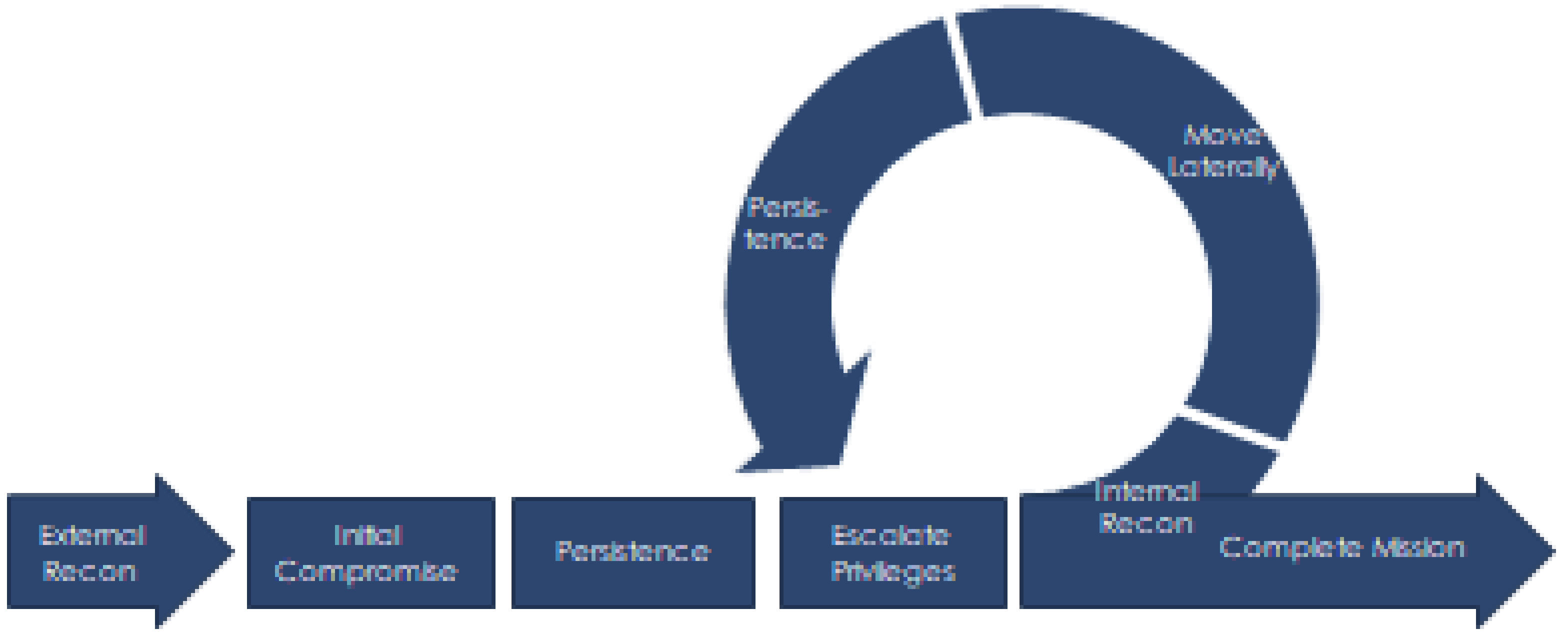
ML-based Network Intrusion Detection: What can be done in practice?

Cédric Lefebvre (Custocy)
Gregory Blanc (Télécom SudParis, Institut Polytechnique de Paris)

Cyber in Occitanie, 8 July 2025, Font-Romeu

A network attack

Une attaque sur un réseau d'entreprise



Un référentiel ?

https://attack.mitre.org/

Name	Description
Reconnaissance	The adversary is trying to gather information they can use to plan future operations.
Resource Development	The adversary is trying to establish resources they can use to support operations.
Initial Access	The adversary is trying to get into your network.
Execution	The adversary is trying to run malicious code.
Persistence	The adversary is trying to maintain their foothold.
Privilege Escalation	The adversary is trying to gain higher-level permissions.
Defense Evasion	The adversary is trying to avoid being detected.
Credential Access	The adversary is trying to steal account names and passwords.
Discovery	The adversary is trying to figure out your environment.
Lateral Movement	The adversary is trying to move through your environment.
Collection	The adversary is trying to gather data of interest to their goal.
Command and Control	The adversary is trying to communicate with compromised systems to control them.
Exfiltration	The adversary is trying to steal data.
Impact	The adversary is trying to manipulate, interrupt, or destroy your systems and data.

Pourquoi regarder le réseau ?

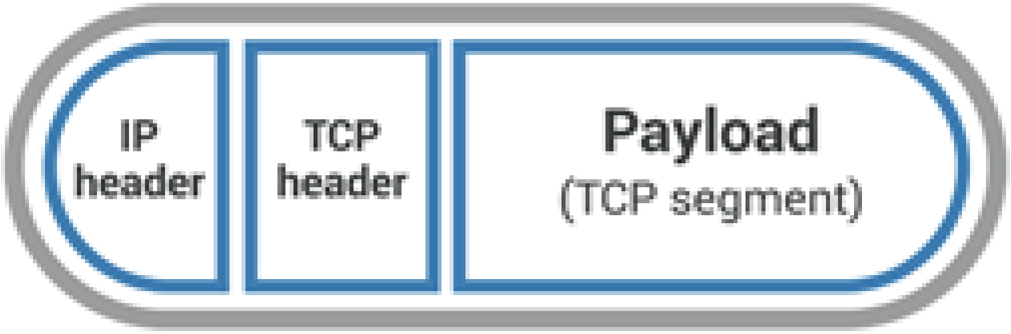
Discovery	The adversary is trying to figure out your environment.
Lateral Movement	The adversary is trying to move through your environment.
Collection	The adversary is trying to gather data of interest to their goal.
Command and Control	The adversary is trying to communicate with compromised systems to control them.
Exfiltration	The adversary is trying to steal data.

Cas concret d'une attaque

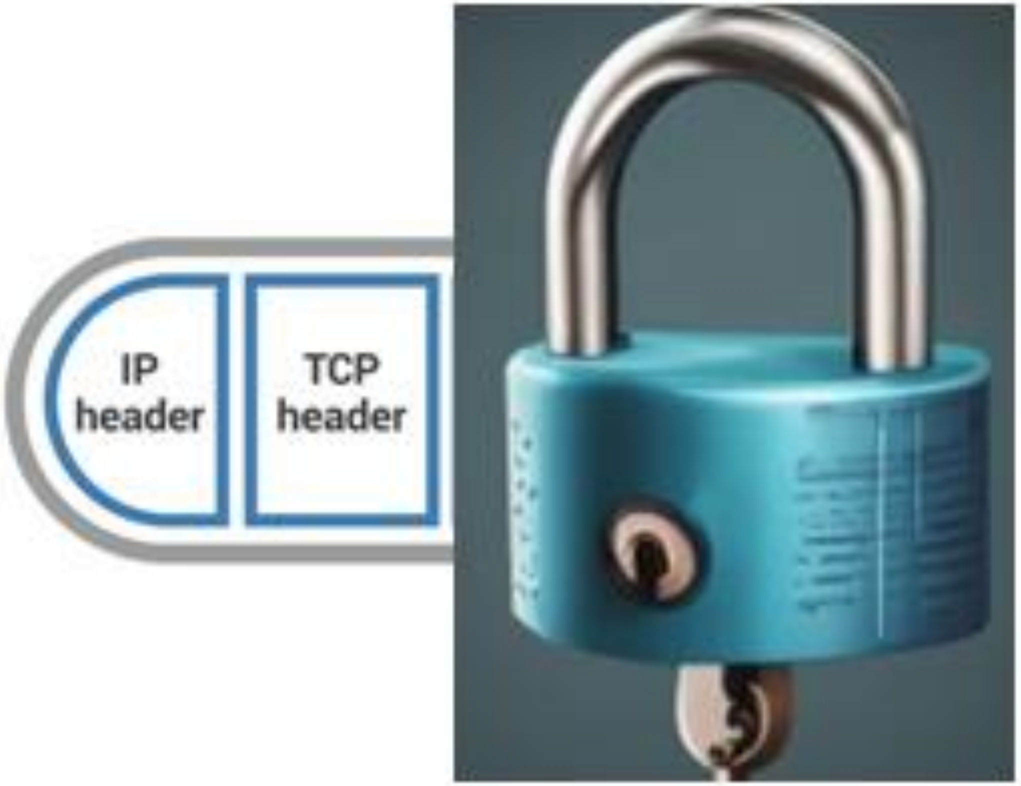
.001	Remote Services: Remote Desktop Protocol	During the SolarWinds Compromise, APT29 used RDP sessions from public-facing systems to internal servers. [5]

Application Layer Protocol: Web Protocols	During the SolarWinds Compromise, APT29 used HTTP for C2 and data exfiltration.[4]

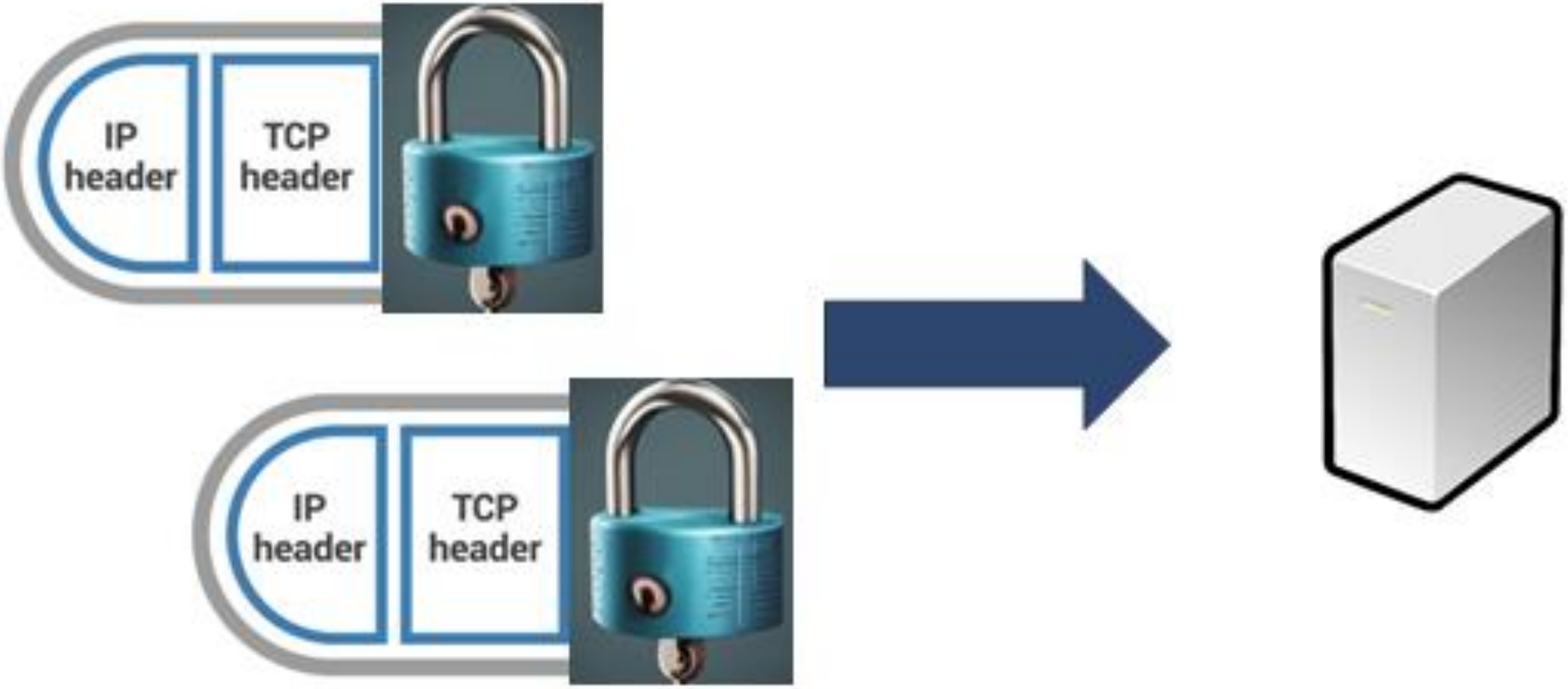
Comment détecter sur le réseau ?



Problème, c'est chiffré



Les méta-données



Caractéristiques visibles

La volumétrie	Les protocoles applicatifs utilisés	La taille et le temps entre les paquets	Qui parle à qui

Quelles procédures / Comment ?

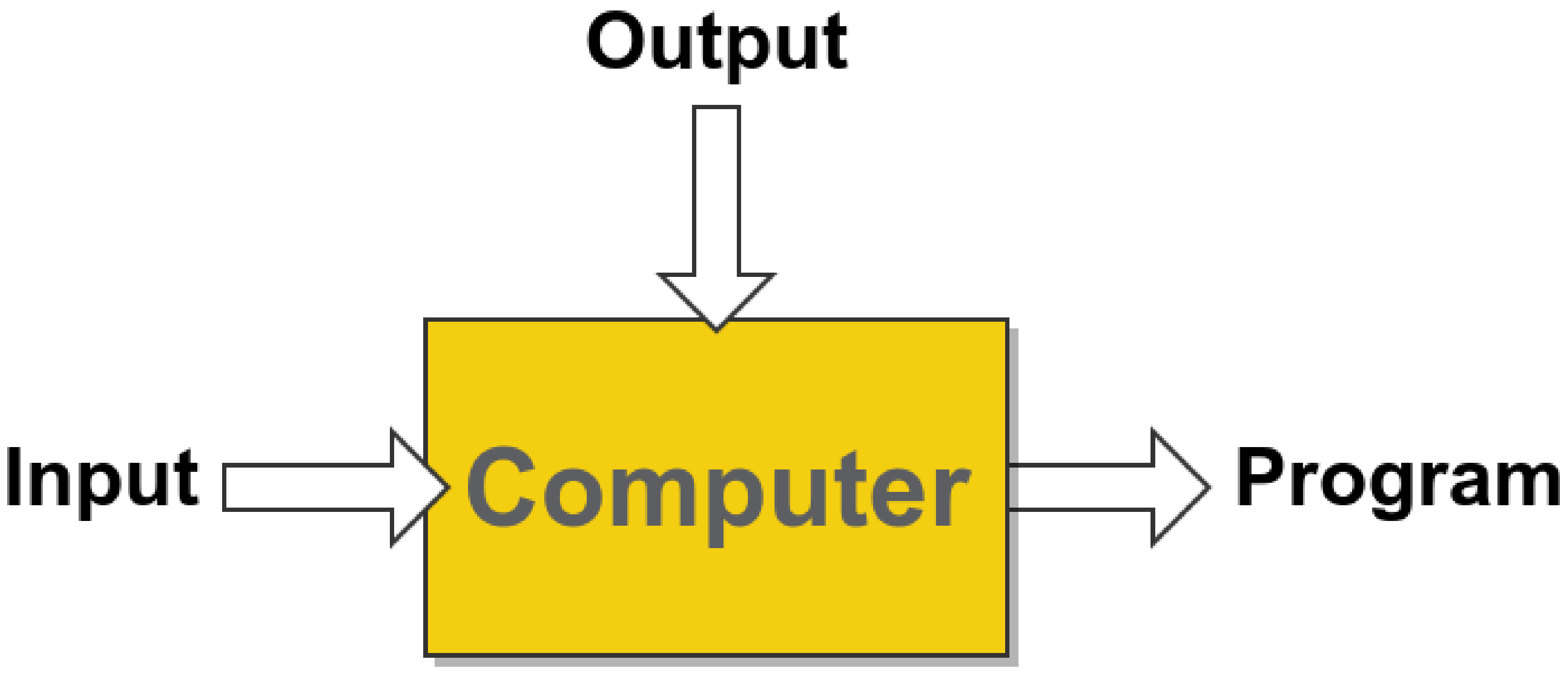
- Command and Control
- Découverte du réseau
- Mouvements latéraux
- Exfiltration de données
- Autres

