Testing Abnormality of a sequence of graphs: Application to Cybersecurity

Clarisse Boinay[1], Christophe Biernacki[2],
Cristian Preda & Thomas Anglade[3]

[1] *Inria and Seckiot*
clarisse.boinay@seckiot.fr
[2] *Inria*
christophe.biernacki@inria.fr
[3] *Inria and Lille University*
cristian.preda@inria.fr
[4] *Seckiot*
thomas.anglade@seckiot.fr

SECKIOT

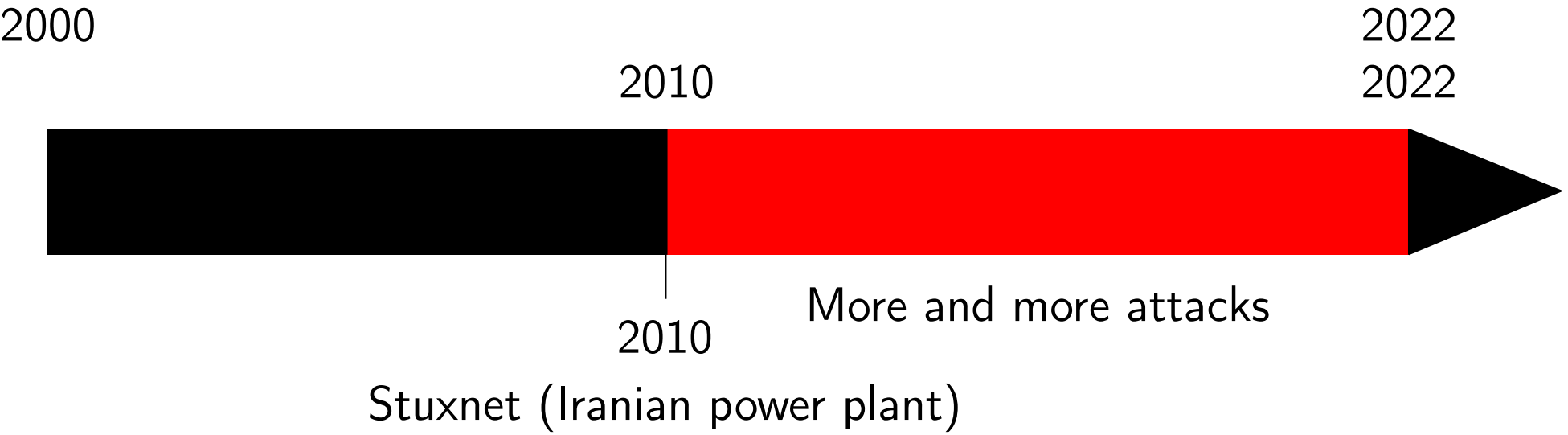
Outline

Operational Technology (OT)

- Part of modern Critical Infrastructures (CI) such as water treatment plants, oil refineries, power grids, and nuclear and thermal power plants
- Composed of sensors and actuators, PLCs, SCADA and HMI

OT aim at controlling the physical process of the firm whereas IT (Information Technology) is all the technologies of information which support all the other processes of the firm. Sometimes there is IT in OT to deal with the information generated by the industrial network.

Attacks in OT: Stuxnet a game-changer



A cyber attack is a malicious attempts to damage, disrupt or gain unauthorised access to computer systems, networks or devices, via cyber means according to the National Cyber Security Centre

Standard approaches to detect an attack

- Solutions in IT not sufficient to stop OT attacks (Raman, Ahmed et Mathur 2021)
- OT Solution based on signature of the previous attacks in the firms (Umer et al. 2022)
- Anomaly detection is the best to stop new attack since it can detect deviation of the normal behaviour (Raman, Ahmed et Mathur 2021)

Thus we focus on anomaly detection

Graph anomaly detection

According to Joshua Neil "Attacks do not happen in isolation on a single endpoint. Instead, they are exhibited across multiple endpoints, and in the communications between these endpoints." So graphs is a natural structure to detect attack.

