



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

Matthieu Mouzaoui, Yufei Han, Gregory Blanc, Gabriel Rilling, Michel Hurfin

► To cite this version:

Matthieu Mouzaoui, Yufei Han, Gregory Blanc, Gabriel Rilling, Michel Hurfin. Breaking GNN-based NIDS via Hard-Label Black-Box Flow Injection Attacks. ESORICS 2026 - 31st European Symposium on Research in Computer Security, Sep 2026, Rome, Italy. <hal-05712664>

HAL Id: hal-05712664

<https://hal.science/hal-05712664v1>

Submitted on 6 Aug 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

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

Matthieu Mouzaoui¹[0009-0006-9104-9230], Yufei Han¹[0000-0002-9035-6718],
Gregory Blanc³[0000-0001-8150-6617], Gabriel Rilling²[0009-0007-6111-1045], and
Michel Hurfin¹[0009-0006-6695-2938]

¹ Inria, Univ. Rennes

² CEA, List, Université Paris-Saclay

³ SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris

Abstract. While Graph Neural Network (GNN)-based Network Intrusion Detection Systems (NIDS) excel by encoding relational traffic structures, this structural dependency introduces novel adversarial vulnerabilities. Yet, existing graph attacks remain impractical, often demanding white-box access, prohibitive query budgets, or exploit-breaking payload manipulations. To overcome these limitations, we propose a practical, hard-label black-box attack that evades detection strictly by injecting valid supplementary traffic flows, perfectly preserving the exploit’s functional integrity. By formulating this combinatorial challenge as a weakly submodular maximization problem, we leverage a local surrogate GNN and an influence-guided greedy search to efficiently identify highly impactful injections. Under standard assumptions on the evasion objective and the surrogate-target alignment, we establish a polynomial-time approximation guarantee for this approach. Empirically, we demonstrate that the proposed flow injection attack can bypass state-of-the-art GNN-based NIDS with minimal query budgets.

Keywords: GNN-based NIDS · Flow Injection Attack · Adversarial Attacks

1 Introduction

Graph Neural Network-based Network Intrusion Detection Systems (GNN-based NIDS) have emerged as the state-of-the-art solution for securing complex communication environments [37,39,34,11,8,22]. Unlike standard models that process traffic flows in isolation, such as Random Forest [10] or Support Vector Machines classifiers [5], GNNs leverage neighborhood aggregation to capture structural dependencies of network traffic, enabling detection of coordinated malicious behaviors such as lateral movement that routinely bypass traditional defenses.

However, the reliance on structural graph representations introduces a novel adversarial risk: by altering the graph’s relational structure, an attacker can manipulate neighborhood aggregation to induce misclassifications. Despite this vulnerability, existing graph-based evasion attacks remain impractical in realistic network environments due to three primary limitations (detailed further

in Section 3). **First**, many approaches rely on continuous feature manipulation (e.g., altering packet sizes or inter-arrival times), as commonly studied in adversarial attacks against GNNs [42]. This creates a severe “inverse-feature mapping” dilemma [25], where mathematical perturbations cannot be translated back into protocol-compliant traffic without breaking the exploit’s functional integrity. **Second**, prior works frequently assume unrealistic white-box access to model gradients or soft-label confidence scores [28,16,14], whereas production NIDS return only binary hard labels. **Third**, current black-box attacks [31,24] rely on unrealistic internal privileges to delete or modify flows between existing end-points or resort to query-based fuzzing [24] that demands massive, repetitive queries that are financially costly and easily flagged by anomaly detectors before convergence. Under *black-box* and *hard-label* constraints, effective attacks against GNN-based NIDS in realistic networks remain largely unexplored.

To overcome these fundamental bottlenecks, we introduce *FIGA* (Flow Injection Graph Attacks), a practical hard-label black-box framework that evades detection strictly through the injection of valid supplementary traffic. Beyond a concrete attack, we demonstrate that under realistic constraints, GNN-based NIDS exhibit a fundamental vulnerability to flow injection. Unlike prior methods that attempt to mutate the malicious payload itself, *FIGA* leaves the original exploit unchanged and instead manipulates the GNN’s spatial context by connecting adversary-controlled flows to the target. As detailed in Section 5, we show that this adversarial risk arises from the neighborhood aggregation mechanisms in GNN models, which underlie a wide spectrum of GNN-based NIDS. By leveraging a local surrogate model to approximate the target’s decision boundary, together with an influence-guided greedy search [33] to navigate the combinatorial space, *FIGA* efficiently identifies highly impactful structural perturbations with minimal query overhead and theoretical guarantees on attack effectiveness. Crucially, *FIGA* mirrors real-world traffic obfuscation tactics: adversaries routinely generate *cover traffic* during scanning or C2 communication [35], and the dynamic entropy of enterprise networks makes crafted injections indistinguishable from ambient traffic. This ensures the attack remains stealthy to both the target NIDS and traditional rate-limiting defences, providing a highly realistic framework for evaluating GNN-based NIDS robustness.

Our main technical contributions include the following perspectives:

Novel Attack Framework: We propose *FIGA*, a practical hard-label black-box evasion framework with proven attack effectiveness. Utilizing a dual-position strategy—leveraging a compromised internal node for offline surrogate training and an external endpoint for flow injection—*FIGA* elegantly circumvents the inverse-feature mapping problem, drastically reduces live query costs, and perfectly preserves exploit integrity.

Theoretical Guarantees: Theorem 1 and Corollary 1 establish that *FIGA*’s influence-guided search is a weakly submodular maximization problem with a

polynomial-time approximation guarantee. Proposition 1 proves that any query-based alternative requires a prohibitively large number of queries per step, making offline surrogate optimization a *necessary* design choice.

Empirical Efficacy & Transferability: Extensive evaluations on the CIC-IDS2017 [27,19] and X-IIoTID [1] datasets show FIGA’s transferability across architectures. With only 100 injected flows and 20000 queries, it achieves a 94.3% Attack Success Rate on average., while two black-box baselines achieves 0%.

2 Related Works

Graph Neural Network-based NIDS Unlike traditional ML models, which process network flows independently and ignore vital relational context such as shared endpoints in coordinated scanning campaigns, GNNs capture these structural dynamics by modeling communication data as graphs. Prior works explore multiple graph representations. Flow-graphs [20,22,8,3] representations encode flows as attributed edges connecting endpoint nodes, as in E-GraphSAGE [22], an extension of GraphSAGE [15] incorporating edge features for intrusion detection. Conversely, line-graphs [30,13] represent flows as attributed nodes, linking them when they share a source or destination endpoint. We adopt a **heterogeneous graph** representation [26], which unifies both formulations while balancing expressiveness and computational efficiency. In this formulation, the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ comprises two disjoint node sets: attributed flow nodes $\mathcal{V}_{\text{fl}}$ and endpoint nodes $\mathcal{V}_{\text{ep}}$ (identified by IP:Port pairs) with randomly initialised features, serving to provide structural context for message-passing. Unattributed edges strictly connect flow nodes to their respective source and destination endpoints.