Refresher on Artificial Intelligence / Machine Learning

Machine learning: from experience

source: underscore.vc

Machine learning: from approximation



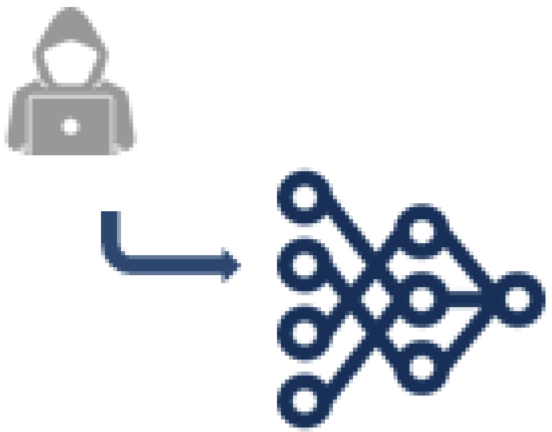
2 types de Machine Learning

Supervisé

Non supervisé

Les NOUVELLES attaques sont difficiles à détecter

Supervisé



Non supervisé

2 types de Machine Learning

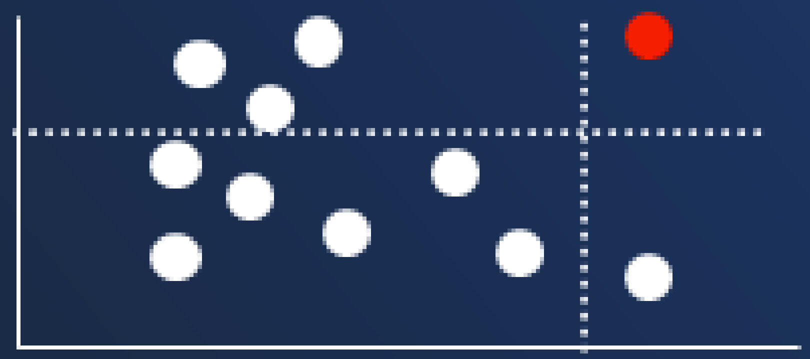
Supervisé

Non supervisé

2 types de Machine Learning

Supervisé

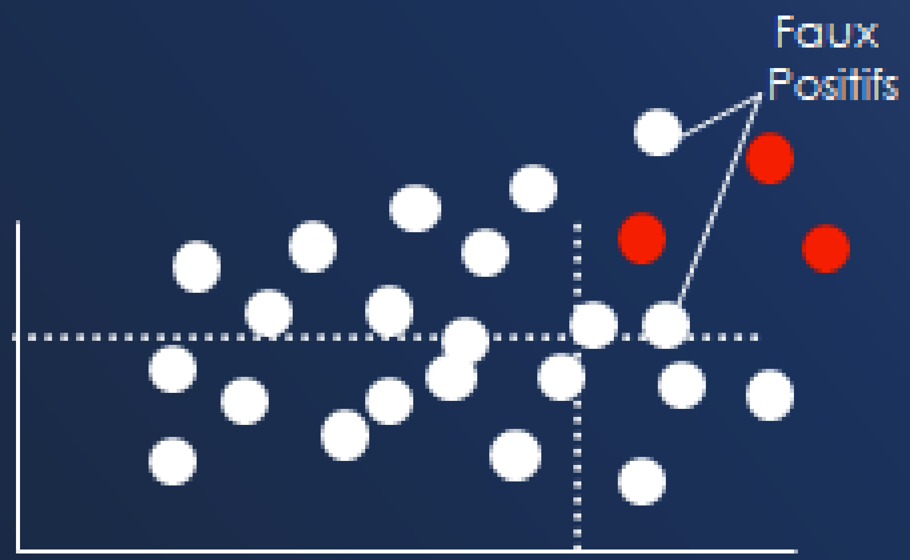
Non supervisé



Trop de faux positifs

Supervisé

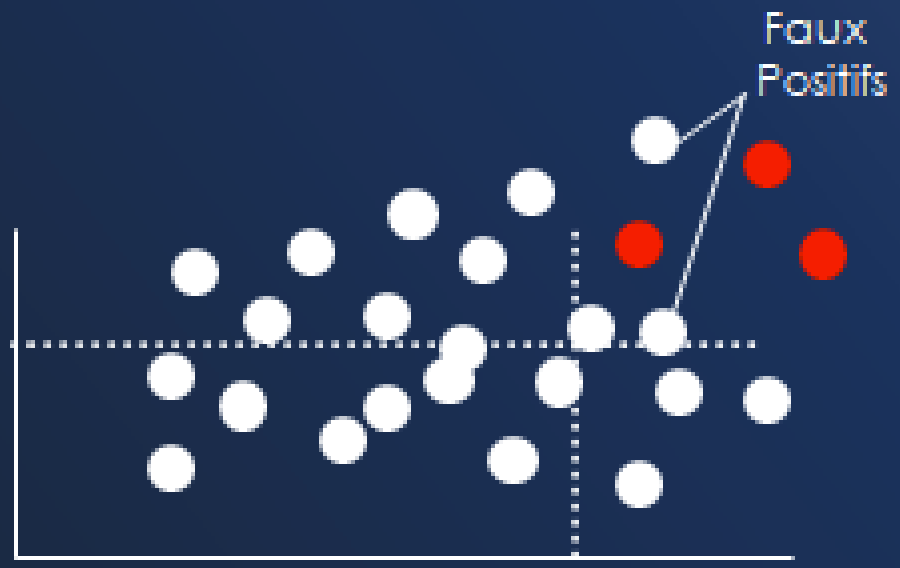
Non supervisé



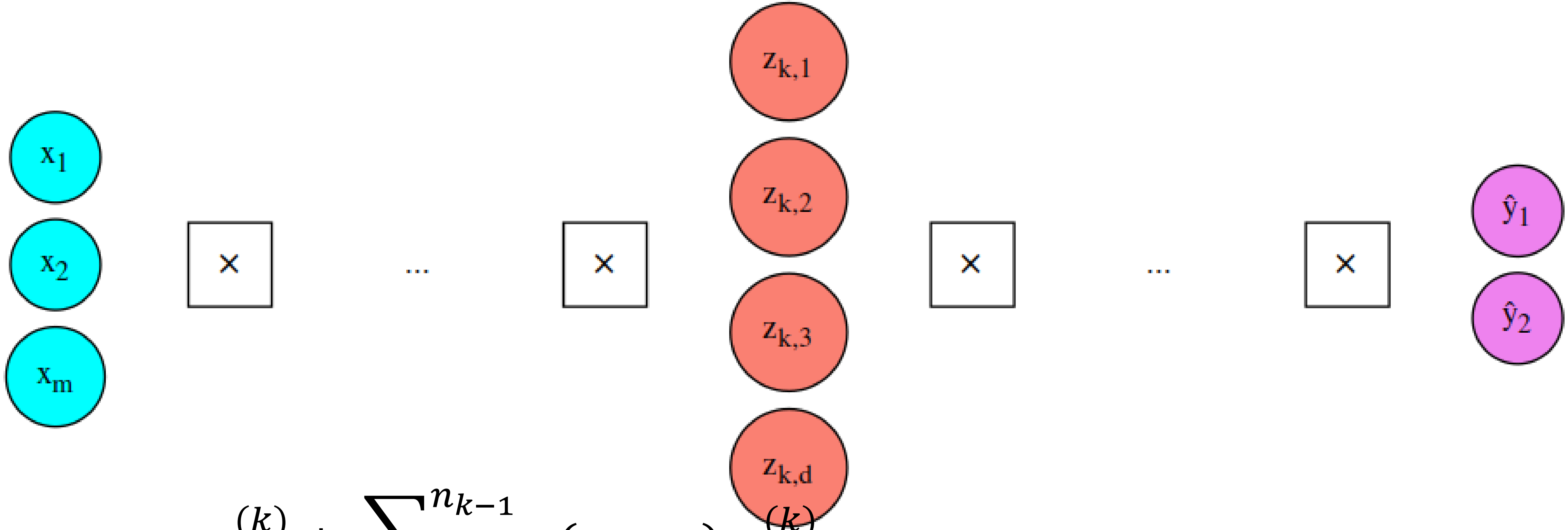
Ces IA sont quand même informatives

Supervisé

Non supervisé



Deep learning: deep neural network



$$z_{k,i} = w_{0,i}^{(k)} + \sum_{j=1}^{n_{k-1}} g(z_{k-1,j}) w_{j,i}^{(k)}$$

source: introtodeeplearning.com

deep neural network’s hidden layer

Deep learning: quantifying loss

$$J(\boxed{W}) = \frac{1}{\boxed{n}} \sum_{i=1}^{n} L(\boxed{f(x^{(i)}; W)}, \boxed{y^{(i)}})$$

weights matrix

number of instances

predicted value

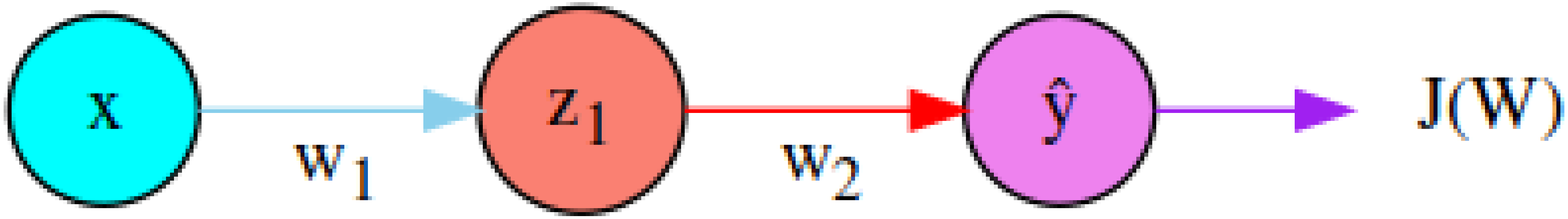
actual value

Deep learning: loss optimization

Objective: find optimal $W^* = \text{argmin}_\text{W}\, \text{J(W)}$

1. Initialize weights randomly
2. Loop until convergence
   1. Compute gradients
   2. Update weights
3. Return weights

Deep learning: backpropagation



Objective: compute gradient $\frac{\partial \mathrm{J(W)}}{\partial \mathrm{W}}$

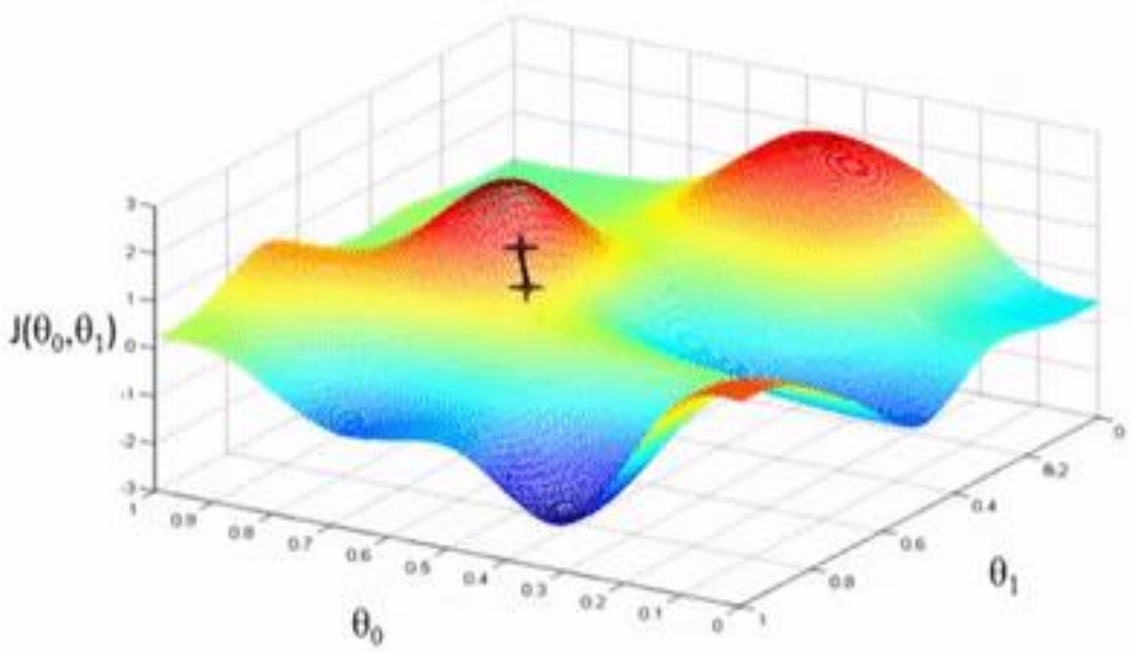
e.g., $\frac{\partial \mathrm{J(W)}}{\partial w_2} = \frac{\partial J(W)}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial w_2}$

$\frac{\partial \mathrm{J(W)}}{\partial w_1} = \frac{\partial J(W)}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial z_1} * \frac{\partial z_1}{\partial w_1}$

Deep learning: gradient descent

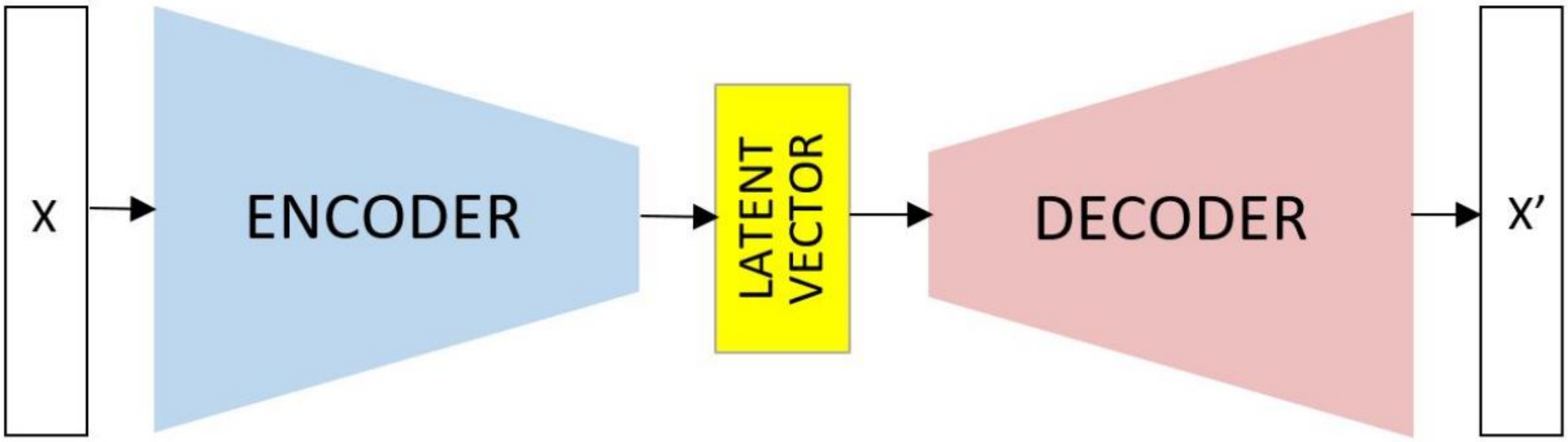
$$W \leftarrow W - \boxed{\eta}\,\frac{\partial J(W)}{\partial W}$$

learning rate



source: machinelearningmastery.com

Application to anomaly detection: AutoEncoders



$$\mathrm{RE} = \sum (\hat{x}_i - x_i)^2$$

Brief state of the art

Cybersecurity + AI = WIN

Easy

Random-Forests-Based Network Intrusion Detection Systems

Publisher: IEEE | Cite This | PDF

Jiong Zhang | Mohammad Zulkernine | Anwar Haque | All Authors

Detection rate (%)	False positive rate (%)
88	2.5

2008

Cybersecurity + AI = WIN

Easy

Random-Forests-Based Network Intrusion Detection Systems

Detection rate (%)	False positive rate (%)
88	2.5

A hybrid network intrusion detection framework based on random forests and weighted k-means

Detection rate (%)	False positive rate (%)
98.5	6

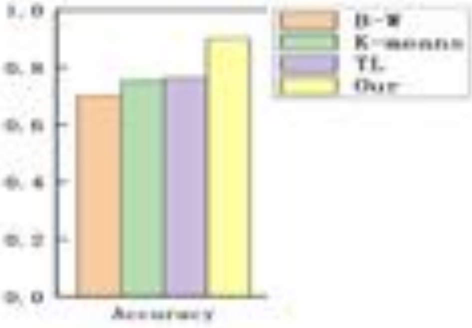
2013

Cybersecurity + AI = WIN

Easy

Multi-Step Attack Detection Based on Pre-Trained Hidden Markov Models

by Xu Zhang [1], Ting Wu [1], Qiuhua Zheng [1,*], Liang Zhai [1], Haizhong Hu [1], Weihao Yin [1], Yingpei Zeng [1] and Chuanhui Cheng [2]

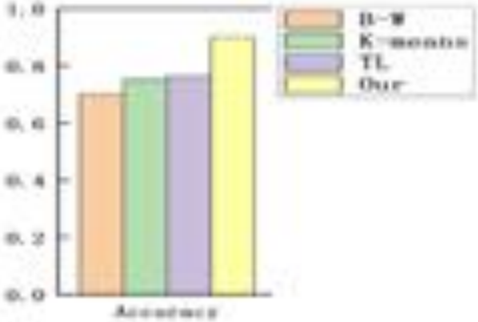


Cybersecurity + AI = WIN

Easy

Multi-Step Attack Detection Based on Pre-Trained Hidden Markov Models

by Xu Zhang [1], Ting Wu [1], Qiuhua Zheng [1,*], Liang Zhai [1], Haizhong Hu [1], Weihao Yin [1], Yingpei Zeng [1] and Chuanhui Cheng [2]



Home > Neural Computing and Applications > Article

A novel approach for APT attack detection based on combined deep learning model

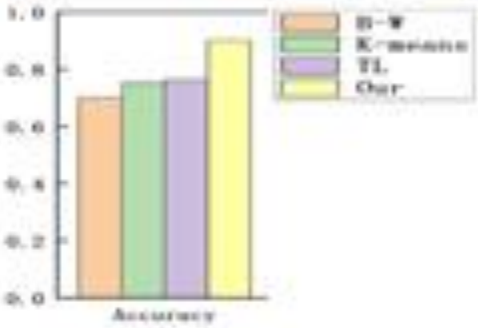
accuracy on all measurements from 93 to 98%.

Cybersecurity + AI = WIN

Easy

Multi-Step Attack Detection Based on Pre-Trained Hidden Markov Models

by Xu Zhang [1], Ting Wu [1], Qiuhua Zheng [1,*], Liang Zhai [1], Haizhong Hu [1], Weihao Yin [1], Yingpei Zeng [1] and Chuanhui Cheng [2]



Home › Neural Computing and Applications › Article

A novel approach for APT attack detection based on combined deep learning model

Original Article | Published: 11 April 2021

accuracy on all measurements from 93 to 98%.

MIF: A multi-step attack scenario reconstruction and attack chains extraction method based on multi-information fusion

CICIDS2017	CTnet	0.99	0.99	0.99	0.99

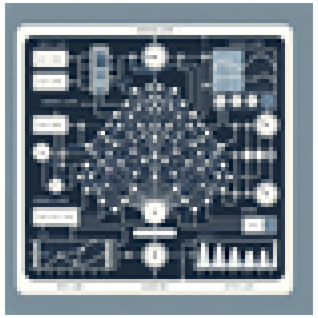
CUSTOCY

But ...

Cybersecurity + AI = WIN

Easy

30 architectures

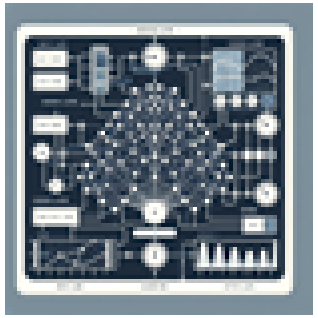


400 000 hyperparameters tested

Cybersecurity + AI = WIN?

Easy

30 architectures



40 000 hyperparameters tested

100 000 flux

4.7% détection
34 234 faux positifs

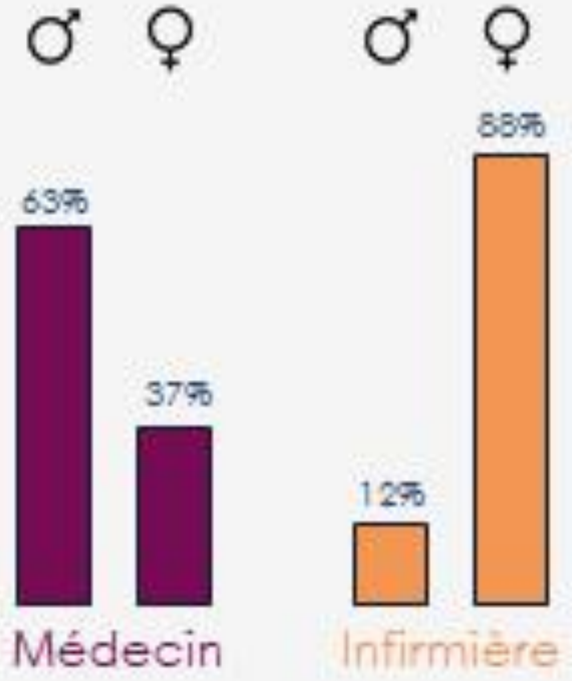
CUSTOCY

Why?

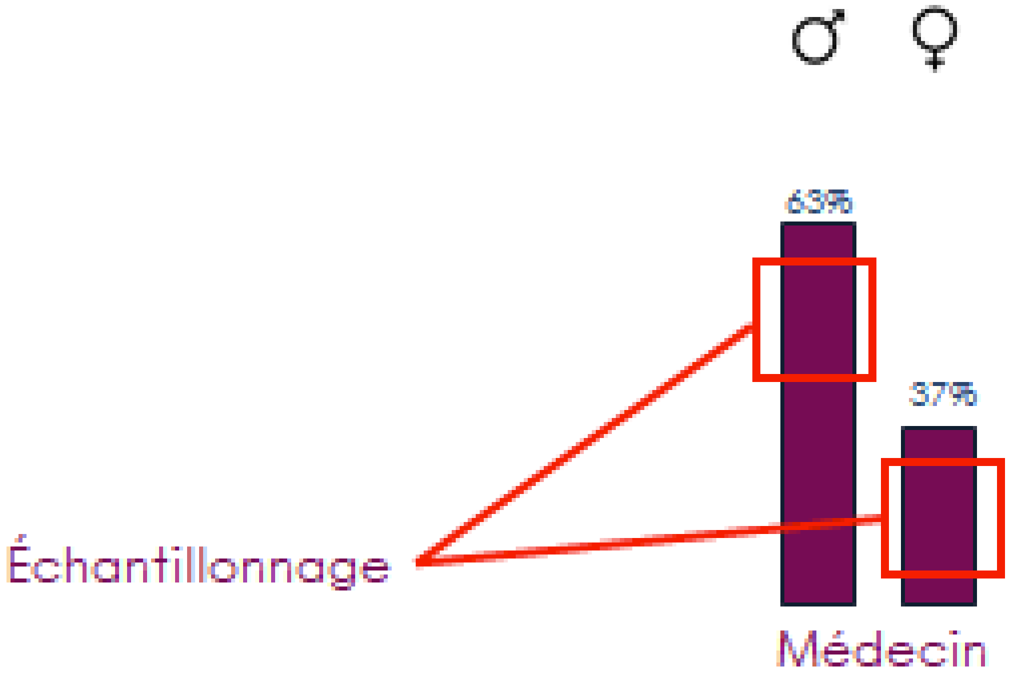
Déséquilibre de classe

Médecin

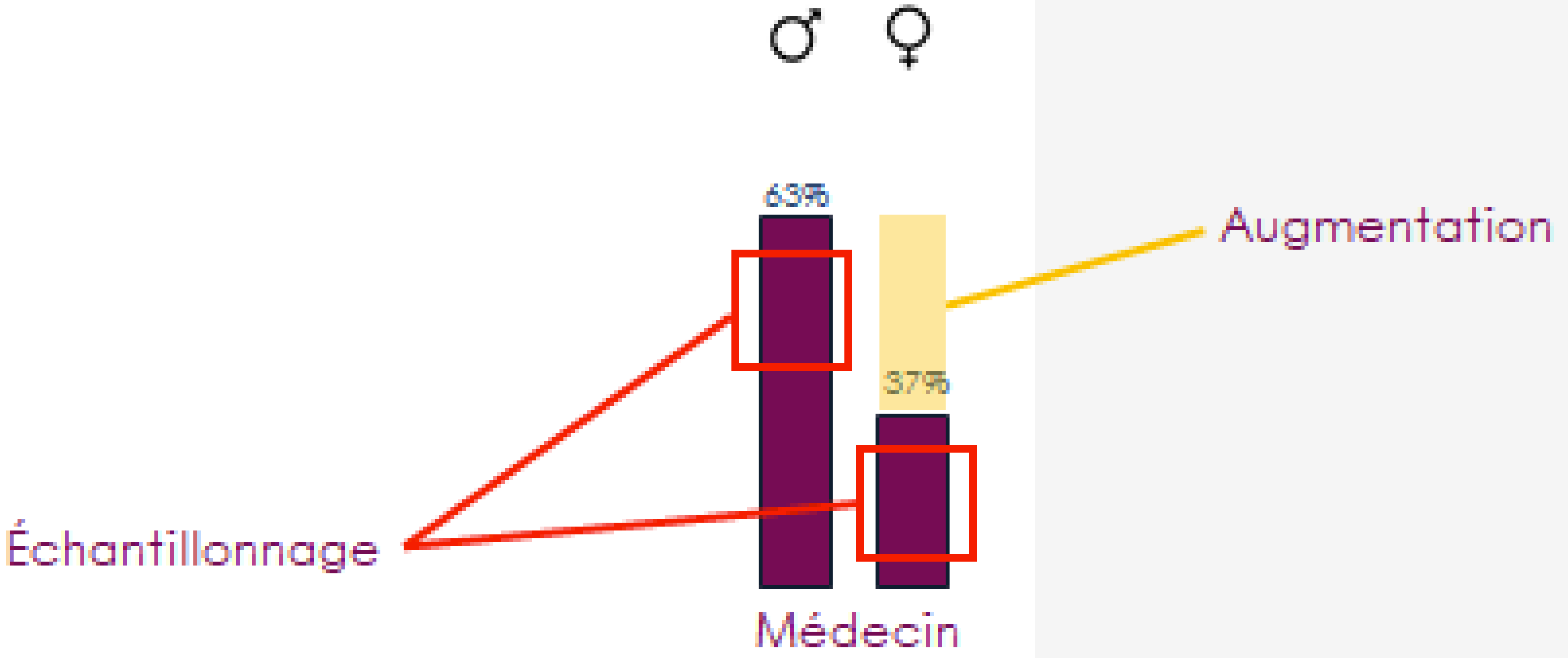
biased	
neighbor	similarity
nurse	1.0121
nanny	0.9035
fiancée	0.8700
maid	0.8674
fiancé	0.8617
mother	0.8612
fiance	0.8611
dentist	0.8569
woman	0.8564



