BACKDOOR DETECTION VIA DENSITY BASED DYNAMIC CLUSTERING

Ms. K.R. [1]Nandhashree, U [2]Jashwanth, J [3]Kalesh, S R [4]Sai Arjun

Assistant Professor[1], UG Scholar[2,3,4], Department of Cyber Security,

SRM Valliammai Engineering College

 DOA:15.05.2025 / DOP: 20.05.2025 ISSN: 3048-619X
Volume 2 Issue 2 [Apr – July 2025] DOI:

ABSTRACT

We introduce Density-based Dynamic Clustering and Centroid Analysis for Universal Defense (DCCAUD) against backdoor attacks. The suggested protection looks at the features of the training dataset in order to identify the existence of backdoor attacks in deep neural network models. In order to provide adaptive clustering that adapts to changing data distributions, DCCA-UD starts by dynamically grouping samples in the training set based on their density. Subsequently, it employs an innovative strategy to identify poisoned clusters by examining the misclassification behavior induced when features from a representative cluster example are integrated with benign samples. This approach remains attack-agnostic, distinguishing it from existing defenses that may only target specific types of backdoor attacks or rely on certain poisoning conditions. Our experiments, conducted across diverse classification tasks and network architectures, encompass various backdoor attack types with both clean and corrupted labels, and a range of triggering signals including global, local, sample-specific, and source-specific triggers. Results demonstrate the effectiveness of DCCA-UD in defending against backdoor attacks, consistently surpassing state-of-the-art techniques across all scenarios. Through this project, we present a robust defense mechanism capable of safeguarding deep neural network models against a broad spectrum of backdoor threats.

INTRODUCTION

Because of their exceptional performance, Deep Neural Networks (DNNs) have become indispensable tools in a wide range of disciplines, particularly in image classification, Natural Language Processing (NLP), and Pattern Recognition. However, the widespread adoption of DNNs has also brought forth new security challenges, notably their vulnerability to backdoor attacks. In these attacks, malicious actors manipulate the training dataset, introducing subtle alterations that enable deceptive behavior within the model. The integrity of DNNs can be compromised when a trained dataset is tainted with poisoned samples, leading to erroneous predictions or malicious behavior under specific conditions.

Maintaining the reliability of DNN models requires the detection of such poisoned data, especially in safety-critical applications like financial systems, autonomous cars, and medical diagnosis. In this study, we employ centroid analysis and clustering to tackle the urgent problem of backdoor assaults on DNNs. Our goal is to identify any poisoned data that may be present in the training dataset by using centroids analysis and clustering techniques. By offering a proactive protection mechanism against backdoor assaults, our method allows for timely mitigation techniques and provides insights into the integrity of DNN models.

Through rigorous experimentation and evaluation, we demonstrate the effectiveness and robustness of our proposed method across diverse datasets, network architectures, and attack scenarios. By enhancing the security posture of DNNs, our work contributes to the advancement of trustworthy AI systems, fostering confidence in their deployment across critical domains.

BACKDOOR ATTACK

In the context of machine learning and cybersecurity, a backdoor is a secret opening or harmful feature that is purposefully added to a system or model to permit unwanted access or control. In the realm of machine learning, a backdoor attack involves the manipulation of training data or model parameters to embed a hidden trigger or pattern. This trigger can subsequently be exploited to manipulate the model's behavior in unintended ways when presented with specific inputs.

RELATED WORKS

At the training dataset level, defense strategies now in use usually rely on dynamic clustering and feature representation or activation pattern analysis. Activation Clustering (AC) [10] is one such technique that clusters samples using the K-means algorithm based on feature representation and is focused on corrupted label attacks. The efficiency of AC declines with low poisoning ratios, but it may still discern whether a class is poisoned or not by examining the relative sizes of clusters.

Xiang et al. introduced the Cluster Impurity (CI) method [13], which utilizes Gaussian Mixture Model (GMM) clustering to identify poisoned samples. CI works against corrupted-label attacks by filtering samples to remove triggering signals. However, its applicability is limited to high-frequency triggers.

