GATv2-NS3 Hybrid IDS: Self-Focusing Simulations for Network Intrusion Detection

AI Agent

Alexander Buyantuev
ITMO,
Innopolis University

Aliaksei Korshuk
Innopolis University

Aleksei Stepin
ITMO,
Innopolis University

Ilya Gusev
Independent Researcher

Vladimir Kubasov
ITMO,
Innopolis University

Vladislav Kulikov
ITMO,
Innopolis University

Artyom Kabanov
ITMO,
Innopolis University

Mikhail Mozikov
AIRI, ISP RAS

Ilya Makarov
AIRI, ISP RAS, Innopolis University

Abstract

Network intrusion detection is prone to data leakage and inflated scores under static evaluation protocols. We present GATv2-NS3, a hybrid IDS that couples Graph Attention Networks v2 with an adaptive NS-3 simulator. Our key idea, *Self-Focusing Simulations*, leverages attention-entropy uncertainty to selectively run packet-level simulations on ambiguous subgraphs, forming a training-time feedback loop that injects QoS signals (latency, jitter, loss, throughput) via a simulation-consistency loss. The results indicate that uncertainty-guided, simulation-grounded learning yields more honest metrics without sacrificing efficiency, advancing practical IDS reliability.

1 Introduction

Network intrusion detection systems (IDS) face fundamental challenges from data leakage and artificial performance inflation that compromise research reliability. Traditional evaluation methodologies lead to overly optimistic performance claims [Madani et al., 2022], with NSL-KDD studies often reporting $> 90\%$ accuracy or $F_1 \approx 0.75$ due to experimental bias [Kus et al., 2022]. This creates a disconnect between academic results and real-world deployment, where IDS systems rarely achieve such performance. Current graph-based IDS approaches suffer from three critical limitations: (1) static evaluation protocols that ignore dynamic network behavior, (2) lack of uncertainty quantification for guiding resource allocation, and (3) absence of adaptive simulation mechanisms for behavior-grounded validation. While Graph Attention Networks show promise [Veličković et al., 2018, Brody et al., 2021], prior work has not leveraged attention uncertainty as a control signal for adaptive simulation fidelity.

Research Question: How can we develop a hybrid IDS framework that combines graph attention mechanisms with adaptive network simulation to achieve realistic intrusion detection performance while allocating computational resources based on model uncertainty?

We introduce GATv2-NS3, a hybrid IDS that couples Graph Attention Networks v2 with an adaptive NS-3 simulator. Our key idea, *Self-Focusing Simulations*, computes attention-entropy uncertainty over nodes and selectively triggers packet-level simulations on ambiguous subgraphs, injecting QoS signals (latency, jitter, loss, throughput) via a simulation-consistency loss. Under a leakage-aware NSL-KDD protocol, GATv2-NS3 attains Accuracy=0.805, Precision=0.699, Recall=0.633, and

$F_1 = 0.649$, substantially below commonly reported figures yet more representative of realistic IDS performance.

1.1 Key Contributions

- **Self-Focusing Simulations**: First use of GATv2 attention-entropy as a control signal for adaptive NS-3 packet-level simulation, dynamically allocating fidelity to ambiguous subgraphs and feeding back QoS signals via a simulation-consistency loss.

- **Leakage-Aware Evaluation**: A rigorous NSL-KDD protocol that removes dataset leakage and standardizes preprocessing/splits, establishing realistic benchmarks for graph-based IDS.

- **Baseline Coverage**: Consistent comparison across graph and classical IDS models on NSL-KDD (GATv2, GraphSAGE, GIN, MLP, Random Forest, Logistic Regression, XGBoost) under the same pipeline.

- **Realistic Performance Insights**: Under the leakage-aware protocol, GATv2-NS3 attains Accuracy=0.805, Precision=0.699, Recall=0.633, and $F_1 = 0.649$; performance across methods concentrates around $F_1 \approx 0.60$–0.65, contrasting with commonly reported $F_1 \approx 0.75$ or $> 90\%$ accuracy.