Déséquilibre de classe

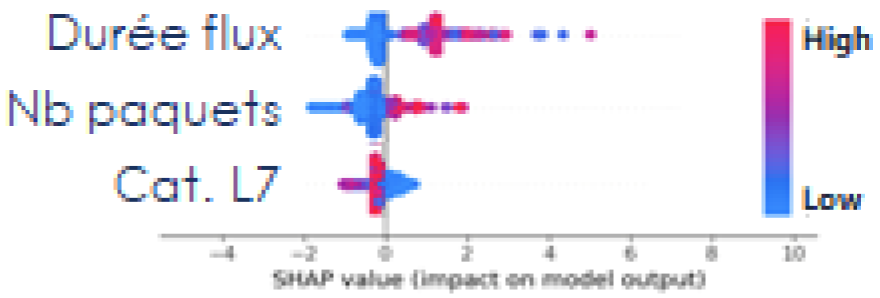


Déséquilibre de classe



Biais inconnus

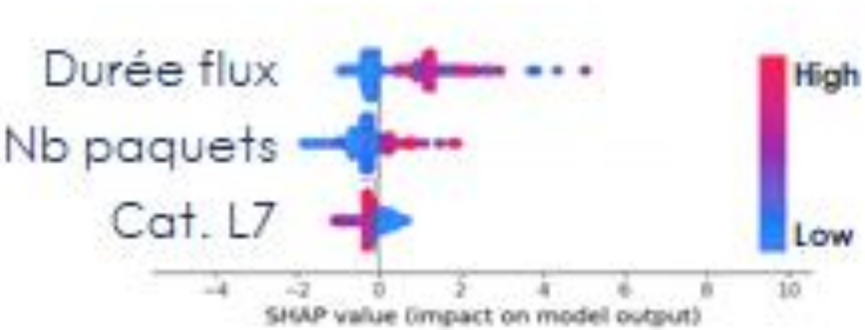
Cas cyber : flux réseau normal classifié en attaque

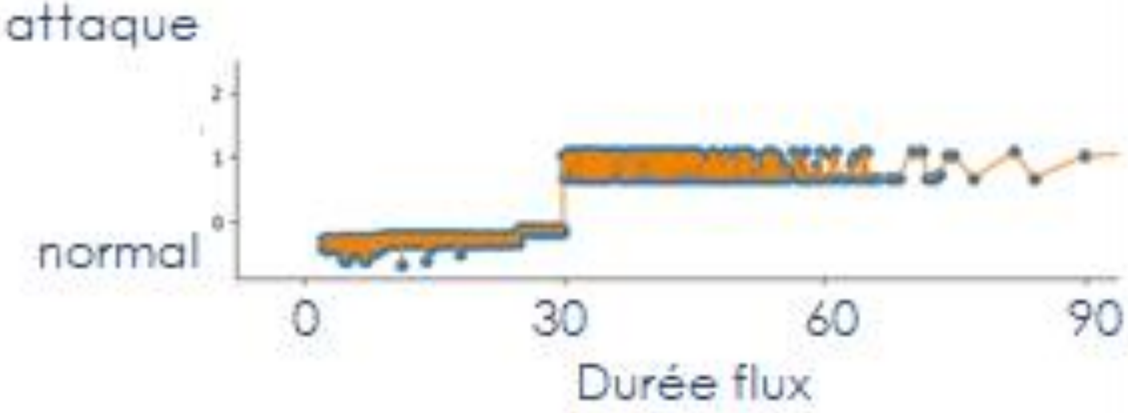


SHAP

Biais inconnus

Cas cyber : flux réseau normal classifié en attaque

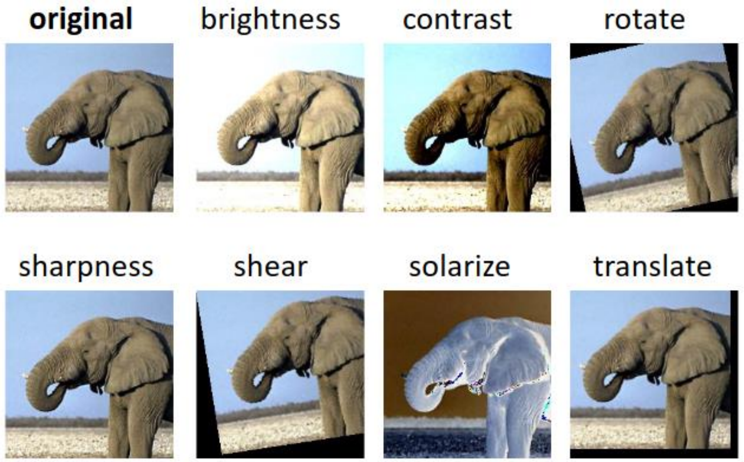




SHAP

PDP

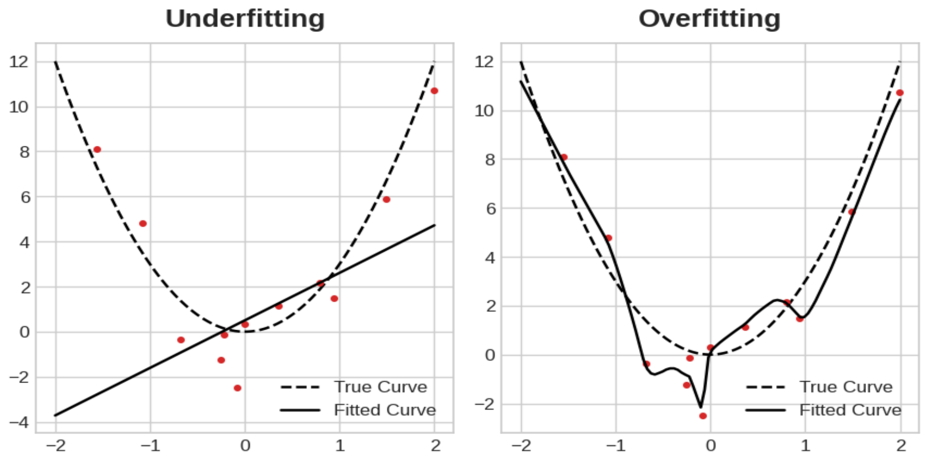
Generalization

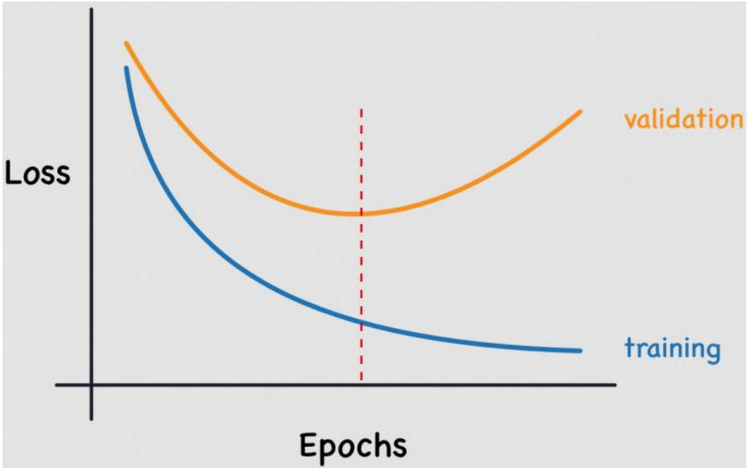


- (Weak) assumption: training and test sets are **independent and identically distributed** (iid)
- Goal: generalize on previously unseen data
- Solutions include **regularization** and **cross-validation**

source: Zhou et al., « Domain Generalization: A survey », IEEE Trans. on Patt. Anal. and ML, 2022

Overfitting





- (Weak) assumption: the more the data fits the model the more reduced loss is
- Goal: improve « signal to noise » ratio
- Solutions include **regularization**, **cross-validation**, **feature selection** or **data augmentation**

source: sourestdeeds.github.io

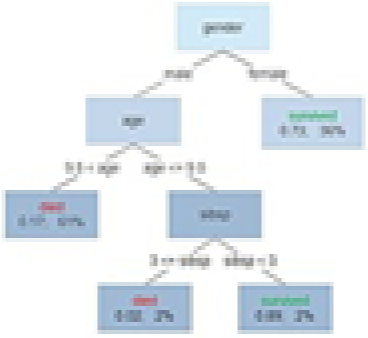
Concept drift

- (Weak) assumption: data distribution is **stationary**
- but not all classes are represented uniformly across the training set
- Well-established features may exhibit gradual drifts (concept changes over time)
- Solutions include:
  - Fine-tuning: to samples exhibiting changes on characteristics prone to change
  - Transfer learning: fit trained models to new unlabeled traces
  - Model extension: structure modification to accommodate new classes

Data Explanation

Explicabilité (& Interprétabilité)

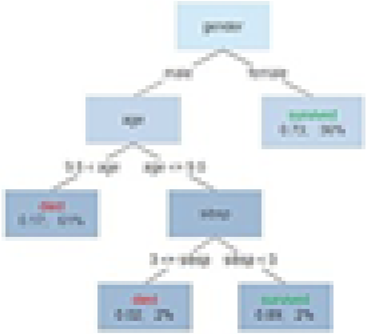
Survie des passagers du Titanic



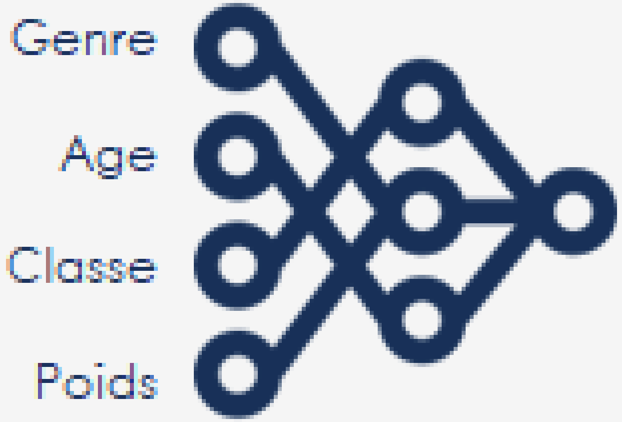
« By design »

Explicabilité

Survie des passagers du Titanic



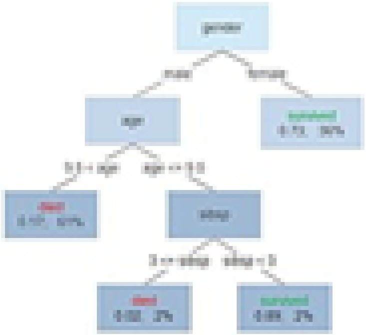
« By design »



Post hoc explainability

Explicabilité

Survie des passagers du Titanic



« By design »

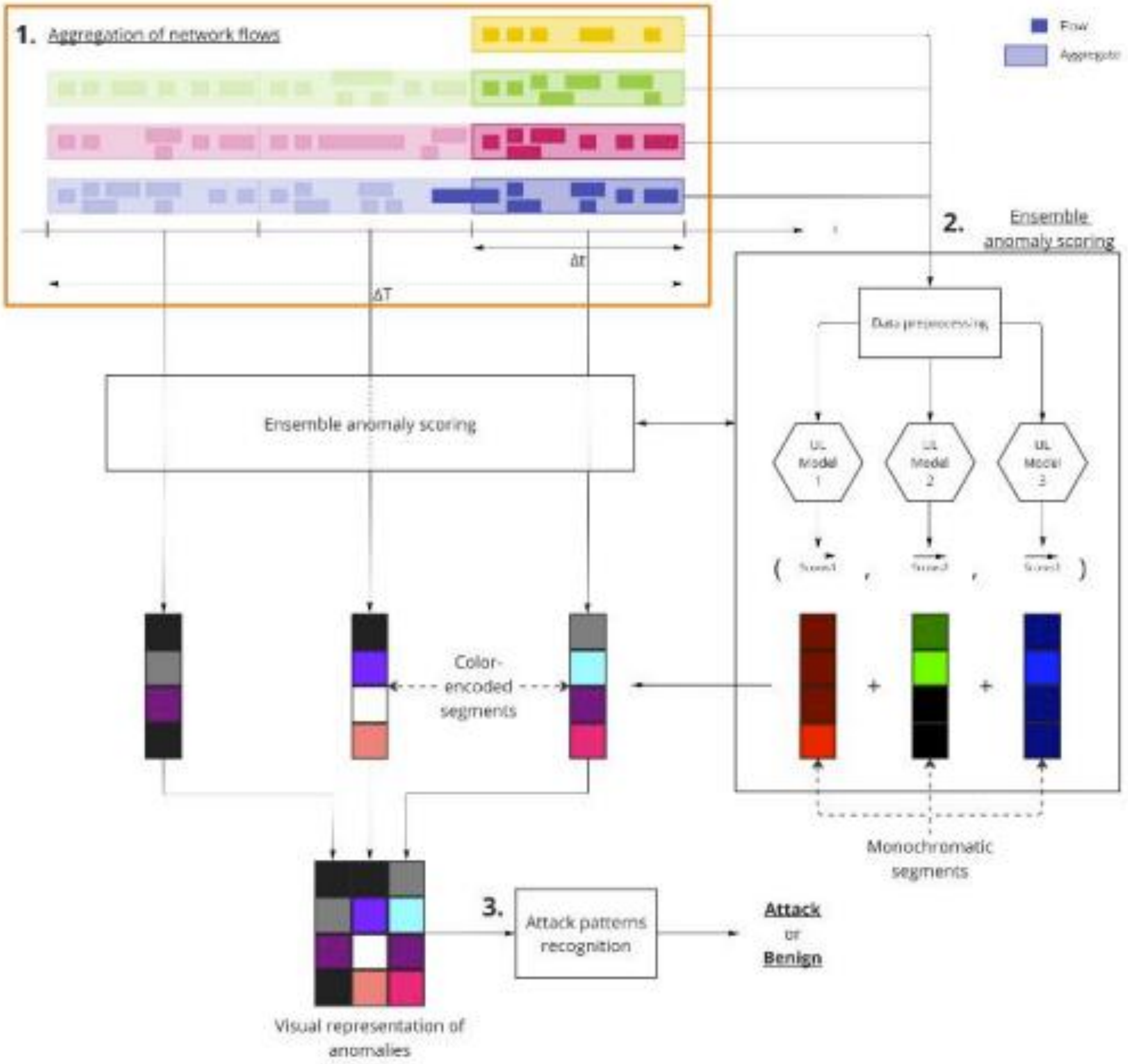
Post hoc explainability

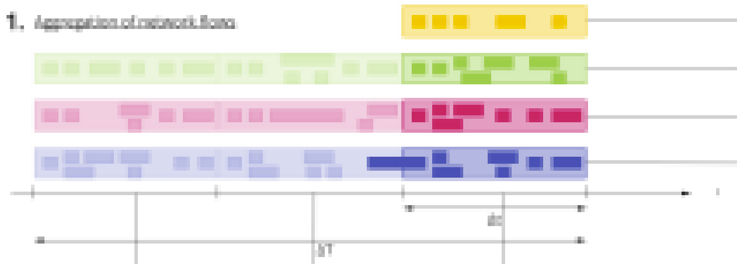
Travaux Céline

Accuracy	99.9%	95%	90%	Expected value
Flow 1	Benign	Attack	Attack	**Benign**
Flow 2	Attack	Attack	Benign	**Attack**
Flow 3	Attack	Attack	Benign	**Attack**
Flow 4	Benign	Benign	Benign	**Benign**
Flow 5	Benign	Benign	Attack	**Attack**

The ensemble learning approach takes advantage of multiple ML models to design more accurate systems :

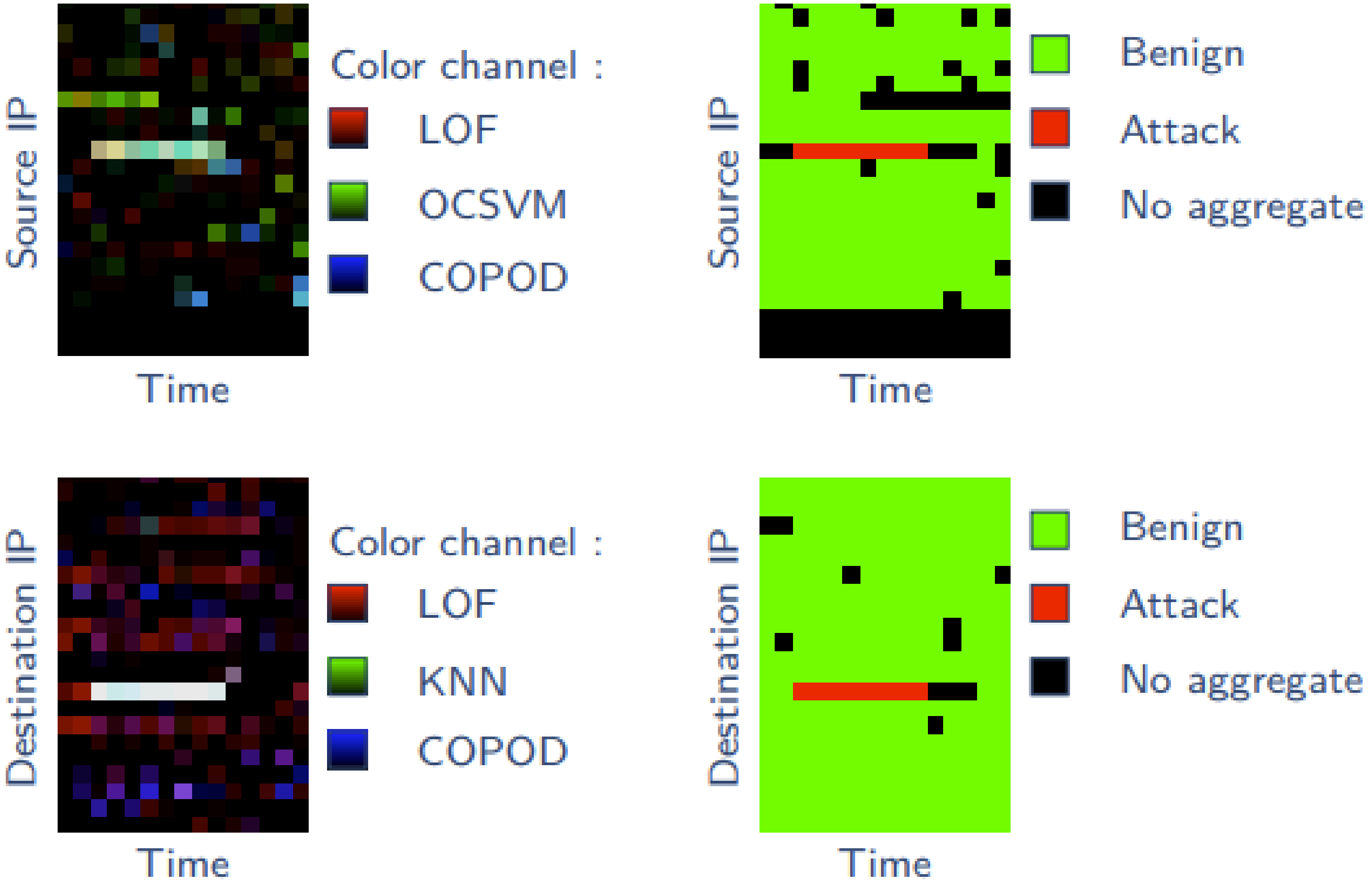
- Bagging
- Boosting
- **Stacking** combines multiple **base learners**.
  - Majority voting
  - Weighted voting depending on the model's performance
  - **Meta-learner**


