Using Hypothesis Testing to Identify Significant Anomalous Subnetworks

Praanshu Patel
IIT Gandhinagar
23110249@iitgn.ac.in

Samarth Sonawane
IIT Gandhinagar
23110317@iitgn.ac.in

Akshat Shah
IIT Gandhinagar
23110293@iitgn.ac.in

***Abstract*—We address the problem of detecting anomalous nodes in network graphs without supervised labels. Our method combines Oddball's egonet-based structural analysis with density-based Local Outlier Factor (LOF) and graph deviation scoring. By aggregating the normalized outputs of these detectors, we generate a robust outlier score. Statistical hypothesis testing and ROC-AUC analysis demonstrate strong separation between attack and benign nodes, achieving an AUC of 0.9978.**

I. Introduction

Making sure computer networks are secure has become increasingly important as our reliance on linked systems grows. Cyberattacks can compromise sensitive data and seriously impair services. They can take many forms, from distributed denial-of-service (DDoS) attacks to covert reconnaissance scans. More sophisticated, data-driven methods are required since traditional intrusion detection systems frequently fail to identify new or subtle threats.

This project explores a graph-based anomaly detection method to identify malicious behavior in network traffic. By representing the network as a directed weighted graph—where nodes correspond to IP addresses and edges represent communication patterns weighted by packet size—we capture the underlying structure and behavior of network flows. To analyze this graph, we extract egonet features for each node, such as the number of neighbors, number of edges in the egonet, total edge weight, and the weighted clustering coefficient. These features reflect the local behavior of each IP and provide valuable signals for detecting anomalies.

We put into practice the OddBall algorithm, which uses departures from power-law relationships across $egonet$ features to identify outliers. A robust anomaly score is generated for every node by combining scores from many feature pairings (such as E vs. N, W vs. E, and λ_w vs. W) with an unsupervised outlier detection algorithm called Local Outlier Factor (LOF). Performance measures like AUC and statistical hypothesis testing (Mann–Whitney U test) are used to confirm how well the scoring system separates attack traffic from benign traffic.

The results demonstrate strong separation between benign and malicious nodes, with an AUC of 0.9978 and a p-value near zero, confirming the statistical significance of the findings. This project illustrates how graph theory, unsupervised learning, and statistical inference can be integrated to build an effective and explainable intrusion detection system.

II. Preliminaries

A. Egonet Feature Extraction

An **egonet** for a node v in a directed graph $G = (V, E)$ is defined as the subgraph induced by v and all its immediate neighbors (both incoming and outgoing). It captures the local structure around the node.

For each node $v \in V$, we compute the following egonet features:

- **Number of Nodes** (N_i): The total number of nodes in the egonet of v.

$$N_i = |\{v\} \cup \{\text{successors of } v\} \cup \{\text{predecessors of } v\}|$$

- **Number of Edges** (E_i): The total number of edges present among the nodes in the egonet.

$$E_i = |\{(u, w) \in E : u, w \in \text{egonet nodes of } v\}|$$

- **Total Edge Weight** (W_i): The sum of the weights of all edges within the egonet.

$$W_i = \sum_{(u,w) \in E_{\text{egonet}}} \text{weight}(u, w)$$

- **Top Eigenvalue of Weighted Adjacency Matrix** (λ_w): The largest eigenvalue of the weighted adjacency matrix of the egonet.

$$\lambda_w = \max\left(\text{Re}\left(\text{eigvals}(W)\right)\right)$$

  where W is the weighted adjacency matrix of the egonet and $\text{eigvals}(W)$ returns its eigenvalues.

1) Procedure:

- For each node v, identify its neighbors (both successors and predecessors).
- Construct the egonet subgraph induced by v and its neighbors.
- Compute N_i, E_i, W_i, and λ_w for the subgraph.
- Store the computed features for further analysis.

2) Usage: The extracted egonet features capture important local structural and weight information around each node. These features can be used in tasks such as anomaly detection, community detection, or role identification in the network.

