IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 8, August 2023

DOI: 10.17148/IJARCCE.2023.12815

Investigation of Layer-wise Feature Analysis for Backdoor Attacks Detection in Deep Neural Networks

Preetha S[1], Nalini M K[2], Mahalakshmi B S[3], Anushka R Dongal[4], Vineetha K[5]

Assistant Professor, ISE, B.M.S. College of Engineering, Bengaluru, India[1,2,3]

Student, ISE, B.M.S. College of Engineering, Bengaluru, India[4,5]

Abstract: Data is the driving force behind the power of modern-day Machine Learning or Deep Learning algorithms. Accuracy and efficiency of these algorithms are largely dependent on the quality of the data that they are trained on; consequently, data poisoning poses a significant threat to these models. Data poisoning attacks present a significant challenge in maintaining the integrity of machine learning models. Currently automated methods and human inspection techniques often fail to identify clean subsets with high precision. In this paper target class of samples for this study's layer-wise feature analysis includes both poisoned and benign samples. It discovers that the key layer-which is frequently overlooked by existing defences is what distinguishes dangerous from innocuous substances. Key layer analysis of characteristic differences between suspicious and benign samples suggests a simple yet effective approach to filter poisoned samples. Effectiveness of the defences has been verified by in-depth experiments on two benchmark datasets.

Keywords: Data poison, machine learning, Deep Neural Networks, Feature Analysis.

I. INTRODUCTION

Data poisoning causes poor performance in security issues and inaccurate predictions in machine learning. Deep learning models are malicious alteration of training data. Monitoring datasets using anomaly classification algorithms and putting in place rigorous training protocols are just a few methods for detecting data poisoning. Introduction of adversarial instances which are constructed inputs intended to deceive the model is one kind of data poisoning. Due to large dimension and non-linear nature of models, detecting data poisoning in deep learning could be difficult. Furthermore, to strengthen the model's resistance against poisoned assaults, adversarial training techniques may be used. A thorough understanding of issue domain and future attack vectors are essential to detect threats properly.

Although Deep Neural Networks (DNN) have been successfully applied to a variety of tasks, training them requires a significant amount of computing power and training data. A small number of training samples can be contaminated by backdoor attacks on DNNs, allowing it to erroneously categorize data with predefined trigger patterns into a target class chosen by adversary. In backdoor detection defenders attempt to ascertain if a suspicious object (model or sample) is malevolent and is one of the most important defensive paradigms. Assuming that poisoned samples have different feature representations than benign samples, majority of backdoor detectors now in use often focus on layer that comes before completely connected layers.

In order to classify images, this research focuses on backdoor attacks and countermeasures. Initial backdoor assault called BadNets picked a few innocent samples at random stamped a trigger patch onto their pictures and then changed their label to target label for creating poisoned copies of those samples. In Clean-label assault paradigm, label attached to the target is congruent to ground-truth labelling of poisoned samples, were offered as a way to get around the issue. Attacks targeting certain samples were added and image warping was chosen as the backdoor trigger. The three primary types of backdoor defences now in use are Input filtering, Input pre-processing, and Model mending.

Input pre-processing seeks to lower the incidence of false positives whereas input filtering seeks to distinguish between benign and poisonous samples according to their specific behaviours. The goal of model repair is to lower the incidence of false positives. This study concentrates on input filtering a practical method for securing deployed DNNs. In order to enhance spectral signature of poisonous samples, a robust covariance estimate of feature representations are offered. It also suggests filtering inputs in response to knowledge that poisoned photos frequently contain certain high-frequency artefacts. Before submitting each input sample to the deployed DNN input pre-processing alters it to obstruct possible trigger patterns and avoid backdoor activation. By adjusting DNNs using benign data, model mending tries to close

DOI: 10.17148/IJARCCE.2023.12815

backdoors in attacked DNNs. Backdoors can be removed by model pruning and competitive model unlearning has been proposed as a way to fix damaged models. Input filtering are used to safeguard deployed DNNs is the main concern of this study.

Feature Analysis by Layers

With an emphasis on identifying poisoned samples, DNNs are examined which are utilized as C- classifiers. Each layer in DNN architecture has weights, biases and functions for activation. To derive class probabilities, softmax is used to calculate DNN output. Observation illustrates attacked DNNs, leveraging class-relevant features for benign data and trigger-related features for poisoned samples predict target label for both benign and poisoned samples. Variations are investigated through layer-wise analysis.