1. Aggregation of network flows
Flow
ΔT
2. Ensemble anomaly scoring
Data preprocessing
UL Model 1
UL Model 2
UL Model 3
Ensemble anomaly scoring
Color-encoded segments
Monochromatic segments
3. Attack patterns recognition
Attack or Benign
Visual representation of anomalies



Feature	Aggregation key	Description
n_dst_ip	IPsrc	Number of destination IP addresses
n_src_ip	IPdst	Number of source IP addresses
n_dst_ports	IPsrc & IPdst	Number of destination ports
n_src_ports	IPsrc & IPdst	Number of source ports
n_fwd_pkts	IPsrc & IPdst	Number of forward packets
n_bwd_pkts	IPsrc & IPdst	Number of backward packets
sum_flx_dur	IPsrc & IPdst	Sum of flows duration
tot_flx	IPsrc & IPdst	Number of flows
sum_pkts_size	IPsrc & IPdst	Sum of packets size
std_pkt_size	IPsrc & IPdst	Standard deviation of packets size

Representation of network anomalies – Denial of Service (DoS)

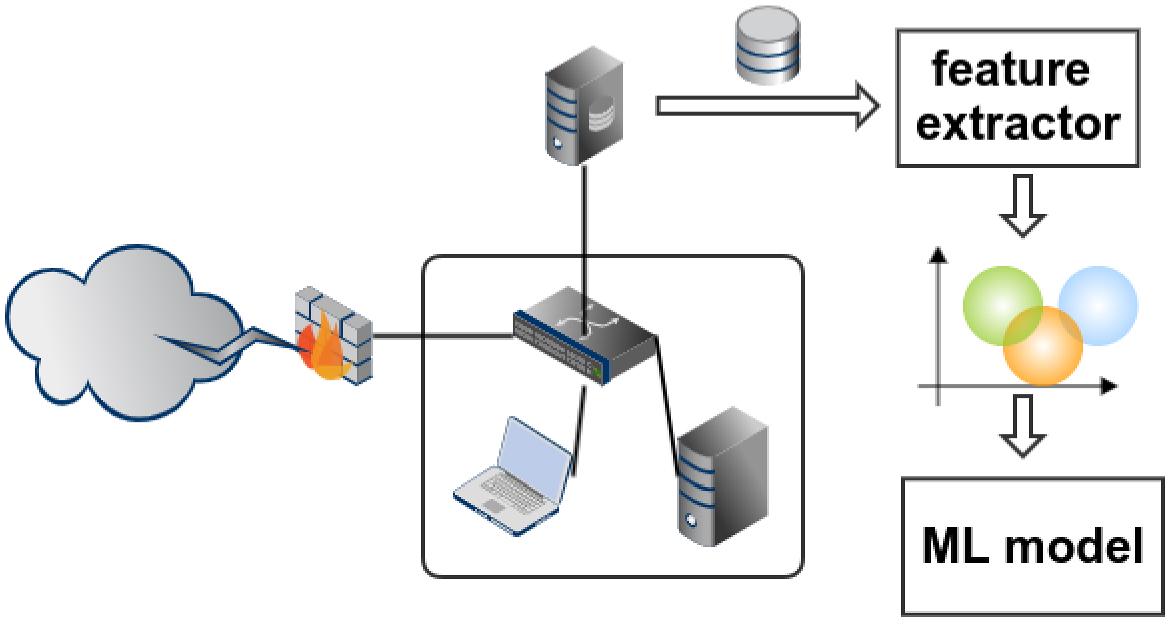


Anomaly scores Labels

Data (or the lack of good traffic data)

Representation

1. Traffic is captured from the data plane as pcap
2. A feature extractor extracts information to represent the traffic in a feature space
   1. Packet-level
   2. Payload-level
   3. Flow-level
3. Representation may be further manipulated
   1. Feature selection
   2. Dimension reduction
   3. Representation learning



NIDS Datasets

Recent study surveyed 89 datasets

- General information
  - year of collection
  - scenario
  - normal and attack traffic types
- Nature of data
  - format
  - number of features
  - anonymized parts of the dataset
- Data volume
  - size
  - duration
- Network properties
  - network type
  - complete capture
- Evaluation
  - split
  - labels

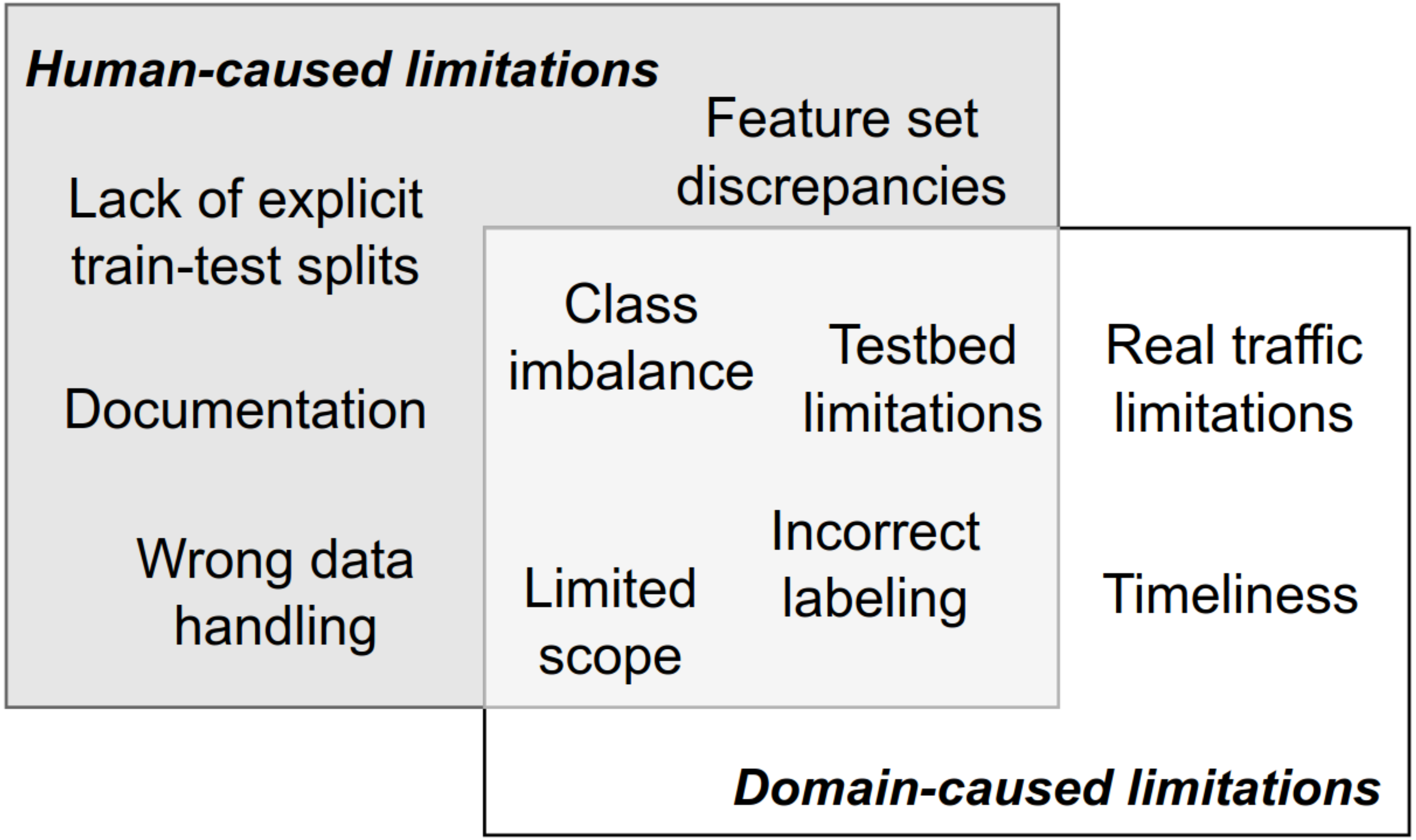
source: Goldschmidt et al., « Network Intrusion Datasets: A Survey, Limitations, and Recommendations », Computers & Security, 2025

Issues with Domain-Specific Properties

- Intra-network variability
  - Computer networks are dynamic and change
- Adversarial environment
  - Attacks attempts to bypass detection
- Inter-network variability
  - Traffic patterns differ among networks
- High cost of errors
  - Unable to balance true and false positives
- Uncertain ground truth & costly labeling
  - Labeling network data is challenging
- Data confidentiality
  - Real-world data might compromise privacy

source: Goldschmidt et al., « Network Intrusion Datasets: A Survey, Limitations, and Recommendations », Computers & Security, 2025

Datasets Limitations



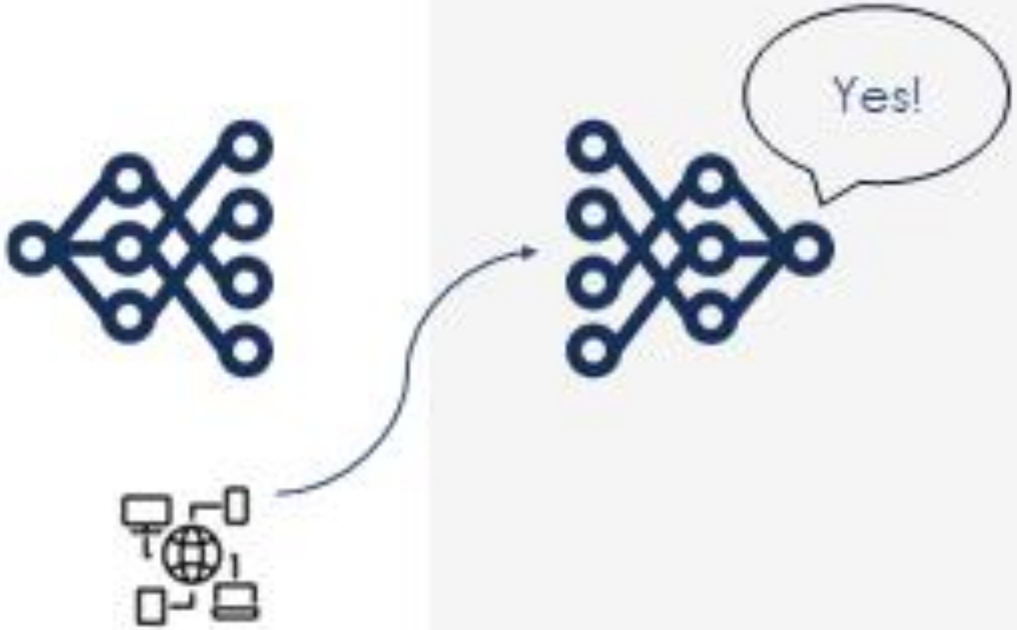
source: Goldschmidt et al., « Network Intrusion Datasets: A Survey, Limitations, and Recommendations », Computers & Security, 2025

IA génératives

générateur

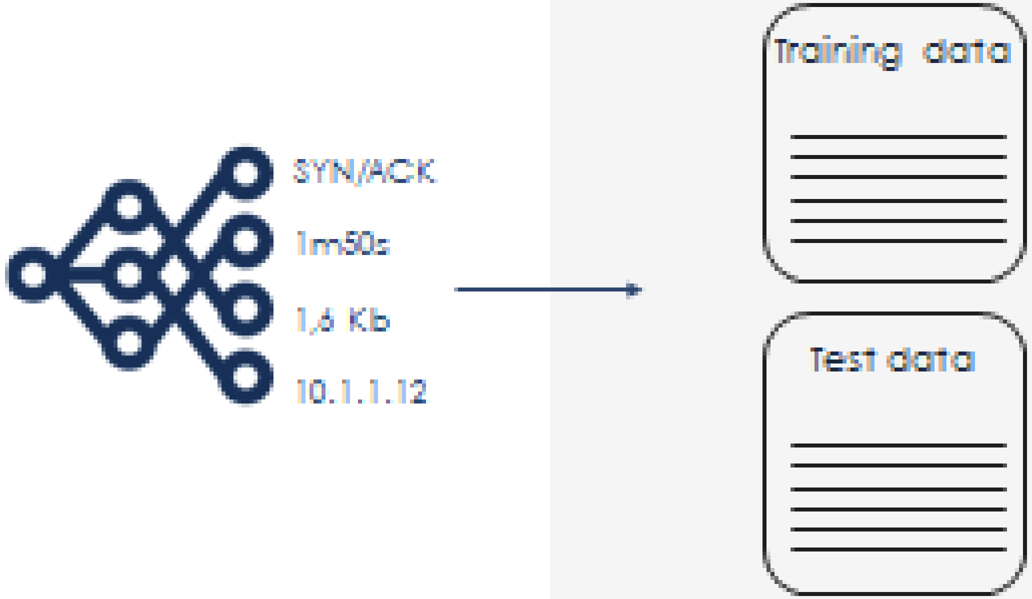
discriminant

IA génératives



IA génératives

IA génératives



Travaux Gabin

Datasets & Flow format

- **TCP** traffic of three malwares ***C&C* channels** (*Emotet, Dridex, Trickbot*) with **benign *HTTPS***.
- **Bi-directional flows**, identified by the ***5-tuple***, with associated **sequence of packets**.
- **Packet** representation **without payload** and a **limited set of features**

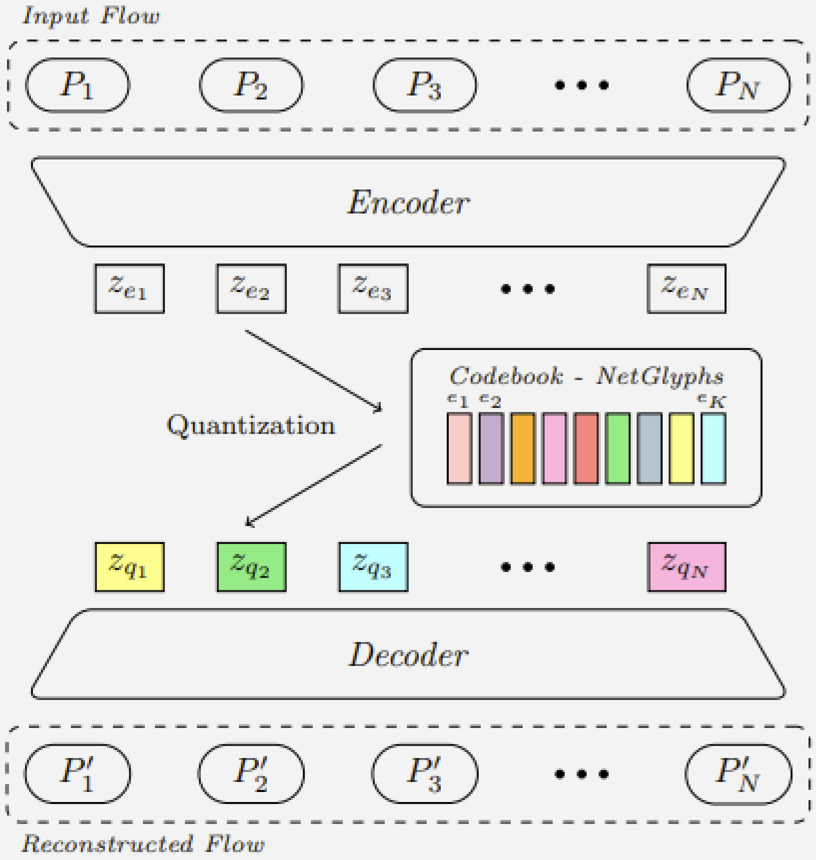
Table 1: Packet features for network traffic representation

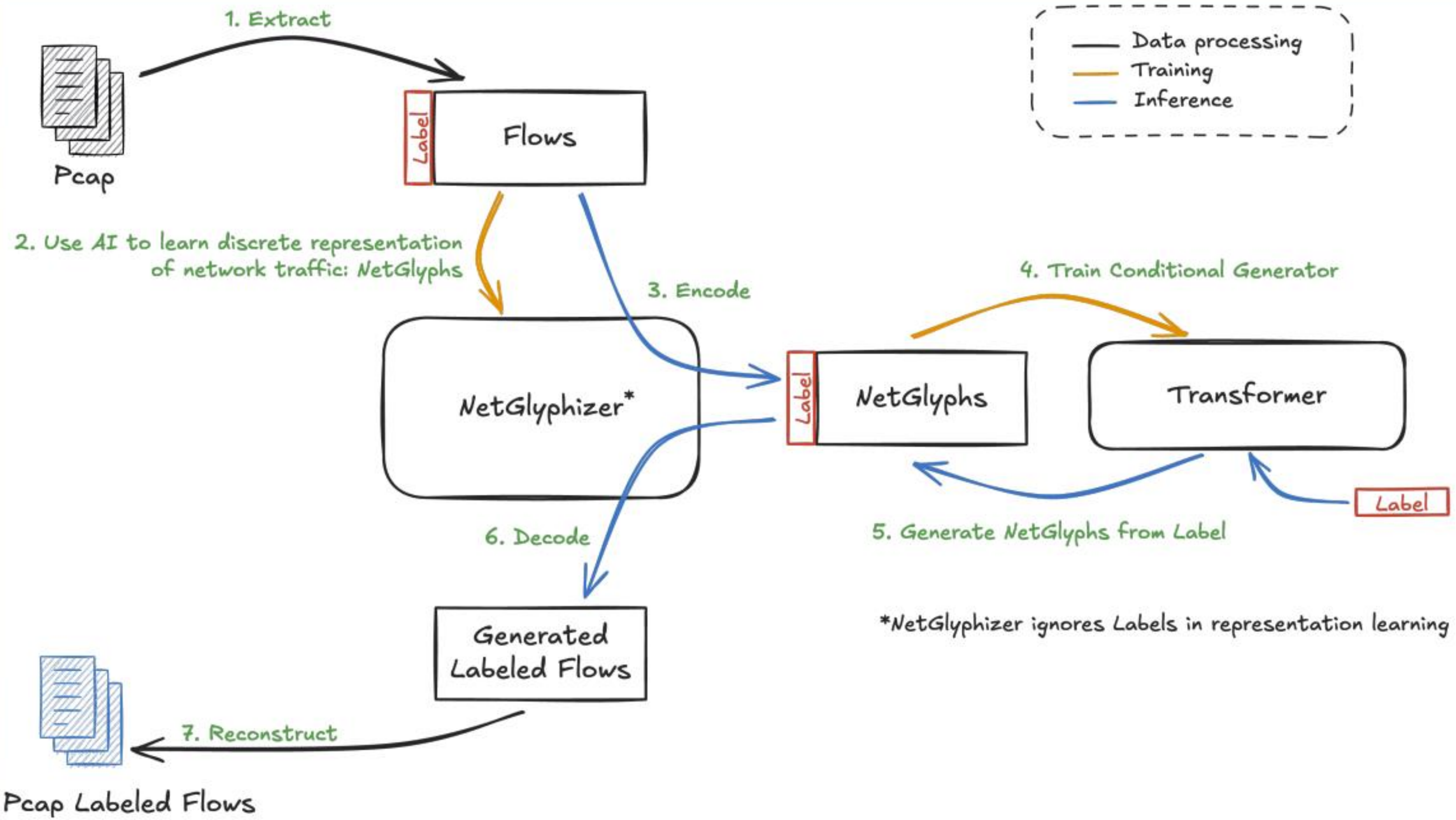
	Type	Example
IAT [1]	*Continuous*	$1.389s$
Payload size	*Numeric*	388
Direction	*Binary*	0 (forward)
Flags	*Categorical*	PA (Psh/Ack)

