Topological Safeguard for Evasion Attack based on the Interpretability of Artificial Neural Network Behavior (Extended Abstract)

Xabier Echeberria-Barrio
Vicomtech Foundation, Basque Research and Technology Alliance (BRTA)
Donostia-San Sebastian, 20009, Spain
xetxeberria@vicomtech.org

***Resumen*—This is an extended abstract of the publication [1]. Deep Learning technology is gaining importance and influence across various fields, including critical areas like healthcare and autonomous vehicles, where decision-making errors can have severe consequences. Concurrently, its widespread adoption has raised new cybersecurity threats, specifically the evasion attack, where an attacker adds specially designed, often imperceptible noise to an input sample, modifying the model's prediction. While researchers have developed various defenses, no single defense is perfect against all known evasion algorithms (such as FGSM, BIM, and PGD). This work introduces a novel evasion attack detector focusing on two key aspects: the information contained in neuron activations and the topology of the targeted Deep Learning model. The detector is motivated by findings that the model's topological structure contains essential information regarding whether an evasion attack is occurring, as these attacks widely alter the previously traversed prediction pathway. A critical finding prior to this work was that no literature analyzed activations while simultaneously maintaining their topological relations for evasion attack detection.**

***Index Terms*—Artificial neural network, Interpretability, Cybersecurity, Adversarial learning, Evasion attack**

I. Introduction

Deep Learning (DL) technology is increasingly deployed across various fields where decision-making errors can have severe consequences. While DL offers significant advances, its widespread adoption has concurrently raised new cybersecurity threats and vulnerabilities.

A major concern is the evasion attack [2], a well-known vulnerability where an attacker adds a specifically designed, often imperceptible noise to an input sample, modifying the model's prediction. Since the introduction of the L-BFG algorithm, researchers have been actively developing defenses, though no single defense is perfect against all known evasion algorithms (such as FGSM, BIM, PGD, and Carlini–Wagner).

This work introduces a novel evasion attack detector focusing on two key aspects: the information contained in neuron activations and the topology of the targeted Deep Learning model. The detector is motivated by findings that the model's topological structure contains essential information regarding whether an evasion attack is occurring. Disturbances caused by evasion attacks widely alter the previously traversed prediction pathway within the model.

The proposed detector uses Graph Convolutional Neural Network (GCN) technology to understand and leverage the target model's topology. To manage the computational complexity associated with modern large models, this approach focuses specifically on analyzing the classifier block (ϕ_C) of the targeted deep learning model, rather than the entire network. The goal is to improve detection rates found in the literature and offer a new methodology in this field.

II. Literature Review

Defenses against evasion attacks often fall into two categories: hardening the model (e.g., adversarial training [6] or dimensional reduction methods like PCA/Autoencoders [5]), or using an external detector (an auxiliary model). External detectors typically use a second classifier to distinguish malicious inputs. While successful, these detectors can sometimes be evaded simultaneously with the main classifier. Similar detector approaches analyze activation values or introduce new branches in the main network [3], [4]. Crucially, the sources emphasize that, prior to this work, no literature was found that analyzed activations while simultaneously maintaining their topological relations for evasion attack detection.

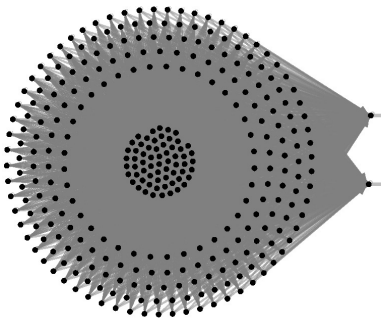


Figura 1: A classifier's behavior graph example, where the central cluster corresponds to the input layer of the classifier, the surrounding nodes are the neurons of the hidden layer, and the last right nodes correspond to the output layer. Notice that the neurons without any connection do not appear.

III. Method

The target deep learning model (ϕ) is conceptualized as having a feature-extractor block (ϕ_F) and a dense classifier block (ϕ_C). The detector operates on ϕ_C, analyzing features computed from its activations when an input image x_i is processed.

