AI for Cybersecurity: Three Applications for Network Security

Pierre-François Gimenez
Inria researcher
PIRAT research team

Summer School – AI-driven Cyber Security
July 1st, 2025

Who am I?

Background

- 2018: PhD on machine learning at IRIT, Toulouse
- Since 2020: Researcher in a security team at Inria, Rennes
- I publish in both AI and security conferences

AI $\cap$ Cybersecurity $=$?

There are many applications of AI to cybersecurity!

- Side channel analysis
- Malware analysis
- Network intrusion detection
- Security data generation

Who am I?

Background

- 2018: PhD on machine learning at IRIT, Toulouse
- Since 2020: Researcher in a security team at Inria, Rennes
- I publish in both AI and security conferences

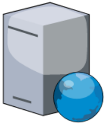
AI $\cap$ Cybersecurity $=$?

There are many applications of AI to cybersecurity!

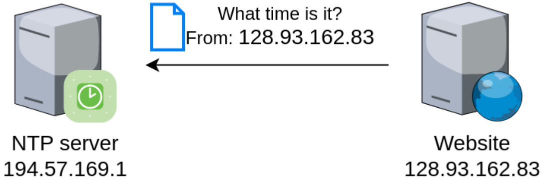
- Side channel analysis
- Malware analysis
- **Network intrusion detection**
- **Security data generation**

The following work were mostly done during Maxime Lanvin and Adrien Schoen PhDs

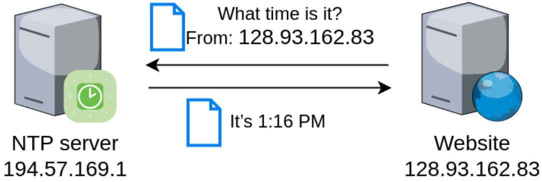
Simple denial of service attack



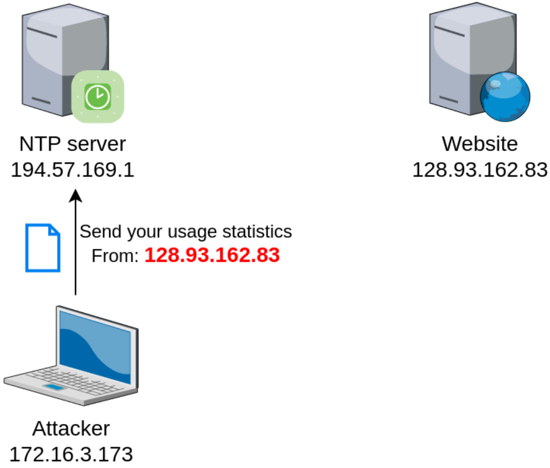
Simple denial of service attack



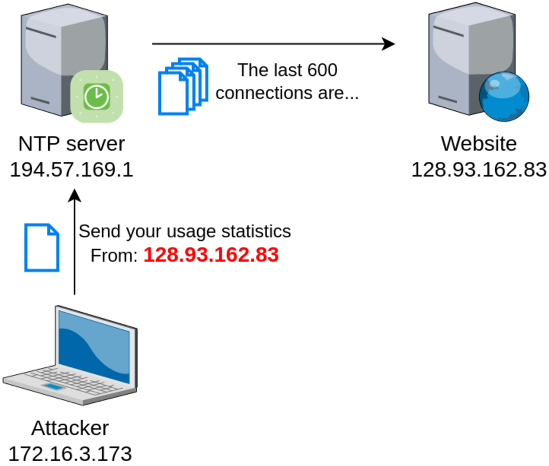
Simple denial of service attack



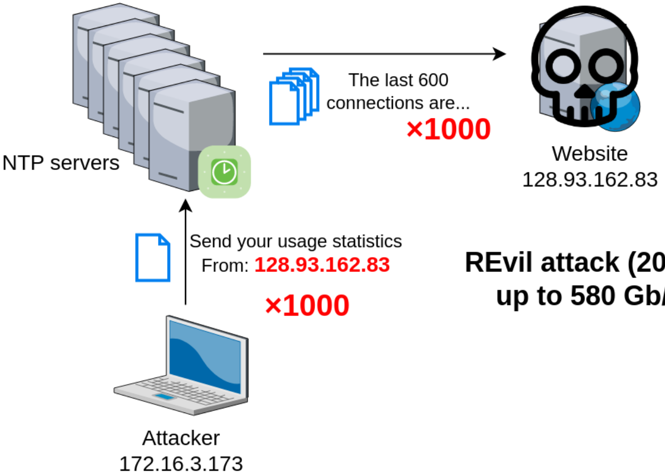
Simple denial of service attack



Simple denial of service attack



Simple denial of service attack

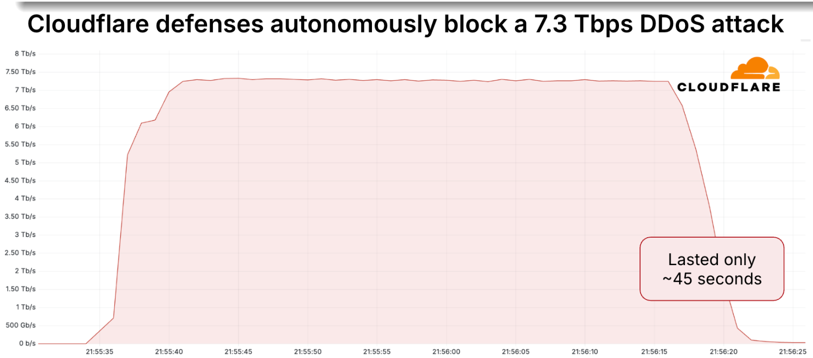


REvil attack (2020):
up to 580 Gb/s

Introduction

Systems are under attack

- Many untargeted, opportunistic attacks like password bruteforce
- Some targeted attacks with a huge power (e.g., DDoS attacks)
- Some very sophisticated attacks months or years in the making (SolarWinds, Stuxnet...)