Tang et al. proposed the Statistical Contamination Analyzer (Scan) [11], which decomposes image/object representations using the Expectation Maximization (EM) algorithm. SCAn detects contaminated classes based on intra-variation and splits representations using Linear Discriminant Analysis (LDA). However, SCAn fails against sample-specific attacks.

To address SCAn's limitation, [14] introduced Beatrix for anomaly detection using Gram matrix statistics. However, Beatrix suffers from the curse of dimensionality with high-dimensional feature spaces. Other defense methods, although available, often rely on unrealistic assumptions regarding the knowledge of the backdoor attack. For example, in order to discover anomalies, [7] and [12] use singular value decomposition (SVD), which requires prior knowledge of the maximum number of poisoned samples.

A trackback tool that requires the defender to identify at least one poisoned sample during testing was introduced by Shan et al. [5]. Some defenses target specific types of attacks, like [2] for clean-label

backdoor attacks or methodologies focusing on training backdoor-free models [3, 4, 7]. Finally, [15] uses the structural relationships between shallow and deep layers to purify models that have been backdoored using self-attention.

DENSITY BASED DYNAMIC CLUSTERING

Density-based dynamic clustering examines the distribution of feature representations in the training dataset to identify potential backdoor attacks in Deep Neural Networks (DNNs). Backdoor attacks can jeopardize the CNN model's performance and integrity by introducing tainted data into the training set.

1.Feature Representation Analysis

In DNNs, each data point is transformed into a high-dimensional feature representation through the network's layers. By clustering these feature representations using density-based methods, we can identify clusters of similar samples within the training dataset.

2.Detecting Anomalies

Backdoor attacks often introduce anomalies or deviations in the data distribution, which may manifest as clusters of poisoned samples with distinct characteristics. Density-based clustering can help identify these anomalous clusters by detecting regions of low density surrounded by regions of high density. These low-density regions may indicate the presence of poisoned data that deviates from the underlying distribution of benign samples.

3.Adaptive Parameter Adjustment

Based on the local density of the data, density-based dynamic clustering modifies its parameters, including the minimum number of points and epsilon neighborhood. This adaptability is crucial for effectively detecting backdoor attacks, as the characteristics of poisoned data clusters may vary in different parts of the feature space. By dynamically adjusting clustering parameters, the algorithm can effectively capture both dense and sparse regions of the data distribution, enhancing its ability to detect anomalous clusters associated with backdoor attacks.

4.Robustness to Varying Data Distributions

Backdoor attacks can manipulate the data distribution in subtle ways, making it challenging to detect poisoned samples using fixed clustering parameters. Density-based dynamic clustering addresses this challenge by adaptively adjusting parameters to accommodate varying data densities and complex structures within the training dataset. This robustness allows the algorithm to effectively detect backdoor attacks across different datasets and network architectures.

Overall, density-based dynamic clustering provides a powerful framework for detecting backdoor attacks in DNNs by analyzing the distribution of feature representations in the training dataset. By leveraging the adaptability and robustness of density-based clustering techniques, we can enhance the resilience of DNN models against backdoor attacks and ensure their integrity in real-world applications.

