vicomtech
MEMBER OF BASQUE RESEARCH & TECHNOLOGY ALLIANCE

Topological Safeguard for Evasion Attack Interpreting the Neural Networks' Behavior

Paris, 17 Sept. 2025

Xabier Echeberria Barrio
Digital Security Researcher

xecheberria@vicomtech.org

www.vicomtech.org

Information

- **Title:** Topological Safeguard for Evasion Attack Interpreting the Neural Networks' Behavior
- **Authors:** Xabier Echeberria-Barrio, Amaia Gil-Lerchundi, Iñigo Mendialdua, Raul Orduna-Urrutia
- **Affiliations:** Vicomtech Foundation, BRTA, and Department of Computer Languages and Systems, University of the Basque Country (UPV/EHU)
- **Journal:** Pattern Recognition
- **Date:** March 2024

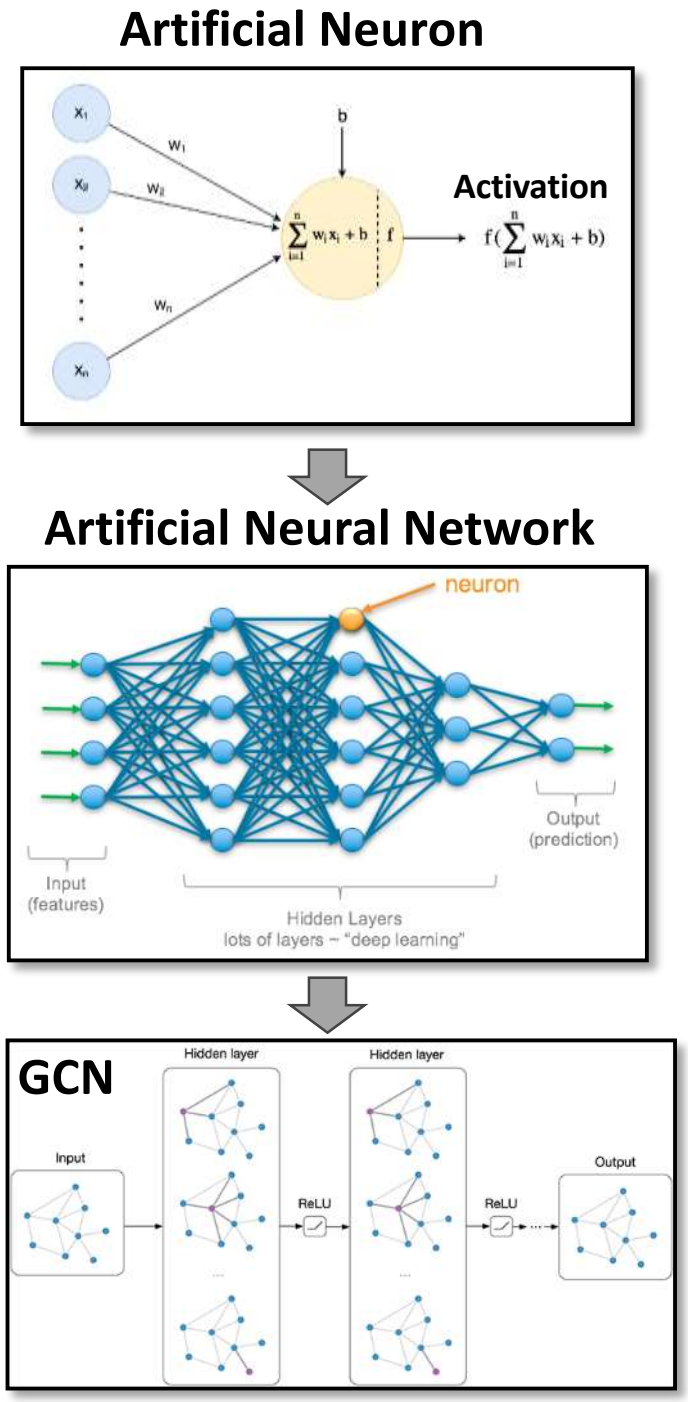
Echeberria-Barrio, X., Gil-Lerchundi, A., Mendialdua, I., & Orduna-Urrutia, R. (2024). Topological safeguard for evasion attack interpreting the neural networks' behavior. *Pattern Recognition*, *147*, 110130.

Overview

- **Problem:** Deep Learning models introduce cybersecurity vulnerabilities, especially evasion attacks that modify decision-making. Existing defenses are not perfect.

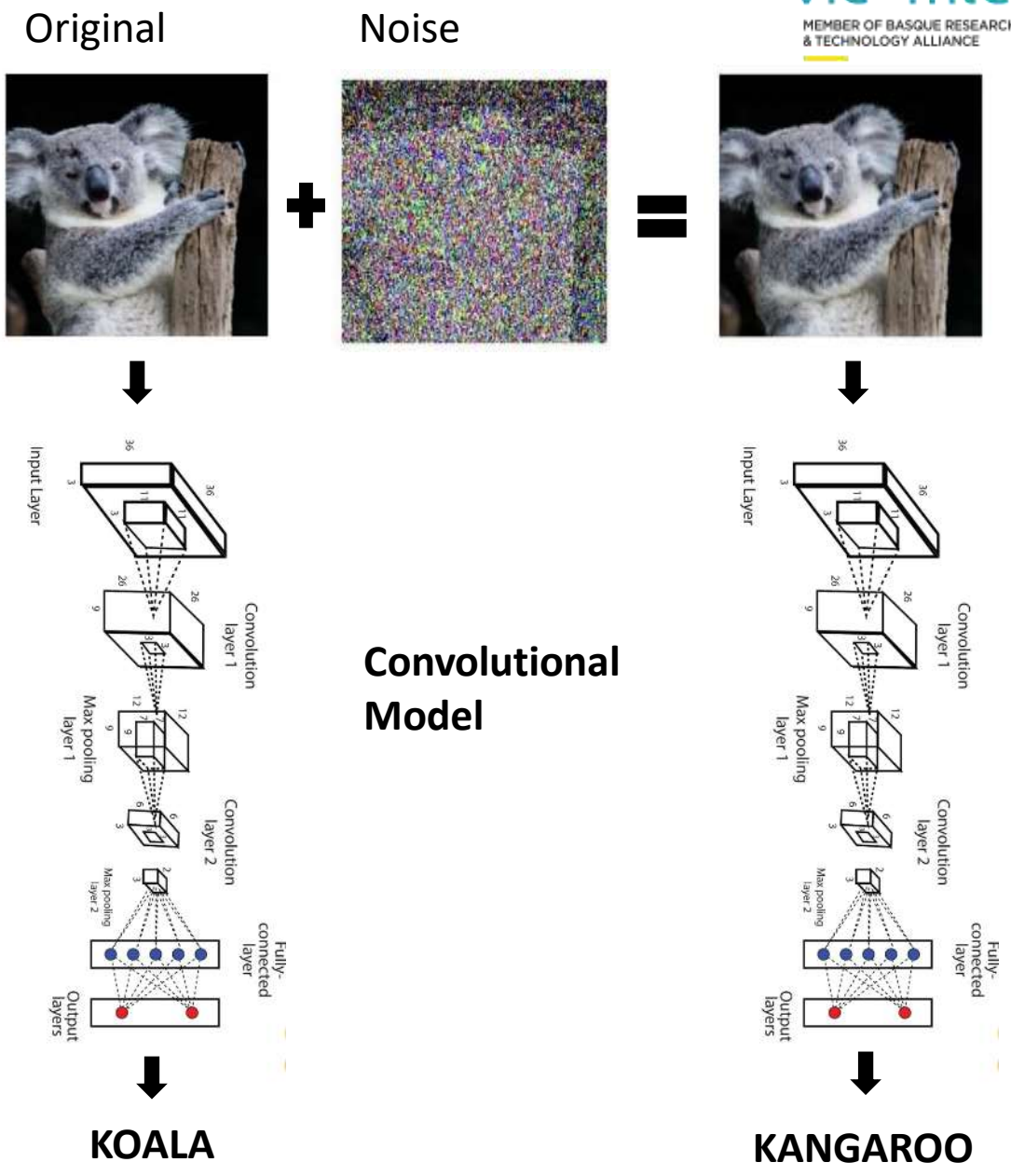
- **Our Solution:** A **novel detector of evasion attacks** that:
  - Analyzes **activations of neurons** when an input sample is injected.
  - Pays attention to the **topology of the targeted deep learning model** (how neurons connect).
  - Uses **Graph Convolutional Neural Network (GCN)** technology to understand model topology.

