LSE Summer week 2022 Day 1

Présenté par: Pierre Parrend, Professeur HDR
Laboratoire Systèmes de l'EPITA (Kremlin-Bicêtre)/ICube (Strasbourg)
Enseignant à l'EPITA Strasbourg

Thanks to organizers!

- Marc, Fabrice, and all others !

Trusted AI for secure critical systems

Présenté par: Pierre Parrend, Professeur HDR
Laboratoire Systèmes de l'EPITA (Kremlin-Bicêtre)/ICube (Strasbourg)
Enseignant à l'EPITA Strasbourg

LSE – some piece of news

LSE Summer week 2022

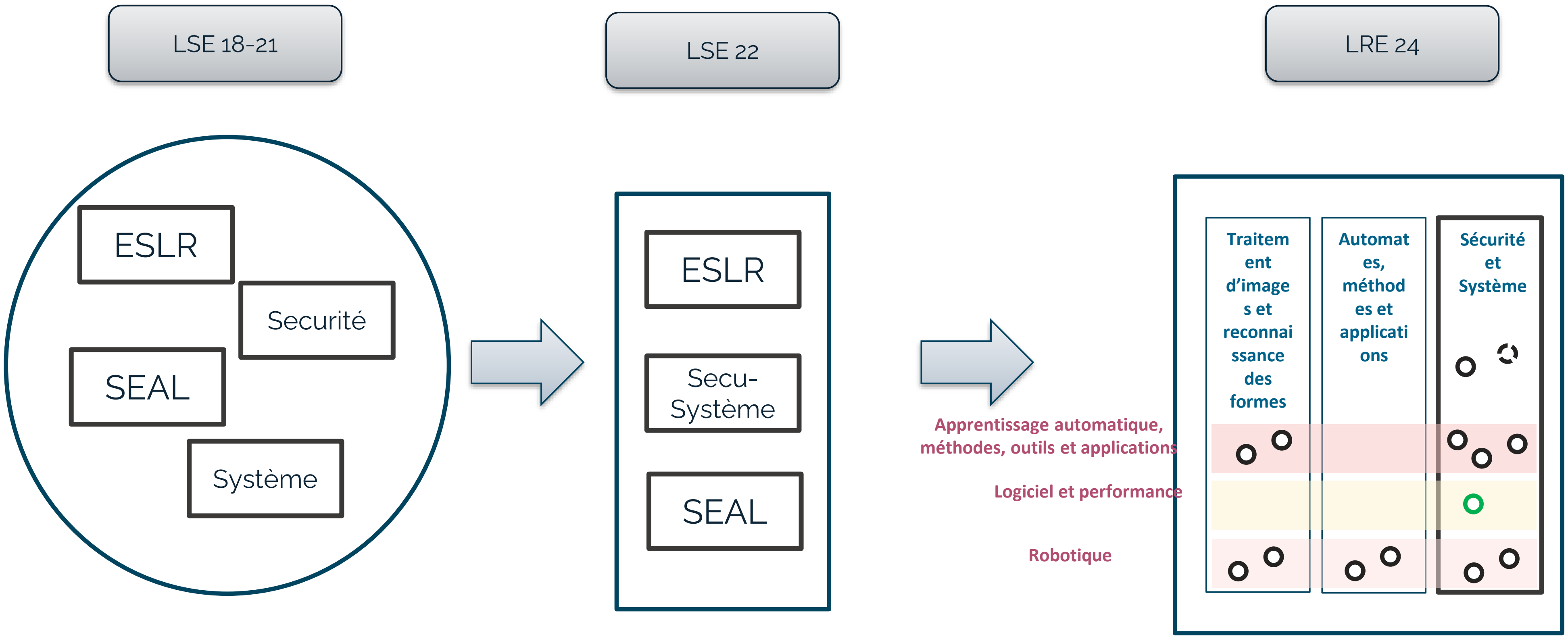
When?	Who	What?
10h	Pierre Parrend	Trusted AI for secure critical systems
10h45	Grégory Blanc	Generating synthetic traffic to improve the robustness of network intrusion detection
11h15	Julius Pfrommer	Industrial Communication with OPC UA – Secure by Design?
11h45	Mark Angustures	Port scans and DDos detection by time series filtering
12h30 - pause		
14h	Chistian Elloh	Anonymisation of DNS requests through blockchain
14h45	Badis Hammi	Is it really easy to detect sybil attacks in C-ITS ?
15h30	Mohammed Badredine Zouhair	Malwares
16h15	Laurent Beaudoin, Loica Avanthey	Cartography of submarine zones with lightweight means

Thursday, 7/7/22

Who	What?
Marc Espie	To cache or not to cache, making pkg_add faster
Martin Grenouilloux	Discovering new ways of attacking AES when trying to do something else
Alex Levigoureux, Antoine Jouan	Work on UEFI driver rootkit with a bare metal hypervisor
Younes Benreguieg	Metrics for graph-based Anomaly Detection
Darius Engler	Writing a bare metal GPU driver for the Raspberry PI 4