In May 2025, an attack delivered 37.4 terabytes in 45 seconds

Information system security

Information system security

- Prevent the attack, detect it, and react
- Detection with **IDS**: *Intrusion Detection System*

Detection relies on observation

- System : OS and applications logs
- Network : network communications

Constraints

- Partial and heterogeneous observations
- Adversarial context: the attacker hides!


```
2024-05-06T23:24:16.806598+02:00
stellar-sheep sshd[16039]: Failed
password for pfg from 192.168.1.36
port 48650 ssh2
```

```
"ts": 1591367999.305988,
"id.orig_h": "192.168.4.76",
"id.resp_h": "192.168.4.1",
"id.resp_p": 53, "proto": "udp",
"service": "dns", "duration":
0.066851, "orig_bytes":
62, "resp_bytes": 141,
"conn_state": "SF", "orig_pkts":
2, "orig_ip_bytes": 118,
"resp_pkts": 2, "resp_ip_bytes":
197
```

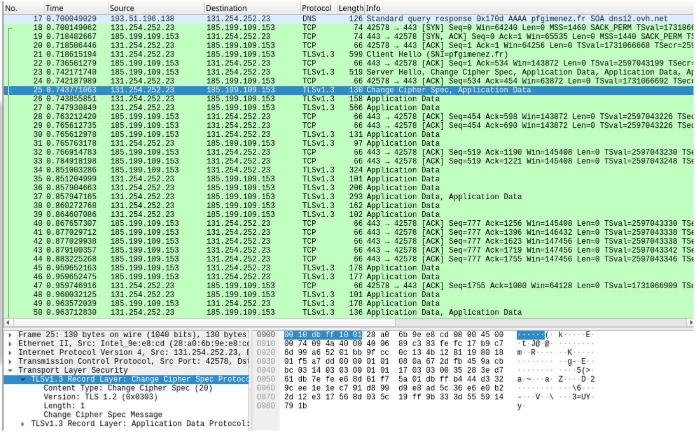
Outline

AI for network intrusion detection

Network data example

Network data

- Raw data consist of packets, regrouped in conversation
- Cybersecurity analysis typically rely on network flow records
- Network flows describe conversations statistically




```
ts,proto,src_ip,dst_ip,dst_port,fwd_packets,bwd_packets,fwd_bytes,bwd_bytes
1730800143,TCP,131.254.252.23,216.58.213.78,443,33,41,5988,1950
```

Two categories of detectors

Signature-based detection

Date: 2024-04-25 10:24:52+02:00
Source IP: 194.57.169.1
Destination IP: 128.93.162.83

Signature : alert udp any any -> any 123 (content:"|00 02 2A|"; offset:1; depth:3; byte_test:1,!&,128,0; byte_test:1,&,4,0; byte_test:1,&,2,0; byte_test:1,&,1,0; threshold: type both, track by_dst,count 2, seconds 60);

Potential attack using NTP!

Signatures database

- \+ quick, clear
- − regular updates, only documented attacks

Anomaly detection

Date: 2024-04-25 10:24:52+02:00
Source IP: 194.57.169.1
Destination IP: 128.93.162.83

Anomaly score: 7,6

Normal behavior model (generally with AI)

- \+ can detect undocumented attacks
- − false positives, no alert description

Two categories of detectors

Signature-based detection	Anomaly detection
Date: 2024-04-25 10:24:52+02:00 **Source IP:** 194.57.169.1 **Destination IP:** 128.93.162.83 **Signature :** alert udp any any -> any 123 (content:"|00 02 2A|"; offset:1; depth:3; byte_test:1,!&,128,0; byte_test:1,&,4,0; byte_test:1,&,2,0; byte_test:1,&,1,0; threshold: type both, track by_dst,count 2, seconds 60); **Potential attack using NTP!**	**Date:** 2024-04-25 10:24:52+02:00 **Source IP:** 194.57.169.1 **Destination IP:** 128.93.162.83 **Anomaly score: 7,6**
Signatures database	Normal behavior model (generally with AI)
+ quick, clear	+ can detect undocumented attacks
– regular updates, only documented attacks	– false positives, no alert description

AI for network security

The constraints of AI

- Typically, AI works on *vectors*
- These vectors must always have the same size
- In practice, it is not always the case

The need of representation

Several techniques are used to transform data into a fixed vector

- Images are rescaled
- Words are split into subwords (tokens)