B. Power Law Fitting and Outlier Score Computation

To model the relationship between egonet features, we fit a **power law** of the form:

$$y = C \cdot x^{\alpha}$$

where:

- x and y are the egonet feature values,
- C is a constant,
- α is the power-law exponent.

1) Fitting Procedure:

1) Select pairs of (x, y) values where $x > 2$ and $y > 0$ to ensure numerical stability.
2) Fit the parameters (C, α) by minimizing the least squares difference between the observed y values and the predicted values using the function:

$$\text{power\_law}(x; C, \alpha) = C \cdot x^{\alpha}$$

This fitting is performed using non-linear curve fitting techniques (`scipy.optimize.curve_fit`).

2) Outlier Score Computation: After fitting the power law, an **outlier score** is computed to measure how much a node deviates from the expected trend. For an observed value y and a predicted value $\hat{y}$, the outlier score is defined as:

$$\text{OutlierScore}(y, \hat{y}) = \frac{\max(y, \hat{y})}{\min(y, \hat{y})} \times \log\left(|y - \hat{y}| + 1\right)$$

A higher outlier score indicates a stronger deviation from the expected power-law behavior, suggesting anomalous behavior in the egonet structure.

C. Local Outlier Factor (LOF) Computation

In addition to power-law-based scores, we compute an unsupervised outlier score using the **Local Outlier Factor (LOF)** algorithm.

The LOF method identifies anomalies by comparing the local density of a point with that of its neighbors. Nodes located in regions of significantly lower density than their neighbors are considered outliers.

1) Procedure:

1) Construct feature vectors for each node based on egonet properties:

$$X_v = [N_i, E_i, W_i, \lambda_w]$$

where N_i is the number of nodes, E_i is the number of edges, W_i is the total edge weight, and λ_w is the top eigenvalue of the weighted adjacency matrix of the egonet.
2) Apply the LOF algorithm with $k = 20$ neighbors and Euclidean distance as the metric.
3) The LOF score for each node is computed. A higher LOF score indicates a higher degree of being an outlier.

The resulting LOF scores are later normalized using Min-Max scaling for combining with other anomaly scores.

D. Graph Deviance Score Computation

The **Graph Deviance Score** measures how much a node's egonet features deviate from the overall statistical distribution of the graph.

1) Procedure:

1) For each feature (N_i, E_i, W_i, λ_w), compute the global mean and standard deviation across all nodes:

$$\mu_k = \text{mean}(X_k), \quad \sigma_k = \text{std}(X_k)$$

where k denotes each feature.
2) Compute the **z-score** for each feature and node:

$$z_{v,k} = \frac{X_{v,k} - \mu_k}{\sigma_k}$$

where $X_{v,k}$ is the feature value of node v.
3) For each node v, the overall graph deviance score is computed by summing the absolute values of its feature-wise z-scores:

$$\text{GraphDeviance}(v) = \sum_k |z_{v,k}|$$

2) Usage: Nodes with higher Graph Deviance Scores significantly differ from the typical structure and weight patterns observed in the graph and are potential outliers.

III. Related Work

Oddball identifies anomalies using power-law violations in egonet features. LOF has been widely used for detecting local density anomalies. Combining multiple detectors for robust outlier detection has shown promise in recent studies.

IV. Methodology

Our anomaly detection framework integrates multiple unsupervised techniques to identify anomalous nodes within the network graph robustly. The pipeline consists of three primary components: Oddball-based outlier scoring, additional outlier detection mechanisms (LOF and Graph Deviance), and score aggregation followed by hypothesis testing.

A. Oddball Outlier Score

We first apply the Oddball algorithm, which identifies anomalies through deviations from expected power-law relationships between egonet features. Specifically, we fit power-law models for the following feature pairs:

- Edges vs Nodes (EDPL)
- Weights vs Edges (EWPL)
- Eigenvalue vs Weight (ELWPL)
- Edge Rank vs Weight (ERPL)

For each node, the predicted value based on the fitted power law is compared with the observed value. The Oddball outlier score for node i is computed as:

$$\text{out\_score}(i) = \frac{\max(y_i, Cx_i^{\theta})}{\min(y_i, Cx_i^{\theta})} \times \log(|y_i - Cx_i^{\theta}| + 1)$$