- Up to our knowledge, no graph anomaly detection in OT
- In IT works in graphs have been already used, for instance:
  - To represent the calls of the functions of binary (Cohen, Yger et Rossi Nov 2021)
  - To model the stream of messages sent between IP adresses:
    - ⋆ To do classification (Xiao et al. 2020 ; Abou Rida, Parrend et Amhaz 2021)
    - ⋆ To do unsupervised learning with community detection, auto-encoder and scan statistics (Ding et al. 2012 ; Neil et al. 2013 ; Leichtnam et al. 2020)

$\Rightarrow$ Only one statistical work to test if there is an anomaly (Neil et al. 2013), otherwise probability models are not used

Our data: dynamical graphs of counting

- N IP adresses communicate over a time $[0, T]$ at different times $t \in [0, T]$ by sending messages.
- The attack can occur in a time interval Δ_t unknown, so the data has to be split with different Δ_t
- We divide the time into intervals of equal length Δ_t

$$[0, T] = \cup_{i=1}^{n} I_i.$$

- The aggregated data is $\boldsymbol{\mathcal{G}} = (\mathcal{G}_i)_{1 \leq i \leq n}$ where for all $1 \leq i \leq n$, $\mathcal{G}_i = (\mathcal{N}, \mathcal{E}_i)$ with the set of nodes $\mathcal{N} = \{1, \ldots, N\}$ and $\mathcal{E}_i$ the set of edges during I_i and there is an edge (k, l) if the IP address k sent at least a message to the IP address l during
- We can construct the adjacency matrices X^i such that $\forall 1 \leq k, l \leq N, X^i_{k,l}$ is the number of messages sent by the IP address k to the IP address l

Our solution: Testing abnormality of a graph of a sequence

1. **Learn a normal behaviour, a distribution $\mathbb{P}_0$, over a sequence of graphs $\mathcal{G} = (\mathcal{G}_i)_{1 \leq i \leq n}$** with a flexible family $\mathcal{F}$ of probability distributions such that $\mathbb{P}_0 \in \mathcal{F}$
   - Let's assume that the graphs $\mathcal{G}_i$ are i.i.d under the distribution $\mathbb{P}_0$
2. **Test if a graph has the normal behaviour**
   - for $i \geq n+1$

$$\begin{cases} H_0 : \mathcal{G}_i \sim \mathbb{P}_0, \\ H_1 : \mathcal{G}_i \sim \mathbb{P}_1. \end{cases}$$

3. **Compute the distribution of the log-likelihood L_0 of the distribution $\mathbb{P}_0$ with a bootstrap to get a quantile $q_{0.05}$.** For $i \geq n+1$, H_0 is said to be true if:

$$L_0(\mathcal{G}_i) \geq q_{0.05}.$$

Candidate families $\mathcal{F}$ for $\mathbb{P}_0$

Given a graph $\mathcal{G}^*$:

- **Stochastic bloc model** with an hyperparameter the number of classes K. A modification of the VEM is made so the SBM is learnt over $\boldsymbol{\mathcal{G}}$. The log-likelihood of $\mathcal{G}^*$ $L_0^K(\mathcal{G}^*)$ which is intractable is approximated with the log-likelihood of the complete data.
- **Gaussian kernel** with the hyperparameter window $h > 0$:

