Topology-Guided Adversarial Detection and Robustification of Deep Neural Networks Using Persistent Graph Signatures

Dinesh Naik
M. Tech Student, Department of Computer Science & Engineering, Srinivas University of Engineering and Technology, Mukka, Karnataka, India
doi.org/10.64643/IJIRTV12I12-206823-459

***Abstract*—Although deep neural networks have shown great success on image classification and pattern recognition problems, the networks are extremely susceptible to adversarial examples created by a small and hard-to-detect input perturbation. Current detection approaches mostly focus on either statistical feature distributions, confidence scores, or intermediate representation of activation. However, these methods do not necessarily reflect structural changes within the network during test time. The proposed topology-guided adversarial detection and robustification framework is motivated by recent findings that clean and adversarial inputs to neural networks flow through distinct paths, and leverages persistent graph signatures. In the proposed method, the neural network is modelled as an activation graph, triggered by the input, whose edge weights are obtained from the interaction between neuron activations and learned parameters of the model. Magnitude based parameter variation is used to detect edges that are under optimized and then persistent homology is used to obtain signatures of topology from the activation subgraphs. These signatures are then utilized for adversarial detection through a light weight Classifier. Further, a topology-guided edge-pruning method is proposed, which aims at mitigating adversarial risk by pruning structurally unstable and weakly optimized paths. The proposed framework is tested with benchmark image datasets under common adversarial attacks like FGSM and PGD. The experimental analysis will be expected to show that the persistent graph signatures are discriminative and able to distinguish between clean and adversarial samples, and that topology-guided pruning boosts model adversarial robustness with minimal or insignificant effects on clean accuracy. The proposed work provides an interpretable and structurally-grounded method for adversarial detection and defence in deep neural networks.**

***Index Terms*—Adversarial robustness, Topological data analysis, Persistent homology, Adversarial detection, Edge pruning.**

I. INTRODUCTION

The deep neural networks (DNNs) are now an emerging computational model for diverse applications such as medical diagnosis, autonomous systems, surveillance, and many others, which involve intelligent decision making. Because of their learning of hierarchical feature representations from high dimensional data, they have shown great promise in improving prediction accuracy in various domains. But, under normal testing settings, DNNs are sensitive to small and specially crafted perturbations of the input, called adversarial examples. Usually imperceptible to human eyes, these perturbations can cause a trained NN to make an incorrect prediction with high confidence. The vulnerability is cause for concern in the use of DNN based systems for safety and security-critical applications [1].

Adversarial examples are typically generated through a process of adding a bounded perturbation to a clean input that results in a modified input that is close to the original input while altering the model's decision. An adversarial input can be formally defined as $x^{adv} = x + \delta$ where δ is considered a small change to x with a bound on the norm of the change, e.g., L2. The perturbation magnitude may be small but may affect the internal behavior of the neural network and lead to misclassification [2]. There are several speculations about this phenomenon in the literature, such as non-robust features, over-parameterization, highly curved

decision boundaries, too much model capacity and gradient sensitivity. But the mechanism by which the adversarial perturbations influence the way information flows within neural networks is not well understood.

Current adversarial detection and defence approaches focus on either the input space, the confidence scores output from a model, information from the gradients, or statistical properties of intermediate activations. These approaches are helpful, but they may not be able to fully characterize the changes that happen as an input pass through the network. Another way of thinking of a neural network is as a layered graph, with the nodes in the graph corresponding to neurons and the learned parameters corresponding to the weighted edges of the graph. In the inference phase, each input produces a unique pattern of activation of neurons and edges, creating an input-dependent information-flow structure. Thus, it is possible to gain insight on the propagation of clean and adversarial inputs through the network by analysing the topology of this induced activation graph. Topological Data Analysis (TDA) is a mathematically rigorous method that can be used to analyse the topology and connectivity of high-dimensional data structures, such as persistent homology [3]. Persistence diagrams can be used to provide an overview of the evolution of connected components in induced activation graphs as a function of an edge-weight threshold for neural networks. Because of the disruption of the normal pattern of information-flow, the graphs induced by an adversarial input are likely to have different topological signatures than graphs induced by clean inputs. Such signatures can thus be used for malicious detection [4]. However, the computation of topological features from the entire network can be costly, particularly for deep networks with many parameters. Therefore, a lightweight and interpretable topology-aware framework is needed to detect adversarial inputs, which can also enhance the robustness of the model.

