Using *Graph Neural Networks* for scientific traffic categorisation

Daniela Brauner

Research Engagement Manager

daniela.brauner@geant.org

TNC

13 June 2025, Brighton

Public

Who we are?

Maarten Meijer
Graduate Intern
UTwente (The Netherlands)
GÉANT Intern

Daniela Brauner
Research Engagement Manager
GÉANT

Doina Bucur
Network data scientist,
Assistant Professor in Computer Science
at UTwente (The Netherlands)

Guy Roberts
Senior Transport Network Architect
GÉANT

Science flows

LHC experiments
data transfers

Satellite data
“ground” distribution

Astronomical
data transfers

… and many others.

Why to identify science flows?

- (Inter)national R&E networks carry a large volume of science traffic
- It is useful to understand the nature of the traffic
  - The science **experiment** and network **activity** involved
  - To better engage better with these researchers and support them
  - Efficient network use, traffic steering, future provisioning and capacity planning
  - Performance measurement (flows, data rates, …)
  - Site traffic profiling, or traffic accounting for shared inter-continental links
  - Quantify global behaviour and analyse trade-offs at scale
- And not only for RENs, scientists are also interested on having more info.

- HOW can we do it?
  - Normal network flows monitoring????
    - Yes, in overlay networks VRFs, e.g., LHCONE, but not full information about the experiments…
    - No, only for specific communities; and the others?

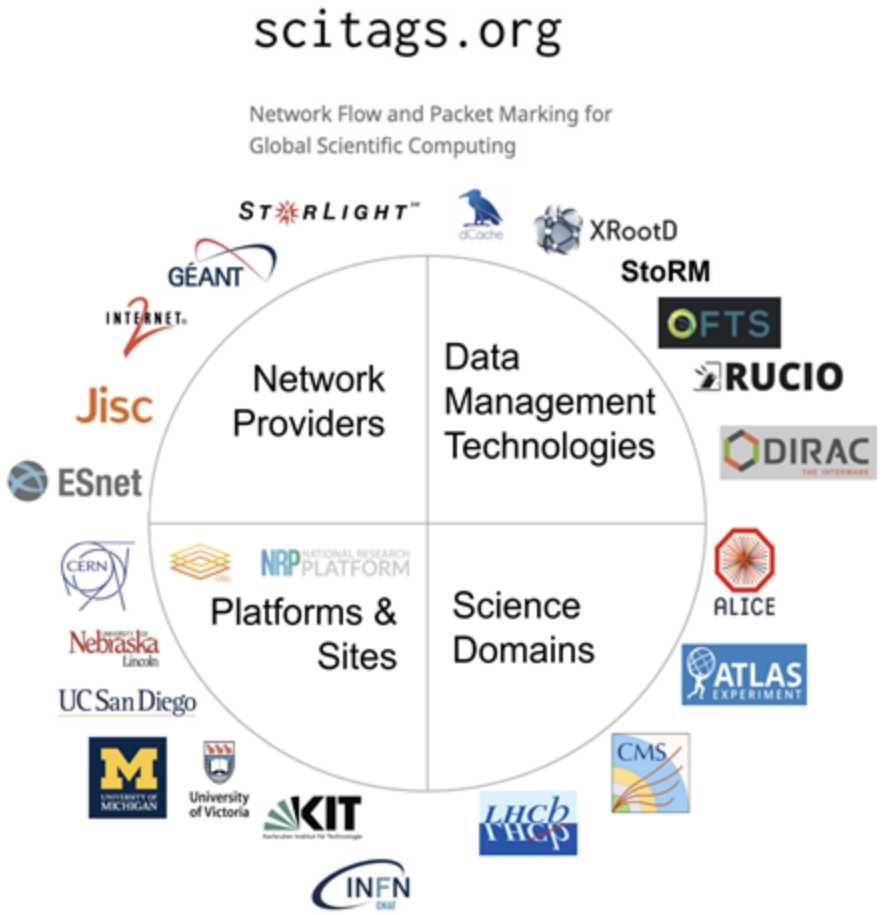
How scientists are doing now?

Marking data packets and flows with experiment and applications IDs for better accounting.

