Superviz Plenary Meeting WP4-PO4.1

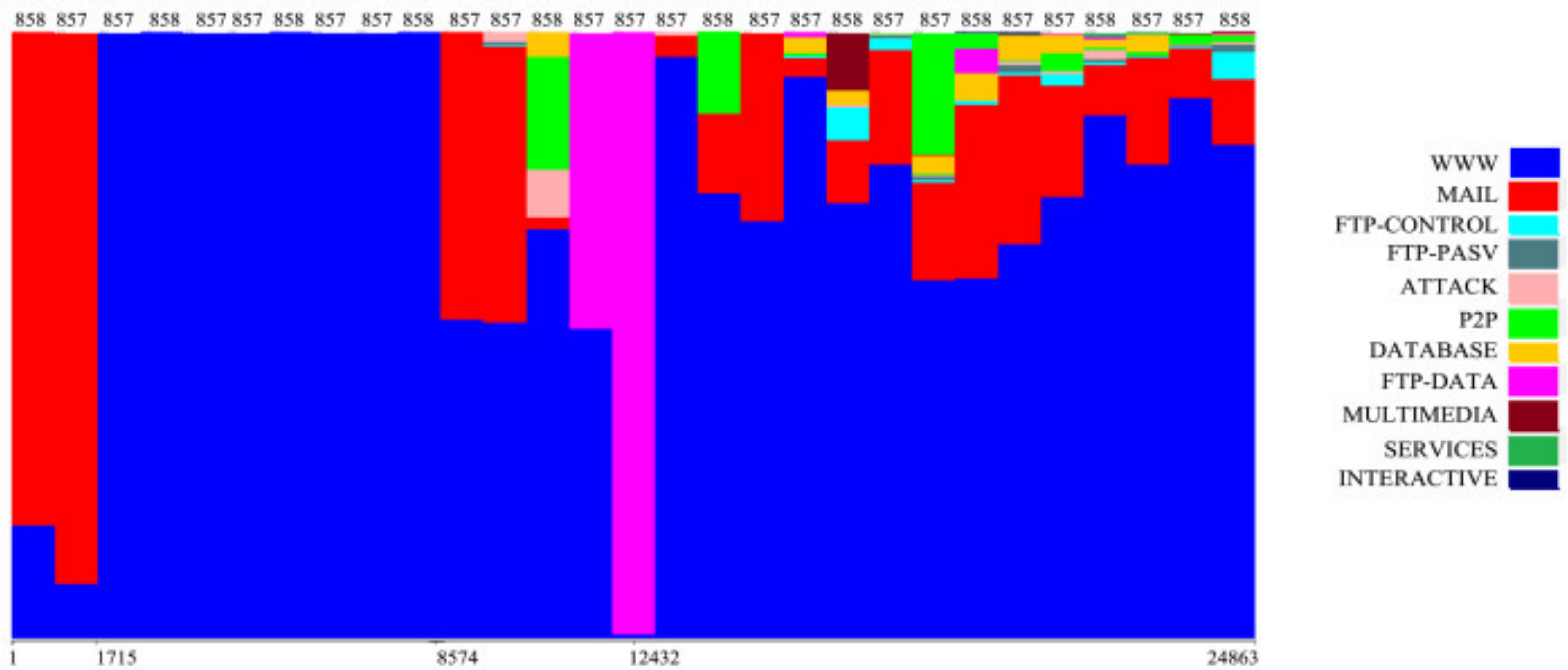
Yufei Han @ INRIA PIRAT

March 11, 2025

@Campus Cyber

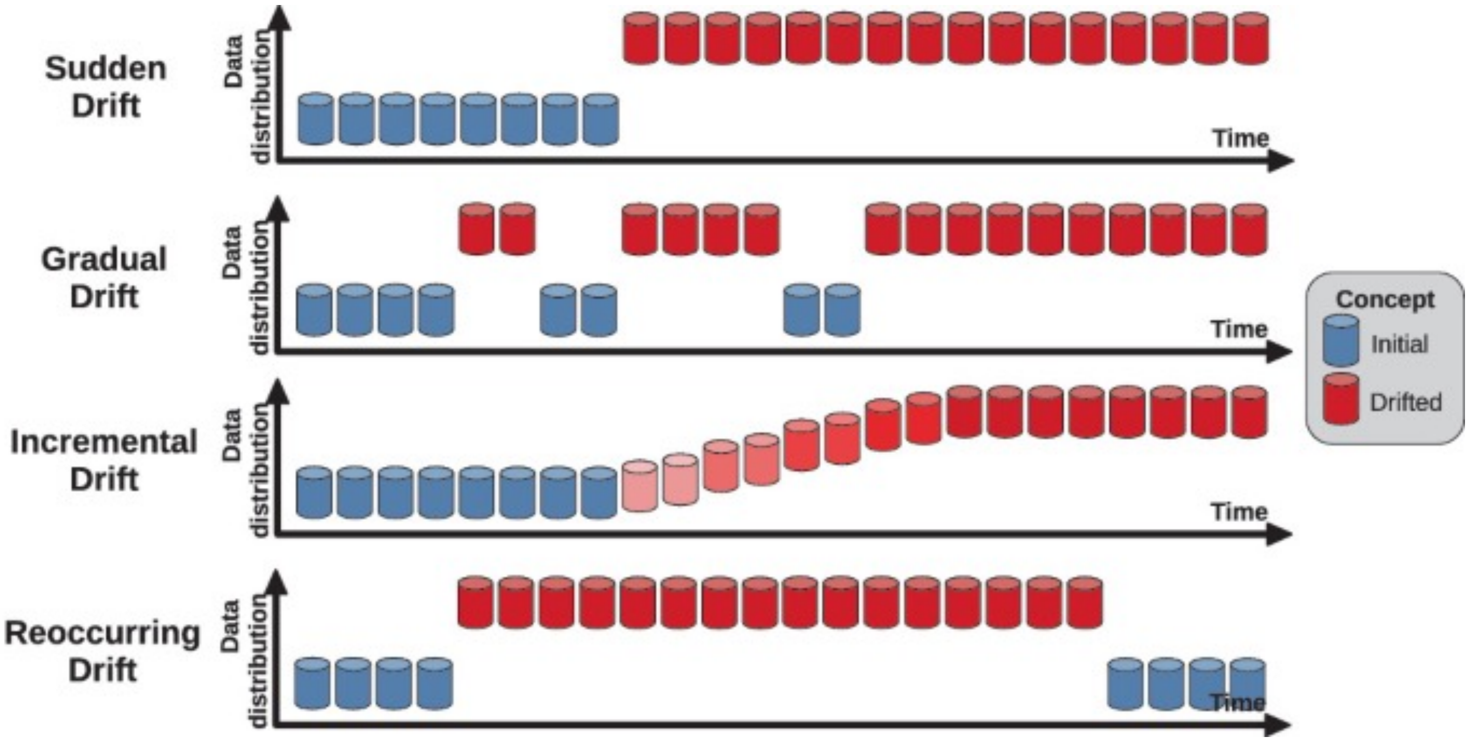
Robust and Transferable Learning for IDS

- What is the bottleneck for ML-based IDS
  - Concept drift of attack behaviours: attackers may change attack techniques to evade detection or exploit new attack surfaces



Robust and Transferable Learning for IDS

- What is the bottleneck for ML-based IDS
  - Concept drift of attack behaviours: attackers may change attack techniques to evade detection or exploit new attack surfaces



Robust and Transferable Learning for IDS

- Theoretically, every ML model has a significantly deteriorated prediction accuracy over concept drifted inputs

Theorem 1 *Let $\theta \in \mathcal{H}$ be a hypothesis, $\epsilon_s(\theta)$ and $\epsilon_t(\theta)$ be the expected risks of source and target respectively, then*

$$\epsilon_t(\theta) \leqslant \epsilon_s(\theta) + \boxed{2d_k(p, q)} + C, \qquad (9)$$