- **Key Outcomes:** Achieves **promising results**, improving detection rates compared to similar defenses in the literature. This approach also offers a new way to develop such detectors.



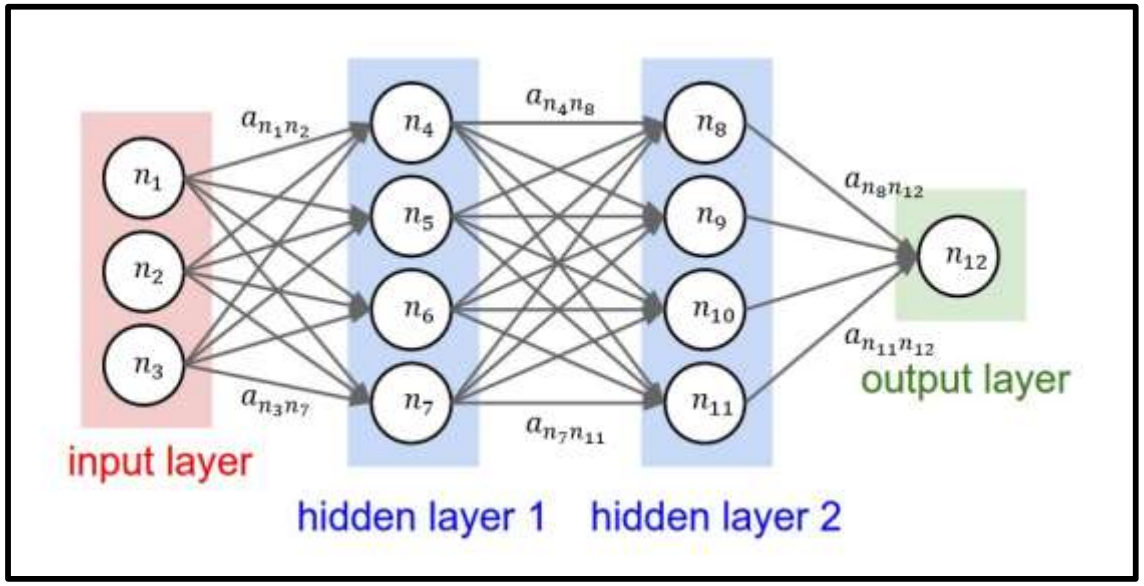
Motivation

- **Deep Learning (DL) Advancements:** Rapid adoption in critical fields like healthcare and autonomous vehicles.
- **New Threats:** These advancements come with significant cybersecurity vulnerabilities, including potential for data leakage and manipulation of model decisions.
- **Focus:** Evasion Attacks
  - A worrisome threat where attackers add **imperceptible noise** to an input sample.
  - This noise **modifies the original output prediction** to cause misclassification.
  - **Examples of Evasion Algorithms:** L-BFGS, Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), Projected Gradient Descent (PGD), and Carlini-Wagner Method.
- **The Challenge:** Despite ongoing research, no perfect defense exists for all known evasion algorithms. Researchers are continuously developing new countermeasures.

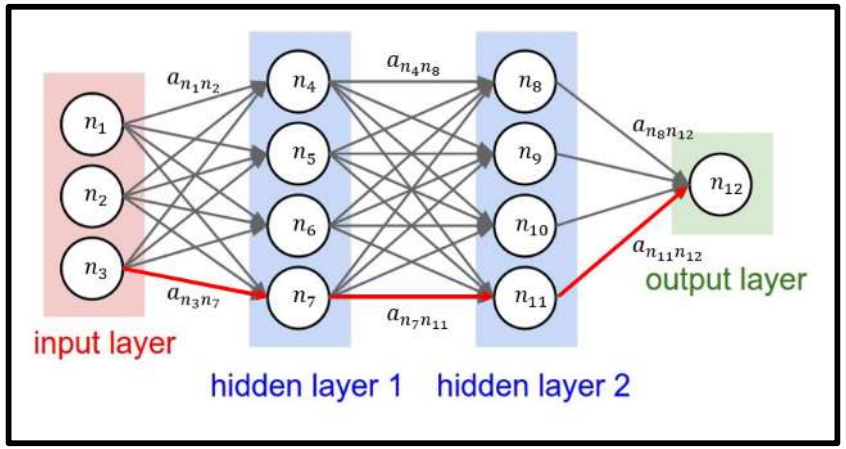


Why Topology Matters

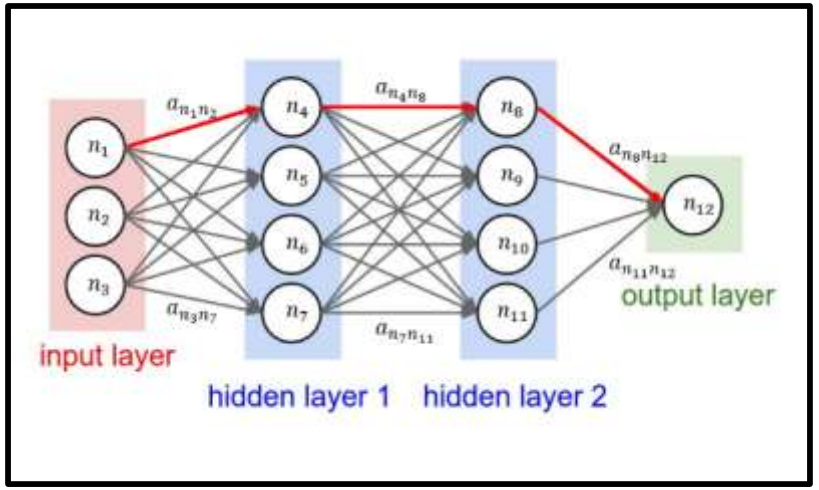
- Small perturbations from evasion attacks can **greatly alter neuron relationships and prediction pathways** within the model.
- Topological information is crucial for understanding the **model's behavior and detecting perturbations.**