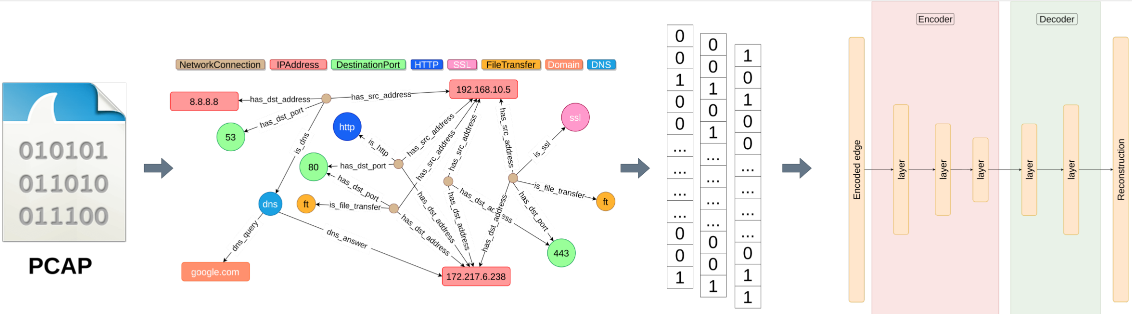
In network security

- Network flow are vectors
- There is no standard way to analyze *packets*

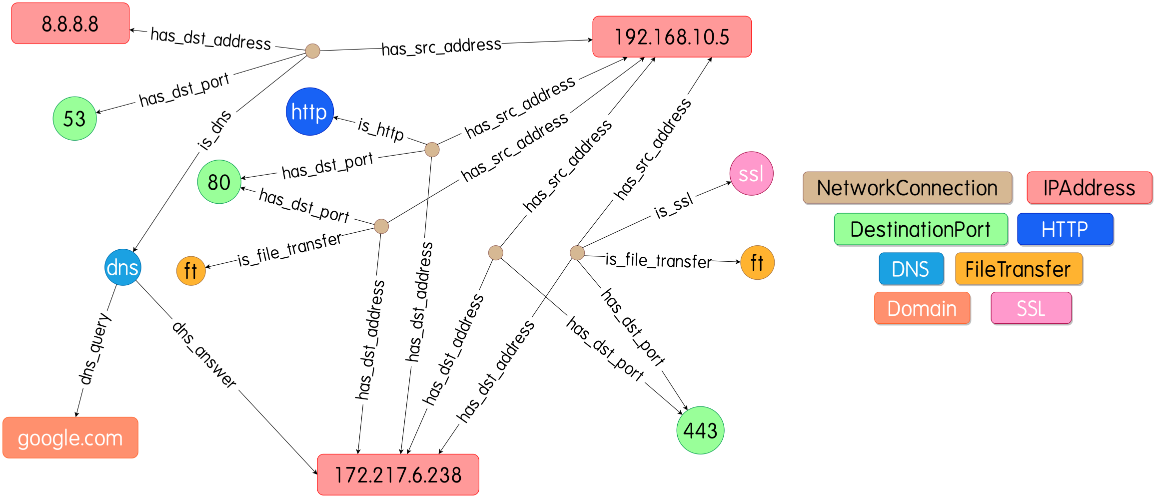
Overview of our approach Sec2graph

Structure of our approach

- Probes capture the network data
- These data are merged into a graph structure
- The graph is transformed into a format usable with a deep learning model
- The model affects an anomaly score to each data point



Security objects graph example



Security objects graph

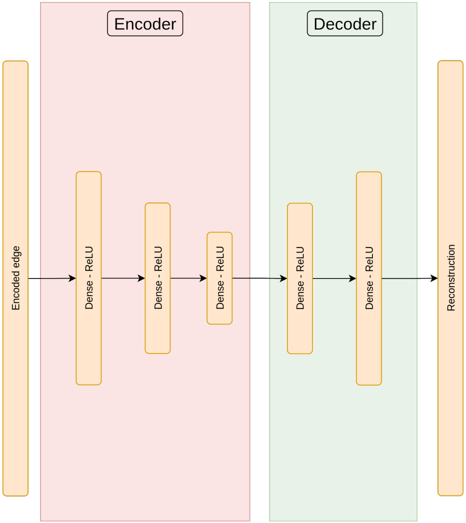
Nodes

- Each node type corresponds to a "security object":
  - protocols: DNS, SSH, DCERPC, SNMP, FTP, DHCP, HTTP, SMTP
  - network data: port, MAC address, IP address, network connection, URI, domain
  - and others
- Nodes contain a set of attributes related to these objects

Edges

- Edges are typed and oriented
- They do not contain attributes
- An edge between two nodes means that these two nodes are found within the same event

Anomaly detection: Autoencoder (AE)



Autoencoder

An autoencoder is a deep learning architecture with a bow-tie shape

Learning

Minimisation of the reconstruction error between the input vector and its reconstructed version

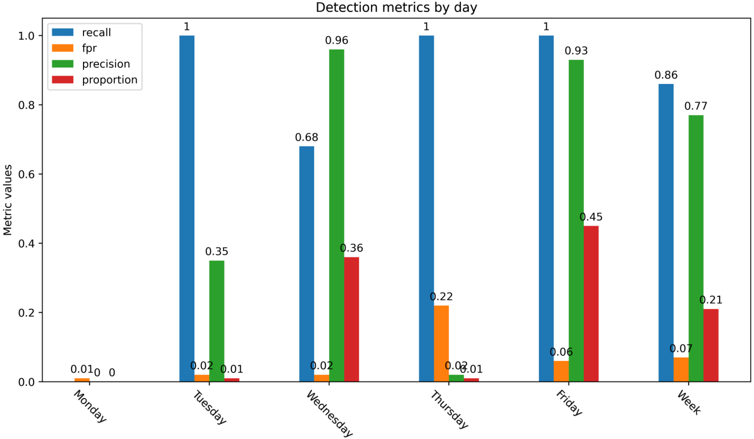
Detection

Raise an alert when the reconstruction error is above a threshold

Performances on CIC-IDS2017

Performances

Recall is mostly good but we have a very high false positive (22%!) on Thursday



Explainable AI for anomaly detection

How to explain the predictions?