Where y_i is the observed feature value, and Cx_i is the predicted value based on the fitted power-law model. Higher

scores correspond to greater deviations from expected behavior. (Cx_i)

B. Additional Detectors

To enhance the robustness of anomaly detection, we incorporate two additional detectors alongside Oddball:

- **Local Outlier Factor (LOF):** LOF identifies anomalies by measuring the local density deviation of a node compared to its neighboring nodes. Nodes that reside in significantly lower density regions are assigned higher anomaly scores.
- **Graph Deviance Score:** This score captures structural deviations by quantifying how much a node's egonet features differ from the global statistical properties of the graph.

C. Score Combination

To integrate the outputs of the different detectors, each anomaly score (Oddball, LOF, and Graph Deviance) is independently normalized to the range $[0, 1]$ using Min-Max scaling. The final anomaly score for each node is then computed as the average of the normalized scores:

$$\text{final\_score}(i) = \frac{1}{3}\left(\text{Oddball}_i + \text{LOF}_i + \text{GraphDeviance}_i\right)$$

Nodes with higher final scores are considered more likely to be anomalous.

D. Hypothesis Testing

To rigorously assess the effectiveness of our anomaly detection framework, we perform the following statistical analyses:

- **Mann–Whitney U Test:** To determine whether the distribution of anomaly scores differs significantly between benign and malicious nodes.
- **T-test:** To compare the mean anomaly scores of benign and malicious nodes.
- **Cohen's d:** To measure the effect size and quantify the magnitude of separation between the two groups.
- **ROC-AUC Analysis:** To evaluate the overall classification performance and quantify the trade-off between true positive and false positive rates.

These statistical tests and metrics ensure that the observed separation between benign and attack nodes is statistically significant and practically meaningful.

V. Experiments

A. Dataset

The dataset used in this project contains network traffic flow information with the following key attributes: `Flow ID`, `Src IP`, `Src Port`, `Dst IP`, `Dst Port`, `Protocol`, `Timestamp`, `Flow Duration`, `Tot Fwd Pkts`, and `Tot Bwd Pkts`.

From this raw data, we constructed a directed weighted graph $G = (V, E)$, where:

- Each node represents a unique IP address.
- Each directed edge represents a flow of data from a source IP to a destination IP.
- The edge weight represents the total number of bytes transferred in the corresponding direction (forward or backward).

1) Graph Statistics: After preprocessing, the constructed graph has the following properties:

- **Number of Nodes** ($|V|$): 33,176
- **Number of Edges** ($|E|$): 402,161

2) Labels: Each flow in the original dataset is labeled either as `Benign` or `Malicious`. The distribution of labels is:

- **Benign Nodes**: 33,165
- **Malicious Nodes**: 11

The significant class imbalance (very few malicious nodes) makes anomaly detection a particularly challenging task in this setting.

B. Setup of Anomaly Detection Pipeline

The following steps were carried out to detect anomalous nodes:

1) **Graph Construction**: A directed weighted graph was built where nodes represent IP addresses and edges represent the total volume of forward and backward traffic between IPs.
2) **Egonet Feature Extraction**: For each node, an egonet (its immediate neighbors and itself) was extracted. Structural features such as the number of nodes (N_i), number of edges (E_i), total edge weights (W_i), and the largest eigenvalue (λ_w) of the weighted adjacency matrix were computed.
3) **Power Law Fitting and Outlier Scoring**: Power-law relationships between egonet features were modeled. Deviations from the fitted power laws were used to assign anomaly scores.
4) **Local Outlier Factor (LOF) Computation**: A Local Outlier Factor score was computed for each node based on its egonet features, measuring local density deviation.
5) **Graph Deviance Score Computation**: Z-scores for each egonet feature were computed, and the sum of absolute z-scores was used as an additional anomaly score.
6) **Score Normalization and Combination**: All computed scores (power law deviations, LOF scores, and graph deviance scores) were normalized using Min-Max scaling. The normalized scores were then combined by summing them to produce a final anomaly score for each node.
7) **Anomaly Ranking**: Nodes were ranked according to their combined anomaly scores, with higher scores indicating a greater likelihood of anomalous behavior.