Saturday, 9/7/22
TO be finalized

A brief history of LSE



Security-Systems Team

Team members

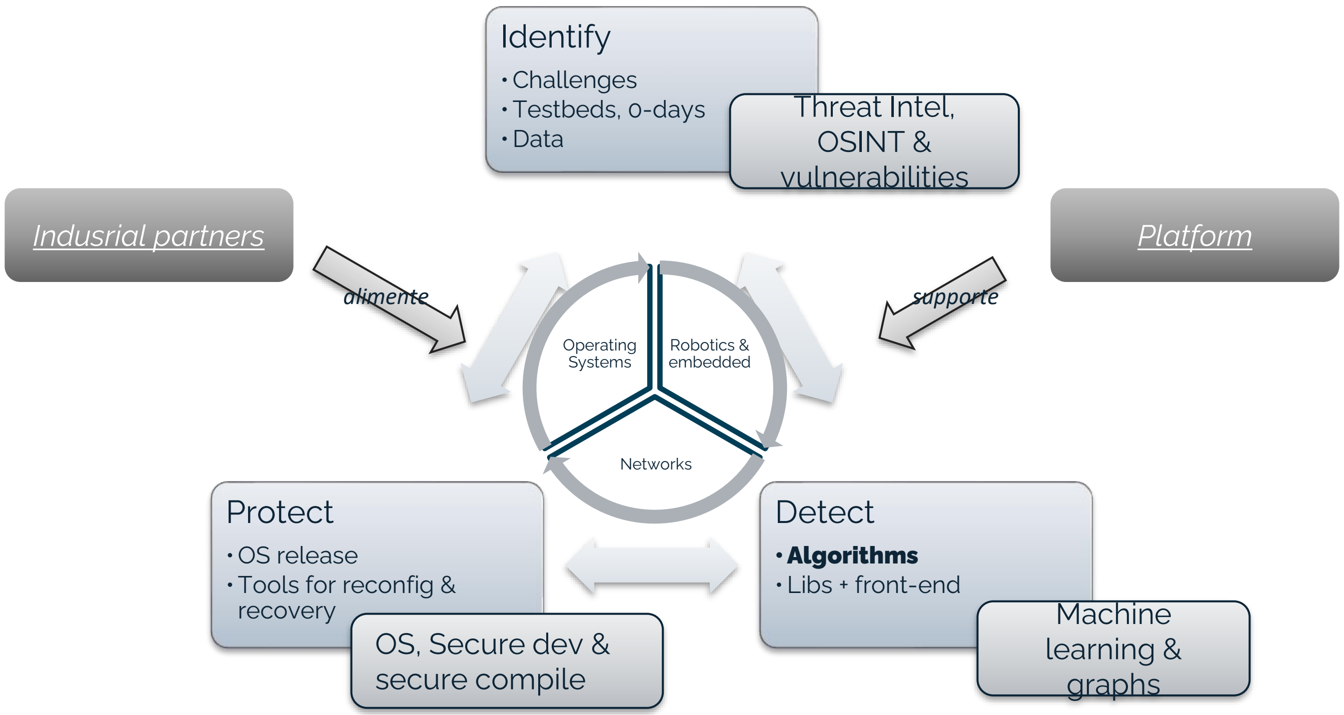




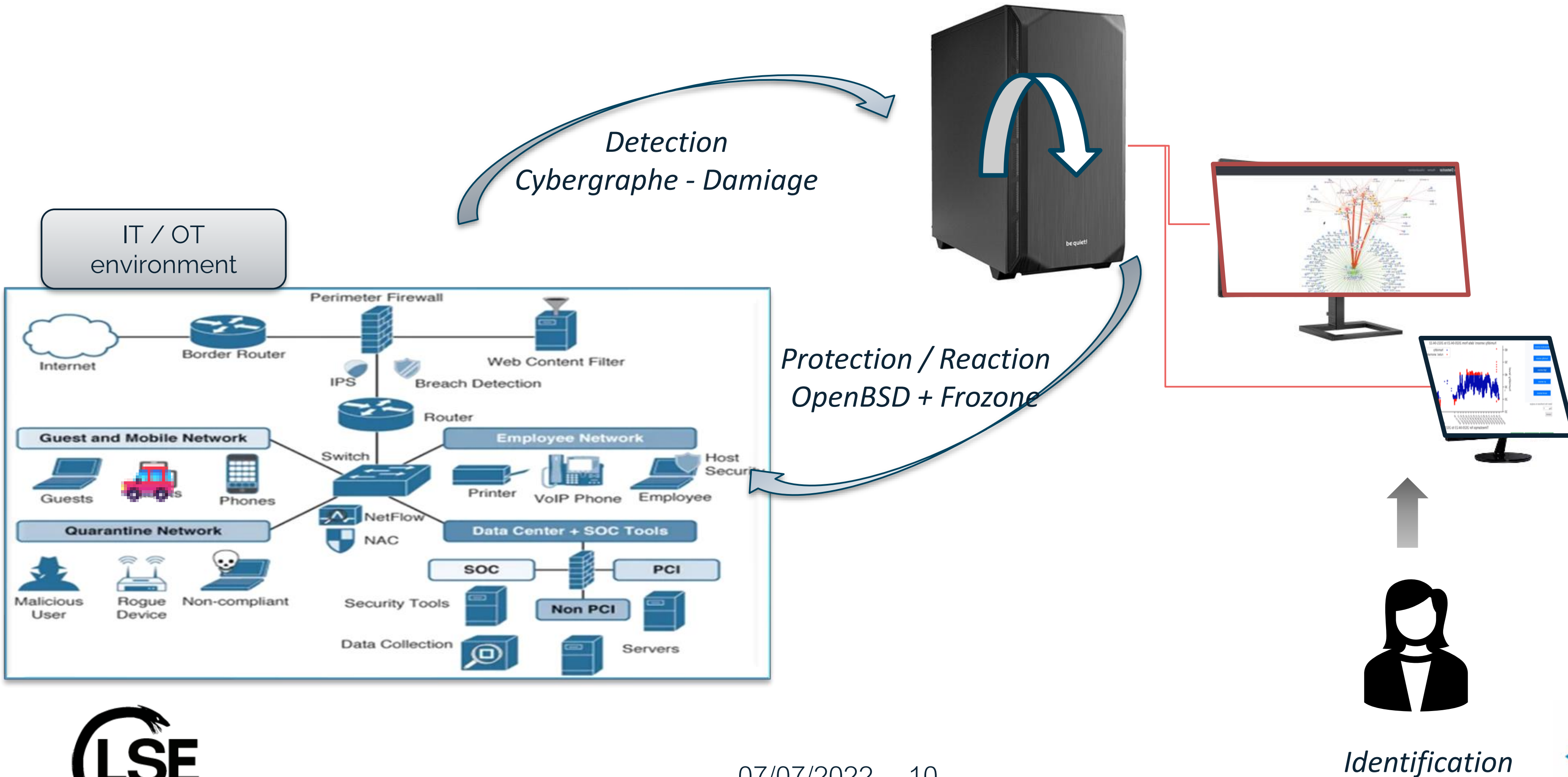


Doctorants: Amani Abou Rida, Julien Michel (, Majed Jaber + YOU)
Younes Benreguieg, Antoine Jouan, Nabih Benazzouz, Sébastien Delsart, Alexis Ehret, Thomas Berlioz, Daniel Frederic, Leo Benito, Mathieu Fourre, Alex Levigoureux, Alexandre Fresnais, Martin Grenouilloux, Pierre-Emmanuel Patry, Cesar Belley, Esteban Blanc, Arthur Cohen, Tanguy Dubroca, Martin Schmidt

Scientific goals – 2022



The SOC cybersecurity use case



Software

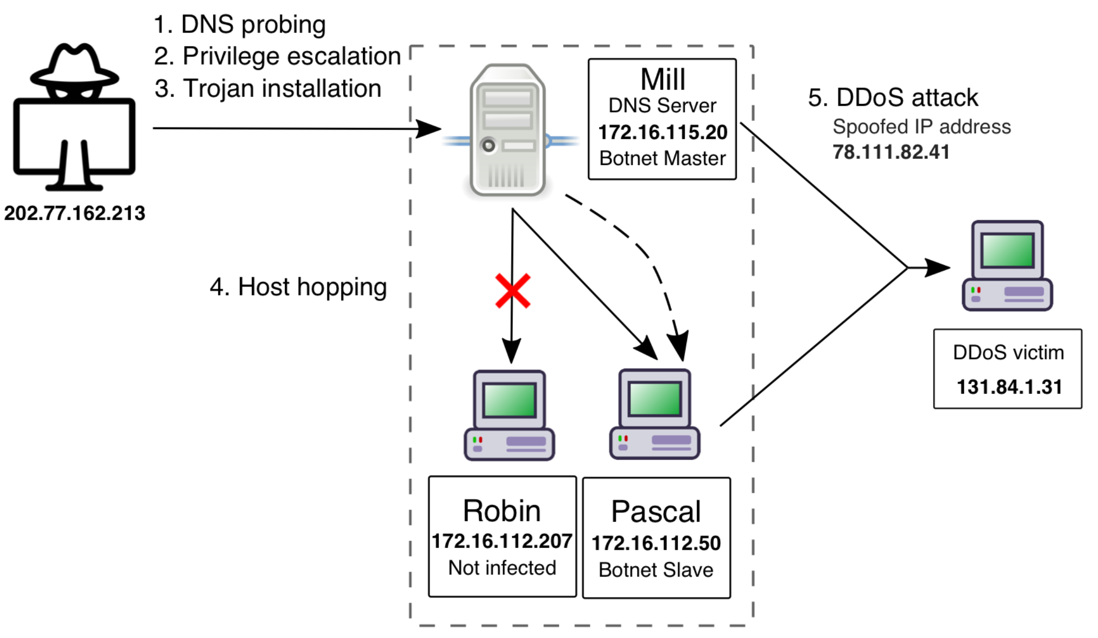
	2018	2019	2020	2021	2022	Total
Research				2	2	2 CREA, Cybergraphe
Open Source	2	2	2	2	1	2 Glibc; OpenBSD
➢ Commit	915	488	209	179	134	OpenBSD (Marc is 1 of 3 maiin contributors)
Google Summer of Code	2		2			4 libvirt, Vulkan, gcc-rs, Radare2
Student projects	9	many	many	many	3	
POCs				1	2	3
EPITA Infrastructure	2	2	2	2		2 Moulinette; infra ACUs
CTF	1	1	1	1	1	1
Total	**7**	**5**	**7**	**8**	**7**	**14**