2 Related Work

IDS Datasets and Evaluation. Classical benchmarks (KDD'99, NSL-KDD) can induce over-optimistic results through preprocessing and split leakage [Tavallaee et al., 2009, Zhipeng et al., 2017]. Modern corpora (UNSW-NB15, CIC-IDS2017/2018, UGR'16, Bot-IoT, ToN-IoT) [Moustafa and Slay, 2015, Sharafaldin et al., 2018, Communications Security Establishment and Canadian Institute for Cybersecurity, 2018, Maciá-Fernández et al., 2018, Koroniotis et al., 2019, Moustafa, 2021] improve realism but still exhibit imbalance and protocol-induced leakage [Kasongo and Sun, 2020]. Surveys and many pipelines report $> 95\%$ accuracy under static splits [Leevy and Khoshgoftaar, 2020, Elsayed et al., 2024], yet such figures often fail to generalize when leakage is controlled [Kus et al., 2022]. We adopt a leakage-aware NSL-KDD protocol to establish realistic baselines.

Graph-based IDS. GNNs encode topology beyond flat features. GraphSAGE [Hamilton et al., 2017], GIN [Xu et al., 2019], and GAT [Veličković et al., 2018] have been adapted to flow/host-level detection [Caville et al., 2022]. Despite promising results, gains frequently rely on static snapshots and single-dataset settings. We build on GATv2 [Brody et al., 2021] and explicitly exploit attention-entropy uncertainty as a control signal, rather than treating attention solely as a representational mechanism.

Uncertainty and Adaptive Learning. Uncertainty estimation (MC dropout [Gal and Ghahramani, 2016], deep ensembles [Lakshminarayanan et al., 2017]) and calibration [Guo et al., 2017] support trustworthy deployment [Talpini et al., 2024]. Active learning reduces labeling cost [Tüzün and Angin, 2024]. Concept-drift methods (INSOMNIA [Andresini et al., 2021], CADE [Yang et al., 2021]) tackle non-stationarity. Distinct from these, we use attention entropy to drive targeted packet-level simulation and integrate QoS feedback via a simulation-consistency loss.

Simulation-based Evaluation and Digital Twins. Network simulators such as ns-3 enable controlled, repeatable experiments [Henderson and Riley, 2020]. Network digital twins provide model-driven testbeds for security and performance studies. Our *Self-Focusing Simulations* extend this paradigm to IDS, steering ns-3 toward ambiguous subgraphs and grounding learning in traffic-level QoS. Enterprise topologies (e.g., Cisco Secure Workload networks [Stanford Network Analysis Project (SNAP), 2024]) illustrate realistic structural patterns for graph-based IDS.

Positioning. Unlike prior graph-based IDS that assume static datasets and uniform evaluation, our framework (i) couples GNN attention with adaptive simulation control, (ii) enforces leakage-aware evaluation, and (iii) yields interpretable forensic artifacts by re-simulating uncertain regions.

3 Methodology

3.1 Problem Formulation

We consider node-wise intrusion detection on a communication graph $G = (V, E, X, A)$ with nodes V (hosts), edges E (communications), node features $X \in \mathbb{R}^{|V|\times d}$, and optional edge features $A \in \mathbb{R}^{|E|\times f}$. Each node $i \in V$ has a class label $y_i \in \{1, \ldots, C\}$. A standard GNN learns $f_\theta : (X, E) \mapsto Z$ with logits $Z \in \mathbb{R}^{|V|\times C}$ and probabilities $P = \mathrm{softmax}(Z)$; training minimizes cross-entropy $\mathcal{L}_{\mathrm{cls}}(Z, y)$.