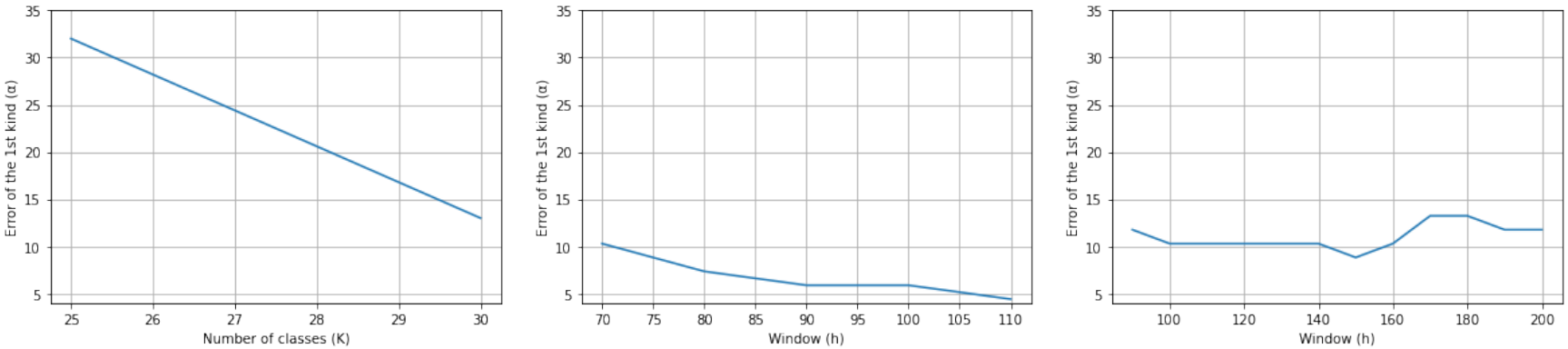
$$L_0^h(\mathcal{G}^*) = \sum_{k \neq l} \log \left(\frac{1}{nh} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{X_{kl}^* - X_{kl}^k}{h})^2} \right).$$

- **Poisson kernel** with the hyperparameter window $h > 0$:

$$L_0^h(\mathcal{G}^*) = \sum_{k \neq l} \log \left(\frac{1}{n} \sum_{i=1}^{n} \frac{(X_{kl}^i + h)^{X_{kl}^*}}{X_{kl}^*!} e^{-(X_{kl}^i + h)} \right).$$

Error of the first kind and tuning of the hyperparameters

The dataset from a firm in OT is split into 3 datasets: a learning dataset to learn $\mathbb{P}_0$, a validation dataset to tune the hyperparameter, a test dataset to estimate the error.



Error of the 1st kind of the validation set for the SBM (left panel), Gaussian kernel (middle), Poisson kernel (right)

Error of the 1st kind (summary)

Model	Hyperparameter	Error of the 1st kind of the test set	Time of learning
SBM	30	4%	46 hours
Gaussian kernel	90	5%	1/2 hours
Poisson kernel	150	10%	1/2 hours

Now we have to define candidate attacks ($\mathbb{P}_1$ distribution)

Hypothesis H_1: Let's test star and path

According to Neil et al. 2013, star and directed path in graphs are typical of cyberattacks.

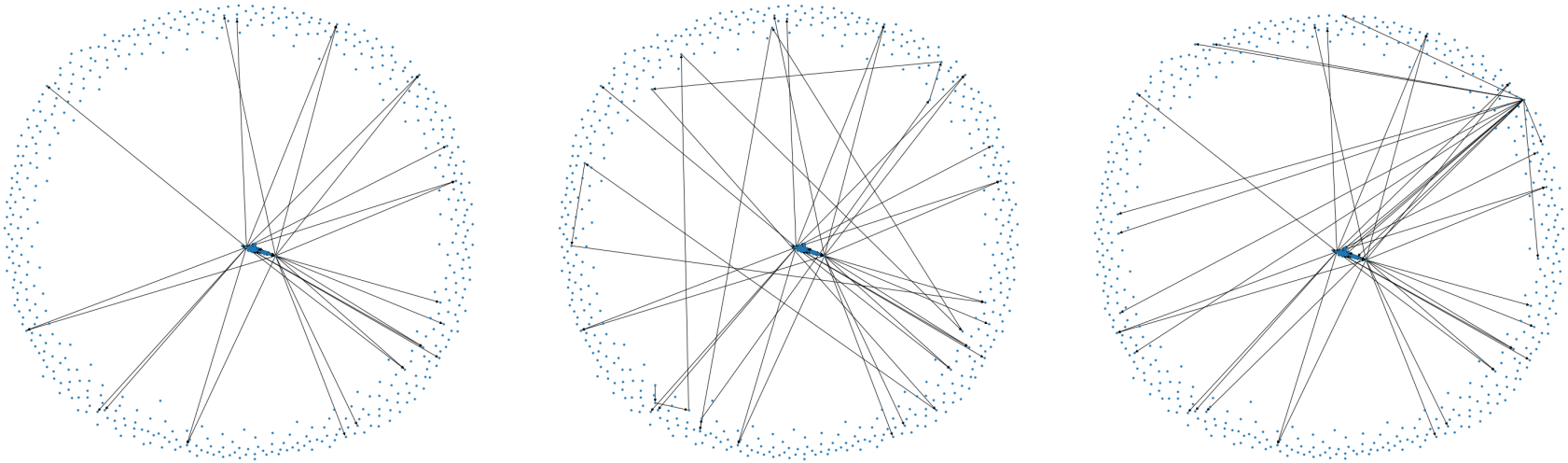
Let's construct a star of size $\theta_s \in [\![1, N-1]\!]$ with β edges on each couple of nodes of the star

