המכללה
האקדמית להנדסה
בראודה
בכרמיאל

Anomaly recognition in citation networks using Beta Wavelet Graph Neural Network

Capstone Project Phase B
24-1-R-13


Maayan Sharvit


Shir Cohen

Supervisor:
Dr. Renata Avros

Introduction

01

Leveraging BWGNNs combined with Isolation Forest.

02

Detecting anomalies nodes in a nested manner.

Requirements

Graph Embedding

Use Node2Vec on graph data

Anomaly Detection

Apply the Isolation Forest approach

Model Flexibility

Dynamic thresholding in link prediction procedure

Model Training

Train the BWGNN model on categorized data

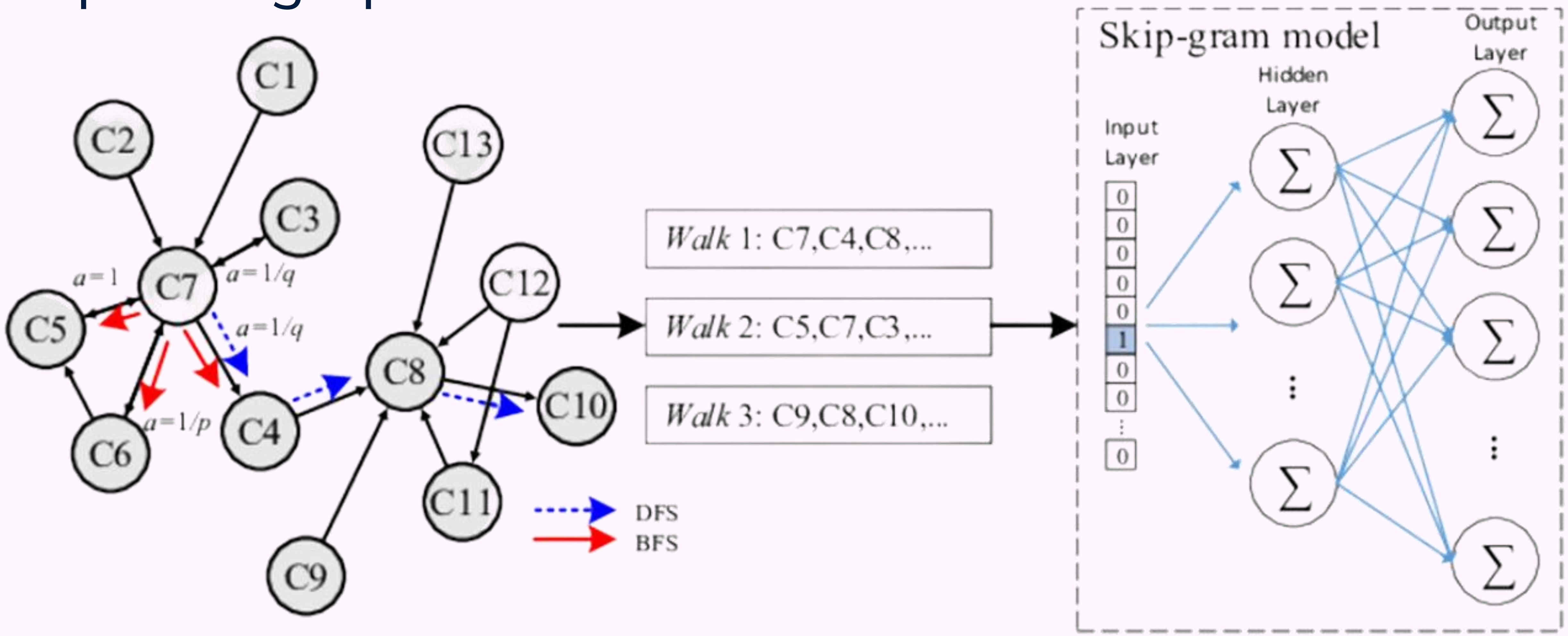
Iterative Refinement

Iterative approach to consider the nested structure of anomalies.

1

Node embeddings

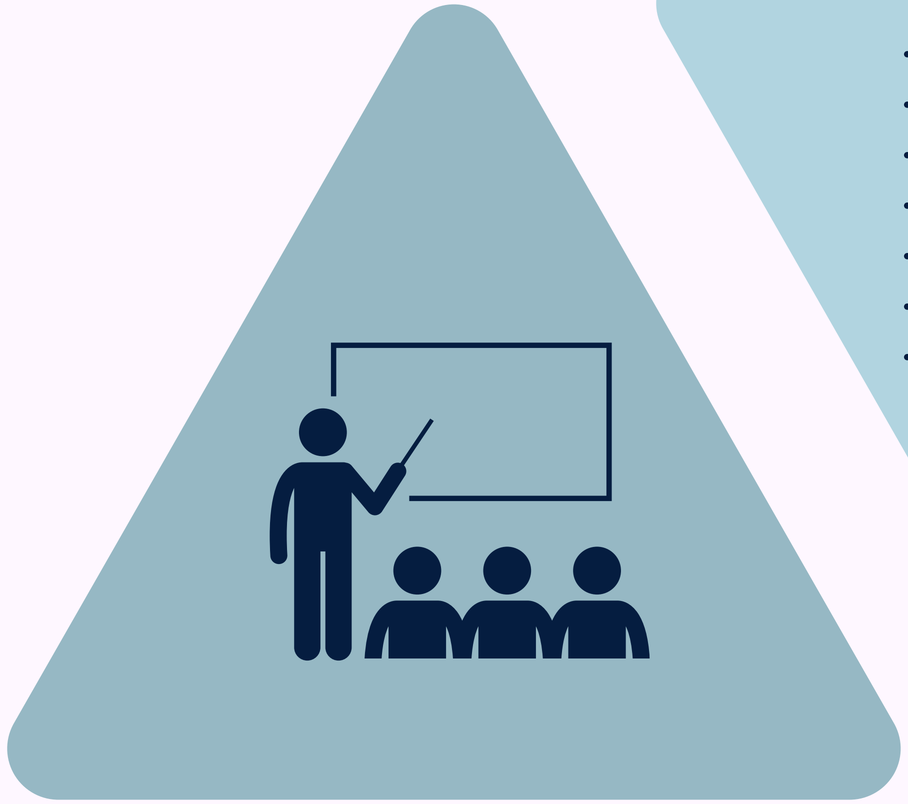
The Node2Vec approach is selected for graph embedding due to its flexibility in capturing the structural properties of nodes without over-relying on specific graph features.

