Graph community metrics as a reliable and time robust tool to detect cyber-attacks

Icube - Laboratoire des sciences de l'ingénieur,de l'informatique et de l'imagerie, UMR 7357
Université de Strasbourg, 67000 Strasbourg, France;

Laboratoire de Recherche de L'EPITA (LRE),
14-16 rue Voltaire, 94270 Le Kremlin-Bicètre, France

julien.michel2@etu.unistra.fr

Directed by : Pierre PARREND

Julien MICHEL
21/06/2024

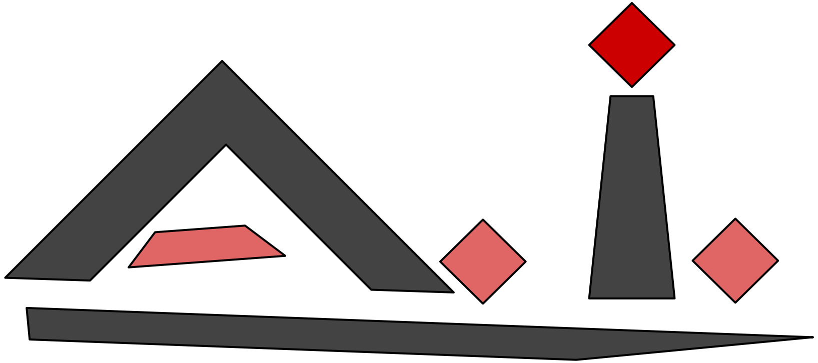
Summary

- Context
- State of the art
- Problems
- Datasets
- Graph community
- Results
- GPML
- Next steps
- Conclusion

Context

BIG DATA :

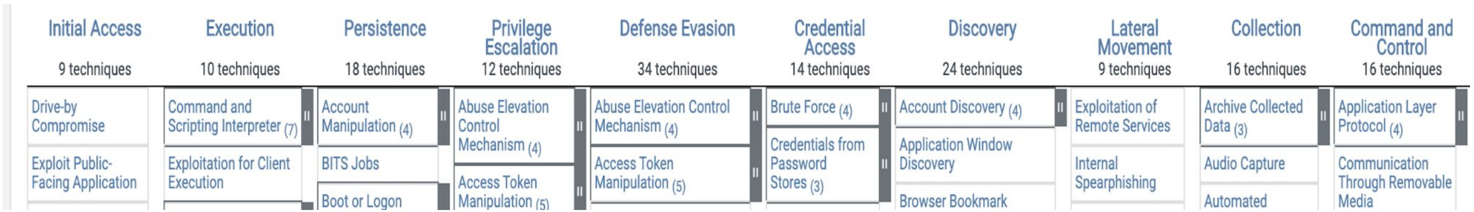
How to manage an ever increasing amount of data ?

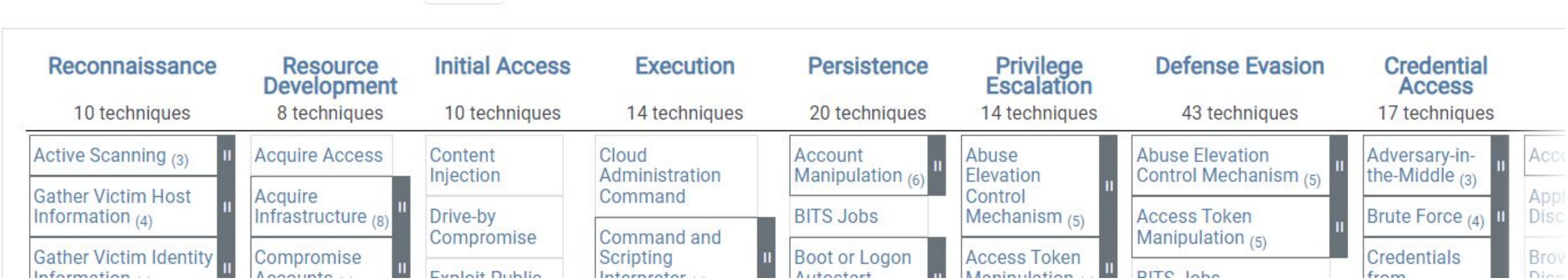


A.I. CHALLENGES :

- **Scalability**
- **Explainability**
- **Time robustness**

Context





Network are changing environment
Attacks are very diverse evolving targets

State of the art

RELATED WORKS FOR ANOMALY DETECTION SURVEY COMPARISON

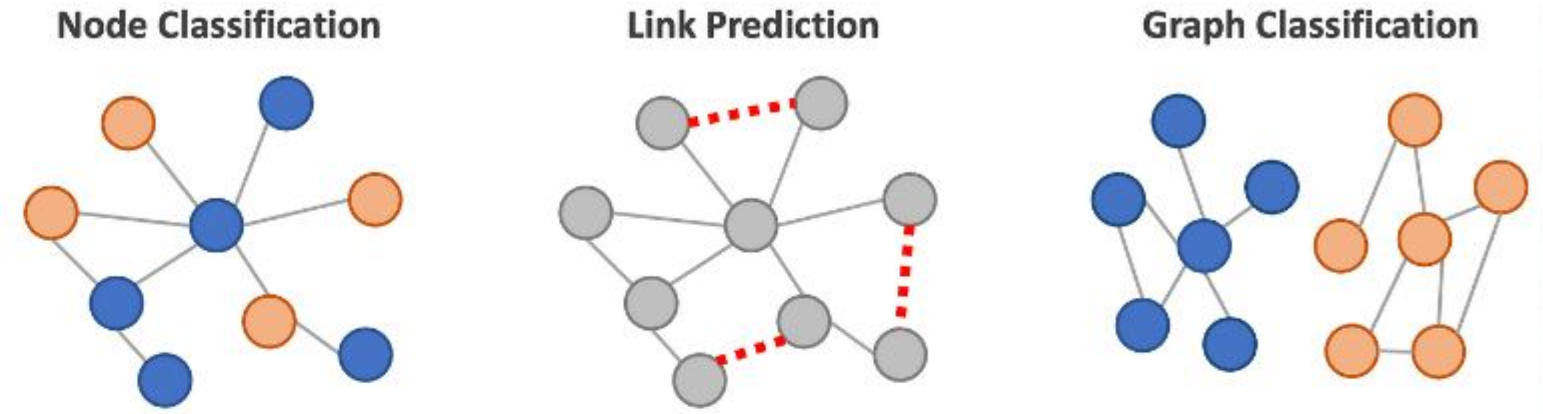
Year →	Before 2020				2020 - 2021					From 2022	
Category \ Papers	Akoglu et al. [3]	Ranshous et al. [4]	Rosetti et al. [5]	Salehi et al. [6]	Magán-Carrión et al. [7]	Pourhabibi et al. [8]	Ma et al. [9]	Nassif et al. [10]	Cook et al. [11]	Chatterjee et al. [12]	Kim et al. [13]
Attack detection	✓	x	x	x	✓	x	✓	✓	x	✓	x
Graph based	✓	✓	✓	✓	x	✓	✓	x	✓	x	✓
Scalability	x	x	✓	x	x	✓	x	x	x	x	x
Dynamicity	✓	✓	✓	✓	x	✓	✓	x	x	x	✓
Time constraint	x	x	x	✓*	x	x	x	x	x	x	x
Time Robustness	x	x	x	x	x	x	x	x	✓	✓	x
Community	✓	✓	✓	✓	x	✓	✓	x	x	x	✓

- Community based approach don’t consider metrics except modularity used for community detection
- Most of the works don’t consider scalability
- None considerer constraints of time such as the one in data stream analysis
- More recent works considered concept drift but have no substantial answer

State of the art : GNN

[15] H. Kim, B. S. Lee, W.-Y. Shin, and S. Lim, “Graph anomaly detection with graph neural networks: Current status and challenges,” IEEE Access, 2022.

- **Very popular**
- **Work with graph structure**
- **Can construct a graph structure from euclidean data -> Embedded prediction to a vector.**



Problems

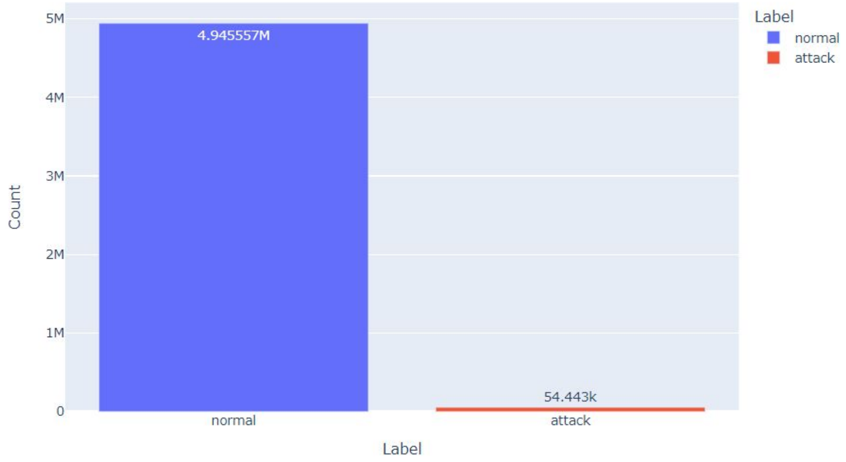
- How to keep a scalable approach ?
- How to be robust to evolution of attacker model ?
- Can explainability be retained ?
- How could poisoning be avoided ?
- Concept drift robustness ?