- Take a normal graph from the test set
- Choose uniformly a node k which will be the center of the star
- Choose uniformly θ_s nodes in $\{1, \ldots, N\}\_\{k\}$
- Add a value β on the edges of the star

We construct similarly a directed path of length $\theta_p \in [\![1, N]\!]$ with β edges on each couple of nodes of the path

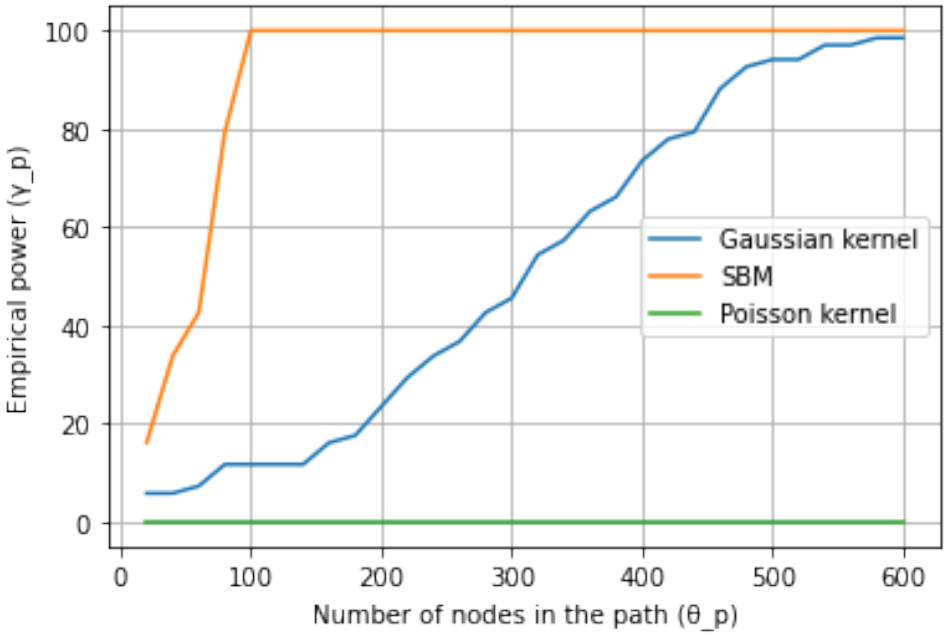
- Take a normal graph form the test dataset
- Choose uniformly θ_p nodes
- Add a value β on the edges of the path