For the task of flow classification, the GNN iteratively updates node embeddings through multi-hop neighborhood aggregation. Specifically, at layer l , the heterogeneous message passing for a relation $\beta \rightarrow \alpha$ (denoting either $\text{ep} \rightarrow \text{fl}$ or $\text{fl} \rightarrow \text{ep}$) between a target node a of type α and its neighbors $\mathcal{N}(a)$ of type β is defined as:

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

where σ is a non-linear activation function and $W_{\beta \rightarrow \alpha}^{(l)}$ denotes relation-specific learnable weight matrices. The aggregation function, $\text{AGGR}()$, depends on the underlying GNN architecture—such as mean-pooling in SAGE [15], weighted summation in GCN [18], or attention mechanisms in GAT [29]. After L propagation layers, the final embeddings of the flow-type nodes are passed to a Multi-Layer Perceptron (MLP) for classification. The model’s parameter set θ thus encapsulates all layer weights and MLP parameters.

Adversarial Attacks Against GNN-based NIDS Adversarial attacks on GNNs are broadly categorized into **feature attacks** [42], which modify node or

edge attributes, and **structural attacks** [2,7,4], which alter the graph topology. In the context of NIDS, manipulating flow features is challenging: continuous optimization over flow features often generates invalid features (e.g., negative byte-counts) or breaks the functionality of an exploit, creating an intractable inverse feature mapping problem [25,23,41,6].

Conversely, structural attacks via flow injection represent a realistic threat vector. Due to the message-passing nature of GNNs, an attacker can indirectly alter the classification of a target flow simply by modifying its structural neighbourhood [12]. However, to remain practically viable in real-world deployments, these structural perturbations require strict network constraints, such as ensuring injected flows are directed only towards valid destination endpoints with active listening ports.

White-box attacks against GNN models require access to the gradient or model parameters of the target GNNs [2,42,4]. In contrast, previous black-box attack methods [36,40,38,32,28,16,14] require the soft-label or classification loss of the target GNNs to guide graph perturbation. [36] assumes the attacker can execute message propagation first on the target graph to identify critical nodes, which is not applicable in the NIDS application. In a realistic **hard-label black-box** setting, the target GNN provides only binary decisions, making the soft-label methods inapplicable. Combinatorial searches for structural perturbations using fuzzing techniques [24,31] require intensive queries, creating a significant query bottleneck. Though [21] adopted language models to generate adversarial flow features in the hard-label setting, their method targets flow-based detection and cannot be applied to GNN-based NIDS. This limitation motivates a surrogate-based approach: the attacker trains a proxy model to approximate the target NIDS decision function, performs the computationally intensive gradient calculations and discrete objective evaluations entirely offline using the proxy model, and transfers the crafted adversarial examples to the target system.

3 Threat Model

This section formalizes the threat model for the adversarial flow-injection scenario evaluated throughout this work. To ensure practical relevance, we define an attacker targeting a GNN-based NIDS under strict black-box constraints, accurately reflecting production environments where internal model states and architectures are never exposed.

3.1 Attacker Knowledge

Following the hard-label black-box setting established in [24], the attacker possesses zero knowledge of the target NIDS’s parameters, architecture, or training data. Crucially, the target NIDS provides only **hard-label outputs** (e.g., detecting flows as benign or malicious) without exposing continuous confidence scores, probabilities, or gradients. This explicitly prevents standard white-box

optimization and score-based attack strategies. The attacker’s knowledge gathering proceeds in two distinct phases. The attacker first passively monitors a segment of the network’s background traffic, without interacting with the target NIDS, collecting real network flows that form the surrogate graph partition. In the surrogate partition labeling phase, the attacker submits its surrogate partition once to the NIDS and retrieves hard-label annotations. Crucially, the submitted network flows are real network traffic, not adversarially modified. This differs from query-based attacks that continuously submit near-identical perturbed graphs, which would be flagged by an anomaly detector. Finally, either through passive traffic sniffing or active reconnaissance before the attack, the attacker knows which **destination endpoints** (i.e., servers with active listening ports) exist within the targeted network.

3.2 Attacker Capabilities

The attacker controls one or more external endpoints that serve as origin points for injecting cover flows into the monitored network alongside the malicious attack flow. Furthermore, the attacker can query the target NIDS as an oracle by transmitting traffic and observing the NIDS’s response. Because aggregate flow features are typically finalised only upon connection termination, we assume the target NIDS evaluates these statistics after the flow naturally ends or reaches a predefined timeout. Consequently, the attacker transmits the complete intended flow and infers the system’s binary classification (the hard-label output) by monitoring subsequent automated defensive actions—such as the origin IP being added to a blocklist, subsequent connection attempts being silently dropped, or receiving post-analysis TCP resets (RST)—ensuring the statistical integrity of the evaluated flow remains intact.

3.3 Constraints on Attacker Actions

Additive-Only Modifications: Lacking core routing privileges, the attacker cannot remove, intercept, or modify existing historical or background traffic. They can only initiate new network flows. Mathematically, the original unperturbed graph remains a strict subgraph of the final adversarial graph ($\mathcal{G} \subseteq \mathcal{G}'$).

Endpoint Reachability: Injected flows must target endpoints with actively listening ports; otherwise, boundary firewalls would drop them. Thus, permissible destinations are strictly bounded to active services, denoted as $\mathcal{V}_{ep}^{dst} \subseteq \mathcal{V}_{ep}$.

Feature Validity: To bypass the inverse-feature mapping problem [25], the attacker does not mathematically mutate flow features. Instead, all injected flows must possess physically transmissible features sampled from a discrete pool of historically observed benign traffic, ensuring protocol compliance and evading elementary statistical sanitary checks.

3.4 Attacker Objective

Our work focuses on **Targeted Evasion**, in which an adversary injects benign network flows to deceive a GNN-based NIDS into misclassifying malicious traffic

as benign. Let $\mathcal{G} = (\mathcal{V}_f \cup \mathcal{V}_{ep}, \mathcal{E})$ be a heterogeneous graph where edges in $\mathcal{E}$ connect flow nodes ($\mathcal{V}_f$) to endpoint nodes ($\mathcal{V}_{ep}$). Edges exist only between flow and endpoint nodes, ensuring the GNN aggregates information strictly through endpoint-flow pathways in both directions. An attacker controlling an external endpoint $v_{ep,atk}$ who needs to generate a malicious flow $v_{f,m}$ against a victim $v_{ep,vic}$ first injects a set of **cover flows**, crafted to hide $v_{f,m}$ from the NIDS.