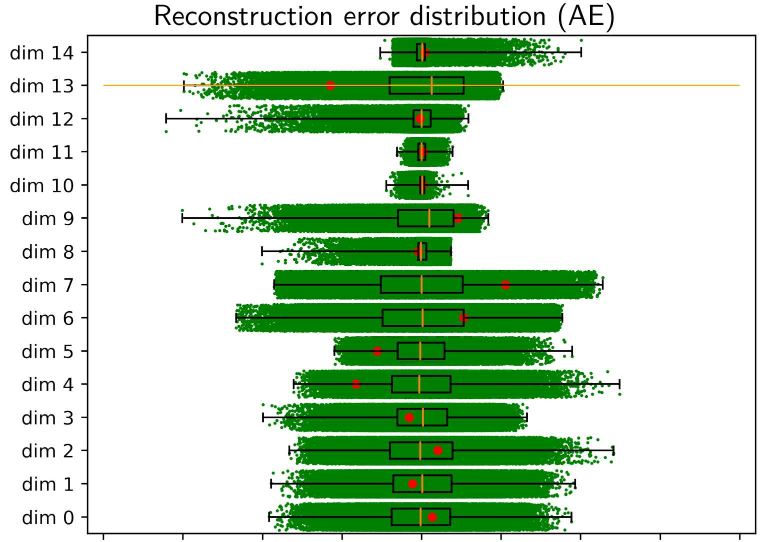
The issue

- Explanations could help us understand the false positives
- There exists a lot of explanation techniques... (LIME, salient maps, counterfactual explanation...)
- ...but little work on explanations for unsupervised learning!

First, naive approach

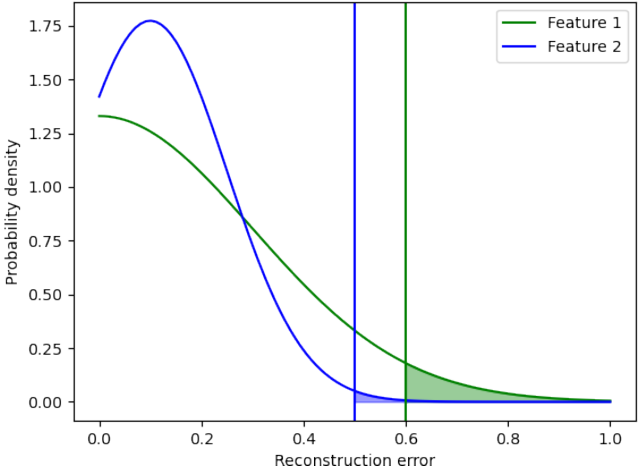
- We can compute the contribution of each feature to the global reconstruction error
- However, we found out this idea does not produce satisfactory explanations:
  - Some features are always difficult to reconstruct because of their high variance
  - Some features are always very faithfully reconstructed, and even a small reconstruction error may reveal an anomaly

What it looks like



Limitations

Comparison of the reconstruction errors of two dimensions



Key Idea

The highest reconstruction error is not always an indication of the most abnormal dimension.

Our approach

This area is called the p-value:

$$p_i = \frac{\#\{r_i \geq e_i\}}{\#\{r_i\}}$$

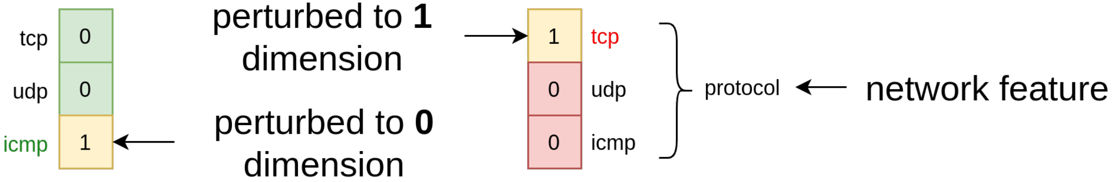
Experimental protocol

Protocol

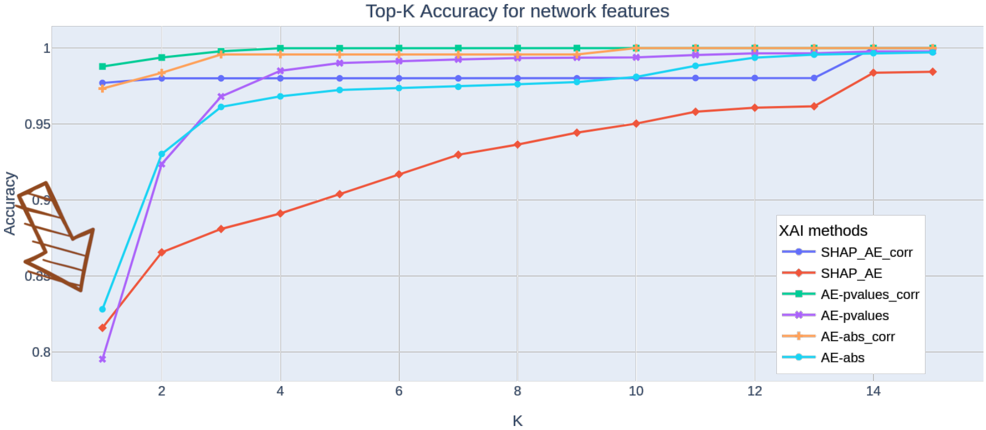
- Inject noise in a known network characteristic of vectors
- Assess ability of XAI methods to find the noisy network characteristic

Experiment with AE-abs (intuitive method), SHAP_AE (state of the art), AE-pvalues (our method)

Example of noise insertion in the protocol characteristic



Benchmark results



Top-K accuracy

Proportion of samples for which the right explanation is among the Top-K explanations. But sometimes several explanations are correct. . .

Several correct explanations

$$1 + 1 = 0$$

Where is the error?

We can all agree there is an error. But where do you think it is?

- 0 should be 2
- $+$ should be $-$
- 1 should be -1
- $=$ should be $>$
- "(mod 2)" is missing
- "is false" is missing

Several correct explanations

$$1 + 1 = 0$$

Where is the error?

