Superviz Plenary Meeting WP4-TH4.1

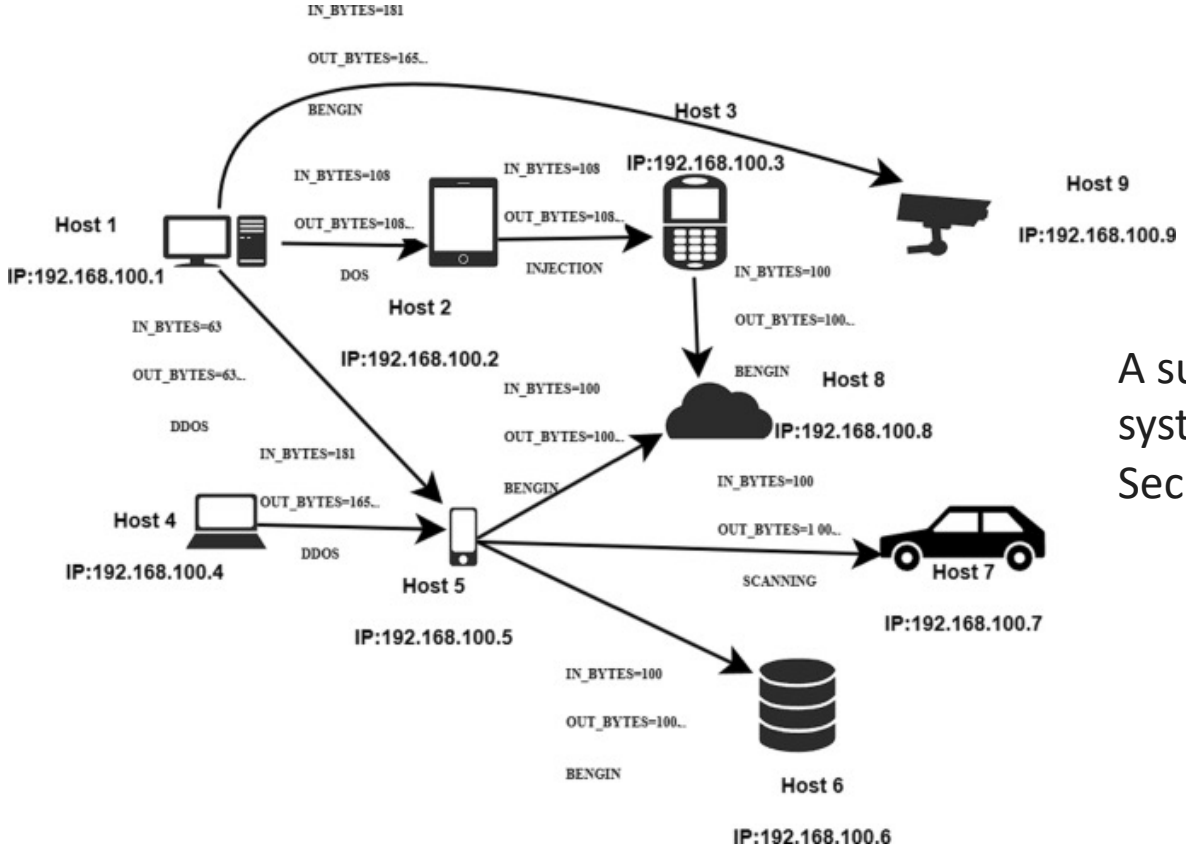
Yufei Han@INRIA PIRAT

March 11, 2025

@Campus Cyber, Paris

Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

