

Breaking GNN-based NIDS via Hard-Label Black-Box Flow Injection Attacks

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

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

PTCC, matinée IA & Cybersécurité



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

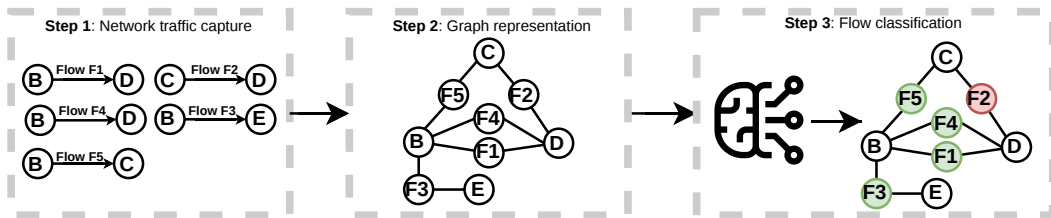


Figure: GNN-based NIDS pipeline. The GNN classifies each flow by aggregating information from its neighbourhood $\Rightarrow$ captures inter-flow dependencies.

GNN: Message Passing

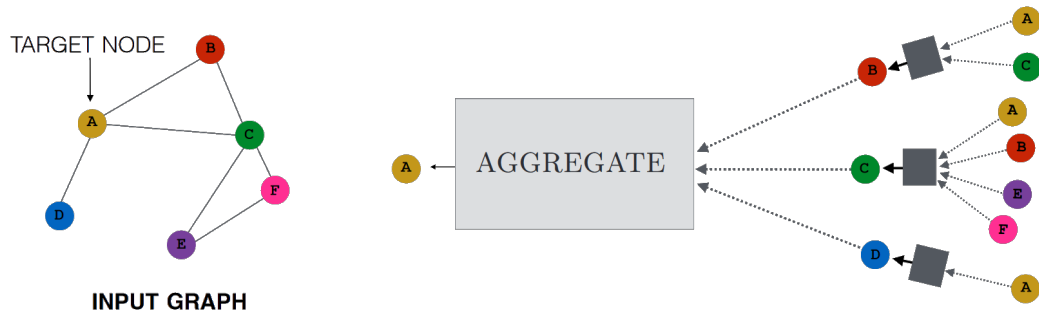


Figure: Graph Neural Network, aggregation. From¹

¹Hamilton.

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 (**cover flows**) or remove flows in the graph

Why feature attacks fail for NIDS

Changing flow *features* could create physically invalid traffic²
 $\Rightarrow$ Only **structural attacks** (injecting flows) are realistic.

²Pierazzi et al. 2020; Merzouk et al. 2022; Vitorino et al. 2023.

Why Existing GNN Attacks Don't Fit NIDS

White-box access

Need gradients or weights.^{abc}
NIDS reality: model is not disclosed.

^aZügner et al. 2018.

^bBojchevski and Günnemann 2019.

^cChen et al. 2020.

Edit existing flows

Modify or delete existing flows.^d
NIDS reality: No access to historical data: **injection only**.

^dBojchevski and Günnemann 2019.

Ignore reachability

Inject to any endpoint.^e
NIDS reality: destination must have an open service port.

^eVenturi et al. 2024.

Realistic constraints: no model parameters, injection-only, reachability

- Many black-box attacks³ rely on **soft labels**; production NIDS expose **hard labels** only (accept/drop).
- Repetitive near-identical queries can be detected and blocked⁴.

³Sheatsley et al. 2022; Wen et al. 2024; Mouzaoui et al. 2026.

⁴Mu et al. 2021.