II. RELATED WORK

Model verification with Convolutional Neural Network and Word Embedding's (MOVCE) was created to combat data poisoning attacks on deep learning vision systems. Advances in cancer detection and auto production have been made possible by deep learning systems, but their improved capacity for learning also leaves them more open to such attacks [1]. Deep neural networks may be exposed to malicious assaults [2] that introduce false information while they are being trained. Algorithm suggested in this study compares parameter distribution calculated using maximum entropy and variational inference to determine network attack. As an illustration, it employs a CNN model from MNIST dataset. Deep learning is an artificial intelligence technique that uses a structured layer of algorithms to process data and produce categorization abilities. Despite its success deep learning is susceptible to security risks such as poisoning attacks. Further study can be done on data poisoning techniques for deep learning, as well as on the synergistic effects of both adversarial examples and data poisoning [3].

How data poisoning can be used to target federated machine learning was first investigated in [4]. A bi-level data poisoning attack formulation that includes three distinct attack types was provided. Future studies of federated transfer learning and vertical (feature-based) learning federation examined data poisoning attacks. Distributed machine learning [5] (DML) can train on enormous datasets when none of the nodes are able to quickly produce reliable findings. As proportion of remote employees rises, security flaws appear making attackers more hazardous. Ana Lucila Sandoval Orozco altered the training results and polluted the dataset in advance of publication. An approach for identifying harmful data [6] makes use of context-specific knowledge about the origin and transformation of data points in the training set. This was the first way to locate causal attacks that include provenance information. The process was presented in two distinct iterations, one intended for data sets that may be partly trusted and the other for untrusted data. This is the first strategy for a defence that makes use of data provenance. They now assume the objectivity of data sources. A Python toolkit called BackdoorBox is created for backdoor learning. It offers a complete set of functions and tools to make it easier to design and analyse backdoor attacks in machine learning models. A deeper knowledge of potential security threats in machine learning systems is made possible by the use of BackdoorBox. It enabled researchers and practitioners to quickly experiment and investigate vulnerabilities and defence mechanisms associated with backdoor attacks [7]. Ahmed et al. [8] examined the dangers and types of data poisoning assaults that pose a threat to development of artificial intelligence and smart technologies. Study explored how such attacks are susceptible and suggested ways for defence and detection. Authors concluded that techniques for detecting anomalies in system behavior are the most effective means of detecting attacks. Recent innovations like the Wasserstein GAN and StyleGAN show potential development in the future.

An effective statistical defence against backdoor assaults on machine learning systems was done in [9]. Models are trained to behave maliciously in backdoor assaults under specified circumstances. The suggested defensive strategy concentrates on identifying and reducing the effects of such attacks by utilising powerful statistical tools. These methods assist in locating and eliminating the influence of tainted training data, allowing the model to operate dependably in the presence of backdoor triggers. The article includes experimental findings that show how the suggested defence mechanism works to strengthen the resistance of machine learning models to backdoor attacks. Case study in [10] presented a black-box attack and assessed its outcomes using realm of medicine. It was observed that mean squared error (MSE) of the regressor increases by 150 percent with just 2% of poisoned samples. Better protection was suggested for nonlinear regression learners like kernel SVR and kernel regression. Regression learning is being used more frequently in mission-critical systems including drug research, financial forecasting and predictive analysis for hedge funds, predictive maintenance and quality assurance. The idea of physical backdoor attacks, where real-world things are altered to trick machine learning models was explored in [11]. These assaults entail changing or adding particular patterns to objects which were noticed by sensors caused the models to behave maliciously. Study addressed all possible dangers and difficulties posed by these attacks including their covert nature and difficulty in being discovered. In order to prevent backdoor attacks on physical

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 8, August 2023
DOI: 10.17148/IJARCCE.2023.12815

systems, it emphasised the necessity for strong defences and counters urging additional study and development cybersecurity field. Machine learning could be a security system's weakest link, according to researchers at the University of Bristol [12]. Security of Drebin, an Android malware detector was examined through definition and execution of related evasion attempts. Machine learning has been widely used in the last 10 years for security-related jobs to combat increasing complexity of contemporary assaults. Machine learning approaches weren't initially intended to deal with adaptive attacks. Study aimed to demonstrate how an antagonist-aware approach to machine learning might enhance system safety.