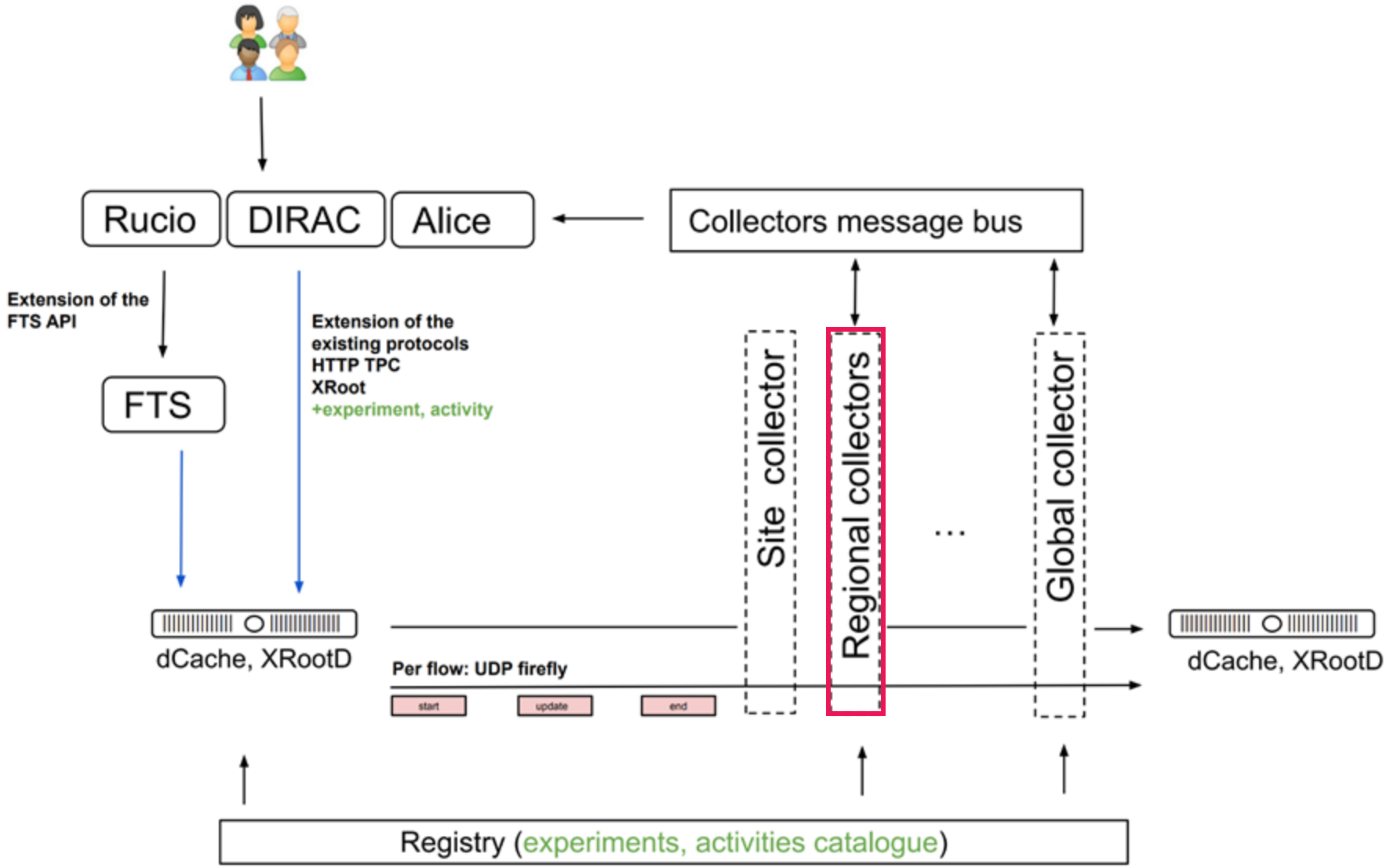
Defining a standard(s) for exchange of information between scientific communities, sites and network operators.