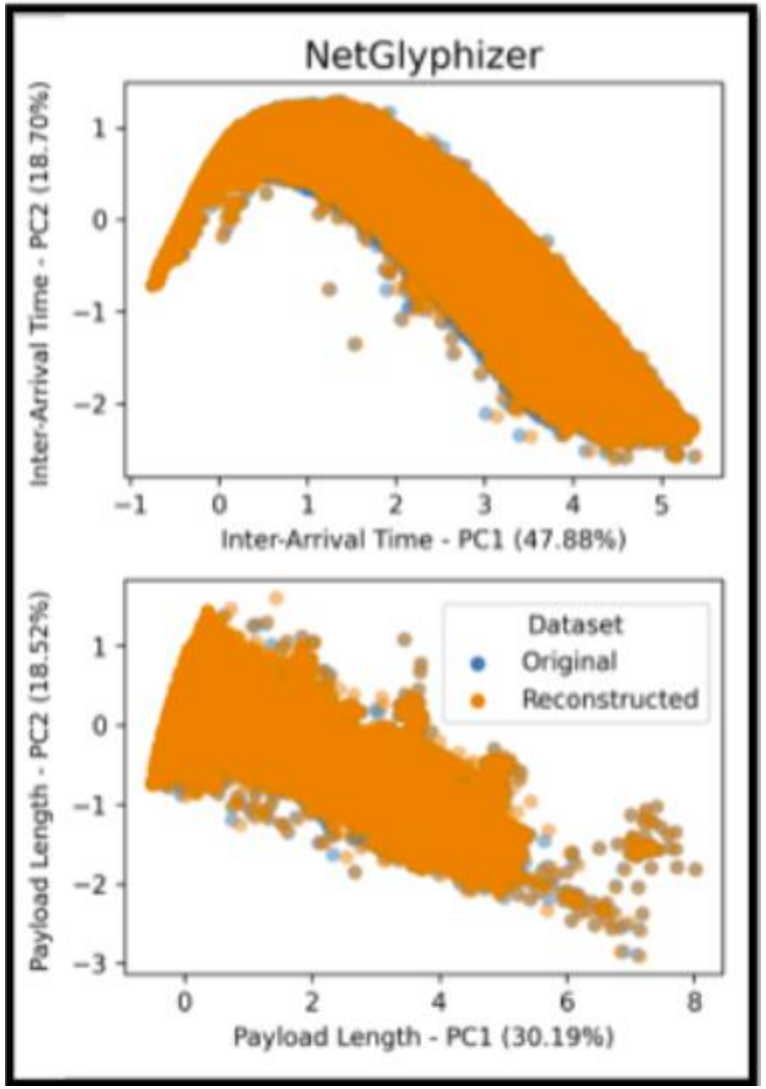
[1] Inter-Arrival Time

NetGlyphizer model

- **Autoencoder** architecture inspired from **VQ-VAE**, adapted to sequence processing. Learns to **discretize network traffic**.




1. Extract
Pcap
Label
Flows
Data processing
Training
Inference
2. Use AI to learn discrete representation of network traffic: NetGlyphs
3. Encode
4. Train Conditional Generator
NetGlyphizer*
Label
NetGlyphs
Transformer
Label
5. Generate NetGlyphs from Label
6. Decode
*NetGlyphizer ignores Labels in representation learning
Generated Labeled Flows
7. Reconstruct
Pcap Labeled Flows

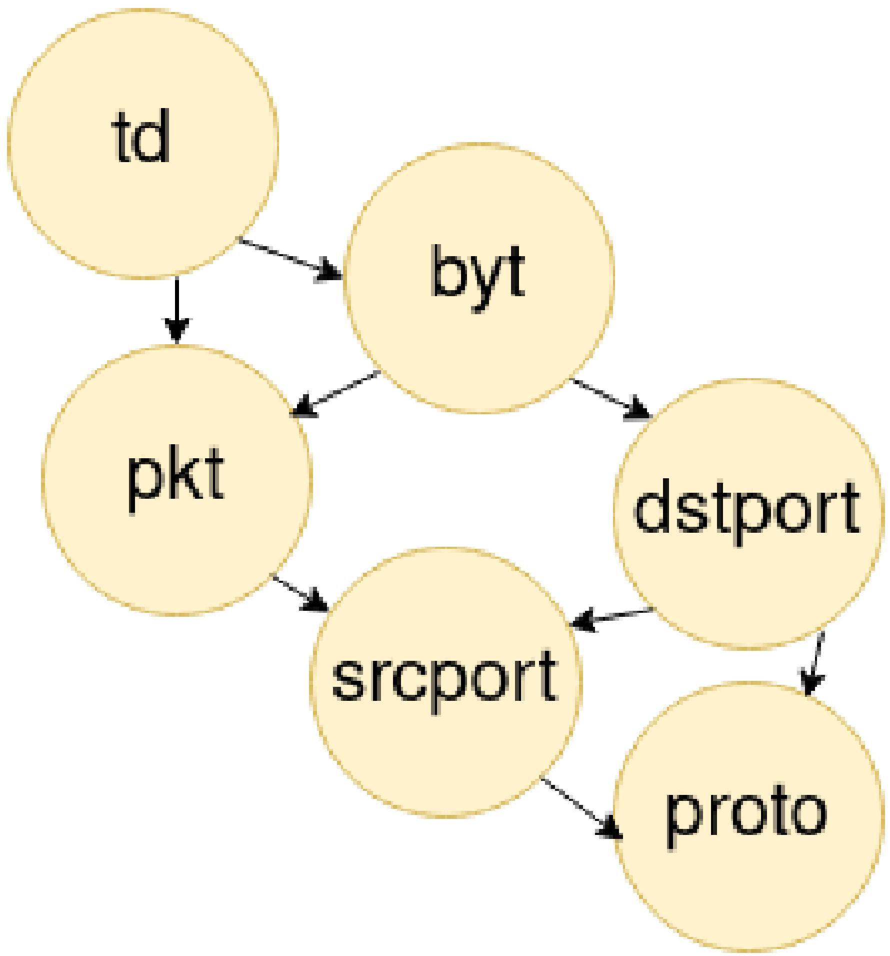


Error Rate

Direction: $< 0,0001\%$
Flags:$<$ 0,001%

Bayesian Network for Traffic Generation

- Focus on legitimate traffic generation: **neglected !**
- Advantages over GANs
  - GANs struggle with feature dependencies and costly computation
  - BNs are efficient, explainable, and handle conditional dependencies
- Learning with BNs: structure learning and Conditional Probability Tables (CPTs)



source: Schoen et al., « A Tale of Two Methods: Unveiling the limitations of GAN and the Rise of Bayesian Networks for Synthetic Network Traffic Generation »,WTMC, 2025

Addressing Challenges inherent to BNs

- Reducing Cardinality of Discrete Features
  - CPT size grows polynomially
    - Group public IPs and ephemeral ports (defined as outside the 30 most commons ports)
- Discretizing Numerical Features
  - BNs require discrete variables
    - Two strategies:
    - Quantile discretization: Equal distribution
    - VGM discretization: Gaussian component-based clustering

source: Schoen et al., « A Tale of Two Methods: Unveiling the limitations of GAN and the Rise of Bayesian Networks for Synthetic Network Traffic Generation »,WTMC, 2025

Synthetic traffic quality evaluation

- Realism: are synthetic flows sampled from the **same distribution** as the source flows?
  - Ex: Contingency Matrix Difference (CMD), Pairwise Conditional Distribution (PCD)
- Diversity: is the synthetic flows' distribution of **similar variance** to the source ones'?
  - Ex: Jensen-Shannon Divergence (JSD), Earth Mover's Distance (EMD)
- Novelty: are synthetic flows **sufficiently different** from source flows?
  - Ex: Membership Disclosure (MD)
- Compliance: do synthetic flows **conform well** to protocol specifications?
  - Ex: Domain Knowledge Check (DKC)

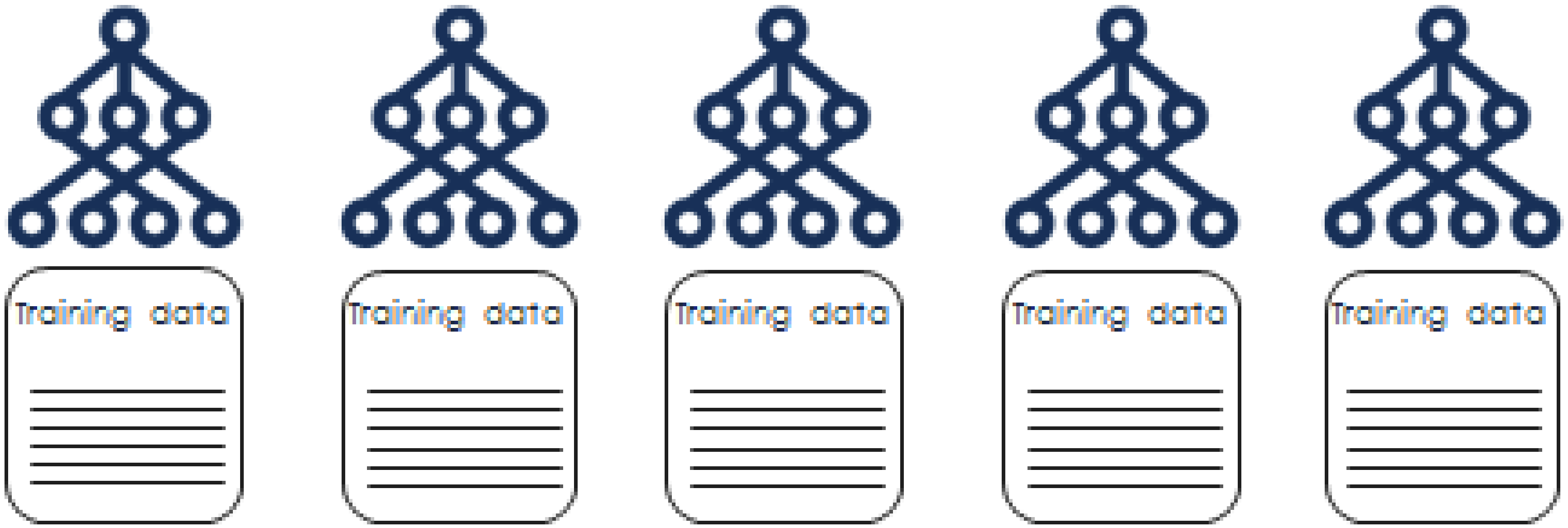
source: Schoen et al., « A Tale of Two Methods: Unveiling the limitations of GAN and the Rise of Bayesian Networks for Synthetic Network Traffic Generation »,WTMC, 2025

Comparison with GAN-based approaches

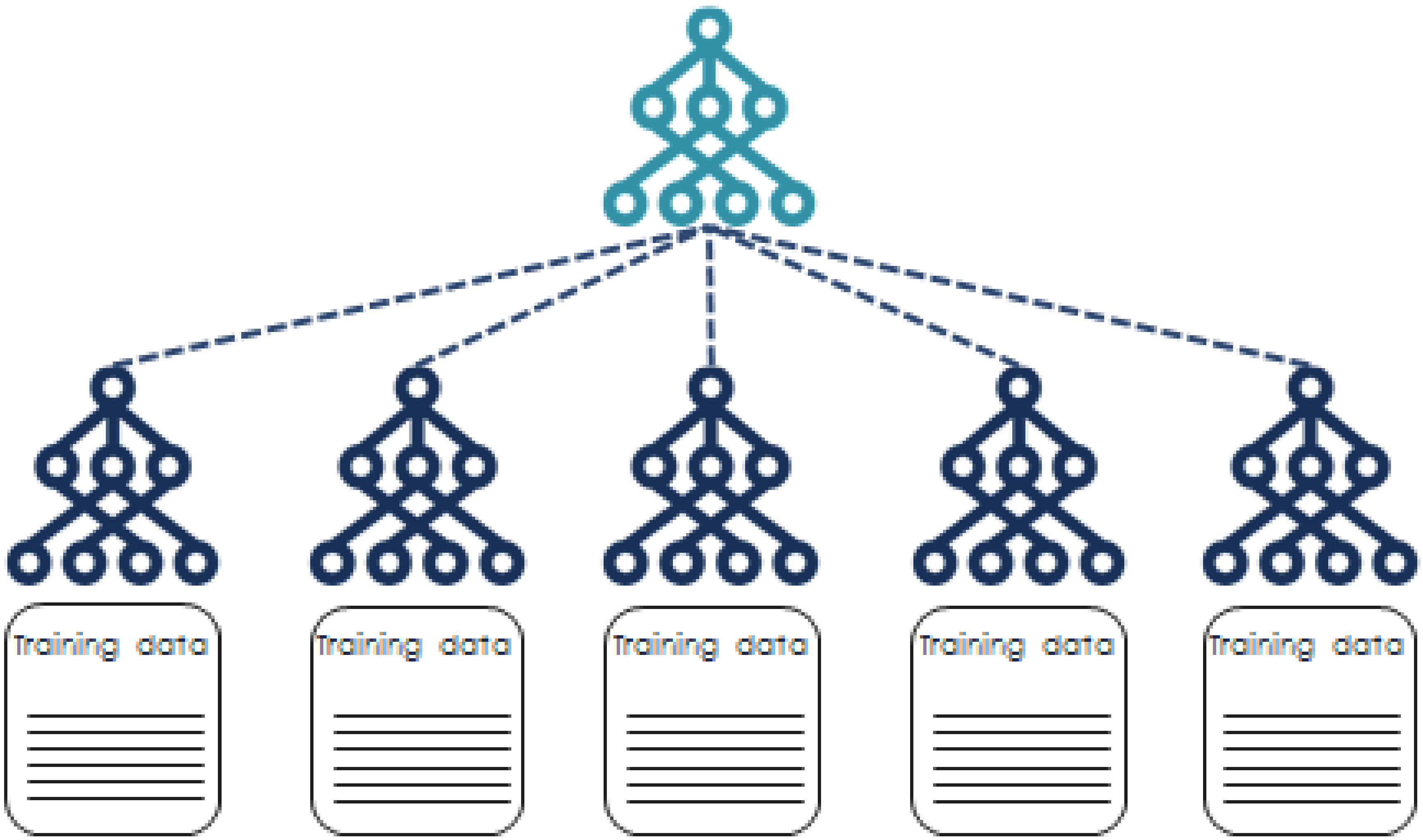
	Description	Real data	Naive	BN_{bins}	BN_{GM}	CTGAN	E-WGAN-GP	NetShare
JSD	Realism and Diversity for categorical features (↓)	**0.067**	0.0068	**0.066**	0.070	0.218	0.105	*0.399*
EMD	Realism and Diversity for numerical features (↓)	**0.002**	**0.002**	0.018	0.007	*0.029*	*0.029*	0.003
CMD	Realism of Correlation between categorical features (↓)	**0.037**	0.223	**0.031**	0.040	0.209	0.050	*0.578*
PCD	Realism of Correlation between numerical features (↓)	**0.373**	*1.222*	**0.452**	0.738	0.863	1.219	0.542
Density	Realism of data distribution (↑)	**0.951**	0.355	0.701	**0.855**	0.486	0.702	*0.027*
Coverage	Diversity of data distribution (↑)	**1.000**	0.805	0.792	**0.998**	0.802	0.996	*0.076*
MD	Novelty (=)	**8.692**	7.519	8.312	8.316	7.447	**8.341**	*5.675*
DKC	Compliance (↓)	**0.006**	0.079	0.005	0.005	0.019	**0.004**	*0.129*

source: Schoen et al., « A Tale of Two Methods: Unveiling the limitations of GAN and the Rise of Bayesian Networks for Synthetic Network Traffic Generation »,WTMC, 2025