III-A. The Behavior Graph

When an input x_i is fed into the classifier ϕ_C, it generates a set of activations. These activations and the neurons of the classifier define a graph representation called the Behavior Graph ($G^{x_i}_{\phi_C}$).

The Behavior Graph is a Weighted Digraph:

- Nodes (N_{ϕ_C}): The neurons of the classifier ϕ_C.
- Weighted Edges ($A^{x_i}_{\phi_C}$): The activations that connect two neurons, defining the influence between them for input x_i.

This graph structure captures the decision-making process, showing how neurons are activating and their relative influence, thus reflecting the targeted model's behavior.

III-B. Classifier Node Attributes

To enrich the topological data for the GCN detector, several neuron attributes are computed based on the behavior graph. These attributes are normalized by layer to ensure comparability across layers with different activation functions.

III-B1. Impact ($I^x_{\phi_C}$): This modified attribute compares a neuron's normalized output activation with the sum of its normalized input activations. It reveals how much a neuron modifies the received values (positively or negatively) during the prediction of x_i. It is a measure of the neuron's influence on the flow of information.

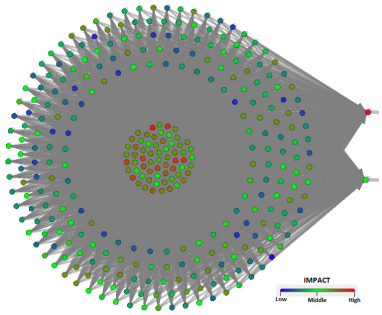


Figura 2: Classifier Behavior Graph example, where the nodes are coloring according to their impact attribute.

III-B2. Influence ($\Xi^x_{\phi_C}$): This attribute highlights the neurons per layer that participate in the prediction with the highest activation values. It is calculated by normalizing activations by layer and then setting a binary value (1 or 0) based on a percentile parameter (p), identifying the most influential neurons in the classification.

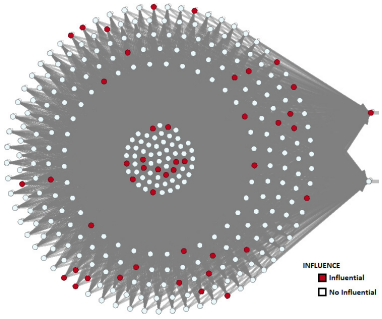


Figura 3: Classifier Behavior Graph example, where the nodes are coloring according to their influence attribute.

III-B3. Input Proportion ($P^x_{\phi_C}$): Proposed specifically in this work, this attribute computes the percentage of non-null input values a neuron receives relative to all possible inputs. This is particularly useful when the classifier uses activation functions (like ReLU) that map values to zero. Since the classifier in the experiment is a fully connected model, all neurons in a specific layer receive the same input proportion value.

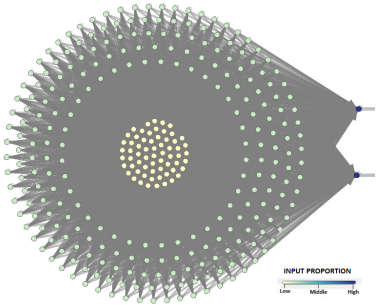
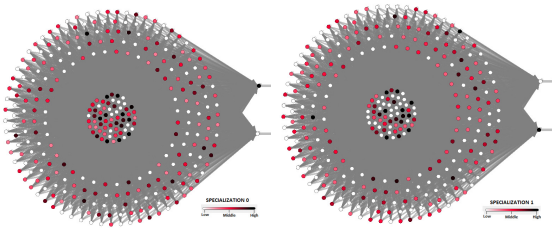


Figura 4: Classifier Behavior Graph example, where the nodes are coloring according to their input proportion attribute.

III-B4. Specialization ($Z^{x,\mathfrak{c}_i,k}_{\phi_C}$): This attribute quantifies a neuron's frequency of participation in predicting a specific class c. It measures how often a neuron's activation is among the k-top values when images of class c are processed.