Detection

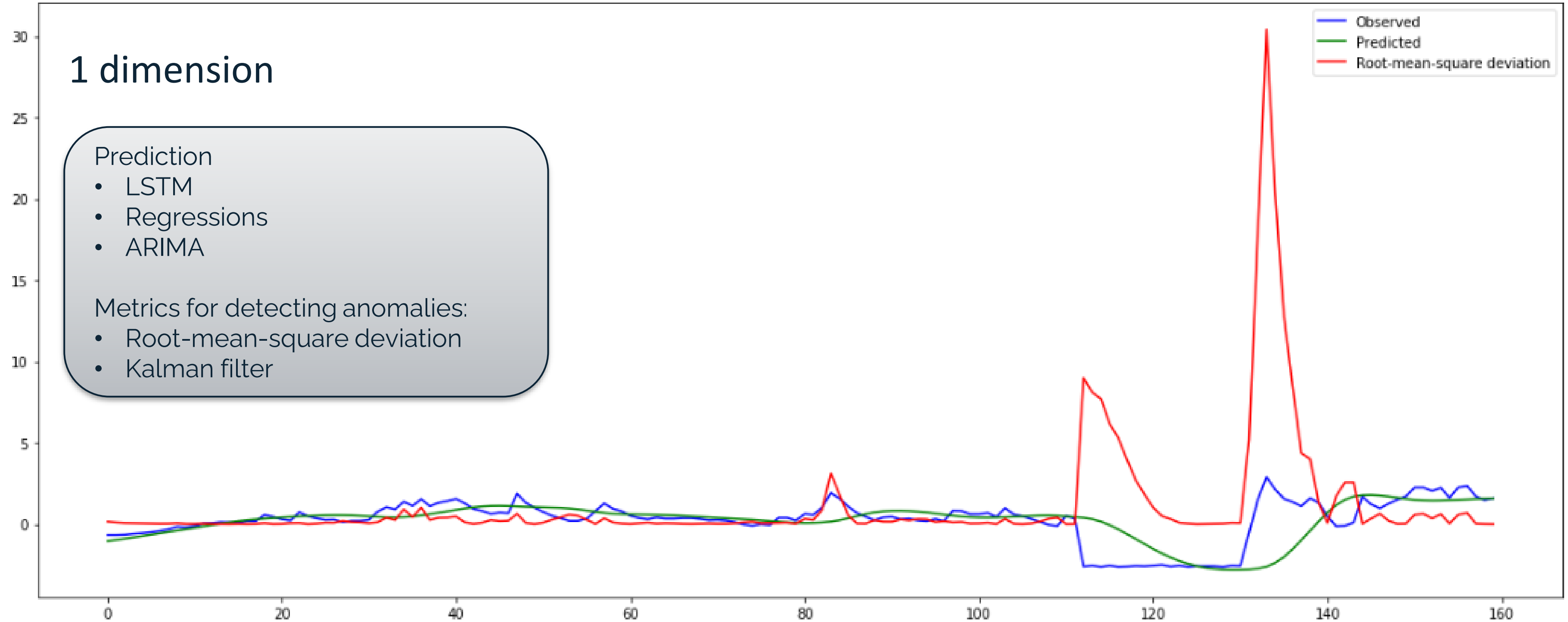
The adversary

- Example Multi-step Attack
  - Dataset DARPA 2000

LLDoS 2.0.2



1-dimension

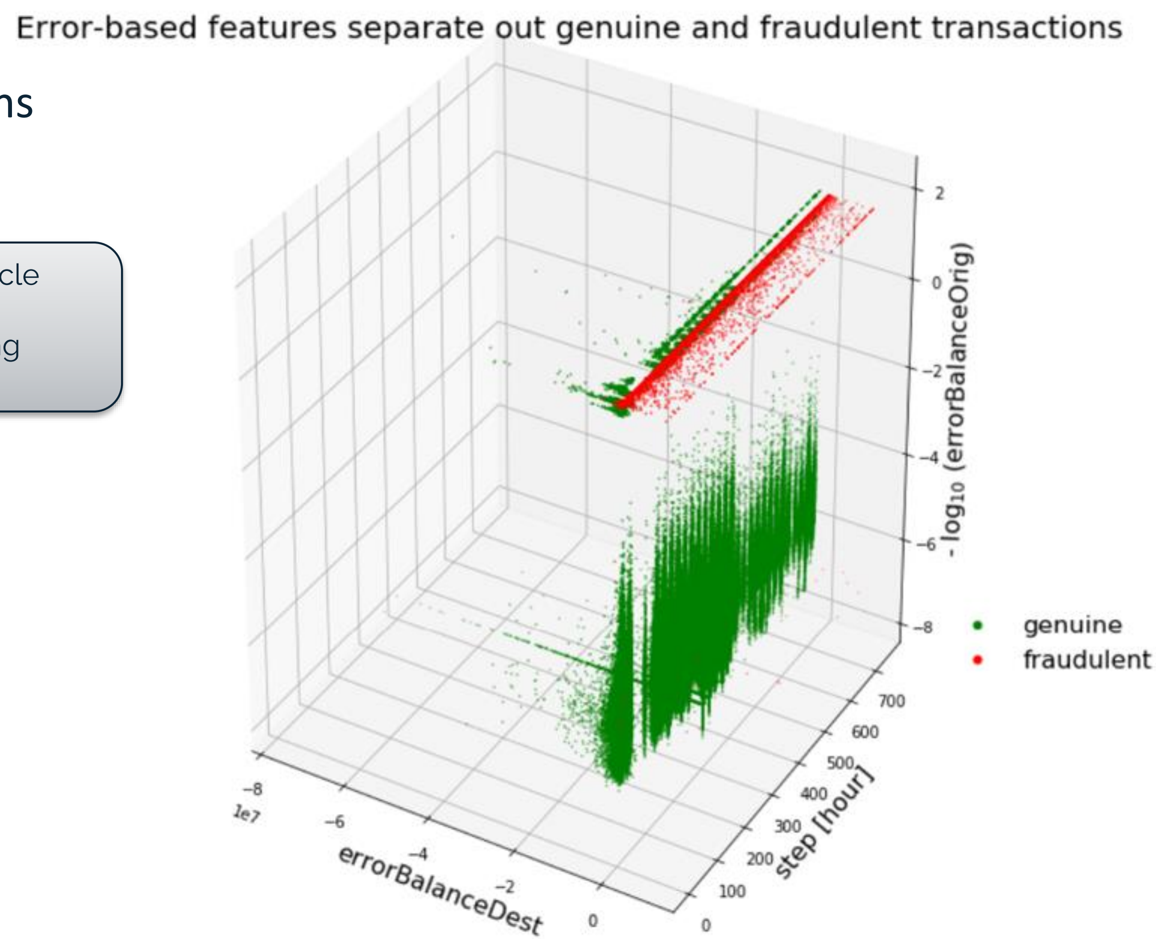



```
Training duration:16.657758951187134
```

N-dimensions

N dimensions

Classification as prediction oracle
- XGBoost
- MLP – Multi-layer processing
- …



Limitations

Low dimensions

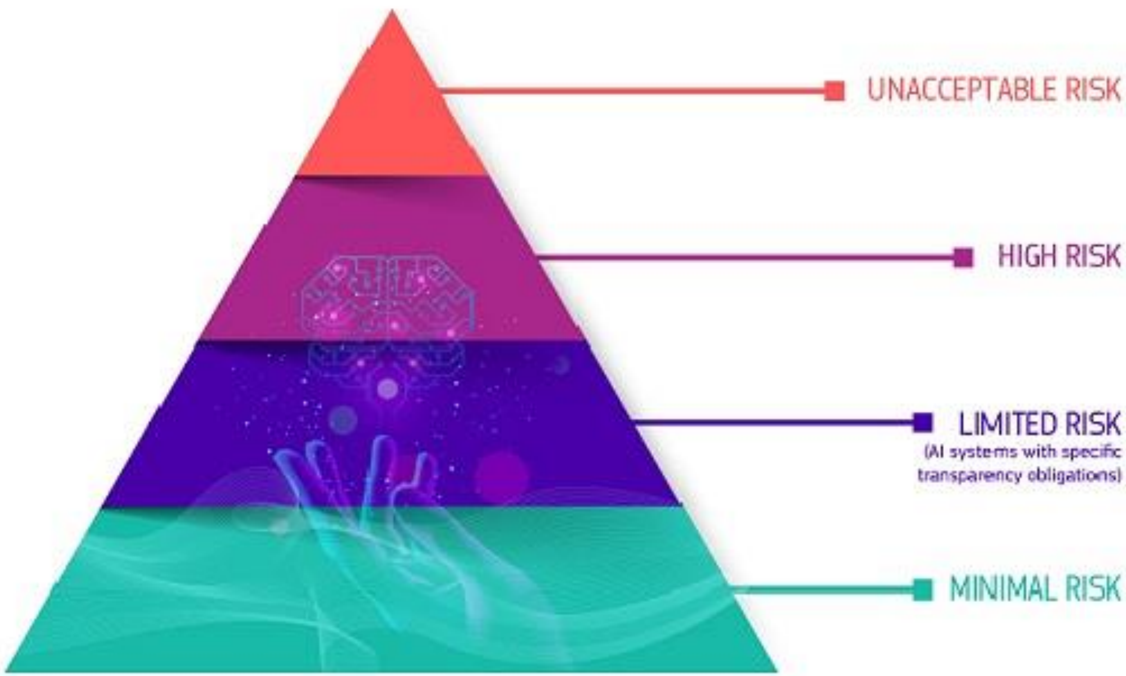
- Efficient for massive events only: DoS, DDoS, scan
- Often imprecise
- Very specific use cases

Bigger dimensions

- Limited to multi-variable correlation
- Requires often manual identification of derived features
- Correlation is not causality

AI and critical systems

What is Trusted IA ?