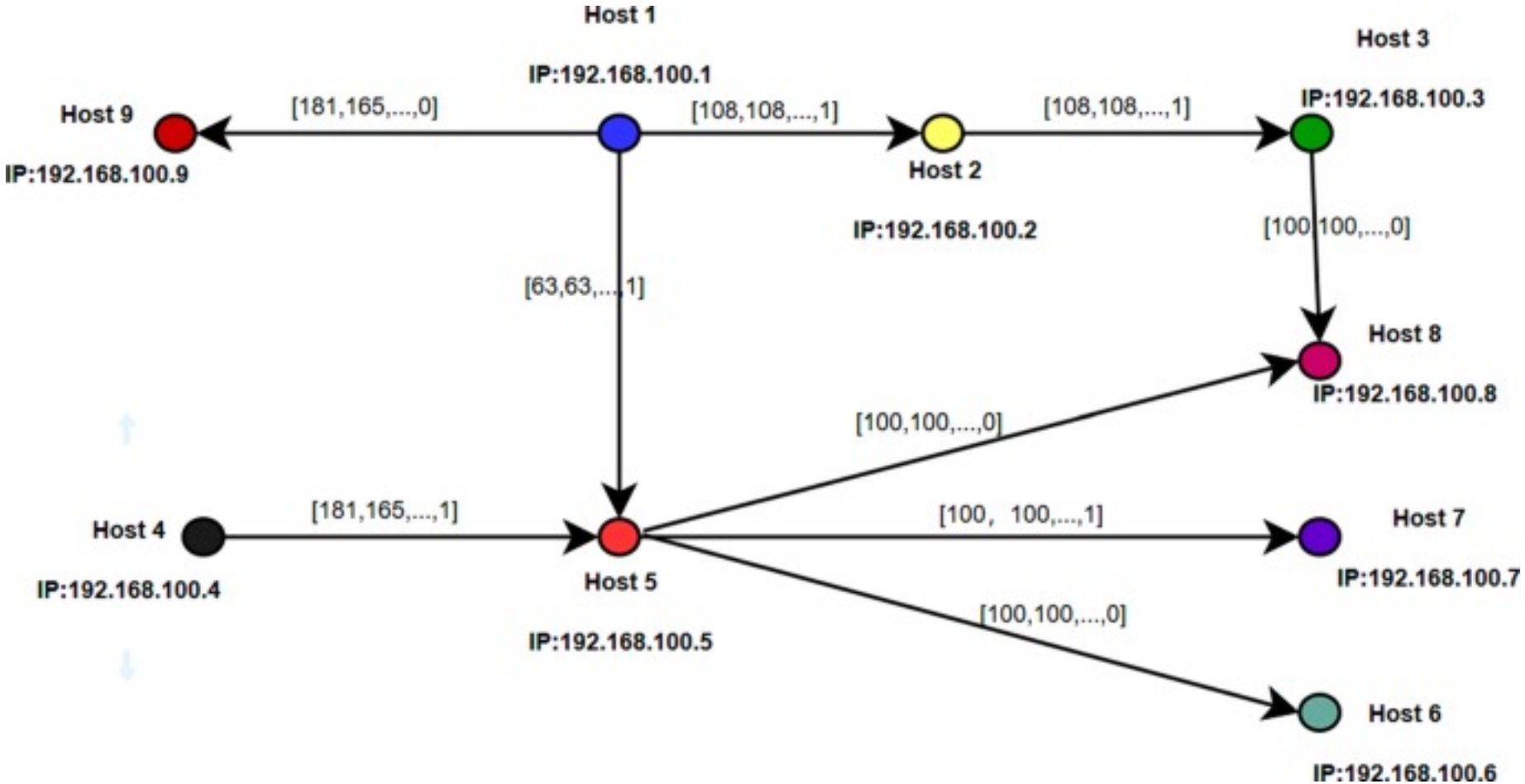
- Why Graph-based Intrusion Detection ?
- Graph provides a structural representation of cyber attack behaviour