We can all agree there is an error. But where do you think it is?

- 0 should be 2
- $+$ should be $-$
- 1 should be -1
- $=$ should be $>$
- "(mod 2)" is missing
- "is false" is missing

Several correct explanations

$$1 + 1 = 0$$

Where is the error?

We can all agree there is an error. But where do you think it is?

- 0 should be 2
- $+$ should be $-$
- 1 should be -1
- $=$ should be $>$
- "(mod 2)" is missing
- "is false" is missing

Several correct explanations

$$1 + 1 = 0$$

Where is the error?

We can all agree there is an error. But where do you think it is?

- 0 should be 2
- $+$ should be $-$
- 1 should be -1
- $=$ should be $>$
- "(mod 2)" is missing
- "is false" is missing

Several correct explanations

$$1 + 1 = 0$$

Where is the error?

We can all agree there is an error. But where do you think it is?

- 0 should be 2
- $+$ should be $-$
- 1 should be -1
- $=$ should be $>$
- "(mod 2)" is missing
- "is false" is missing

Several correct explanations

$$1 + 1 = 0$$

Where is the error?

We can all agree there is an error. But where do you think it is?

- 0 should be 2
- $+$ should be $-$
- 1 should be -1
- $=$ should be $>$
- "(mod 2)" is missing
- "is false" is missing

Several correct explanations

$$1 + 1 = 0$$

Where is the error?

We can all agree there is an error. But where do you think it is?

- 0 should be 2
- $+$ should be $-$
- 1 should be -1
- $=$ should be $>$
- "(mod 2)" is missing
- "is false" is missing

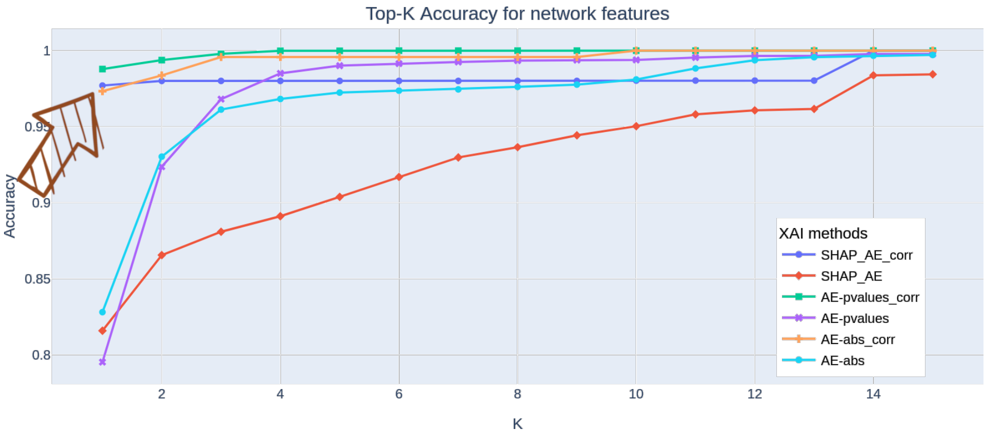
Several correct explanations

$$1 + 1 = 0$$

Where is the error?

We can all agree there is an error. But where do you think it is?

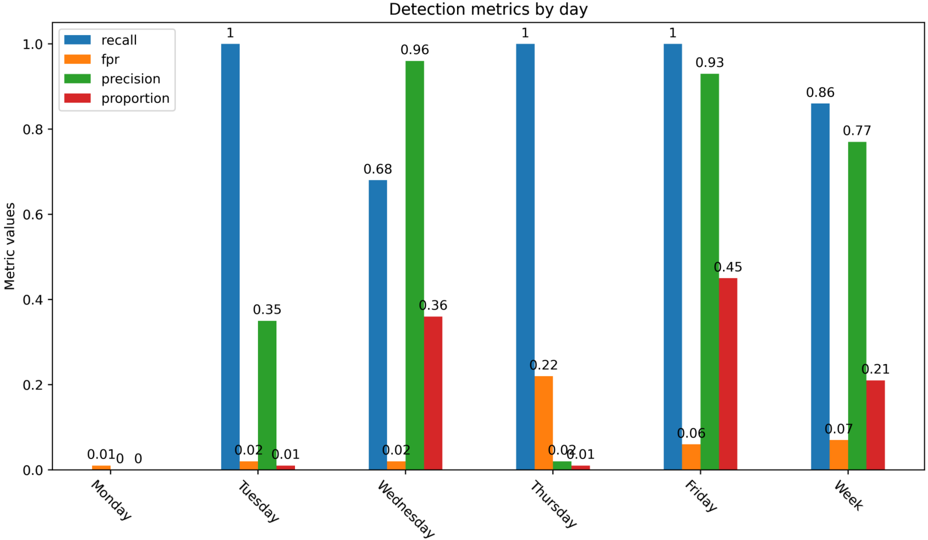
- 0 should be 2
- $+$ should be $-$
- 1 should be -1
- $=$ should be $>$
- "(mod 2)" is missing
- "is false" is missing



A more realistic evaluation

Evaluation modification: accepting correlated features as correct explanations

Remember that?...



What is the issue with CIC-IDS2017?

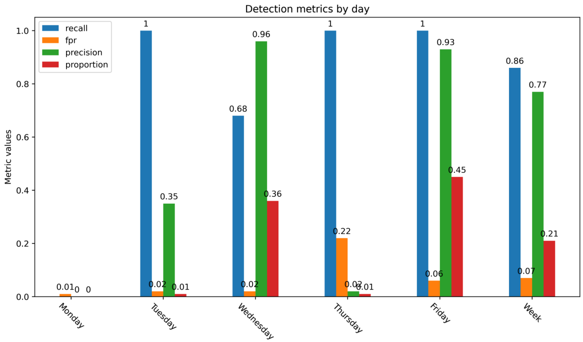
Not only one...