Two options are being pursued:

- Packet marking: encoding experiment/activity directly in packets (tag in the IPv6 flowlabel field)
- Flow marking - sending a separate UDP packet (**firefly**) with metadata (Identifying the experiment and activity of traffic)



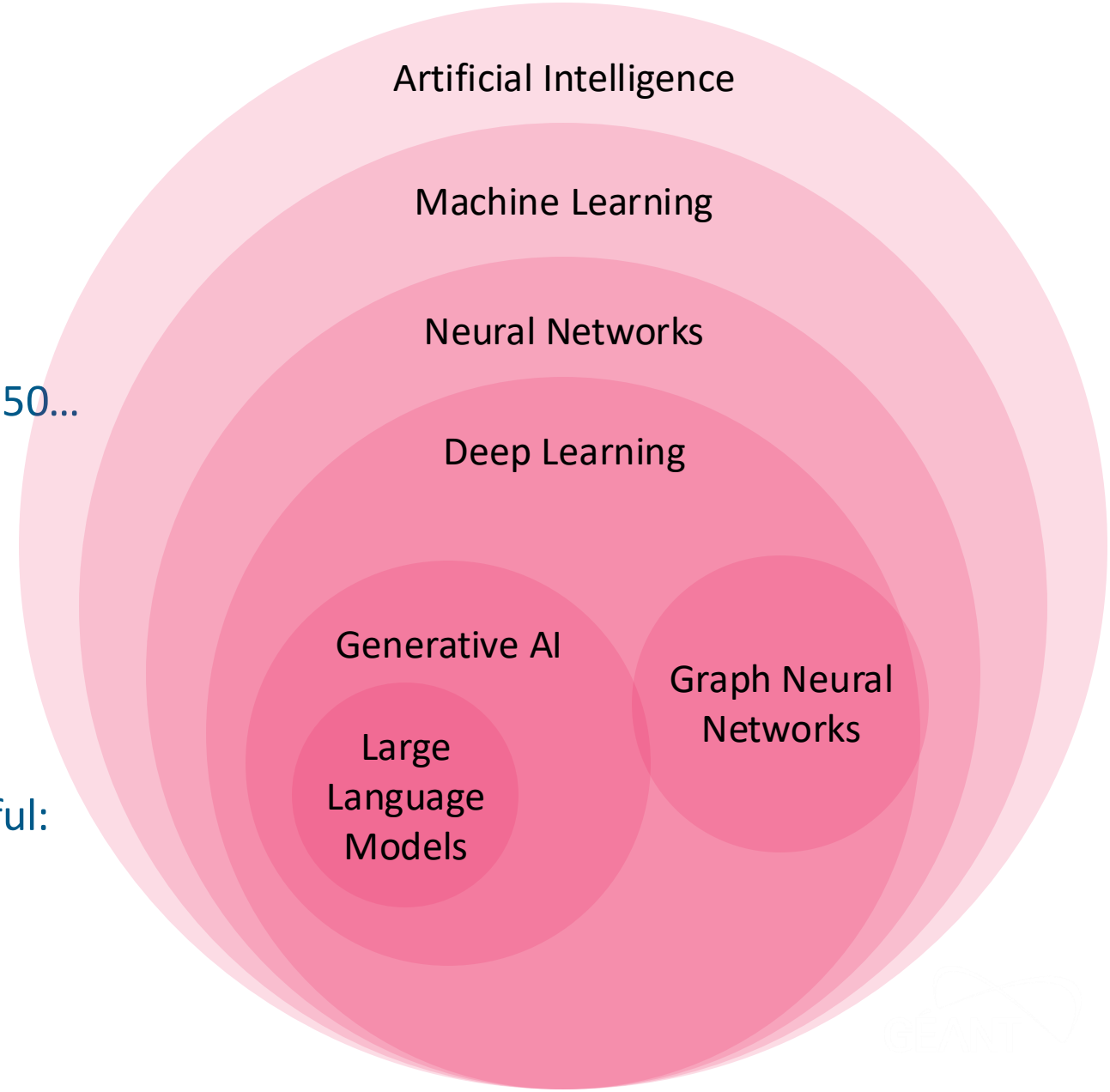
How scitags work - fireflies today



But not all domains yet!