SAMPLE 1

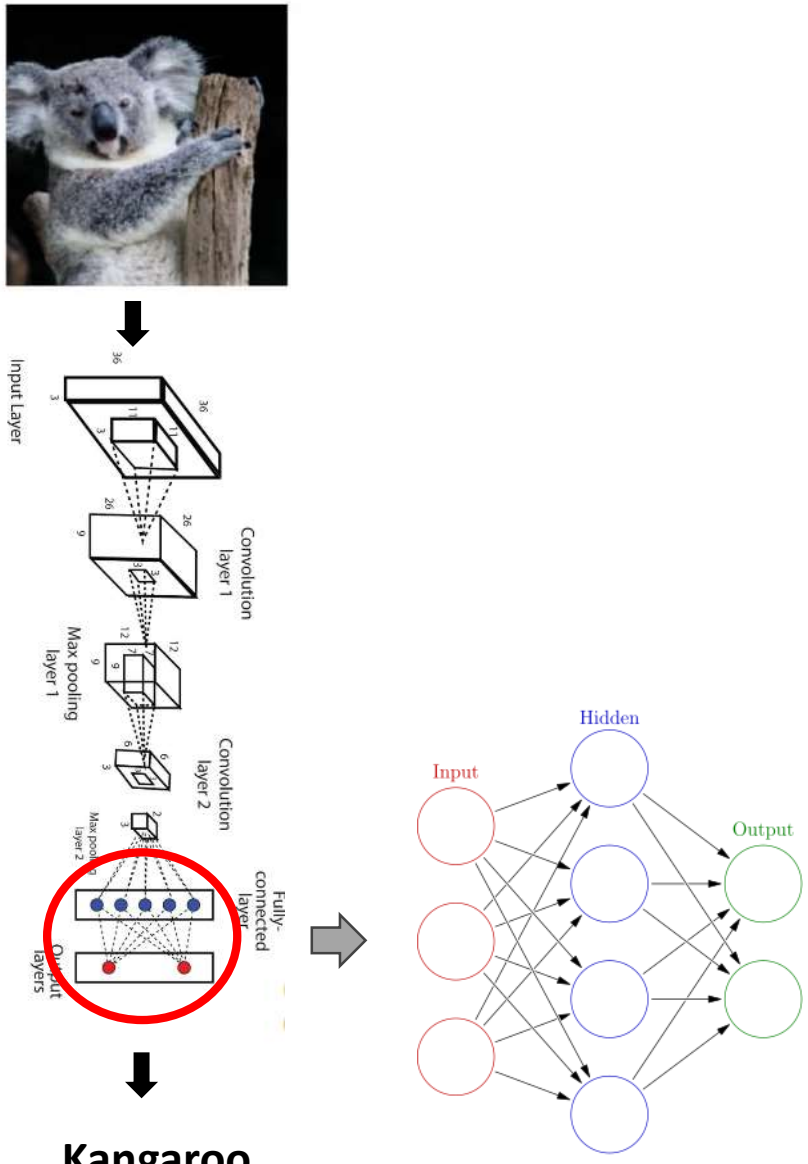


SAMPLE 2



Proposal

- **Core Idea:** Develop an evasion attack detector based on graph neural network technology.
- **Target:** Focuses on the classifier part of the deep learning model.
- **Process:**
  1. **Preprocessing:** Convert the classifier's behavior for an input into a **behavior graph**.
  2. **Attribute Extraction:** Compute novel neuron attributes that capture detailed behavioral and topological information.
  3. **GCN Detector:** A **Graph Convolutional Neural Network (GCN)** consumes these graphs and attributes to identify adversarial inputs.
- **Goals:**
  - Improve existing **detection rates**.
  - Introduce a **new way to develop detectors** in this field by explicitly leveraging topology.
  - Potentially provide **detailed information about vulnerable neurons** (future work)



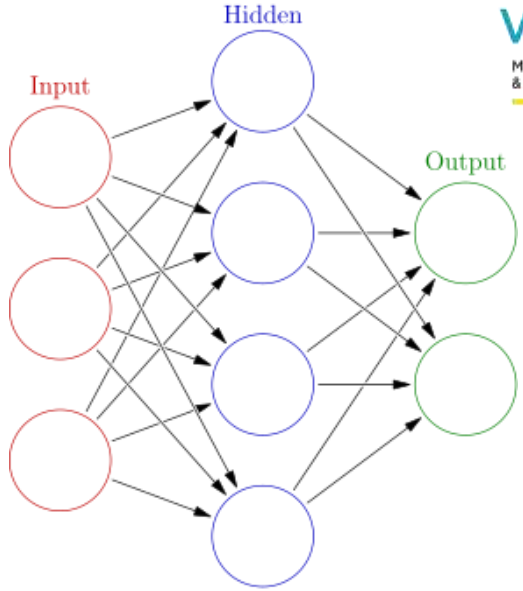
Preprocessing

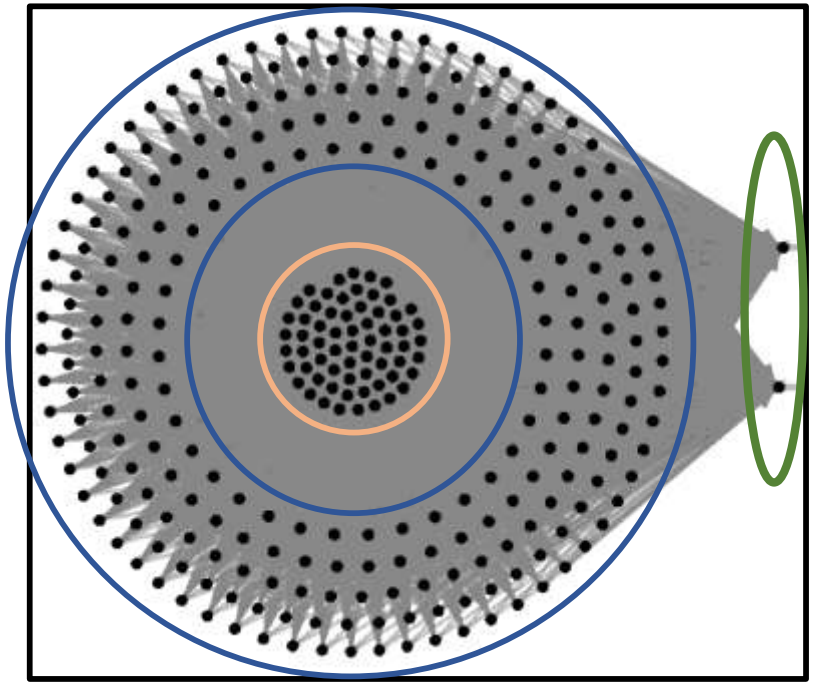
- **Definition:** A Weighted Digraph that represents the classifier's behavior for a given input image.
  - **Components:**
    - **Nodes:** The individual neurons of the classifier.
    - **Edges:** The connections between neurons, showing the flow of information.
    - **Weighted Edges:** The activation values of neurons, quantifying their influence or how strongly they fire.

- **Significance:** This graph precisely shows how neurons are activating and influencing decisions, providing a direct representation of the targeted model's behavior.

- **Flexibility:** Can be generated for any classifier architecture, regardless of the number of hidden layers.

- **Origin:** First conceptualized by Echeberria et al. in [1] to visualize model behavior.