Attack Formulation Operating under black-box and hard-label constraints, the attacker optimizes this structural perturbation using an offline surrogate model g_S , and subsequently transfers the crafted cover flows to the targeted NIDS g_T . We assume the NIDS operates in time window-based mode, processing the relational structure of flows captured within each window. Let $g_S(v_f | \mathcal{G}) \in [0, 1]$ be the surrogate model’s predicted probability that a flow node v_f is malicious given a context defined by a graph $\mathcal{G}$. The final output of g_S for v_f is a *benign* label if $g_S(v_f | \mathcal{G}) < 0.5$ and a *malicious* label otherwise. To shift the surrogate’s prediction of the target malicious flow $v_{f,m}$ from *malicious* to *benign*, the attacker alters the graph $\mathcal{G}$ to increase the probability defined in 2:

$$\ell(\mathcal{G}) = 1 - g_S(v_{f,m} | \mathcal{G}) \quad (2)$$

Evasion succeeds on g_S when $\ell(\mathcal{G}) > 0.5$. The attack is executed iteratively starting from a graph $\mathcal{G}^{(0)}$, which represents the set of flows (malicious and background traffic) captured within the target NIDS’s current evaluation window. The attacker estimates the injection of cover flows into this local context to ensure that the graph of flows updated with the injected flows evades detection upon submission to the target GNN-based NIDS model. At step t , the attacker updates the graph $\mathcal{G}^{(t-1)}$ by adding a new cover flow $v_{f,c}^{(t)}$: $\mathcal{G}^{(t)} = \mathcal{G}^{(t-1)} \oplus v_{f,c}^{(t)}$. The cover flow $v_{f,c}(v_{ep}, \mathbf{x}_f)$ originates from the attacker’s endpoint $v_{ep,atk}$ and is fully parameterized by its destination $v_{ep} \in \mathcal{V}_{ep}^{dst}$. Its features $\mathbf{x}_f \in \mathcal{X}_{benign}$, where $\mathcal{X}_{benign}$ is a set of features from flows labelled as benign in the used datasets, described in Sect. 6.1. More realistically, the attacker could use features from the collected flows that are classified as benign by g_S . The pair $(v_{ep}^{(t)}, \mathbf{x}_f^{(t)})$ can be obtained using Eq. 3

$$(v_{ep}^{(t)}, \mathbf{x}_f^{(t)}) = \arg \max_{v_{ep} \in \mathcal{V}_{ep}^{dst}, \mathbf{x}_f \in \mathcal{X}_{benign}} \ell \left(\mathcal{G}^{(t-1)} \oplus v_{f,c}(v_{ep}, \mathbf{x}_f) \right). \quad (3)$$

The iterative injection continues until success ($\ell(\mathcal{G}^{(t)}) > 0.5$) or until a maximum budget N is reached, ensuring the perturbation remains stealthy and operationally minimal.

4 Attack Method

FIGA operates in two phases: *surrogate model training* and *iterative flow injection*. Its workflow is summarised in Figure 1 and Algorithm 1.

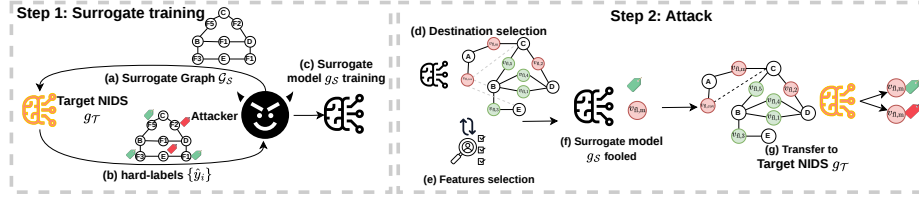


Fig. 1: FIGA pipeline. Phase 1: the attacker queries the target NIDS to obtain hard labels and trains an offline surrogate g_S . Phase 2: influence-guided flow injection is performed entirely on g_S and transferred to g_T .

4.1 Surrogate Model Training (Lines 1–2)

The first phase of the attack is the construction of a differentiable approximation of the target GNN g_T (**Lines 1–2** of Algorithm 1). The attacker trains a surrogate GNN g_S to approximate the decision boundaries of g_T on Q passively collected flows, submitting them once to g_T to obtain hard-label annotations Y_S used as ground truth. The transferability of the subsequent attack relies directly on the quality of this approximation, governed by the query budget Q . Its impact on transferability is analyzed in Section 6.2.

4.2 Iterative Flow Injection

With the surrogate model established, the attacker generates flows to inject offline, without interacting with the target NIDS, through a greedy iterative process. Since the GNN’s neighborhood aggregation dynamically shifts the latent embedding space after each structural modification, it is difficult to design a single bulk injection. Instead, at each step $t \in \{1, \dots, N\}$, the attacker evaluates the current graph state $\mathcal{G}^{(t-1)}$ to heuristically select and inject one more flow, progressively diluting the malicious signal of the flow $v_{fl,m}$ until the evasion objective is achieved or the maximum injection budget N is exhausted.

Endpoint Selection (Lines 5–6) To efficiently navigate the massive combinatorial search space of the graph topology, the attacker first identifies the most structurally vulnerable destination endpoint. By computing the gradient of the evasion objective ℓ with respect to the final embedding h_v of every valid endpoint $v \in \mathcal{V}_{ep}^{dst}$, the attacker isolates the specific neighborhood pathway that exerts the greatest mathematical influence over the target flow’s classification outcome:

$$v_{ep}^{(t)} = \arg \max_{v \in \mathcal{V}_{ep}^{dst}} \left\| \frac{\partial \ell}{\partial h_v} \right\| \quad (4)$$

Flow Feature Selection (Lines 8–9) Once the destination endpoint is chosen, the attacker determines the features of the injected flow (**Lines 8–9** of Algorithm

1). To this end, we compute the directional gradient of the objective ℓ with respect to the flow features (**Line 8**). The attacker then searches the candidate pool $\mathcal{X}_{benign}$ composed of the features of historically observed benign flows and selects the flow feature exhibiting the highest cosine similarity with this gradient (**Line 9**). Formally:

$$\mathbf{x}_{fl}^{(t)} = \arg \max_{\mathbf{x}_{fl} \in \mathcal{X}_{benign}} \left\{ \text{Cosine} \left(\frac{\partial \ell}{\partial \mathbf{x}_{fl}}, \mathbf{x}_{fl} \right) \right\} \quad (5)$$

This aims to increase the attack objective in the correct direction with realistic, benign flow features. Following injection (**Line 10**), the surrogate is re-evaluated; if evasion is successful, the process terminates early to maximise stealth (**Line 11**). We construct the feature pool $\mathcal{X}_{benign}$ by applying k -means clustering to historical flows with benign labels, selecting the medoids of each cluster as representative features for various communication protocols.

Algorithm 1 FIGA: Flow Injection Graph Attack Pipeline

Require: Target NIDS g_T ; Surrogate graph $\mathcal{G}_S$; Valid endpoint pool $\mathcal{V}_{ep}^{dst}$; Feature pool $\mathcal{X}_f$; Injection budget N ; Objective ℓ ; the set of flows including the target malicious flow $\mathcal{G}^0$

Ensure: Adversarial graph $\mathcal{G}_{adv}$

Phase 1: Offline Surrogate Training

1: $Y_S \leftarrow$ Query g_T with Q flows in $\mathcal{G}_S$ to obtain binary hard labels

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

Phase 2: Influence-Guided Flow Injection

3: **for** $t = 1$ to N **do**

4: *// Structural Selection via Influence Functions*

5: $v_{ep}^{(t)} \leftarrow \arg \max_{v \in \mathcal{V}_{ep}^{dst}} \left\| \frac{\partial \ell}{\partial \mathbf{h}_v} \right\|$ $\triangleright$ Find node with highest embedding influence

6: *// Feature Selection via Gradient Alignment*

7: $\mathbf{d} \leftarrow \frac{\partial \ell}{\partial \mathbf{x}_f}$ $\triangleright$ Compute directional gradient in feature space

8: $\mathbf{x}_{fl}^{(t)} \leftarrow \arg \max_{\mathbf{x}_{fl} \in \mathcal{X}_{benign}} \text{Cosine}(\mathbf{d}, \mathbf{x}_{fl})$ $\triangleright$ Pick best-matching benign features