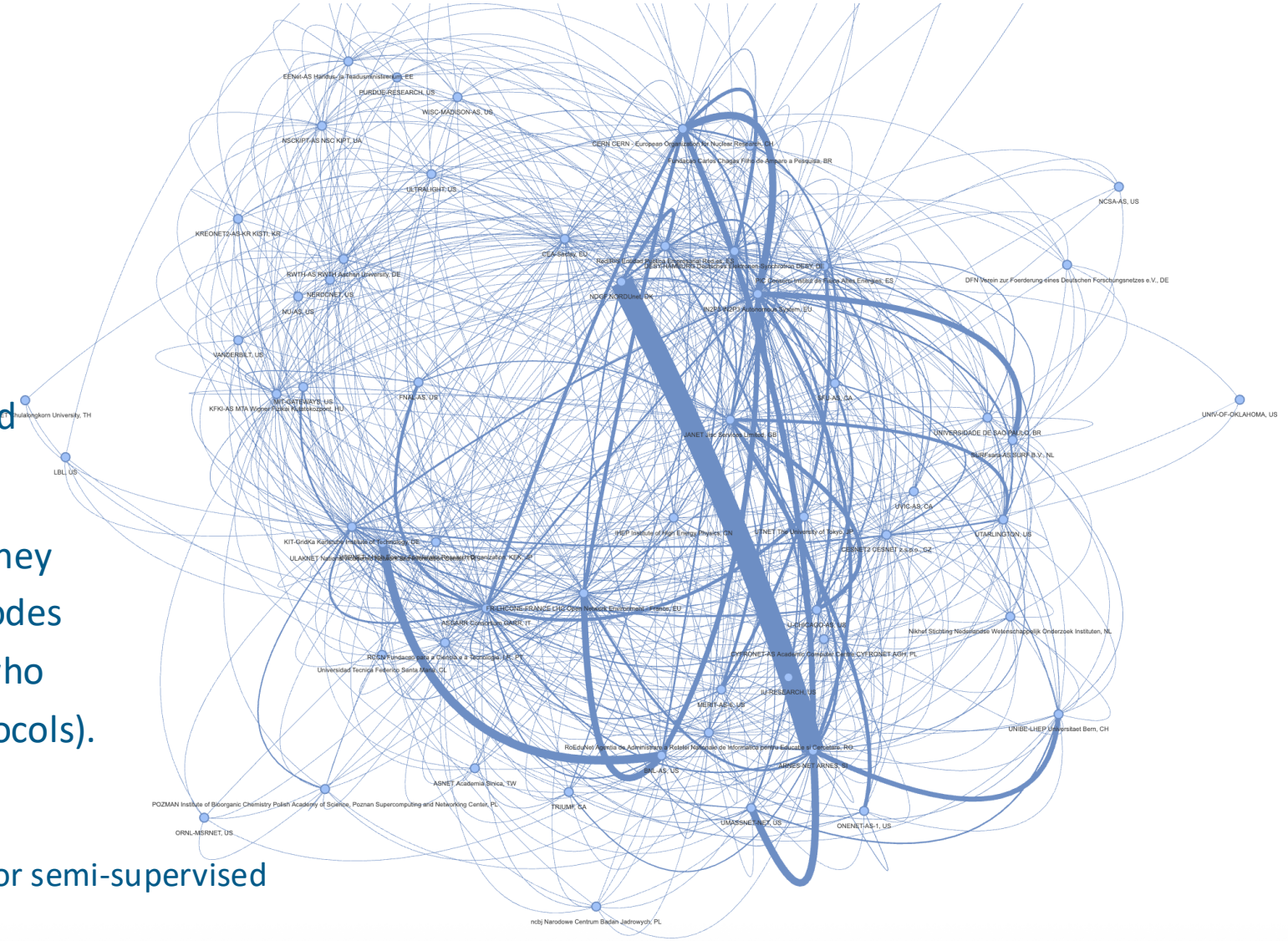
Why machine learning is a candidate for this?