Data poisoning attacks aim to compromise security and functionality of machine learning models [13]. Data poisoning assaults come in a variety of forms including label Model update attacks, feature injection attacks, and flipping assaults all have various effects. To identify and eliminate malicious data samples or strengthen the models, researchers have suggested a number of protection measures. Machine learning systems that have been trained using user-provided data are susceptible to data poisoning attacks. Worst-case loss of defence against a determined attacker is still not fully known, despite recent work proposing a number of attacks and defensive tactics. Empirical findings demonstrate that both Modified National Institute of Standards and Technology database (MNIST-1-7) and Dogfish datasets are resistant to attack even when simple defences are used. Internet Movie Database (IMDB) sentiment dataset [14] can experience a test error increase of up to 23% with just 3% poisoned data. However, most important component in the field of machine learning namely training data originates directly from outside sources.

According to a new survey of industry experts' data poisoning is the most annoying threat to machine learning models. Other threats include model theft and adversarial attacks. A benchmarking and assessment framework for various poison attacks on image classifiers was developed in [15]. Attacks that alter training data to change the behavior of the system are increasingly sophisticated. Meta Poison is a first-order method for approximating bi-level optimization using meta-learning. It has been demonstrated to be more efficient than currently used clean-label poisoning techniques. Meta Poison can also be used for degenerate purposes such as mislabelling one class as a different one of your choice. Study raised awareness of this important threat vector and establishes a benchmark for future data poisoning research [16].

Data poisoning attacks can affect machine learning systems employed in high-stakes situations like autonomous driving, biometrics and cybersecurity. Attacks like these take place when a tiny percentage of tainted data points are added to the training set which can significantly alter the decision boundary. With only 3% poisoned data, it could produce considerable test errors for databases like Enron and IMDB. Our findings highlight the need for stronger defences against data poisoning assaults. Solutions were based on two concepts, it coordinated attacks to put poisoned points close together and structures each assault as a limited optimization problem with constraints designed to ensure that poisoned points remain undetected [17]. The surge in processing power and data accessibility have been credited with recent developments in machine learning.

However, malicious intervention with training data can compromise validity of models. [18] discussed data poisoning assaults and preventative methods before discussing open questions regarding testing models for data poisoning. [19] checks whether task-independent web-scale prior training in NLP might be applied to other domains with results. We discover that using this equation leads to the emergence of comparable behaviors in the discipline of machine vision and talk about the societal ramifications of this field of inquiry. CLIP models acquire a wide range of skills during retraining in order to maximize their training aim. Then, by using natural language prompting, this task learning may be used to enable reset-shot translation to several existing datasets. Although there is still considerable space for improvement, the performance of this technique may, at the appropriate scale, compete with task-specific supervised models.

III. METHODOLOGY

A. Data Pre-processing

Collect a dataset that includes both benign and potentially poisoned samples. The benign samples represent normal, clean data, while the potentially poisoned samples may contain hidden malicious patterns. Pre-process the data by removing any irrelevant or noisy samples. This step ensures that the model focuses on relevant patterns and avoids being misled by outliers or irrelevant data points. Perform data normalization or scaling to bring the features of the samples to a similar range. This step helps prevent certain features from dominating the learning process due to their larger magnitudes. We split the dataset into training and validation sets. The training set is used to optimize the model's parameters, while the validation set is used to assess the model's performance and prevent overfitting.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 8, August 2023

DOI: 10.17148/IJARCCE.2023.12815

B. Model Training

Design a Deep Neural Network (DNN) architecture suitable for the task at hand. This involves determining the number and type of layers, activation functions, and other hyper parameters. Figure 1 shows the DNN architecture. Initialize the model's weights and biases randomly or using pre-trained weights if available. Feed the training data into the model and calculate the predicted outputs. Compare the predicted outputs with the actual labels of the samples to compute the training loss, which represents the model's error. The goal is to minimize the training loss and improve the model's ability to learn patterns and make accurate predictions. Repeat the training process for multiple epochs, where each epoch represents a complete pass through the entire training dataset. This repetition helps the model refine its parameters and learn more complex representations.