9: $\mathcal{G}^{(t)} \leftarrow \mathcal{G}^{(t-1)} \oplus (v_{ep}^{(t)}, \mathbf{x}_{fl}^{(t)})$ $\triangleright$ Perform structural injection

10: **if** $\ell > 0.5$ on g_S **then break** $\triangleright$ Early stopping for stealth

11: **end for**

12: **return** $\mathcal{G}_{adv} \leftarrow \mathcal{G}^{(t)}$

5 Attack Feasibility Analysis

Theorem 1 (Weak Submodularity of FIGA). *Let $\mathcal{G} = (\mathcal{V}_f \cup \mathcal{V}_{ep}, \mathcal{E})$ be a heterogeneous graph representation of network traffic. Subject to additive-only and endpoint reachability constraints ($\mathcal{V}_{ep}^{dst} \subseteq \mathcal{V}_{ep}$), let $\mathcal{S}_{inj} = \{(v_{ep}^{(i)*}, \mathbf{x}_{fl}^{(i)*})\}_{i=1}^N$ be a set of injected cover flows within an attack budget N . The adversary, controlling endpoint $v_{ep,atk} \in \mathcal{V}_{ep}$, seeks to evade detection for a target malicious flow node $v_{fl,m} \in \mathcal{V}_f$.*

Let $m = \text{AGGREGATE}(\{\mathbf{h}_u : u \in \mathcal{N}(v_{\mathcal{H},m})\})$ be the mean-pooling based aggregated neighborhood message at node $v_{\mathcal{H},m}$, which serves as the continuous latent representation used for classification. Let $\ell_{\text{sur}}(m)$ and $\ell_{\text{tar}}(m)$ denote the evasion objectives for the surrogate (g_S) and target (g_T) models, respectively. Assume ℓ_{sur} is L -Lipschitz and μ -strongly concave with respect to m . If the cross-model gradient alignment is bounded such that for any message shift Δm induced by a single injection, $|\langle \nabla \ell_{\text{tar}}(m) - \nabla \ell_{\text{sur}}(m), \Delta m \rangle| \leq \epsilon$, then maximizing the surrogate objective $\ell_{\text{sur}}(m(\mathcal{S}_{\text{inj}}))$ is a γ_{sur} -weakly submodular maximization problem [9], where γ_{sur} is the submodularity ratio of ℓ_{sur} . Furthermore, the target evasion loss is bounded by:

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

ensuring that maximizing the surrogate evasion loss effectively maximizes a strict lower bound of the target model's evasion objective.

We adopt mean-pooling as a first-order proxy for general GNN message passing to establish a tractable mathematical baseline. While GCN [18] and GAT [29] introduce dynamic normalisation and attentions, respectively, the fundamental mechanism of neighbour feature aggregation remains a weighted summation.

Corollary 1 (Optimality Guarantee of the Influence-Function Search).

Let $\mathcal{I}(v_{\mathcal{H}}, g_S)$ be the influence function approximating the marginal gain $\ell_{\text{sur}}(\mathcal{G} \cup \{v_{\mathcal{H}}\}) - \ell_{\text{sur}}(\mathcal{G})$ without computing full GNN forward-backward passes. The proposed FIGA method, as given in Algorithm 1, acts as a discrete greedy maximisation algorithm. Selecting the injected set $\hat{\mathcal{S}}_N$ iteratively provides, under the previously stated assumptions, a $(1 - e^{-\gamma_{\text{sur}}})$ -approximation of the optimal injected flow set $\mathcal{O}_{\text{sur}}^*$, converging in polynomial-time $\mathcal{O}(N \cdot |\mathcal{V}_{\mathcal{H}}^{\text{pool}}| \cdot \mathcal{C}_{\text{inf}})$, where $\mathcal{C}_{\text{inf}}$ is the computational complexity of the influence score.

Proposition 1 (Query Complexity of Query-Based Attack Baselines).

Under hard-label access, any query-based zeroth-order attack without the surrogate model that attempts to identify the marginal gain of candidate destination endpoints $v_{ep} \in \mathcal{V}_{ep}^{\text{dst}}$ at a single greedy step t requires at least

$$Q^{(t)} \geq \Omega(|\mathcal{V}_{ep}^{\text{dst}}| \cdot |\mathcal{X}_{\text{val}}|) \quad (6)$$

queries towards the target GNN model to estimate, where $\Omega(\cdot)$ denotes an asymptotic lower bound, with constant accuracy, which endpoint $v_{ep}^{(t)}$ maximizes the marginal evasion gain $\ell(\mathcal{G}^{(t-1)} \oplus (v_{ep}, \mathbf{x}_{\mathcal{H}}))$.

This theoretical framework grounds the feasibility and severity of our black-box flow injection attack. **First**, Corollary 1 demonstrates that FIGA bypasses the NP-hard complexity of combinatorial graph poisoning, guaranteeing polynomial-time convergence and real-world scalability. Conversely, Proposition 1 proves that query-based baselines, lacking this surrogate heuristic, require a prohibitive number of live target queries with repetitive graph patterns, proportional to

Table 1: 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

the product of reachable endpoints and the candidate feature pool. **Second**, Theorem 1 ensures that this computational efficiency does not compromise effectiveness. As the surrogate objective is weakly submodular and transfers to the target within a bounded gap, our greedy optimization secures a tight $(1 - e^{-\gamma_{\text{sur}}})$ -approximation of the globally optimal perturbation. Thus, the algorithm reliably converges on highly potent structural injections that evade the target NIDS, even under stringent budget and query constraints. We empirically verify the stated assumptions: the Lipschitz and gradient alignment conditions hold across injection trajectories, and while the local curvature assumption is often violated in practice, convergence is unaffected. Full proof and code implementation are provided in the accompanying repository: <https://github.com/MouzaouiMatthieu/FIGA-Flow-Injection-Graph-Attacks>

6 Experiments

6.1 Experimental Setup

Datasets Experiments are conducted on the CIC-IDS2017 dataset [27] (with label corrections from [19]) as well as X-IIoTID [1]. We partitioned the flow sets using a temporal-order-aware split. In practice, the target NIDS is trained and deployed first, after which the attacker collects subsequent background traffic to train their surrogate before launching the flow injection attack. The first partition is used to train the target NIDS, the attacker collects subsequent traffic $\mathcal{G}_{\mathcal{S}}$ to train the surrogate model. The last partition, $\mathcal{G}_{\mathcal{T}}$, representing recent data, is used for attack design and evaluation. Datasets are partitioned into $\mathcal{G}_{\text{train}}$, $\mathcal{G}_{\mathcal{S}}$ and $\mathcal{G}_{\mathcal{T}}$ (see Table 1). $\mathcal{G}_{\text{train}}$ is used exclusively to train the target NIDS prior to deployment; the attacker has no access to it and collects $\mathcal{G}_{\mathcal{S}}$ only from subsequent background traffic.

GNN Models We evaluate three heterogeneous GNN architectures: GraphSAGE [15], a heterogeneous graph formulation equivalent to the E-GraphSAGE NIDS, GCN [18], and GAT [29], each instantiated with 2 and 4 message-passing

layers, indicated as **2L** and **4L**. Hyperparameter details and the intrusion detection accuracy of these models are given in Appendix 8.2 and 8.1.

