



# Anomaly Detection in Network Data

A Unsupervised Learning Approach

Samuel Ghalayini



# Table of contents

**01** Introduction

**02** Anomalies

**03** Data

**04** Model

.....





# Introduction


## Motivation

Chose to explore anomaly detection after surveying diverse datasets (power grids, machinery, network traffic) and found network data especially compelling for its cybersecurity impact.

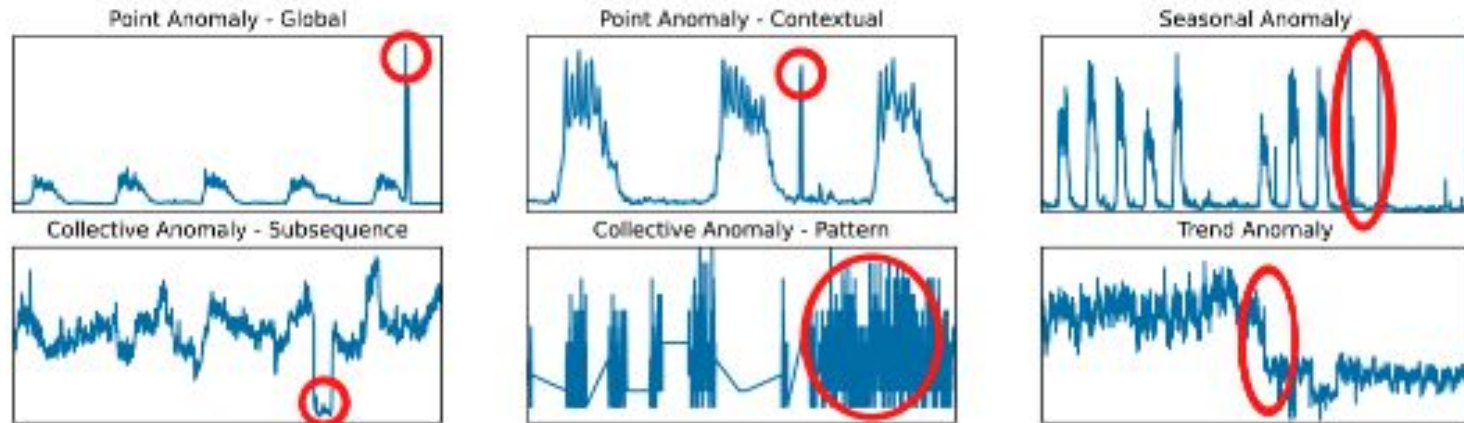
## Scope

Leverage CESNET-Timeseries24 data at three aggregation levels (10 min, 1 hr, 1 day) to detect outliers in features like packet/flow counts and TCP/UDP ratios.

Apply an unsupervised pipeline—EDA with correlation plots, box plots, and KDEs; PCA for dimensionality reduction; and K-Means clustering—to flag anomalous IP addresses without labeled examples.



# Anomalies



8. <https://www.nature.com/articles/s41597-025-04603-x>



# Data

## Features:

id\_time, n\_flows, n\_packets, n\_bytes, sum\_n\_dest\_asn, average\_n\_dest\_asn, std\_n\_dest\_asn, sum\_n\_dest\_ports, average\_n\_dest\_ports, std\_n\_dest\_ports, sum\_n\_dest\_ip, average\_n\_dest\_ip, std\_n\_dest\_ip, tcp\_udp\_ratio\_packets, tcp\_udp\_ratio\_byte, dir\_ratio\_packets, dir\_ratio\_bytes, avg\_duration, avg\_ttl, timestamp

Aggregation	Number of Time Slots
10 minutes	40,297
1 hour	6,717
1 day	279

## Tree of datasets

ip\_addresses\_sample/  
├── agg\_10\_minutes/  
├── agg\_1\_hour/  
└── agg\_1\_day/

institutions/  
├── agg\_10\_minutes/  
├── agg\_1\_hour/  
└── agg\_1\_day/

institutions\_subnets/  
├── agg\_10\_minutes/  
├── agg\_1\_hour/  
└── agg\_1\_day/



# Exploratory Data Analysis and Demo of Notebook

---

---

---



# Model



## PCA:

Scaled with a RobustScaler to mitigate outliers, then reduced to the minimum number of principal components explaining 95% of the total variance.

Focused on the directions of maximum variance guided by feature skewness and kurtosis to retain the most representative structure in the data.

## K-Means:

Used the elbow method to narrow down k, then selected the final cluster count by maximizing the silhouette score for best cohesion and separation.

Flagged anomalies by converting each point's distance to its cluster centroid into z-scores and labeling any IP address more than  $3\sigma$  away as an outlier.



# Results

Category	Aggregation	Features	PCA Dims	K-Means k	Anomalies (count/total)
IP	10min	44	1	2	5 / 1000
IP	1hr	36	1	2	5 / 1000
IP	1day	36	1	2	5 / 1000
instit	10min	36	4	2	7 / 283
instit	1hr	36	5	2	11 / 283
instit	1day	36	2	2	5 / 283
instit_subnet	10min	36	4	2	7 / 548
instit_subnet	1hr	36	3	2	6 / 546
instit_subnet	1day	36	2	2	10 / 546





# Discussion and Conclusion

- Gained deep insights into K-Means and the power of unsupervised methods as the “cake” of ML
- Unsupervised workflows require intensive data understanding, tuning, and iteration
- Initial results highlight bulk IP anomaly detection is useful for flagging suspicious actors

## Improvements:

- Integrate real network traffic for richer feature context
  - Aim to move from anomaly detection to attack classification
  - Plan to incorporate labeled “attack” data to train and validate clustering models
-



# References

1. <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
2. <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
3. [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html)
4. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html>
5. <https://www.geeksforgeeks.org/how-to-calculate-skewness-and-kurtosis-in-python/>
6. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html>
7. <https://zenodo.org/records/13382427>
8. <https://www.nature.com/articles/s41597-025-04603-x>

Github: [https://github.com/sam-ghala/fault\\_detection](https://github.com/sam-ghala/fault_detection)