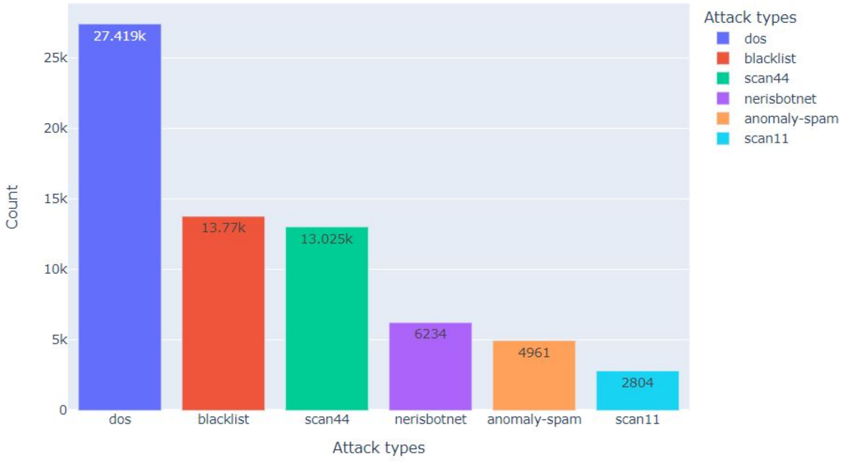
Datasets : UGR16

Date time	Duration	Source IP	Destination IP	Source Port	Destination Port	Protocol	Flag	Forwarding status	ToS	Packets	Bytes	Label
2016-07-27 13:43:29	0.0	143.72.8.137	42.219.158.161	53	43192	UDP	.A....	0	0	1	214	background
2016-07-27 13:43:29	0.0	42.219.154.119	143.72.8.137	60185	53	UDP	.A....	0	0	1	72	background
2016-07-27 13:43:30	0.0	42.219.154.107	143.72.8.137	48598	53	UDP	.A....	0	0	1	77	background
2016-07-27 13:43:30	0.0	42.219.154.98	143.72.8.137	51465	53	UDP	.A....	0	0	1	63	background
2016-07-27 13:43:30	0.0	43.164.49.177	42.219.155.26	80	37934	TCP	.A...F	0	0	1	52	background

- Background data gathered from march to august 2016
- Simulated attacks from the last week of july and august in the background data (DoS and Port Scan)
- Re-inserted some attacks detected using anomaly detection (Spam and Botnet)
- Some unnoticed attacks may still be labelled as background

Datasets : UGR16





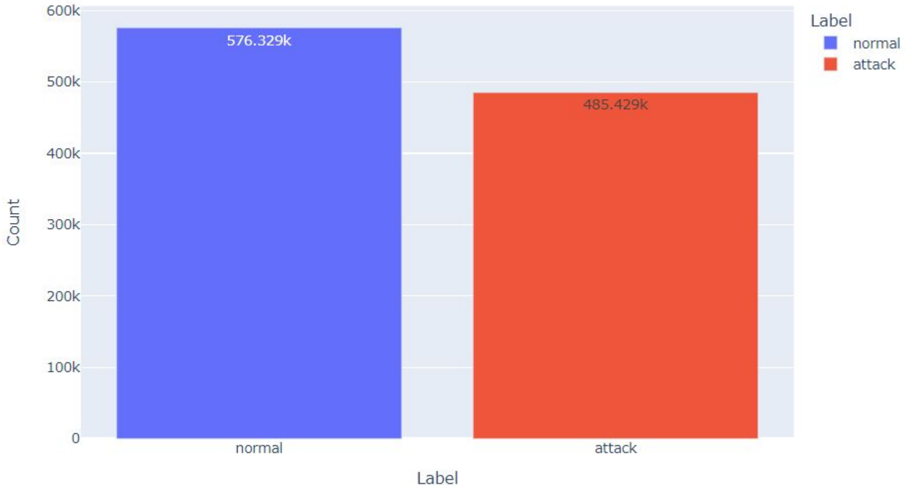
Datasets : Kitsune

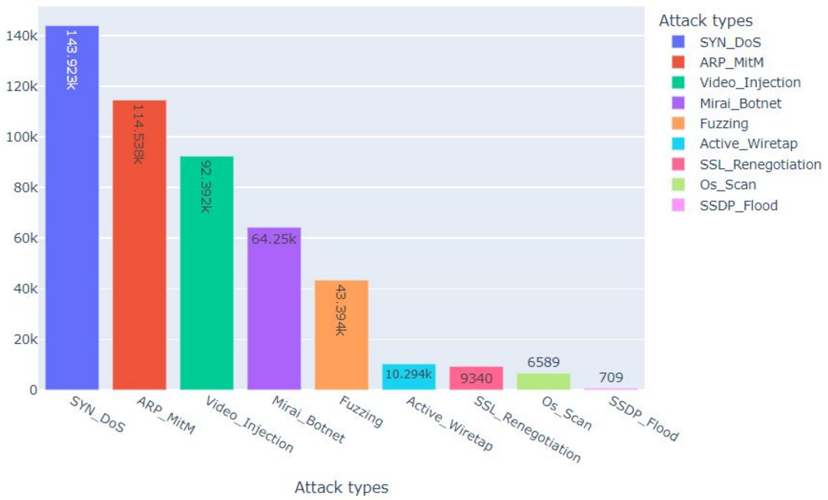
Attack Type	Attack Name	Tool	Description: *The attacker...*	Violation	Vector	# Packets	Time [min.]
Recon.	**OS Scan**	Nmap	*...scans the network for hosts, and their operating systems, to reveal possible vulnerabilities.*	**C**	1	**1,697,851**	52.2
	Fuzzing	SFuzz	*...searches for vulnerabilities in the camera's web servers by sending random commands to their cgis.*	**C**	3	2,244,139	85.5
Man in the Middle	**Video Injection**	Video Jack	*...injects a recorded video clip into a live video stream.*	**C, I**	1	2,472,401	33.4
	ARP MitM	Ettercap	*...intercepts all LAN traffic via an ARP poisoning attack.*	**C**	1	2,504,267	28.2
	Active Wiretap	Raspberry PI 3B	*...intercepts all LAN traffic via active wiretap (network bridge) covertly installed on an exposed cable.*	**C**	2	4,554,925	95.6
Denial of Service	**SSDP Flood**	Saddam	*...overloads the DVR by causing cameras to spam the server with UPnP advertisements.*	**A**	1	4,077,266	40.8
	SYN DoS	Hping3	*...disables a camera's video stream by overloading its web server.*	**A**	1	2,771,276	52.8
	SSL Renegotiation	THC	*...disables a camera's video stream by sending many SSL renegotiation packets to the camera.*	**A**	1	6,084,492	65.6
Botnet Malware	**Mirai**	Telnet	*...infects IoT with the Mirai malware by exploiting default credentials, and then scans for new vulnerable victims network.*	**C, I**	**X**	764,137	118.9

Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: An ensemble of autoencoders for online network intrusion detection,” in The Network and Distributed System Security Symposium (NDSS) 2018

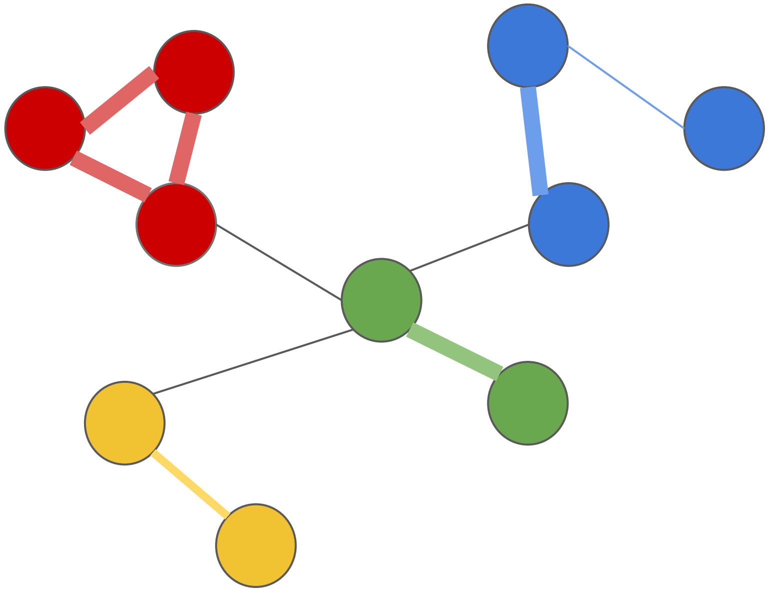
Formatted for ML
Lot of “efficient” features but ….

Datasets : Kitsune





Graph community



Groups of nodes more connected to each others than to the other nodes of the graph.

In general a graph partition is obtained by maximizing the modularity.

$Size_i$ Number of nodes in community i
V_{all} : The number of nodes in the graph

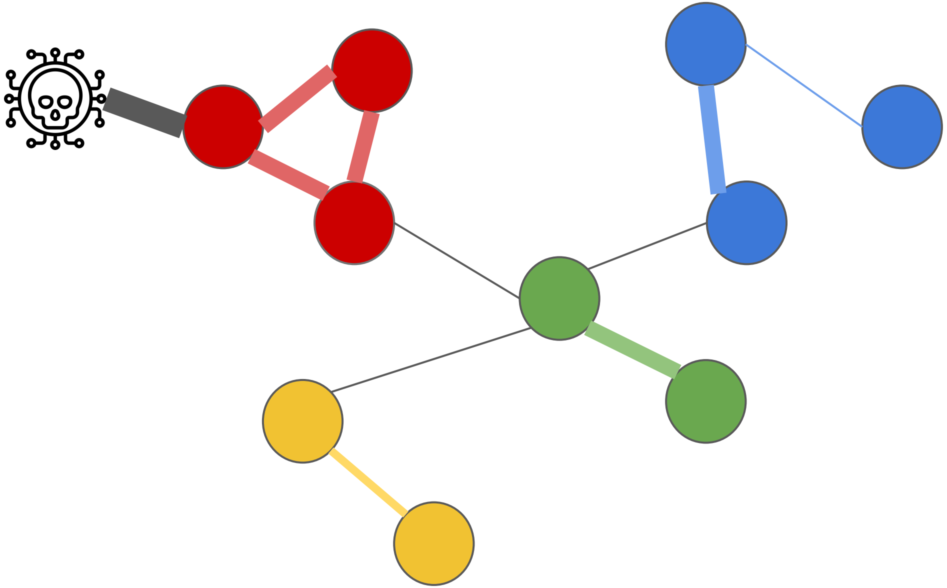
$$= Mod - Cov - \frac{\sum \frac{M_{all}}{V_{all}^2} . Size_i^2}{M_{all}}$$