(a) $0-$specialization (b) $1-$specialization

Figura 5: Classifier Behavior Graph example, where the nodes are coloring according to their specialization attributes.

III-C. Data Preprocessing and Detector Architecture

The preprocessing phase takes a set of adversarial and original images X and the target model ϕ to generate the enriched dataset for the detector. For each image x in X, the process generates:

- The Adjacency Matrix ($Ad^x_{\phi_C}$), representing the topology of ϕ_C.
- The set of all computed node attributes

$$\Delta^{x,p,k,t}_{\phi_C} = (I^x_{\phi_C}, \Xi^x_{\phi_C}, P^x_{\phi_C}, Z^{x,c_i,k}_{\phi_C}) \tag{1}$$

- A binary label

$$\lambda^x_{\phi_C} = \begin{cases} 1, & \text{if is an adversarial example} \\ 0, & \text{if is not an adversarial example.} \end{cases}$$

, indicating whether the image is adversarial (1) or not (0).

-

GCN Technology: Graph Neural Network (GNN) technology, specifically Graph Convolutional Networks (GCNs), was

chosen for the detector because of its ability to collect and process topological information through the graph structure. The selected architecture uses a sequence of GCN layers (with hyperbolic tangent activation) followed by one-dimensional convolution layers and a final classifier block.

IV. Experimental Setup and Scenario

IV-A. Scenario Details

The experiment uses a target deep learning model comprising a VGG16 network as the feature-extractor (ϕ_F) and a dense neural network as the classifier (ϕ_C). The classifier in this specific scenario uses the sigmoid activation function. The scenario employs the breast cancer dataset, which contains two image classes (non-cancer and cancer).

Adversarial examples were generated using three algorithms, all with the same perturbation budget ($\epsilon = 5/255$):

- Fast Gradient Sign Method (FGSM) (1 step).
- Basic Iterative Method (BIM) (10 steps).
- Projected Gradient Descent Method (PGD) (40 steps).

Four datasets were created: FGSM, BIM, PGD, and a combined Total dataset (53,908 total images, 26,954 adversarial). The parameter p for Influence was fixed at 0.5, and k for Specialization was fixed at 10.

IV-B. Attribute Analysis

An analysis of the generated node attributes showed that they were generally lowly correlated with each other and, importantly, none was directly correlated with the adversarial label. This confirms that a complex model, such as the GCN detector, is necessary to associate the attributes with the label.

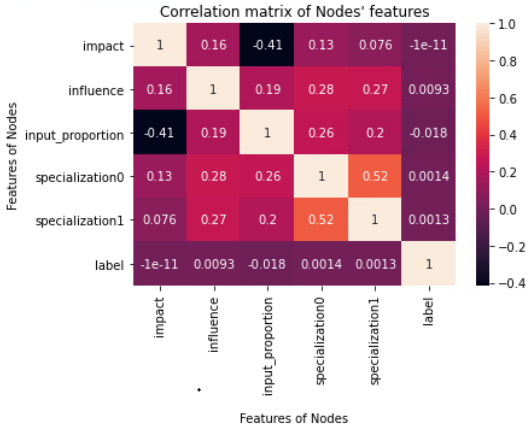


Figura 6: Correlation between the node features in the generated dataset

V. Results and Discussion

Four detectors were trained: one for each specific attack (FGSM, BIM, PGD) and one trained on the combined data (Total detector).

V-A. Detection Performance

The detectors achieved promising results in identifying adversarial examples (see Table I):

- BIM Detector: Highest accuracy at 0.969 (±0.0075).
- PGD Detector: High accuracy at 0.9573 (±0.0037).
- Total Detector (Generalized): Accuracy of 0.9219 (±0.0057), demonstrating effectiveness regardless of the attack type.
- FGSM Detector: Lowest accuracy at 0.8832 (±0.0023).