- Labeling issue: CIC-IDS2017 has a scan attack on Thursday that is not corrected labeled. About 70,000 flows of scan are labeled as "benign"!
- Duplication issue: probably due to a badly configured probe, on average 500,000 packets are duplicated per day. It caused the CSV files to contain bad data
- Shortcut learning possible: the tools use their default user agent
- And a few minors issues

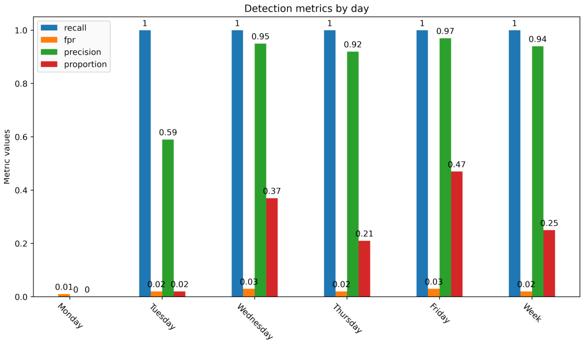
Corrected CIC-IDS2017 files: `https://gitlab.inria.fr/mlanvin/crisis2022`

These results make us confident in the usefulness of our explanation method

Updated results on CIC-IDS2017



Before CIC-IDS2017 correction



After CIC-IDS2017 correction

Flawed datasets

Public dataset

- Most IDS research relies on public dataset
- It allows for reproducible results and comparison between methods
- A few datasets are popular: NSL-KDD, CIC-IDS-2017/2018, and a few others

Criticisms

We are not the only ones finding issues in datasets

- NSL-KDD is still used but obsolete
- 4 articles have been published on issues on CIC-IDS-2017 alone
- Other datasets are also criticized

Common issues: unrealistic testbed, duration too low, badly configured tool and probe...

Alternatives

Real data

- Difficult to obtain/share due to confidentiality and privacy reasons
- Typically not labeled

Testbeds

- Difficult to create: it must include fake users with online activity with a wide range of behaviors
- Slow: we need one month to generate one month of data

Data generation with AI

- Could be much faster than testbed
- Is AI mature enough? How to explain the generation process and to evaluate the data?

AI for synthetic data generation

GenAI: GANs

Generative Adversarial Networks

Two neural networks compete: one to generate fake data, the second one to find whether some data is fake or genuine

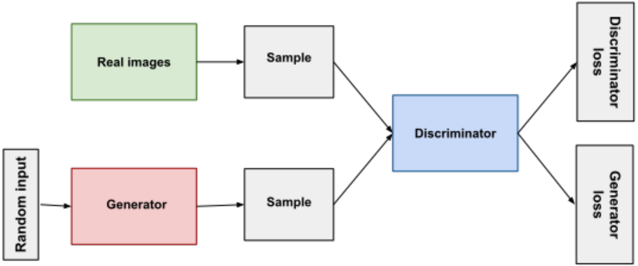
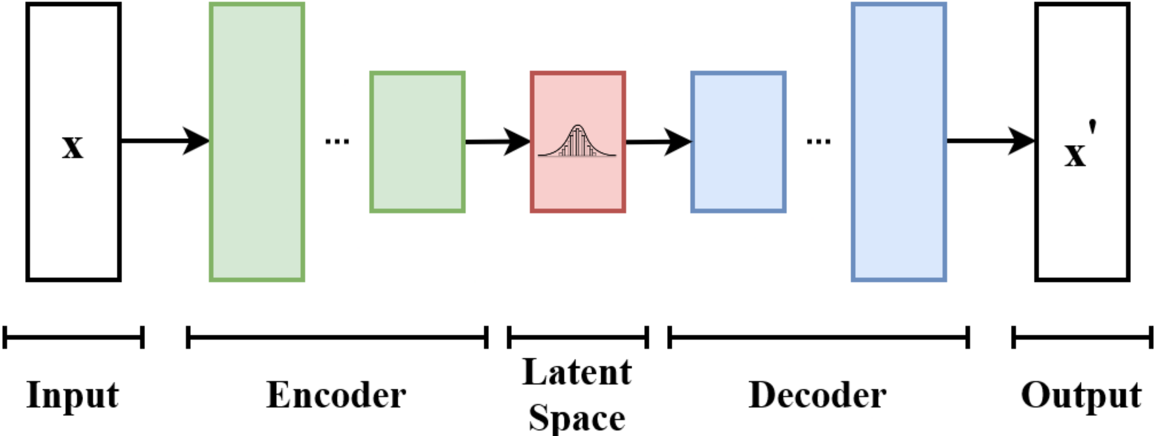


Image generated with StyleGAN (2019)

Variational AutoEncoders

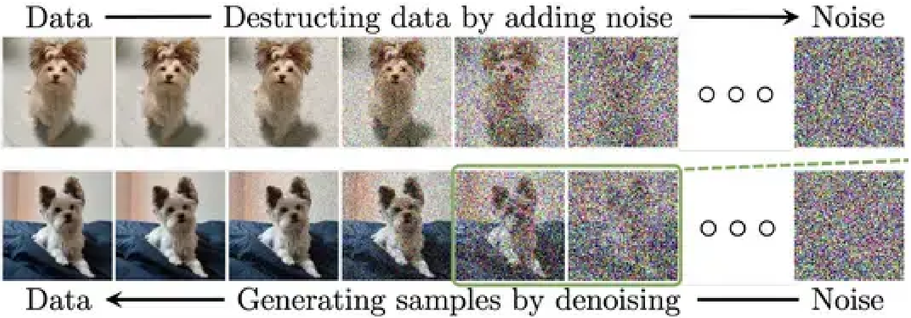
An autoencoder used to generate data by decoding random vectors in the latent space



