Incremental graph processing for detection of network attacks

Younes Benreguieg[1] Pierre Parrend[1]

[1]Security and Systems group
LRE - EPITA Research Laboratory

Research Seminar, 29 June 2023

How can we effectively manage anomaly detection and visualization in large-scale, complex network traffic data?

- What mechanisms can we use to detect specific attack patterns in network traffic data?
- How can we effectively identify anomalies in network traffic data given their volume and complexity?
- How can we create an effective data visualization and analysis tool for these data?

Network Attacks Detection Issues

How can we effectively manage anomaly detection and visualization in large-scale, complex network traffic data?

- What mechanisms can we use to detect specific attack patterns in network traffic data?
- How can we effectively identify anomalies in network traffic data given their volume and complexity?
- How can we create an effective data visualization and analysis tool for these data?

Overall Problem

The goal is to work with very large datasets.

Table of Contents

1. Application Demonstration
2. Cybergraph and its Architecture
   - Architecture
   - Attack patterns
   - CI/CD & Deployment
3. Anomalies detection
   - Anomalous
   - AMAD
   - Benchmark
4. Future Works and Conclusion

Table of Contents

1. Application Demonstration
2. Cybergraph and its Architecture
   - Architecture
   - Attack patterns
   - CI/CD & Deployment
3. Anomalies detection
   - Anomalous
   - AMAD
   - Benchmark
4. Future Works and Conclusion

Cybergraph

Let's do a demo!