When the target model is defended by the proposed detector, the original success rate of the evasion algorithms drops significantly. For example, the success rate of the BIM attack fell from 0.9378 (original model) to 0.0290 (defended model).

Tabla I: The results obtained from the developed detectors

Metrics	FGSM detector	BIM detector	PGD detector	Total detector
Loss	$0{,}3623 \pm 0{,}0052$	$0{,}1394 \pm 0{,}0204$	$0{,}1852 \pm 0{,}0060$	$0{,}2566 \pm 0{,}0057$
Accuracy	$0{,}8832 \pm 0{,}0023$	$0{,}969 \pm 0{,}0075$	$0{,}9573 \pm 0{,}0037$	$0{,}9219 \pm 0{,}0057$

V-B. Validation of Topological Information

To validate the core hypothesis regarding the importance of topology, the GCN detector was compared against a basic MLP detector lacking the ability to capture topological information (see Table II).

The GCN detector performed considerably better in all cases:

- Total Detector (GCN): 0.9219 accuracy.
- MLP Detector: 0.8465 accuracy.

This outcome supports the investigation's original idea that topological information is crucial for understanding and detecting anomalies in deep learning models.

Tabla II: The detectors were trained with each dataset obtaining different results. The Total column shows the results obtained by training the detector with the total dataset.

Detectors	FGSD attack	BIM attack	PGD attack	Total
Total detector (GCN)	0,8832	0,969	0,9573	0,9219
MLP detector	0,8140	0,9170	0,8930	0,8465

V-C. Cross-Analysis and Transferability

Evaluation across different datasets showed the detector's ability to transfer detection patterns between adversarial types. Detectors trained on BIM or PGD attacks performed well when tested against other attack types. For instance, the BIM detector achieved high accuracy against the PGD test data (0.9637), and the PGD detector performed well against BIM test data (0.9501) [see Table III]. This cross-analysis suggests that training a single, generalized detector (like the Total detector) may not be strictly necessary, as detectors trained on BIM or PGD attacks can cover those threats effectively.

Tabla III: The results obtained from the combination of the trained detectors and the different datasets.

	FGSM test	BIM test	PGD test
FGSM detector	0,8832	0,7097	0,6923
BIM detector	0,8704	0,969	0,9637
PGD detector	0,8645	0,9501	0,9573
Total detector	0,8435	0,9614	0,9524

V-D. Attribute Contribution Analysis

Evaluating the contribution of each defined attribute (Impact, Influence, Input Proportion, Specialization) showed varying levels of importance (see Table IV).

- Impact and Influence attributes provided the most information for detection.
- Impact was most informative for BIM and PGD detection.
- Influence contained the most relevant information for the FGSM threat.
- Input Proportion and Specialization performed worse when used alone. The lower utility of Input Proportion in this scenario is likely due to the use of the sigmoid activation function in the classifier, which rarely returns null values, thus reducing the information this attribute can provide.

Tabla IV: The results are obtained in the total dataset from the attribute detectors. The first column shows the results of the detectors in the total dataset. The rest columns detail the accuracy of each detector in the different attacks.

Detectors	(Accuracy ± std, Loss ± std)	FGSD attack	BIM attack	PGD attack
Impact detector	(0,9047 ± 0,0081, 0,3104 ± 0,0239)	0,8053	0,9560	0,9502
Influence detector	(0,9131 ± 0,0075, 0,2865 ± 0,0215)	0,8493	0,9301	0,9276
Proportion detector	(0,8531 ± 0,0060, 0,4132 ± 0,0133)	0,8148	0,8664	0,8636
Specialization detector	(0,8598 ± 0,0063, 0,4159 ± 0,0134)	0,8182	0,8891	0,8714
Total detector	(0,9219 ± 0,0057, 0,2566 ± 0,0057)	0,8435	0,9614	0,9524

V-E. Ablation Analysis

A subsequent ablation analysis confirmed that all attributes provided extra, complementary information to the detection process, as the Total detector achieved the best generalized results (see Table V).