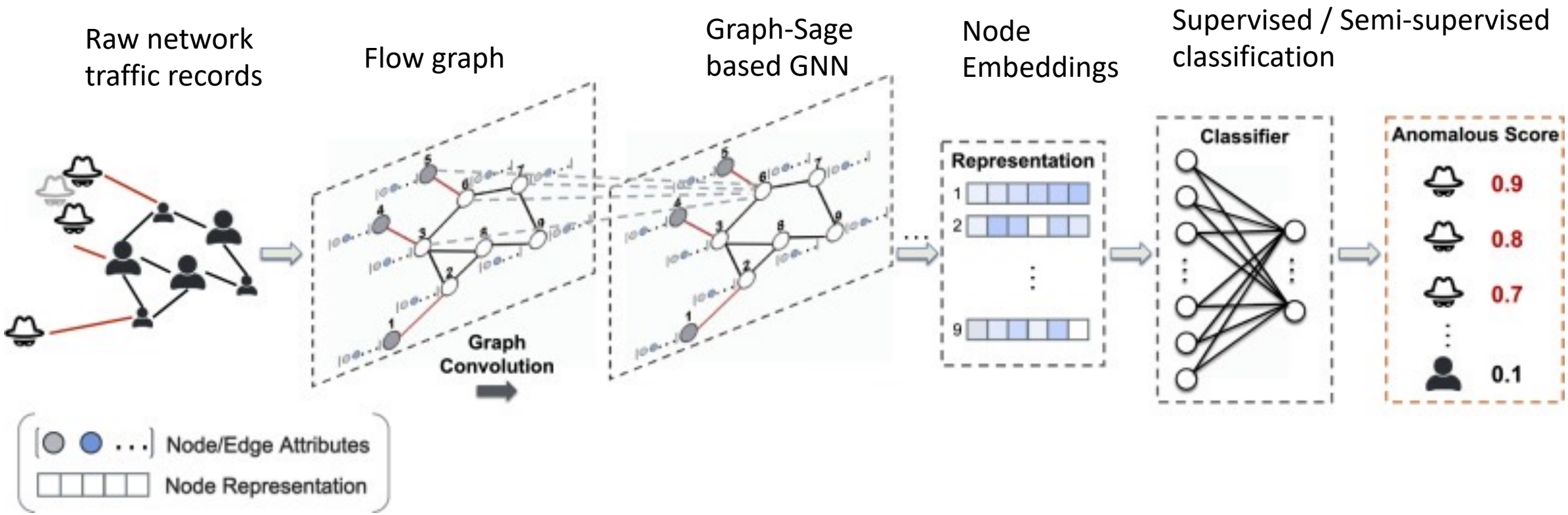
A survey on graph neural networks for intrusion detection systems: Methods, trends and challenges, Computer & Security, Vol.141, June 2024

Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

- Target: **Flow Graph** based representation of network traffics



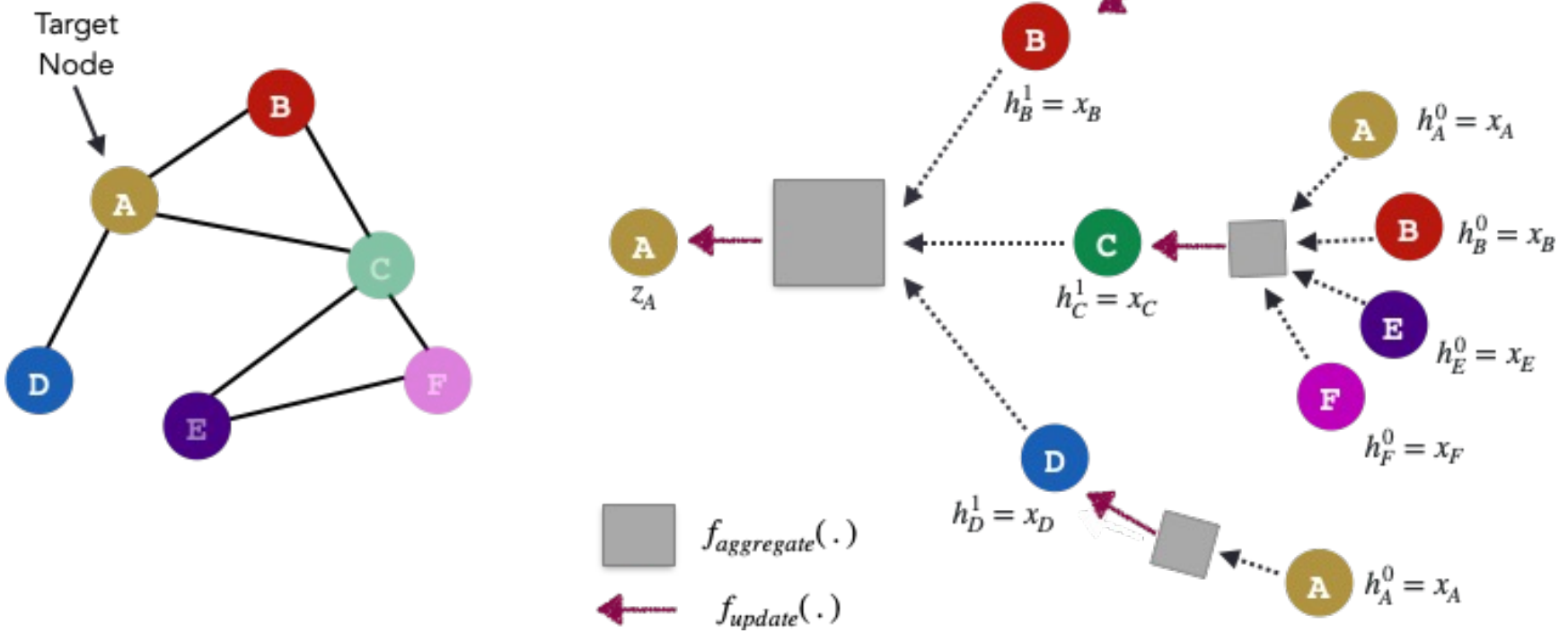
Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection



Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

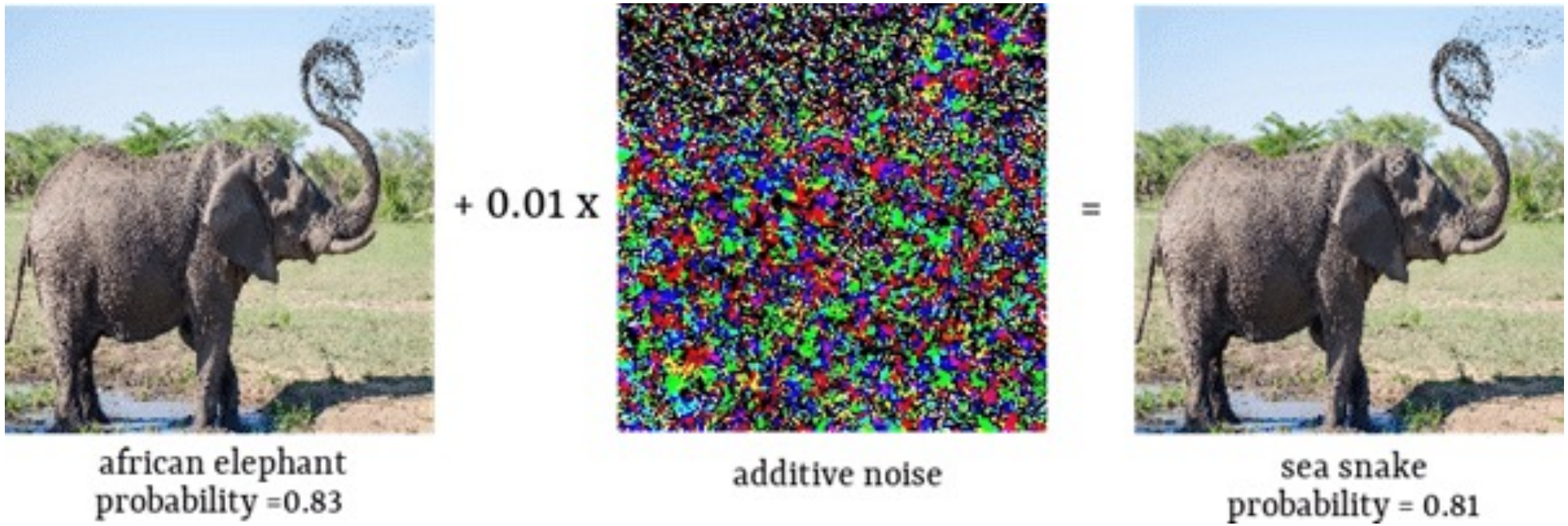
- A few words about Graph-Sage based GNN models