Cybergraph

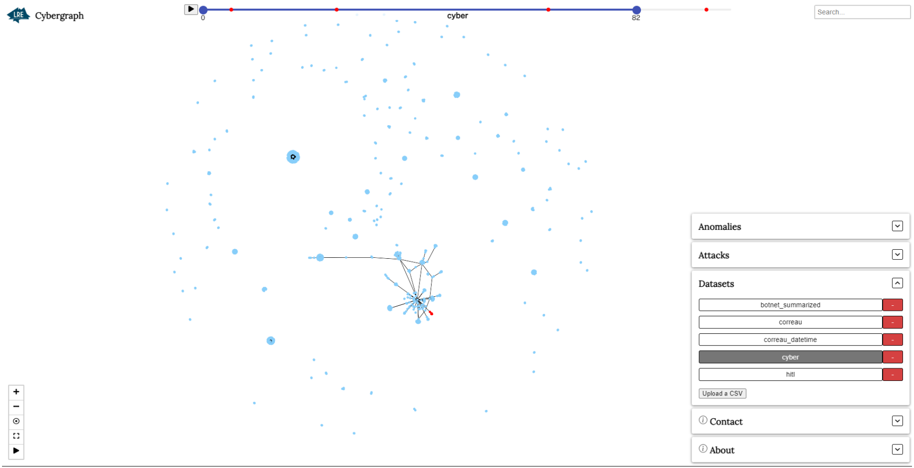

Figure: Cybergraph with UGR'16 sample dataset

Cybergraph

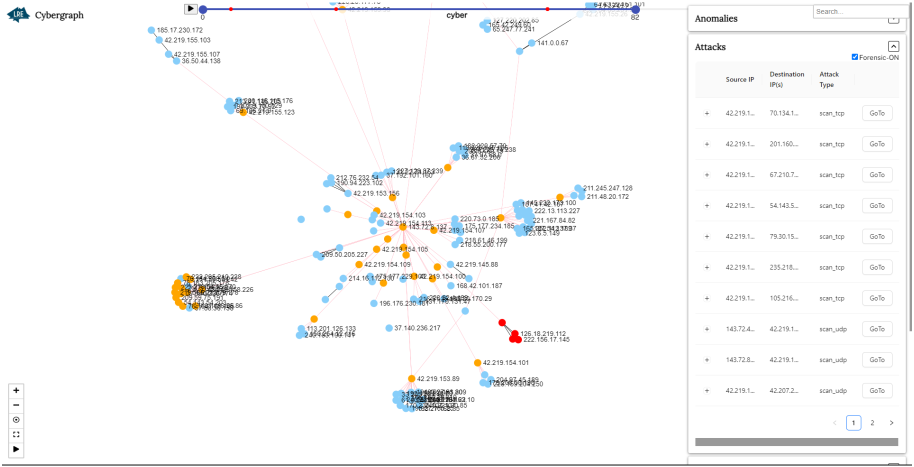

Figure: Cybergraph in forensic mode

Table of Contents

1 Application Demonstration

2 Cybergraph and its Architecture
- Architecture
- Attack patterns
- CI/CD & Deployment

3 Anomalies detection
- Anomalous
- AMAD
- Benchmark

4 Future Works and Conclusion

Cybergraph: Architecture

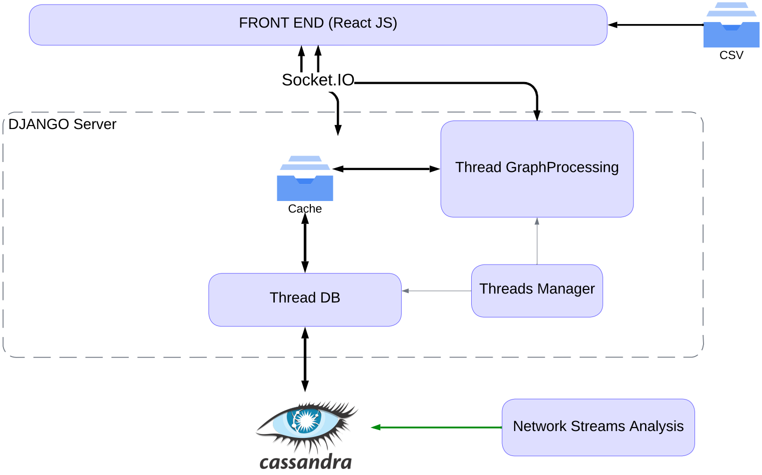

Figure: Cybergraph's structure

Attack patterns

- **detect_dos()**: Detects Denial of Service attacks by identifying unusual high traffic from a single source.
- **detect_ddos()**: Detects Distributed Denial of Service attacks by spotting coordinated high traffic from multiple sources.
- **detect_scan_tcp()**: Detects TCP scanning activities typically used for identifying network vulnerabilities.
- **detect_scan_udp()**: Detects UDP scanning activities which are usually indicative of reconnaissance efforts.

Remark

Graph patterns will be enhanced with the trustseclearn library.

CI/CD & Deployment

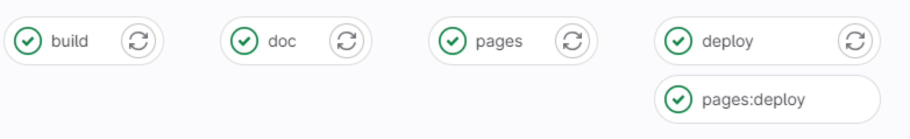

Figure: CI pipeline

Figure: Azure Cosmos DB

heroku

Table of Contents

1. Application Demonstration
2. Cybergraph and its Architecture
   - Architecture
   - Attack patterns
   - CI/CD & Deployment
3. Anomalies detection
   - Anomalous
   - AMAD
   - Benchmark
4. Future Works and Conclusion

Comparing GAD algorithms

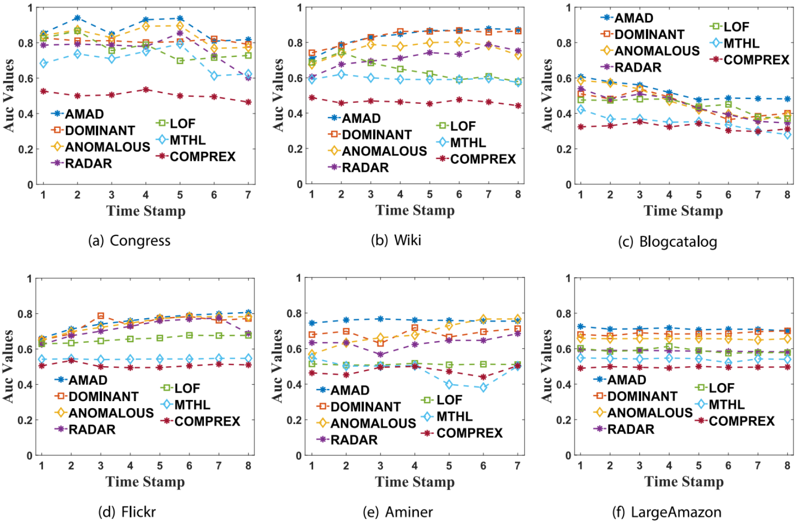

Figure: Time-evolving anomaly detection performance of different methods.[1]