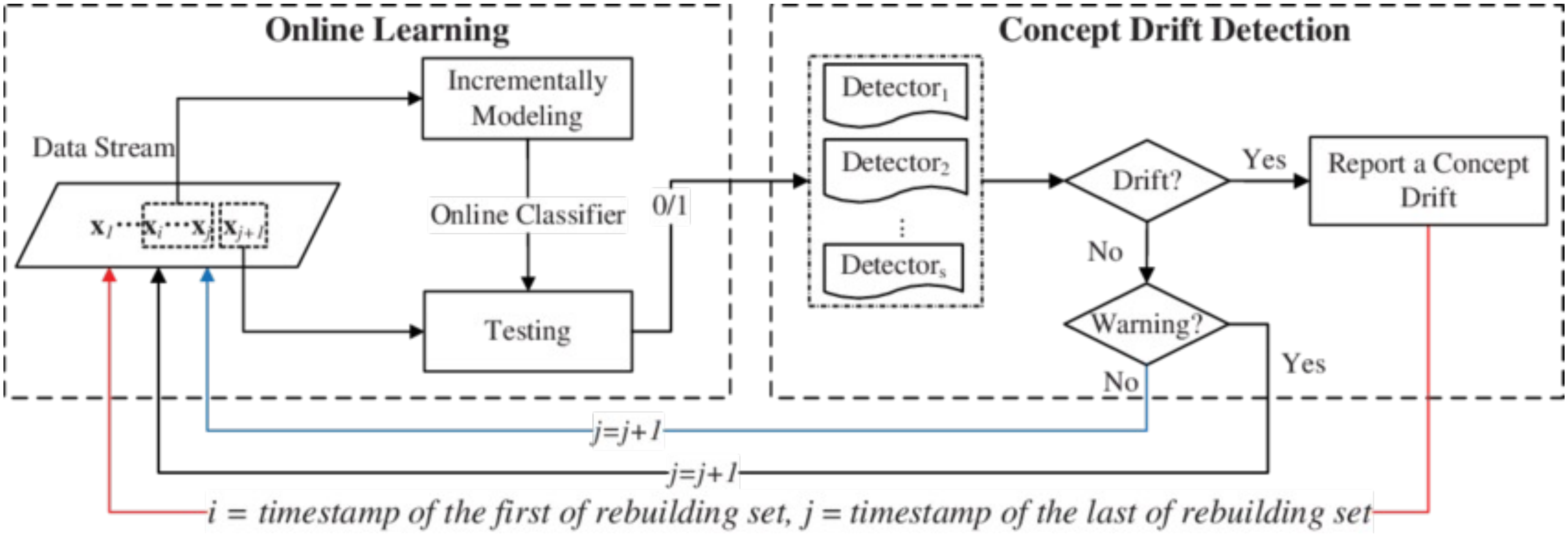
where C is a constant for the complexity of hypothesis space and the risk of an ideal hypothesis for both domains.

Distribution gap between training and testing data

Michel. I. Jordan et al, Learning Transferable Features with Deep Adaption Networks, ICML 2015

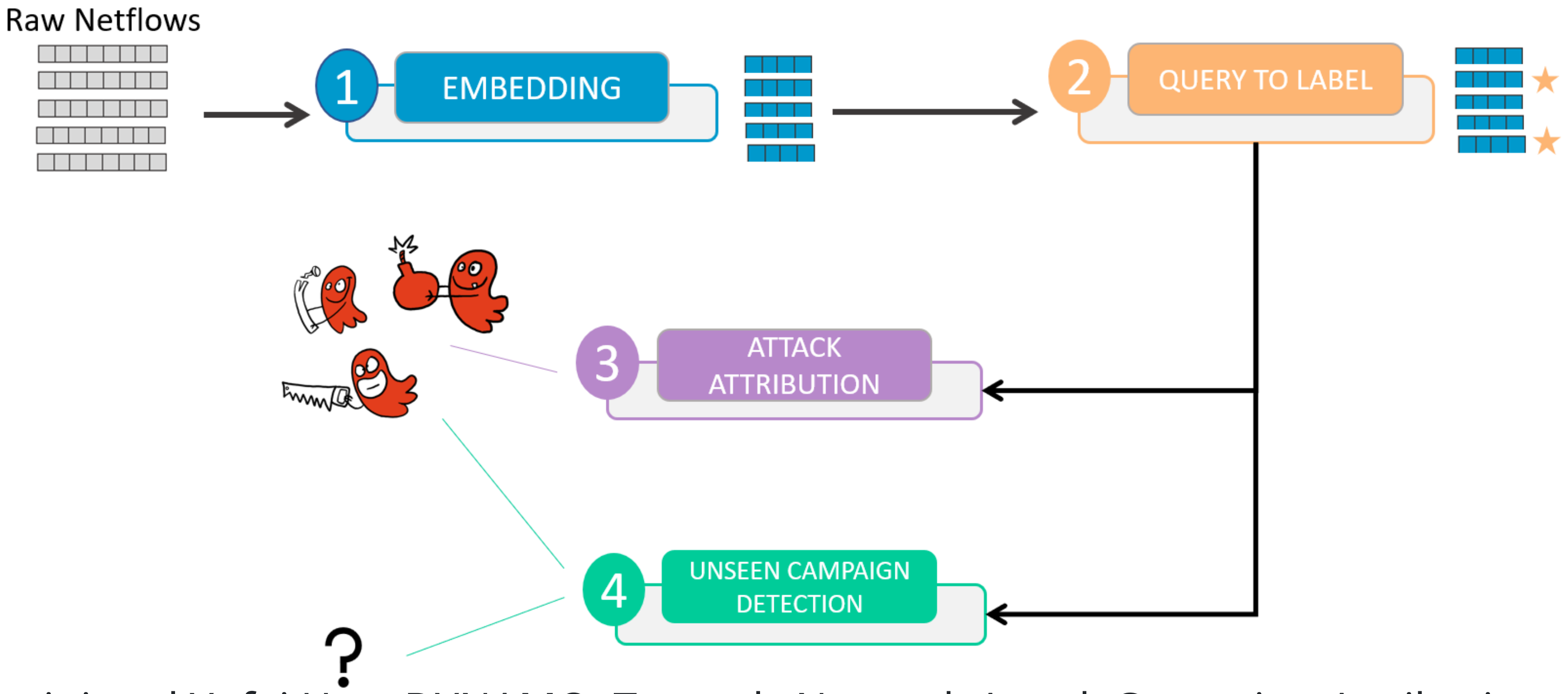
Robust and Transferable Learning for IDS