Black-box Attack Baselines We compare two black-box GNN attack baselines. Score-based attacks require soft-label access. In the hard-label setting, it naturally reduces to a random fuzzing-based attack, referred to as *Random*. [24], referred to as *Hard-GNN*, assumes hard-label oracle access. For a fair comparison, it was originally applied for graph-level classification. We adapt it to handle node classification on heterogeneous graphs of flows, restricting modifications to edge additions between cover flow nodes and valid destination endpoints.

Evaluation Metrics We evaluate targeted evasion attacks using a maximum budget of $N = 100$ injected cover flow nodes. We exclusively attack a diverse test set of malicious flow nodes ($v_{f,m}$) in $\mathcal{G}_{\mathcal{T}}$ that are initially correctly detected by the surrogate and target models. An attack is deemed successful when both the surrogate ($g_{\mathcal{S}}$) and target ($g_{\mathcal{T}}$) models are forced to classify $v_{f,m}$ as benign. We define the Attack Success Rate (ASR) as the percentage of these malicious flows in $\mathcal{G}_{\mathcal{T}}$ successfully evaded. We use $\mathbf{ASR}_s$ (resp., $\mathbf{ASR}_t$) to denote ASR on the offline surrogate (resp., target) GNN-based NIDS. Given the maximum number of queries Q and injected flows N , a high $\mathbf{ASR}_s$ coupled with a minimal gap to $\mathbf{ASR}_t$ indicates strong attack transferability. We also report the intrusion detection F1-scores of the surrogate and target model on these targeted flows as $\mathbf{F1}_s$ and $\mathbf{F1}_t$ respectively. Lower $\mathbf{F1}_t$ indicates a stronger attack.

Our evaluation focuses on answering the following **Research Questions (RQs)**:

- RQ1: How effective is FIGA against widely used GNN architectures for NIDS compared to baseline methods, under an identical query budget?
- RQ2: What impact does the candidate flow feature selection strategy of FIGA have on the attack success?
- RQ3: Do different surrogate-to-target GNN-based NIDS pairs show significantly different attack transferability gap? Which GNN architecture proves most or least vulnerable to the flow injection attack?

6.2 Attack Effectiveness Evaluation

We evaluate the effectiveness of FIGA across varying query budgets (Q) compared to the baseline methods, addressing RQ.1.

Protocol The query budget (Q) ranges from 100 to 20000 allowed queries, representing less than 10% of the available surrogate traffic partition. For each value of Q , a corresponding surrogate model $g_{\mathcal{S}}$ is trained on the partition $\mathcal{G}_{\mathcal{S}}$ using exactly Q hard-labeled samples queried from the target NIDS $g_{\mathcal{T}}$. Thus, the actual graph used by the attacker to train $g_{\mathcal{S}}$ is a subgraph of $\mathcal{G}_{\mathcal{S}}$. Non-labeled nodes are discarded for training. To strictly isolate the impact of the query budget and eliminate any variance caused by feature selection, the candidate feature pool

is restricted to a single static vector: the centroid of benign flow features. This restriction isolates the impact of query budget from feature selection, which will be evaluated in Sect. 6.3. The maximum flow injection budget remains fixed at $N = 100$.

Table 2: $\mathbf{ASR}_t$ and $\mathbf{F1}_t$ of evasion attacks across GNN-based NIDS using 4-layer architectures on CIC-IDS2017 and X-IIoTID, for a given query budget Q .

Target NIDS	Q	SAGE-4L		GCN-4L		GAT-4L	
		ASR _{<i>t</i>}	F1 _{<i>t</i>}	ASR _{<i>t</i>}	F1 _{<i>t</i>}	ASR _{<i>t</i>}	F1 _{<i>t</i>}
<i>CIC-IDS2017</i>							
SAGE-4L	10 ²	0.870	0.230	0.710	0.450	0.740	0.413
	10 ³	0.990	0.020	0.990	0.020	0.969	0.059
	10 ⁴	1.000	0.000	1.000	0.000	1.000	0.000
	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
	10 ³	0.990	0.020	0.990	0.020	0.970	0.039
	10 ⁴	0.990	0.020	0.990	0.020	0.990	0.020
	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
	10 ³	1.000	0.000	1.000	0.000	0.980	0.039
	10 ⁴	1.000	0.000	1.000	0.000	1.000	0.000
	2 × 10 ⁴	1.000	0.000	1.000	0.000	1.000	0.000
<i>X-IIoTID</i>							
SAGE-4L	10 ²	0.450	0.710	0.939	0.115	0.620	0.551
	10 ³	0.530	0.640	0.951	0.093	0.620	0.519
	10 ⁴	0.813	0.320	0.950	0.095	0.870	0.230
	2 × 10 ⁴	0.94	0.113	0.951	0.093	0.89	0.198
GCN-4L	10 ²	0.280	0.837	1.000	0.000	0.660	0.508
	10 ³	0.320	0.810	1.000	0.000	0.690	0.473
	10 ⁴	0.320	0.810	1.000	0.000	0.750	0.400
	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
	10 ³	0.620	0.551	0.910	0.165	0.620	0.551
	10 ⁴	0.840	0.276	0.910	0.165	0.810	0.319
	2 × 10 ⁴	0.910	0.165	0.910	0.165	0.870	0.230
<i>Query-Based Attack Baselines</i>							
Random	2 × 10 ⁴	0.000	1.000	0.000	1.000	0.000	1.000
Hard-GNN	2 × 10 ⁴	0.000	1.000	0.000	1.000	0.000	1.000

FIGA demonstrates highly effective attacks under tight query budgets On CICIDS-2017, FIGA achieves ASR_t exceeding 70% with only 100 queries for most surrogate-target pairs, with performance saturating on average near $Q = 1000$. On X-IIoTID, convergence requires a larger surrogate budget, saturating on average near $Q = 10,000$. Crucially, these query counts represent only 0.4% and 5% of the surrogate subset $\mathcal{G}_S$ in the two datasets, respectively. Table 2 reports the full per-pair results. At these thresholds, the surrogate model achieves an F1 score > 0.8 relative to the target NIDS g_T , accurately approximating its decision boundary. Crucially, **for our proposed method, Q represents a minimal fraction of real background traffic that the attacker passively monitors and submits to the target NIDS for hard labels.** These queries consist entirely of graphs of real network flows. In contrast, the trial-and-error baselines repeatedly query nearly identical perturbed graphs, yielding easily detectable repetitive query patterns. Indeed, as per Proposition 1, any query-based zeroth-order attack must issue at least $\Omega(|\mathcal{V}_{ep}^{dst}| \cdot |\mathcal{X}_{val}|)$ queries per greedy step. With $|\mathcal{V}_{ep}^{dst}|$ in the order of thousands in our datasets (Table 1), the total budget of queries is insufficient. Both baselines degenerate to an uninformed trial-and-error, producing repetitive queries differing by only a single injection and would be detected as an anomaly long before any convergence. **During the attack phase, FIGA achieves a successful evasion while injecting only a minimal number of flows.** On the CIC-IDS2017 dataset, FIGA injects an average of fewer than 5 flows to reach $\text{ASR} > 80\%$ against GCN, SAGE, and GAT targets, and fewer than 29 flows on X-IIoTID.

Two baselines fail despite massive number of queries Both baselines fail completely (0% ASR_t), as Proposition 1 predicts that hard-label query-based search requires prohibitively many target queries, making convergence impractical.