Threat Model

Capabilities:

- Controls **one external endpoint** v_a
- Injects new nodes (cover flows) toward reachable destinations
 - **Injection-only**: cannot remove or modify existing flows
 - **Reachability**: target must have open service ports
- Queries the NIDS as an **oracle**
(one query per flow)

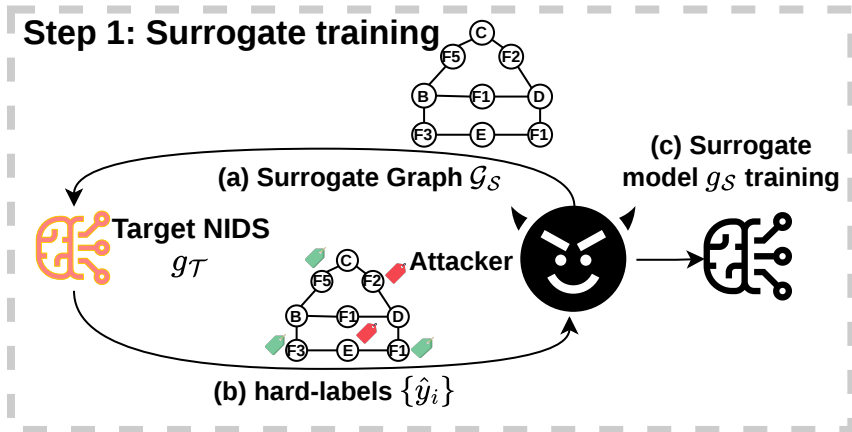
Knowledge:

- **Black-box**: no model weights or gradients
- **Hard-label**: accepts or drops flows, no confidence scores
- Knows which destination endpoints are reachable

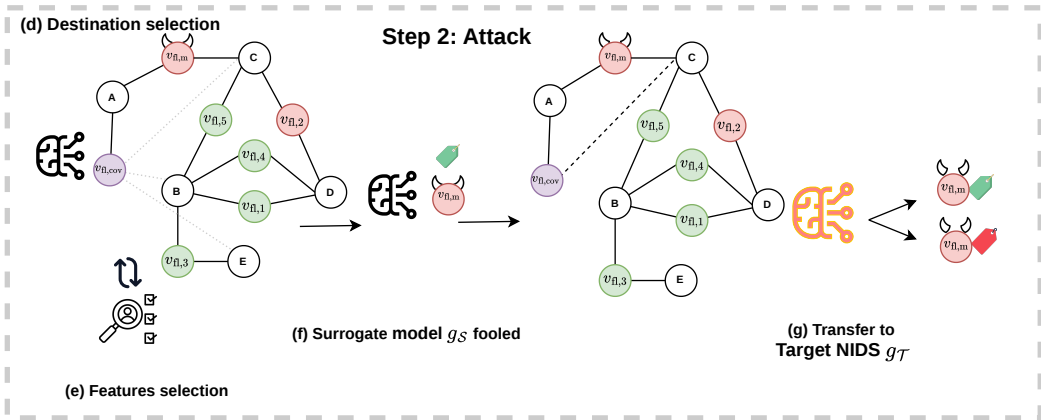
Objective: targeted evasion

Make the target NIDS g_T misclassify a specific malicious flow $v_{fl,m}$ as benign, by injecting a sequence of **cover flows**.

Attack Method (Step 1)



Attack Method (Step 2)



Flow Injection Graph Attack (FIGA) main steps

(d) Destination selection

$$v_{ep}^{(t)} = \arg \max_{v \in \mathcal{V}_{ep}^{dst}} \left\| \frac{\partial \ell}{\partial \mathbf{h}_v} \right\|$$

Pick the reachable endpoint whose embedding has the largest influence on the evasion objective.

ℓ = attacker's evasion objective

(e) Feature selection

$$\mathbf{x}_{fl}^{(t)} = \arg \max_{\mathbf{x}_{fl} \in \mathcal{X}_{benign}} \text{Cosine} \left(\frac{\partial \ell}{\partial \mathbf{x}_{fl}}, \mathbf{x}_{fl} \right)$$

Pick the flow in the pool that is best aligned with the gradient direction.

Computation

Both computed on the surrogate model.