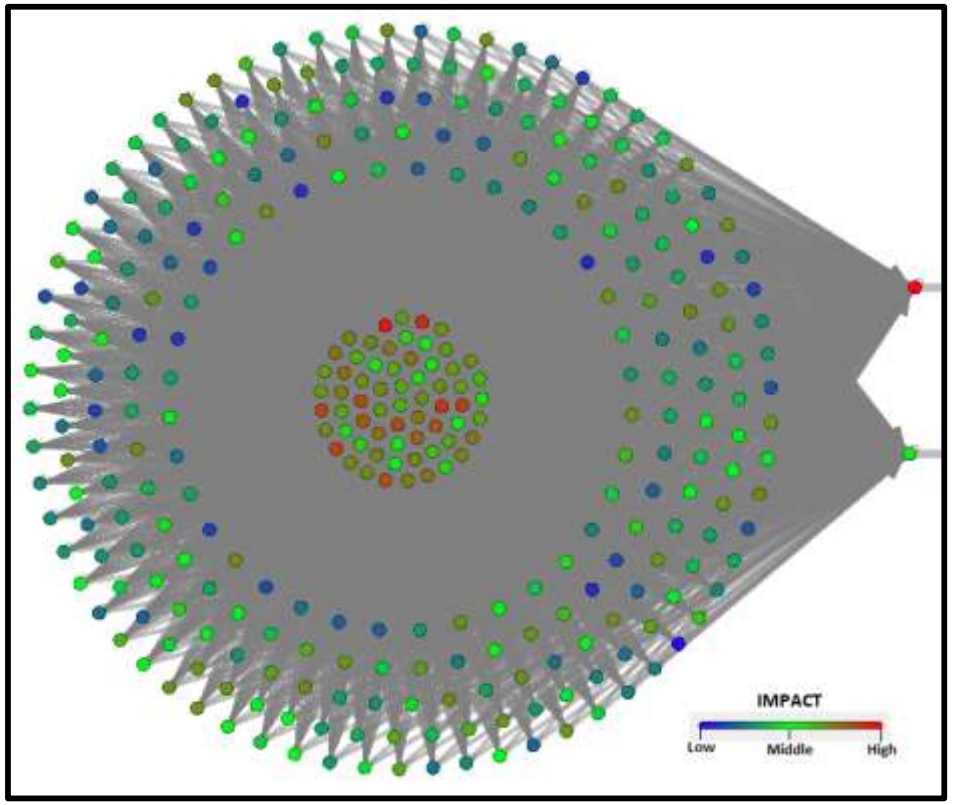




Attribute Extraction

Impact

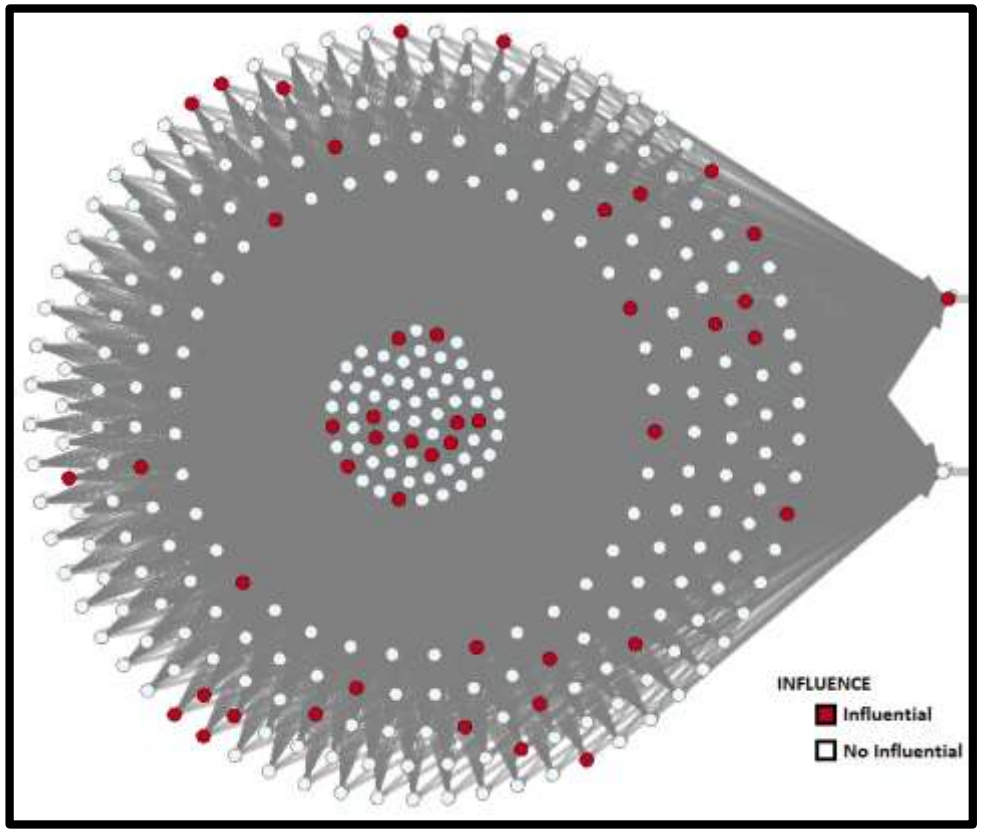
- **Definition:** Measures how a neuron modifies the values it receives to influence the prediction.
- **Modification:** We normalize activations by layer to address scale differences between various activation functions (e.g., ReLU vs. Sigmoid), an improvement over prior definitions.
- **Interpretation:**
  - **>> 0:** High positive modification
  - **> 0:** Slight positive modification
  - **= 0:** Keeps values equal
  - **< 0:** Slight negative modification
  - **<< 0:** High negative modification
- **Visualization:** Blue for negative impact, Red for positive, Green for null



Attribute Extraction

Influence

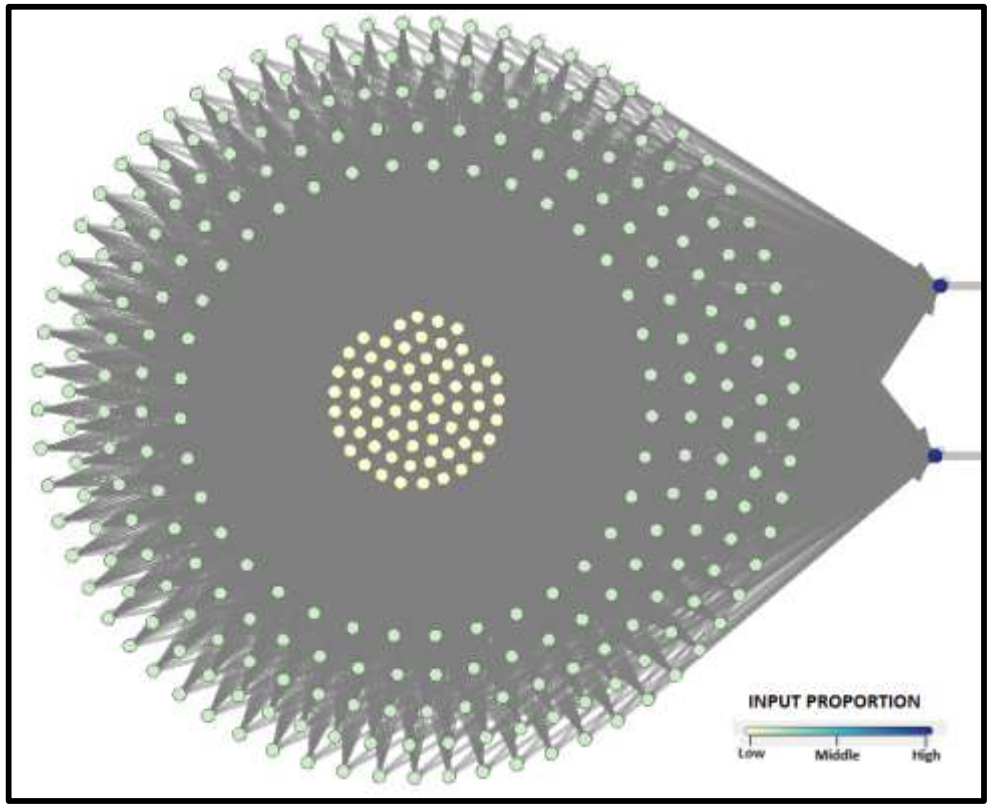
- **Definition:** Highlights neurons that participate in the prediction with the highest activation values within their layer.
- **Modification:** Adapted from Hohman et al. for dense neural networks, also uses layer-wise normalization.
- **Mechanism:** Sorts normalized activations by layer; assigns '1' to top neurons whose cumulative activation reaches a parameter p (e.g., 0.5), '0' otherwise.
- **Visualization:** Highlighted (red) for influential neurons.



Attribute Extraction

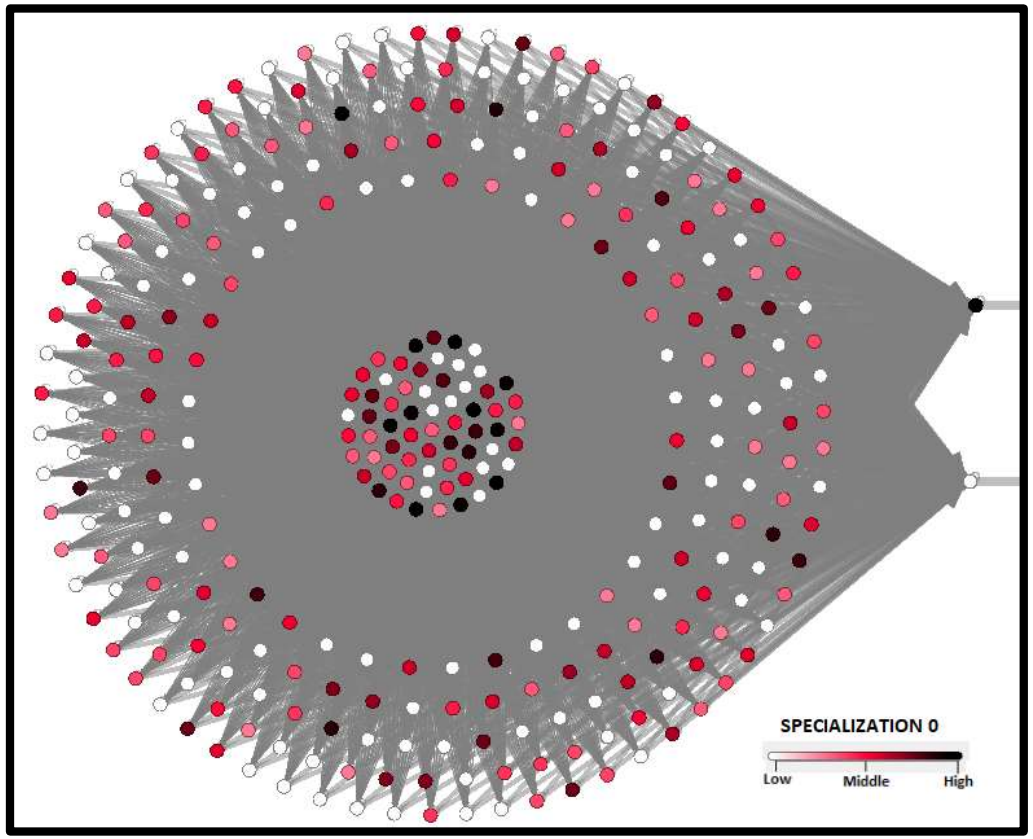
Input Proportion

- **Definition:** The percentage of non-null input values a neuron receives, relative to all possible inputs it could receive based on topology.
- **Information Content:** Highly dependent on the activation function of the targeted model.
- **ReLU:** Provides useful information as it maps values to zero.
- Sigmoid: Less informative as it rarely produces null values.
- **Relevance:** Helps detect adversarial examples by identifying "weird" null activation patterns (e.g., a "dog" image with cat-like null activations).
- **Visualization:** Colors nodes based on their input proportion