Social scoring by governments

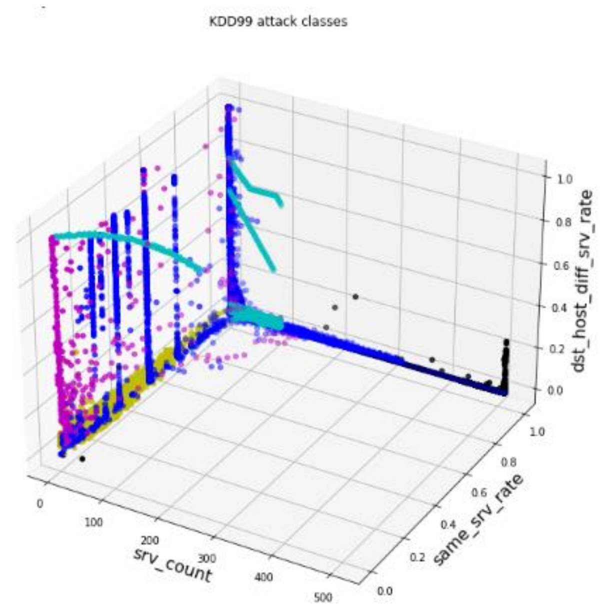
Industrial applications, medicine, education

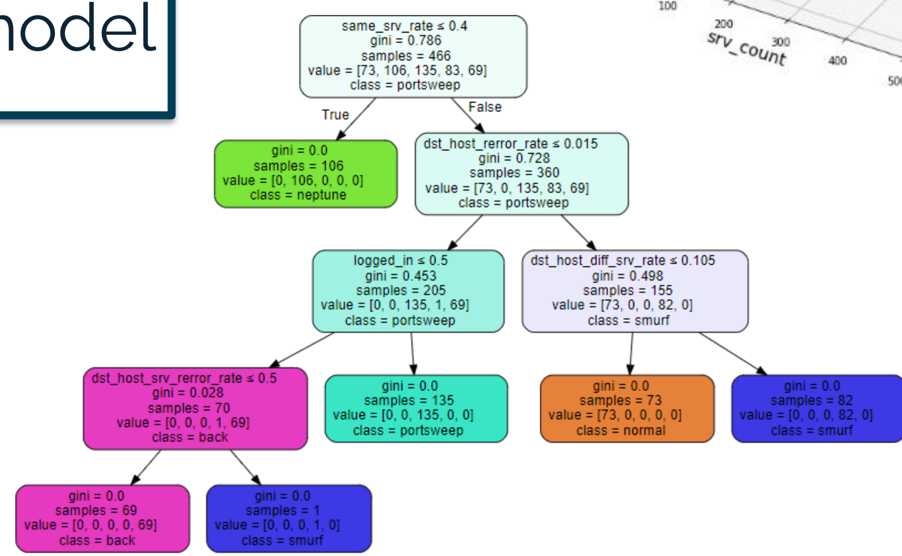
Chatbots

Video Games

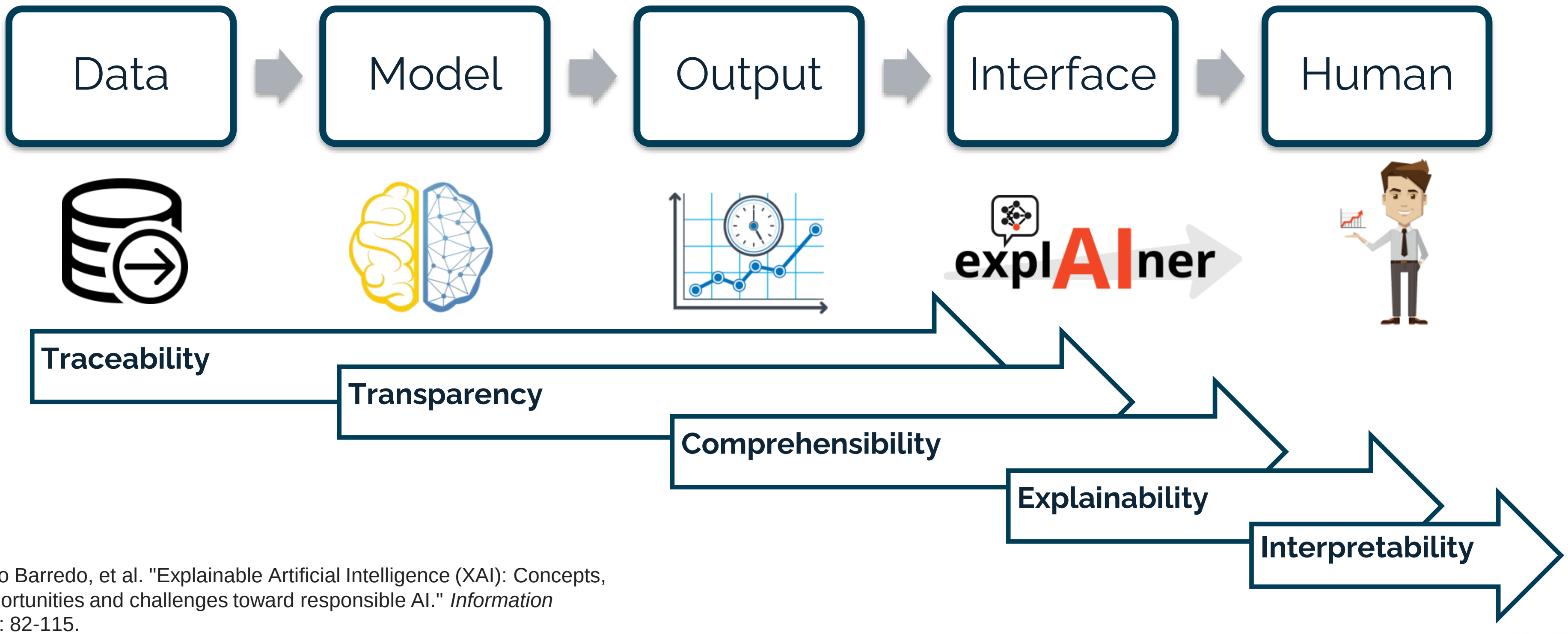
Human oversight

- Visualisation of data
- Visualisation of model



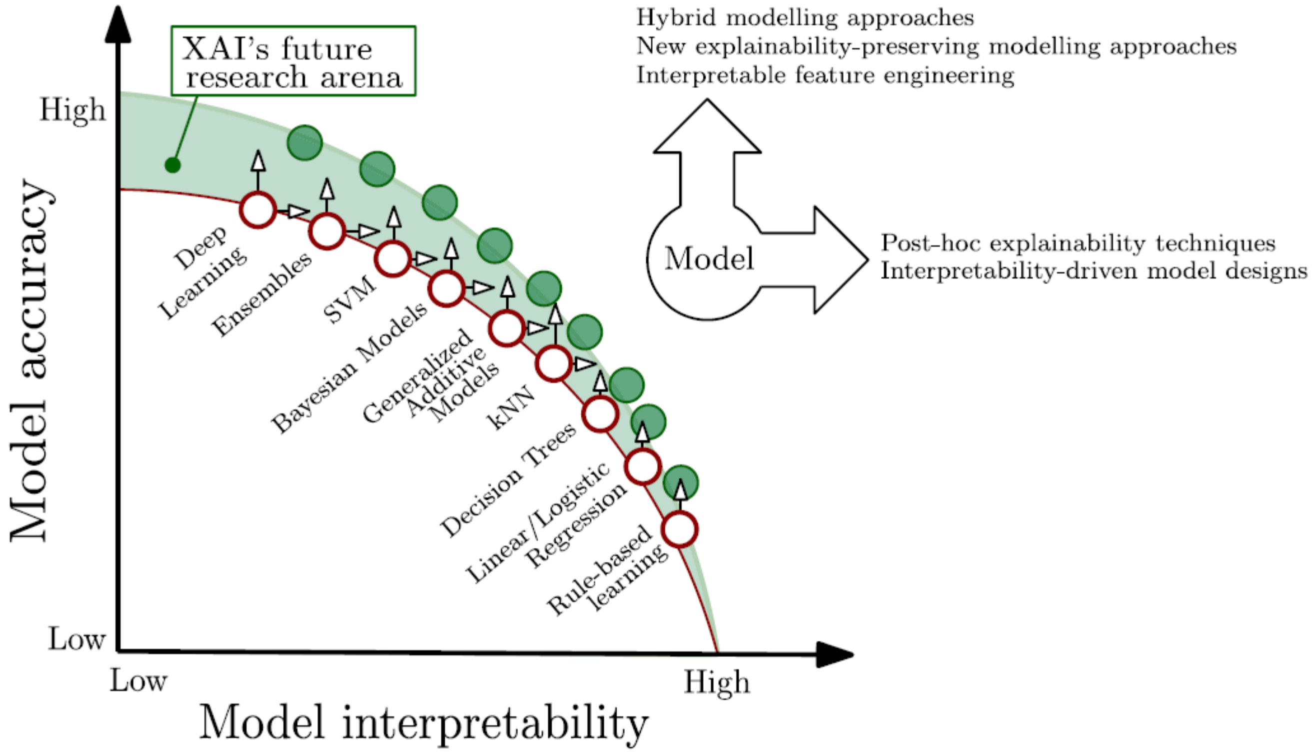


Towards explainability



Arrieta, Alejandro Barredo, et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." *Information fusion* 58 (2020): 82-115.