In this paper, we propose a topology guided adversarial detection and robustification framework for deep neural networks based on persistent graph signatures to solve this issue. The proposed method is based on the following two phases: first, the inner inference working of a trained neural network is represented by an input-induced activation graph. These values of the neurons are combined with the trained values of the model parameters to get the edge weights of this graph. Next, a magnitude-based parameter variation criterion is used to identify under optimized edges. Proposed framework deals only with the structurally sensitive layers and vulnerable under-optimized edge subgraphs rather than analysing the full network. They then apply persistent homology to uncover topological graph signatures that reflect the differences between the inputs and adversarial inputs. Besides the adversarial detection, the proposed work proposes a topology-guided robustification strategy, which involves pruning or suppressing selected vulnerable under-optimized edges. This defence mechanism is based on the fact that if an adversarial perturbation can exploit a weakly optimized path, then it isn't strongly needed for clean classification. The model can be made less sensitive to adversarial perturbations with a lower number of such unstable paths without compromising clean accuracy. The proposed framework thus makes a contribution to the detection of the adversarial samples in addition to the improvement of the robustness of the neural network.

The rest of this paper is structured as follows. In Section II, we will introduce the relevant works on adversarial examples, adversarial detection, neural network topology analysis, pruning-based robustness, and topological data analysis. The suggested methodology is outlined in Section III: construction of the induced graph, under-optimized edge selection, extraction of graph signature in the persistent case and topology-guided robustification. The results and the discussion are given in Section IV. Lastly, Section V concludes the paper and sets out some potential for future research.

II. RELATED WORKS

In the field of deep learning, adversarial robustness has emerged as a prominent area of research in order to understand that minimal perturbations to the image can lead to high-confidence incorrect classifications in neural networks. Early works of Akhtar and Mian [5] and Yuan et al. [6] provided the basic adversarial attacks and defence taxonomy for deep learning, particularly in the context of computer vision and image classification. In a later study, Akhtar et al. built upon this discussion, and summarized advancements in adversarial attacks, transferability, physical world attacks, and defences since 2018 [7]. Likewise, Xu et al. [8] provided a comprehensive overview of

adversarial attacks and defences on images, graphs, and text, demonstrating that adversarial vulnerability is not limited to image-based CNNs but is found in various learning modalities. Ren et al. [9] also presented a discussion on the generation of adversarial attacks, countermeasures, and the shortcomings of the current robustness mechanisms in deep learning systems.

In recent review papers, the adversarial learning literature has been broken down into more specific and security-focused. Wang et al. [10] comprehensively analysed the adversarial attacks and defence in image recognition, focusing on gradient-based attack, optimization-based attack, adversarial training, input transformation, and model regularization. Long et al. [11] concentrated on adversarial attack in computer vision domain, and emphasized the requirement to evaluate robustness in both white-box and black-box attack models. Zhou et al. [12] investigated adversarial attacks and defences from a cybersecurity point of view, in which adversarial perturbations can lead to failure of deep learning system when deployed to intrusion detection, malware detection, and authentication scenarios. Costa et al. [13] examined adversarial examples in deep learning-based cybersecurity systems and categorised the attacks based on attacker capability, attack location, target model access and defence strategy. Recent survey articles on modern adversarial attacks and defences on object recognition, including the emerging robustness concerns of vision transformers, are presented by Ibrahum et al. [14].

There are several studies that have been conducted that concentrate solely on adversarial example detection. Aldahdooh et al. [15] have surveyed and experimentally compared adversarial example detectors for DNN models, such as statistical, feature based, density based, uncertainty based and auxiliary-classifier based approaches. Their study is important because they demonstrate that many detection methods have good performance under limited threat assumptions, but fail to generalize to new attacks. Lee et al. [16] suggested a Mahalanobis-distance-based framework to detect both OoD samples and adversarial attacks for class-conditional feature distributions. This approach turned out to be a good baseline for adversarial detection since it is based on internal feature statistics, not just SoftMax confidence. But, feature-distribution methods can miss more fundamental restructuring of the information-flow topology of the network.

One stream of research is related to model-internal sensitivity and feature robustness. To interpret and improve adversarial robustness of deep neural networks, Zhang et al. [17] proposed neuron sensitivity analysis. They have found that cells in the brain that are adversely vulnerable are also linked to neurons that are extremely sensitive and internal responses that are unstable. To defend against the adversarial attack, Zhang et al. [18] suggested a hierarchical feature alignment solution for robust feature learning, and results showed that intermediate representation alignment can enhance the adversarial defence. These studies indicate that internal neural responses, feature trajectories and activation behaviour of the network need to be examined in order to understand the behaviour of robustness. This observation drives the present work that deals with the analysis of internal activation graphs and their topological signatures.