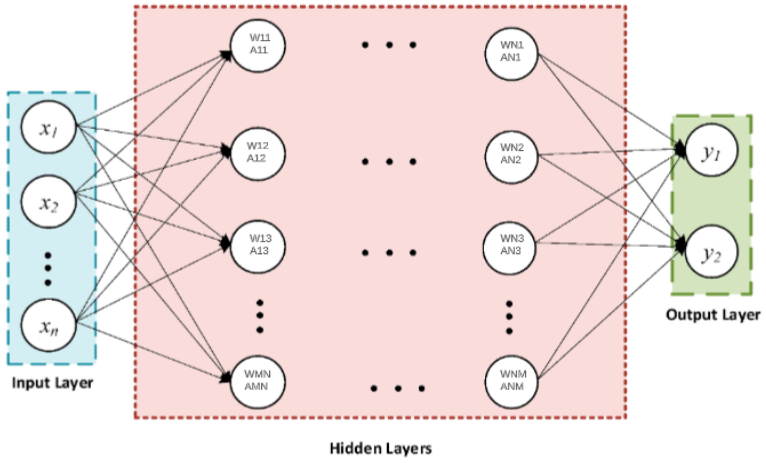

Fig 1: DNN Architecture

C. Feature Extraction

After training the DNN model, relevant features are extracted from each layer. This process involves obtaining the outputs or representations of the intermediate layers. Deep neural networks extract features at different abstraction levels. Early layers capture low-level features like edges, while deeper layers capture high-level concepts. Features can be obtained by accessing intermediate layer outputs or using techniques like activation maximization. Analyzing these features helps understand underlying patterns. In the process of estimating layer-wise centroids, the mean or centre points for each layer of interest in the trained DNN model are calculated. These centroids serve as reference points, capturing the typical features found in benign samples. By computing cosine similarity between the features of an incoming sample and the estimated centroids at each layer of interest, we can quantitatively measure the degree of similarity or alignment between sample's features and centroids. This cosine similarity metric helps determine how closely the features align with the centroids, providing insights into the sample's resemblance to benign patterns. By utilizing an algorithm or metric, we can identify the layer of interest where the feature differences between benign and potentially poisoned samples are most pronounced. This analysis involves analyzing the cosine similarity values across different layers and selecting the layer that exhibits the largest discrepancies or most significant differences. Identifying the layer of interest allows us to focus our further analysis on the layer that is most likely affected by potential poisoning or backdoor attacks.

D. Finding the Potentially Poisoned Inputs

When an incoming sample is suspicious or flagged as potentially poisoned, compute the cosine similarity between the sample's features and the estimated centroids at the layer of interest and the preceding layers.

IV. PROPOSED SYSTEM

The proposed approach uses layer-wise feature analysis to protect the backdoor attacks by comparing DNN characteristics for benign and poisoned samples. It identifies the critical layer with the greatest feature variations, which is often overlooked in existing defences. The system uses a simple backdoor detection method, comparing suspicious samples with benign samples. Extensive testing using benchmark datasets demonstrates its efficiency in recognizing and filtering poisoned samples.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 8, August 2023

DOI: 10.17148/IJARCCE.2023.12815

V. IMPLEMENTATION

A. Estimating Layer-wise Centroids

Model mending attempts to seal backdoors in exploited DNNs by modifying the DNNs using innocuous data. Model pruning can get rid of backdoors, and competing model unlearning was suggested as a technique to correct broken models. Input filtering is the primary focus of this study since it helps protect deployed DNNs.

B. Computing Cosine Similarities

We calculate the cosine similarity between the predicted centroids at every layer of interest and the features retrieved via the incoming sample (xs). In order to provide helpful data for subsequent analysis, this phase measures the degree to which The centroids' characteristics resemble those of the input sample.

C. Identifying the Layer of Interest (LOI)

Layer of Interest (LOI) is determined using the below mentioned algorithm in order to concentrate our investigation on the layers that are the most useful. Once found, compute a per-sample sum of the cosine similarity in the LOI and both layers that come before it. This enables to observe the behavior of the layers as a whole and gain an understanding of the patterns displayed by both healthy and possibly poisonous samples. Algorithm for loi is as represented below.