GenAI: diffusion models

Diffusion models

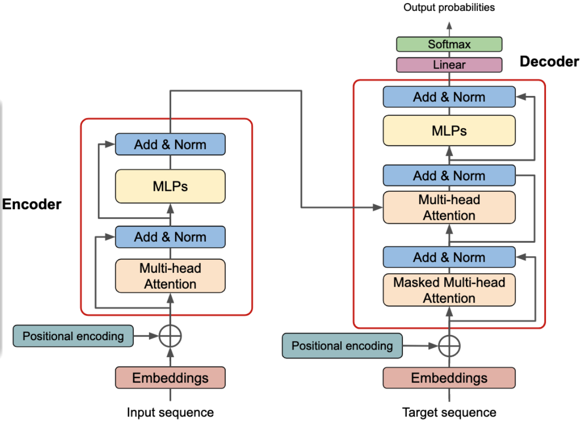
A model trained to "denoise" data. Applied several times in a row to create images from noise.



GenAI: LLMs

Transformers

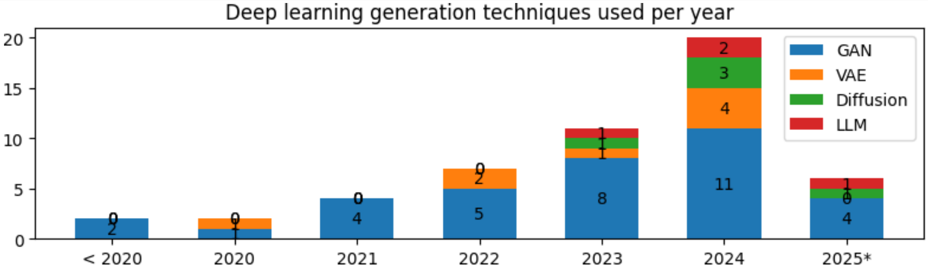
- A model that predicts the next token based on the previous ones. The generation focuses on the relevant tokens in the context window
- It is the base of LLMs: ChatGPT, Gemini, Mistral, Llama, etc.



GenAI for network generation

And in network generation?

- A quick growth of works on synthetic network traffic generation
- All previous techniques are used to generate synthetic network traffic
- However, the quality of the generated data is still low
- Lack of explainability makes progress slower



GenAI for network generation

A big limitation: dependencies within the data

- Intra-flow dependency
  - the port depends on the destination IP
  - the number of packets depends on the application protocol
- Inter-flow dependency:
  - DNS query then HTTP(S)
  - IMAP request then HTTP(S)

Our work

We propose FlowChronicle as an explainable generation method not based on deep learning

FlowChronicle

FlowChronicle: a novel approach

- Pattern language
  - Captures intra-flow and inter-flow dependencies
  - Summarizes data with non-redundant patterns
- Data generation
  - Produces realistic traffic respecting protocols
  - Preserves temporal dependencies
- Explainability
  - Patterns are interpretable and auditable

FlowChronicle

What is a pattern?

Frequently occurring substructure in data

Pattern Mining

- Define the set of possible patterns, named the "pattern language"
- Find a small set of patterns that best describes the data
- More precisely, we use the patterns to compress the data: higher the compression, better the patterns

Pattern description

Pattern language

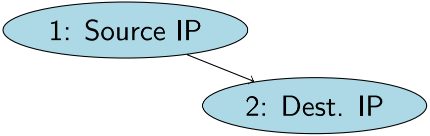
Each pattern has two parts: a partially defined flow, and a Bayesian network

- Fixed values are defined in the partial flow
- the distribution of Free variables is defined in the Bayesian network
- Reused variables are always equal to some Free variable

Partial flows

Source IP	Dest. IP	Dest. Port
β_A	8.8.8.8	53
A	β	80

Bayesian Network



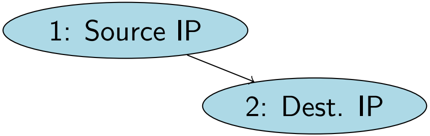
In reality there are more columns!

Pattern description

Partial flows

Source IP	Dest. IP	Dest. Port
β_A	8.8.8.8	53
A	β	80

Bayesian Network



Example

- Here, there are two flows
- The first flow is contacting 8.8.8.8 on port 53 (DNS). The source IP is random
- The second flow has the same source IP as the first flow, and is contacting a destination IP that is random and depends on the first source IP, on port 80 (HTTP)

Our goal is to learn ("mine") such patterns

Pattern mining algorithm

Pattern Search:

1. Initialize Model with an empty pattern
2. Generate Pattern Candidates from existing patterns $p \in M$.
   - By extending with an attribute
   - By merging existing patterns
3. Test candidates for addition:
   - Cover the datasets with the patterns
   - Add patterns when it reduces MDL score: $L(D \mid M) + L(M)$

Dataset cover

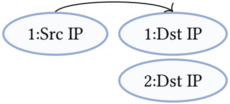
Model — Pattern and Bayesian Network:

ϵ :

$$\begin{bmatrix} \beta & \beta & \beta \end{bmatrix}$$

p :

$$\begin{bmatrix} \beta_A & \beta & 993 \end{bmatrix}$$
$$\begin{bmatrix} A & \beta & 80 \end{bmatrix}$$



q :

$$\begin{bmatrix} \beta_A & 8.8.8.8 & 52 \end{bmatrix}$$
$$\begin{bmatrix} A & \beta_B & 443 \end{bmatrix}$$
$$\begin{bmatrix} B & \beta & 3306 \end{bmatrix}$$

