

Robustness Evaluation of Graph Neural Network-based Network Intrusion Detection Systems against Adversarial Flow Injection

M. Mouzaoui¹ Y. Han¹ G. Blanc² G. Rilling³ M. Hurfin¹

¹Inria, Univ. Rennes ²Télécom SudParis, IPP ³CEA, LIST, Univ. Paris-Saclay

SECRYPT 2026



Supported by the French National Research Agency (ANR) under France 2030 (SuperViz ANR-22-PECY-0008)

GNN-based NIDS: Graph Representation

- **Flow node** (V_{fl}): one per network flow, with feature vector
- **Endpoint node** (V_{ep}): one per IP:Port pair
- **Connection**: each flow connects to its source and destination endpoints

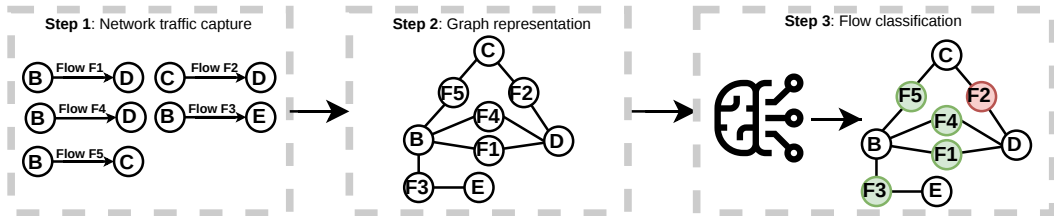


Figure: GNN-based NIDS pipeline. The GNN classifies each flow by aggregating information from its neighbourhood $\Rightarrow$ captures inter-flow dependencies.

Adversarial Attacks on GNNs

Graph structure is discrete

Edges either exist or not $\Rightarrow$ no continuous gradient over structure $\Rightarrow$ **greedy search**, not gradient descent.

Feature-based attacks

Change flow attributes (bytes, duration, packet count. . .)

Structure-based attacks

Add or remove flows in the graph

Why feature attacks fail for NIDS

Changing flow *features* could create physically invalid traffic¹
 $\Rightarrow$ Only **structural attacks** (injecting flows) are realistic.

¹Pierazzi et al. 2020; Merzouk et al. 2022; Vitorino et al. 2023.

Why Existing GNN Attacks Don't Fit NIDS

White-box access

Need model gradients or weights.^a

NIDS reality: model is not disclosed.

^aZügner et al. 2018.

Edit existing flows

Modify or delete flows in the graph.^b

NIDS reality: No access to historical data — **injection only**.

^bBojchevski and Günnemann 2019.

Ignore reachability

Inject to any endpoint.^c

NIDS reality: destination must have an open service port.

^cVenturi et al. 2024.

Realistic attack constraints:

Realistic flow injection: No model parameters, injection-only, endpoint reachability.

Threat Model

Capabilities:

- Controls **one external endpoint** v_a
- Injects new flow nodes toward reachable destinations
 - **Injection-only**: cannot remove or modify existing flows
 - **Reachability**: target must have open service ports

Knowledge:

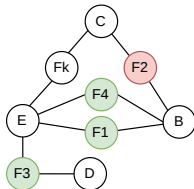
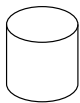
- **Black-box**: no model weights or gradients
- **Soft-label**: logit query outputs only
- Statistical patterns of benign/malicious traffic

2 distinct objectives:

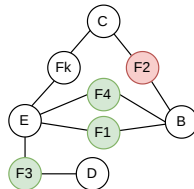
- **Evasion**: misclassify a malicious flow as benign by injecting one or several flows
- **DoS**: raise the False Positive Rate