Attribute Extraction
Specialization

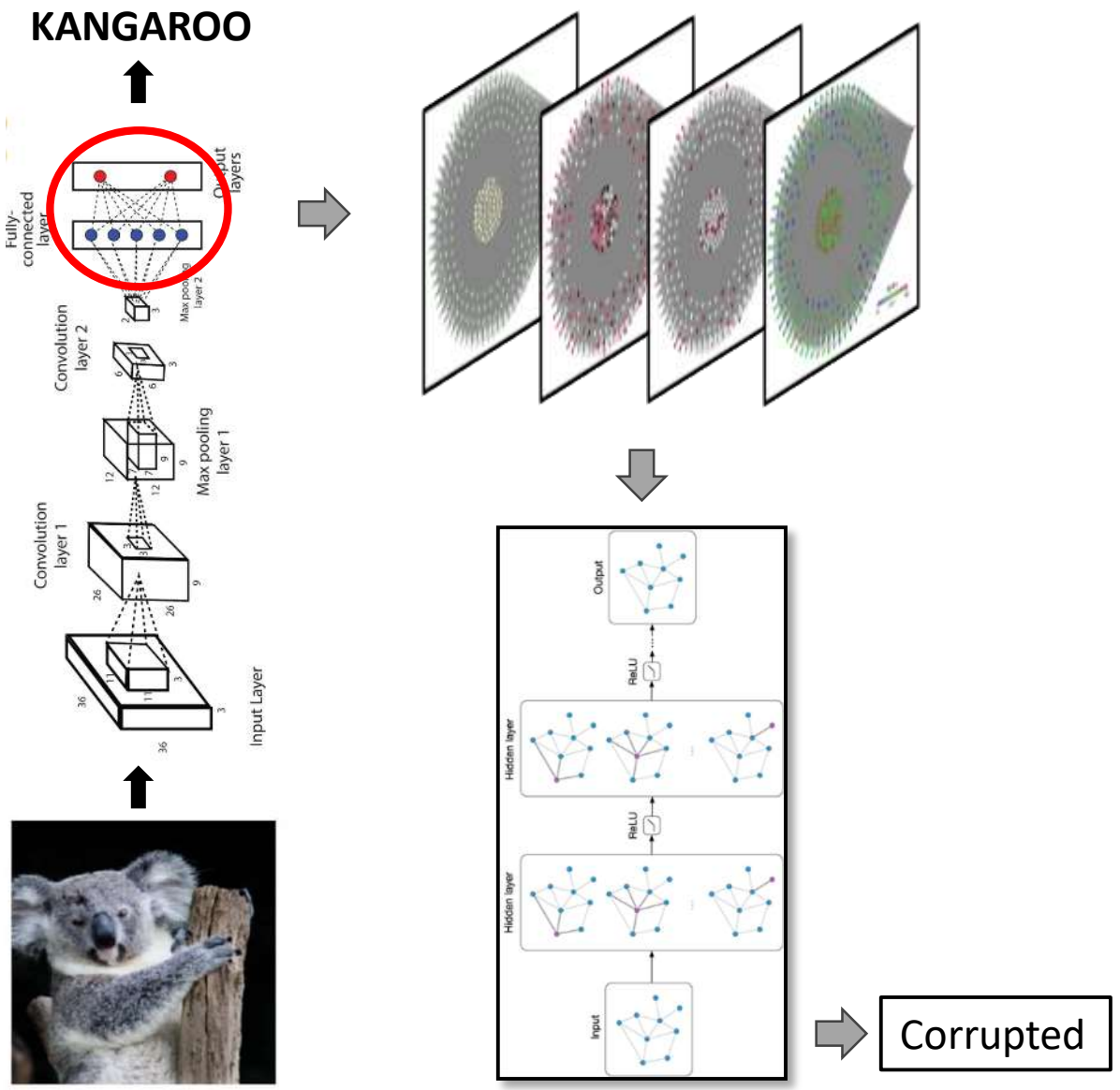
- **Definition:** Measures the frequency with which a neuron actively participates in predicting a concrete class (e.g., 'Class 0-specialization', 'Class 1-specialization').
- **Modification:** Instead of solely relying on null activation values, it considers if a neuron's activation is among the k-top activations within its layer. This is crucial for models with non-ReLU activation functions.
- **Mechanism:** For each class, it calculates how often a neuron's activation falls into the top-k activations across all images of that class.
- **Visualization:** A white-red-black scale to indicate low to high specialization



Detection Dataset

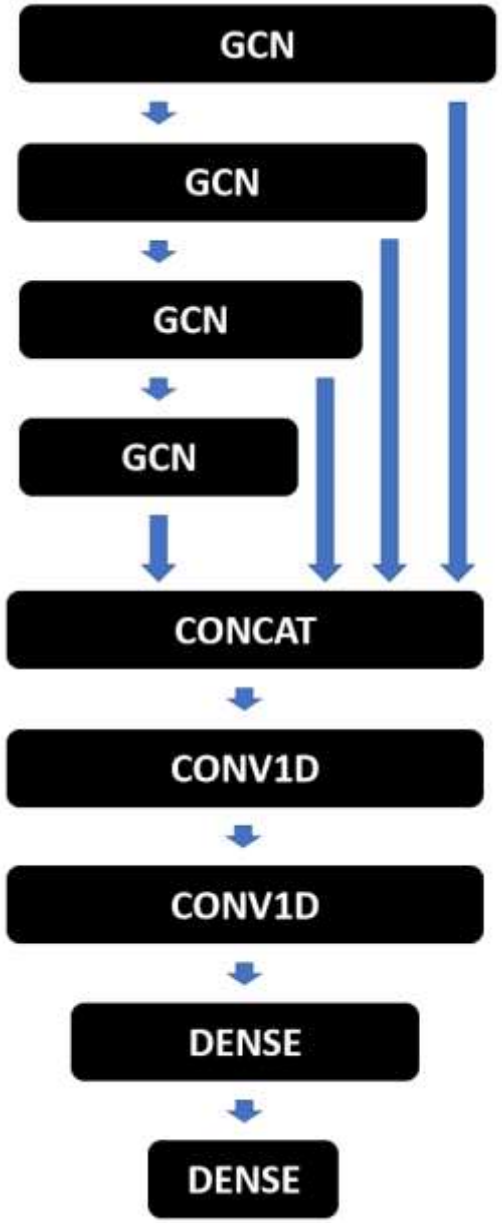
Data Preprocessing Pipeline (for each image):

1. **Behavior Graph Generation:** Compute behavior graph for the classifier.
2. **Adjacency Matrix:** Represents the connectivity of the graph.
3. **Attribute Computation:** Calculate Impact, Influence, Input Proportion, and t Specialization attributes for all neurons.
4. **Attribute Aggregation:** Group these into a feature vector per neuron.
5. **Labeling:** Assign a label (1 for adversarial, 0 for legitimate) for the detector's training