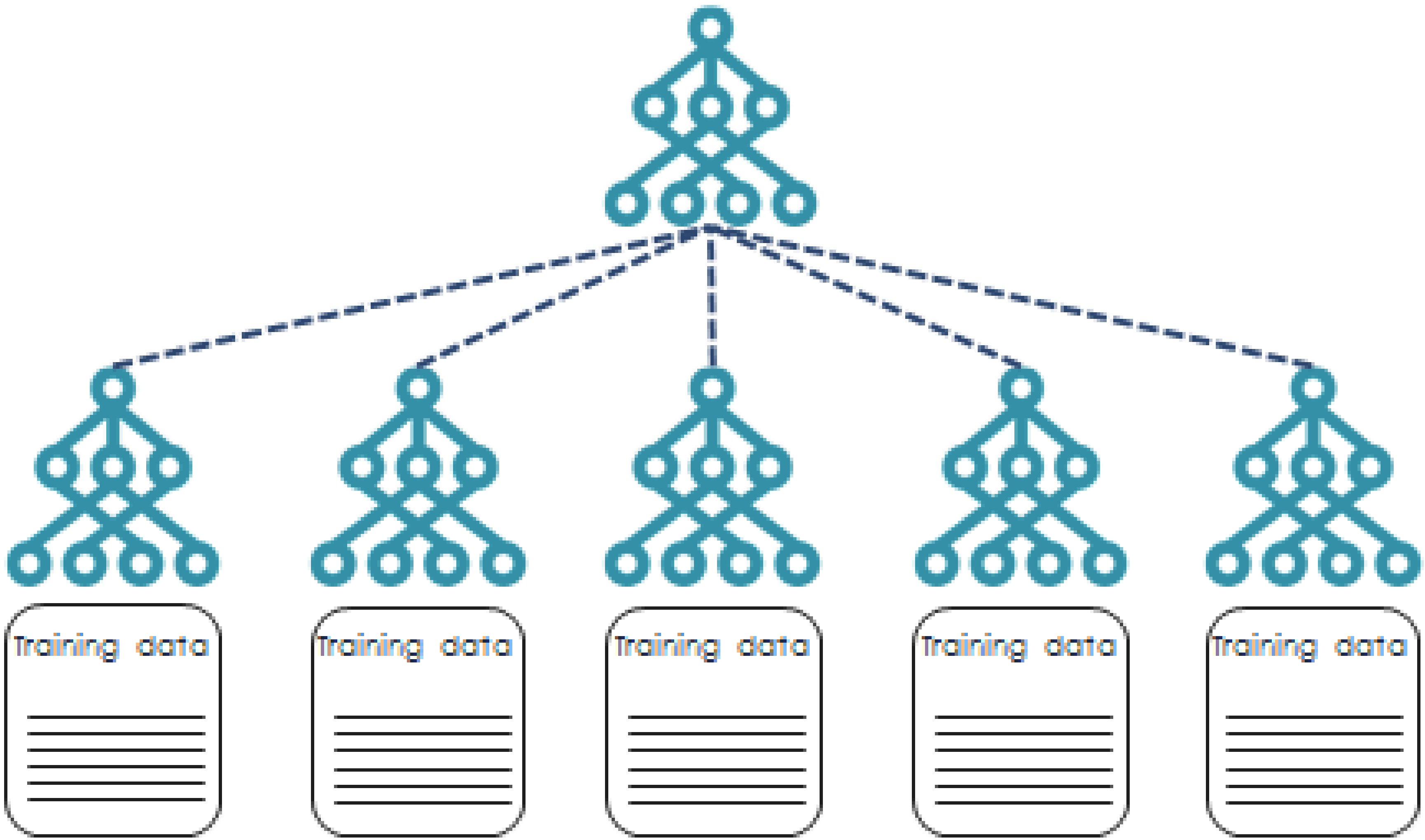
Apprentissage fédéré



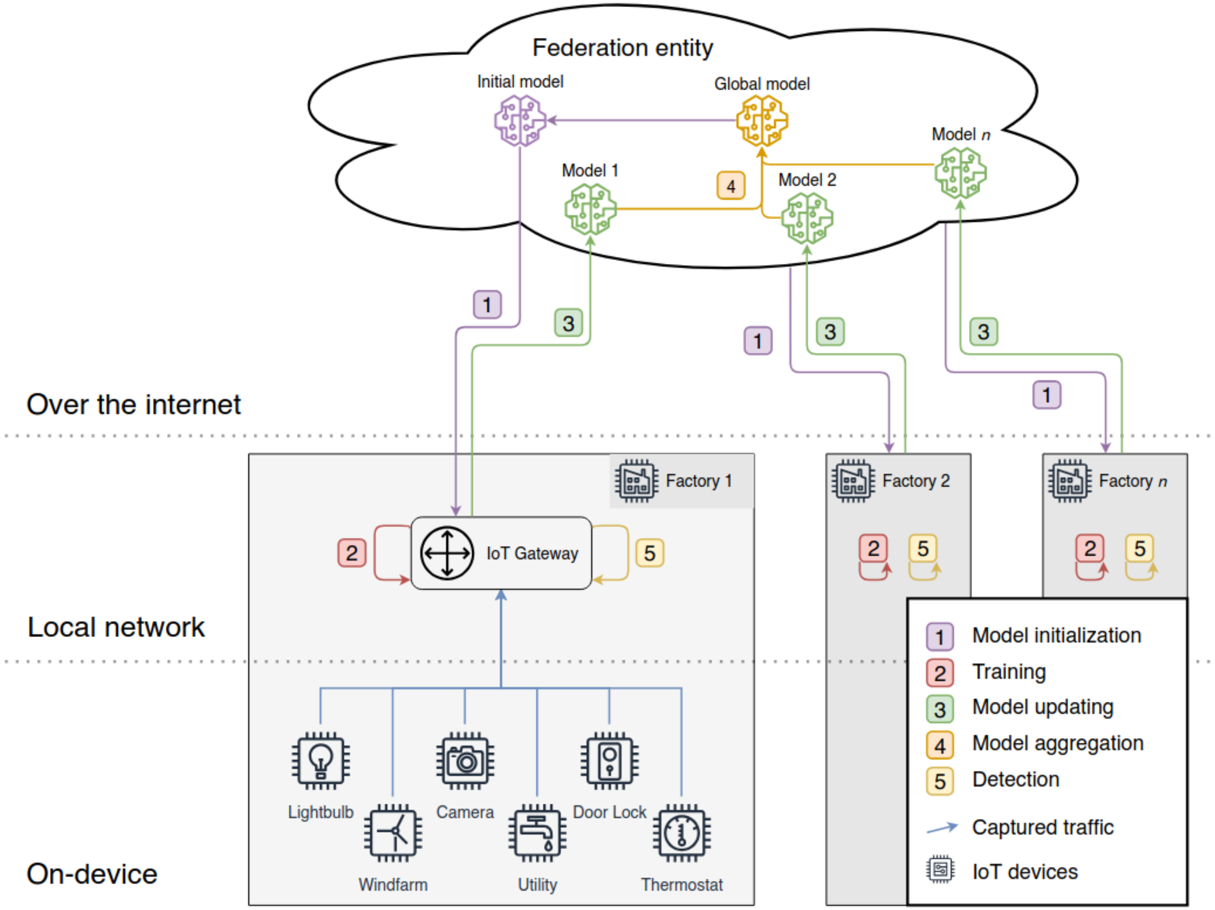
Apprentissage fédéré



Apprentissage fédéré

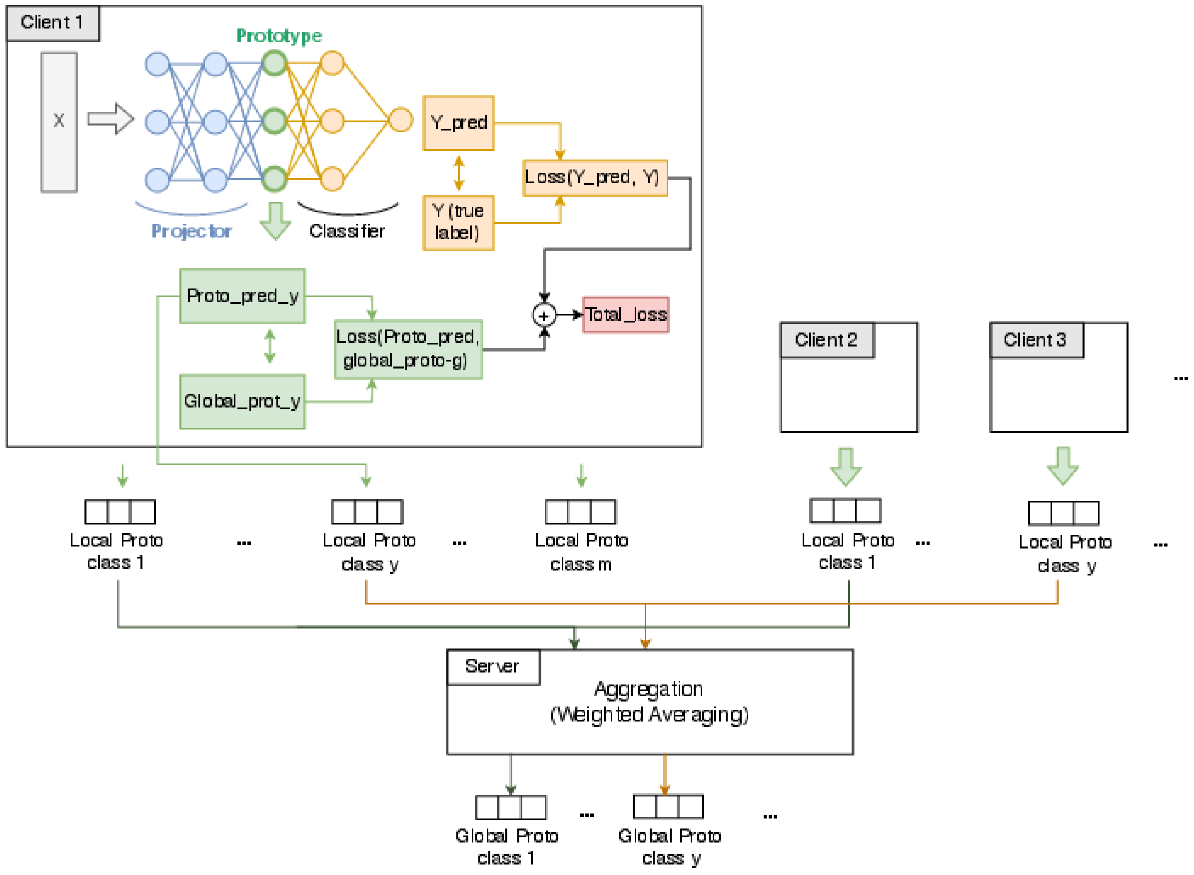


FL-based Intrusion Detection System



source: Lavaur et al., « The evolution of federated learning-based intrusion detection and mitigation: A survey », IEEE Trans. on Net. and Serv. Mgmt, 2022

Collaborative Detection : Knowledge Sharing



source: Chennoufi et al., « PROTEAN: Federated Intrusion Detection in Non-IID Environments through Prototype-Based Knowledge Sharing », ESORICS, 2025

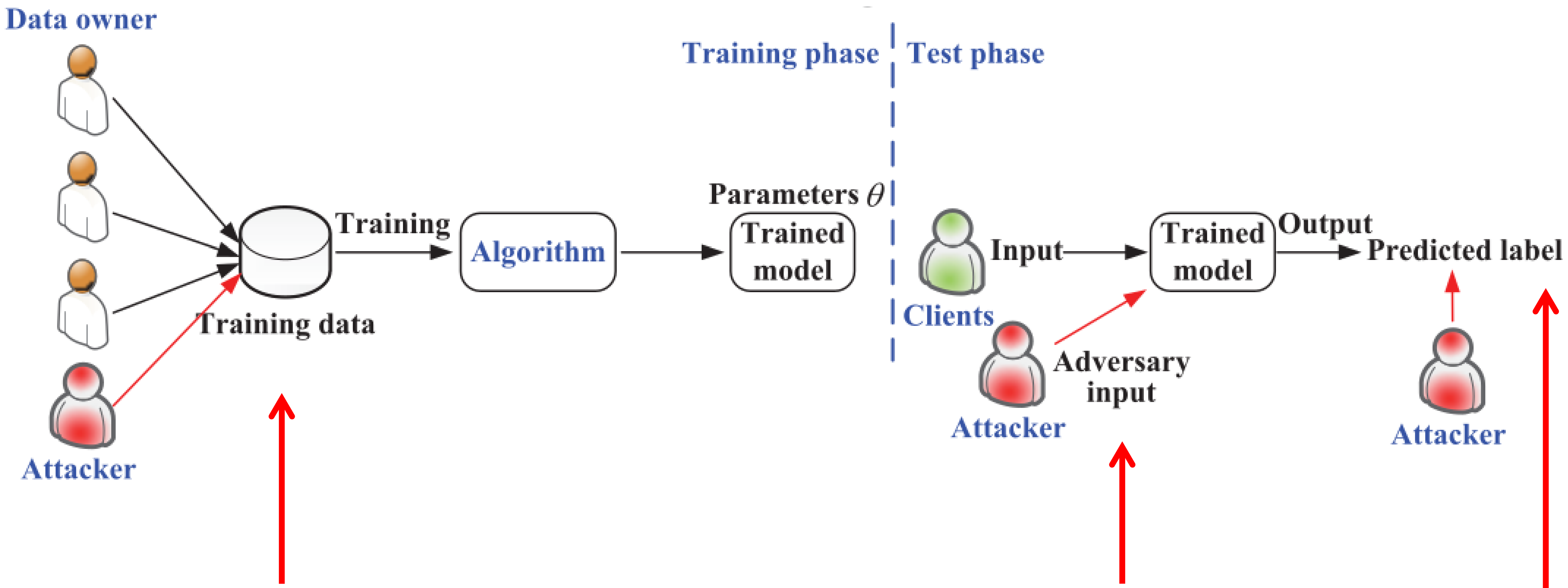
Issues with FL-based IDS

- Knowledge sharing
  - Sharing prototypes improves learning less-represented classes
    - PROTEAN enables zero-shot learning
- Collaboration evaluation
  - Unbalanced data distribution obtained using Dirichlet distribution
- Privacy risk
  - Sharing prototypes does not significantly increase data leakage
- *Byzantine resilience*
  - Label flipping affects classical aggregation algorithms
    - What about FPL/PROTEAN?

source: Chennoufi et al., « PROTEAN: Federated Intrusion Detection in Non-IID Environments through Prototype-Based Knowledge Sharing », ESORICS, 2025

Avoid being detected

Threats against ML Systems



Data poisoning, backdoor

Adversarial examples

Model theft, Membership inference

source: Xue et al., « Machine learning security: Threats, countermeasures, and evaluations », IEEE Access, 2020

Evasion attacks: threat model and problem formulation

- Knowledge restriction
  - White box
  - Grey box
  - Black box
- Attack objective
  - Untargeted
  - Targeted

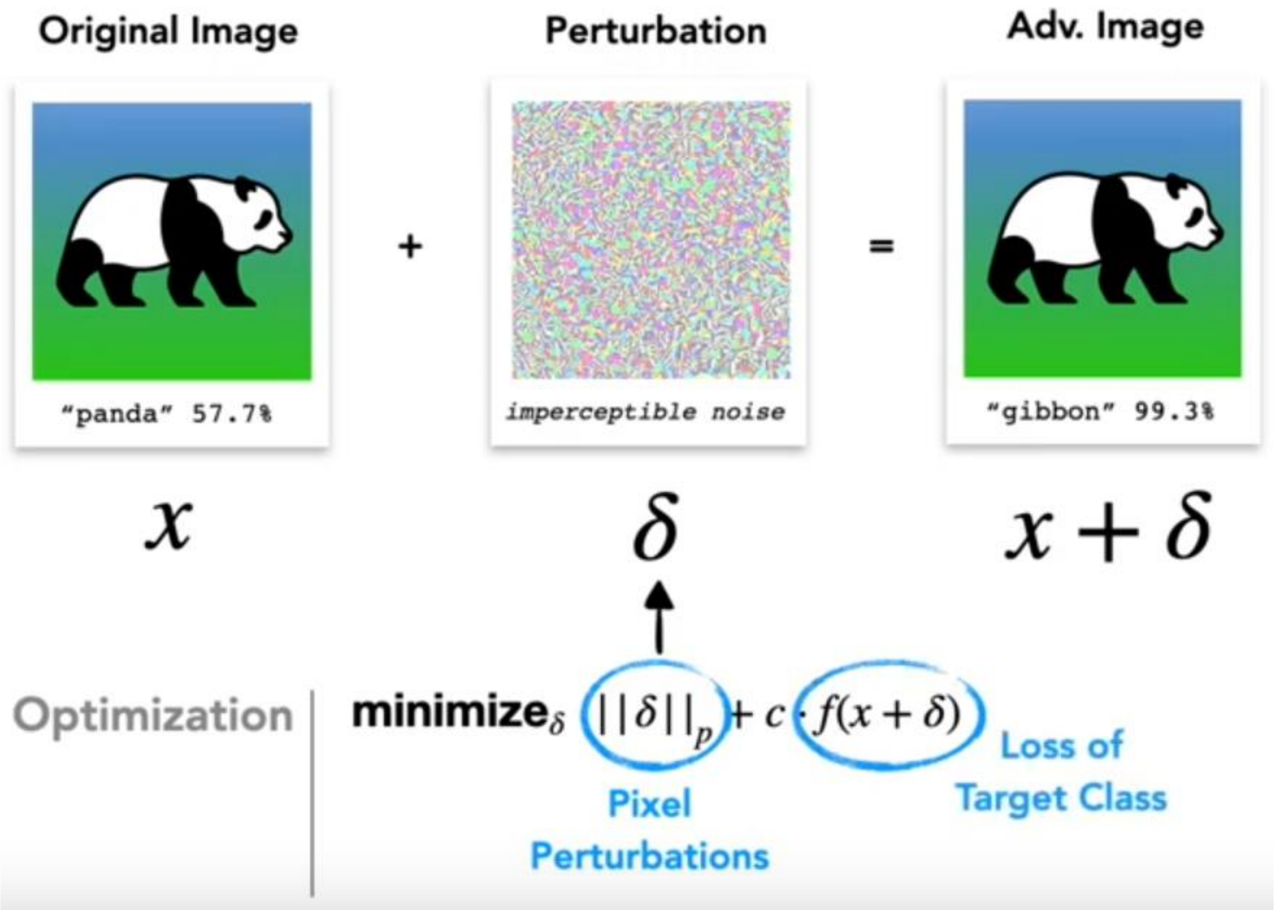
Minimize:

$$\mathrm{D}(\mathrm{x}, \mathrm{x} + \delta)$$

Such that:

- $\mathrm{C}(\mathrm{x} + \delta) = t$ (class constraint)
- $\mathrm{x} + \delta \in [0, 1]^n$ (validity constraint)

Evasion: feature-space attacks



source: Pierazzi et al., « Intriguing properties of adversarial ML attacks in the problem space », IEEE Symposium on Security and Privacy, 2020

Properties of adversarial examples

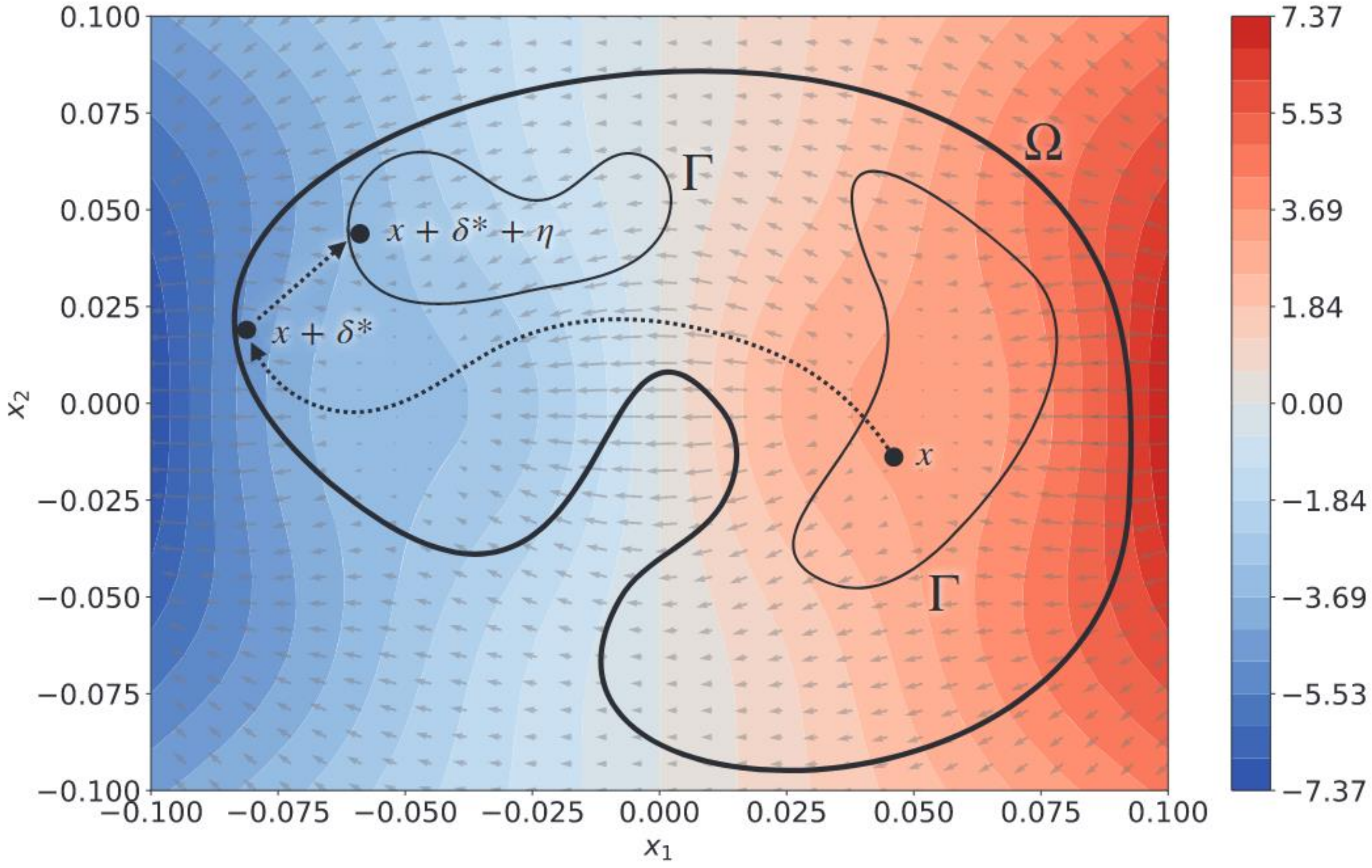
- Perturbations
  - What features, amount of noise, distance from unperturbed sample
- Domain constraints
  - Syntactic constraints (according to specifications, to types, to exclusiveness (e.g., 1-hot encoding))
  - Semantic links, i.e., dependency between features (computed from one or several features, across one or many samples)
- Manipulation space

Are adversarial examples against NIDS practical?