- How to reach this goal ?



Robust and Transferable Learning for IDS

- Active Learning as a protocol for incrementally update the knoweldge for network attack classification



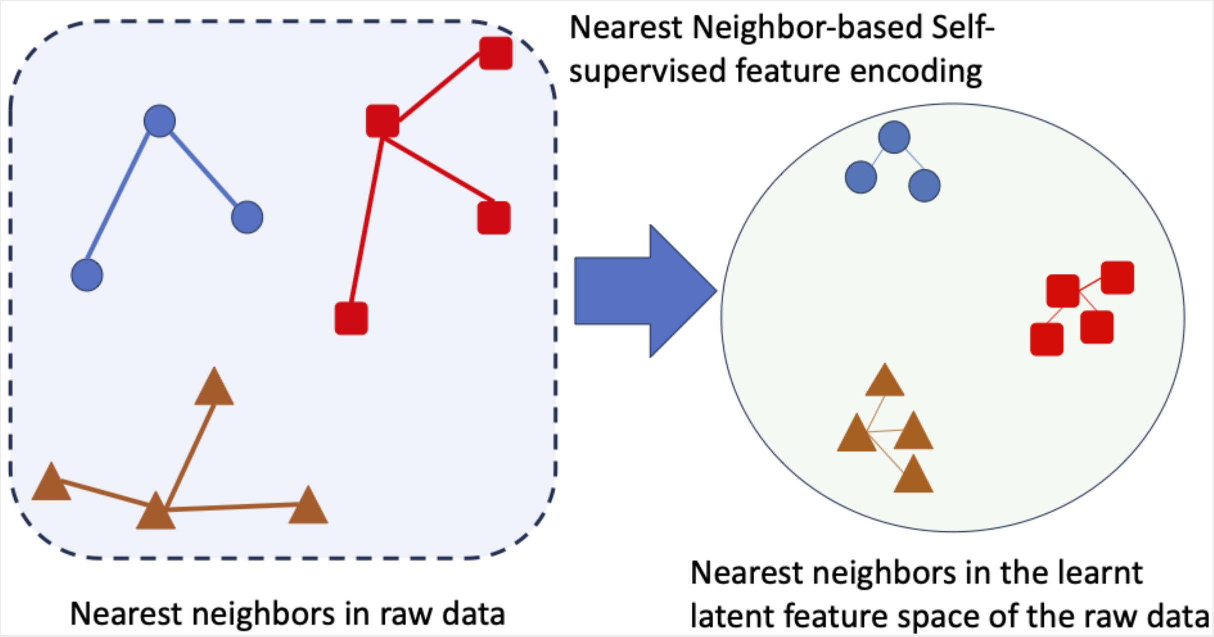
Helene Orsini and Yufei Han, DYNAMO: Towards Network Attack Campaign Attribution via Density-Aware Active Learning, https://hal-emse.ccsd.cnrs.fr/UNIV-UBS/hal-04877620v1