M_{in} : The number of edge with both vertex in same community
M_{all} : The number of edge in the graph

$$Cov = \frac{M_{in}}{M_{all}}$$

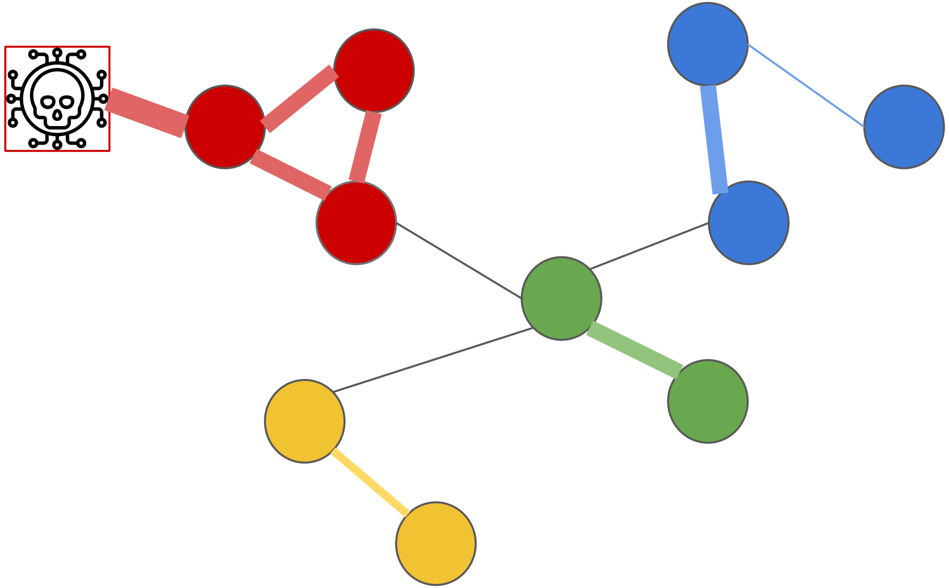
[9] H. S. Pattanayak, H. K. Verma, and A. L. Sangal, “Community detection metrics and algorithms in social networks,” in 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC) 2018, pp. 483–489.

Why Graph community ?



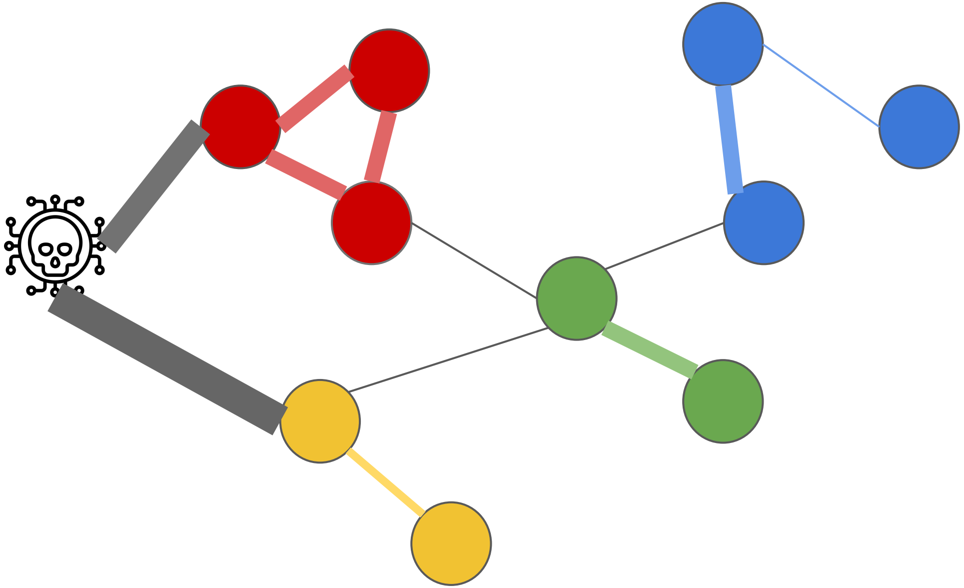
Intense number of communications from the attacker to the target, like port scanning 1to1 or DoS by flooding.

Why Graph community ?



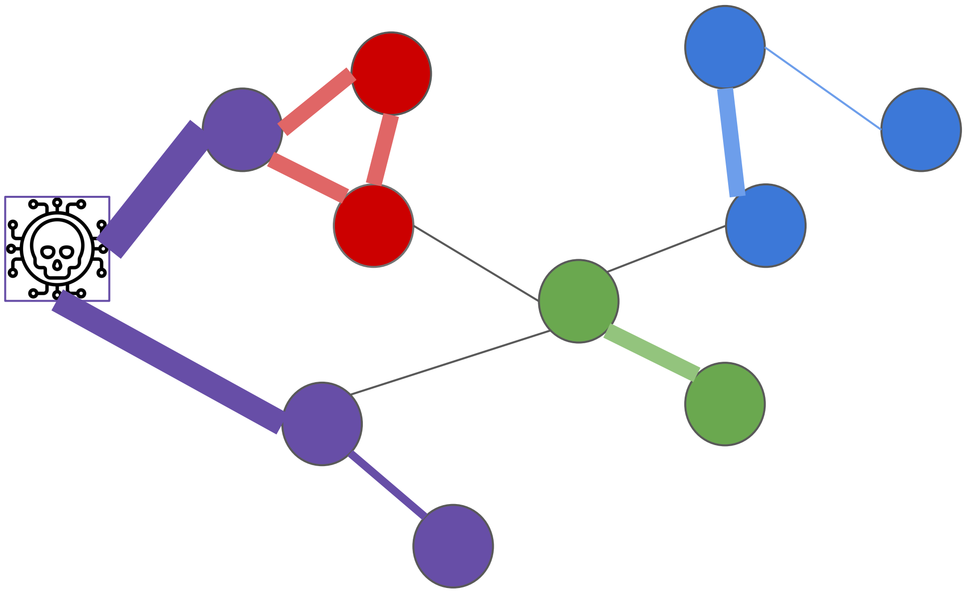
Intense number of communications from the attacker to the target, like **port scanning 1to1** or **DoS by flooding**.

Why Graph community ?



Typically similar to the behavior of a **Man in the Middle type of attack**

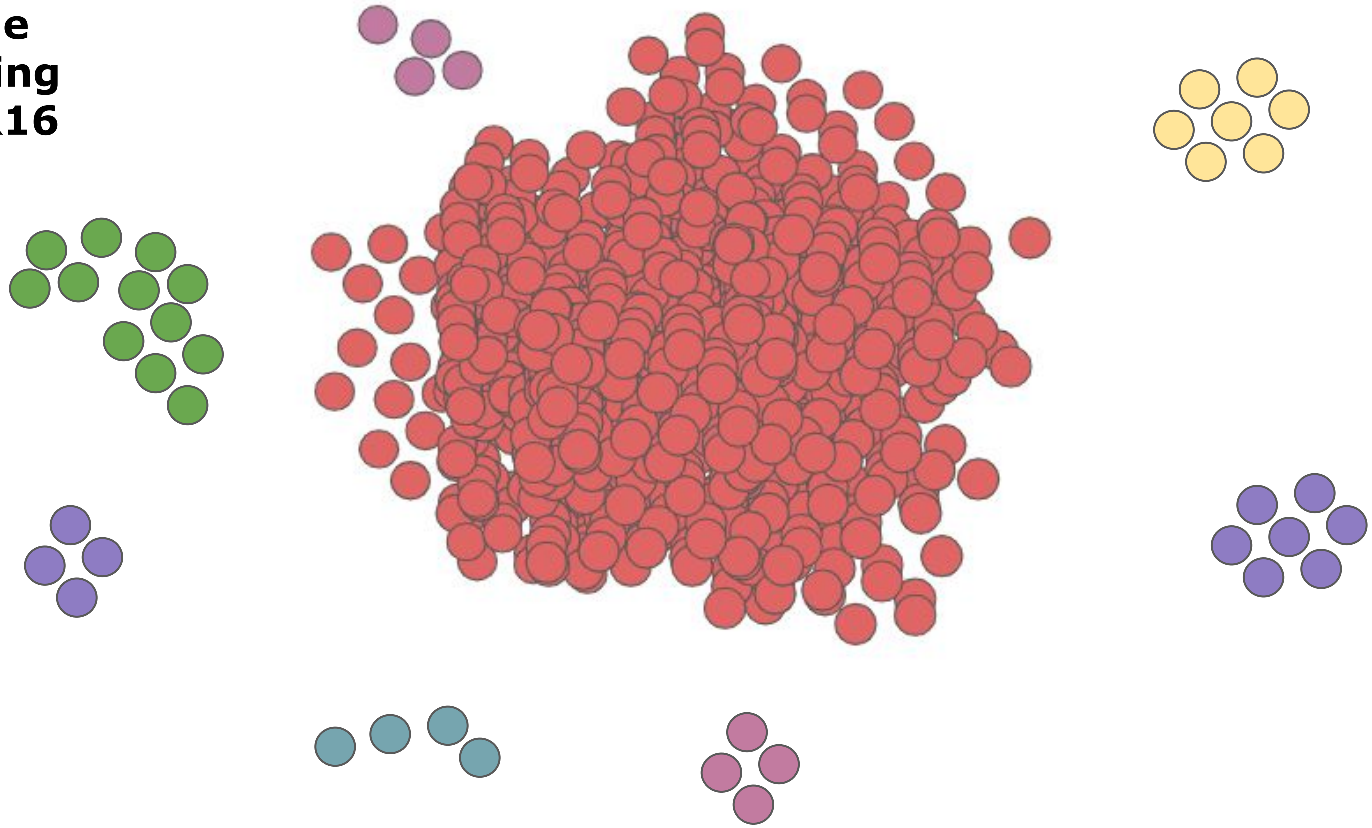
Why Graph community ?