Attack Effectiveness (CIC-IDS2017)

Target	Queries	Surrogate					
		SAGE-4L		GCN-4L		GAT-4L	
		ASR _t	F1 _t	ASR _t	F1 _t	ASR _t	F1 _t
SAGE-4L	10 ²	0.870	0.230	0.710	0.450	0.740	0.413
SAGE-4L	10 ³	0.990	0.020	0.990	0.020	0.969	0.059
SAGE-4L	10 ⁴	1.000	0.000	1.000	0.000	1.000	0.000
SAGE-4L	2 × 10 ⁴	1.000	0.000	1.000	0.000	1.000	0.000
GCN-4L	10 ²	0.990	0.020	0.990	0.020	0.970	0.039
GCN-4L	10 ³	0.990	0.020	0.990	0.020	0.970	0.039
GCN-4L	10 ⁴	0.990	0.020	0.990	0.020	0.990	0.020
GCN-4L	2 × 10 ⁴	0.990	0.020	0.990	0.020	0.990	0.020
GAT-4L	10 ²	1.000	0.000	1.000	0.000	0.720	0.438
GAT-4L	10 ³	1.000	0.000	1.000	0.000	0.980	0.039
GAT-4L	10 ⁴	1.000	0.000	1.000	0.000	1.000	0.000
GAT-4L	2 × 10 ⁴	1.000	0.000	1.000	0.000	1.000	0.000

Baselines (Random, Hard-GNN⁵): **ASR_t** = 0.000 / **F1_t** = 1.000 at all budgets.

⁵Mu et al. 2021.

Table: Attack Success Rate, F1-score on CIC-IDS2017 dataset, $n = 100$ injected cover flows.

Attack Effectiveness (X-IIoTID)

Table: Attack Success Rate, F1-score on X-IIoTID dataset, $n = 100$ injected cover flows.

Target	Queries	Surrogate					
		SAGE-4L		GCN-4L		GAT-4L	
		ASR _t	F1 _t	ASR _t	F1 _t	ASR _t	F1 _t
SAGE-4L	10 ²	0.450	0.710	0.939	0.115	0.620	0.551
SAGE-4L	10 ³	0.530	0.640	0.951	0.093	0.870	0.230
SAGE-4L	10 ⁴	0.813	0.320	0.950	0.095	0.890	0.198
SAGE-4L	2 × 10 ⁴	0.940	0.113	0.951	0.093	0.890	0.198
GCN-4L	10 ²	0.280	0.837	1.000	0.000	0.660	0.508
GCN-4L	10 ³	0.320	0.810	1.000	0.000	0.690	0.473
GCN-4L	10 ⁴	0.320	0.810	1.000	0.000	0.750	0.400
GCN-4L	2 × 10 ⁴	0.660	0.508	1.000	0.000	0.750	0.400
GAT-4L	10 ²	0.780	0.361	0.910	0.165	0.650	0.519
GAT-4L	10 ³	0.620	0.551	0.910	0.165	0.620	0.551
GAT-4L	10 ⁴	0.840	0.276	0.910	0.165	0.810	0.319
GAT-4L	2 × 10 ⁴	0.910	0.165	0.910	0.165	0.870	0.230

Baselines (Random, Hard-GNN⁶): **ASR_t** = 0.000 / **F1_t** = 1.000 at all budgets.

⁶Mu et al. 2021.

Does Feature Selection Matter?

<i>n</i> injected cover flows	FIGA		FIGA-Random	
	ASR_t↑	F1_t↓	ASR_t↑	F1_t↓
2	0.763	0.384	0.224	0.874
3	0.813	0.315	0.432	0.724
4	0.841	0.275	0.433	0.724
5	0.947	0.101	0.549	0.622
25	0.992	0.015	0.375	0.769
50	0.952	0.092	0.605	0.567
100	0.957	0.082	0.660	0.508

dataset = CIC-IDS2017, 20,000 queries, pool of 15 candidate flows, target SAGE-2L.