Robust and Transferable Learning for IDS

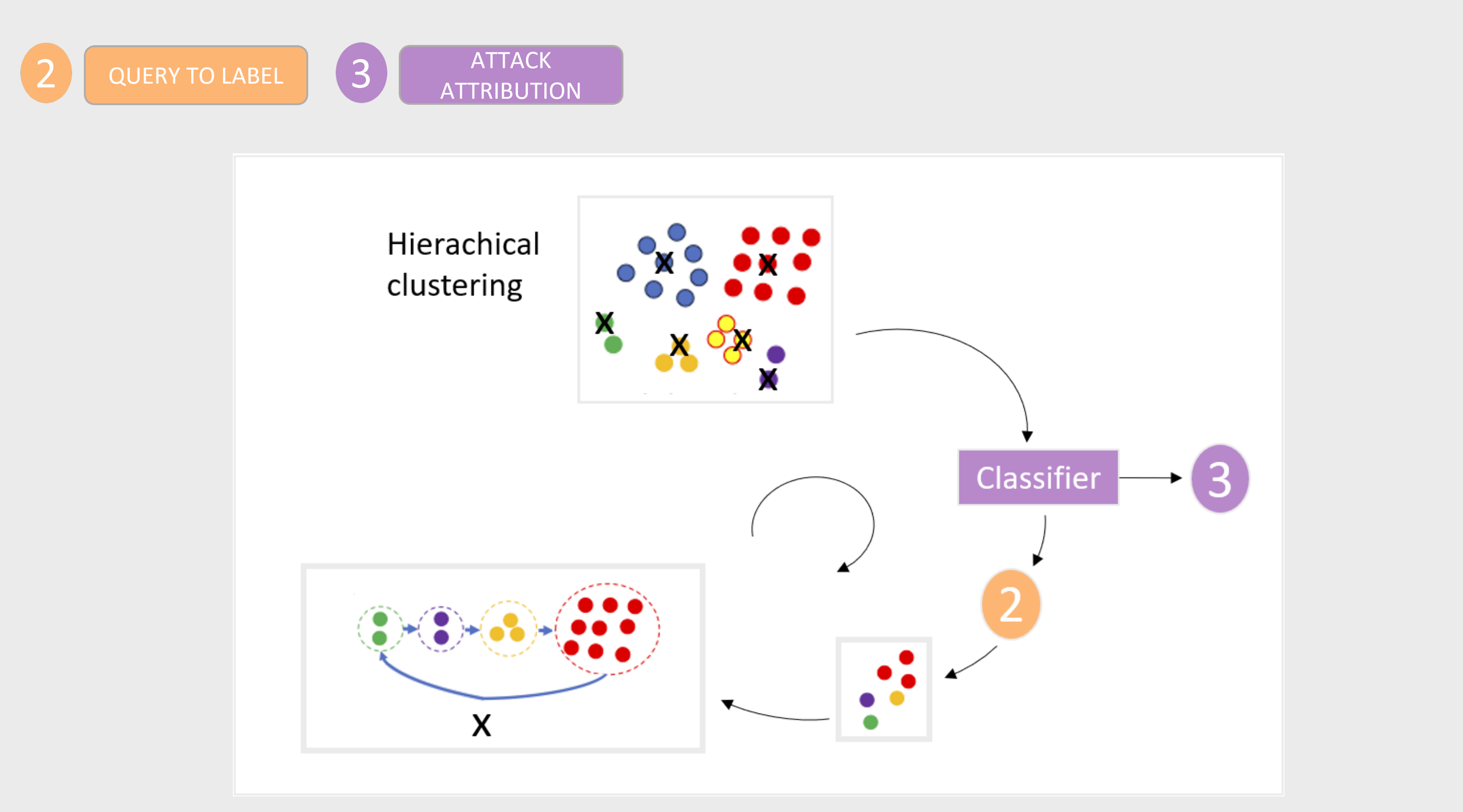
1 EMBEDDING

Raw feature vectors from Netflow : GraphSage method

$$\theta^* = \arg\min_{\theta} - \frac{1}{nK} \sum_{i=1}^{n} [\sum_{k=1}^{K} log(\sigma(h_\theta^T(x_i) h_\theta(x_i^{NN,k}))) - \lambda \sum_{j, x_v \notin \mathrm{KNN}(x_i)} log(\sigma(-h_\theta^T(x_i) h_\theta(x_v)))]$$