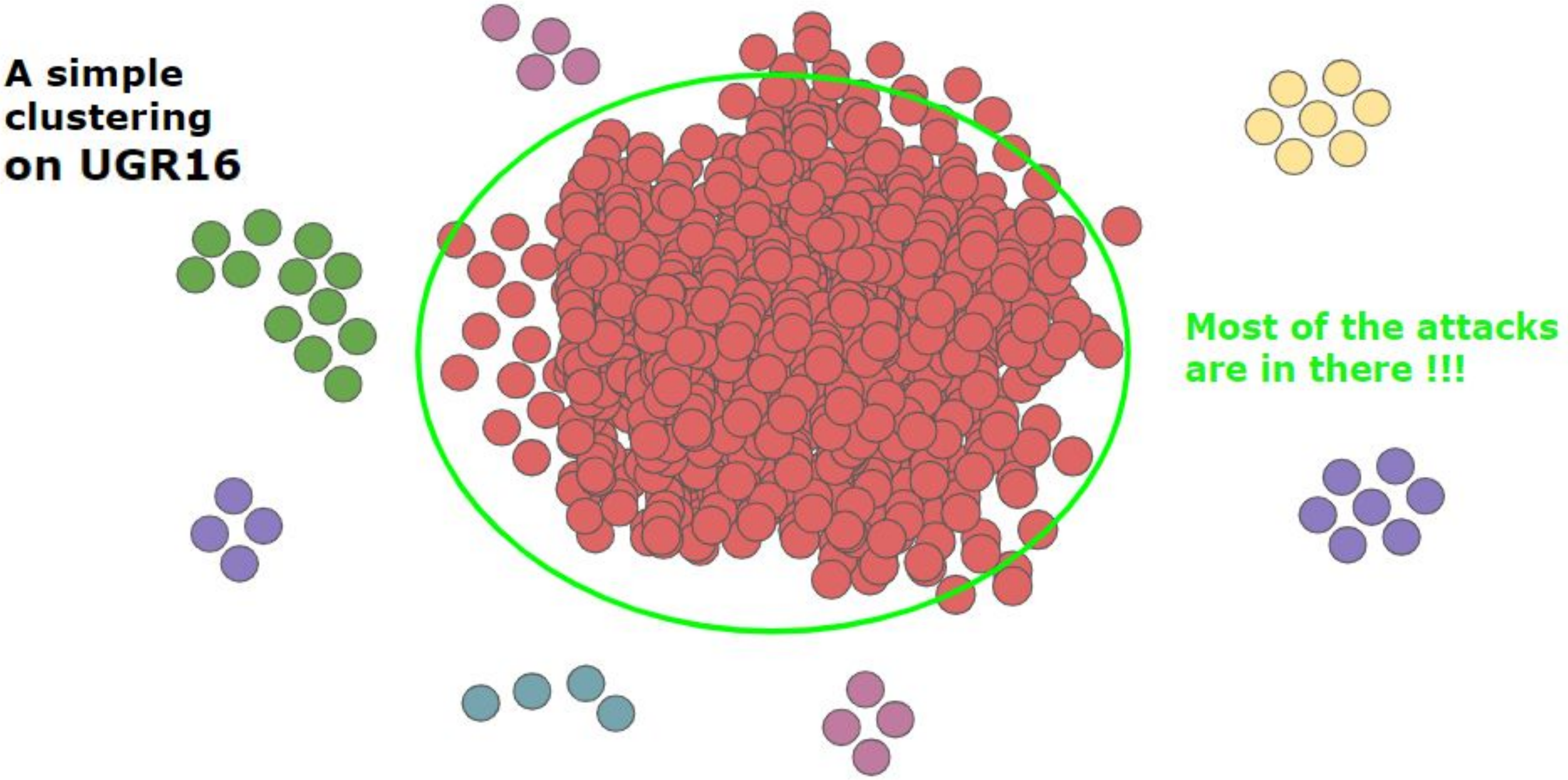
Typically similar to the behavior of a Man in the Middle type of attack

Why Graph community metrics ?

A simple clustering on UGR16

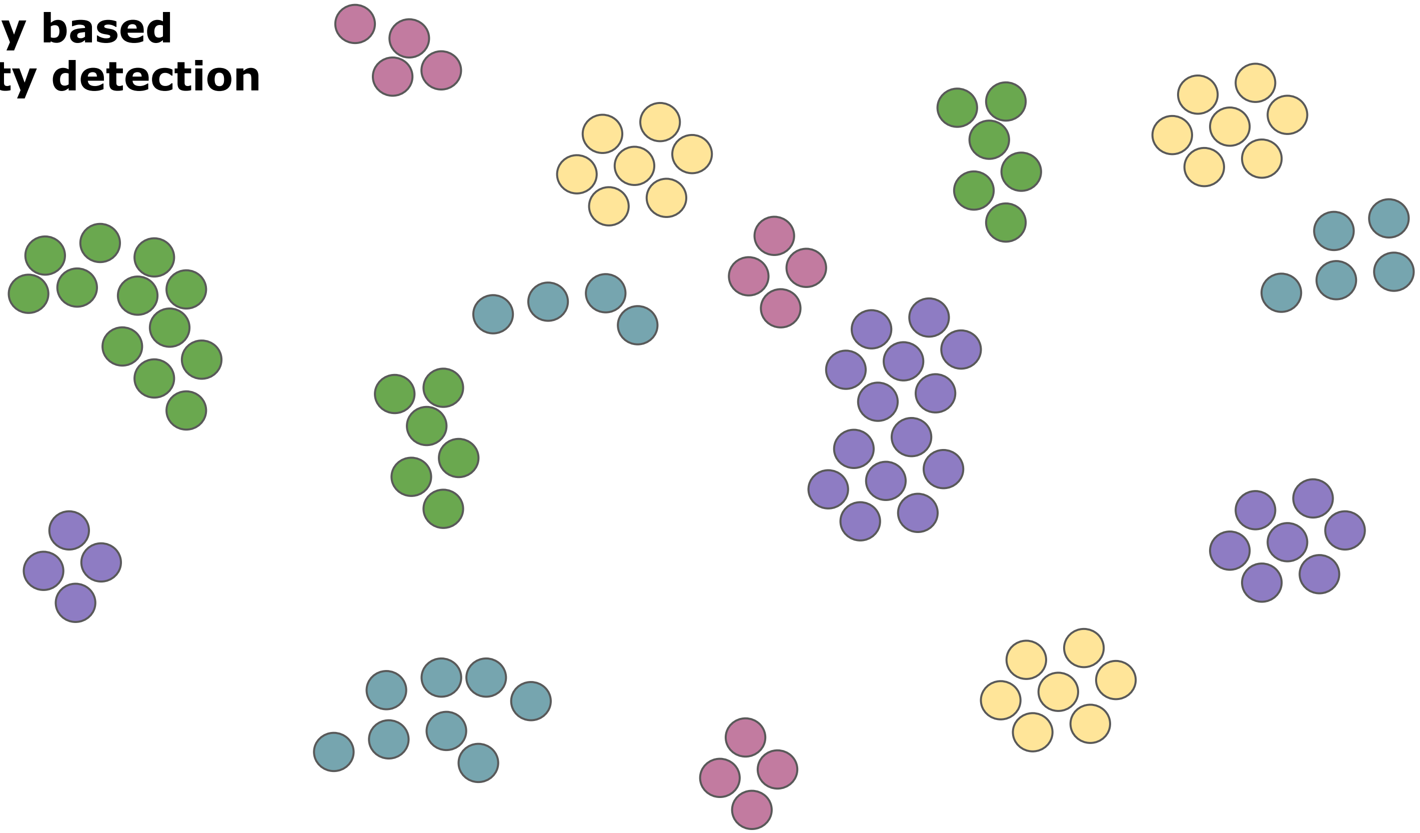


Why Graph community metrics ?



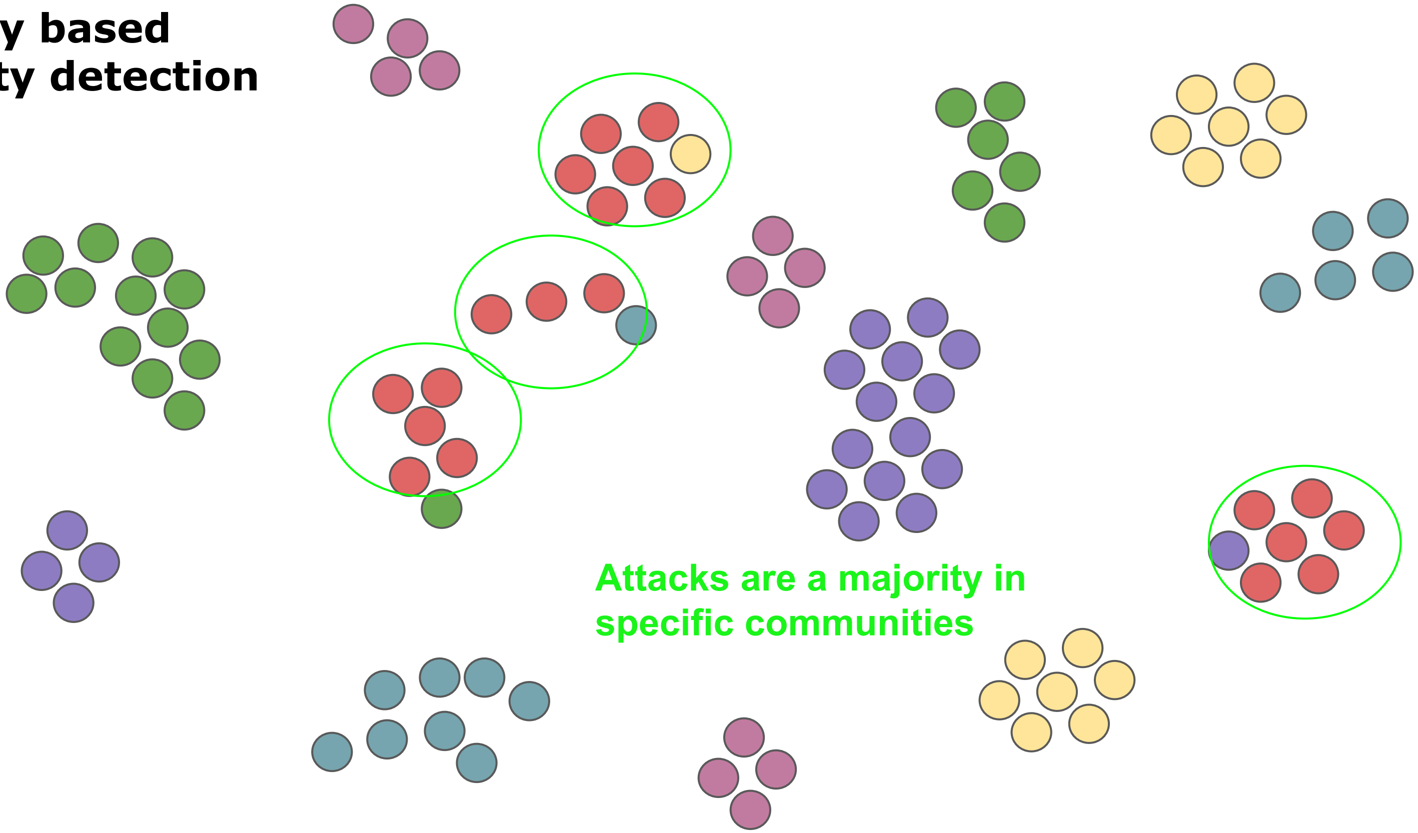
Why Graph community metrics ?