Answer to RQ1 On CICIDS-2017, FIGA achieves high ASR already at $Q=500$, with average saturation near $Q=2,000$. On X-IIoTID, higher budgets are needed for some pairs, but FIGA outperforms both baselines which fail entirely with even $Q = 10^4$. This empirically validates the intractable query complexity of the baselines in Proposition 1 and the attack convergence and effectiveness guarantee of Theorem 1 and Corollary 1. The influence-guided offline search captures the most impactful flow injections, making the query volume a non-bottleneck. The empirical results across GCN and GAT show an identical saturation pattern in ASR, suggesting that the approximation guarantee of Corollary 1 is architecturally robust, even when the mean-pooling assumption in Theorem 1 is relaxed.

6.3 Impact of the Candidate Flow Feature Selection Strategy

This section addresses RQ.2. We evaluate how the diversity of the candidate flow feature pool used in the attack and the specific feature selection strategy affect

the overall performance of the evasion attack. Moreover, repeatedly injecting the same flows would raise detectability concerns.

Protocol To isolate the effect of feature selection from the query budget, we conduct experiments in this section with CIC-IDS2017 data using the maximum available query budget of 20,000. We fix the maximum number of injected flows to 5 based on the results observed in Sect 6.2. The pool size (number of clusters) is varied across $k \in \{5, 10, 15, 20, 25, 30\}$. FIGA’s gradient-informed flow feature selection is evaluated across these different pool sizes k against a target SAGE-2L NIDS with 2-layer GNN surrogates. We also compared against *FIGA-Random* which uniformly samples from the flow pool, regardless of the gradient alignment (Eq.5). The comparison for the two strategies is reported in Table 3.

Extending the pool size does not improve FIGA’s effectiveness We report ASR and F1 on the target model (SAGE-L2) for different k . When $k \leq 15$, ASR and F1 on the target model exceed 0.96 and $F1_t$ is lower than 0.1, respectively, indicating successful attacks. However, increasing k to 30 induces lower $\mathbf{ASR}_t$ between 0.96 and 0.85, and raises $F1_t$ to 0.26. A possible explanation is that a larger pool may lead the injected flows to overfit the local surrogate model, thereby reducing their transferability to the target model. In summary, statistical analysis reveals no significant correlation between the size of the feature pool k and the final attack success rate (Spearman’s $\rho = 0.328$, $p = 0.329$).

Gradient-informed features selection boosts the attack effectiveness

Table 3 demonstrates that FIGA’s gradient-informed features selection provides an initial advantage over uniform random selection. At fixed feature pool size ($k = 15$), with a SAGE-2L as target NIDS, and 2-layer GNNs as surrogates, FIGA achieves $\mathbf{ASR}_t$ of 0.957, with a maximum of 100 injected cover flow nodes, while *FIGA-Random* achieves an $\mathbf{ASR}_t$ of 0.660 in the same settings. Moreover, the $\mathbf{ASR}_t$ is already 0.763 with only 2 injected cover flow nodes using **FIGA**, which is 3 times higher than that of *FIGA-Random* ($\mathbf{ASR}_t = 0.224$). By leveraging gradient information, FIGA forces the selected features to make a directionally coherent contribution toward the benign decision boundary, achieving high ASR with very few injections. In contrast, the random strategy achieves only moderate success given a sufficiently high injection budget ($n = 100$).

Answers to RQ.2 Feature selection plays a secondary but operationally vital role. FIGA’s gradient-informed approach vastly outperforms the random guess-based scheme by aligning injected features into a directionally coherent contribution to the GNN’s aggregated message. Crucially, a larger feature pool does not systematically correlate with higher attack success, as most benign features produce functionally similar effects on the GNN’s spatial context. This grants attackers considerable flexibility: *rather than requiring an exhaustive dataset, an adversary can curate a small pool of high-confidence benign flows to efficiently*

Table 3: Evasion performance across feature selection strategies, with varying number of injected flows n . The number of queries is set to 20,000, and the pool size is fixed to $k = 15$, on CIC-IDS2017.

n	FIGA				FIGA-Random			
	Surrogate		Transfer		Surrogate		Transfer	
	$\mathbf{ASR}_s \uparrow$	$\mathbf{F1}_s \downarrow$	$\mathbf{ASR}_t \uparrow$	$\mathbf{F1}_t \downarrow$	$\mathbf{ASR}_s \uparrow$	$\mathbf{F1}_s \downarrow$	$\mathbf{ASR}_t \uparrow$	$\mathbf{F1}_t \downarrow$
2	0.904	0.175	0.763	0.384	0.302	0.822	0.224	0.874
3	0.937	0.119	0.813	0.315	0.479	0.685	0.432	0.724
4	0.952	0.092	0.841	0.275	0.479	0.685	0.433	0.724
5	0.952	0.092	0.947	0.101	0.559	0.612	0.549	0.622
25	0.992	0.015	0.992	0.015	0.484	0.681	0.375	0.769
50	1.000	0.000	0.952	0.092	0.668	0.499	0.605	0.567
100	0.998	0.005	0.957	0.082	0.731	0.425	0.660	0.508

maximize the benign signal during message-passing, guaranteeing evasion with minimal injections.

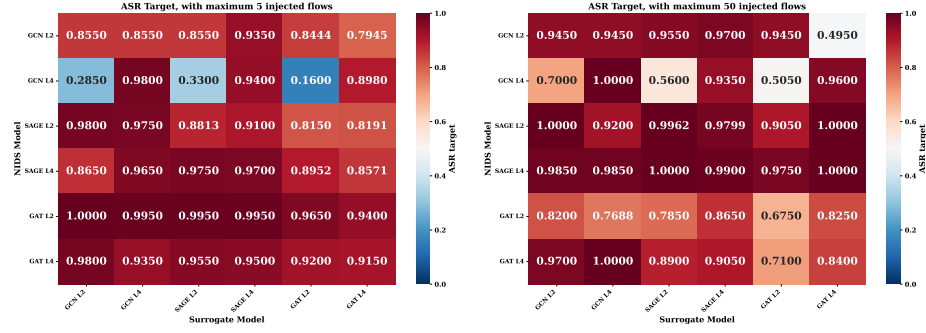
6.4 Attack Transferability Gap

In this section, we focus on addressing RQ.3, investigating how GNN model architectural discrepancies between the surrogate and target affect attack efficacy. **Protocol** We evaluate the surrogate-to-target attack transferability across all 6 target architectures (SAGE, GCN, GAT $\times$ L2/L4), with a feature pool size $k=15$, a number of queries $Q=20,000$, and cosine-similarity feature selection.

Broad transferability across architectures With only 5 injected flows on the CIC-IDS2017, and 50 on X-IIoTID, FIGA achieves high $\mathbf{ASR}_t$ across most surrogate-target pairs, shown in Figure 2. The adversarial risk posed by the flow injection threat pervades a wide spectrum of GNN-based NIDS.

In Figure 2, we observe the lowest attack transferability of 2-layer surrogates against the 4-layer GCN target. Specifically, using 2-layer GCN, GraphSAGE, and GAT surrogates yields ASRs of only 0.285, 0.33, and 0.16 on GCN-L4 on CIC-IDS2017 (Figure 2a), respectively. Conversely, 4-layer surrogates, especially GCN-L4 and SAGE-L4, maintain the highest transferred ASR over the target models. The same pattern is observed on X-IIoTID (Figure 2b). Deeper surrogates generate denser perturbations that transfer better across architectures, whereas GAT-based attacks are more architecture-specific.