Graphs with star and path

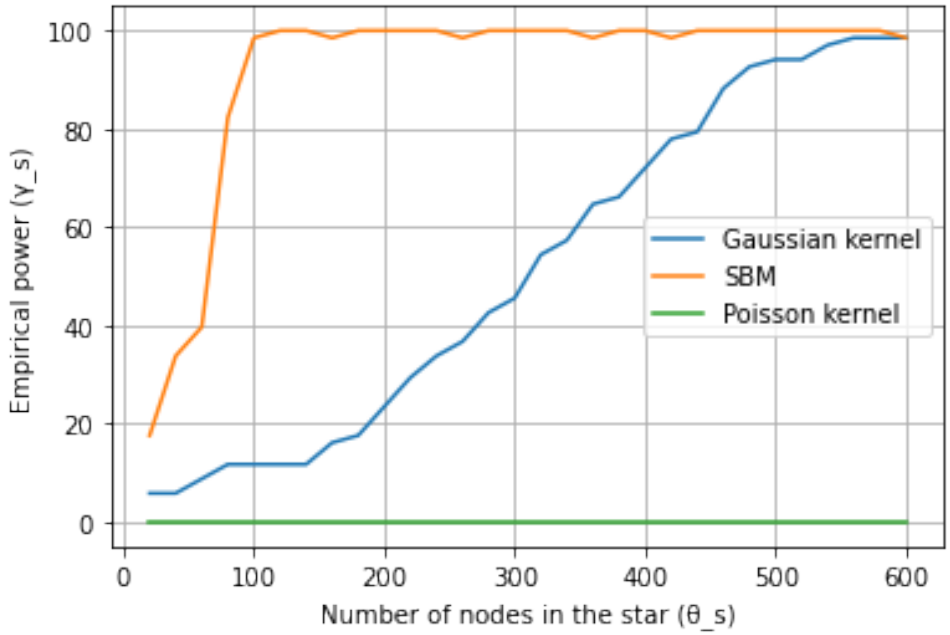


Normal graph (left), graph with a path (middle), graph with a star (right)

Empirical power



Empirical power for the path



Empirical power for the star

The SBM is way better than the Gaussian kernel and the Poisson kernel.

An other solution: Testing abnormality of a sequence of graphs

As a recall, we observe $\mathcal{G} = (\mathcal{N}, \mathcal{E}_i)_{1 \leq i \leq n}$

$$\begin{cases} H_0 : \forall i \in [\![1, n]\!], \text{no abnormality} \\ H_1 : \exists i \in [\![1, n]\!], \text{there is an abnormality} \end{cases}$$

The test can be reformulated thanks to the Scan statistics

In Priebe et al. 2005, we define the scan region for the scale $k \in \mathbb{N}$ the location the vertex $v \in \mathcal{N}$ and the closed kth-order neighborhood of v in $\mathcal{G}_i$ to be the induced graph of $N_k^i(v)$ denoted by:

$$\Omega(N_k^i(v))$$

with vertices $V(\Omega(N_k^i(v))) = N_k^i(v)$ and edges
$E(\Omega(N_k^i(v))) = \{(v, w) \in \mathcal{E}_i : v, w \in N_k^i(v)\}$
The time-dependent locality statistic is:

$$\phi_{k,i}(v) = |E(\Omega(N_k^i(v)))| \tag{1}$$

Scan Statistics

A vertex-standardized locality statistic with a window width τ is:

$$\tilde{\phi}_{k,i}(v) = \frac{(\phi_{k,i}(v) - \hat{\mu}_{k,i,t,\tau}(v))}{\max(\hat{\sigma}_{k,i,t,\tau}(v), 1)}$$

where $\hat{\mu}_{k,i,t,\tau}(v)) = \frac{1}{\tau}\sum_{t'=i-\tau}^{i-1} \phi_{k,t'}(v)$ and
$\hat{\sigma}^2_{k,i,t,\tau}(v) = \frac{1}{\tau-1}\sum_{t'=i-\tau}^{i-1}(\phi_{k,t'}(v) - \hat{\mu}_{k,i,t,\tau}(v))^2$
The standardized scan statistics is:

$$\tilde{M}_{k,i} = \max_{v} \tilde{\phi}_{k,i}(v)$$

we consider a temporally-normalized version of $\tilde{M}_{k,i}$

$$S_{k,i} = \frac{\tilde{M}_{k,i} - \tilde{\mu}_{k,i,L}}{max(\tilde{\sigma}_{k,i,L}, 1)}$$

where $\tilde{\mu}_{k,i,L}$ and $\tilde{\sigma}_{k,i,L}$ are the running mean and the standard deviation of $\tilde{M}_{k,i}$ based on the most recent L time step.