Modularity based community detection

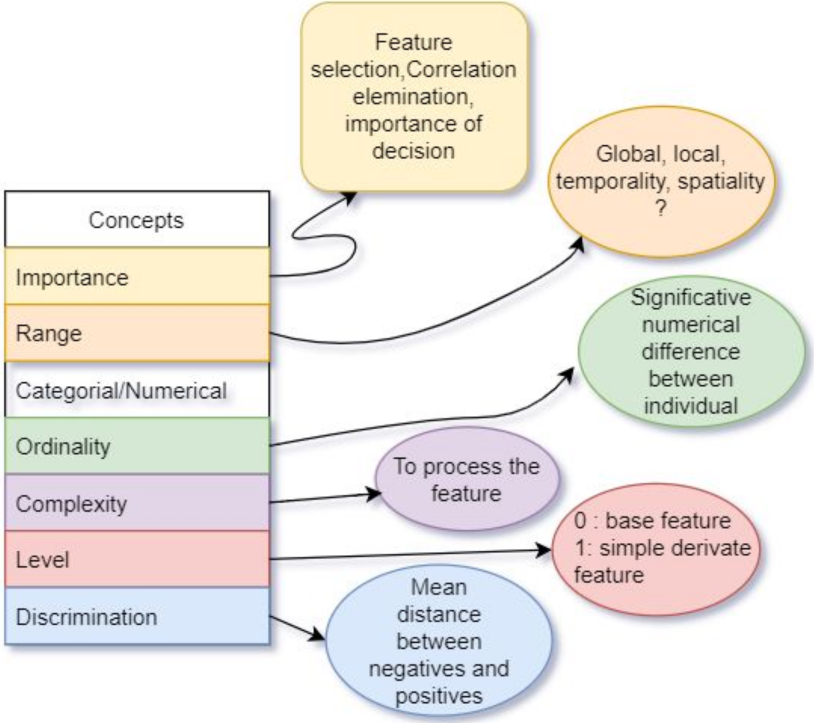


Why Graph community metrics ?

Modularity based community detection



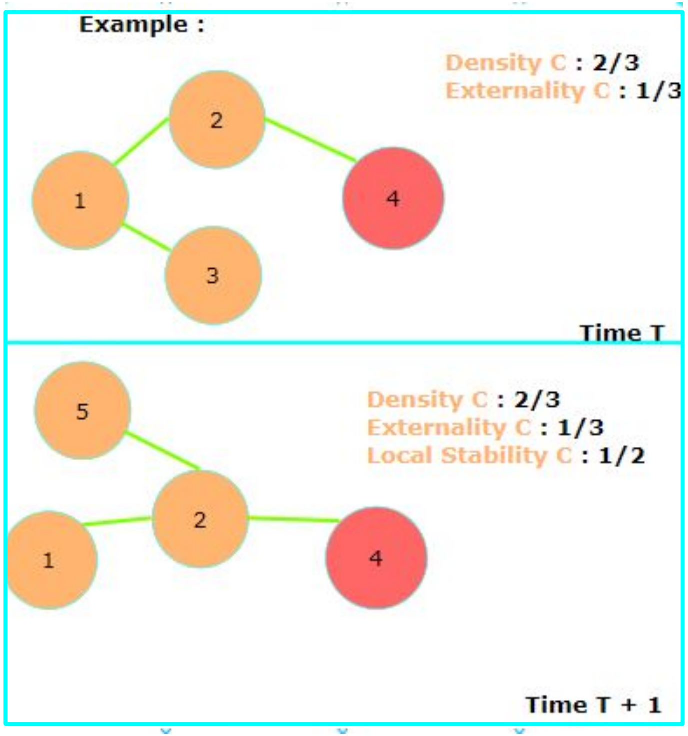
Why Graph community metrics ?



- Features are an important aspect if not the most important in anomalies detection.
- You need to keep only relevant features
- They need to discriminate positive and negative
- They need to be computable in your study case

Why dynamic community metrics ?

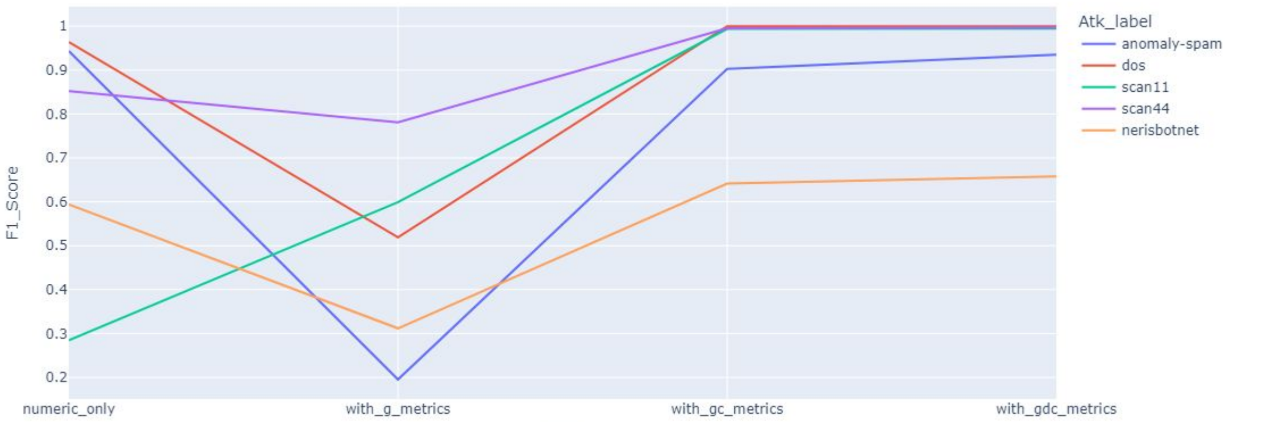
- Few nodes
  or
- Few edges
  can have high impact on community values



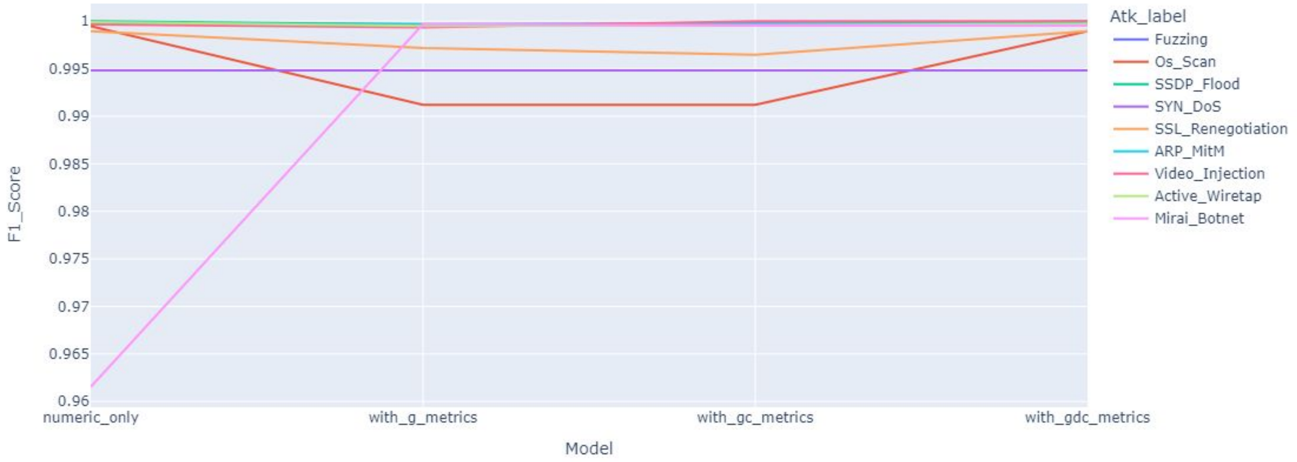
We define Stability as a value of distance between 2 state of the same community.