Attack Method

Step 1: Model creation

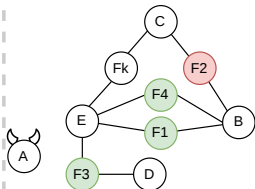


(b) Graph creation

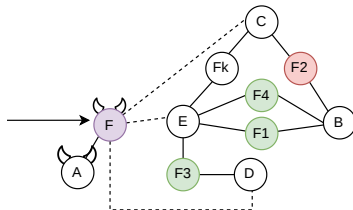


(d) Model in inference mode

Step 2: Adversarial attacks



(e) Attacker endpoint addition



(f) Flow injection, greedy search

TP	FN
FP	TN

(g) FPR increase



(h) Flow misclassification

DoS:

- **FPR** (False Positive Rate): ratio of benign flows misclassified as malicious out of all benign flows
- FPR: 0.042 $\rightarrow$ 0.043 with 3,000 injected flows $\Rightarrow$ **Infeasible** in practice

Evasion:

- **ASR** (Attack Success Rate): ratio of malicious flows evading detection out of all tried malicious flows
- **84.5% ASR** (GraphSAGE-2L, X-IloTID), median **6 injected flows** to succeed

Key observation

Rapid gain in the first few injections, then a plateau

RQ2 – What Drives Attack Cost?

- Higher degree $\Rightarrow$ more flows to dilute
- Malicious purity $> 50\%$
 $\Rightarrow 2\times$ more flows needed
- 91.2% of attacks target the **victim** endpoint
(receiving the initial malicious flow)

Degree	Mean	Median
= 1	5.4	4
= 2	7.0	5
3–5	11.9	4
6–50	45.0	45
> 50	25.2	11

Table RQ2: Number of required cover flows by **endpoint degree**, against GraphSAGE-2L on X-IIoTID.

Key observation

Local structure determines attack difficulty: endpoint degree and neighbourhood purity predict attack cost.

RQ3 & RQ4 – Stealth and Transferability

Side effects (RQ3): no label flips outside target, within **1–2 hops** of victim — stealthy and local.

Transferability (RQ4): strong to **GCN** (similar aggregation).

Layers	GraphSAGE		GCN		GAT		GIN	
	2	4	2	4	2	4	2	4
Attack Success Rate	—	0.905	0.872	0.384	0.113	0.357	0.143	0.317

Table RQ4: Cross-architecture ASR at fixed injection budget K — attack designed against **GraphSAGE-2L**

Key observation

Transfer depends on **aggregation similarity** to the surrogate (GraphSAGE = mean pooling).

Conclusion & Open Problems

Findings:

- **84.5% evasion ASR**, median 6 flows, realistic constraints
- DoS effectively **infeasible**: benign communities resist
- Attack cost driven by **local structure** (endpoint degree + neighbourhood purity)
- **Stealthy** (1–2 hop locality) and **transferable** across architectures

Open Problems:

- **Hard-label setting**: logit access is not always realistic
⇒ attacker only sees accept/drop
- Joint topology + feature optimisation
- Temporal graph dynamics
- **Robust architectures** by design: bound individual node influence

Thank you!

Questions?