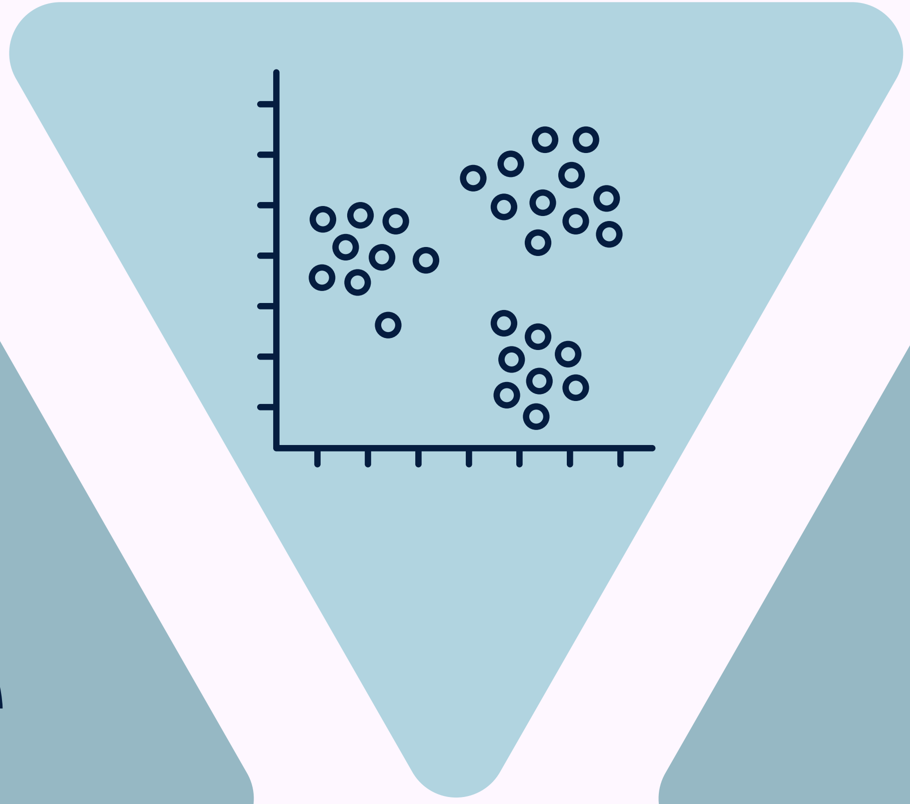


The conceptual framework of node2vec.

2 Anomaly detection

Initially considered K-means clustering, but it is limited by data distribution assumptions and its struggle with high-dimensional data.





Aiming to train the network anomaly set must be initialized.

Adopted the Isolation Forest algorithm operating flexibility and suitability on complex datasets.

3

Threshold

1

A threshold-based approach to classify nodes as anomalous using isolation scores is adopted, setting thresholds of 5%, 10%, and 20%.

2

The nodes are categorized into kernel (normal) and residual (anomalous) sets.

3

Training is conducted using these two classes, updated during the iterations.

4

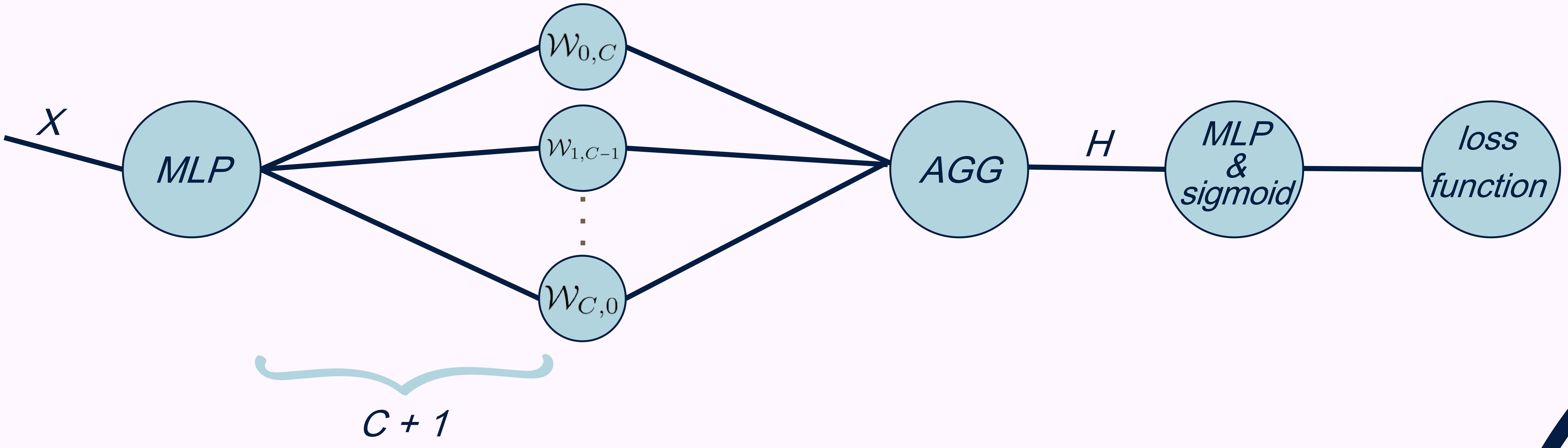
Iterations

The iterative process of identifying and removing anomalies ensured a progressively adjusted graph, enabling more precise results in subsequent iterations.