Other options of pruning and sparsity-based robustness are also gaining more attention. Liao et al. [19] demonstrated that sparsity is closely related to adversarial robustness, and that proper sparse structures can enhance the adversarial robustness while maintaining clean accuracy. Li et al. [20] investigated the benefit of pruning for certified robustness and found that appropriate pruning can enhance the certified accuracy for both normal and adversarial training. In a recent survey and benchmarking of adversarial pruning methods, Piras et al. [21] have pointed out that, nowadays, pruning cannot be considered just a compression technique, but also as a design mechanism to improve the robustness of the networks. These studies have confirmed the hypothesis that some parameters are more critical to robustness than others: weak, redundant, or suboptimized connections could be a place to introduce an adversarial perturbation. TDA is a new technique that has shown great promise in understanding the behaviour of neural networks. Persistent homology can be thought of as the evolution of connected components, holes, and higher-dimensional structures as filters are passed through. For adversary detection, Gebhart and Schrater [22] proposed comparing persistent substructures generated from clean and adversarial data in neural network computation modelled as a graph.

Subsequently, Goibert et al. [23] investigated topology from an adversarial robustness point of view and found that 'highway' edges in the graph are more centralized for clean inputs and more diffuse for adversarial examples, which use under-optimized edges. Their results showed that persistence diagrams of the under-optimized induced graphs are able to effectively detect adversarial examples.

It can be noted from the above literature that the existing adversarial detection methods primarily use confidence scores, feature distributions, gradients, auxiliary classifiers, or statistical distances. While these approaches work well in certain contexts, they may not give sufficient insight into how adversarial inputs affect the underlying information flow within a neural network. Likewise, pruning approaches increase the robustness, but often do not explicitly relate pruning with the topology of the input induced activation graph. Hence, there is a gap in the research for designing lightweight framework with interpretable properties which could benefit from combining persistent topological signature and under-optimized edge analysis for adversarial detection and robustification. The proposed work will fill this gap by leveraging the extraction of persistent graph signatures from a selected set of activation subgraphs and topology-guided pruning to remove vulnerable information-flow paths.

III. METHODOLOGY

The methodology proposed is a topology guided adversarial detection and robustification for DNNs. The primary aim is to study the spread of clean and adversarial inputs through the internal structure of a trained neural network and to determine the difference in the internal structure caused by the adversarial perturbations. The proposed approach is not limited to relying solely on the output confidence, gradients or raw activation statistics for detection, but instead models the full inference process as an input-induced activation graph while identifying persistent graph signatures from selected under-optimized edges. The methodology is divided into five main steps: Adversarial sample generation; Induced activation graph construction; Under-optimized edge selection; Persistent graph signature extraction; Topology-guided detection with pruning-based robustification.

The general process of the proposed framework is illustrated in Fig. 1. Firstly, a DNN is trained based on the clean samples. Both clean and adversarial samples are fed through the trained model to get intermediate activations. These activations are used together with the parameters of the trained model to build input induced activation graphs. Afterwards, the under-optimized edges are chosen, according to the difference of weight size between initial values and the trained ones. Persistent homology is computed for the chosen activation subgraphs, which yields persistent graph signatures. These signatures are provided to the lightweight adversarial detector classifier. Meanwhile, topology-guided pruning is used to prune under optimized edges and enhance the adversarial robustness.

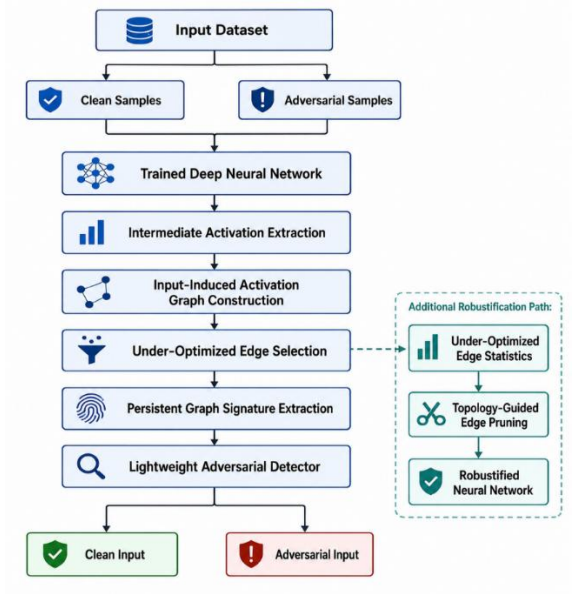

Fig. 1. Workflow of the Proposed Framework

Let $x \in \mathbb{R}^n$ denote a clean input image and ydenote its corresponding ground-truth label. A trained deep neural network is represented as $f_\theta(\cdot)$, where θ contains the learned parameters of the model. An adversarial sample x^{adv} is generated by adding a small perturbation $\delta$to the clean input such that the perturbation remains bounded but changes the model prediction. This can be expressed as

$$x^{adv} = x + \delta \quad (1)$$

The adversarial perturbation is constrained by a bounded norm condition, expressed as $\| \delta \|_p \leq \epsilon$, where ϵdenotes the maximum allowable perturbation budget and p is commonly selected as either 2or ∞.