Key observation

Gradient-guided selection beats random sampling

Transferability Across Architectures

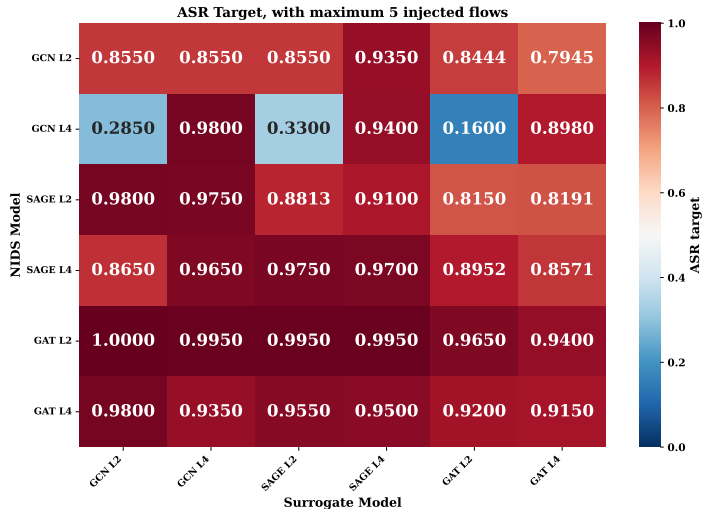


Figure: Attack Success Rate, on CIC-IDS2017 dataset

Transferability Across Architectures (X-IloTID)

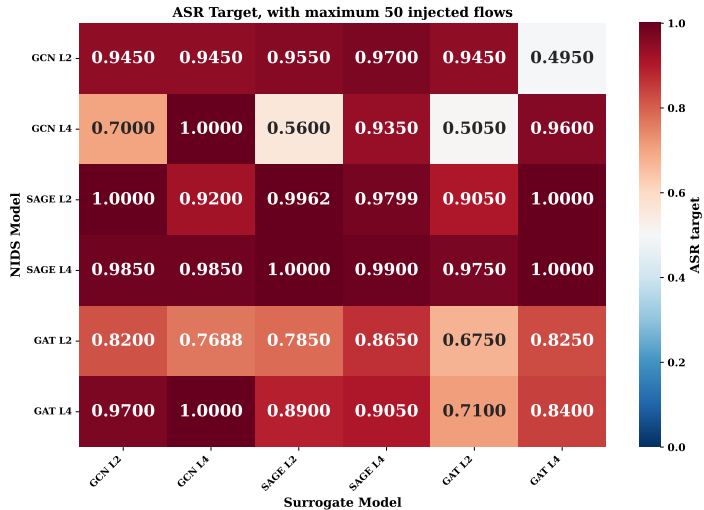


Figure: X-IloTID: ASR_t , max. 50 cover flow nodes.

Findings

- Practical hard-label, black-box flow injection attack (**FIGA**) still effective⁷
- Influence-guided search is outperforming other methods in a tight budget
- Strong transferability & ASR across diverse GNN architectures

Open Problems

- Evaluation limited to two benchmarks
- Robustification: context-aware defenses are needed, such as adversarial training with structural perturbations

⁷Code: github.com/MouzaouiMatthieu/FIGA-Flow-Injection-Graph-Attacks

Merci !

Questions ?

`matthieu.mouzaoui@inria.fr`

-  Al-Hawawreh, Muna, Elena Sitnikova, and Neda Aboutorab (Mar. 2022). “X-IloTID: A Connectivity-Agnostic and Device-Agnostic Intrusion Data Set for Industrial Internet of Things”. In: *IEEE Internet of Things Journal* 9.5, pp. 3962–3977. ISSN: 2327-4662. DOI: 10.1109/JIOT.2021.3102056. (Visited on 04/21/2026).
-  Bojchevski, Aleksandar and Stephan Günnemann (June 2019). “Adversarial Attacks on Node Embeddings via Graph Poisoning”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 695–704.

References II

-  Chen, Jinyin, Huiling Xu, Jinhuan Wang, Qi Xuan, and Xuhong Zhang (2020). “Adversarial Detection on Graph Structured Data”. In: *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*. PPMLP’20. New York, NY, USA: Association for Computing Machinery, pp. 37–41. ISBN: 978-1-4503-8088-1. DOI: 10.1145/3411501.3419424.
-  Lanvin, Maxime, Pierre-François Gimenez, Yufei Han, Frédéric Majorczyk, Ludovic Mé, and Éric Total (2023). “Errors in the CICIDS2017 Dataset and the Significant Differences in Detection Performances It Makes”. In: *Risks and Security of Internet and Systems*. Ed. by Slim Kallel, Mohamed Jmaiel, Mohammad Zulkernine, Ahmed Hadj Kacem, Frédéric Cuppens, and Nora Cuppens. Cham: Springer Nature Switzerland, pp. 18–33. ISBN: 978-3-031-31108-6.