Results : XGBoost F1-score comparison

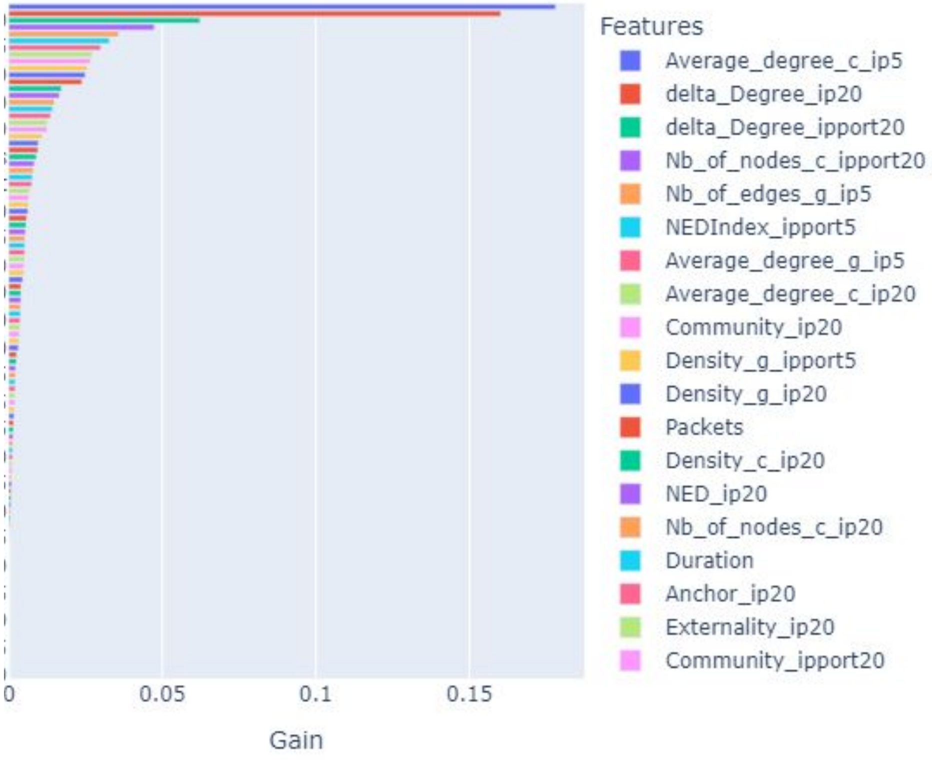
UGR16



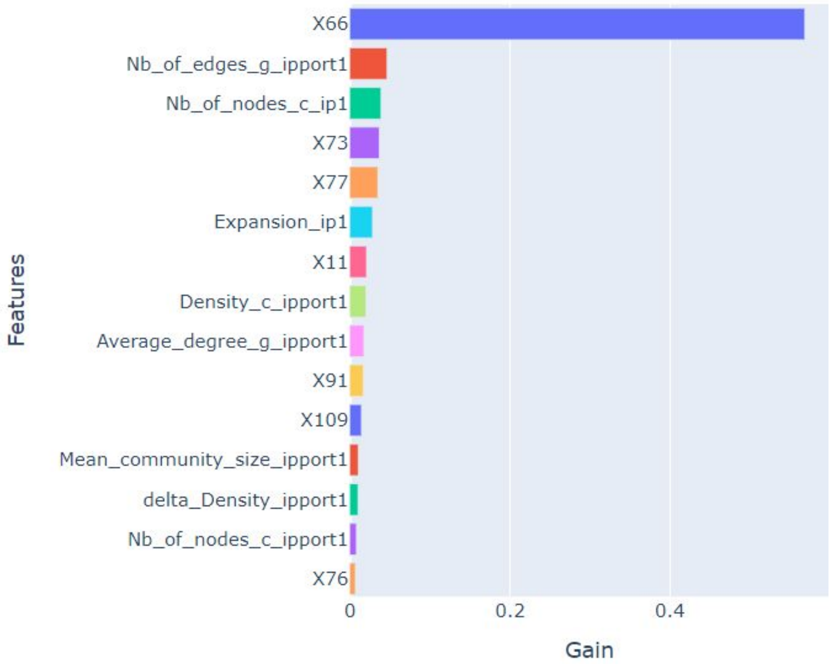
Kitsune



Results: Importance gain

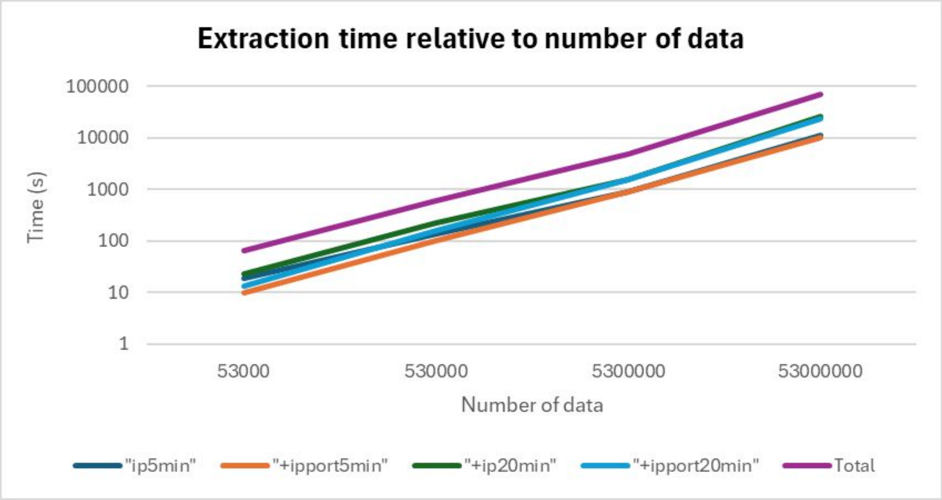


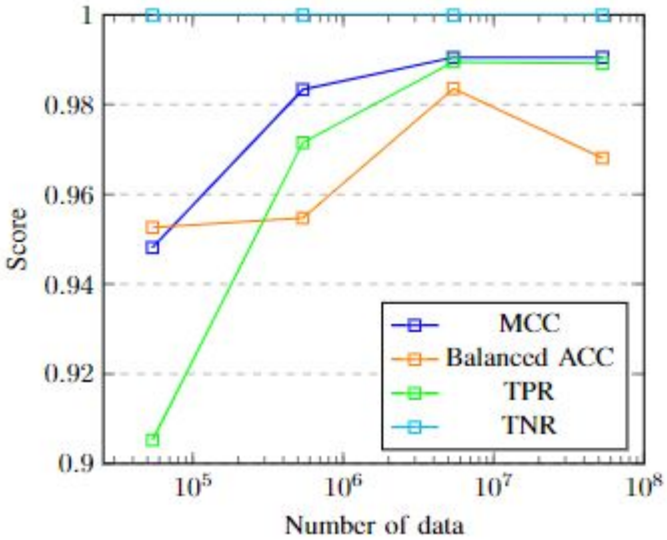
UGR16



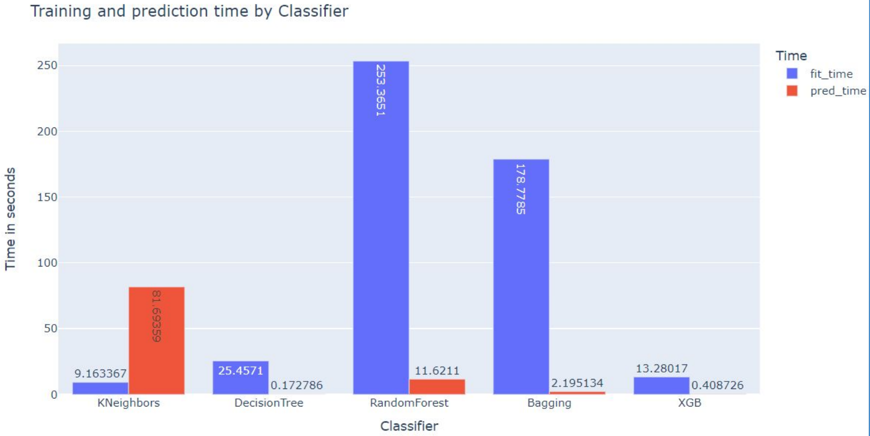
Kitsune

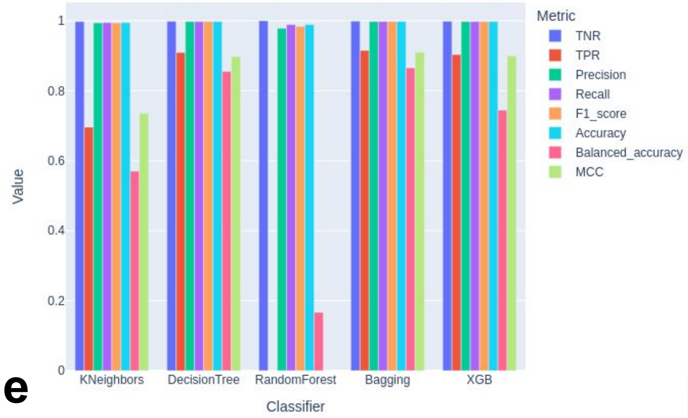
Results : Scalability



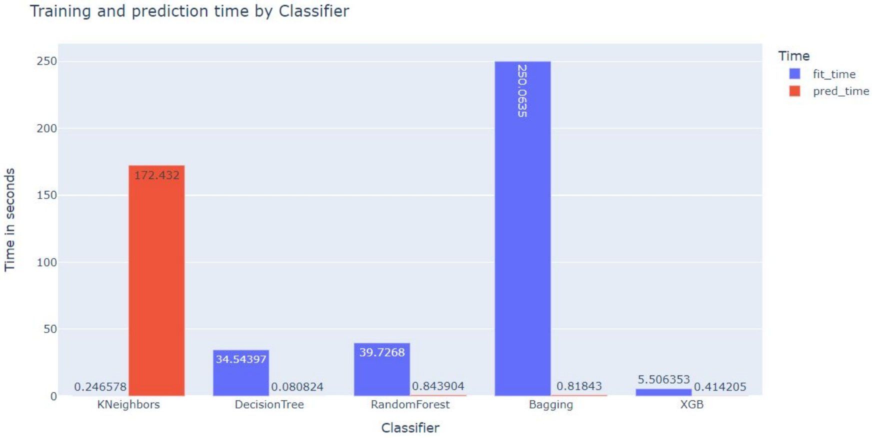


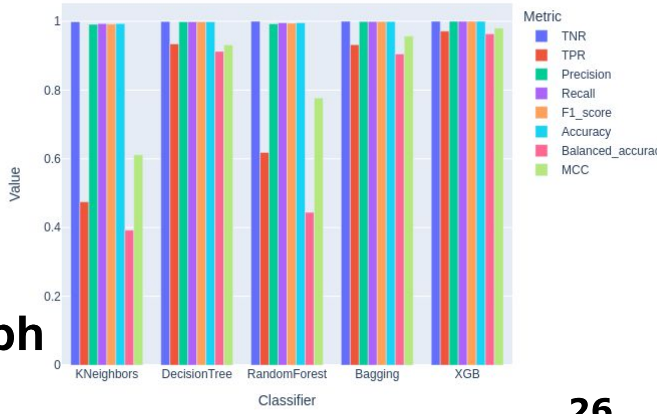
Results : Scalability - UGR16





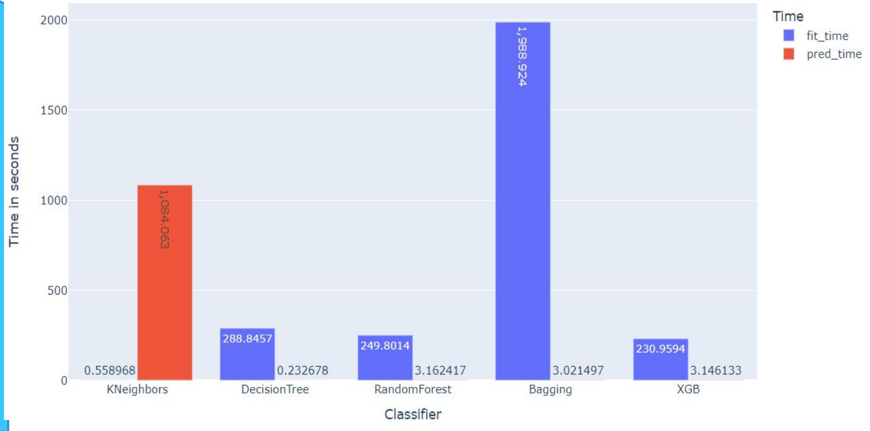
Baseline

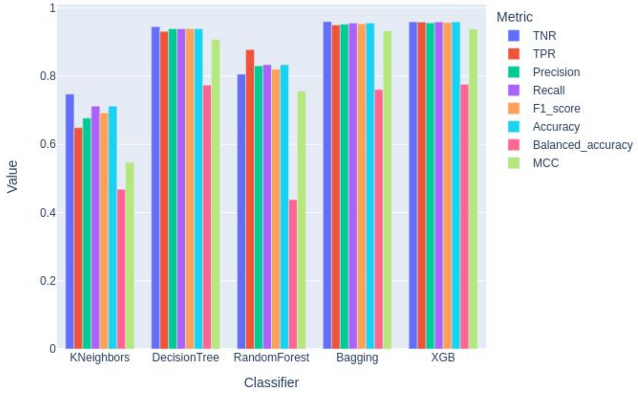




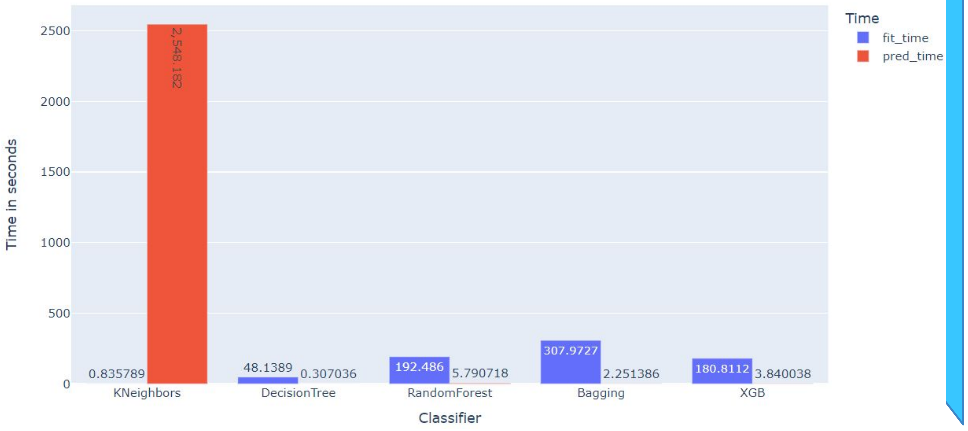
Dynamic graph communities

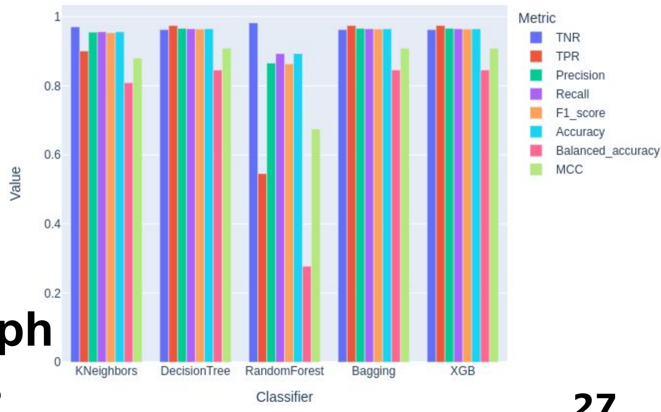
Results : Scalability - Kitsune





Baseline





Dynamic graph communities

Results : Performance comparison

Datasets	Attacks	Precision		Recall		Balance Accurracy		F1-Score		Best Baseline
		Baseline	DGC	Baseline	DGC	Baseline	DGC	Baseline	DGC	
UGR16	Nerisbotnet	0,6875	**0,8457**	**0,6409**	0,5381	0,6442	**0,6919**	**0,6634**	0,6577	Bagging
	Scan11	0,8133	**0,9988**	0,7426	**0,9905**	0,7779	**0,9947**	0,7763	**0,9947**	
	Scan44	0,9239	**0,9992**	0,9332	**0,9956**	0,9286	**0,9974**	0,9286	**0,9974**	
	Spam	0,9608	**0,9814**	**0,927**	0,8924	**0,9439**	0,9369	**0,9436**	0,9348	Xgboost
	DoS	0,9359	**0,9998**	0,9943	**1**	0,9651	**0,9999**	0,9642	**0,9999**	
Kitsune	DoS	**0,9882**	0,7849	**0,9995**	0,587	**0,9943**	0,6859	**0,9943**	0,6716	
	SSL_Renegotiation	0,3571	**0,6984**	0,3694	**0,8371**	0,3632	**0,7678**	0,3631	**0,7615**	CART
	Mirai_botnet	**0,9994**	0,9973	**0,9986**	0,9765	**0,999**	0,9869	**0,999**	0,9868	Bagging
	Active_Wiretap	0,7286	**0,9435**	0,6178	**0,8971**	0,6732	**0,9203**	0,6686	**0,9197**	Xgboost
	Video_injection	**0,946**	0	**0,999**	0	**0,9725**	0	**0,9718**	0	
	ARP_MiTM	**0,9516**	0,9167	**0,9982**	0,9758	**0,9748**	0,9463	**0,9743**	0,9454	
	SSDP_Flood	0,7554	**0,8064**	0,6261	**1**	0,6907	**0,9031**	0,6847	**0,8928**	
	Os_Scan	0,4964	**1**	**0,4931**	0,0026	**0,4948**	0,0052	0,4948	**0,5013**	CART
	Fuzzing	**0,9085**	0,6037	0,9004	**0,9095**	**0,9045**	0,7566	**0,9045**	0,7257	Xgboost

- For UGR16, DGC use both base features and dgc features
- For Kitsune, DGC use only graph features