Robust and Transferable Learning for IDS



Robust and Transferable Learning for IDS

4 UNSEEN CAMPAIGN DETECTION

Pu learning

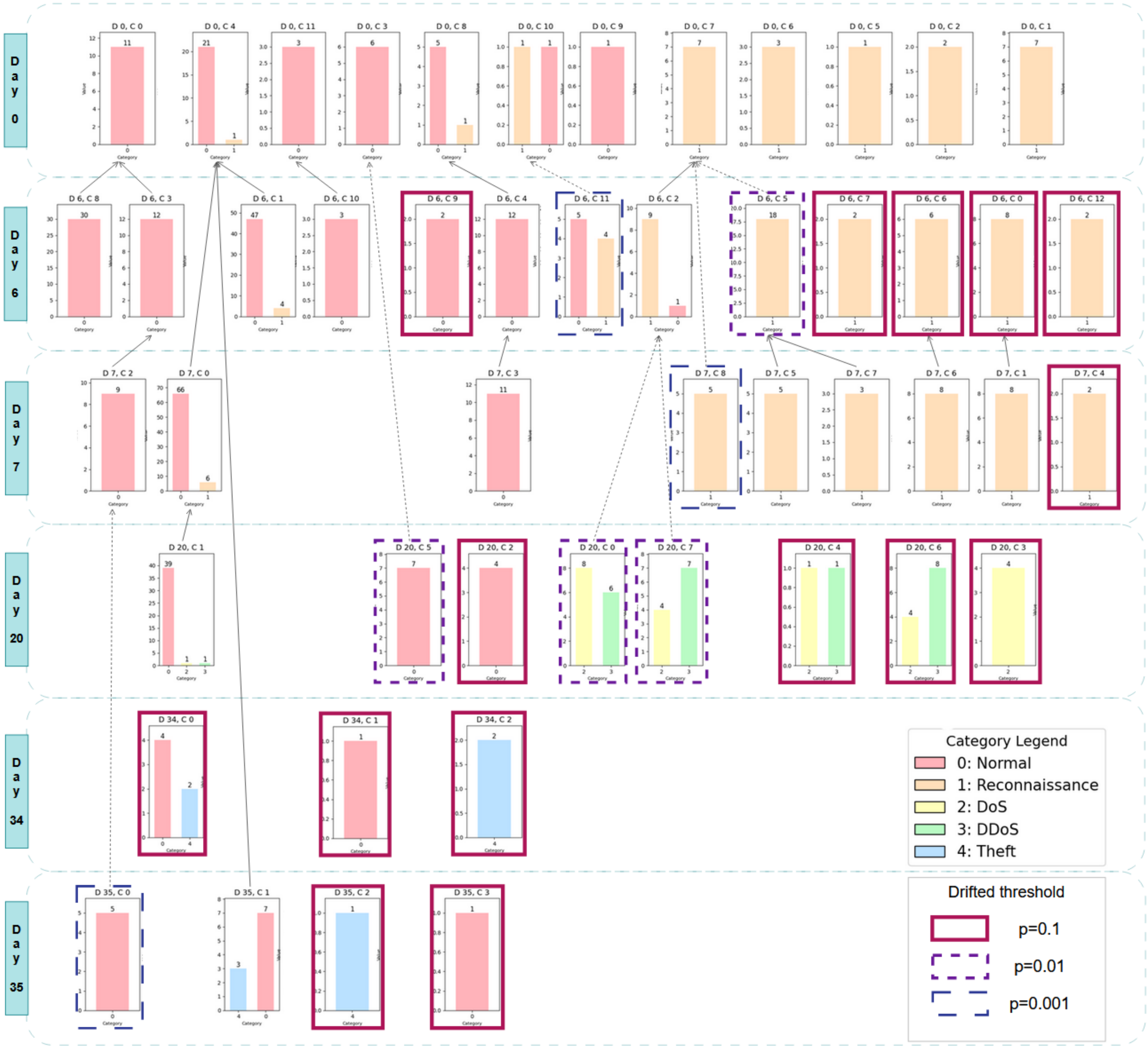
Train a classifier to distinguish between positive and negative.