- It scales to large, high-dimensional datasets (network flows) – more than 20 attributes, 30, 50… 15 min over 17.8 million flows
- It can adapt to the dynamics of science, scientist activities changes over time (new tools, new paths, new experiments) – not rule based
- No need pre-tagged data;
- Classification/Clustering models are super useful: anomaly detection and other applications;



Our experiment

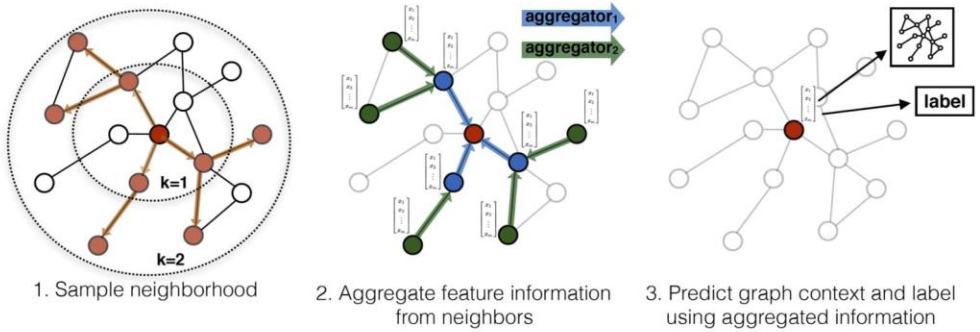
- Network traffic forms a graph structure;
- So… Graph Neural Networks (GNN) naturally model the **relational structure** and **topology** of network traffic;
- **GNNs are designed to learn on graphs** — they aggregate information from neighboring nodes to capture **contextual relationships** (e.g., who talks to whom, how often, over which protocols).
- Learning type:
  - Supervised, unsupervised, self-supervised or semi-supervised

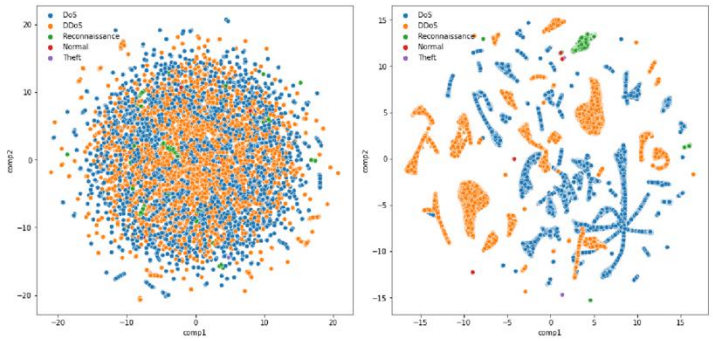


15 min of Netflow data
250.000 LHCONE VRF

Our reference

- Experiments on GNNs – self-supervised for intrusion detection
  - E-GraphSAGE[1], Anomal-E[2]
  - Outlier detection and clustering (unsupervised)





(a) raw data **(b)** edge embedded data

Fig. 4: Visualisation of dimensionality reduction a) Sample of BoT-IoT raw validation data, b)Sample of edge embeddings generated by E-GraphSAGE (Multiclass).

Table 6: NF-CSE-CIC-IDS2018-v2 results (4% contamination).