ARCHITECTURE DIAGRAM

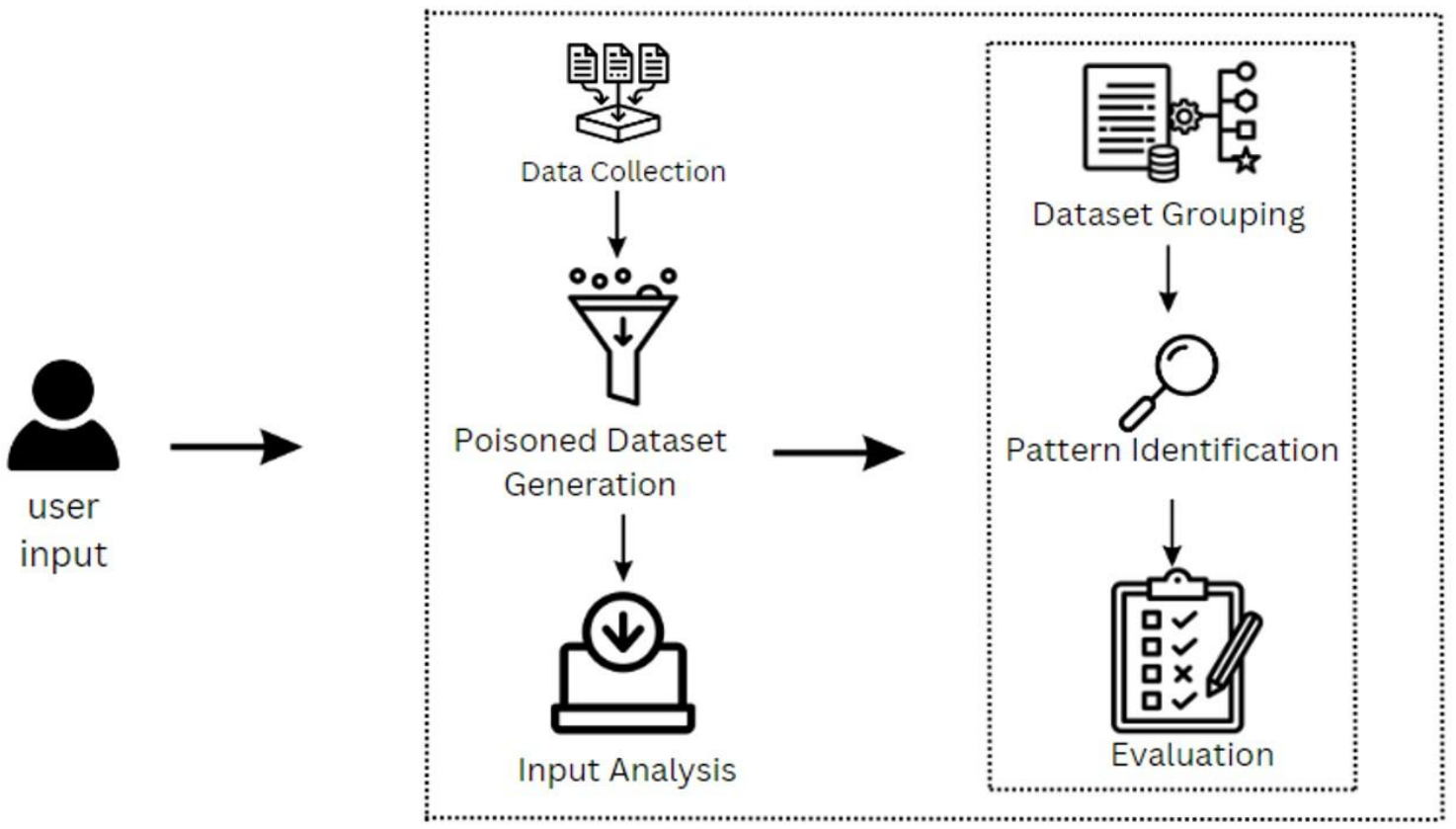

Figure 1: Backdoor detection

An all-encompassing strategy to protect Deep Neural Networks (DNNs) from backdoor attacks is shown in the architecture diagram. The system begins by gathering user input, which is then passed to the dataset poisoning mechanism. Here, the user data is combined with the pre-trained dataset to create a poisoned dataset, introducing subtle alterations that mimic potential backdoor attacks. Subsequently, the poisoned dataset is utilized to train the detection system, enabling it to identify and classify instances of poisoned attacks effectively. After training, the detection system is put into action to protect DNN models from backdoor attacks. This proactive defense ensures the integrity and dependability of AI systems in a variety of domains and applications.

MODULES

CREATION OF DATASET

Ten categories and sixty thousand color images, each measuring thirty-two pixels, make up the CIFAR-10 dataset. These categories include a wide range of everyday objects, including cars, animals,

and landscapes. The dataset, which is divided into 10,000 test photos and 50,000 training images, allows for a complete evaluation of machine learning models.