Learning phase: **Positive and Unlabelled** (*only some of the positive examples in the training data are labeled and none of the negative examples are*)

$$g_\phi^{pu} = \arg\min_{\phi} \frac{\pi}{n_p} \sum_{x_i \in S} [\ell(g_\phi^{\mathrm{pu}}(h_\theta(x_i)), y_i = +1) - \ell(g_\phi^{\mathrm{pu}}(h_\theta(x_i)), y_i = -1)] + \frac{1}{n_u} \sum_{x_i \in X_{\mathrm{unlabeled}}} \ell(g_\phi^{\mathrm{pu}}(h_\theta(x_i)), y_i = -1)$$

Robu ꞏꞏꞏ)S

• Extenc



Pacl

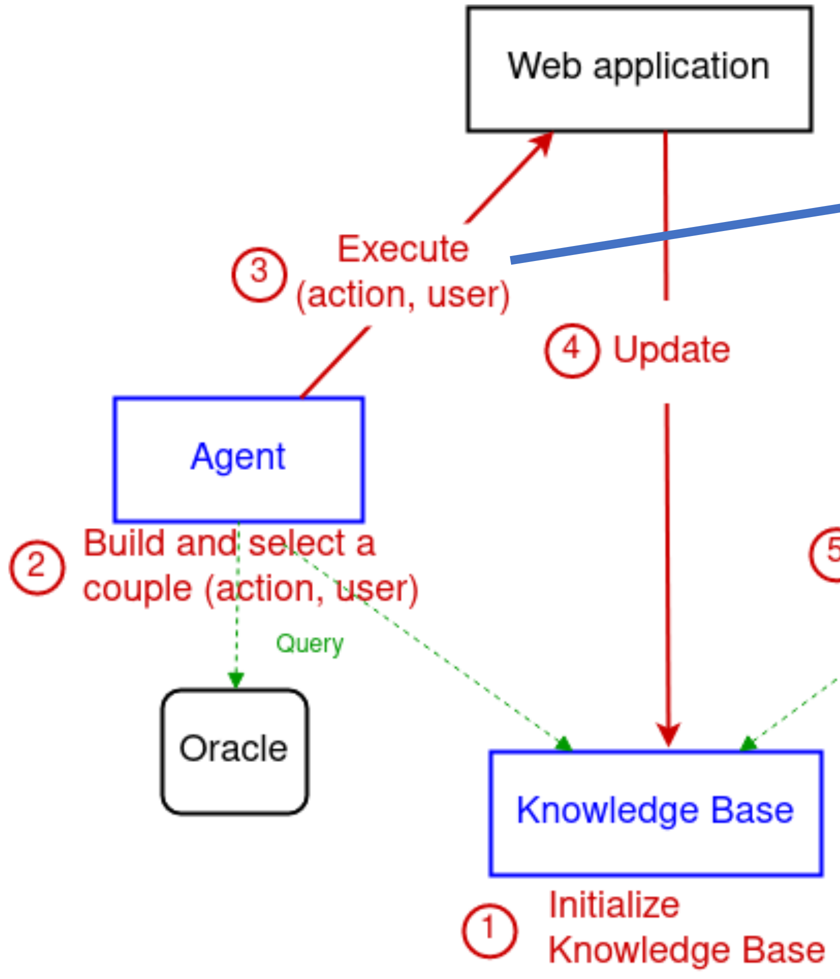
pack
pack
pack
pack
pack
pack
pack
pack

Robust and Transferable Learning for IDS

- Back to the question “How to reach a robust IDS facing concept drift ?”
- Can we predict the attack behaviour ?
- Natan Talon et al, SCWAD: Automated Pentesting of Web Applications, https://inria.hal.science/hal-04874868v1/document

Robust and Transferable Learning for IDS

Vulnerabilities, e.g. Broken Access Control and Reflected Cross-Site Scripting.