Detector Model

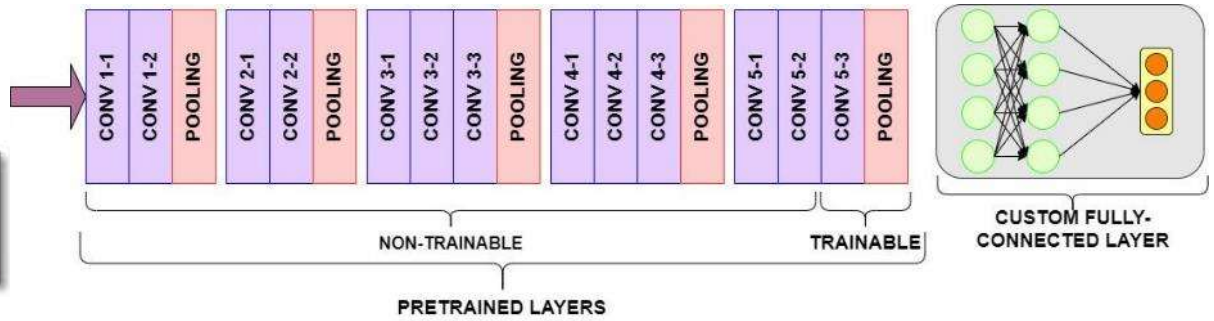
- **Rationale:** GCNs are ideal for leveraging **topological information** and the inherent structure-property relationship in deep learning models.
- **Architecture:** Adopted from a successful fake news detection model.
- **GCN Block:** Multiple GCN layers (hyperbolic tangent activation) to capture graph-level features.
- **Convolutional Block:** 1D convolution layers (ReLU activation) to process concatenated GCN outputs.
- **Classifier Block:** Dense layers (sigmoid activation) for final adversarial/legitimate classification



Experimental Setup

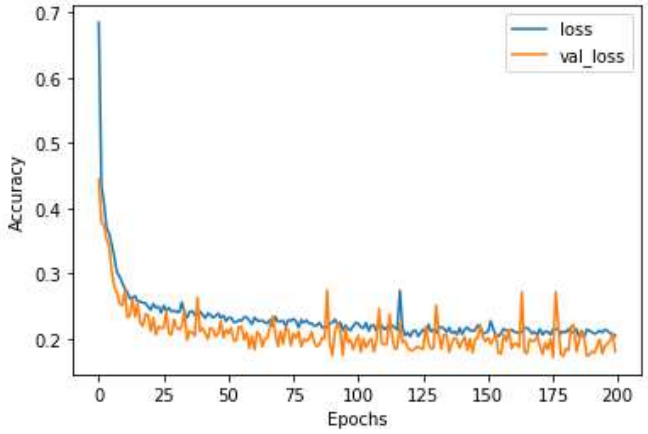
- **Target Model:**
  - **Feature Extractor**: VGG16 Neural Network.
  - **Classifier:** A dense neural network implemented with a Sigmoid activation function. (Note: Differs from which used ReLU).
- **Dataset: Breast Cancer Dataset**
  - **Two classes:** Non-cancer (Class 0) and Cancer (Class 1).
  - **Split:** 70% training, 30% test (stratified split to preserve class proportions).
- **Adversarial Examples Generation:**
  - **Algorithms:**
    - **Fast Gradient Sign Method (FGSM)**
    - **Basic Iterative Method (BIM)**
    - **Projected Gradient Descent (PGD)**
  - **Perturbation Budget:** $\epsilon = 5/255$ for all.
  - **Optimization Steps:** FGSM (1 step), BIM (10 steps), PGD (40 steps).

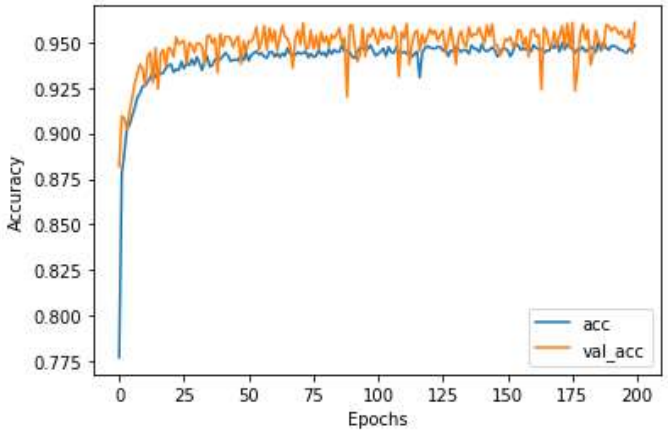
- **Generated Datasets for Detector:**
  - **FGSM Dataset**: (9,489 adversarial, 9,489 normal)
  - **BIM Dataset:** (9,378 adversarial, 9,378 normal)
  - **PGD Dataset:** (8,087 adversarial, 8,087 normal)
  - **Total Dataset:** (26,954 adversarial, 26,954 normal) - combined from all three.
- **Attribute Parameters:** Influence p = 0.5, Specialization k = 10, Number of classes t = 2.



Results

- **Trained Detectors:** FGSM, BIM, PGD, and Total (trained on respective datasets).
- **Accuracies:**
  - **BIM Detector: 96.90%** (Highest performance)
  - **PGD Detector: 95.73%**
  - **Total Detector: 92.19%**
  - **FGSM Detector: 88.32%** (Lowest, FGSM is a less refined attack)
- **Training Stability:** No significant overfitting observed; loss and accuracy curves converged well.
- **Impact on Attack Success Rate:**
  - **Original Model (No Defense):** FGSM (0.4744), BIM (0.9378), PGD (0.8087)**.**
  - **Defended Model (with our detector):**
    - **FGSM attack success reduced to 0.0554**
    - **BIM attack success reduced to 0.0290**
    - **PGD attack success reduced to 0.0345**
- **Conclusion:** Our detector significantly **reduces the success rate** of evasion attacks.





Results

Comparison with Literature

1. **Auxiliary Model Detectors:**
   - Compared our detector with leading auxiliary model detectors: **LID, NSS, and KD+BU.**
   - **Our detector significantly outperforms all of them:**
     - **FGSM: 0.8832 (Our)** vs. 0.8076 (NSS - 2nd best)
     - **PGD: 0.9573 (Our)** vs. 0.8142 (NSS - 2nd best)
     - **BIM: 0.969 (Our)** vs. 0.8028 (NSS - 2nd best)
   - **Performance Improvement:** Our detector shows **8-16% higher accuracy** compared to the second-best detector (NSS) across different attacks.

2. **Similar Method: Pawlicki et al.**
   - **Theoretical Comparison** (empirical difficult due to different data: network traffic vs. images).
   - **Pawlicki et al.:** Uses a long 1D vector of all activation values, without topological consideration.
   - **Our Work:**
     - **More scalable:** Focuses on the classifier block only.
     - **Topologically aware:** Explicitly considers and combines activation values with the model's topology, generating richer features..

Results
Topological Information

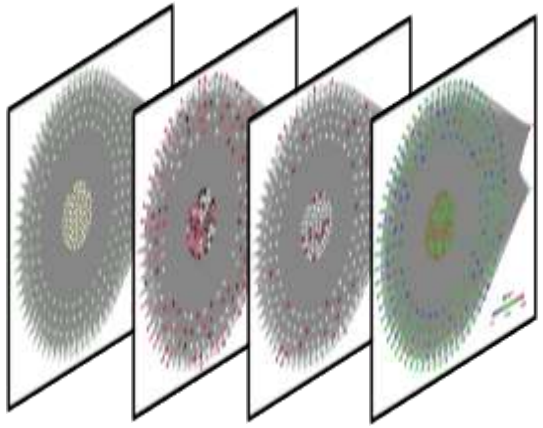
- **Hypothesis:** Topological information, captured by GCNs, is essential for effective detection.

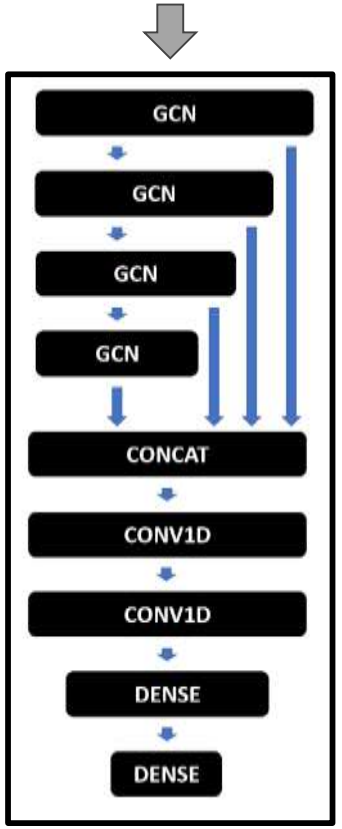
- **Comparison:** Our proposed GCN detector vs. a basic MLP detector.
  - **MLP Detector:** Used a single hidden layer with 200 neurons, mirroring the dense part of our GCN architecture, but without topological awareness.