The challenge: getting accuracy and interpretability back together



[Arrieta2020]

Trusted graphs

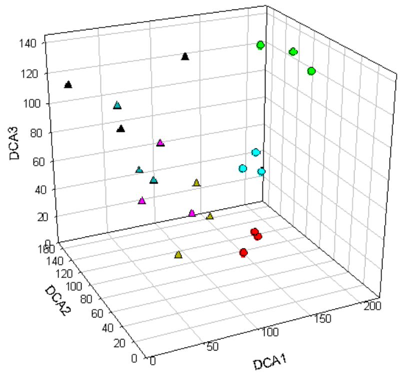
From ML to Graph learning

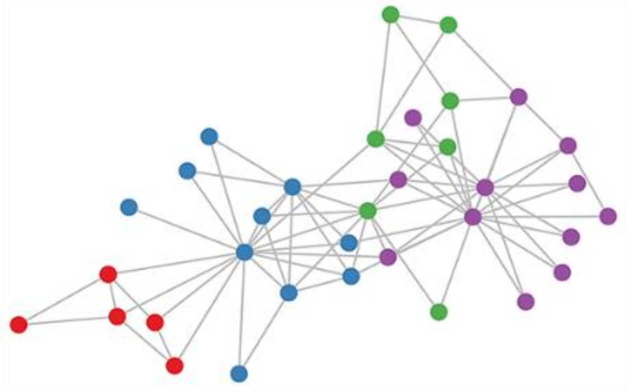
Euclidean domains
1,.., n dimensional

Machine learning-based network anomaly detection methods such as one-class support vector machines (OSVM), autoencoders (AE) , and isolation forests (IS).

Non-Euclidean domains

Graph learning such as graph analysis, graph embedding, graph neural network

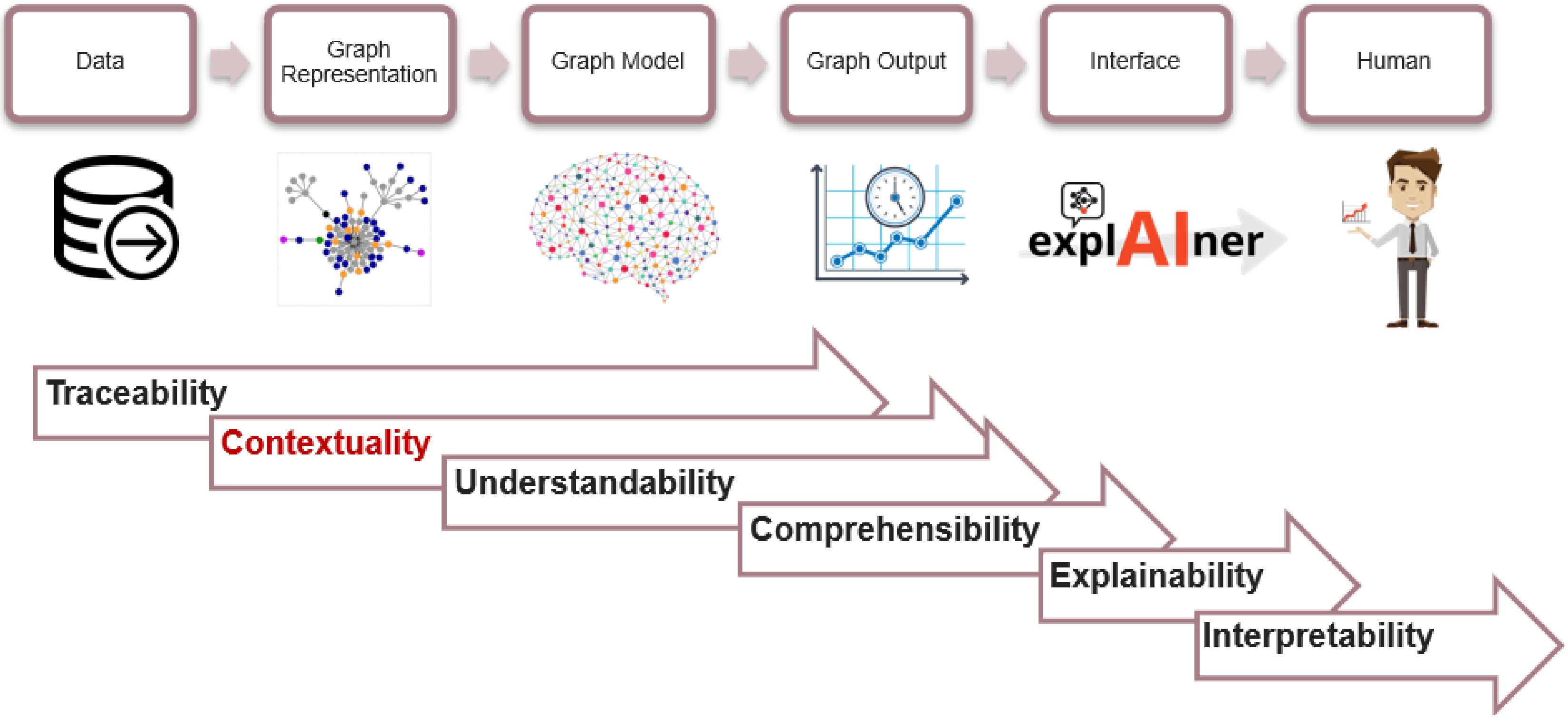




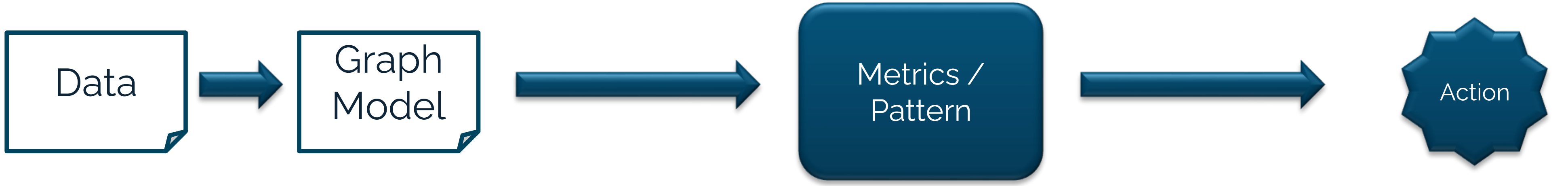
Expectations

- **Traceability:** graph nodes can support data embedding. Interactions are visible!
- **Transparency:** most graph algos are intrinsicaly transparent (not all)
- **Comprehensibility:** output is typically a graph part (not always)
- **Explainability:** making interactive graph plots is straightforward
- **Interpretability**: see all repvious points