Criterion	Value intervals			Non-binary values			Multiple categories		
Dataset	NSL-KDD	UNSW-NB15	CIDDS-01	NSL-KDD	UNSW-NB15	CIDDS-01	NSL-KDD	UNSW-NB15	CIDDS-01
FGSM	100%	100%	100%	100%	100%	100%	100%	100%	100%
BIM	100%	100%	100%	100%	100%	100%	100%	100%	100%
DeepFool	100%	100%	100%	100%	100%	100%	100%	100%	100%
C&WL_2	99.38%	99.55%	99.01%	100%	99.97%	99.92%	0%	0%	0%
C&WL_∞	73.70%	93.15%	98.97%	75.46%	93.38%	99.82%	28.26%	48.83%	0.22%
C&WL_0	70.27%	32.77%	0.43%	58.01%	15.19%	99.74%	0.24%	0.02%	0.48%
JSMA	0.01%	6.52%	0%	31.93%	68.32%	0.67%	31.02%	68.32%	0.67%

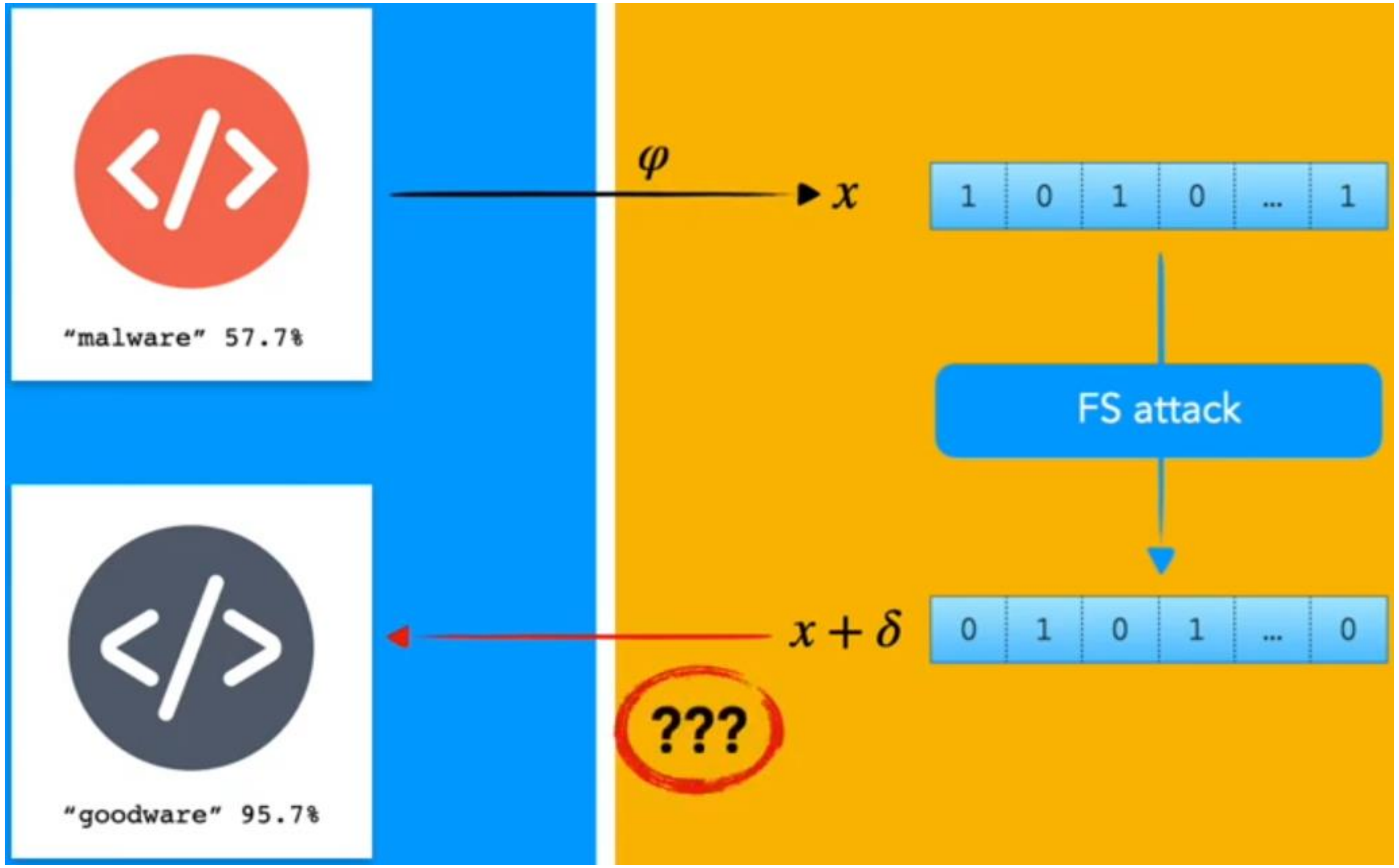
source: Merzouk et al., « Investigating the practicality of adversarial evasion attacks on network intrusion detection », Annals of Telecommunications 77 (11), 2022

Feature space vs. Problem space



source: Pierazzi et al., « Intriguing properties of adversarial ML attacks in the problem space », IEEE Symposium on Security and Privacy, 2020

Inverse feature-mapping problem



source: Pierazzi et al., « Intriguing properties of adversarial ML attacks in the problem space », IEEE Symposium on Security and Privacy, 2020

Problem-space constraints

- Find the sequence of valid transformations **T** such that an object *z* of label *y* is misclassifed as *t* i.e., we want to transform *z* to:

$$z' = T(z)$$

such that $\varphi(z\ ) = x + \delta$ and z' is <u>valid and realistic</u>

  - Available transformations (T): which modifications can be performed in the problem space
  - Preserved semantics (Υ): while mutating z to z' , wrt specific features abstractions which the attacker aims to be resilient against
  - Plausibility (Π): (qualitative) properties must be preserved in mutating z to z', so that z' appears realistic upon manual inspection
  - Robustness to preprocessing (Λ): determines which non-ML techniques could disrupt the attack

Problem-space attack: image domain

- **Threat model:** *perfect knowledge* on a DL-based image (pixels) classifier
- T: modification of pixel values (integer between 0 and 255)
- Υ: constrained perturbation to prevent image from becoming an image from another class
- Π: none explicitly considered (back in 2017)
- Λ: constrained perturbation to prevent changes from being *perceptible to a human*
- **Search strategy:** gradient-driven with no side effects

source: Carlini et Wagner, « Towards evaluating the robustness of neural networks », S&P, 2017

Problem-space attack: code domain

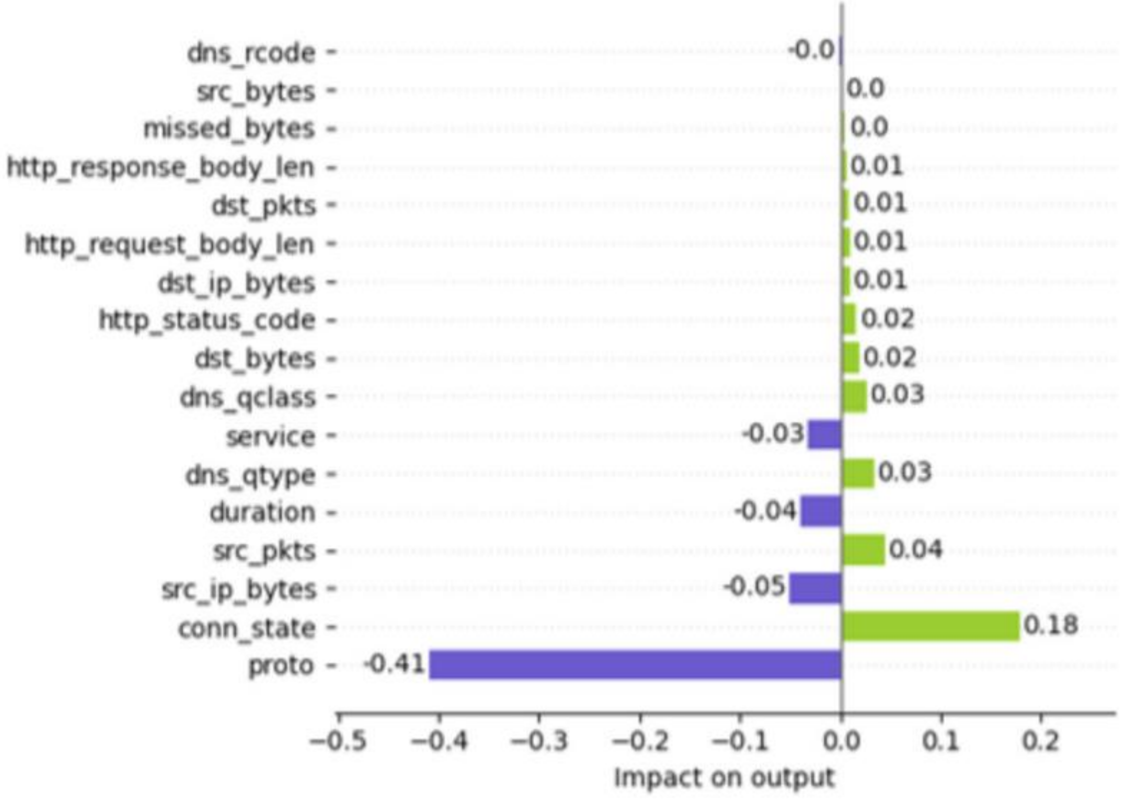
- **Threat model:** *zero knowledge* on any static analysis features (AST, PDG, CFG) classifier
- $\mathcal{T}$: transplantation of semantically-equivalent benign ASTs
- Υ: preservation of malicious semantics by construction (AST-based transplantation)
- Π: robust to removal of function/variable name inconsistencies
- Λ: by construction if no obsolete objects are used
- **Search strategy:** problem-driven (search of sub-AST graphs in benigh samples); side effects are incurred

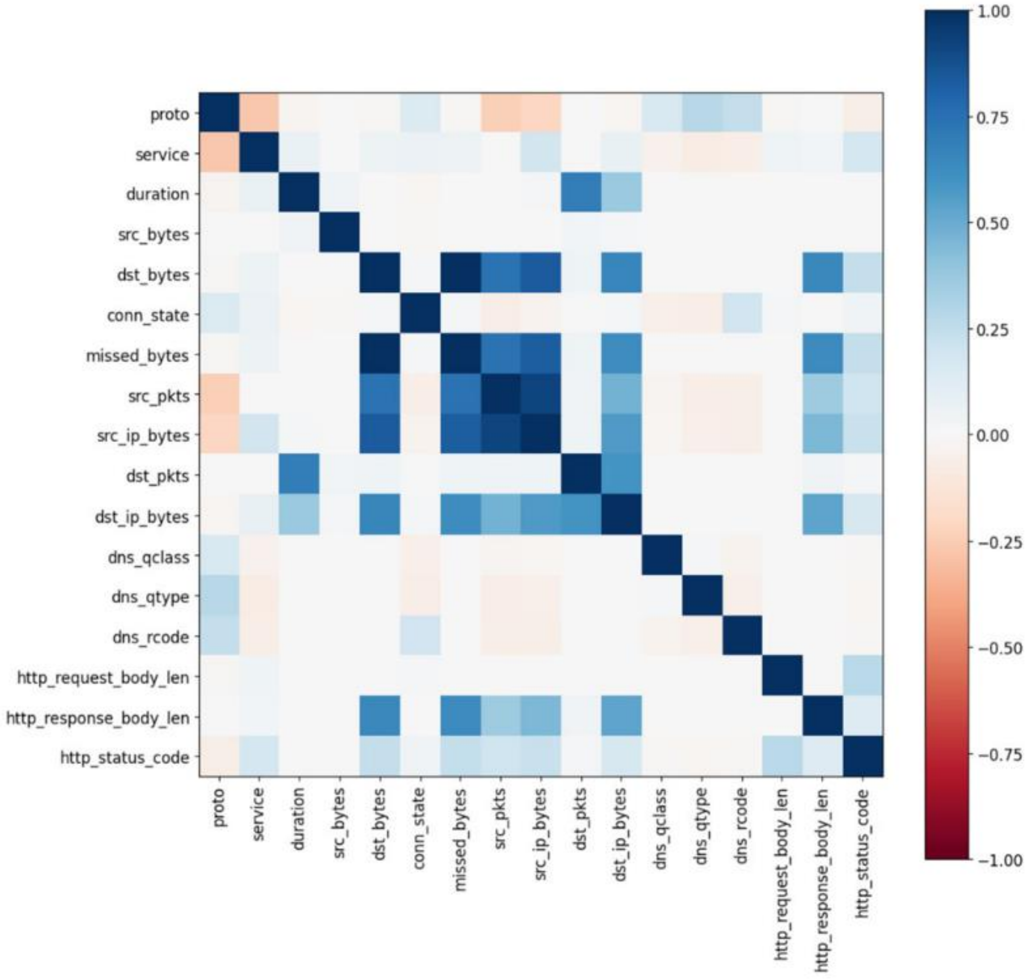
source: Fass et al., « HideNoSeek: Camouflaging Malicious JavaScript in Benign ASTs », ACM CCS, 2019

XAI-driven Black-box Attack

1. Analyze the target’s model decisions, in part. negatives, with KernelSHAP
2. Select k most important features, which are problem-space compliant
3. Plot true positives and negatives in the k-dimensional space
4. Validate features after computing a correlation heatmap
5. Chose candidate features to perturb
6. Implement perturbations in the problem space

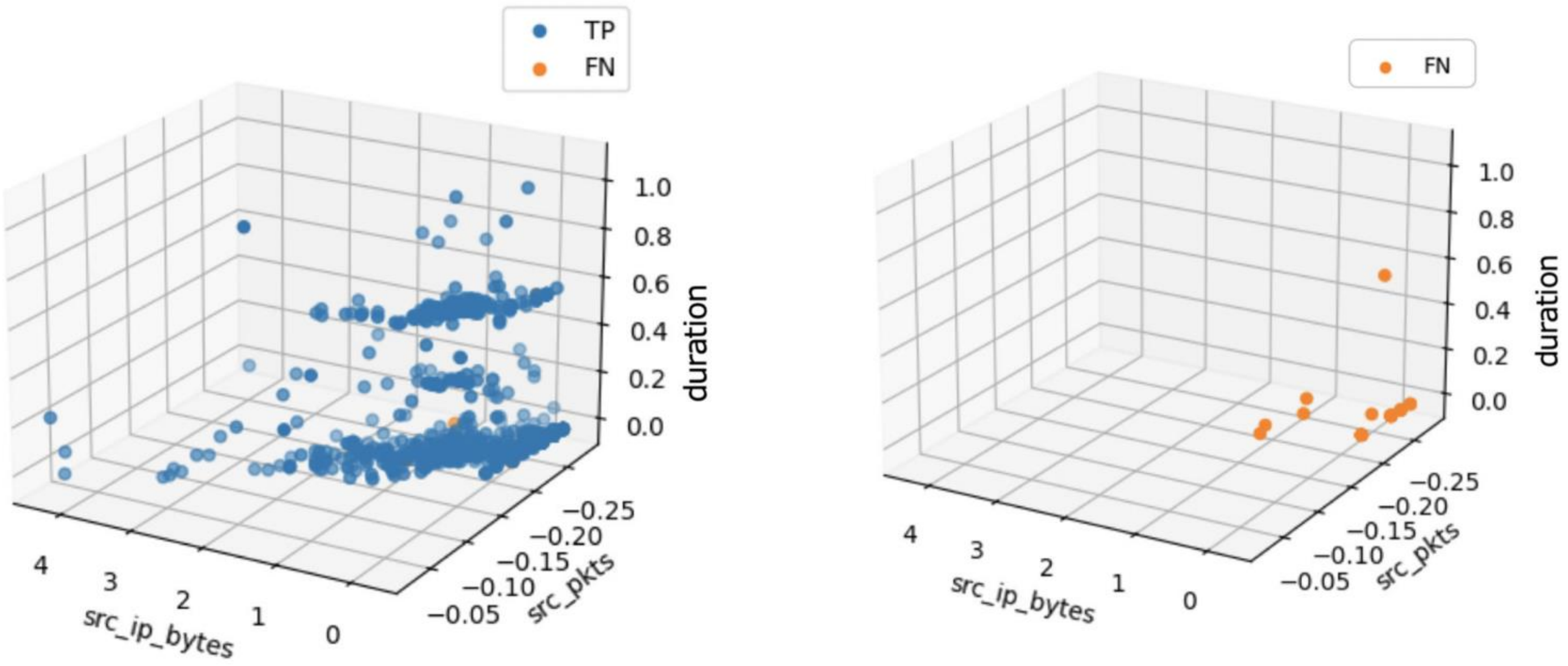
XAI-driven Attack Use Case: XSS





source: Okada et al., « XAI-driven black-box adversarial attacks on network intrusion detectors », Intl Journal of Inf. Sec., 2025

XAI-driven Attack Use Case: XSS



source: Okada et al., « XAI-driven black-box adversarial attacks on network intrusion detectors », Intl Journal of Inf. Sec., 2025

Evasion defenses

- **Adversarial training:** include adversarial examples in the training set
- **Obfuscated gradients:** disrupt gradient-descent by masking
- **Defensive distillation:** transfer knowledge to a new NN which is trained with probability vectors as output instead of class labels
- **Feature squeezing:** reduce dimensionality by filtering unnecessary features
- **Feature removal:** remove most vulnerable features
- **Adversarial detection:** estimate density estimations (for example, on the last layer) compared to the training set of a class (e.g., benign)
- **Adversarial query detection:** detect the similarity among a group of queries

source: Debicha et al., « Review on the feasibility of adversarial evasion attacks and defenses for network intrusion detection systems », arXiv, 2023

Practical

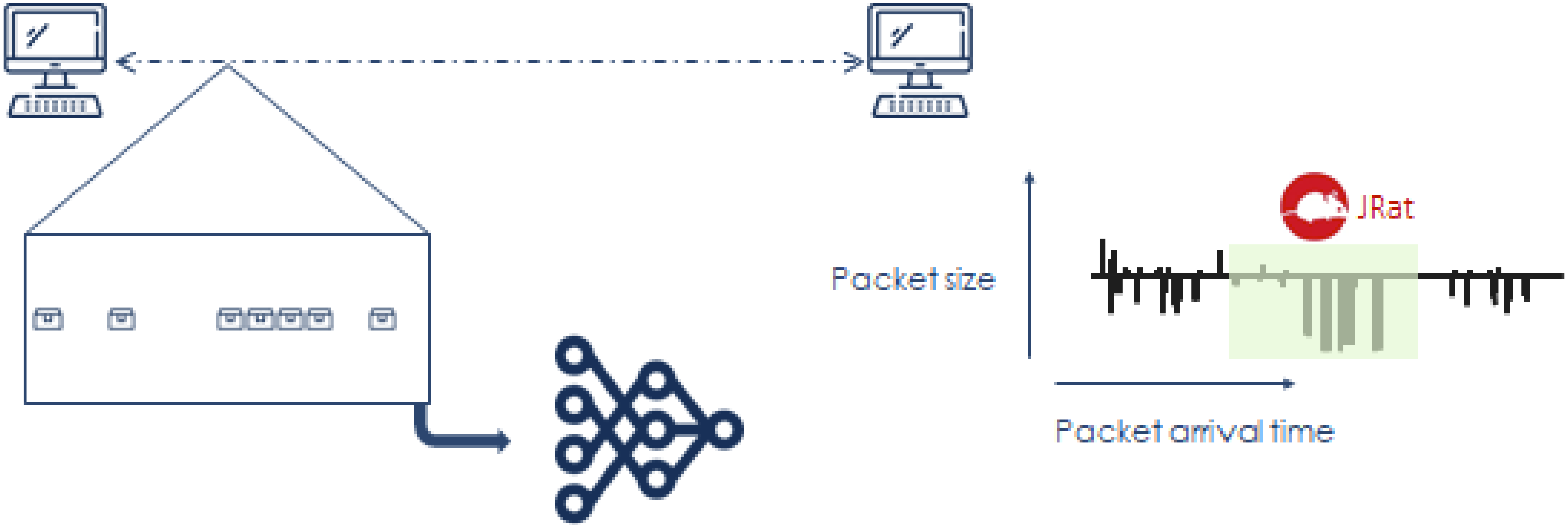
Need to work in « real time » and « real environment »

- Real time meaning ?