Answer to RQ3 High attack transferability is achieved across most surrogate-target pairs, which confirms that the proposed flow injection attack is effective against a wide spectrum of GNN-based NIDS. For most surrogate-victim pairs, FIGA achieves $\mathbf{ASR}_t > 0.80$ within the tested injection budget. However, defenders should note that deeper architectures with symmetric normalisation, specifically GCN-L4, demonstrate the highest natural resilience to this threat.



(a) CIC-IDS2017 dataset: $\mathbf{ASR}_t$ of evasion attacks, with a maximum of 5 cover flow nodes injected. (b) X-IIoTID dataset: $\mathbf{ASR}_t$ of evasion attacks, with a maximum of 50 cover flow nodes injected.

Fig. 2: Comparison of $\mathbf{ASR}_t$ of evasion attacks across CIC-IDS2017 and X-IIoTID datasets, with different pairs of NIDS and surrogate models. The feature pool size is fixed at $k = 15$, and the number of queries to $Q = 20,000$, for a maximum number of injected flow nodes enough to make the $\mathbf{ASR}_t$ converge.

7 Conclusion

In this work, we present FIGA, a practical hard-label, black-box flow injection attack to evaluate the adversarial risk of GNN-based NIDS. By leveraging off-line surrogate modeling to preserve exploit integrity, FIGA strictly adheres to stringent operational and query constraints. Theoretically, under the stated assumptions about the evasion objective and the surrogate-target alignments, we establish that its influence-guided search is a weakly submodular maximization problem, guaranteeing polynomial-time convergence to near-optimal perturbations. Empirically, FIGA demonstrates strong transferability and superior attack success rates across diverse GNN architectures, drastically outperforming existing baselines in query efficiency. By exposing these structural vulnerabilities, we highlight the need for context-aware defenses. While our evaluation is conducted on two widely used benchmarks, extending FIGA to larger-scale enterprise network environments with dynamic topology and higher traffic variability remains an important direction for future work. Future work will also explore advanced robustification, though we conjecture standard adversarial training may struggle to mitigate purely structural manipulations.

Acknowledgments. This work has been partially supported by the French National Research Agency (ANR) under the France 2030 label (SuperViz ANR-22-PECY-0008). Experiments presented in this paper were carried out using the Grid’5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see <https://www.grid5000.fr>).

References

1. Al-Hawawreh, M., Sitnikova, E., Aboutorab, N.: X-IIoTID: A Connectivity-Agnostic and Device-Agnostic Intrusion Data Set for Industrial Internet of Things. *IEEE Internet of Things Journal* **9**(5), 3962–3977 (Mar 2022). <https://doi.org/10.1109/JIOT.2021.3102056>
2. Bojchevski, A., Günnemann, S.: Adversarial attacks on node embeddings via graph poisoning. In: Chaudhuri, K., Salakhutdinov, R. (eds.) *Proceedings of the 36th ICML. Proceedings of Machine Learning Research*, vol. 97, pp. 695–704. PMLR (Jun 2019), <https://proceedings.mlr.press/v97/bojchevski19a.html>
3. Chang, L., Branco, P.: Graph-based solutions with residuals for intrusion detection: The modified e-GraphSAGE and e-ResGAT algorithms. *ArXiv abs/2111.13597* (2021)
4. Chen, J., Xu, H., Wang, J., Xuan, Q., Zhang, X.: Adversarial detection on graph structured data. In: *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*. p. 37–41. PPMLP’20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3411501.3419424>
5. Chen, W.H., Hsu, S.H., Shen, H.P.: Application of SVM and ANN for intrusion detection **32**(10), 2617–2634. <https://doi.org/10.1016/j.cor.2004.03.019>
6. Chernikova, A., Oprea, A.: FENCE: Feasible evasion attacks on neural networks in constrained environments. *ACM Trans. Priv. Secur.* **25**(4) (Jul 2022). <https://doi.org/10.1145/3544746>
7. Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., Song, L.: Adversarial attack on graph structured data. In: Dy, J., Krause, A. (eds.) *Proceedings of the 35th ICML. Proceedings of Machine Learning Research*, vol. 80, pp. 1115–1124. PMLR (Jul 2018)
8. Deng, P., Huang, Y.: Edge-featured multi-hop attention graph neural network for intrusion detection system. *Computers & Security* **148**, 104132 (Jan 2025). <https://doi.org/10.1016/j.cose.2024.104132>
9. Elenberg, E.R., Khanna, R., Dimakis, A.G., Negahban, S.: Restricted strong convexity implies weak submodularity. *The Annals of Statistics* **46**(6B), 3538–3568 (2018)
10. Farnaaz, N., Jabbar, M.: Random Forest Modeling for Network Intrusion Detection System. *Procedia Computer Science* **89**, 213–217 (2016). <https://doi.org/10.1016/j.procs.2016.06.047>
11. Farrukh, Y.A., Wali, S., Khan, I., Bastian, N.D.: XG-NID: Dual-modality network intrusion detection using a heterogeneous graph neural network and large language model **287**, 128089. <https://doi.org/10.1016/j.eswa.2025.128089>
12. Finkelshtein, B., Baskin, C., Zheltonozhskii, E., Alon, U.: Single-node attacks for fooling graph neural networks. *Neurocomputing* **513**, 1–12 (Nov 2022). <https://doi.org/10.1016/j.neucom.2022.09.115>
13. Friji, H., Olivereau, A., Sarkiss, M.: Efficient Network Representation for GNN-Based Intrusion Detection. In: *Applied Cryptography and Network Security: 21st International Conference, ACNS 2023, Kyoto, Japan, June 19–22, 2023, Proceedings, Part I*. pp. 532–554. Springer-Verlag, Berlin, Heidelberg (Jun 2023). https://doi.org/10.1007/978-3-031-33488-7_20
14. Galli, D., Venturi, A., Marasco, I., Marchetti, M.: Evaluating explainability of graph neural networks for network intrusion detection with structural attacks. In: *Proceedings of the 2025 Joint National Conference on Cybersecurity (ITASEC) and SERICS. CEUR Workshop Proceedings*, vol. 3962. CEUR-WS, Bologna, Italy (Feb 2025)

15. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 1025–1035. NIPS’17, Curran Associates Inc., Red Hook, NY, USA (2017)
16. Ibitoye, O., Shafiq, O., Matrawy, A.: Analyzing adversarial attacks against deep learning for intrusion detection in iot networks. In: 2019 IEEE Global Communications Conference (GLOBECOM). pp. 1–6 (2019). <https://doi.org/10.1109/GLOBECOM38437.2019.9014337>
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
18. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th ICLR, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017)
19. Lanvin, M., Gimenez, P.F., Han, Y., Majorczyk, F., Mé, L., Totel, É.: Errors in the cids2017 dataset and the significant differences in detection performances it makes. In: Kallel, S., Jmaiel, M., Zulkernine, M., Hadj Kacem, A., Cuppens, F., Cuppens, N. (eds.) Risks and Security of Internet and Systems. pp. 18–33. Springer Nature Switzerland, Cham (2023)
20. Lin, L., Zhong, Q., Qiu, J., Liang, Z.: E-GRACL: An IoT intrusion detection system based on graph neural networks. The Journal of Supercomputing **81**(1), 42 (Oct 2024). <https://doi.org/10.1007/s11227-024-06471-5>
21. Liu, Z., Zhao, Y., Liu, Z., Li, Q., Fu, C., Zhou, G., Xu, K.: A hard-label black-box evasion attack against ML-based malicious traffic detection systems. In: NDSS (2026)
22. Lo, W.W., Layeghy, S., Sarhan, M., Gallagher, M., Portmann, M.: E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT. In: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. pp. 1–9 (Apr 2022). <https://doi.org/10.1109/NOMS54207.2022.9789878>
23. Merzouk, M.A., Cuppens, F., Boulahia-Cuppens, N., Yaich, R.: Investigating the practicality of adversarial evasion attacks on network intrusion detection. Annals of Telecommunications **77**(11-12), 763–775 (Dec 2022). <https://doi.org/10.1007/s12243-022-00910-1>
24. Mu, J., Wang, B., Li, Q., Sun, K., Xu, M., Liu, Z.: A Hard Label Black-box Adversarial Attack Against Graph Neural Networks. In: Proceedings of ACM CCS 2021. pp. 108–125. ACM, Virtual Event Republic of Korea (Nov 2021). <https://doi.org/10.1145/3460120.3484796>
25. Pierazzi, F., Pendlebury, F., Cortellazzi, J., Cavallaro, L.: Intriguing properties of adversarial ML attacks in the problem space. In: Proceedings of IEEE SP Oakland, 2020. pp. 1332–1349 (2020). <https://doi.org/10.1109/SP40000.2020.00073>
26. Pujol-Perich, D., Suarez-Varela, J., Cabellos-Aparicio, A., Barlet-Ros, P.: Unveiling the potential of Graph Neural Networks for robust Intrusion Detection. ACM SIGMETRICS Performance Evaluation Review **49**(4), 111–117 (Jun 2022). <https://doi.org/10.1145/3543146.3543171>
27. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: International Conference on Information Systems Security and Privacy (2018)
28. Sheatsley, R., Papernot, N., Weisman, M., Verma, G., McDaniel, P.: Adversarial examples for network intrusion detection systems. Journal of Computer Security **30**(5), 727–752 (Jan 2022)

29. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: Proceedings of ICLR 2018 (2018)
30. Venturi, A., Ferrari, M., Marchetti, M., Colajanni, M.: ARGANIDS: A novel Network Intrusion Detection System based on adversarially Regularized Graph Autoencoder. In: Proceedings of the 38th ACM Symposium on Applied Computing. pp. 1540–1548. Association for Computing Machinery (Jun 2023)
31. Venturi, A., Stabili, D., Marchetti, M.: Problem space structural adversarial attacks for Network Intrusion Detection Systems based on Graph Neural Networks (Mar 2024). <https://doi.org/10.48550/arXiv.2403.11830>
32. Wang, B., Li, Y., Zhou, P.: Bandits for structure perturbation-based black-box attacks to graph neural networks with theoretical guarantees. In: 2022 IEEE/CVF Conference on CVPR. pp. 13369–13377 (2022)
33. Wang, B., Lin, M., Zhou, T., Zhou, P., Li, A., Pang, M., Li, H., Chen, Y.: Efficient, direct, and restricted black-box graph evasion attacks to any-layer graph neural networks via influence function. In: Proceedings of the 17th ACM WSDM. pp. 693–701 (2024)
34. Wang, M., Yang, N., Weng, N.: K-getnid: Knowledge-guided graphs for early and transferable network intrusion detection. IEEE Transactions on Information Forensics and Security **19**, 7147–7160 (2024). <https://doi.org/10.1109/TIFS.2024.3431932>
35. Wang, T., Cai, X., Nithyanand, R., Johnson, R., Goldberg, I.: Effective attacks and provable defenses for website fingerprinting. In: Proceedings of the 23rd USENIX Conference on Security Symposium. p. 143–157. SEC’14, USENIX Association
36. Wen, L., Liang, J., Yao, K., Wang, Z.: Black-box adversarial attack on graph neural networks with node voting mechanism. IEEE Trans. on Knowl. and Data Eng. **36**(10), 5025–5038 (Oct 2024)
37. Zhang, A., Zhao, Y., Zhou, C., Zhang, T.: Resacag: A graph neural network based intrusion detection. Comput. Electr. Eng. **122**(C) (Mar 2025)
38. Zhao, M., Zhang, J.: Highly imperceptible black-box graph injection attacks with reinforcement learning. Proceedings of the AAAI Conference on Artificial Intelligence **39**(12), 13357–13364 (Apr 2025)
39. Zhong, M., Lin, M., Zhang, C., Xu, Z.: A survey on graph neural networks for intrusion detection systems: Methods, trends and challenges. Computers & Security **141**, 103821 (Jun 2024). <https://doi.org/10.1016/j.cose.2024.103821>
40. Zhu, P., Pan, Z., Liu, Y., Tian, J., Tang, K., Wang, Z.: A general black-box adversarial attack on graph-based fake news detectors. In: Proceedings of the 33rd IJCAI. IJCAI ’24 (2024)
41. Zolbayer, B.E., Sheatsley, R., Mcdaniel, P., Weisman, M., Zhu, S., Zhu, S., Krishnamurthy, S.V.: Generating practical adversarial network traffic flows using NIDS-GAN. ArXiv **abs/2203.06694** (2022)
42. Zügner, D., Akbarnejad, A., Günnemann, S.: Adversarial Attacks on Neural Networks for Graph Data. In: Proceedings of the 24th ACM SIGKDD. pp. 2847–2856 (2018)

8 Appendix

8.1 Model hyperparameters

Each model features a hidden node embedding dimension of 120 neurons and an output embedding dimension of 128; the GAT variant utilises 4 attention heads.

The detection performance of these models is reported in the appendix 8.2. The models are trained on $\mathcal{G}_{\text{train}}$ using the Adam [17] optimiser with a learning rate of 10^{-3} and weight decay of 10^{-4} for up to 500 epochs, utilising early stopping based on validation loss.

8.2 Detection performances of GNN-based NIDS

Table 4: Target NIDS performance on CICIDS-2017.

Model	Accuracy	Precision	Recall	F1	AUC
GCN L2	0.985	0.977	0.997	0.985	0.987
GCN L4	0.986	0.978	0.997	0.986	0.988
SAGE L2	0.941	0.980	0.909	0.941	0.988
SAGE L4	0.924	0.980	0.877	0.924	0.985
GAT L2	0.979	0.977	0.984	0.979	0.989
GAT L4	0.946	0.979	0.921	0.946	0.984

Table 5: Target NIDS performance on X-IIoTID (G_{test}).

Model	Accuracy	Precision	Recall	F1	AUC
SAGE L2	0.787	0.780	0.700	0.779	0.896
SAGE L4	0.732	0.770	0.537	0.711	0.787
GCN L2	0.865	0.803	0.908	0.864	0.954
GCN L4	0.860	0.828	0.850	0.858	0.957
GAT L2	0.699	0.655	0.631	0.692	0.808
GAT L4	0.724	0.870	0.420	0.682	0.872