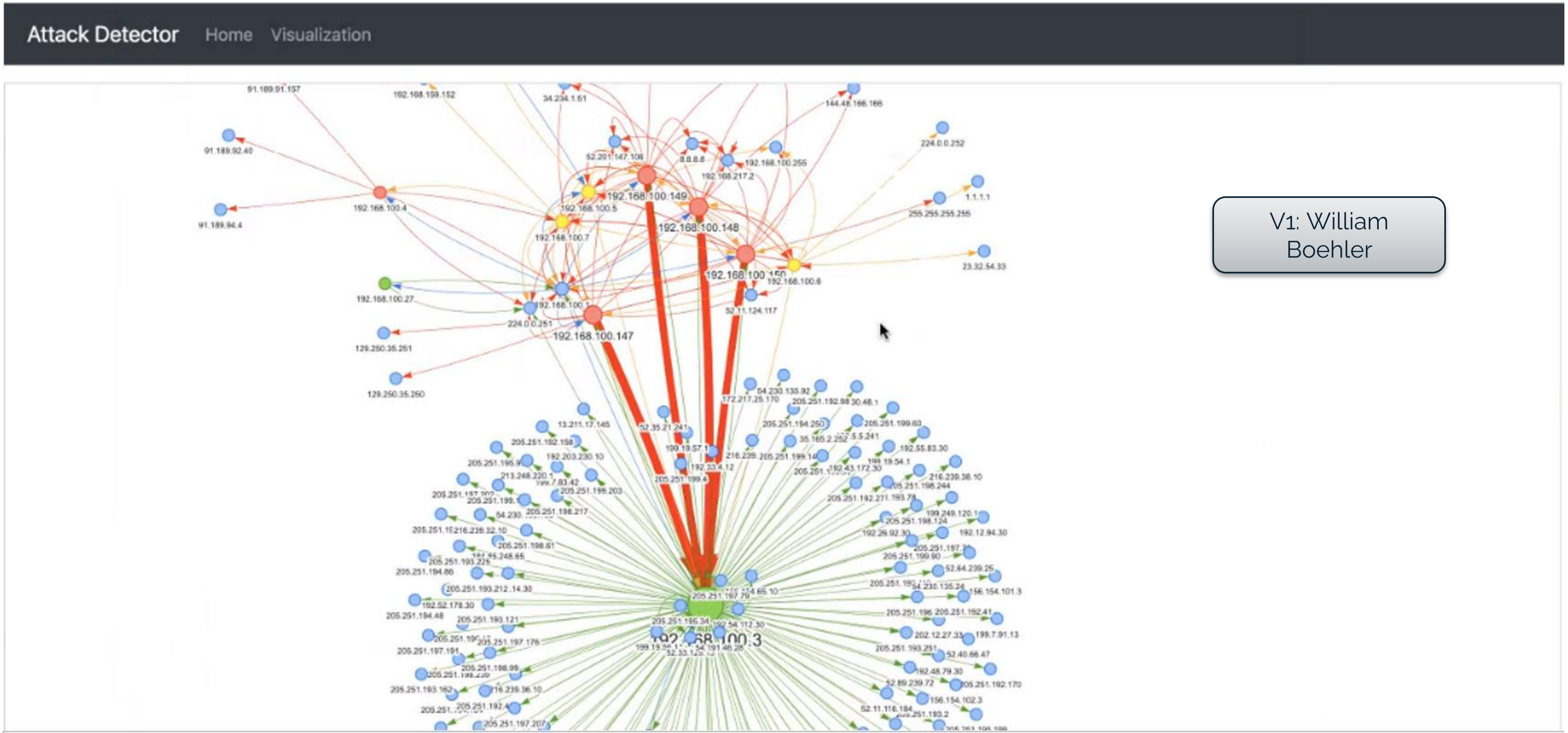
Explainability in graph models



Machine learning with graphs 1/3
First order search

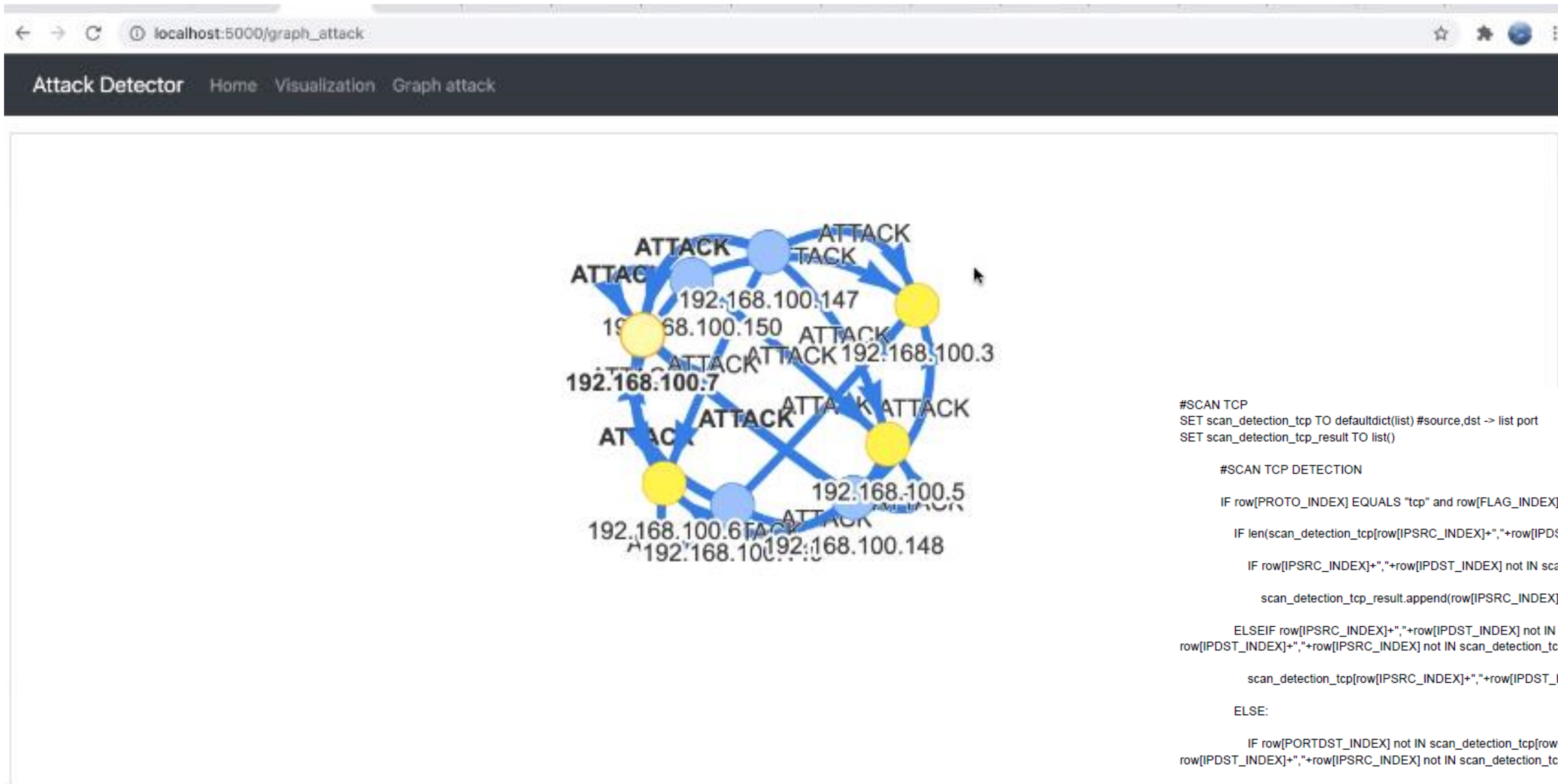


The Cybergraph tool: low hanging fruits



V1: William Boehler

The Cybergraph tool




```
#SCAN TCP
SET scan_detection_tcp TO defaultdict(list) #source,dst -> list port
SET scan_detection_tcp_result TO list()

        #SCAN TCP DETECTION

        IF row[PROTO_INDEX] EQUALS "tcp" and row[FLAG_INDEX] EQUALS "RST":

          IF len(scan_detection_tcp[row[IPSRC_INDEX]+","+row[IPDST_INDEX]]) >= SCAN_TCP_TRESHOLD:

            IF row[IPSRC_INDEX]+","+row[IPDST_INDEX] not IN scan_detection_tcp_result:

              scan_detection_tcp_result.append(row[IPSRC_INDEX]+","+row[IPDST_INDEX])

          ELSEIF row[IPSRC_INDEX]+","+row[IPDST_INDEX] not IN scan_detection_tcp and
row[IPDST_INDEX]+","+row[IPSRC_INDEX] not IN scan_detection_tcp:

            scan_detection_tcp[row[IPSRC_INDEX]+","+row[IPDST_INDEX]].append(row[PORTDST_INDEX])

          ELSE:

            IF row[PORTDST_INDEX] not IN scan_detection_tcp[row[IPSRC_INDEX]+","+row[IPDST_INDEX]] and
row[IPDST_INDEX]+","+row[IPSRC_INDEX] not IN scan_detection_tcp:

              scan_detection_tcp[row[IPSRC_INDEX]+","+row[IPDST_INDEX]].append(row[PORTDST_INDEX])
```