This approach makes it possible to build a robust and adaptable anomaly detection system for graph-based datasets.

5
Apply the BWGNN

BWGNN is an improved GNN, which employs Beta wavelets in parallel for each signal. These wavelets act as band-pass filters that since anomalous nodes distribute their spectral energy in high frequencies, capture anomalies effectively.



Final Solution

- Node2Vec for generating embeddings
- combining BWGNN and Isolation Forest for effectiveness
- Isolation Forest's dynamic thresholding
- Iterative refinement
- Visualizations and detailed results

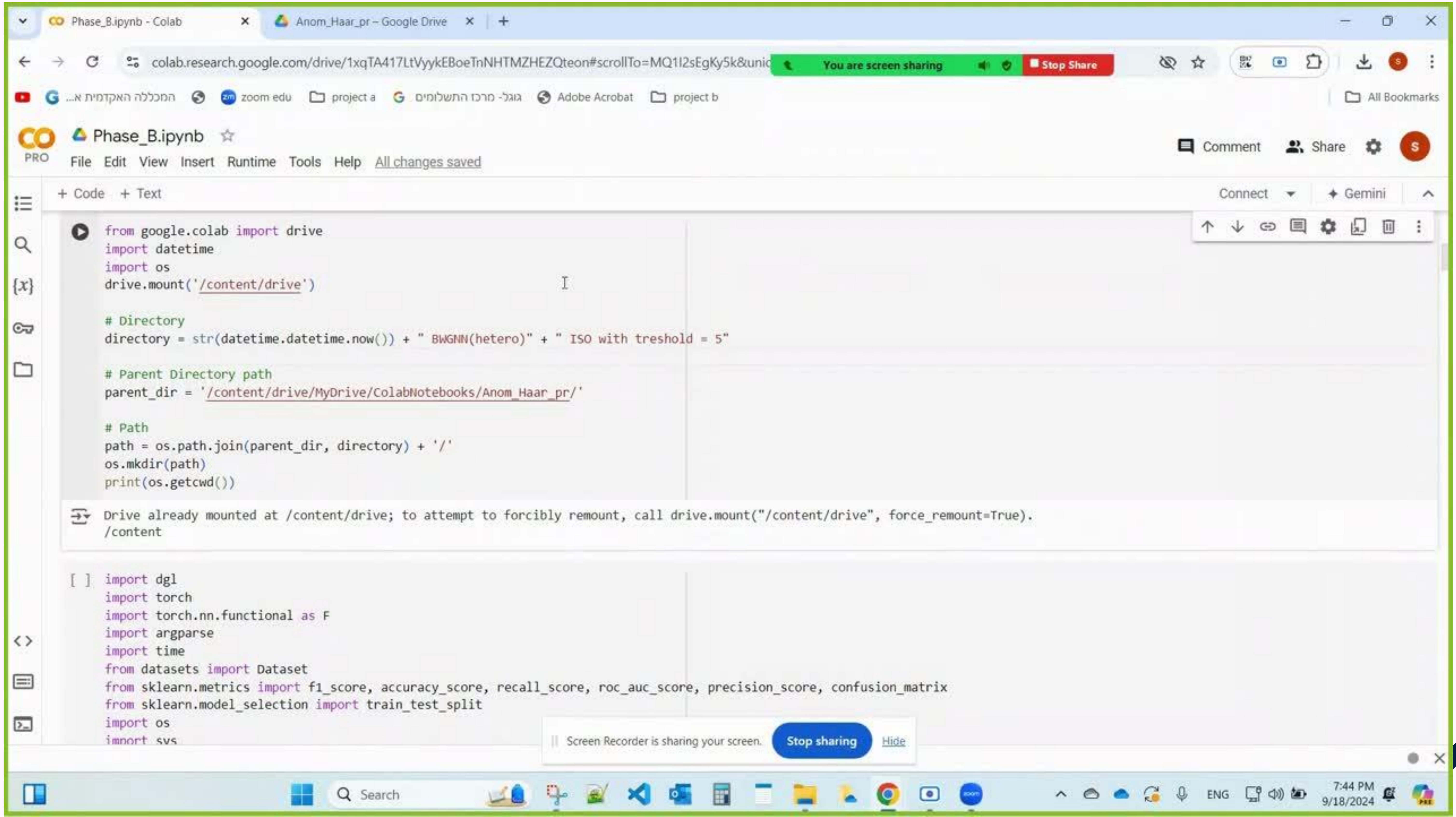
Final Solution

Six numerical experiments, including both BWGNN Hetero and Homo types, according to 5%, 10%, and 20% thresholding.

The results of each experiment are accurately documented.

Comparing the impact of different threshold settings and model variations on the system's performance.

Live Demonstration



Outputs

Node ID	Degree	Is Anomalous	Iteration 0	Iteration 1	Iteration 2	Iteration 3
0	3	0	0	0	0	0
1	3	0	0	0	0	0
2	5	0	0	0	0	0
3	1	0	0	0	0	0
4	5	0	0	0	0	0
5	3	0	0	0	0	0
6	4	0	0	0	0	0
7	1	0	0	0	0	0
8	3	0	0	0	0	0
9	2	0	0	0	0	0
10	2	0	0	0	0	0
11	2	0	0	0	0	0
12	4	0	0	0	0	0
13	2	0	0	0	0	0
14	5	0	0	0	0	0

Anomalies detected across all iterations.