Each image is represented in RGB format, with three color channels (red, green, and blue), allowing for a diverse range of visual features to be captured. This dataset's relatively low resolution makes it computationally tractable for training and experimenting with machine learning models while still presenting significant challenges due to its real-world complexity.

How to Use the Module,

- Installation

Ensure that you have Python installed on your system. You can download the module cifar10_dataset.py from our repository or create it with the provided code snippet.

- Importing the Module

Once you have the module, you can import it into your Python script or Jupyter Notebook

- Accessing Training Data

You can retrieve a batch of training data along with their corresponding labels using the get_train_batch method

- Visualizing CIFAR-10 Images

You can visualize CIFAR-10 images using libraries like Matplotlib.

IMPLEMENTATION OF TRAINING MODEL

This module is to mainly focus the initialization phase in the training process of CIFAR-10 involves setting up the foundational components required for training a convolutional neural network (CNN) model. This phase encompasses initializing the model architecture, optimizer, and loss function, as well as defining additional parameters essential for the training process.

Once the model architecture is defined, train the model on the CIFAR-10 training set. The training process typically involves the following steps:

- Initialization:

Describe the CNN model's architecture, which will be used to train it on the CIFAR-10 dataset. This entails defining the quantity and arrangement of fully connected, pooling, and convolutional layers.

Model architecture selection has a big impact on how well the model performs on the classification task and how well it can learn from the data.

Choose appropriate activation functions for the hidden layers of the model.

Rectified Linear Unit (ReLU) is a popular option because of its ease of use and efficiency in addressing the vanishing gradient issue.

- Training Loop:

The training process entails batching the training data, feeding them into the model, calculating the loss, and adjusting the model parameters through backpropagation. Over a predetermined number of epochs, this iterative process is carried out, allowing the model to improve its performance as it gains experience with the training set of data.

- Hyperparameter Tuning:

To maximize the model's performance on the CIFAR10 dataset, hyperparameter tuning entails experimenting with various hyperparameters, including the learning rate, batch size, optimizer selection, and model architecture.

EVALUATION AND VALIDATION

Assess the model's accuracy and generalization abilities by running it through the CIFAR-10 test set after it has been trained. To gauge the trained CNN model's effectiveness and capacity for generalization, test it on a different validation set. Track metrics like recall, accuracy, precision, and F1-score to determine how well the model performs in relation to certain parameters.

COMPARISON AND PREDICTION

Evaluate the model's predictions on sample images from the test set to understand its strengths and weaknesses. To predict with a trained CNN model, execute a forward pass by inputting data (e.g., images) into the model. During this pass, the input data traverses through the network layers, and the model calculates the predicted output (e.g., class probabilities) for each input sample. To generate forecasts that are meaningful, interpret the model's output. The output for classification tasks usually consists of class probabilities produced by the output layer's SoftMax activation function. For the input sample, the class with the highest probability is regarded as the predicted class.

FUTURE ENHANCEMENT

It could include integrating advanced anomaly detection techniques and leveraging deep learning models for improved accuracy and adaptability. Additionally, incorporating real-time monitoring capabilities and enhancing the methodology to handle dynamic and evolving attack scenarios would be beneficial. Integration with threat intelligence feeds and automated response mechanisms could further strengthen the system's resilience against sophisticated backdoor attacks. Moreover, exploring ensemble methods that combine multiple clustering algorithms or integrating explainable AI techniques to provide insights into detected anomalies could enhance the interpretability and effectiveness of the detection system, ensuring robust protection against emerging threats in increasingly complex cybersecurity landscapes.

CONCLUSION

Finally, in order to protect Deep Neural Networks (DNNs) from backdoor attacks, our study presents Density-based Dynamic Clustering and Centroid Analysis for Universal Defense (DCCA-UD). We tackled the crucial problem of identifying tainted data in the training dataset, which can jeopardize the accuracy and efficiency of DNN models.