Tabla V: The results are obtained in the total dataset from the non-attribute detectors. The first column shows the results of the detectors in the total dataset. The rest columns detail the accuracy of each detector in the different attacks.

Detectors	(Accuracy ± std, Loss ± std)	FGSD attack	BIM attack	PGD attack
Non-impact detector	(0,9101 ± 0,0077, 0,2978 ± 0,0109)	0,8437	0,9453	0,9285
Non-influence detector	(0,9125 ± 0,0031, 0,2794 ± 0,0021)	0,8345	0,9560	0,9394
Non-proportion detector	(0,9057 ± 0,0055, 0,3056 ± 0,0127)	0,8472	0,9490	0,9425
Non-specialization detector	(0,9192 ± 0,0013, 0,2659 ± 0,0056)	0,8443	0,9613	0,9520
Total detector	(0,9219 ± 0,0057, 0,2566 ± 0,0057)	0,8435	0,9614	0,9524

V-F. Comparison with Literature

Auxiliary Model Comparison: The proposed detector achieved superior detection rates compared to well-known auxiliary detectors (LID, NSS, and KD+BU) found in the literature for this specific scenario [97, 98, Table 12]. For instance, against the BIM attack, the proposed detector achieved 0.969 accuracy, significantly improving upon the NSS detector (0.8028). Theoretical Comparison to Similar Methods: A theoretical comparison was made with the work of Pawlicki et al., which also analyzes activation values. Their approach treats activations as a single, long 1D vector, ignoring topology and considering all activations of the targeted model. This novel detector improves on that by focusing only on the classifier block for scalability and crucially, by integrating the target model's topology as an essential feature for detection.

VI. Conclusion and Future Work

This work successfully developed a novel evasion attack detector that achieves outstanding results by incorporating and leveraging the topological information of the targeted neural network's classifier block, utilizing Graph Convolutional Networks. The introduction of four novel or adapted node attributes—Impact, Influence, Input Proportion, and Specialization—enriches the dataset, providing the GCN with detailed information about the model's behavior. The analysis supports the importance of topology in understanding and detecting model perturbations.

Future research aims to refine the system by:

1. Testing different GCN architectures and optimizing hyperparameters.
2. Using interpretability techniques on the trained detector to identify vulnerable neurons, which could lead to the generation of local defenses.
3. Extending the analysis to the highly complex feature extractor block (ϕ_F) of the model. This extension would require implementing simplification techniques, such as aggregations or graph splitting, to manage the immense number of parameters involved.

Aknowledgements

This work was partially supported by the European Commission under the Horizon Europe Programme as part of the FALCON and TESTUDO projects (Grant Agreements No. 101121281 and No. 101121258)

Referencias

[1] Echeberria-Barrio, X., Gil-Lerchundi, A., Mendialdua, I., Orduna-Urrutia, R.: "Topological safeguard for evasion attack interpreting the neural networks' behavior", en *Pattern Recognition*, 147, 110130, 2024.

[2] Jiang, W., Li, H., Liu, S., Luo, X., Lu, R.: "Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles". *IEEE transactions on vehicular technology*, 69(4), 4439-4449, 2020.

[3] Pawlicki, M., Choraś, M., Kozik, R.: "Defending network intrusion detection systems against adversarial evasion attacks". *Future Generation Computer Systems*, 110, 148-154, 2020.

[4] Metzen, J. H., Genewein, T., Fischer, V., Bischoff, B.: .on detecting adversarial perturbations". *arXiv preprint arXiv:1702.04267*, 2017.

[5] Sahay, Rajeev and Mahfuz, Rehana and El Gamal, Aly: Çombatting adversarial attacks through denoising and dimensionality reduction: A cascaded autoencoder approach". *53rd Annual conference on information sciences and systems (CISS)*, 2(3), 1–6, 2019.

[6] Yu, X., Smedemark-Margulies, N., Aeron, S., Koike-Akino, T., Moulin, P., Brand, M., ... Wang, Y.: İmproving adversarial robustness by learning shared information". *Pattern Recognition*, 134, 109054, 2023.