Iteration 1	Iteration 2	Iteration 3	Iteration 4
518	386	343	265

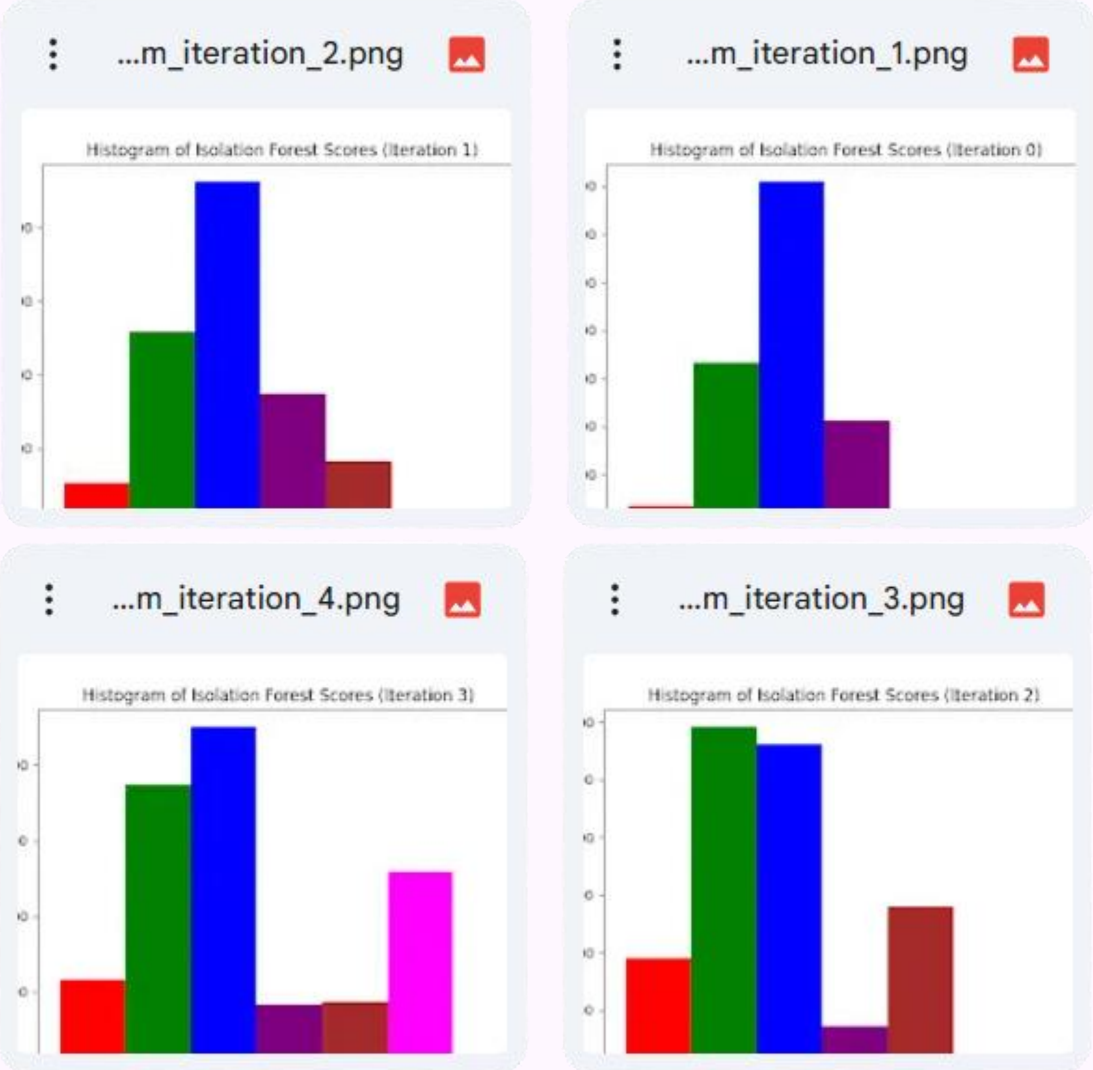
Sum of anomalies in each iteration.