Grinsztajn, Léo, Edouard Oyallon, and Gaël Varoquaux. "Why do tree-based models still outperform deep learning on typical tabular data?." Advances in neural information processing systems 35 (2022): 507-520.

Results : Optimisation ?

- **Simple lasso for features selection**
- **RandomSearch for hyperparameter tuning**

Graph Processing for Machine Learning

Algorithm 1 Community propagation algorithm

Require: $G1, G2$ {Two graphs}
Require: C_{G1}, C_{G2} {List of centers in G1 and G2}
Require: $Index_N \in G1 = Index_N \in G2$

$Center\_Where \leftarrow []$ {Void list for center position}
$Not\_in \leftarrow |C_{G1}|$
for $i \in C_{G2}$ **do**
  if $i \in G1$ **then**
    $Center\_Where \cup i.community \in G1$
  else
    $Center\_Where \cup Not\_in$
    $Not\_in \leftarrow Not\_in + 1$
  end if
end for
for $N \in G2$ **do**
  $N.old\_community \leftarrow N.community$
  $N.community \leftarrow N.community \in Center\_Where$
end for

Ensure: $G2$ {G2 is updated with propagated communities}

- Better accessibility for graph data for about any dataset
- Dynamic community specific algorithm
- General tool for visualisation of network data for machine learning

https://github.com/lre-security-systems-team/gpml

Next steps

Concept drift :

The characteristics of the target you are trying to detect are changing with passing time and this target is itself in an environment that is evolving with passing time

Feature 1 = x
Feature 2 = y
Feature 3 = z

Is attack
Can bite

Feature 1 = w
Feature 2 = y
Feature 3 = nz

Isn’t attack
Don’t bite

Next steps

We can decide to make rules :

Feature 1 = x Feature 2 = y Feature 3 = z	Is attack Can bite

→ Feature 3 = z then is attack

Next steps

~~Feature 3 = z then is attack~~

The problem is that at any point in time :

Feature 1 = x
Feature 2 = y
Feature 3 = z

Is attack
Can bite

→

Feature 1 = x
Feature 2 = y
Feature 3 = nz

Is attack
Can bite

Feature 1 = w
Feature 2 = y
Feature 3 = nz

Isn't attack
Don't bite

→

Feature 1 = w
Feature 2 = y
Feature 3 = z

Isn't attack
Don't bite

Next steps

Then what we are looking for :

Feature 1 = x
Feature 2 = y
Feature 3 = z

Is attack
Can bite

Feature 1 = w
Feature 2 = y
Feature 3 = nz

Isn't attack
Don't bite

are features that may not be visible on data at first glance but are property of attack models

Conclusion

Getting good features is very important to detection !