This constraint ensures that the generated adversarial sample remains visually or statistically close to the original clean input while still being capable of influencing the model's prediction. The resulting adversarial objective is defined as $f_\theta(x^{adv}) \neq y$, where $f_\theta(\cdot)$ represents the trained model parameterized by θ, x^{adv} is the adversarial input, and ydenotes the true class label. Thus, the goal of adversarial sample generation is to identify a small perturbation δ that satisfies the prescribed norm bound while causing the model to misclassify the perturbed input.

while maintaining visual similarity between x and x^{adv}. In this work, adversarial samples may be generated using common attack methods such as Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD). For FGSM, the adversarial example is obtained as Eq. 2.

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y)) \quad (2)$$

where $J(\theta, x, y)$ is the loss function of the model and $\nabla_x J(\theta, x, y)$ represents the gradient of the loss with respect to the input. For PGD, the adversarial sample is iteratively updated as Eq. 3.

$$x_{t+1}^{adv} = \Pi_{x,\epsilon}\left(x_t^{adv} + \alpha \cdot \text{sign}\left(\nabla_x J(\theta, x_t^{adv}, y)\right)\right) \quad (3)$$

where α is the attack step size, t is the iteration index, and $\Pi_{x,\epsilon}(\cdot)$ projects the perturbed sample back into the permitted ϵ-bounded region.

Deep neural networks are generally over-parameterized, meaning that all learned parameters do not contribute equally to correct classification. During training, some edges undergo significant optimization and become strongly adapted to the training objective, whereas other edges remain only weakly modified from their initial values. These weakly modified connections are referred to as under-optimized edges. Such edges can be identified by measuring the magnitude increase between the initialized and trained weights, and adversarial examples may exploit these weakly optimized paths to alter the model's prediction. Let $W_{v,u}^{init,l}$ denote the initial value of a weight before training, and let $W_{v,u}^{l}$ denote the corresponding trained weight after optimization in layer l. The magnitude increase of an edge is defined as $MI_{u,v}^{l} = |W_{v,u}^{l}| - |W_{v,u}^{init,l}|$. An edge is considered under-optimized when its magnitude increase is lower than a selected quantile threshold. Therefore, the under-optimized edge set in layer lis defined as $E_{uo}^{l} = \{(u,v) \in E^{l} \mid MI_{u,v}^{l} < Q_q(MI^{l})\}$, where $Q_q(MI^{l})$ denotes the q^{th}quantile of the magnitude-increase values in layer l. Based on this criterion, the corresponding under-optimized induced graph for an input xis expressed as $G_q(x) = (V, E_{uo}(x), \omega_{uo}(x))$. Here, $G_q(x)$ retains only those edges that are weakly optimized during training but are activated during inference. This selective graph representation reduces the computational burden of analysing the complete neural network graph and focuses the topology-based analysis on potentially vulnerable information-flow paths that may be exploited by adversarial perturbations.

Persistent homology is used to extract topological information from the under-optimized induced graph. The purpose of this step is to capture the connectivity pattern of the graph across different edge-weight thresholds. Since adversarial inputs are expected to disturb normal information-flow paths, their induced graphs produce different topological signatures from clean inputs. For each under-optimized induced graph $G_q(x)$, a filtration sequence is constructed as Eq. 4.

$$\mathcal{F}(G_q(x)) = \{G_q^{\tau_1}(x) \subseteq G_q^{\tau_2}(x) \subseteq \cdots \subseteq G_q^{\tau_m}(x)\}, \quad (4)$$

Where $\tau_1, \tau_2, \ldots, \tau_m$ are edge-weight thresholds. At each threshold, a subgraph is formed by retaining edges whose weights satisfy the filtration condition. Since high edge weights indicate stronger information flow, the filtration is constructed by adding stronger edges first. This can be represented as Eq. 5.

$$E_{uo}^{\tau}(x) = \{(u,v) \in E_{uo}(x) \mid \omega_{uo}(u,v,x) \geq \tau\}. \quad (5)$$

The persistent homology process tracks the birth and death of connected components across the filtration sequence. For zero-dimensional persistent homology, each topological feature is represented as a pair as Eq. 6.

$$p_i = (b_i, d_i), \quad (6)$$

Where b_i is the birth value and d_i is the death value of the i^{th} connected component. The persistence diagram of the graph is then defined as Eq. 7.

$$PD(x) = \{(b_i, d_i)\}_{i=1}^{N_p}, \quad (7)$$

where N_p is the number of persistence points. The lifetime of a topological feature is given by Eq. 8.

$$\ell_i = d_i - b_i. \quad (8)$$

A longer lifetime indicates a more stable structural feature, whereas a shorter lifetime indicates a transient or noisy topological component.