Iteration 2 results.csv

Iteration 3 results.csv

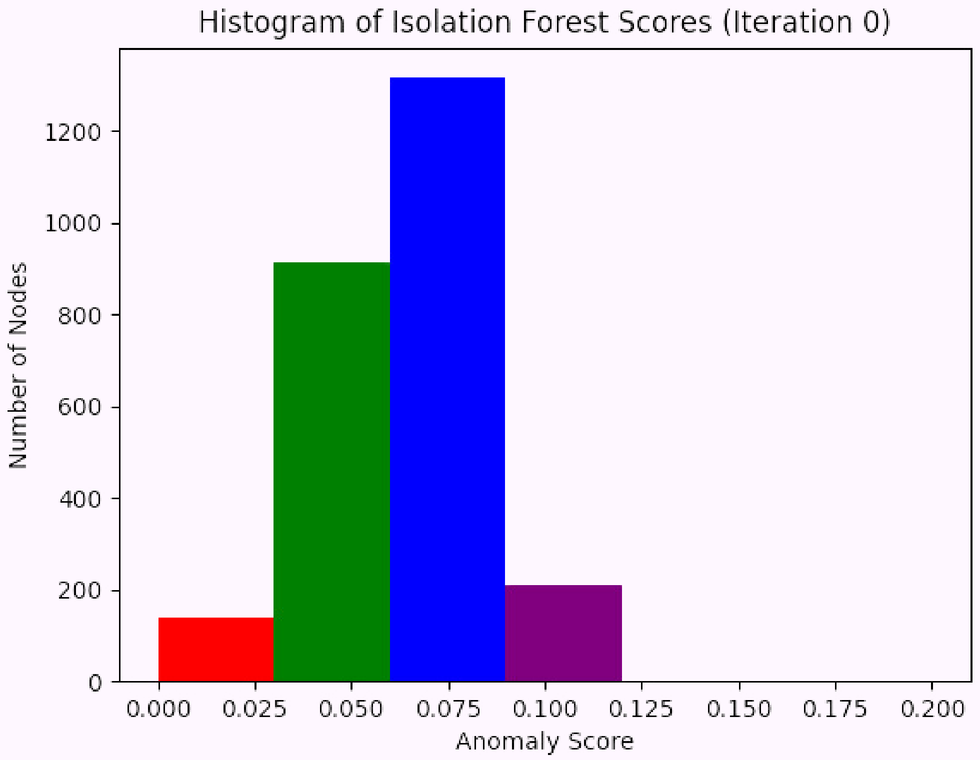
Iteration 4 results.csv

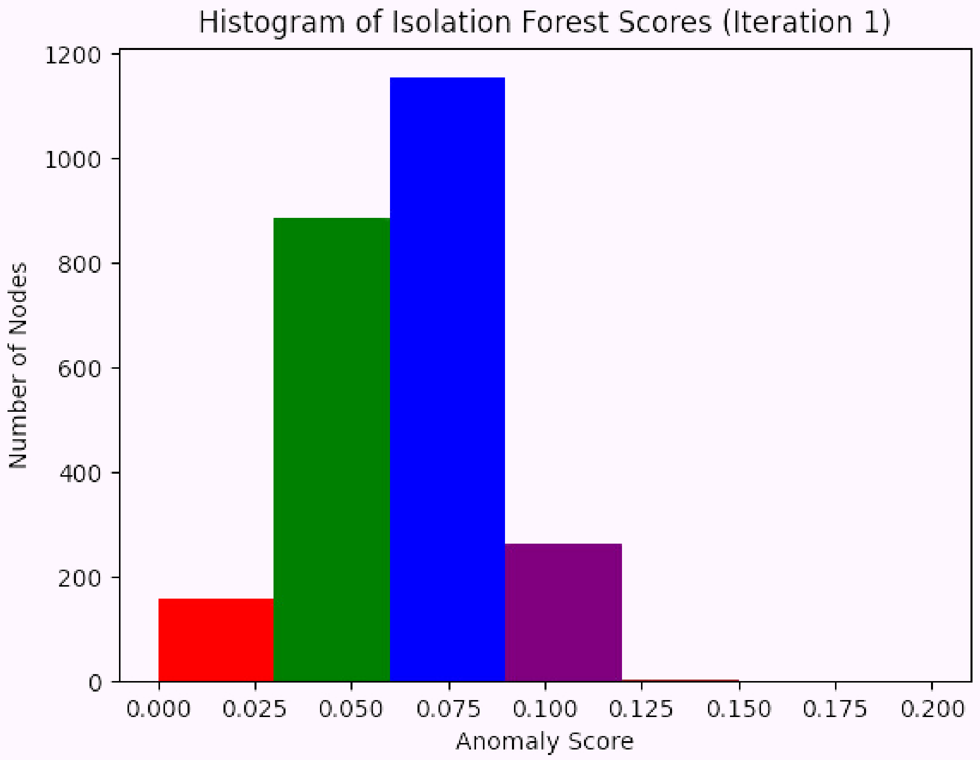
Anomalous nodes identified in each cycle.