Input : Similarities of Cosine $\{cs_tL/2$, …… $cs_tL\}$for potential target class t
maxdiff ←$cs_tL/2+1$ - $cs_tL/2$; LOI_t ←[L/2] +1;
for l €{L/2} +2
ldiff ← cs_t - cs_tl-1 ;
if ldiff > maxdiff then
maxdiff ←ldiff ; LOI_t ←l;
return LOI_t

Algorithm identifies LOI using cosine similarities for target class 't'. initializes 'maxdiff' and 'LOIt', with 'maxdiff' being set as the difference in cosine similarity between layer 'bL/2c+1' and 'bL/2c' for target class 't'. '. The algorithm loops from 'bL/2c + 2' to the final layer 'L'. After the loop ends, the algorithm returns the final 'LOLt' value, representing the layer of interest for the target class 't'. Main goal is to determine the layer with the most notable difference in cosine similarities for a given target class.

D. Detection of Potentially Poisoned Inputs

Two actions are considered for every suspicious incoming input (xs) the trained model (fs) categorizes as belonging to class t. Then, at the three layers previously indicated, compute cosine similarity between the input sample and the estimated centroids. Then, using the mean determined in the preceding step, compare total similarities. Consider the input to be possibly tainted if the total of similarities is below the mean by a certain threshold or number of standard deviations. Algorithm for Layer wise feature analysis to detect backdoor attack is as shown below

Input : Suspicious trained DNN fs ; validation samples Xval ; Threshold T ; Suspicious input xs
Output : Boolean value (True/False) tells id xs is poisoned
for each potential target class t €{1,…..C} do -> An offline loop conducted for one time only

Xtval ← Split t's benign samples from Xval
Layers features generated by fs for {xi for all Xtval}
Estimate t's centroid at layer l €{[L/2],….L}
Cosine similarity (ail,atl) similarity of ail to its centroid
Aggregate computed benign similarities at layer l
LOIt ← IdentifyLayerOfInterest({cst[L/2], ….., cstL})
Calculate Mean, STD
ISPoisoned ← False
for each potential target class t €{1,……C} do
if ys = t then
Calculate cosine similarity (aSl,atl)
if css < (μt – T x σ t) then
IsPoisoned ← True
return Ispoisoned
Procedure IdentifyLayerOfInterest ({cs[L/2], ….csL})

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 8, August 2023

DOI: 10.17148/IJARCCE.2023.12815

```
max diff ← cs[L/2]+1 – cs[L/2]
LOI←- [L/2] +1
  for l €{[L/2] +2,…., L}do
      ldiff ← cs – csl-1
       if  ldiff  >maxdiff then
           maxdiff ←l diff
           LOI← l
   return LOI
```

Algorithm detects if input sample `xs` is poisoned using the suspicious deep neural network `fs`. Initialize target class `t`. Loop through potential classes. Separate benign samples from `t` as `Xtval`. Compute DNN `fs` layer features for `Xtval` and store in `a`. Estimate class `t` centroid at each layer `l` using mean of `a`. Calculate cosine similarity between sample features and layer centroids, save in `csl`. Compute `csˆl` for benign samples in each layer `l`. Find `LOIt` through aggregate similarities. Calculate mean (`μt`) and standard deviation (`σt`) of `csi` for benign samples `t`. Set `IsPoisoned` as `False`. Predict `yˆs` for `xs`. Loop through possible target class `t`. If `yˆs` matches `t`, compute cosine similarities for `xs` at `LOIt-2`, `LOIt-1`, and `LOIt`. Store in `csLOIts`, `csLOIt-1s`, and `csLOIt`. Calculate cosine similarity for `xs` (`css`). If `css` < (μt - τ × σt), it's poisoned. Set `IsPoisoned` as `True`. Check if `IsPoisoned` is true. Compare similarity of input sample to benign samples from other target classes. Suspicious inputs with dissimilar similarity patterns may indicate poisoning.

E. Leveraging cloud computing to train DNN

Cloud computing popularity for training Deep Neural Networks is due to its scalable and flexible resources, enabling efficient training through distributed computing and parallel processing. It is a pay-as-you-go model eliminates upfront infrastructure investments and allows cost optimization based on training requirements. Cloud-based environments provide accessibility, ease of use, pre-configured machine learning frameworks, and centralized storage and computing resources. This approach enables data sharing, model collaboration, and collective training efforts enabling researchers to overcome resource limitations, optimize costs, enhance cooperation and accelerate innovation in deep learning. Figure 2 and 3 represents examples of benign and poisoned data.