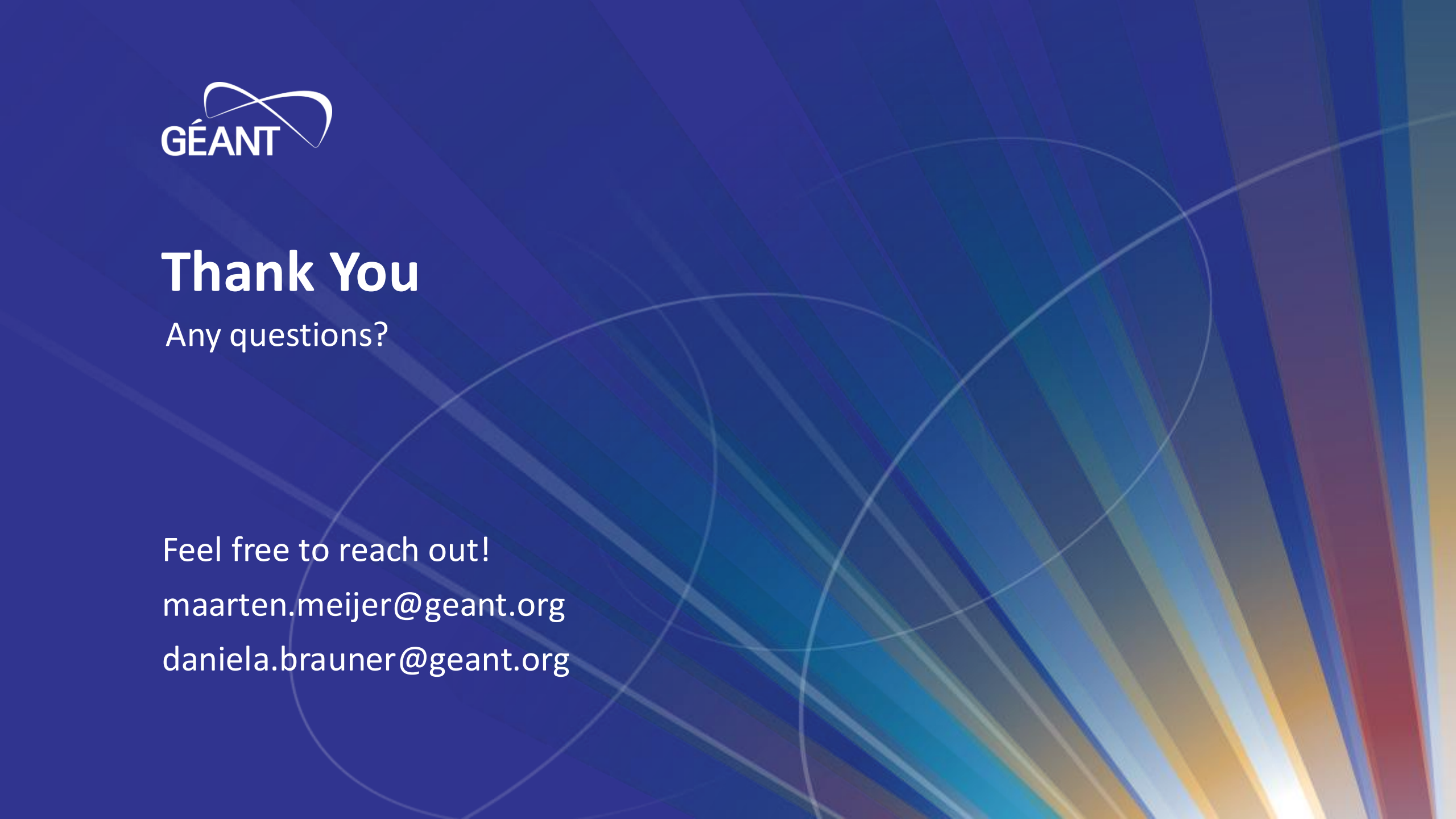
	Raw Features			Embeddings		
	Acc	Macro F1	DR	Acc	Macro F1	DR
PCA	85.91%	73.76%	74.71%	97.11%	92.57%	79.16%
IF	86.1%	74.09%	75.39%	89.79%	81.11%	91.84%
CBLOF	94.61%	86.18%	69.16%	97.80%	94.38%	82.67%
HBOS	88.81%	78.82%	84.22%	96.86%	91.89%	77.79%

[1] *Lo, W. W., Layeghy, S., Sarhan, M., Gallagher, M., & Portmann, M. (2022, April 25). E-GraphSAGE: A graph neural network based intrusion detection system for IoT.*
[2] *Caville, E., Lo, W. W., Layeghy, S., & Portmann, M. (2022). Anomal-E: A self-supervised network intrusion detection system based on graph neural networks.*

Testing experiment

- 500.000 flows
- NVIDIA A16 GPU (8x16GB VRAM), 72 CPU cores, 256GB memory

Process	Time taken
Loading and preprocessing	13s
Constructing graph	60s
Training Graph Neural Network	8m 54s (converging at 1700 train iterations)
Running model on test data	Instant!



Thank You

Any questions?

Feel free to reach out!

maarten.meijer@geant.org

daniela.brauner@geant.org