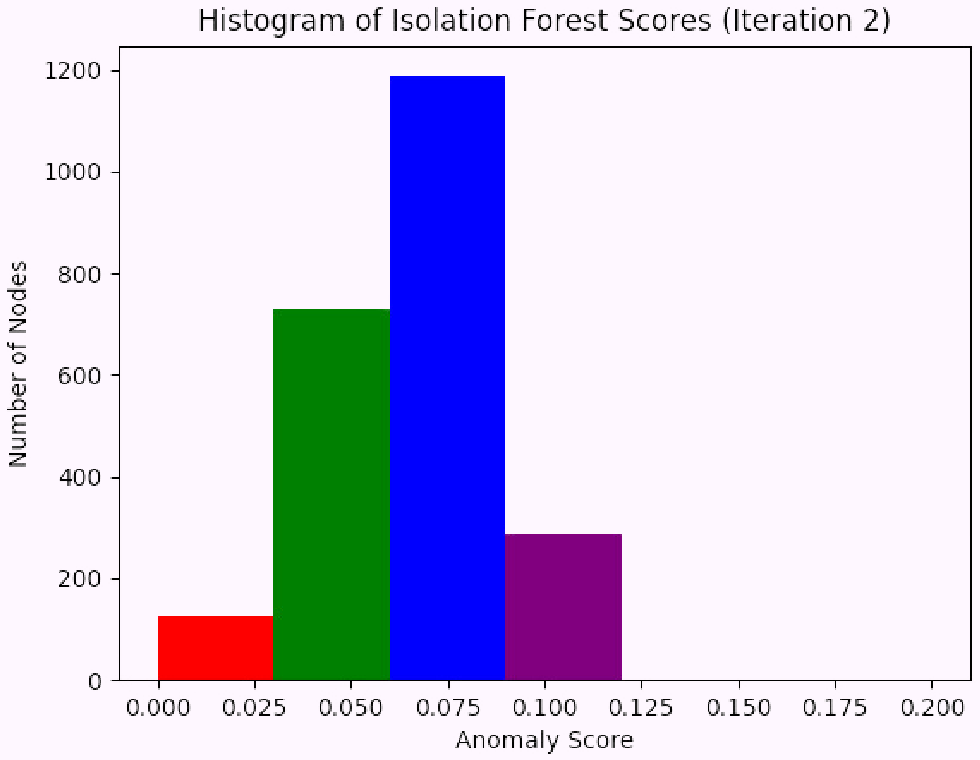


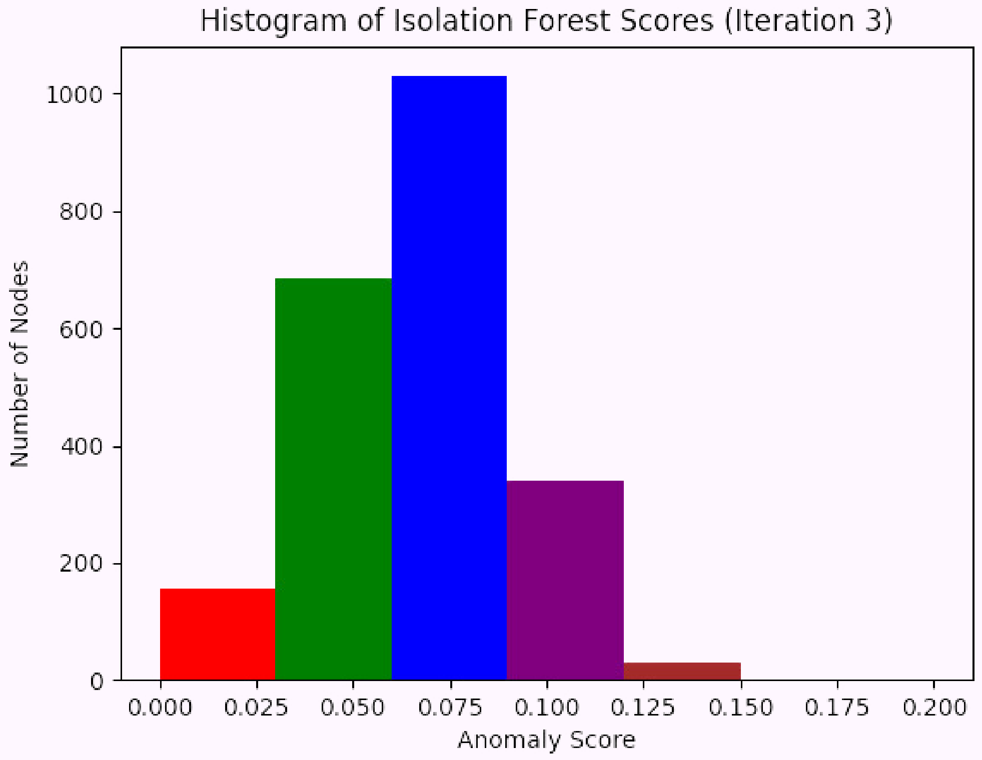
Distribution of isolation forest scores by nodes.

Threshold 5









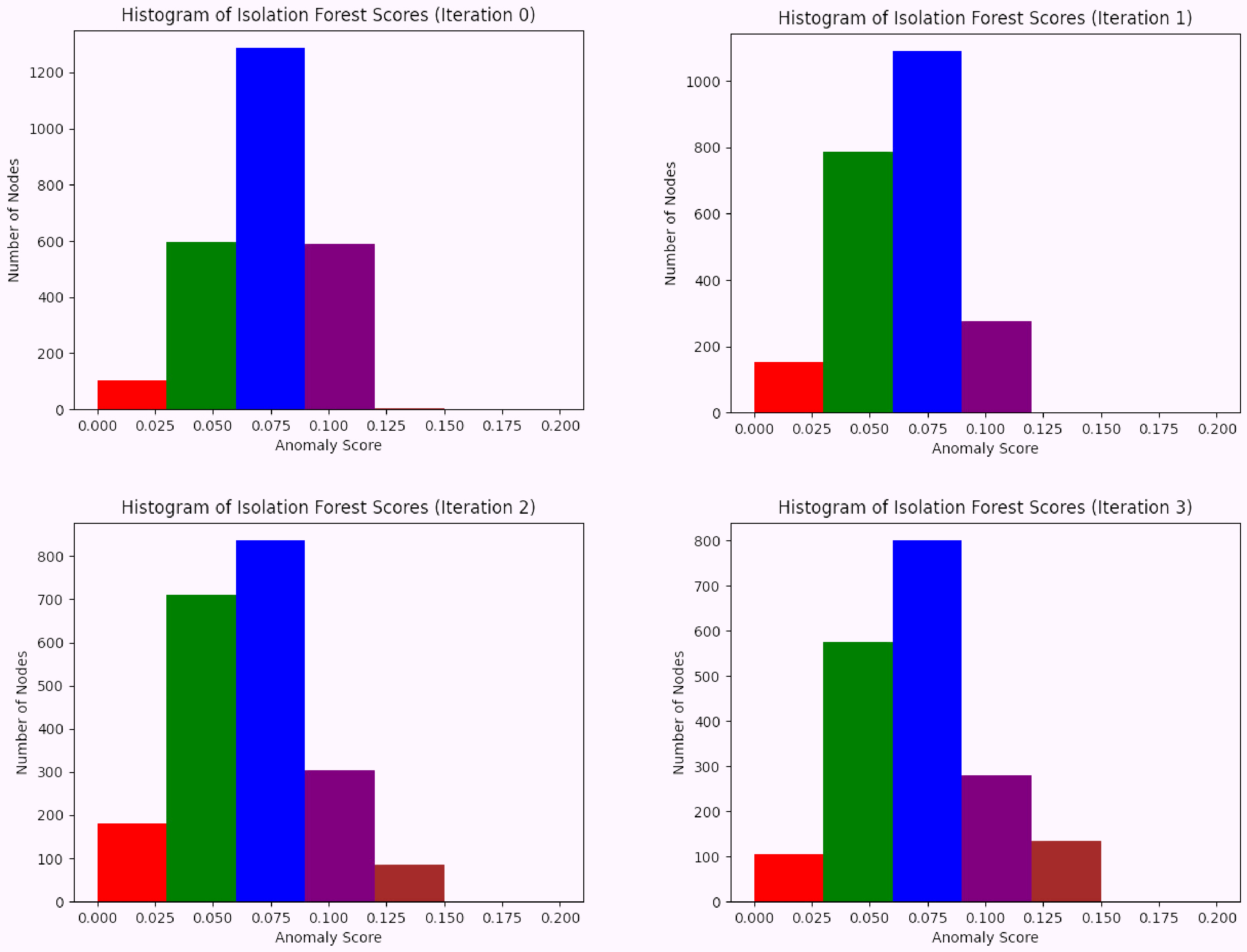
Remain stable across all iterations.
This indicate a gentle approach to detecting anomalies.