GNN = A Recurrent Message Passing Process across K-hop neighbours



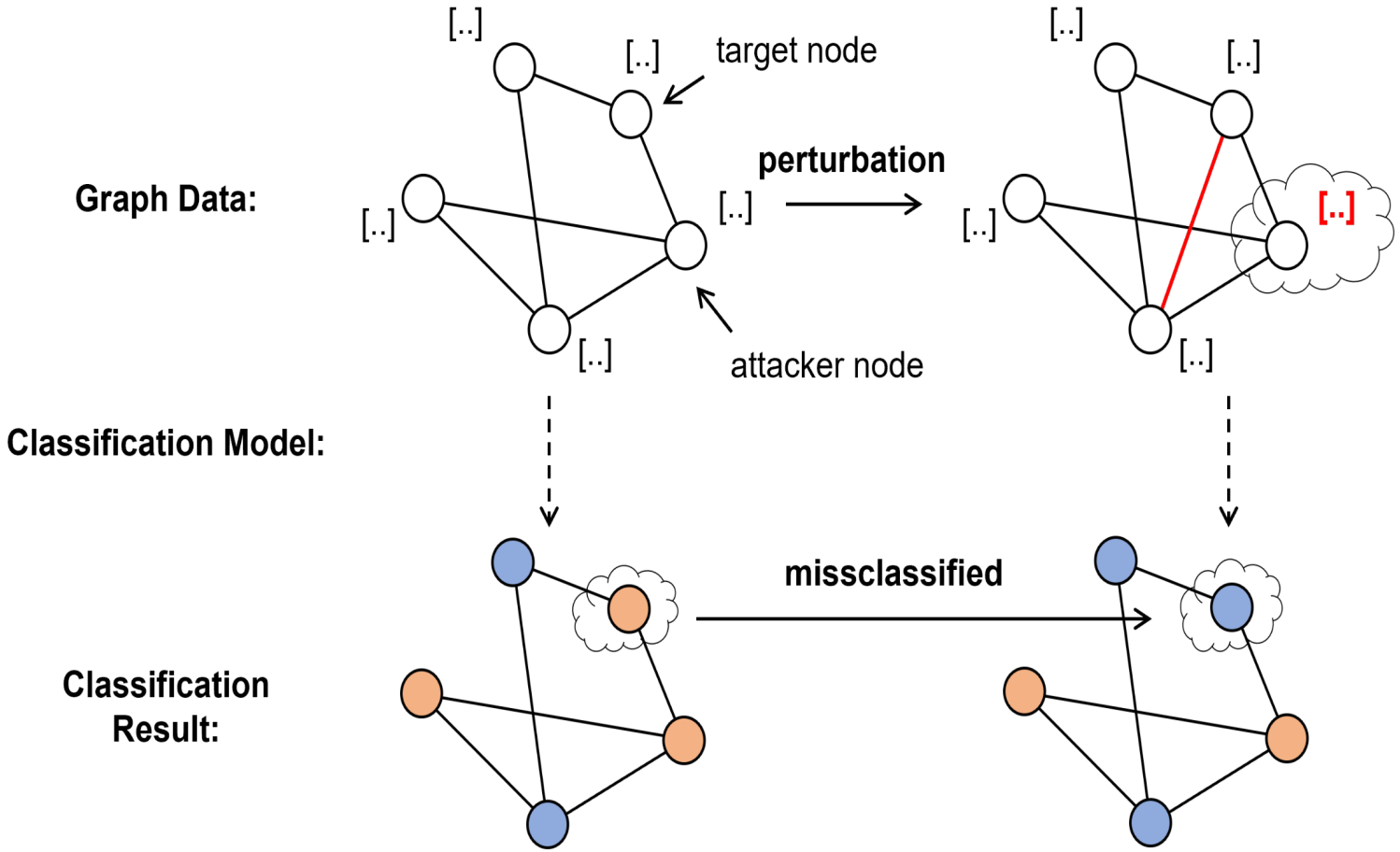
Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