BotIOT

Attacks table

127.0.0.1:5000/attacks_table

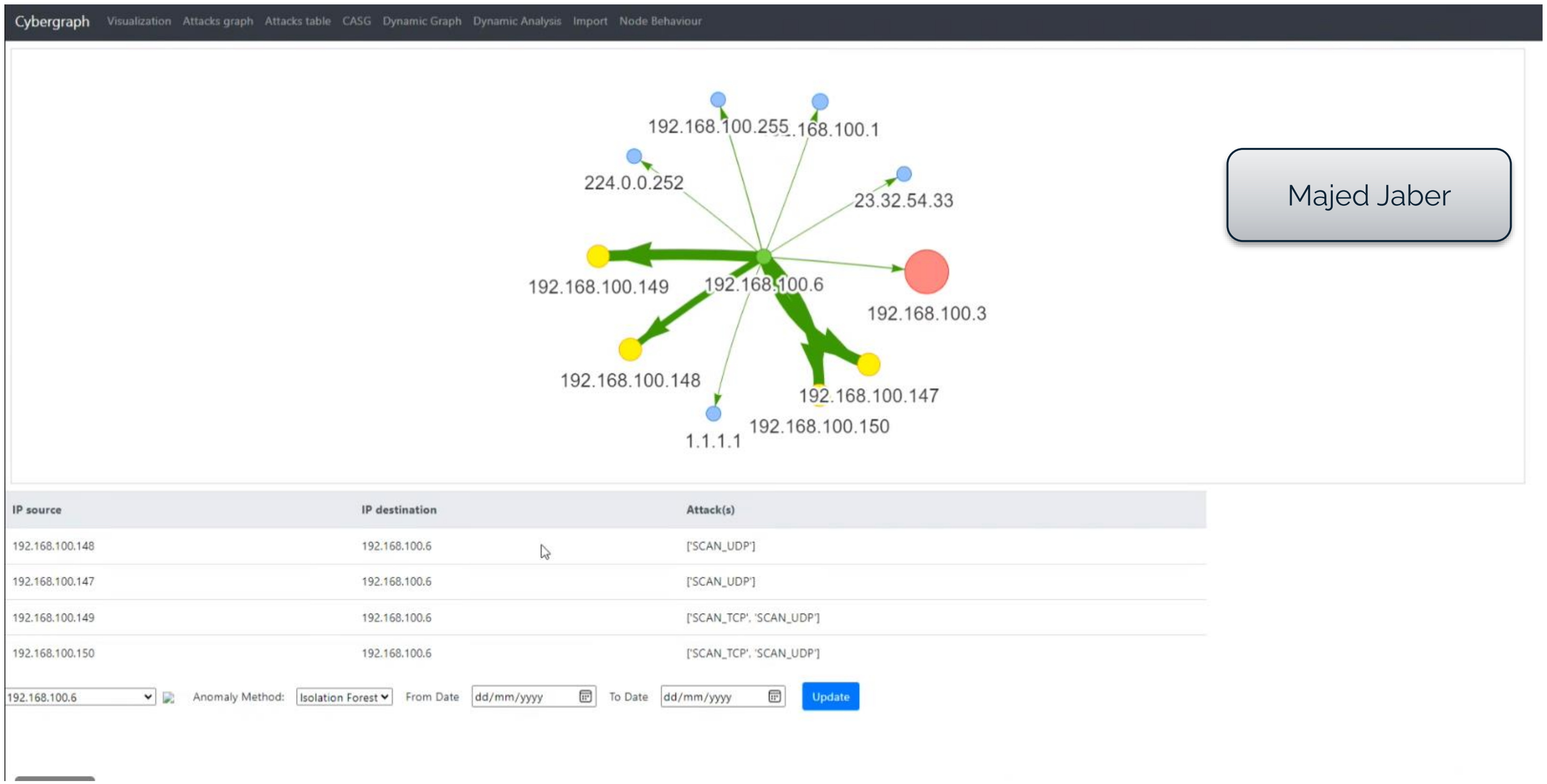
Cybergraph Visualization Attacks graph Attacks table CASG Import Dynamic Graph Dynamic Analysis

Select database : botiot ☑ Scan TCP ☑ Scan UDP ☑ DoS ☑ DDoS ☑ MITM ☑ Island Hopping Update

Current database in use : botiot

IP source	IP destination	Attack(s)
192.168.100.150	192.168.100.5	['SCAN_TCP']
192.168.100.150	192.168.100.7	['SCAN_UDP', 'SCAN_TCP']
192.168.100.150	192.168.100.6	['SCAN_UDP', 'SCAN_TCP']
192.168.100.150	192.168.100.3	['SCAN_UDP', 'SCAN_TCP']
192.168.100.149	192.168.100.6	['SCAN_UDP', 'SCAN_TCP']
192.168.100.149	192.168.100.3	['SCAN_UDP', 'SCAN_TCP']
192.168.100.149	192.168.100.7	['SCAN_UDP', 'SCAN_TCP']
192.168.100.149	192.168.100.5	['SCAN_TCP']
192.168.100.148	192.168.100.7	['SCAN_UDP', 'SCAN_TCP']
192.168.100.148	192.168.100.5	['SCAN_TCP']
192.168.100.148	192.168.100.3	['SCAN_UDP', 'SCAN_TCP']
192.168.100.147	192.168.100.5	['SCAN_TCP']
192.168.100.147	192.168.100.7	['SCAN_UDP', 'SCAN_TCP']
192.168.100.147	192.168.100.3	['SCAN_UDP', 'SCAN_TCP']
192.168.100.6	192.168.100.147	['SCAN_TCP']
192.168.100.6	192.168.100.148	['SCAN_TCP']

Node behaviour analytics



Majed Jaber

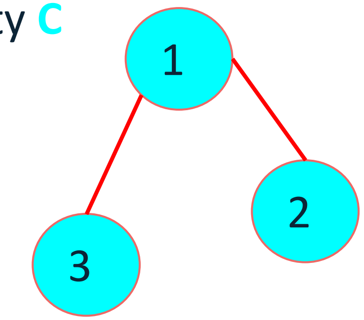
Machine learning with graphs 2/3
Graph features, Euclidian ML

Some graph metrics: communities

- **Density d**

0 =< d =< 1
Rate of number of connections between nodes in a community wrt number of possible connections number of connections

Example:
Community C



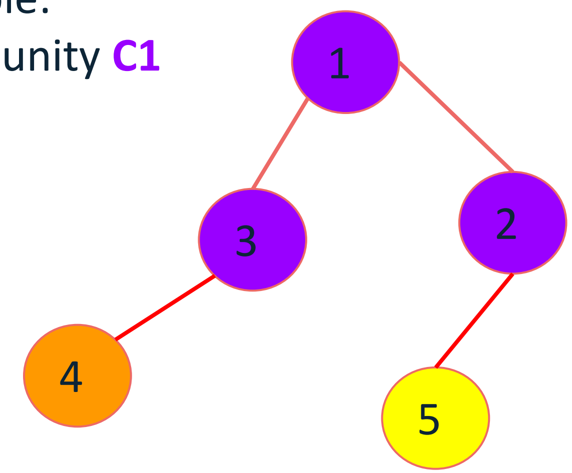
d =2/3

- 2 connections in C (1,2) and (1,3)
- 3 possible connections (1,2), (1,3) et (2,3)

- **Externality e**

0 =< e =< 1
Rate of edges with 1 end node not in the **C1** community and 1 in **C1** wrt. the total number of edges having at least 1 end node in **C1**