Slightly decreasing across all iterations.
This indicate that nodes are being marked and removed from the middle areas.

Slightly increasing across all iterations.
This increase suggests that as the model processes more data and starts refining its detection.

Appears In this range only in the last iteration.
Which suggests that the model become more sensitive to certain types of behaviors that were not sufficiently pronounced in earlier iterations.

Threshold 10

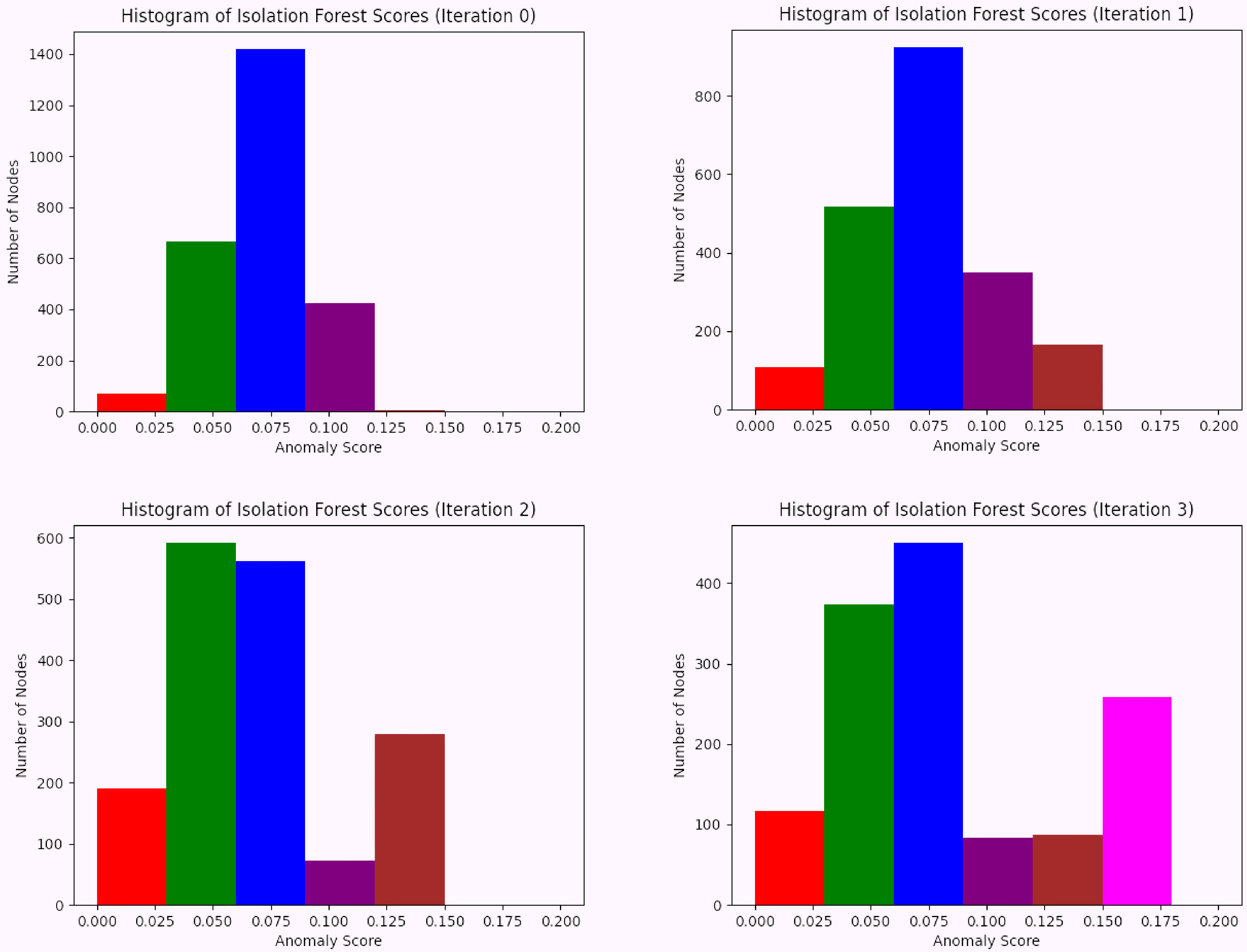


Increases across iterations, indicating effective detection over time. However, there is a minor reduction in the final iteration, as some anomalies undergo reevaluation

Decreases over the iterations, which indicates that the model is becoming more precise in differentiating between normal and anomalous nodes.

Appears in the third iteration and increases, which suggests that the model is becoming more effective at distinguishing between what it considers normal behavior and anomalies.

Threshold 20



The count of nodes in this range is unstable because the model is dynamically adjusting its classification boundaries

Decrease that that anomaly is being detected and nodes from the mid-levels are filtered to the edges.

Increase in nodes that reflect the model's success in anomaly and normal detection.
The decrease at the end could reflect the model adjusting its sensitivity or re-evaluating nodes.

Large increase happened due to deeper insights gained from the data after multiple iterations.

Insights all histograms

5

Overly restrictive identification of anomalies, risking missing subtler anomalous patterns.