-  Merzouk, Mohamed Amine, Frédéric Cuppens, Nora Boulahia-Cuppens, and Reda Yaich (Dec. 2022). “Investigating the Practicality of Adversarial Evasion Attacks on Network Intrusion Detection”. In: *Annals of Telecommunications* 77.11-12, pp. 763–775. ISSN: 0003-4347, 1958-9395. DOI: 10.1007/s12243-022-00910-1. (Visited on 02/12/2024).
-  Mouzaoui, Matthieu, Yufei Han, Gregory Blanc, Gabriel Rilling, and Michel Hurfin (Aug. 2026). “Robustness Evaluation of Graph Neural Network-Based Network Intrusion Detection Systems against Adversarial Flow Injection”. In: *23rd International Conference on Security and Cryptography*, pp. 860–867. ISBN: 978-989-758-858-7. (Visited on 08/24/2026).

References IV

-  Mu, Jiaming, Binghui Wang, Qi Li, Kun Sun, Mingwei Xu, and Zhuotao Liu (2021). “A Hard Label Black-Box Adversarial Attack against Graph Neural Networks”. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. Ccs '21. New York, NY, USA: Association for Computing Machinery, pp. 108–125. ISBN: 978-1-4503-8454-4.
-  Pierazzi, Fabio, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro (2020). “Intriguing Properties of Adversarial ML Attacks in the Problem Space”. In: *Proceedings of IEEE SP oakland, 2020*, pp. 1332–1349. DOI: 10.1109/SP40000.2020.00073.
-  Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani (2018). “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”. In: *International Conference on Information Systems Security and Privacy*.

-  Sheatsley, Ryan, Nicolas Papernot, Michael J. Weisman, Gunjan Verma, and Patrick McDaniel (Jan. 2022). “Adversarial examples for network intrusion detection systems”. In: *Journal of Computer Security* 30.5, pp. 727–752. ISSN: 0926-227X.
-  Venturi, Andrea, Dario Stabili, and Mirco Marchetti (Mar. 2024). *Problem Space Structural Adversarial Attacks for Network Intrusion Detection Systems Based on Graph Neural Networks*. DOI: 10.48550/arXiv.2403.11830. arXiv: 2403.11830 [cs]. (Visited on 03/20/2024).
-  Vitorino, João, Isabel Praça, and Eva Maia (Aug. 2023). “Towards Adversarial Realism and Robust Learning for IoT Intrusion Detection and Classification”. In: *Annals of Telecommunications* 78.7-8, pp. 401–412. ISSN: 0003-4347, 1958-9395. DOI: 10.1007/s12243-023-00953-y. arXiv: 2301.13122 [cs]. (Visited on 02/12/2024).

-  Wen, Liangliang, Jiye Liang, Kaixuan Yao, and Zhiqiang Wang (Oct. 2024). “Black-Box Adversarial Attack on Graph Neural Networks With Node Voting Mechanism”. In: *IEEE Transactions on Knowledge and Data Engineering* 36.10, pp. 5025–5038. ISSN: 1041-4347. DOI: 10.1109/TKDE.2024.3380750. (Visited on 05/20/2026).
-  Zügner, Daniel, Amir Akbarnejad, and Stephan Günnemann (2018). “Adversarial Attacks on Neural Networks for Graph Data”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Kdd '18. New York, NY, USA: Association for Computing Machinery, pp. 2847–2856. ISBN: 978-1-4503-5552-0. DOI: 10.1145/3219819.3220078.

Datasets

Experiments are conducted on the CIC-IDS2017 dataset (with label corrections)⁸ as well as X-IIoTID⁹. We partitioned the flow sets using a temporal-order-aware split.