Reformulating the test

$$\begin{cases} H_0 : \forall i, S_{1,i} \leq q \\ H_1 : \exists i \text{ such that } S_{1,i} > q \end{cases}$$

Error of the 1st kind

Power

To test H_1, I construct as follow a star of size θ_s

- Simulate a sequence of graphs with the Stochastic Bloc Model
- Choose uniformly a graph i^* in the sequence
- Choose uniformly a node which will be the center, and then θ_s nodes in the other nodes
- Add an edge to the graph i^* at each couple of node of the star

Thank you for your attention

Abou Rida, Amani, Pierre Parrend et Rabih Amhaz (2021). "Anomaly Detection for CyberSecurity Using Inductive Node Embedding with Convolutional Graph Neural Networks". In : Complex Networks and their Applications 2021, 30 novembre - 2 décembre Madrid, Spain. url : https://hal.archives-ouvertes.fr/hal-03393640.

Cohen, Roxane, Florian Yger et Fabrice Rossi (Nov 2021). "Adding semantic to level-up graph-based Android malware detection". In : Complex Networks and their Applications 2021, 30 novembre - 2 décembre Madrid, Spain. url : https://hal.archives-ouvertes.fr/hal-03393640.

Ding, Qi et al. (2012). "Intrusion as (Anti)Social Communication : Characterization and Detection". In : Proceedings of the 18th ACM SIGKDD International Conference on Know KDD '12. Beijing, China : Association for Computing Machinery, p. 886-894. isbn : 9781450314626. doi : 10.1145/2339530.2339670. url : https://doi.org/10.1145/2339530.2339670.

- Leichtnam, Laetitia et al. (juin 2020). "Sec2graph : Network Attack Detection Based on Novelty Detection on Graph Structured Data". In : DIMVA 2020 : 17th Conference on Detection of Intrusions and Malware, a T. 12223. Lecture Notes in Computer Science. Lisbon, Portugal, p. 238-258. doi : 10.1007/978-3-030-52683-2_12. url : https://hal.inria.fr/hal-02950489.
- Neil, Joshua et al. (2013). "Scan Statistics for the Online Detection of Locally Anomalous Subgraphs". In : Technometrics 55.4, p. 403-414. doi : 10.1080/00401706.2013.822830. eprint : https://doi.org/10.1080/00401706.2013.822830. url : https://doi.org/10.1080/00401706.2013.822830.
- Priebe, Carey et al. (oct. 2005). "Scan Statistics on Enron Graphs". In : Computational Mathematical Organization Theory 11, p. 229-247. doi : 10.1007/s10588-005-5378-z.
- Raman, Gauthama, Chuadhry Mujeeb Ahmed et Aditya Mathur (2021). "Machine learning for intrusion detection in industrial control systems : challenges and lessons from experimental evaluation". In :

IEEE Transactions on Industrial Informatics. doi : 10.1186/s42400-021-00095-5.


Umer, Muhammad Azmi et al. (2022). "Machine learning for intrusion detection in industrial control systems : Applications, challenges, and recommendations". In : International Journal of Critical Infrastructure Protection, p. 100516. issn : 1874-5482. doi : https://doi.org/10.1016/j.ijcip.2022.100516. url : https://www.sciencedirect.com/science/article/pii/S1874548222000087.


Xiao, Qingsai et al. (2020). "Towards Network Anomaly Detection Using Graph Embedding". In : Computational Science – ICCS 2020 12140, p. 156-169.