10

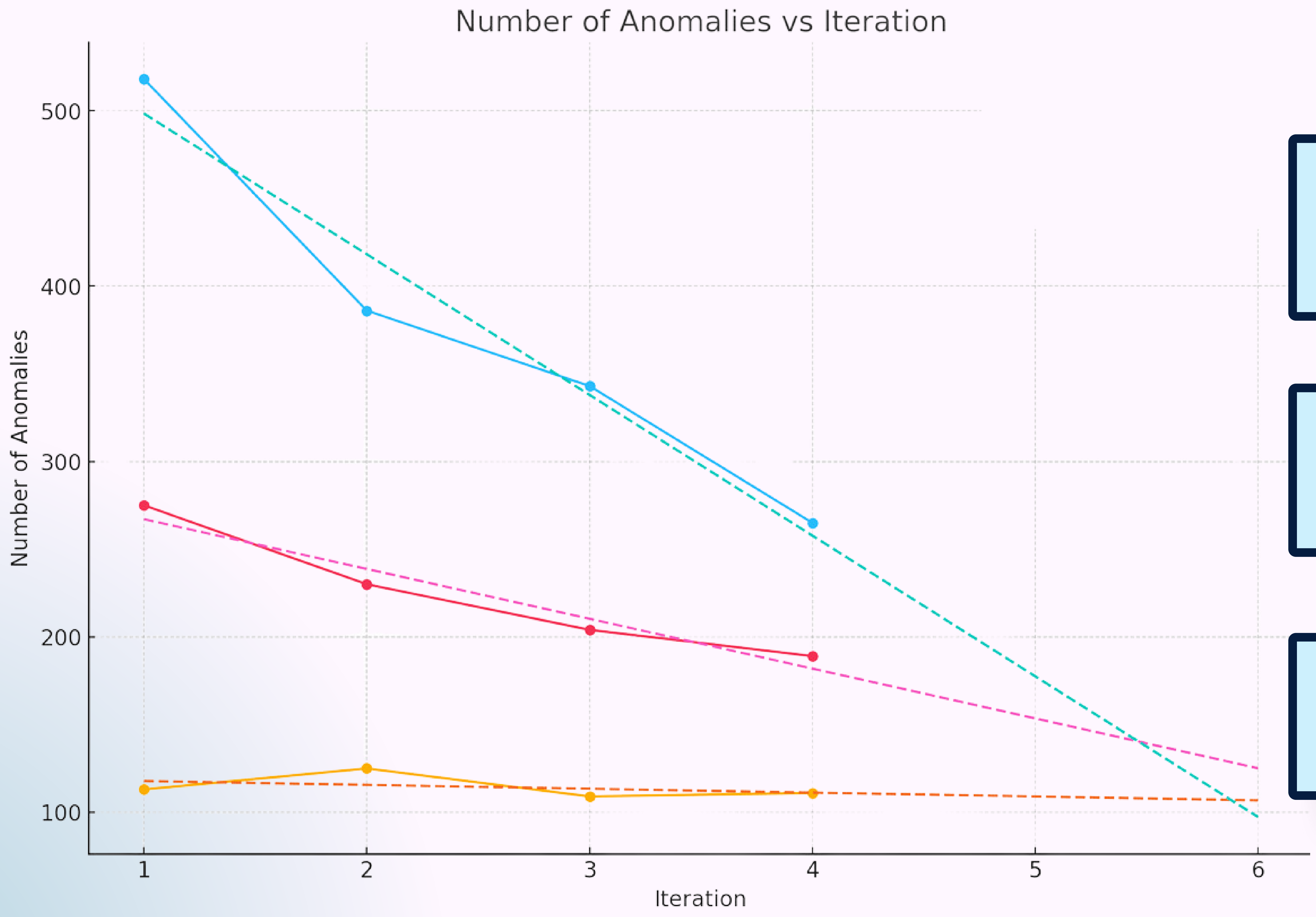
A balance between sensitivity and specificity.

20

Broader identification of anomalies, but with increased risk of misclassifying normal nodes as anomalous.

Lower thresholds are ideal for precision-focused tasks, while higher thresholds are better for environments where missing an anomaly could be detrimental.

Nested Iterations



Prediction for Future Iterations

The predicted trend shows that future iterations will continue to detect fewer anomalies and converge.

Closer to Kernel Nodes

As iterations progress, the process of detecting and removing anomalies allows the model to get closer to identifying the kernel nodes which considered as normal.

Nested Anomalies

The iterative reduction of anomalies reflects the "nested" nature of the anomaly detection process.

Conclusions

01

The choice of four iterations was effective, allowing the model to refine anomaly detection. Further iterations are expected to converge as the kernel set increasingly represents normal nodes in nested anomalies.

02

The results align with those achieved by our supervisors, confirming the validity of our approach.

03

Histograms across thresholds (5%, 10%, 20%) were similar but had key differences:
At 20%, more nodes were flagged, making it assertive in detection but less precise.
At 5%, only extreme anomalies were captured, missing subtler ones.
The 10% threshold balanced precision and recall.?

04

While the results are good, we could explore alternative methods like Kernel Sum to potentially improve anomaly detection.

Verification Plan

	Name	Description	Expected Output
Case 1	Insufficient Data	The graph contains too few nodes or edges to effectively train the model.	A warning message is displayed, suggesting providing more data.
Case 2	Incorrect Anomaly Threshold	The user sets an inappropriate threshold for classifying nodes as anomalies.	The model might misclassify nodes as normal or anomalous.
Case 3	Evaluation Metrics Mismatch	The chosen evaluation metrics (e.g., precision, recall, F1-score) are not aligned with the specific task or dataset.	The model's performance might be misinterpreted.
Case 4	Model Training Failure	The model fails to train due to issues like hyperparameter tuning, convergence problems, or insufficient computational resources.	An error message is displayed, indicating the failure and suggesting potential solutions.

Thank you for listening