ANOMALOUS: GAD on Attributed Networks

Method: It approximates the original data (attributes and adjacency matrix) through CUR decomposition, which enables more interpretable selection of instances and attributes.
Regularization: Utilizes regularization terms γ (for residuals and network-attribute correlation) and ω (for row and column sparsity of selection matrix).
Result: Ranks anomalies based on residual errors.

ANOMALOUS: GAD on Attributed Networks

Method: It approximates the original data (attributes and adjacency matrix) through CUR decomposition, which enables more interpretable selection of instances and attributes.
Regularization: Utilizes regularization terms γ (for residuals and network-attribute correlation) and ω (for row and column sparsity of selection matrix).
Result: Ranks anomalies based on residual errors.

Complexity

iterations x $(O(n^2 d) + O(n^2))$[2]

ANOMALOUS: GAD on Attributed Networks

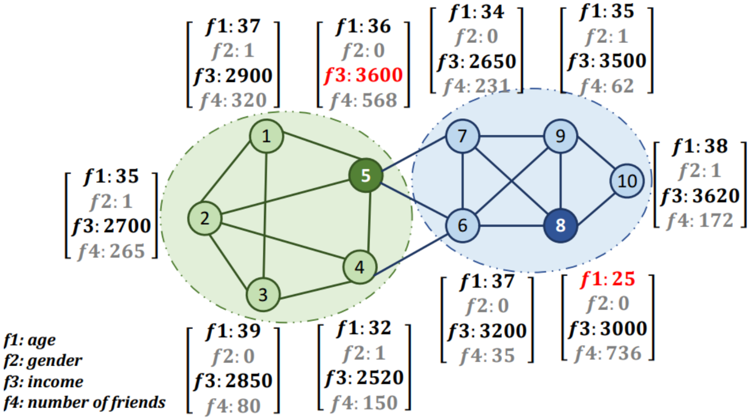

Figure: A toy example for anomaly detection on representative attributes via attribute selection.[2]

AMAD: GAD on time-evolving attributed networks

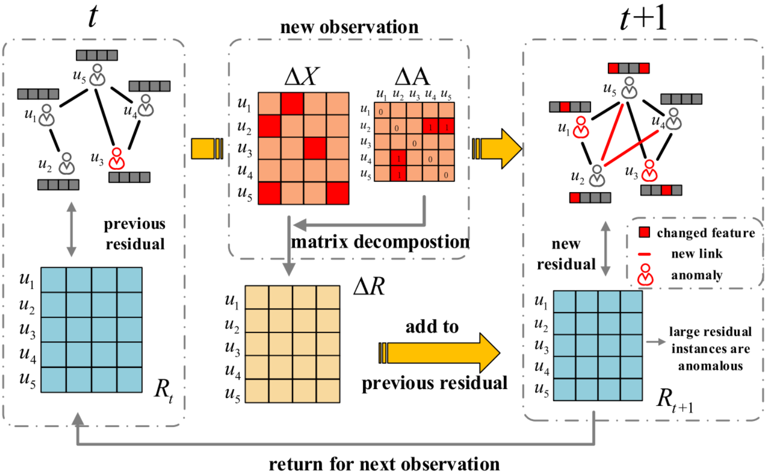

Figure: The workflow of the AMAD method [3]

AMAD Optimization: Incremental Updating and Caching

Incremental Laplacian Matrix Update: Row and column update per new data point, time complexity $O(n^2)$ to $O(n)$.

Incremental W and R Updates: Utilizes updated Laplacian, potential complexity reduction to $O(n)$.

Intermediate Results Caching: Accelerate later iterations, potential $O(1)$ calculations.