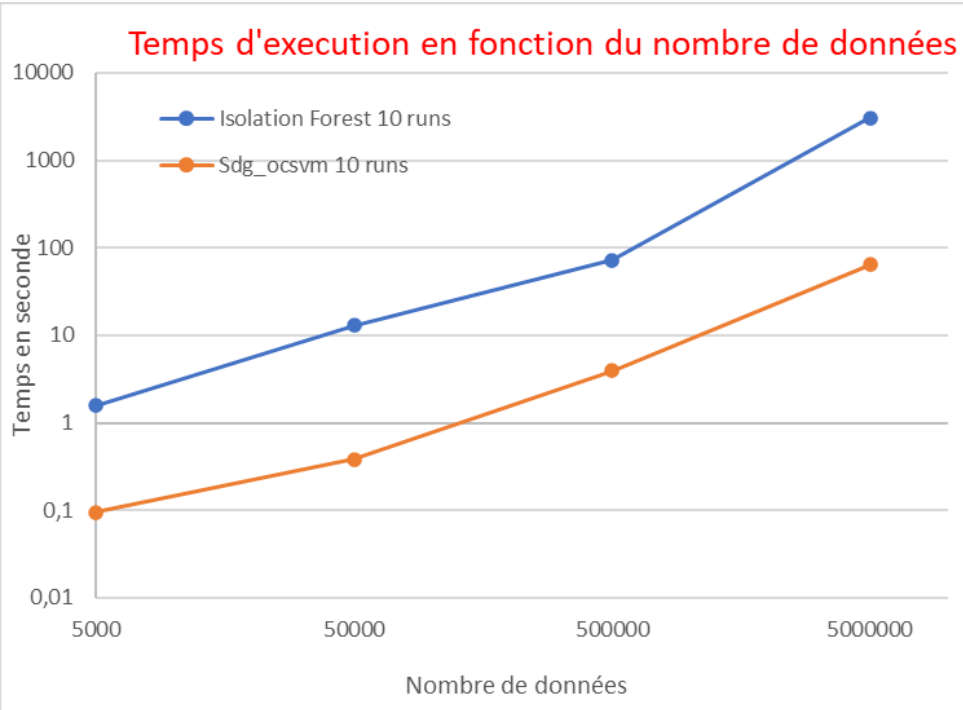
Example:
Community **C1**

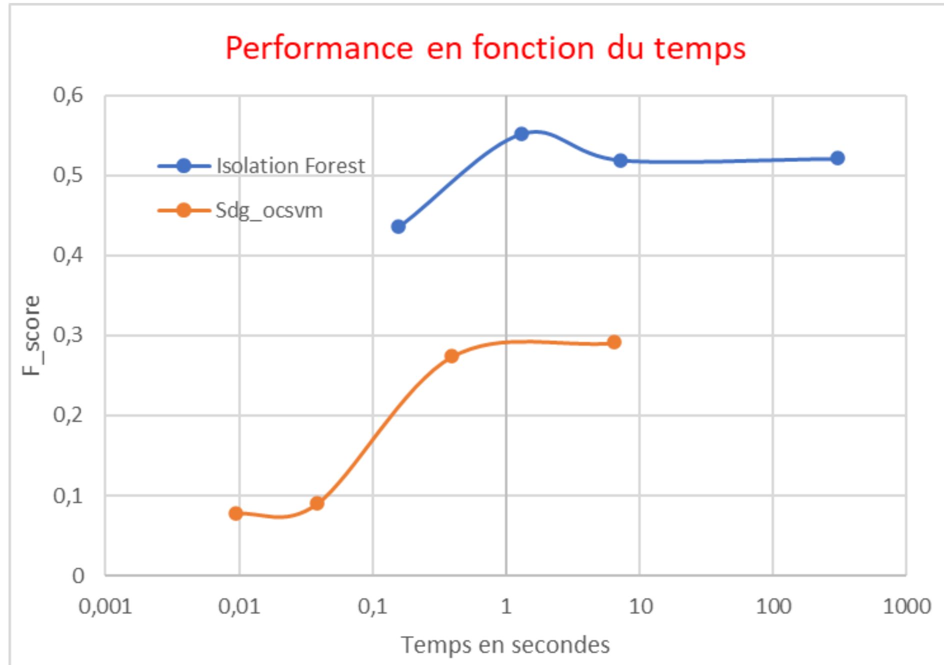


e = 0,5

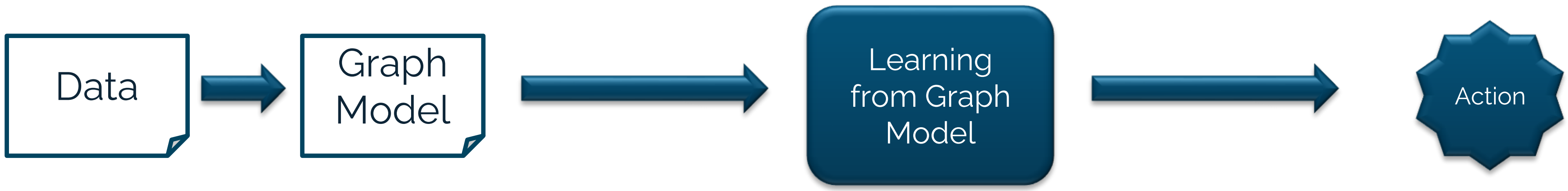
- 2 edges with end node outside **C1** (3,4) et (2,5)
- 4 edges with at least 1 node in **C1** (1,2), (1,3) et (2,3)

Benchmarking learning incl extracted graph features





Machine learning with graphs 3/3
Graph learning



Metrics for evaluating explainability

- Fidelity

difference of accuracy between the original predictions and the new predictions after masking out important input features

$$Fidelity = \frac{1}{N}\sum_{i=1}^{N}(f(G_i)_{yi} - f(G_i^{(1-m_i)})_{yi})$$

$$Infidelity = \frac{1}{N}\sum_{i=1}^{N}(f(G_i)_{yi} - f(G_i^{(m_i)})_{yi})$$

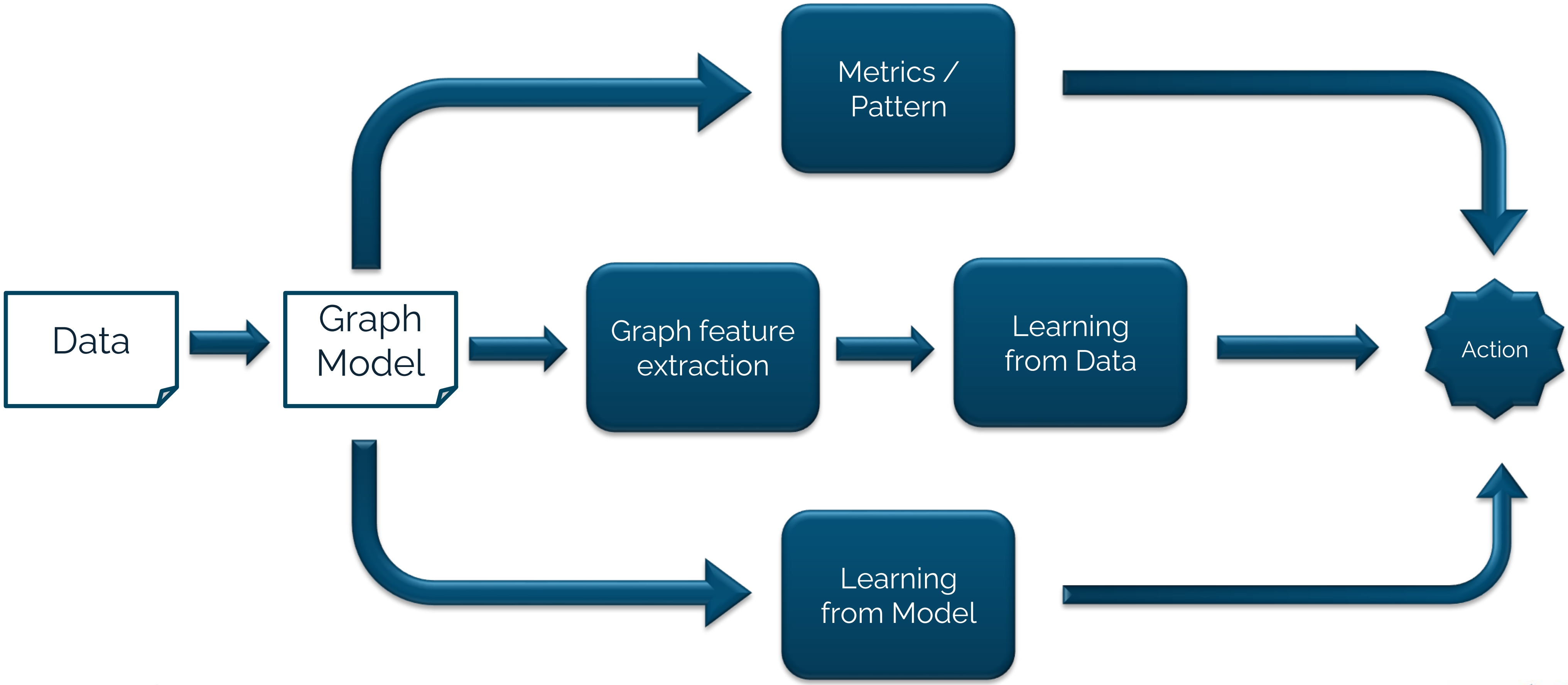
- Sparsity

the fraction of features selected as important by explanation methods

$$Sparsity = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{|m_i|}{|M_i|})$$

Pope, Phillip E., et al. "Explainability methods for graph convolutional neural networks." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

Machine learning with graphs
Summary



Future challenges

Scalability

- In small (IoT)
- In big (network core)
- Frugal algorithms becomes a must

Dynamic behaviour

- Reduce the batch size
- … and avoid poisoning when learning is adapted!

Next application domains

- XDGMed – Medical IoT
- Correau – Industrial systems - water distribution

This is a call for PhD Students …

And future interns !

Merci !!