- What is an adversarial attack ?



Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

- An adversarial attack against Graph Neural Network-based models

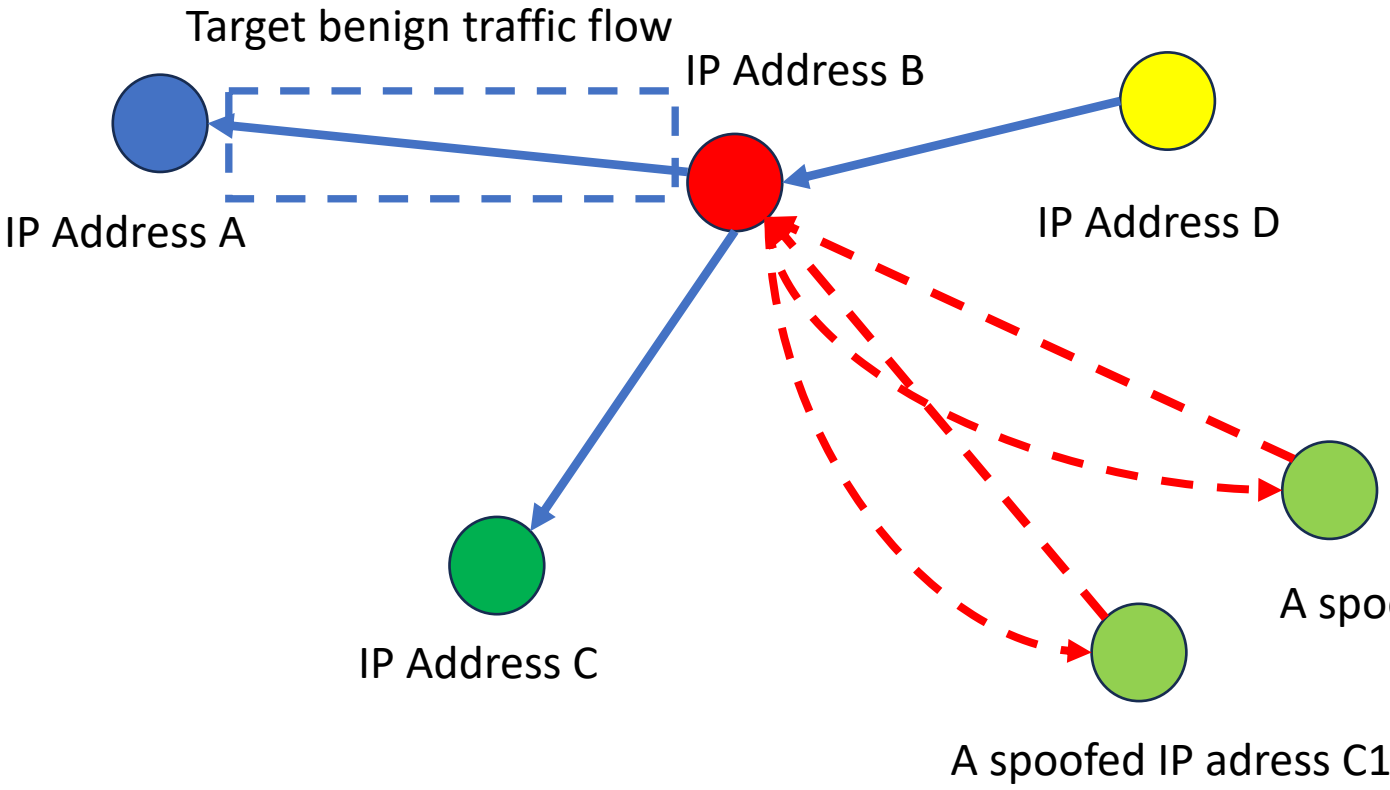


Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

- What is the problem space constraint ?
- Any change to the input graph should not break the attack / normal traffic flows
  - An attack is still an attack
- Any change to the input graph should be made consistent with the deployed communication protocols
- In the end, any change to the input graph should be made compatiable with the profiles of real traffic flows between IP addresses.

Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

- An example of problem-space adversarial attacks



Problem-space constrant:

Edge addition: An attacker can send traffic flows from a spoofed IP address to an IP address existing in the raw traffic records.

What edges to add: The added traffic flows should be added in a way consistent with the communication protocol, such as TCP-IP

Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

- Mathematical nature of the edge-moification-based adversarial attacks

$$\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{E}\}$$

Adjacency matrix

Node attributes

Edge attributes

$$\boldsymbol{A}^* = \arg\max_{\boldsymbol{A}'} \ell(\mathcal{M}(\boldsymbol{A}', \boldsymbol{X}, \boldsymbol{E}, \boldsymbol{Y}))$$

$$s.t. \ \|\boldsymbol{A} - \boldsymbol{A}'\|_{L_0} \leq 2b$$

Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

- Every GNN is vulnerable to edge modification-based adversarial attacks

Let $\mathbf{y}$ be the target class label of the evasion attack. The goal is to make $f_y(\hat{\mathbf{x}})$ deviate from $f_y(\mathbf{x}) = 0$. In other words, we aim at maximizing $f_y(\hat{\mathbf{x}})$, so that $\mathbf{x}$ is modified to get $\mathbf{y}$ assigned to it. The evasion attack task can then be formulated as a process of set function optimization, defined as

$$S^* = \arg\max_{|S| \leq K} g(S)$$

$$\text{where} \quad g(S) = \max_{l \subset S} f_y(\hat{\mathbf{x}}), \quad l = diff(\mathbf{b}, \hat{\mathbf{b}})$$