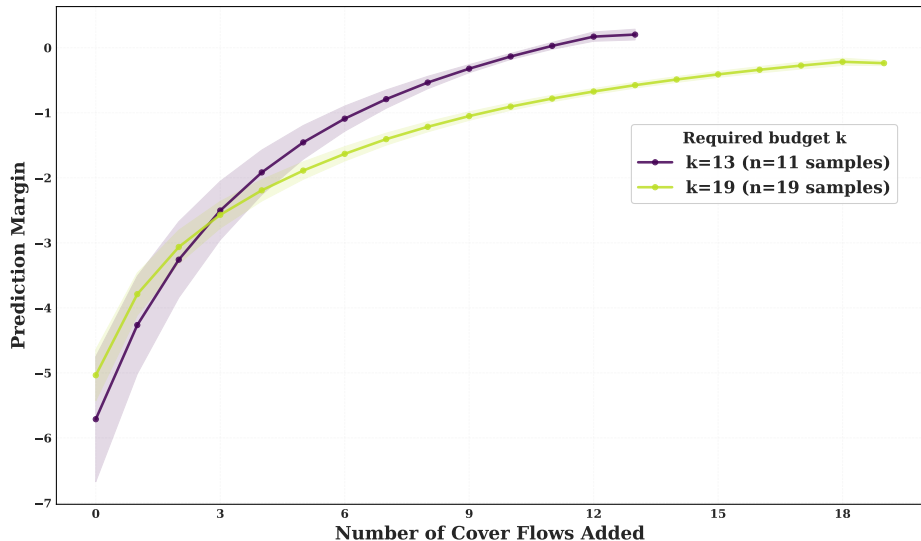
`matthieu.mouzaoui@inria.fr`

References I

-  Bojchevski, Aleksandar and Stephan Günnemann (2019). “Adversarial Attacks on Node Embeddings via Graph Poisoning”. In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR, pp. 695–704.
-  Merzouk, Mohamed Amine, Frédéric Cuppens, Nora Boulahia-Cuppens, and Reda Yaich (2022). “Investigating the Practicality of Adversarial Evasion Attacks on Network Intrusion Detection”. In: *Annals of Telecommunications* 77.11–12, pp. 763–775.
-  Pierazzi, Fabio, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro (2020). “Intriguing Properties of Adversarial ML Attacks in the Problem Space”. In: *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1332–1349. DOI: 10.1109/SP40000.2020.00073.
-  Venturi, Andrea, Dario Stabili, and Mirco Marchetti (2024). “Problem space structural adversarial attacks for network intrusion detection systems based on graph neural networks”. In: *arXiv preprint arXiv:2403.11830*.

-  Vitorino, João, Isabel Praça, and Eva Maia (2023). “SoK: Realistic adversarial attacks and defenses for intelligent network intrusion detection”. In: *Computers & Security* 134, p. 103433. ISSN: 0167-4048. DOI: <https://doi.org/10.1016/j.cose.2023.103433>. URL: <https://www.sciencedirect.com/science/article/pii/S0167404823003437>.
-  Zügner, Daniel, Amir Akbarnejad, and Stephan Günnemann (2018). “Adversarial Attacks on Neural Networks for Graph Data”. In: *SIGKDD*, pp. 2847–2856.

Backup – RQ1: Evasion Margin Trajectory (Figure 3)



Backup – Greedy Flow Injection (Algorithm 1)

Input: graph G , model f_θ , budget K , attack mode

- 1 Add attacker endpoint v_a ; for evasion, also inject attack flow f_{ab} with malicious features
- 2 **For** $t = 1, \dots, K$:
 - 1 For each reachable endpoint v : temporarily inject cover flow ($v_a \rightarrow v$), query f_θ
 - 2 Compute marginal gain $\Delta \ell_{\text{atk}}$; keep best endpoint v^*
 - 3 For evasion: early stop if prediction margin $\delta_{v_{ab}} > 0$

Theoretical guarantee

Objective is **weakly submodular** [Wang et al., 2020]: greedy achieves $(1 - e^{-\gamma})$ -approximation of the optimum in $\mathcal{O}(K \cdot |\mathcal{V}_{ep}^{dst}|)$ model queries.

Backup – Transferability across Representations (Table 3)

ASR (%) for attacks designed against heterogeneous GraphSAGE-2L,
evaluated on E-GraphSAGE targets (X-IloTID)

Target model	K	$K+1$	$K+2$
E-GraphSAGE (2 layers)	86.6	90.2	90.2
E-GraphSAGE (4 layers)	87.2	91.2	91.1

K = minimum flows needed on the surrogate. Attack transfers well across representations.

