503.09302.

[16] M. A. Hanif, N. Chattopadhyay, B. Ouni, and M. Shafique, "Survey on Backdoor Attacks on Deep Learning: Current Trends, Categorization, Applications, Research Challenges, and Future Prospects," *IEEE Access*, vol. 13, pp. 93190–93221, 2025, doi: 10.1109/ACCESS.2025.3571995.

[17] Y. Zeng, M. Pan, H. A. Just, L. Lyu, M. Qiu, and R. Jia, "Narcissus: A Practical Clean-Label Backdoor Attack with Limited Information," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, Nov. 2023, pp. 771–785, doi: 10.1145/3576915.3616617.

[18] O. Mengara, A. Avila, and T. H. Falk, "Backdoor Attacks to Deep Neural Networks: A Survey of the Literature, Challenges, and Future Research Directions," *IEEE Access*, vol. 12, pp. 29004–29023, 2024, doi: 10.1109/ACCESS.2024.3355816.

[19] B. Nelson *et al.*, "Misleading Learners: Co-opting Your Spam Filter," in *Machine Learning in Cyber Trust*, Boston, MA: Springer US, 2009, pp. 17–51, doi: 10.1007/978-0-387-88735-7_2.

[20] H. Aghakhani, D. Meng, Y.-X. Wang, C. Kruegel, and G. Vigna, "Bullseye Polytope: A Scalable Clean-Label Poisoning Attack with Improved Transferability," in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, Sep. 2021, pp. 159–178, doi: 10.1109/EuroSP51992.2021.00021.

[21] C. Zhu *et al.*, "Transferable Clean-Label Poisoning Attacks on Deep Neural Nets," in *Proceedings of Machine Learning Research*, May 2019, pp. 7614–7623. [Online]. Available at: https://proceedings.mlr.press/v97/zhu19a.html.

[22] J. Geiping *et al.*, "Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching," *ICLR 2021 - 9th Int. Conf. Learn. Represent.*, pp. 1–24, Sep. 2020. [Online]. Available at: https://arxiv.org/pdf/2009.02276.

[23] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: a defence against trojan attacks on deep neural networks," in *Proceedings of the 35th Annual Computer Security Applications Conference*, Dec. 2019, pp. 113–125, doi: 10.1145/3359789.3359790.

[24] N. Peri *et al.*, "Deep k-NN Defense Against Clean-Label Data Poisoning Attacks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12535 LNCS, Springer, Cham, 2020, pp. 55–70, doi: 10.1007/978-3-030-66415-2_4.

[25] S. Hong, N. Carlini, and A. Kurakin, "Diffusion Denoising as a Certified Defense Against Clean-Label Poisoning Attacks," in *ICLR 2024 Conference*, 2024, pp. 1–12. [Online]. Available at: https://openreview.net/forum?id=aAE44ivBtx.

[26] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, "Backdoor Defense via Decoupling the Training Process," *ICLR 2022 - 10th Int. Conf. Learn. Represent.*, pp. 1–25, Feb. 2022. [Online]. Available at: https://arxiv.org/pdf/2202.03423.

[27] M. Zolotukhin, D. Zhang, T. Hämäläinen, and P. Miraghaei, "On Attacking Future 5G Networks with Adversarial Examples: Survey," *Network*, vol. 3, no. 1, pp. 39–90, Dec. 2022, doi: 10.3390/network3010003.

[28] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," pp. 1-60, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[29] B. Chen *et al.*, "Detecting backdoor attacks on deep neural networks by activation clustering," *CEUR Workshop Proc.*, vol. 2301, pp. 1–8, 2019, [Online]. Available at: https://ceur-ws.org/Vol-2301/paper_18.pdf.

[30] G. Cohen, G. Sapiro, and R. Giryes, "Detecting Adversarial Samples Using Influence Functions and Nearest Neighbors," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 14441–14450, doi: 10.1109/CVPR42600.2020.01446.