Graph community metrics seems relevant to the detection of cyber attacks

Dynamic graph community metrics have shown to be highly important features to detection

In particular some metrics have shown to be relevant for different datasets and type of attacks

An approach which fulfill the constraint of scalability has been set up

Thank you !

[1] W. Robertson, G. Vigna, C. Kr¨ugel, and R. Kemmerer, “Using generalization and characterization techniques in the anomaly-based detection of web attacks.” in NDSS, 01 2006.
[2] T. Zoppi, A. Ceccarelli, T. Capecchi, and A. Bondavalli, “Unsupervised anomaly detectors to detect intrusions in the current threat landscape,” ACM/IMS Trans. Data Sci., vol. 2, no. 2, apr 2021. [Online]. Available: https://doi.org/10.1145/3441140
[3] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, “Anomaly detection in dynamic networks: a survey,” WIREs Computational Statistics, vol. 7, no. 3, pp. 223–247, 2015. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1347
[4] R. Paudel, T. Muncy, and W. Eberle, “Detecting dos attack in smart home iot devices using a graph-based approach,” in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 5249–5258.
[5] E. Navruzov and A. Kabulov, “Detection and analysis types of ddos attack,” in 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). IEEE, 2022, pp. 1–7.
[6] C. Regan, M. Nasajpour, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and K.-K. R. Choo, “Federated iot attack detection using decentralized edge data,” Machine Learning with Applications, vol. 8, p. 100263, 2022.
[7] X. Tao, Y. Peng, F. Zhao, P. Zhao, and Y. Wang, “A parallel algorithm for network traffic anomaly detection based on isolation forest,” International Journal of Distributed Sensor Networks, vol. 14, no. 11, p. 1550147718814471, 2018.
[8] L. Akoglu, H. Tong, and D. Koutra, “Graph based anomaly detection and description: a survey,” Data mining and knowledge discovery, vol. 29, pp. 626–688, 2015.
[9] G. Rossetti and R. Cazabet, “Community discovery in dynamic networks: a survey,” ACM computing surveys (CSUR), vol. 51, no. 2, pp. 1–37, 2018.
[10] M. Salehi and L. Rashidi, “A survey on anomaly detection in evolving data: [with application to forest fire risk prediction],” ACM SIGKDD Explorations Newsletter, vol. 20, no. 1, pp. 13–23, 2018.
[11] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, “Fraud detection: A systematic literature review of graph-based anomaly detection approaches,” Decision Support Systems, vol. 133, p. 113303, 2020.
[12] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, “A comprehensive survey on graph anomaly detection with deep learning,” IEEE Transactions on Knowledge and Data Engineering, 2021.
[13] A. A. Cook, G. Mısırlı, and Z. Fan, “Anomaly detection for iot time-series data: A survey,” IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6481–6494, 2019.
[14] A. Chatterjee and B. S. Ahmed, “Iot anomaly detection methods and applications: A survey,” Internet of Things, vol. 19, p. 100568, 2022.
[15] H. Kim, B. S. Lee, W.-Y. Shin, and S. Lim, “Graph anomaly detection with graph neural networks: Current status and challenges,” IEEE Access, 2022.
[16] K. Liu, Y. Dou, Y. Zhao, X. Ding, X. Hu, R. Zhang, K. Ding, C. Chen, H. Peng, K. Shu et al., “Bond: Benchmarking unsupervised outlier node detection on static attributed graphs,” Advances in Neural Information Processing Systems, vol. 35, pp. 27 021–27 035, 2022.
[17] S. B. Park, H. J. Jo, and D. H. Lee, “G-idcs: Graph-based intrusion detection and classification system for can protocol,” IEEE Access, 2023.
[18] L. Leichtnam, E. Totel, N. Prigent, and L. M´e, “Sec2graph: Network attack detection based on novelty detection on graph structured data,” in Detection of Intrusions and Malware, and Vulnerability Assessment: 17th International Conference, DIMVA 2020, Lisbon, Portugal, June 24–26, 2020, Proceedings 17. Springer, 2020, pp. 238–258.
[19] T. Liu, Z. Li, H. Long, and A. Bilal, “Nt-gnn: Network traffic graph for 5g mobile iot android malware detection,” Electronics, vol. 12, no. 4, p. 789, 2023.
[20] Y. Chen, Z. Ye, H. Zhao, Y. Wang et al., “Feature-based graph backdoor attack in the node classification task,” International Journal of Intelligent Systems, vol. 2023, 2023.

[21] A. Bojchevski and S. G¨unnemann, “Adversarial attacks on node embeddings via graph poisoning,” in International Conference on Machine Learning. PMLR, 2019, pp. 695–704.
[22] M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” Physical review E, vol. 69, no. 2, p. 026113, 2004.
[23] M. Rosvall, D. Axelsson, and C. T. Bergstrom, “The map equation,” The European Physical Journal Special Topics, vol. 178, no. 1, pp. 13–23, 2009.
[24] D. A. Huffman, “A method for the construction of minimum-redundancy codes,” Proceedings of the IRE, vol. 40, no. 9, pp. 1098–1101, 1952.
[25] J. M. Kumpula, M. Kivel¨a, K. Kaski, and J. Saram¨aki, “Sequential algorithm for fast clique percolation,” Physical review E, vol. 78, no. 2, p. 026109, 2008.
[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” Journal of Statistical Mechanics: Theory and Experiment, vol. 2008, no. 10, p. P10008, oct 2008. [Online]. Available: https://dx.doi.org/10.1088/1742-5468/2008/10/P10008
[27] V. A. Traag, L. Waltman, and N. J. Van Eck, “From louvain to leiden: guaranteeing well-connected communities,” Scientific reports, vol. 9, no. 1, p. 5233, 2019.
[28] N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” Physical review. E, Statistical, nonlinear, and soft matter physics, vol. 76, p. 036106, 10 2007.
[29] P. Pons and M. Latapy, “Computing communities in large networks using random walks,” in Computer and Information Sciences - ISCIS 2005, p. Yolum, T. G¨ung¨or, F. G¨urgen, and C. ¨Ozturan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 284–293.
[30] P. Mane, S. Shanbhag, T. Kamath, P. Mackey, and J. Springer, “Analysis of community detection algorithms for large scale cyber networks,” 2016.
[31] K. M. Carter, N. Idika, and W. W. Streilein, “Probabilistic threat propagation for malicious activity detection,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 2940–2944.
[32] J.-H. Park and H.-Y. Kwon, “Cyberattack detection model using community detection and text analysis on social media,” ICT Express, vol. 8, no. 4, pp. 499–506, 2022.
[33] J. Jia, Z. Dong, J. Li, and J. W. Stokes, “Detection of malicious dns and web servers using graph-based approaches,” in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 2625–2629.
[34] R. Francisquini, A. C. Lorena, and M. C. Nascimento, “Community-based anomaly detection using spectral graph filtering,” Applied Soft Computing, vol. 118, p. 108489, 2022.
[35] G. Rossetti, “Rdyn: graph benchmark handling community dynamics,” Journal of Complex Networks, vol. 5, no. 6, pp. 893–912, 2017.
[36] C. Vehlow, F. Beck, P. Auw¨arter, and D. Weiskopf, “Visualizing the evolution of communities in dynamic graphs,” in Computer Graphics Forum, vol. 34, no. 1. Wiley Online Library, 2015, pp. 277–288.
[37] C. Fu, Q. Li, K. Xu, and J. Wu, “Point cloud analysis for ml-based malicious traffic detection: Reducing majorities of false positive alarms,” in Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, pp. 1005–1019.
[38] E. Altinisik, F. Deniz, and H. T. Sencar, “Provg-searcher: A graph representation learning approach for efficient provenance graph search,” in Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, pp. 2247–2261.
[39] B. Wang and N. Z. Gong, “Attacking graph-based classification via manipulating the graph structure,” in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2023–2040.

[40] Y. Zhu, Y. Lai, K. Zhao, X. Luo, M. Yuan, J. Ren, and K. Zhou, “Binarizedattack: Structural poisoning attacks to graph-based anomaly detection,” in 2022 IEEE 38th International Conference on Data Engineering (ICDE). Los Alamitos, CA, USA: IEEE Computer Society, may 2022, pp. 14–26. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICDE53745.2022.00006
[41] L. Akoglu, M. McGlohon, and C. Faloutsos, “Oddball: Spotting anomalies in weighted graphs,” in Advances in Knowledge Discovery and Data Mining: 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part II 14. Springer, 2010, pp. 410–421.
[42] M. Q. Pasta and F. Zaidi, “Topology of complex networks and performance limitations of community detection algorithms,” IEEE Access, vol. 5, pp. 10 901–10 914, 2017.
[43] R. Tibshirani, “Regression shrinkage and selection via the lasso,” Journal of the Royal Statistical Society Series B: Statistical Methodology, vol. 58, no. 1, pp. 267–288, 1996.
[44] H. S. Pattanayak, H. K. Verma, and A. L. Sangal, “Community detection metrics and algorithms in social networks,” in 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018, pp. 483–489.
[45] J. Yang and J. Leskovec, “Defining and evaluating network communities based on ground-truth,” in Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, ser. MDS ’12. New York, NY, USA: Association for Computing Machinery, 2012. [Online]. Available: https://doi.org/10.1145/2350190.2350193
[46] M. K. Rahman, “Nedindex: A new metric for community structure in networks,” in 2015 18th International Conference on Computer and Information Technology (ICCIT). IEEE, 2015, pp. 76–81.
[47] G. Maci´a-Fern´andez, J. Camacho, R. Mag´an-Carri´on, P. Garc´ıa-Teodoro, and R. Ther´on, “Ugr’16 dataset.”
[48] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: An ensemble of autoencoders for online network intrusion detection,” in The Network and Distributed System Security Symposium (NDSS) 2018, 2018.