Data and Pattern Windows:

Time	**Src IP**	**Dst IP**	**Port**
12	134.96.235.78	142.251.36.5	993
56	134.96.235.129	8.8.8.8	52
89	134.96.235.78	212.21.165.114	80
113	134.96.235.129	198.95.26.96	443
145	198.95.26.96	198.95.28.30	3306
156	134.96.235.78	134.96.234.5	21
178	134.96.235.36	185.15.59.224	993
206	134.96.235.36	128.93.162.83	80

Loss function

Length of data given the model:

$$L(D \mid M) = \sum_{p \in M} \left(L_{\mathbb{N}}(|W_p|) + L(W_p)\right)$$

where:

$$L(W_p) = \sum_{i=1}^{|W_p|} \left(L(t_1 \text{ of } w_i) + \sum_{k=2}^{|p|} L(t_k \text{of } w_i \mid t_{i-1}) \right) - \log(Pr(w_i | BN_p, \{w_j | j < i\}))$$

Length of Model:

$$L(M) = L_{\mathbb{N}}(|M|) + \sum_{p \in M} L(p)$$

Length of one pattern:

$$L(p) = L_{\mathbb{N}}(|p|) + \left(\sum_{j=1}^{|p|} L(X[j]|p) \right) + L(BN_p)$$

Data quality evaluation

Hard to evaluate

- No standard metrics
- Evaluation often partial

Proposition

A set of evaluating metrics:

Realism	: could the data actually exist?
Diversity	: do we generate the diversity of behavior from the training set?
Novelty	: can the generator create data absent from the training set?
Compliance	: do the generated data comply with the technical specifications?

We do not consider privacy yet

Experimental protocol

Training data

We use the CIDDS 001 dataset: train on one week of traffic and generate one week of traffic

Baselines

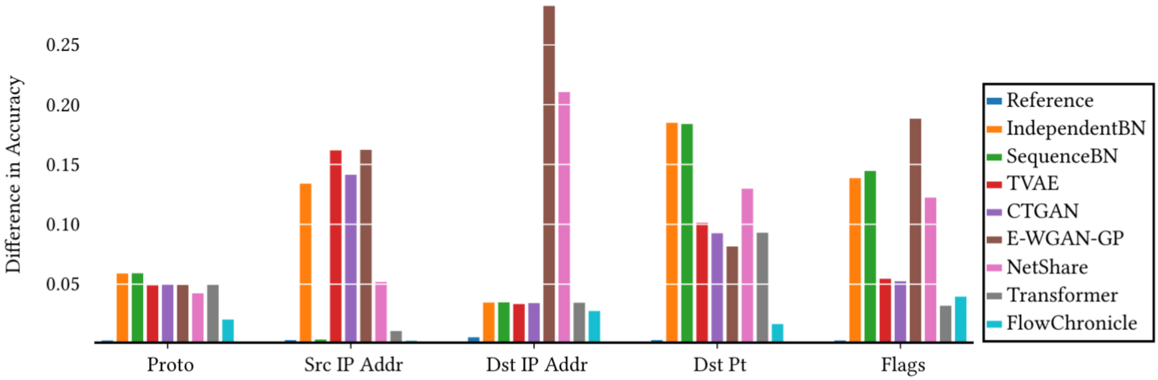
We compare FlowChronicle with:

- Bayesian networks
- VAE
- GAN
- Transformers
- "Reference": actual data from the same dataset to simulate the best generative method

FlowChronicle: generation quality

	Density	CMD	PCD	EMD	JSD	Coverage	DKC	MD	Rank
	Real. $\uparrow$	*Real.* $\downarrow$	*Real.* $\downarrow$	*Real./Div.* $\downarrow$	*Real./Div.* $\downarrow$	*Div.* $\uparrow$	*Comp.* $\downarrow$	*Nov.* $=$	*Average Ranking*
Reference	**0.69**	**0.06**	**1.38**	**0.00**	**0.15**	**0.59**	**0.00**	**6.71**	-
IndependentBN	0.24	0.22	2.74	*0.11*	0.27	0.38	0.05	5.47	5.25
SequenceBN	0.30	**0.13**	2.18	0.08	0.21	0.44	**0.02**	5.51	3.875
TVAE	0.49	0.18	1.84	**0.01**	0.30	0.33	0.07	5.17	4.125
CTGAN	**0.56**	0.15	**1.60**	0.01	**0.15**	**0.51**	*0.11*	**5.70**	**3.0**
E-WGAN-GP	*0.02*	0.34	*3.63*	0.02	0.38	*0.02*	0.07	4.66	7.0
NetShare	0.32	0.28	**1.47**	0.03	0.36	0.22	0.05	3.82	5.25
Transformer	**0.62**	*0.78*	3.62	**0.00**	*0.55*	0.03	0.05	*3.75*	*5.375*
FlowChronicle	0.41	**0.03**	2.06	0.02	**0.10**	**0.59**	**0.02**	**5.87**	**2.125**

FlowChronicle: temporal generation quality



Overall, FlowChronicle outperforms other GenAI techniques and is explainable

Conclusion

Conclusion

AI + Cybersecurity = ♡

- There are many applications of AI to cybersecurity
- I presented three of them:
  - Network intrusion detection
  - Explainable AI for anomaly detection
  - Synthetic network traffic generation

Current limits of AI

- AI is not a silver bullet for cybersecurity (yet)
- AI-based IDS still raise too many false positives
- Lack of explainability is a big drawback
- Generation performances are not that great

But AI's progress is fast and some of these limits could soon disappear