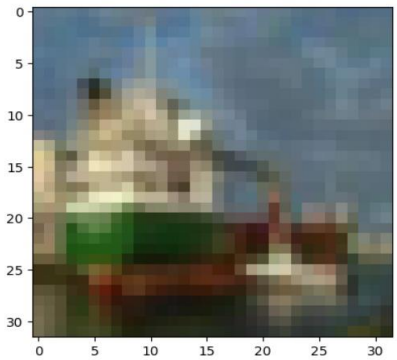

Fig 2: Example of a benign data

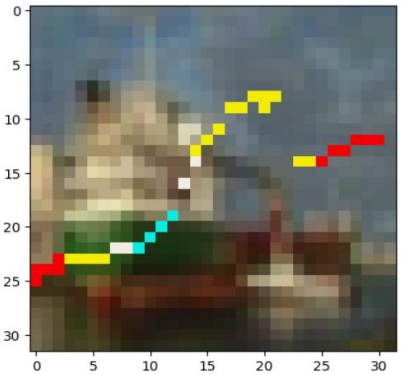

Fig 3: Example of a poisoned data

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 8, August 2023

DOI: 10.17148/IJARCCE.2023.12815

VI. RESULTS

GTSRB dataset evaluates current strategies and finds that their strategy exceeds cutting-edge techniques in identifying assaults that have low TPR or high FPR. The method's detection performs similarly to cutting-edge approaches and its TPR or FPR significantly beats them. Results confirm the detection's effectiveness. Figure 4 shows the identification of layer of interest (LOI) for the classes and Figure 5 depicts the impact of detection thresholds on TPR and FPR.

```
Number of layers: 10
Class:0, Layer of maximum difference:9
Class:1, Layer of maximum difference:8
Class:2, Layer of maximum difference:9
Class:3, Layer of maximum difference:9
Class:4, Layer of maximum difference:9
Class:5, Layer of maximum difference:9
Class:6, Layer of maximum difference:8
Class:7, Layer of maximum difference:9
Class:8, Layer of maximum difference:9
Class:9, Layer of maximum difference:9
```

```
Tau:0.5
TPR:100.00, FPR:27.45
TPR-STD:0.00, FPR-STD:2.05
------------------------------
Tau:1
TPR:99.98, FPR:16.04
TPR-STD:0.01, FPR-STD:1.02
------------------------------
Tau:1.5
TPR:99.88, FPR:7.67
TPR-STD:0.04, FPR-STD:0.50
------------------------------
Tau:2
TPR:99.76, FPR:3.29
TPR-STD:0.04, FPR-STD:0.38
```

Fig 4: Identify the layer the layer of interest (LOI) for the classes

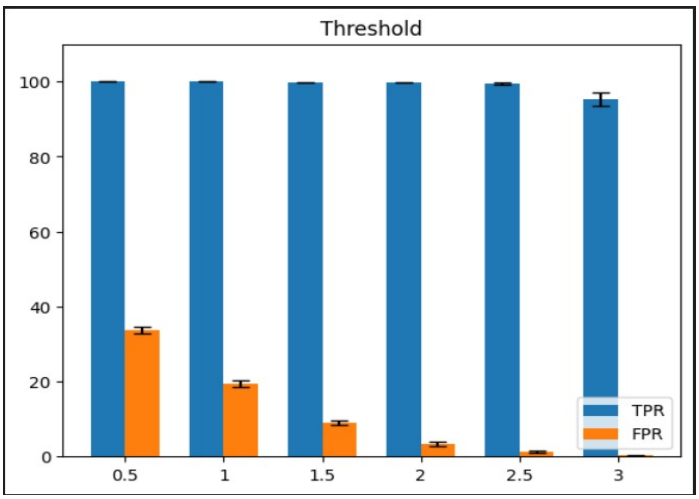

Fig 5: Impact of detection thresholds on TPR and FPR

VII. CONCLUSION AND FUTURE ENHANCEMENTS

Layer-wise feature analysis is used to study the behaviour of healthy and poisoned data produced by attacked DNNs in this article. Crucial layer is observed which is easily spotted based on the behaviours of benign samples, is likely to be where the typical contrast between benign & poisoned samples is largest. Based on this observation, the proposed scheme is a simple yet effective backdoor detection method for determining if a suspected diagnostic sample is poisoned by comparing its properties to those of a few neighbouring benign samples. Experiments utilising benchmark datasets demonstrated the effectiveness of detection. Study benefits in gaining a better understanding of DNN attack strategies in order to develop backdoor defences and more secure systems.