Table: Each dataset is partitioned into three non-overlapped sets of network flows. We keep 50% of the flows for the NIDS training, and the last 20% for $\mathcal{G}_{\mathcal{T}}$

Dataset	Split	# Endpoints	# Flows	# Benign	# Malicious
CICIDS-2017	$\mathcal{G}_{\text{train}}$	167134	429424	195949	233475
	$\mathcal{G}_{\mathcal{S}}$	119787	257652	117569	140083
	$\mathcal{G}_{\mathcal{T}}$	92195	171779	78380	93399
X-IIoTID	$\mathcal{G}_{\text{train}}$	179785	299788	169406	130382
	$\mathcal{G}_{\mathcal{S}}$	124987	179701	100724	78977
	$\mathcal{G}_{\mathcal{T}}$	90297	117453	67065	50388

⁸Sharafaldin et al. 2018; Lanvin et al. 2023.

⁹Al-Hawawreh et al. 2022.

Backup – Algorithm 1: FIGA Pipeline

Require: target $g_{\mathcal{T}}$; surrogate graph $\mathcal{G}_S$; valid endpoint pool $\mathcal{V}_{ep}^{dst}$; feature pool $\mathcal{X}_{fl}$; budget N ; objective ℓ

1 Phase 1 – Offline surrogate training

1 $Y_S \leftarrow$ query $g_{\mathcal{T}}$ with Q flows in $\mathcal{G}_S$ to obtain hard labels

2 Train local surrogate GNN g_S using $(\mathcal{G}_S, Y_S)$

2 Phase 2 – Influence-guided flow injection, for $t = 1, \dots, N$:

1 *Structural selection:* $v_{ep}^{(t)} \leftarrow \arg \max_{v \in \mathcal{V}_{ep}^{dst}} \left\| \frac{\partial \ell}{\partial \mathbf{h}_v} \right\|$

2 *Feature selection:* $\mathbf{d} \leftarrow \frac{\partial \ell}{\partial \mathbf{x}_{fl}}$, then $\mathbf{x}_{fl}^{(t)} \leftarrow \arg \max_{\mathbf{x}_{fl} \in \mathcal{X}_{benign}} \text{Cosine}(\mathbf{d}, \mathbf{x}_{fl})$

3 Inject: $\mathcal{G}^{(t)} \leftarrow \mathcal{G}^{(t-1)} \oplus (v_{ep}^{(t)}, \mathbf{x}_{fl}^{(t)})$

4 Early stop if $\ell > 0.5$ on g_S (stealth)

3 Return $\mathcal{G}_{adv} \leftarrow \mathcal{G}^{(t)}$

Theoretical guarantee

Objective is weakly submodular $\Rightarrow$ greedy achieves a $(1 - e^{-\gamma})$ -approximation of the optimum in $\mathcal{O}(N \cdot |\mathcal{X}_{fl}^{pool}| \cdot C_{inf})$.

Backup – Theoretical Guarantees

Theorem 1 – Weak Submodularity of FIGA

Under additive-only + endpoint-reachability constraints (and standard Lipschitz / curvature / gradient-alignment assumptions), maximising the surrogate evasion objective ℓ_{sur} is a γ_{sur} -weakly submodular problem. Moreover

$$\ell_{tar}(m(\mathcal{S}_{inj})) \geq \ell_{sur}(m(\mathcal{S}_{inj})) - N\epsilon,$$

so maximising the surrogate loss lower-bounds the target's evasion loss.

Corollary 1 – Optimality Guarantee

The influence-guided greedy search (Algorithm 1) gives a $(1 - e^{-\gamma_{sur}})$ -approximation of the optimal injected set $\mathcal{O}_{sur}^*$, converging in polynomial time $\mathcal{O}(N \cdot |\mathcal{X}_{fl}^{pool}| \cdot C_{inf})$.

Proposition 1 – Query Complexity of Baselines

Any hard-label, query-based (surrogate-free) zeroth-order attack needs at least

$$Q^{(t)} \geq \Omega(|\mathcal{V}_{ep}^{dst}| \cdot |\mathcal{X}_{val}|)$$

queries per greedy step to identify the best injection with constant accuracy.

Backup – Graph Representation & Message Passing

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 $\mathbf{z}$.