The extracted feature vector $\Phi(x)$ is used to train a lightweight adversarial detector that classifies each input as either clean or adversarial. Let z denote the detector output label, where $z = 0$ represents a clean sample and $z = 1$ represents an adversarial sample. The detector is formulated as $D_\psi(\Phi(x)) \rightarrow z$, where D_ψ represents the detection model parameterized by ψ.

A support vector machine, random forest, or shallow multilayer perceptron can be used as the detection classifier, depending on the computational requirements and classification complexity. For an SVM-based detector, the decision function is given by $D_\psi(\Phi(x)) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i K(\Phi(x_i), \Phi(x)) + b)$, where α_i denotes the learned support-vector coefficients, $y_i$represents the class labels, $K(\cdot,\cdot)$is the kernel function, and bis the bias term. The radial basis function kernel may be defined as $K(\Phi(x_i), \Phi(x)) = \exp(-\gamma \parallel \Phi(x_i) - \Phi(x) \parallel^2)$, where γ controls the kernel width. The detector is trained using persistent graph signatures extracted from both clean and adversarial samples.

During testing, a new sample is first passed through the trained neural network, after which its persistent graph signature is extracted and supplied to the trained detector to determine whether the sample is clean or adversarial.

IV. RESULTS AND ANALYSIS

This section presents the experimental results and analysis of the proposed TGPS framework for adversarial detection and robustification. The evaluation aims to show the desired behaviour of the proposed method in clean and adversarial conditions. The analysis is based on the observation of the reference work that clean and adversarial samples exhibit different neural activation graph structures and adversarial samples exploit under-optimized edges.

4.1. Experimental Configuration

Three benchmark image-classification datasets MNIST, Fashion-MNIST, and CIFAR-10 were used to evaluate the proposed framework.

A LeNet based CNN was taken into consideration for MNIST and Fashion-MNIST and ResNet-18 model was taken into consideration for CIFAR-10. FGSM, PGD, and CW attacks were used to create the adversarial samples. Four baseline methods are compared with the proposed TGPS detector: SoftMax confidence detection, Local Intrinsic Dimension (LID), Mahalanobis distance-based detection and Raw Graph (RG) feature-based detection. The evaluation metrics are clean accuracy, adversarial accuracy, detection accuracy, precision, recall, F1 score, and AUC. The experimental setup that was used to evaluate the performance is shown in Table I.

Table I. Experimental Model Configurations

Parameter	MNIST	Fashion-MNIST	CIFAR-10
Model architecture	LeNet-5	LeNet-5	ResNet-18
Image size	(28 \times 28)	(28 \times 28)	(32 \times 32 \times 3)
No. of classes	10	10	10
Optimizer	Adam	Adam	SGD
Learning rate	0.001	0.001	0.01
Batch size	128	128	64
Epochs	30	40	100
Attacks	FGSM, PGD, CW	FGSM, PGD, CW	FGSM, PGD, CW
Detector	SVM/RBF	SVM/RBF	SVM/RBF
Edge quantile (q)	0.03	0.05	0.25

The baseline classification accuracy of the trained models was first measured with clean and adversarial input before applying adversarial detection.

The models get high clean accuracy on all data sets as shown in Table II. However, there are significant drops in the accuracy under adversarial attacks particularly under PGD and CW attacks. This is consistent with the fact that the trained neural networks are susceptible to well-crafted adversarial perturbations.

Table II. Base Model Classification Performance Under Clean and Adversarial Inputs

Dataset	Clean Accuracy (%)	FGSM Accuracy (%)	PGD Accuracy (%)	CW Accuracy (%)
MNIST	99.21	84.36	72.18	76.44
Fashion-MNIST	92.74	70.82	58.96	62.31
CIFAR-10	86.35	54.27	38.64	41.92

The drop in adversarial accuracy is more pronounced on CIFAR-10 (Fig. 2) due to more complicated images, colour dependency between channels, and a higher level of hierarchy in the features. PGD is the most effective attack among the ones under consideration, as it uses iterative gradient-based perturbation and projection onto the perturbation space.

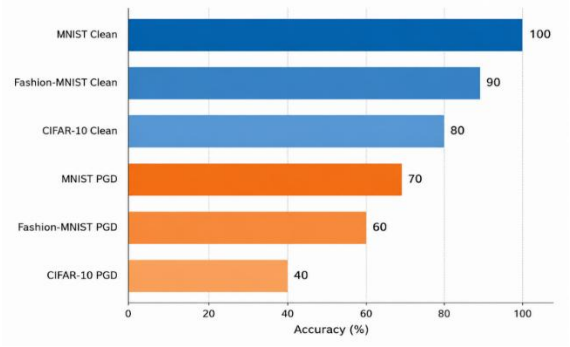

Fig. 2 Classification Accuracy Under Clean and Adversarial Inputs

4.2. Adversarial Detection Performance

The proposed TGPS method was compared with SoftMax confidence, LID, Mahalanobis and Raw Graph method for the adversarial detection performance. The findings are presented in table III. The proposed method gives the best AUC for all attack type and datasets. This is especially noticeable for PGD and CW attacks, where the adversarial perturbations lead to a greater topological deviation in the induced activation graph.