AMAD Optimization: Incremental Updating and Caching

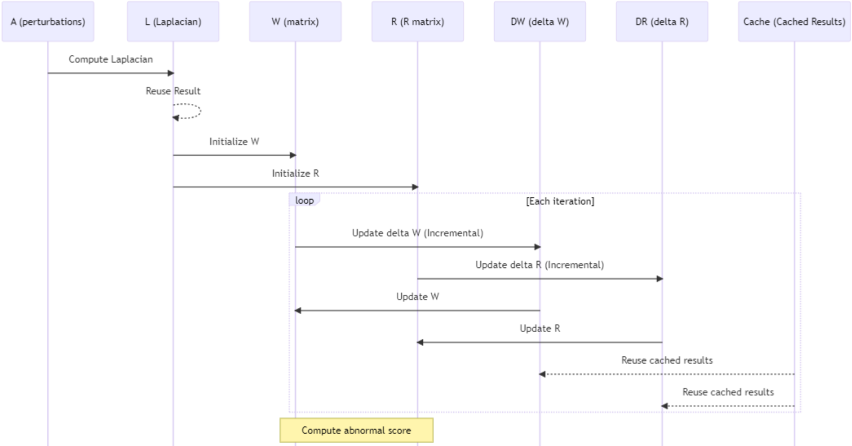

Figure: AMAD method description

Benchmark

	nodes	edges	attributes	anomaly
Wiki	2.405	10.976	4.973	1%
UGR'16 Sample	48219	55742	14	1.33%

Table: Informations about used datasets [3]

	Anomalous	AMAD	AMAD(Opti)
Wiki	0.78	0.82	0.82
UGR'16 Sample	0.52	0.79	0.80

Table: Detection performances, AUC values

	Anomalous	AMAD	AMAD(Opti)	Improvement
Wiki	579.20(s)	147.30(s)	**95.5(s)**	+54,2%
UGR'16 Sample	1542.57(s)	950.49(s)	**652.10(s)**	+45,8%

Table: Average running time

Table of Contents

1. Application Demonstration
2. Cybergraph and its Architecture
   - Architecture
   - Attack patterns
   - CI/CD & Deployment
3. Anomalies detection
   - Anomalous
   - AMAD
   - Benchmark
4. Future Works and Conclusion

Conclusion

- **Cybergraph** and its attack pattern detection functions effectively handle large network traffic datasets.
- **Anomalous** and **AMAD** optimize anomaly detection.
- Incremental updating and caching techniques in AMAD notably reduce computational complexity.

Next steps

- **Cybergraph**
  - Pursue the test phase with bigger datasets
  - Integrate trustseclearn library
  - Allow pattern detectors to be added via the cybergraph's front-end
  - Deploying cybergraph with partners
  - Add GAD algorithms
  - Add a supervisor model, using patterns and anomalies
- **AMAD**
  - Test with different datasets
  - Parallelization
  - Move to C++

Next steps: Supervisor

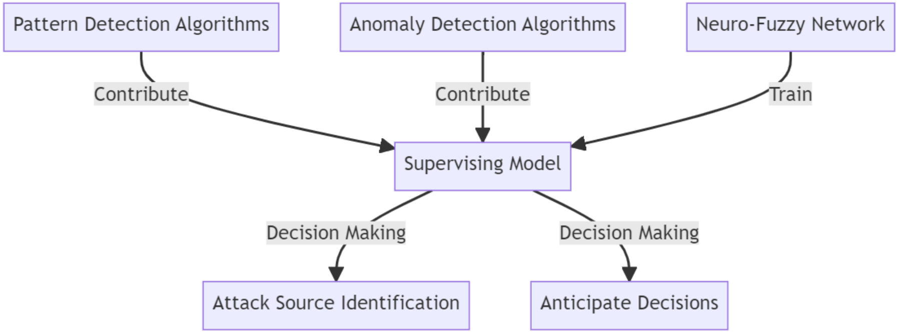

Figure: Supervisor generating attacks decisions structure

References

[1] Stephen Ranshous et al. "Anomaly detection in dynamic networks: a survey". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 7.3 (2015), pp. 223–247.

[2] Zhen Peng et al. "ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks.". In: *IJCAI*. 2018, pp. 3513–3519.

[3] Luguo Xue et al. "An anomaly detection framework for time-evolving attributed networks". In: *Neurocomputing* 407 (2020), pp. 39–49.

Questions

Thank you for your attention!

Questions?