- **Accuracies:**
  - **GCN Detector (Total):** 92.19%
  - **MLP Detector (Total):** 84.65%

- **GCN vs. MLP - Individual Attacks:**
  - **FGSM: GCN (0.8832)** vs. MLP (0.8140)
  - **BIM: GCN (0.969)** vs. MLP (0.9170)
  - **PGD: GCN (0.9573)** vs. MLP (0.8930)

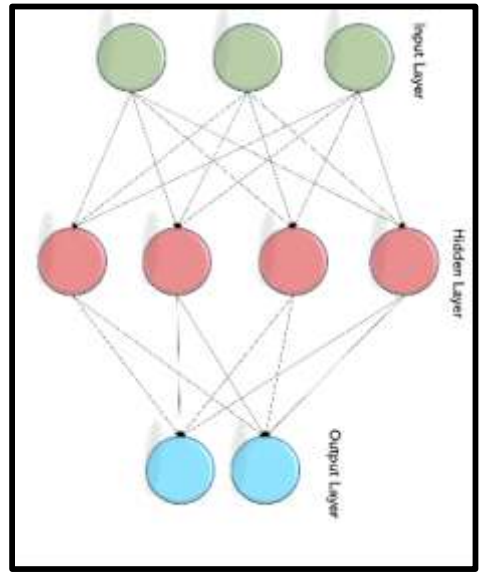
- **Conclusion:** The GCN detector performs considerably better in all cases, unequivocally supporting the importance of topology in understanding deep learning models and detecting attacks.





92.19%

$(a_{11} \quad a_{12} \quad \cdots \quad a_{nn})$

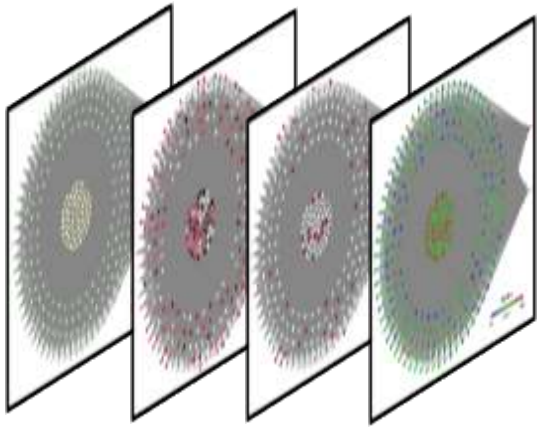


84.65%

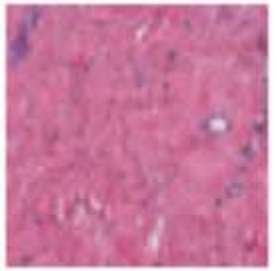
Results

Detector Transferability

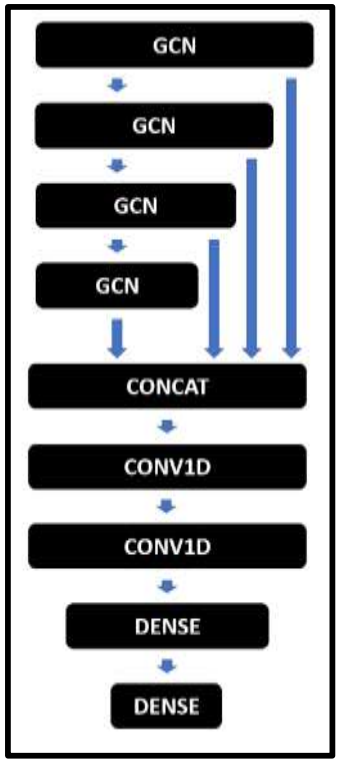
- **Question:** Can a detector trained on one attack type successfully detect others?
- **Experiment:** Each trained detector (FGSM, BIM, PGD, Total) evaluated against all adversarial datasets.
- **Key Findings:**
  - **FGSM detector:** Shows reduced performance on BIM (0.7097) and PGD (0.6923) datasets.
  - **BIM detector:** Performs outstandingly across all datasets (FGSM: 0.8704, BIM: 0.969, PGD: 0.9637).
  - **PGD detector:** Also performs well across all datasets (FGSM: 0.8645, BIM: 0.9501, PGD: 0.9573), though slightly less than BIM.
- **Conclusion:** A 'Total detector' is not always necessary. Detectors trained on BIM or PGD attacks demonstrate strong transferability, effectively covering other adversarial threats.
- **Insight:** The BIM detector consistently outperformed the PGD detector, even on PGD attacks. This suggests the BIM dataset might contain 'less noise' (fewer ineffective modifications) compared to PGD, allowing for better generalization