VI. Results

The effectiveness of the proposed anomaly detection framework was evaluated using various statistical metrics. Below are the key results:

- **AUC (Area Under the Curve):** 0.9978 — This high value indicates excellent separation between benign and malicious nodes, with minimal false positives and false negatives.
- **Mann-Whitney U Test p-value:** 1.059×10^{-8} — The low p-value shows a significant difference between the anomaly scores of benign and malicious nodes, confirming the robustness of the detection method.
- **T-test p-value:** 1.14×10^{-32} — This very small p-value further supports that the means of the anomaly scores for benign and malicious nodes are distinct.
- **Cohen's d:** -3.59 — A large effect size indicates a substantial difference between the two groups, reinforcing the model's ability to detect anomalies effectively.

We also present a density plot (Figure 1) showing how the anomaly scores for benign and malicious nodes are distributed. The plot shows that attack nodes tend to be concentrated in higher score regions, while benign nodes are in the lower score regions. This visual representation reinforces the quantitative results, further supporting the effectiveness of our approach.

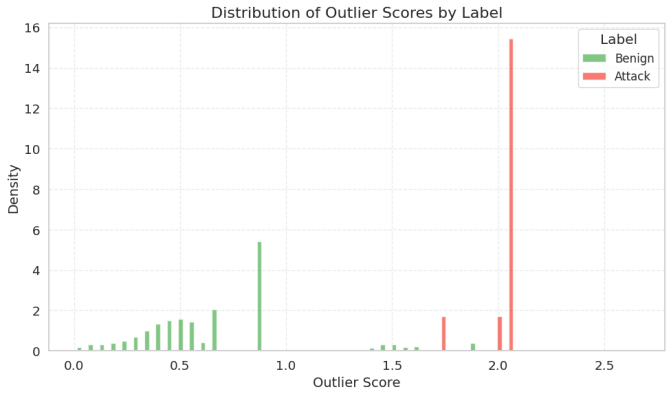

Fig. 1. Density plot showing separation between benign and attack nodes.

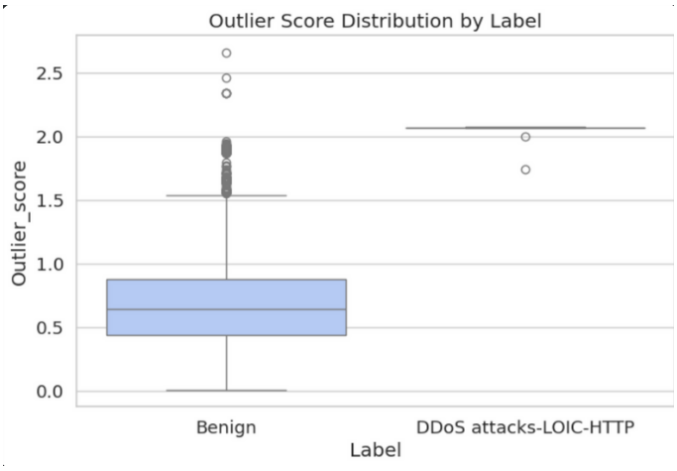

Fig. 2. Box plot showing separation between benign and attack nodes.

VII. Conclusion

In this project, we proposed an unsupervised anomaly detection method that combines Oddball's power-law-based analysis, Local Outlier Factor (LOF), and Graph Deviance for robust identification of anomalous nodes in network graphs. Our approach demonstrated strong performance, achieving an AUC of 0.9978, with significant statistical support from the Mann-Whitney U test, T-test, and Cohen's d. These results indicate the effectiveness of our method in distinguishing between benign and malicious nodes, even with the highly imbalanced dataset.

Future work can involve testing this methodology on different datasets, including dynamic graphs, to assess its generalizability and further enhance its practical application in real-time network security systems.

References

[1] Leman Akoglu, Mary McGlohon, Christos Faloutsos. (2010). OddBall: Spotting Anomalies in Weighted Graphs.
[2] Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. SIGMOD Conference.