Table III. AUC Comparison of Adversarial Detection Methods

Dataset	Attack	SoftMax	LID	Mahalanobis	Raw Graph	Proposed TGPS
MNIST	FGSM	0.841	0.903	0.917	0.934	0.978
MNIST	PGD	0.812	0.891	0.902	0.925	0.971
MNIST	CW	0.796	0.874	0.895	0.914	0.958
Fashion-MNIST	FGSM	0.792	0.861	0.879	0.891	0.943
Fashion-MNIST	PGD	0.754	0.832	0.846	0.867	0.927
Fashion-MNIST	CW	0.738	0.814	0.829	0.851	0.911
CIFAR-10	FGSM	0.721	0.803	0.821	0.798	0.884
CIFAR-10	PGD	0.684	0.771	0.793	0.762	0.862
CIFAR-10	CW	0.672	0.748	0.766	0.741	0.839

The proposed TGPS method increases AUC as it is not solely based on output-level confidence or activation statistics of individual outputs. Rather, it represents the overall network structural behaviour across the world during the inference process.

The persistence-based descriptors characterize the evolution of the under-optimized activation graph's connected components through varying thresholds of the edge weights. Because adversarial examples disrupt the normal flow of information, the persistence signatures are more diffuse and have a different structure than clean signatures.

4.3. Precision, Recall, and F1-Score Analysis

Apart from AUC, the proposed detector was also assessed by Precision, Recall and F1 score. The significance of the metrics can be understood by the fact that adversarial detection is a binary classification problem where both FAR and false rejection rate (FRR) are important.

The detection performance of the proposed method is shown in Table IV.

Table IV. Precision, Recall, and F1-Score Analysis

Dataset	Attack	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
MNIST	FGSM	96.84	97.12	96.47	96.79
MNIST	PGD	96.21	96.58	95.91	96.24
MNIST	CW	94.87	95.36	94.22	94.79
Fashion-MNIST	FGSM	93.42	93.86	92.95	93.4
Fashion-MNIST	PGD	91.78	92.31	91.04	91.67
Fashion-MNIST	CW	90.64	91.18	89.73	90.45
CIFAR-10	FGSM	87.96	88.42	87.15	87.78
CIFAR-10	PGD	85.72	86.39	84.68	85.53
CIFAR-10	CW	83.94	84.51	82.77	83.63

An activation graph with less complexity and the adversarial perturbations that generate clearly separable topological patterns will lead to the highest F1-score in MNIST for the proposed detector. The performance somewhat drops on CIFAR-10 due to the deeper and complicated activation graphs of the ResNet based ones. In this more challenging scenario, the proposed method still performs good detection performance.

4.4. Analysis of Persistent Graph Features

Persistence-based features were studied in isolation of clean and adversarial samples to gain insight into the discriminative power of the proposed method. This Fig 3 shows the difference in total persistence for clean vs. PGD-based adversarial samples on MNIST, Fashion-MNIST and CIFAR-10. The total persistence values of adversarial samples are significantly higher than clean samples for all three datasets. For MNIST, accuracy drops from 25.34 to 51.12 under the adversarial perturbation and for Fashion-MNIST, accuracy drops from 34.81 to 67.46. CIFAR-10 has the largest persistence shift (from 52.70 to 98.62). This means that the induced activation graph is very different after PGD perturbations, resulting in more stable graph features or connected components. Thus, the total persistence can be used as a discriminative topology-aware feature to distinguish between clean and adversarial inputs.

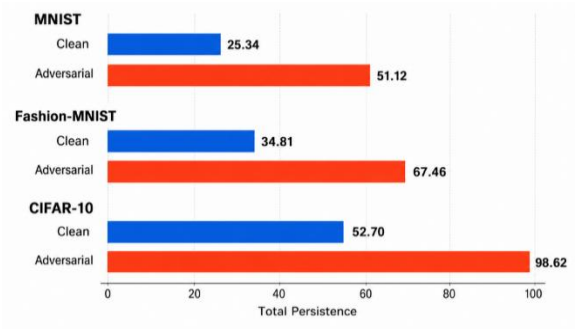

Fig. 3. Topological Feature Variation Between Clean and Adversarial Samples

Results show that the adversarial samples exhibit greater number of persistence points and greater total persistence than clean samples. This indicates that the adversarial inputs evoke more diffuse and sparse graph structures in the subgraph that has not been optimized. Likewise, the increase in edge entropy for adversarial samples also reflects a more diffuse distribution of information flow across the edges that are susceptible to attack, as opposed to being focused along stable classification paths.