Activations obtained by modifying the input data with BIM



BIM Modified

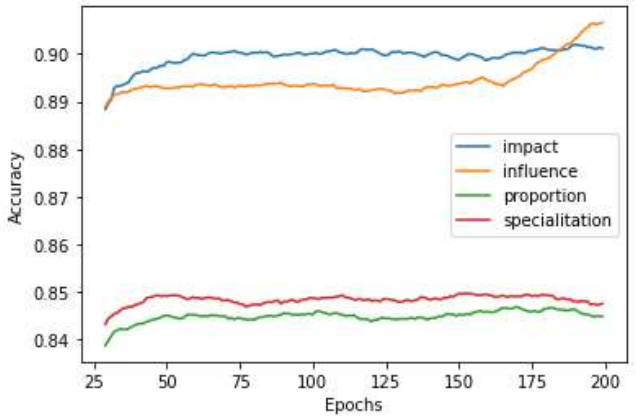


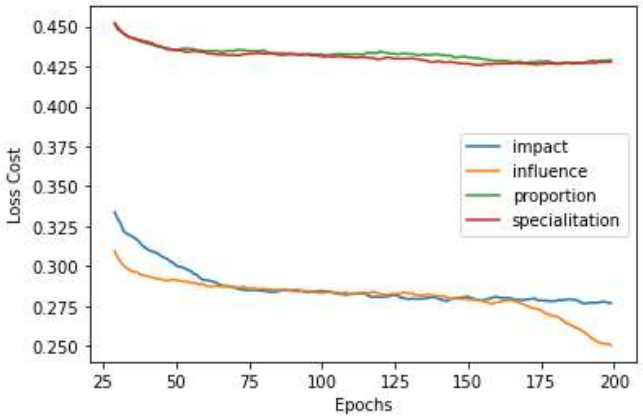
PGD detector

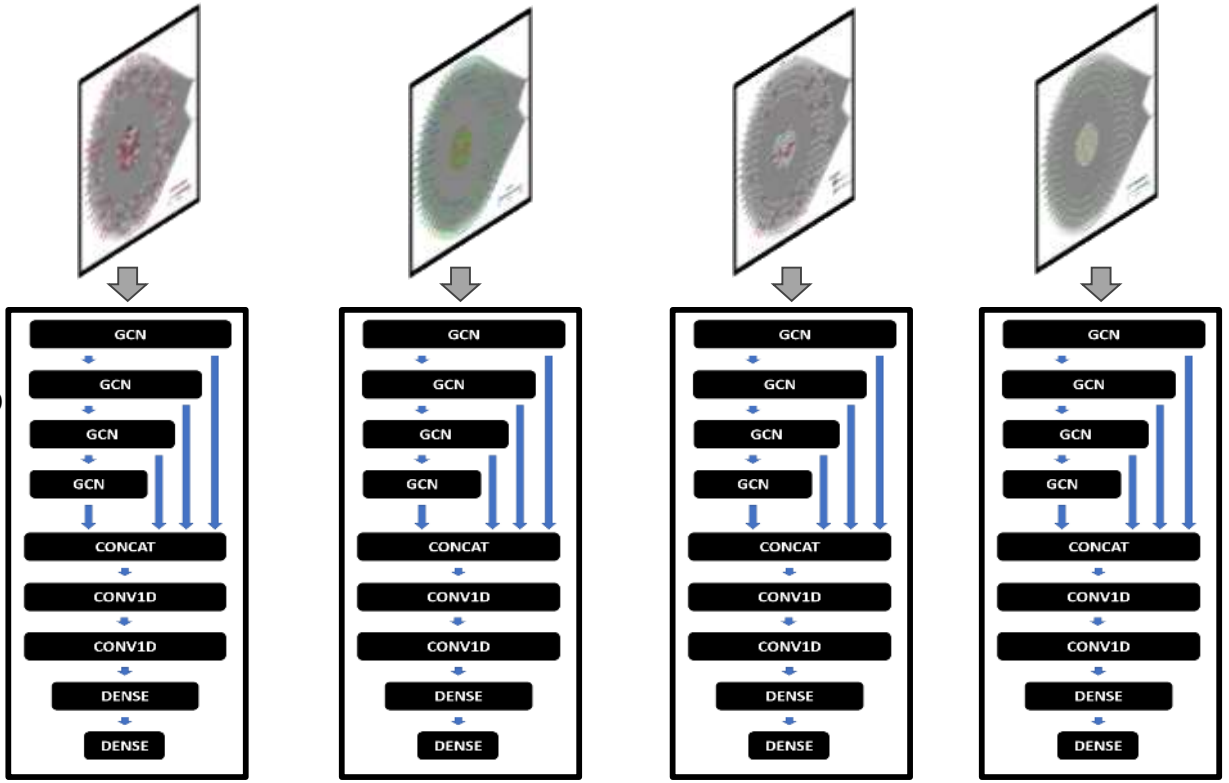
Results

Attribute Contribution

- **Experiment:** Detectors trained using **only one attribute** (Impact, Influence, Input Proportion, or Specialization) on the Total dataset.
- **Purpose:** To understand the individual contribution of each attribute to detection performance.
- **Accuracies:**
  - **Influence detector:** Highest overall accuracy (0.9131), most helpful for global detection and FGSD attacks**.**
  - **Impact detector:** High accuracy (0.9047), most informative for BIM and PGD attacks.
  - **Input Proportion detector:** Lowest accuracy (0.8531).
  - **Specialization detector:** Also lower (0.8598).
- **Note on Input Proportion:** Its low performance is likely due to the Sigmoid activation function in our target model, which rarely produces null values.
  - This attribute **could be more valuable** with activation functions that map values to zero (e.g., ReLU).
- **Conclusion:** While Influence is globally most informative, **Impact is crucial for BIM/PGD**, and all attributes potentially complement each other.

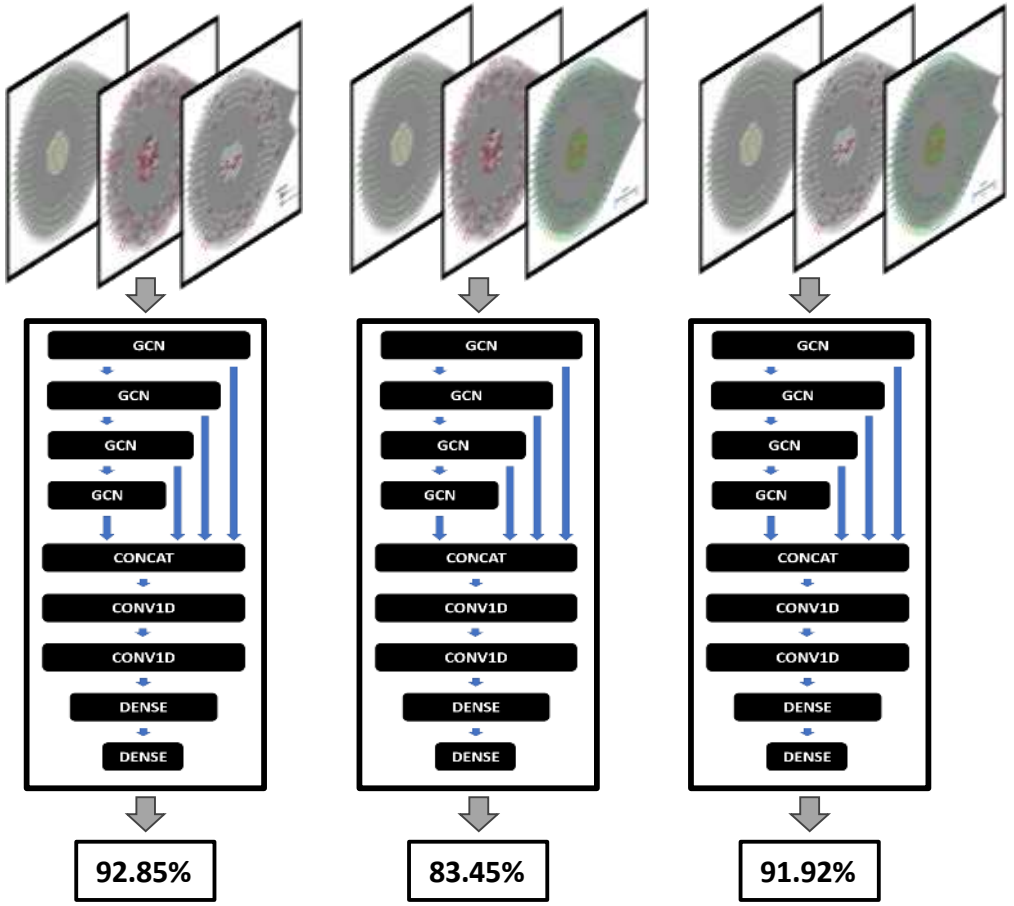






Results
Ablation Analysis

- **Experiment:** Detectors trained with all attributes EXCEPT one (Non-Impact, Non-Influence, Non-Proportion, Non-Specialization) on the Total dataset.
- **Purpose:** To identify the unique and non-redundant contribution of each attribute to the full model's performance.
- **Accuracies:**
- **Total detector (for reference):** 0.9219
    - **Non-Influence detector:** Shows the most significant reduction for **FGSM attack** detection (0.8345 vs. 0.8435 for Total), confirming Influence's unique role there.
    - **Non-Impact detector:** Leads to the most significant reductions for **BIM (0.9453 vs. 0.9614) and PGD (0.9285 vs. 0.9524) attacks**, highlighting Impact's critical role for these attacks.
    - **Non-Specialization detector:** Performance (0.9192) is very close to the Total detector, suggesting its information might largely be covered by other attributes in this scenario.
- **Overall:** All attributes generally provide **extra information**, though their specific impact varies by attack type.



Conclusions

- **Novel Detector:** The development of a **novel evasion attack detector** with a successful performance.
- **Key Innovation:** The integration of **topological information** (behavior graphs) into the detection process using **Graph Convolutional Networks** (GCNs).
- **Validation of Topology:** Our GCN detector consistently **outperformed a basic MLP detector**, confirming the crucial importance of model topology for attack detection.
- **Attribute Contributions:** All defined neuron attributes contribute, with **Influence** being globally most significant and **Impact** crucial for BIM/PGD attacks. Input Proportion's value is context-dependent (activation function).
- **Transferability:** Detectors trained on **BIM or PGD attacks show strong transferability**, effectively covering other adversarial threats without needing a universal 'Total detector'.
- **Impact:** Our safeguard significantly **reduces the success rates of evasion attacks** on target models

vicomtech
MEMBER OF BASQUE RESEARCH & TECHNOLOGY ALLIANCE

Future Work

- **Detector Optimization:**
  - Exploring **different GCN architectures and optimizing hyperparameters.**
  - Evaluating the method in **more diverse scenarios and datasets.**

- **Interpretability and Explainability:**
  - Using the detector's embeddings to understand the relationship **between topological information and neuron behavior.**
  - Identifying **vulnerable neurons** to generate targeted **local defenses.**

- **Broader Threat Detection:**
  - Extending the defense to detect other threats that cause activation anomalies, such as **poisoning and trojaning attacks.**

- **Full Model Analysis:**
  - Expanding the analysis to include the **feature extractor part** of the model (requires advanced preprocessing due to high parameter count).

Thank You

Thank You for your attention!

Questions?

xetxeberria@vicomtech.org

www.linkedin.com/in/xecheberria

Xabier Echeberria Barrio
Digital Security Researcher