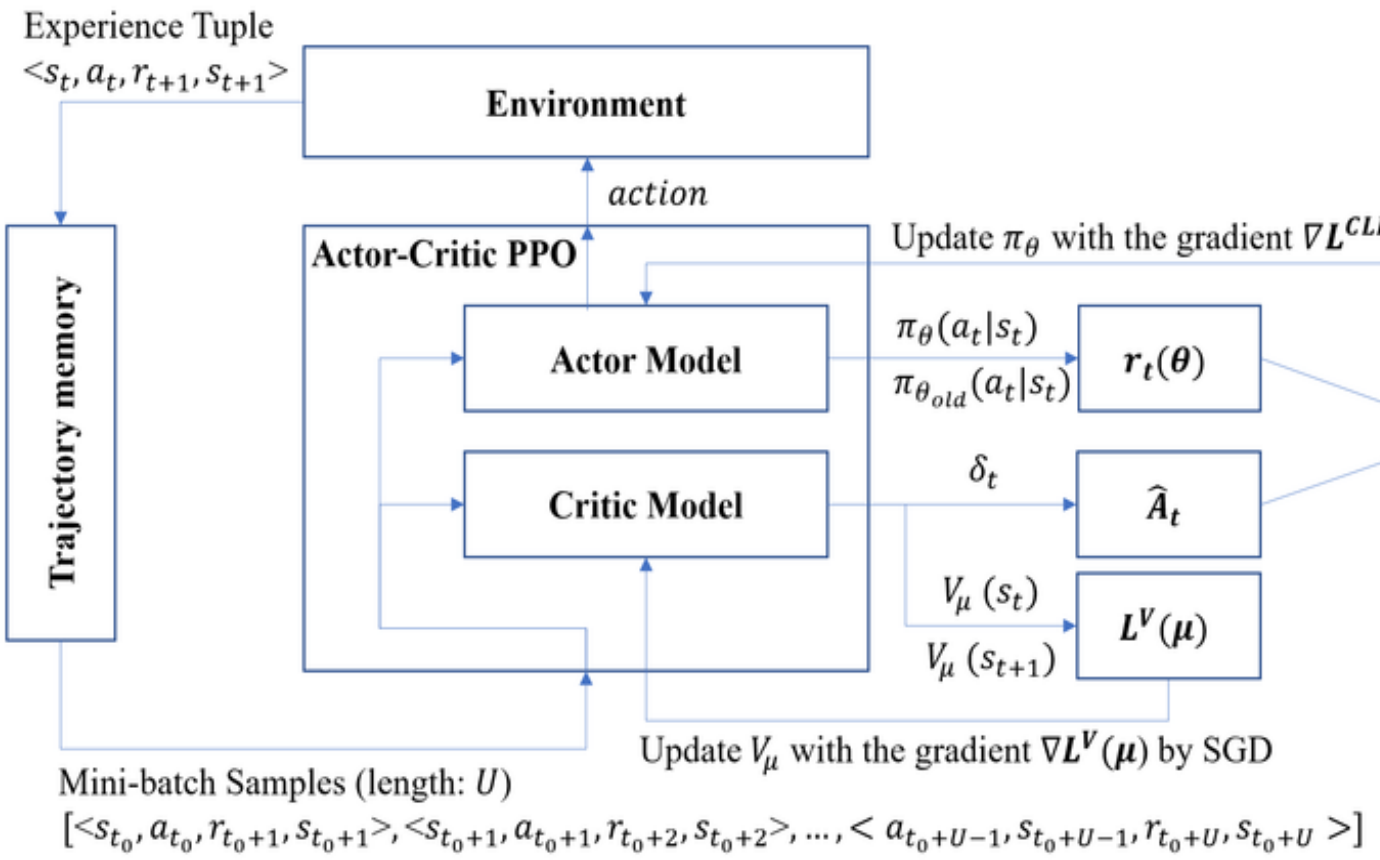
Action set

Command	Parameters	Expected behavior on the Web application
`AccessWebPage`	`u`: *url*	change current page for the current user
`SendHttpForm`	`x`: *xpath* , `input`: *dict* { *key: value* }	send the form identified by `x` and filled with `input` to the web application
`SearchData`	`data`: *string*	return `true` if `data` has been found in the current page
`SetCookie`	`cookie`: *dict* { *key: value* }	add `cookie` to user's cookies if this cookie doesn't exists and replace it otherwise

User login: user name
User credentials: The credentials for the application
User cookies: The set of cookies, active or not
User pages: The pages visited by the user and/or those to which the user has a link
User current page: The current page visited by the user
User allowed paths: All the links to the pages reachable by the user

Robust and Transferable Learning for IDS

- Learn an attack policy with reinforcement learning



Key question to answer:

Given the current staste of the target web application and the vulnerability to explore, what shall be the most likely action that an attack can take to compromise the application ?

Thanks !