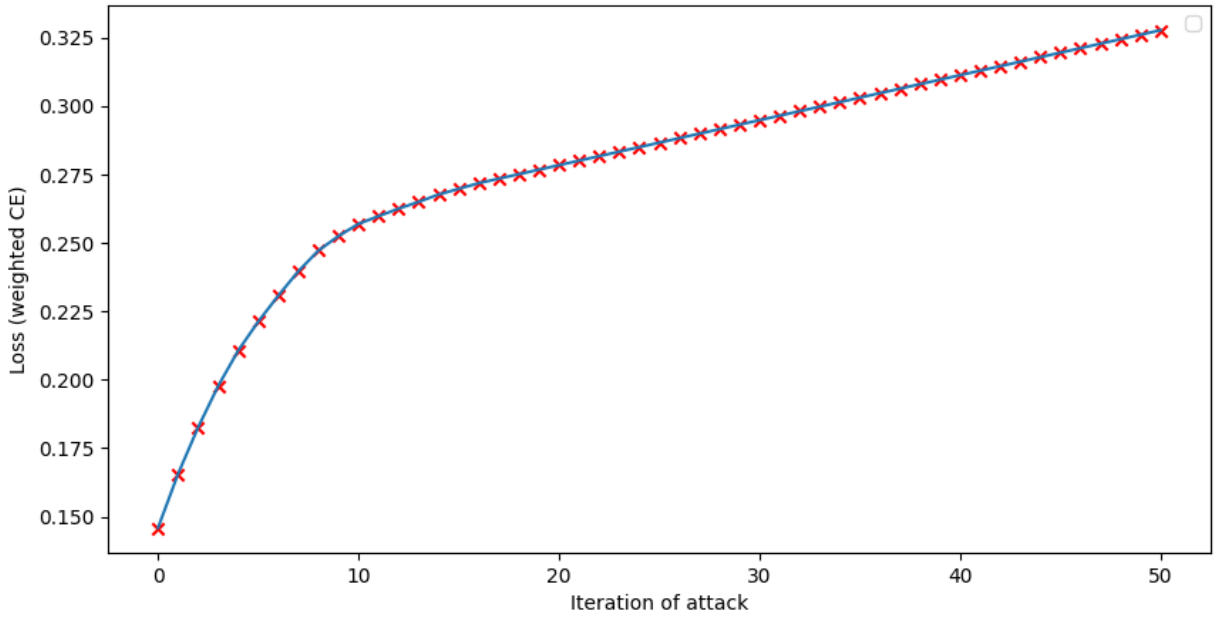
This is in nature a weakly-submodular function maximization problem , which can be solved with Greedy Search with a polynormal complexity.

Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

Algorithm 1: Unconstrained Adversarial Attack

Input: Original flow graph $\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{E}\}$, attack budget b, target model $\mathcal{M}$, loss function ℓ
Output: Modified flow graph $\mathcal{G}' = \{\boldsymbol{A}', \boldsymbol{X}, \boldsymbol{E}'\}$

Initialize $\boldsymbol{A}' \leftarrow \boldsymbol{A}$, $\boldsymbol{E}' \leftarrow \boldsymbol{E}$;
for $i \leftarrow 1$ **to** b **do**
  $max\_loss \leftarrow -\infty$;
  for *each controlled node* u *in* $\mathcal{G}$ **do**
    for *each possible destination node* $v \neq u$ **do**
      Simulate adding edge (u, v) to $\boldsymbol{A}'$, with corresponding edge features $\boldsymbol{E}'_{(u,v)}$;
      Compute loss: $current\_loss \leftarrow \ell(\mathcal{M}(\boldsymbol{A}', \boldsymbol{X}, \boldsymbol{E}'), \boldsymbol{Y})$;
      if $current\_loss > max\_loss$ **then**
        $max\_loss \leftarrow current\_loss$;
        $best\_edge \leftarrow (u, v)$;
  Add $best\_edge$ to A', update $\boldsymbol{E}'$ with its features;
return $\mathcal{G}' = \{\boldsymbol{A}', \boldsymbol{X}, \boldsymbol{E}'\}$;



Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

- Questions remaining to answer
- We focus on activating an excessively large fase alarm rate as the attack goal, but can we also generate evasive samples (increasing the false negative rate ?)
  - Adding benign traffic flows targeting at one IP address in the dataset ?
- Divide the raw network traffics according to their temporal orders.
  - Attackers add adversarial manipulations to historical data, while aiming to deliver adversarial attack impacts over future traffic flows
- The impact of the degree of the target IP address
- Transferability of the attack
  - If an adversarial attack can mislead GNN-based intrusion detection systems, can it also mislead a flow-statistic based intrusion detection model (or not) ?
  - Think about other GNN models

Problem Space-constrained Adversarial Attack against Graph-based Intrusion Detection

- Next step
- Elevating both false alarm and false negative rate via attack
- Where to add edges and what edges to add ?
- Transferable attacks across both the graph-based IDS and statistics-based IDS
- Publication plan:
  - Matthieu Mouzaoui at the second year of this thesis.
  - 2 papers submitted by the end of this year.

Thanks !