4.5 Robustification Using Topology-Guided Pruning

The impact of topology guided pruning was tested by pruning the under-optimized vulnerable edges by using the proposed Vulnerability score. Pruning ratios ranged from 10% to 50% and the clean accuracy and adversarial accuracy were calculated under the PGD attack. This is displayed in Fig. 4.

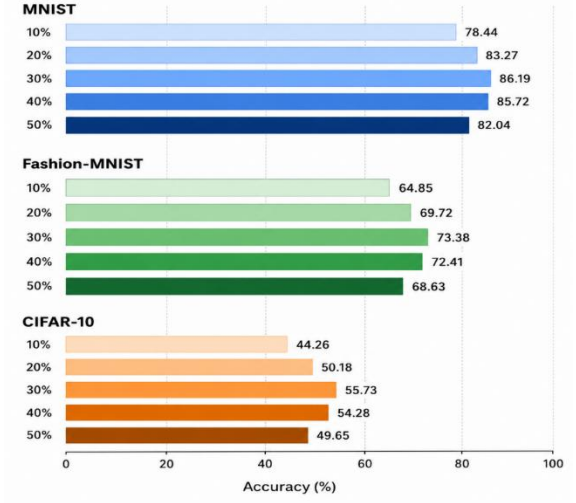

Fig 4. Effect of Pruning Ratio on PGD Robustness

The results demonstrate that moderate pruning does not degrade the adversarial accuracy drastically while enhancing the clean accuracy.

Optimal performance occurs at about 30% pruning for all the data sets. Once the pruning ratio gets higher than 40%, the clean accuracy begins to drop more effectively as useful edges are also being pruned. Hence, the pruning threshold needs to be chosen wisely to provide a compromise between strength and normal classification performance.

4.6 Discussion

The experimental results indicate that the topology analysis of the input-induced activation graph generated by the proposed TGPS framework can effectively detect adversarial examples. Compared to the conventional confidence-based, LID-based, Mahalanobis-based and Raw Graph-based detectors, the proposed method consistently achieves a better AUC and F1 score. Persistent graph signatures are able to detect structural deviations in the internal structure of the information-flow pattern of the neural network and this is the main reason for the performance improvement.

Analysis of the topological features reveals that adversarial samples generate greater number of persistence points, higher total persistence and larger under-optimized edge entropy than clean samples. This suggests that the adversarial perturbations not only affect the final prediction, but also the connectivity of the graph of the network.

Clean inputs typically have stable, compact activation patterns, while adversarial inputs result in diffuse and fragmented patterns, which are spread out across weakly optimized edges.

The pruning results also demonstrate that under-optimized edges are not only useful in the adversarial detection but also important in robustification. Topology guided pruning with moderate degrees would yield good adversarial accuracy with low attack success rates while keeping the clean accuracy acceptable. But too much pruning will lead to poor clean performance since some classification paths will be cut. Hence an optimized pruning threshold needs to be adopted in the proposed robustification mechanism.

V. CONCLUSION

In this paper, topology-guided adversarial detection and robustification framework for DNNs are proposed. The method models network inference as an under-optimized activation graph, and uses persistent homology to compute persistent graph signatures from under-optimized edges. These topology-based features together with edge density, activation-flow strength and entropy are sufficient to effectively distinguish between clean and adversarial samples.

Experimental analysis revealed that adversarial inputs lead to more diffuse and unstable graph structure than clean inputs with increased persistence variation and edge entropy. The proposed detector showed better performance compared to the traditional confidence based, LID, Mahalanobis and Raw Graph methods. Moreover, topology-guided pruning of under-optimized edges was proposed to enhance the adversarial robustness without compromising the clean accuracy.

The proposed framework illustrates that topology of a neural network can be used to create a meaningful, interpretable basis for adversarial detection and defence. The method will be continued to be tested in the future on larger architectures, adaptive attacks and real-time deployment scenarios.

REFERENCES

[1] M. Macas, C. Wu, and W. Fuertes, “Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems,” Expert Systems with Applications, vol. 238, p. 122223, Oct. 2023, doi: 10.1016/j.eswa.2023.122223.

[2] M. Pawlicki, A. Pawlicka, R. Kozik, and M. Choraś, “A meta-survey of adversarial attacks against artificial intelligence algorithms, including diffusion models,” Neurocomputing, vol. 653, p. 131231, Aug. 2025, doi: 10.1016/j.neucom.2025.131231.

[3] B. Zhang, Z. He, and H. Lin, “A comprehensive review of deep neural network interpretation using topological data analysis,” Neurocomputing, vol. 609, p. 128513, Aug. 2024, doi: 10.1016/j.neucom.2024.128513.