Overall, our project contributes to advancing the field of adversarial machine learning by providing a reliable and attack-agnostic defense mechanism against backdoor attacks in DNNs. By enhancing the security posture of DNN models, we aim to foster trust and confidence in their deployment across critical domains, safeguarding against malicious manipulation of training data and ensuring the integrity of AI systems in real world applications.

REFERENCES

[1] W. Jiang, X. Wen, J. Zhan, X. Wang, Z. Song and C. Bian, ‘Critical Path-Based Backdoor Detection for Deep Neural Networks,’ in IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 3, pp. 4032-4046, March 2024.

[2] Z. Yao et al., ‘Reverse Backdoor Distillation: Towards Online Backdoor Attack Detection for Deep Neural Network Models,’ in IEEE Transactions on Dependable and Secure Computing, vol. 70, no. 12, pp. 12102-12114, 2024.

[3] Y. Li et al., ‘NTD: Non-Transferability Enabled Deep Learning Backdoor Detection,’ in IEEE Transactions on Information Forensics and Security, vol. 19, pp. 104-119, 2024.

[4] Z. Chen, S. Yu, M. Fan, X. Liu and R. H. Deng, 'Privacy-Enhancing and Robust Backdoor Defense for Federated Learning on Heterogeneous Data,' in IEEE Transactions on Information Forensics and Security, vol. 19, pp. 693-707, 2024.

[5] H. Qiu et al., 'Toward a Critical Evaluation of Robustness for Deep Learning Backdoor Countermeasures,' in IEEE Transactions on Information Forensics and Security, vol. 19, pp. 455-468, 2024.

[6] X. Zhao, H. Wu and X. Zhang, 'Effective Backdoor Attack on Graph Neural Networks in Spectral Domain,' in IEEE Internet of Things Journal, vol. 11, no. 7, pp. 12102-12114, 2024.

[7] L. A. C. Ahakonye, C. I. Nwakanma, J. M. Lee and D. -S. Kim, 'X-HDNN: Explainable Hybrid DNN for Industrial Internet of Things Backdoor Attack Detection,' in IEEE Transactions on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2023, pp. 10531057,2023.

[8] W. Guo, B. Tondi and M. Barni, 'A Temporal Chrominance Trigger for Clean-Label Backdoor Attack Against Anti-Spoof Rebroadcast Detection,' in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 6, pp. 4752-4762, Nov.-Dec. 2023.

[9] T. Qiu and Y. -J. Li, 'Fast LDP-MST: An Efficient Density-Peak-Based Clustering Method for LargeSize Datasets,' in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 5, pp. 4767-4780, May 2023.

[10] Z. Xiang, D. J. Miller, and G. Kesidis, 'Detection of Backdoors in Trained Classifiers Without Access to the Training Set,' in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 3, pp. 1177-1191, March 2022.

[11] M. Fan, Z. Si, X. Xie, Y. Liu, and T. Liu, 'Text Backdoor Detection Using an Interpretable RNN Abstract Model,' in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 4117-4132, 2021.

[12] Debnath and M. Song, 'Fast Optimal Circular Clustering and Applications on Round Genomes,' in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 18, no. 6, pp. 2061-2071, 1 Nov.-Dec. 2021.

[13] G. Sun, R. He, B. Ai, C. Huang and Z. Zhong, 'Dynamic Clustering of Multipath Components for Time-Varying Propagation Channels,' in IEEE Transactions on Vehicular Technology, vol. 70, no. 12, pp. 13396-13400, Dec. 2021.

[14] J. Wang, C. Zhu, Y. Zhou, X. Zhu, Y. Wang and W. Zhang, 'From Partition-Based Clustering to Density-Based Clustering: Fast Find Clusters With Diverse Shapes and Densities in Spatial Databases,' in IEEE Access, vol. 6, pp. 1718-1729, 2021.

[15] K Elaiyaraja, MS Kumar, B Chidhambararajan," Enhanced Effective Generative Adversarial Networks Based LRSD and SP Learned Dictionaries with Amplifying CS" - Machine Learning and IoT for Intelligent Systems, 2021.