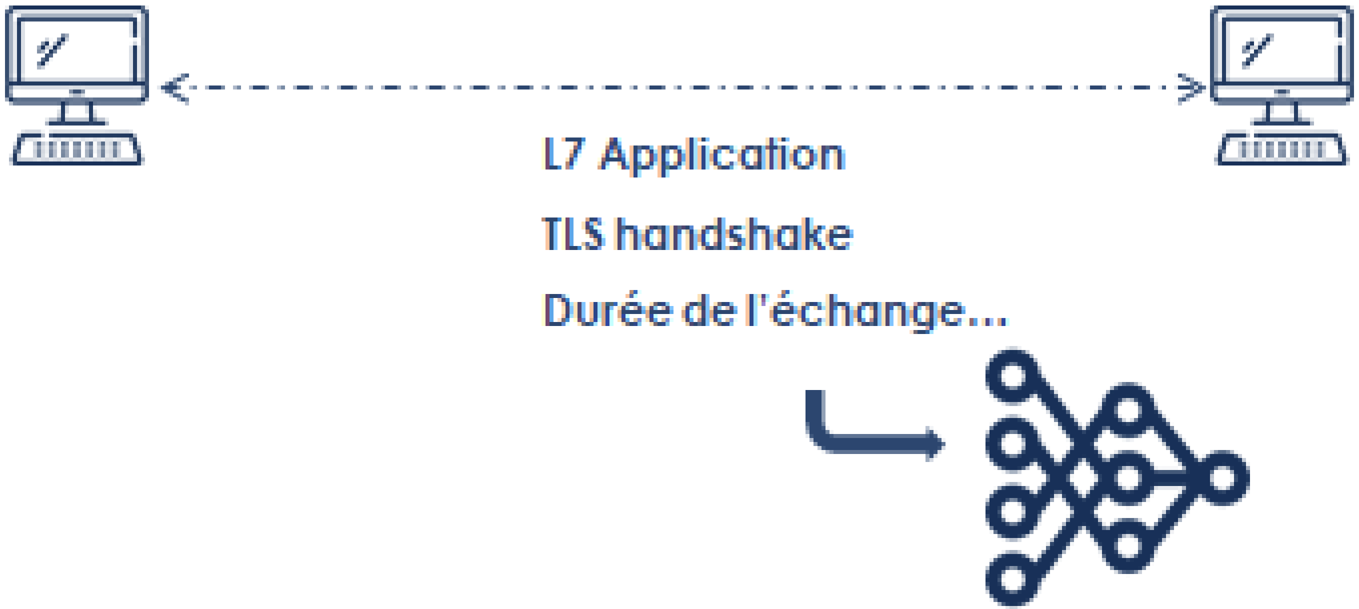
- How many flows / second ?

Custocy models

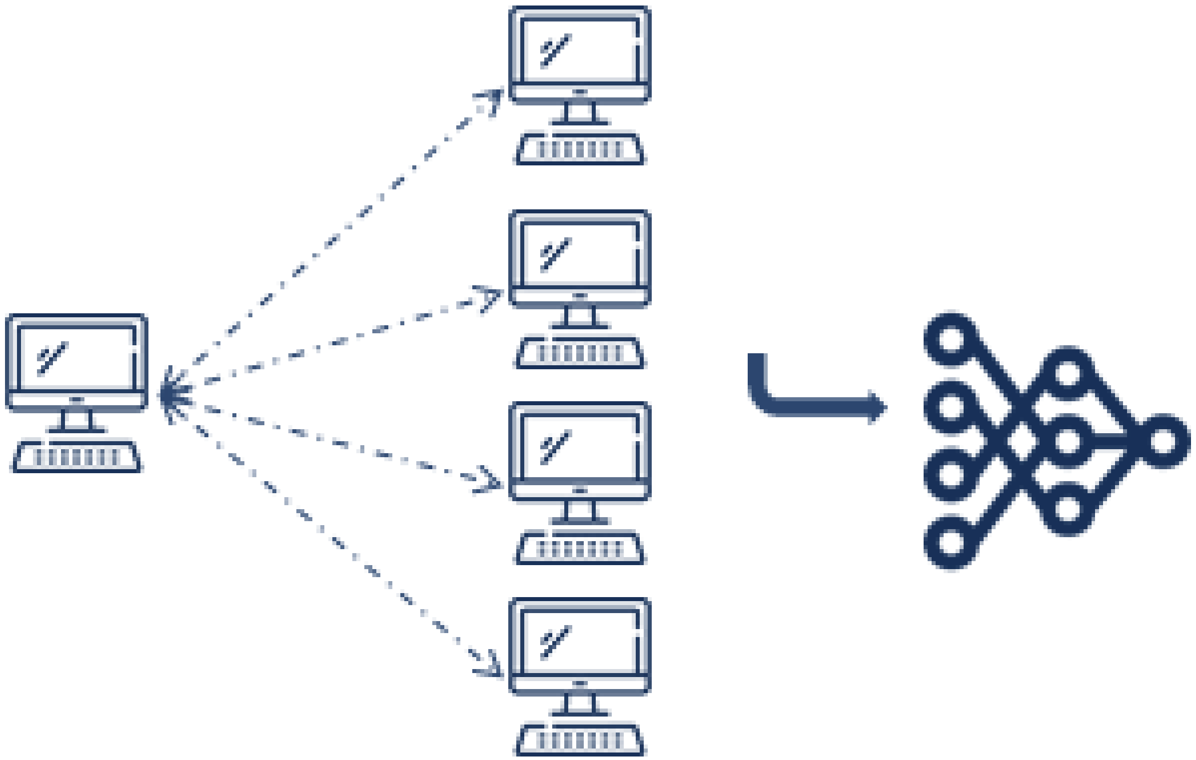
Weak model 1 : network packets



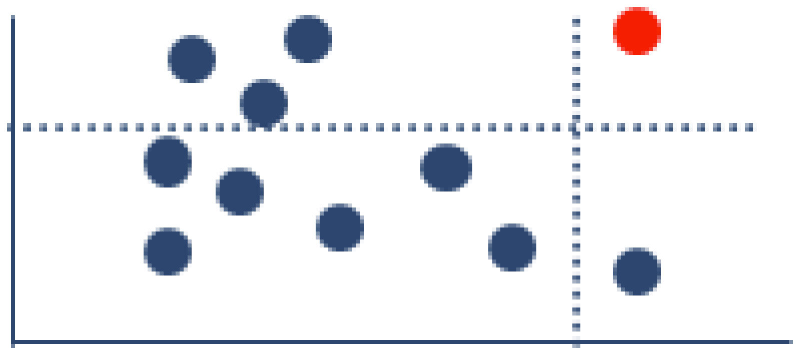
Weak model 2 : network flows



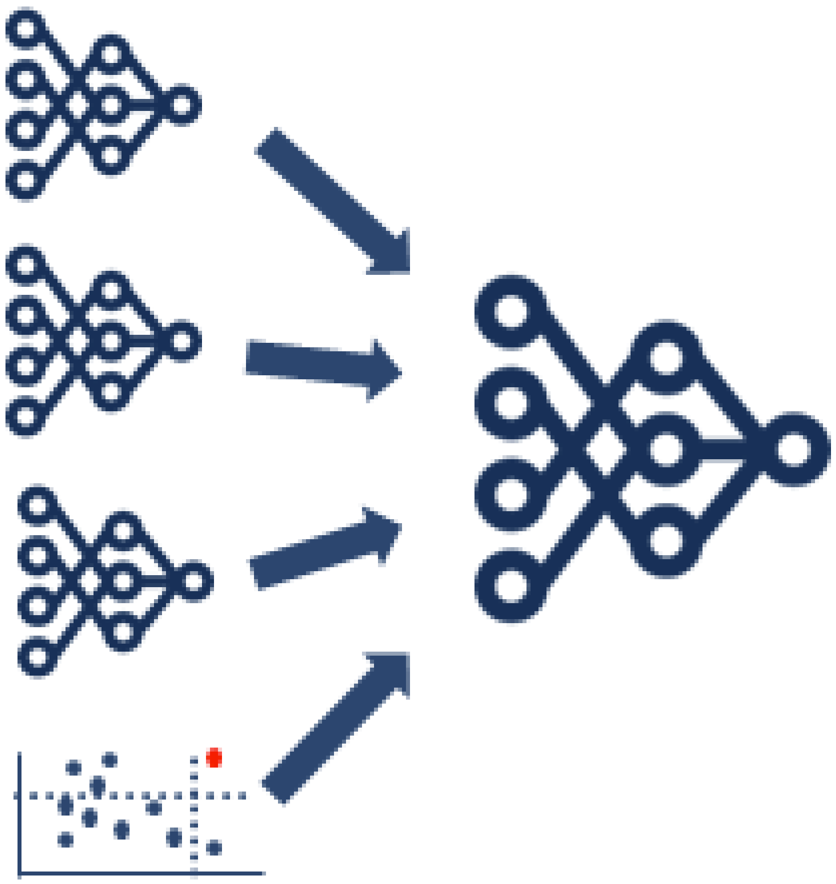
Weak model 3 : aggregation of flows



Weak model 4 : analyse comportementale



4 Weak models : 1 strong model



Ceci est une attaque

Conclusion

Appendices

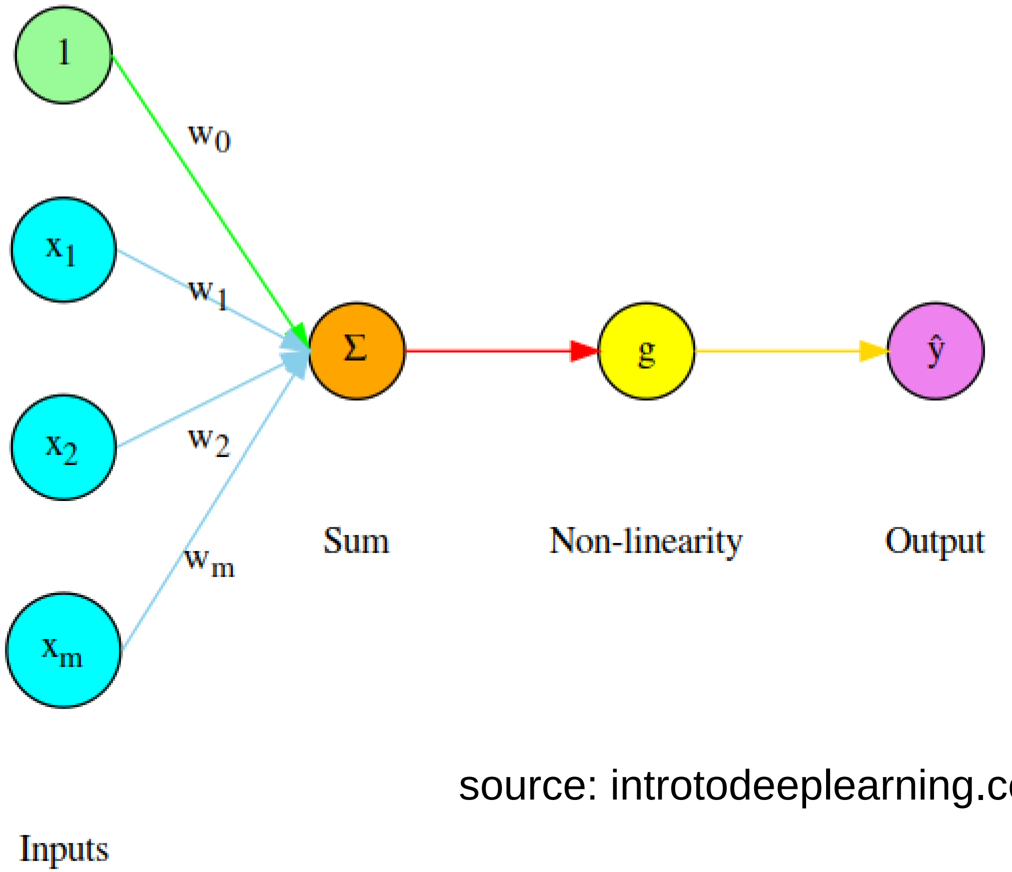
Deep learning: the perceptron

$$\hat{y} = g\left(w_0 + \sum_{j=1}^{m} x_j w_j\right)$$

single neuron computation

$$\hat{y} = g\left(w_0 + X^T W\right)$$

matrix notation

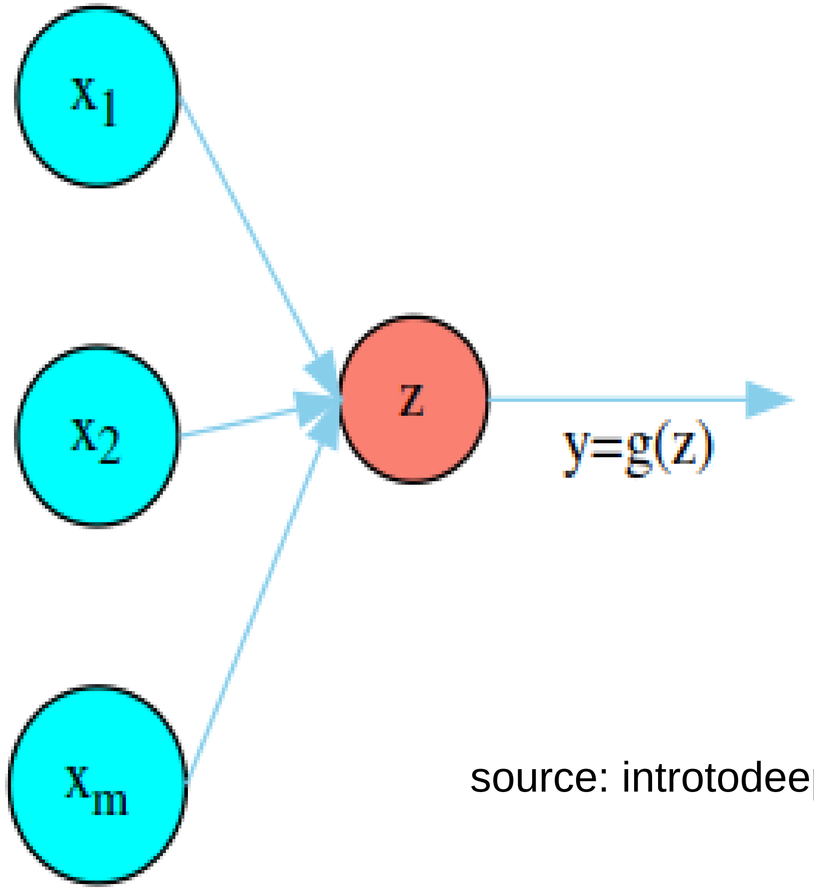


source: introtodeeplearning.com

Deep learning: the perceptron

$$z = w_0 + \sum_{j=1}^{m} x_j w_j$$

simplified input vector

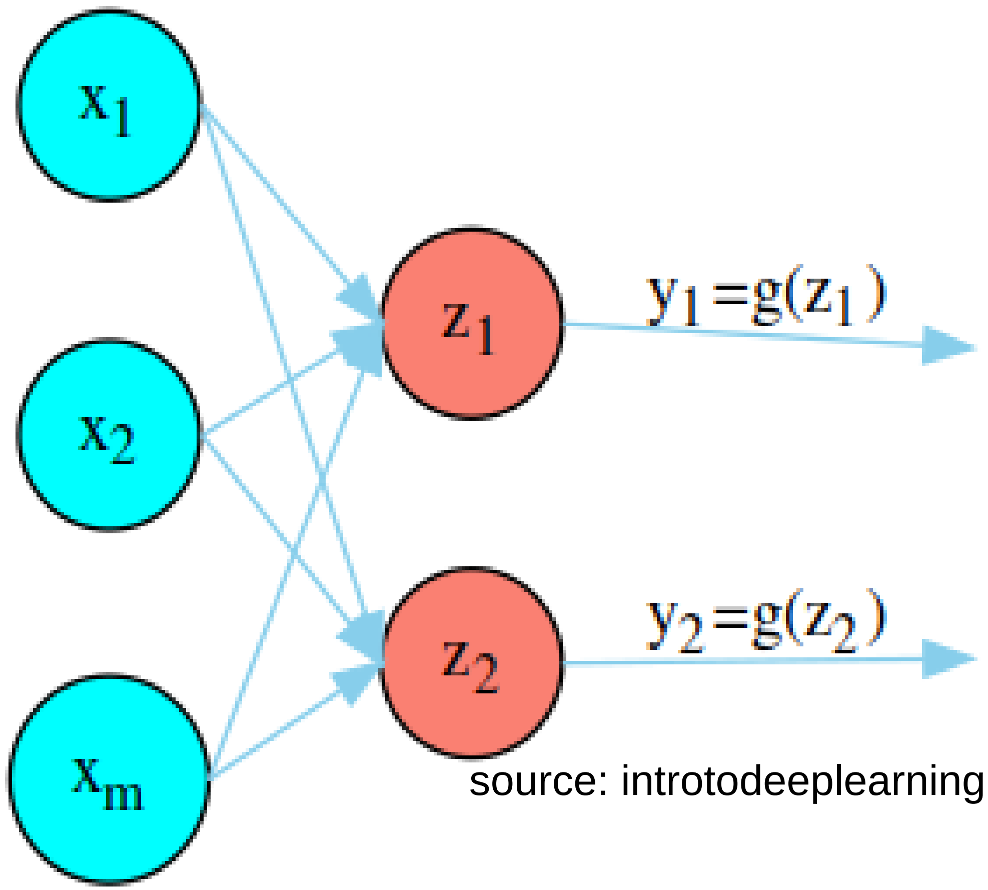


source: introtodeeplearning.com

Deep learning: multi-output perceptron

$$z_i = w_{0,i} + \sum_{j=1}^{m} x_j w_{j,i}$$

multi-output perceptron



source: introtodeeplearning.com

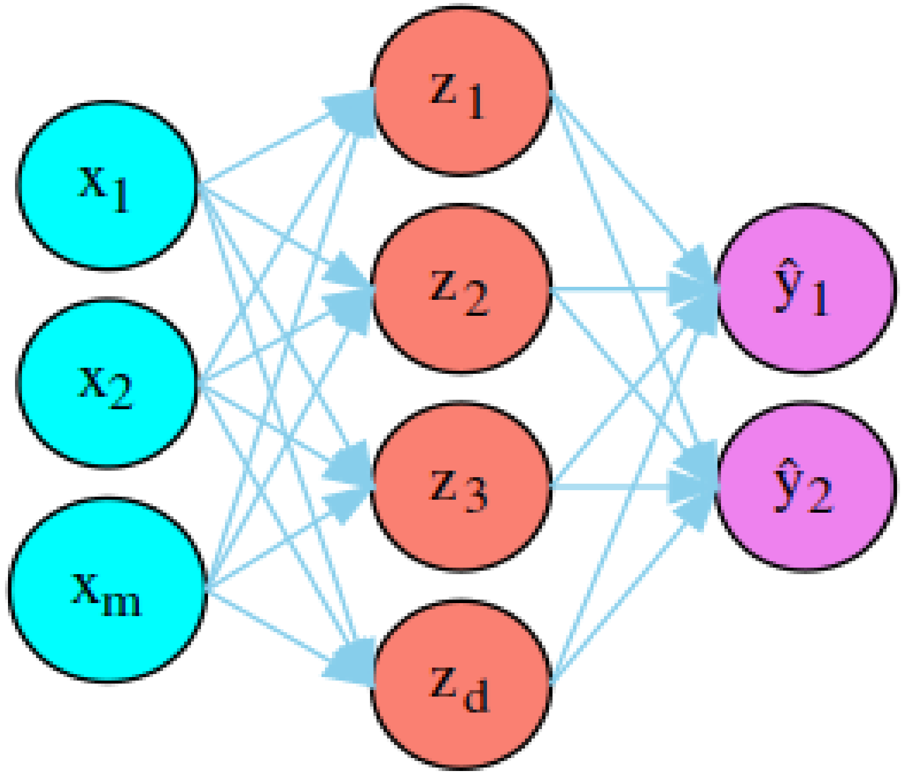
Deep learning: hidden layers

$$z_i = w_{0,i}^{(1)} + \sum_{j=1}^{m} x_j w_{j,i}^{(1)}$$

hidden layer

$$\hat{y}_i = g(w_{0,i}^{(2)} + \sum_{j=1}^{d} g(z_j) w_{j,i}^{(2)})$$

single neural network's final output



source: introtodeeplearning.com