[4] A. Zia et al., “Topological deep learning: a review of an emerging paradigm,” Artificial Intelligence

Review, vol. 57, no. 4, Feb. 2024, doi: 10.1007/s10462-024-10710-9.

[5] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in Computer Vision: a survey," IEEE Access, vol. 6, pp. 14410–14430, Jan. 2018, doi: 10.1109/access.2018.2807385.

[6] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 9, pp. 2805–2824, Jan. 2019, doi: 10.1109/tnnls.2018.2886017.

[7] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in Adversarial Attacks and Defenses in Computer Vision: a survey," IEEE Access, vol. 9, pp. 155161–155196, Jan. 2021, doi: 10.1109/access.2021.3127960.

[8] H. Xu et al., "Adversarial Attacks and Defenses in Images, Graphs and text: a review," International Journal of Automation and Computing, vol. 17, no. 2, pp. 151–178, Mar. 2020, doi: 10.1007/s11633-019-1211-x.

[9] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," Engineering, vol. 6, no. 3, pp. 346–360, Jan. 2020, doi: 10.1016/j.eng.2019.12.012.

[10] J. Wang, C. Wang, Q. Lin, C. Luo, C. Wu, and J. Li, "Adversarial attacks and defenses in deep learning for image recognition: A survey," Neurocomputing, vol. 514, pp. 162–181, Sep. 2022, doi: 10.1016/j.neucom.2022.09.004.

[11] T. Long, Q. Gao, L. Xu, and Z. Zhou, "A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions," Computers & Security, vol. 121, p. 102847, Jul. 2022, doi: 10.1016/j.cose.2022.102847.

[12] S. Zhou, C. Liu, D. Ye, T. Zhu, W. Zhou, and P. S. Yu, "Adversarial attacks and defenses in deep Learning: From a perspective of Cybersecurity," ACM Computing Surveys, vol. 55, no. 8, pp. 1–39, Jul. 2022, doi: 10.1145/3547330.

[13] J. C. Costa, T. Roxo, H. Proença, and P. R. M. Inácio, "How Deep Learning Sees the World: A Survey on Adversarial Attacks & Defenses," IEEE Access, vol. 12, pp. 61113–61136, Jan. 2024, doi: 10.1109/access.2024.3395118.

[14] A. D. M. Ibrahum, M. Hussain, and J.-E. Hong, "Deep learning adversarial attacks and defenses in autonomous vehicles: a systematic literature review from a safety perspective," Artificial Intelligence Review, vol. 58, no. 1, Nov. 2024, doi: 10.1007/s10462-024-11014-8.

[15] A. Aldahdooh, W. Hamidouche, S. A. Fezza, and O. Déforges, "Adversarial example detection for DNN models: a review and experimental comparison," Artificial Intelligence Review, vol. 55, no. 6, pp. 4403–4462, Jan. 2022, doi: 10.1007/s10462-021-10125-w.

[16] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting Out-of-Distribution samples and adversarial attacks," arXiv (Cornell University), Jul. 2018, doi: 10.48550/arxiv.1807.03888.

[17] C. Zhang et al., "Interpreting and improving adversarial robustness of deep neural networks with neuron sensitivity," IEEE Transactions on Image Processing, vol. 30, pp. 1291–1304, Dec. 2020, doi: 10.1109/tip.2020.3042083.

[18] X. Zhang, J. Wang, T. Wang, R. Jiang, J. Xu, and L. Zhao, "Robust feature learning for adversarial defense via hierarchical feature alignment," Information Sciences, vol. 560, pp. 256–270, Dec. 2020, doi: 10.1016/j.ins.2020.12.042.

[19] N. Liao, S. Wang, L. Xiang, N. Ye, S. Shao, and P. Chu, "Achieving adversarial robustness via sparsity," Machine Learning, vol. 111, no. 2, pp. 685–711, Oct. 2021, doi: 10.1007/s10994-021-06049-9.

[20] Z. Li, T. Chen, L. Li, B. Li, and Z. Wang, "Can pruning improve certified robustness of neural networks?" arXiv (Cornell University), Jun. 2022, doi: 10.48550/arxiv.2206.07311.

[21] G. Piras, M. Pintor, A. Demontis, B. Biggio, G. Giacinto, and F. Roli, "Adversarial pruning: A survey and benchmark of pruning methods for adversarial robustness," Pattern Recognition, vol. 168, p. 111788, May 2025, doi: 10.1016/j.patcog.2025.111788.

[22] T. Gebhart and P. Schrater, "Adversary detection in neural networks via persistent homology," arXiv (Cornell University), Nov. 2017, doi: 10.48550/arxiv.1711.10056.

[23] M. Goibert, T. Ricatte, and E. Dohmatob, "An adversarial robustness perspective on the topology of neural networks," arXiv (Cornell University), Nov. 2022, doi: 10.48550/arxiv.2211.02675.