Enhancing robustness against advanced attacks ensures transferability of defence mechanisms across domains, datasets and network architectures. Development of real-time detection algorithms, improved interpretability, exploration of collaborative defence strategies, continuous monitoring networks for backdoor attacks, and deploying the defence mechanism in practical environments like financial institutions and healthcare systems are essential steps to be considered. Continuous monitoring and adaptation are also crucial for ensuring the effectiveness of the defence mechanism.

DOI: 10.17148/IJARCCE.2023.12815

REFERENCES

[1]. Raghavan, Vijay, Thomas Mazzuchi, and Shahram Sarkani. "An improved real time detection of data poisoning attacks in Deep Learning Vision systems." Discover Artificial Intelligence 2.1 (2022): 18. H. Chacon, S. Silva and P. Rad, "Deep Learning Poison Data Attack Detection".

[2]. Tang, Di, et al. "Demon in the variant: Statistical analysis of {DNNs} for robust backdoor contamination detection." 30th USENIX Security Symposium (USENIX Security 21). 2021.

[3]. Sun, Gan, et al. "Data poisoning attacks on federated machine learning." IEEE Internet of Things Journal 9.13 (2021): 11365-11375.

[4]. Chen, Yijin, et al. "Data poison detection schemes for distributed machine learning." IEEE Access 8 (2019): 7442-7454.Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, and Jaehoon Amir Safavi. 2017.

[5]. Baracaldo, Nathalie, et al. "Mitigating poisoning attacks on machine learning models: A data provenance based approach." Proceedings of the 10th ACM workshop on artificial intelligence and security. 2017.

[6]. Li, Yiming, et al. "BackdoorBox: A python toolbox for backdoor learning." arXiv preprint arXiv:2302.01762 (2023).

[7]. Ahmed, Ibrahim M., and Manar Younis Kashmoola. "Threats on machine learning technique by data poisoning attack: A survey." Advances in Cyber Security: Third International Conference, ACeS 2021,

[8]. Penang, Malaysia, August 24–25, 2021, Revised Selected Papers 3. Springer Singapore, 2021.

[9]. Hayase, Jonathan, and Weihao Kong. "Spectre: Defending against backdoor attacks using robust covariance estimation." International Conference on Machine Learning. 2020.

[10]. Müller, Nicolas, Daniel Kowatsch, and Konstantin Böttinger. "Data poisoning attacks on regression learning and corresponding defenses." 2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC). IEEE, 2020.

[11]. Jagielski, Matthew, et al. "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning." 2018 IEEE symposium on security and privacy (SP). IEEE, 2018.

[12]. Chen, Xinyun, et al. "Targeted backdoor attacks on deep learning systems using data poisoning." arXiv preprint arXiv:1712.05526 (2017).

[13]. Gu, Tianyu, et al. "Badnets: Evaluating backdooring attacks on deep neural networks." IEEE Access 7 (2019): 47230-47244.

[14]. Steinhardt, Jacob, Pang Wei W. Koh, and Percy S. Liang. "Certified defenses for data poisoning attacks." Advances in neural information processing systems 30 (2017).

[15]. Schwarzschild, Avi, et al. "Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks." International Conference on Machine Learning. PMLR, 2021.

[16]. Huang, W. Ronny, et al. "Metapoison: Practical general-purpose clean-label data poisoning." Advances in Neural Information Processing Systems 33 (2020): 12080-12091.

[17]. Koh, Pang Wei, Jacob Steinhardt, and Percy Liang. "Stronger data poisoning attacks break data sanitization defenses." Machine Learning (2022): 1-47.

[18]. Cinà, Antonio Emanuele, et al. "Machine learning security against data poisoning: Are we there yet?." arXiv preprint arXiv:2204.05986 (2022).

[19]. Xu, Youjiang, et al. "Faster meta update strategy for noise-robust deep learning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