Backup – Full Transferability Results (Table 2)

ASR and TPR@FPR for attacks designed against GraphSAGE-2L, evaluated on target architectures (X-IIoTID)

Target	L	K	K+1	K+2	TPR@FPR
GAT	2	0.113	0.113	0.119	0.963
GAT	4	0.357	0.479	0.642	0.601
GCN	2	0.872	0.930	0.948	0.953
GCN	4	0.384	0.479	0.633	0.956
GIN	2	0.143	0.216	0.275	0.983
GIN	4	0.317	0.463	0.581	0.984
GraphSAGE	4	0.905	0.942	0.942	0.972

Backup – Degree Analysis: Full Table (Table 1)

Required cover flows by endpoint degree — GraphSAGE-2L, X-IloTID

Degree Group	n	Mean	Median	Std Dev
$deg = 1$	54	5.37	4.0	6.24
$deg = 2$	118	7.03	5.0	6.41
$deg = 3-5$	96	11.89	4.0	13.15
$deg = 6-50$	45	45.04	45.0	31.12
$deg > 50$	15	25.20	11.0	34.20

91.2% of attacks target the endpoint **directly connected** to the victim flow.

Backup – Attack Formulation

C1: Additive-only

Cannot modify or remove existing flows — preserves temporal consistency.

C2: Endpoint reachability

Injections only toward destinations with reachable service ports.

Optimization:

$$G^* = \arg \max_{\substack{G' \text{ satisfies} \\ \text{C1, C2}}} \ell_{\text{atk}}(G')$$

Evasion

$$\ell_{\text{evasion}} = z_{ab}[0] - z_{ab}[1]$$

Maximize benign margin of attack flow f_{ab} .

DoS

$$\ell_{\text{DoS}} = \sum_{j \in \mathcal{B}_{\text{test}}} (z_j[1] - z_j[0])$$

Maximize malicious logit of benign flows.

NP-hard

Combinatorial problem $\Rightarrow$ greedy approximation (Algorithm 1). Objective is weakly submodular $\Rightarrow$ $(1 - e^{-\gamma})$ -approximation guarantee.

Heterogeneous graph: $G = (V_{fl} \cup V_{ep}, \mathcal{E})$

- V_{fl} : flow nodes with feature vectors
- V_{ep} : endpoint (IP:Port) nodes
- Each flow connects to source & destination endpoint

Message passing (relation $\beta \rightarrow \alpha$, layer l):

$$m_{\mathcal{N}(a)}^{(l+1)} = \text{AGGR}\{h_b^{(l)}, \forall b \in \mathcal{N}(a)\}$$
$$h_a^{(l+1)} = \sigma\left(W_{\beta \rightarrow \alpha}^{(l)} \text{CONCAT}\left(h_a^{(l)}, m_{\mathcal{N}(a)}^{(l+1)}\right)\right)$$

Prediction margin (binary):

$$\delta_i = z_i[0] - z_i[1]$$

$\delta_i > 0$: benign $\delta_i < 0$: malicious

Aggregation (AGGR) by architecture:

- GraphSAGE, GCN: mean pooling
- GAT: weighted attention
- GIN: weighted sum

After L layers, each flow embedding is passed to an MLP classifier returning logits $\mathbf{z}$.

GNN: Message Passing

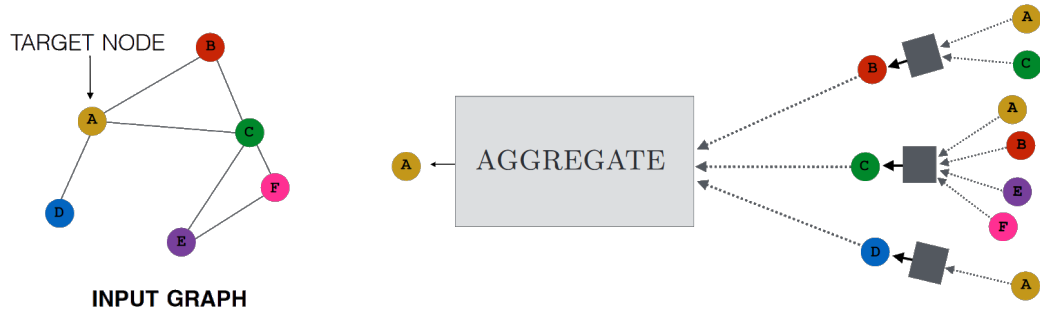


Figure: Graph Neural Network, aggregation.