GATv2-NS3 augments this with uncertainty-guided simulation feedback that affects training (not inference). Let the final-layer GATv2 attention coefficients be α_{ijh} over incoming edges $(j \to i)$ and heads $h = 1, \ldots, H$. We define per-node attention-entropy uncertainty

$$U_i \;=\; -\frac{1}{H}\sum_{h=1}^{H}\sum_{j\in\mathcal{N}(i)} \alpha_{ijh}\,\log(\alpha_{ijh} + 10^{-10}).$$

An adaptive threshold selects uncertain nodes:

$$\tau \;=\; \tau_0 + \beta\cdot \mathrm{std}(U), \qquad T \;=\; \{\, i \in V \mid U_i > \tau \,\},$$

and we extract a k-hop subgraph $G_S = (V_S, E_S, X_S)$ around T with $k =$ `subgraph_hops` (default $k = 2$) and capacity $|V_S| \le 1000$; if $T = \varnothing$ or $|V_S| > 1000$, simulation is skipped.

A simulator $\mathcal{M}$ runs ns-3 on G_S and returns per-node QoS metrics

$$M \;=\; \mathcal{M}(G_S) \in \mathbb{R}^{|V_S|\times 4}, \qquad M = \big[\,\text{latency, jitter, packet\_loss, throughput}\,\big],$$

z-normalized per metric before use. Training adds (i) a simulation-consistency loss aligning per-node feature variance and QoS variance,

$$\mathcal{L}_{\mathrm{sim}}(X_S, M) \;=\; \Big\|\,\mathrm{norm}\big(\mathrm{var}(X_S)\big) \;-\; \mathrm{norm}\big(\mathrm{var}(M)\big)\,\Big\|_2^2,$$

and (ii) attention regularization toward target entropy $U^\star$,

$$\mathcal{L}_{\mathrm{att}}(U) \;=\; \big(U - U^\star\big)^2_{\mathrm{mean}}.$$

The total objective is

$$\mathcal{L}(\theta; t) \;=\; \mathcal{L}_{\mathrm{cls}}(Z, y) \;+\; \lambda_1(t)\,\mathcal{L}_{\mathrm{sim}}(X_S, M) \;+\; \lambda_2(t)\,\mathcal{L}_{\mathrm{att}}(U),$$

with $\lambda_1(t) = \lambda_1^{(0)} e^{-\gamma t}$ and $\lambda_2(t) = \lambda_2^{(0)}(1 + \rho t)$. If no simulation is triggered, $\mathcal{L}_{\mathrm{sim}} = 0$. Inference uses f_θ alone: $\hat{y}_i = \arg\max_c Z_{ic}$; U is reported for interpretability.

3.2 Self-Focusing Simulations Framework

3.2.1 GATv2 Architecture and Attention Uncertainty

We use GATv2 [Brody et al., 2021] with $L = 3$ layers, hidden dimension $d_h = 128$, $K = 8$ heads, dropout $p = 0.3$, and LeakyReLU slope $\alpha = 0.2$. The final layer provides attention coefficients used to compute U_i as defined above.

3.2.2 Adaptive Simulation Control

Simulations are considered every N epochs (N=5 during training) and restricted to uncertain regions. The adaptive threshold uses entropy parameters τ_0=0.5, β=0.3. Subgraphs are k=2-hop neighborhoods (`subgraph_hops`) with a capacity limit $|V_S| \le 1000$. The simulator outputs per-node metrics M and applies z-normalization. When ns-3 bindings are unavailable, a feature-aware stub generates realistic variability; otherwise, packet-level tracing uses perturbation ranges: bandwidth 10–100 Mbps, delay 1–50 ms, packet drops 5–15%, jitter 10–50 ms.

3.2.3 Multi-Objective Training

Training minimizes the objective defined in the Problem Formulation with configuration-derived schedules: $\lambda_1^{(0)}$=0.1, γ=0.001, $\lambda_2^{(0)}$=0.01, ρ=0.0001 and target entropy $U^\star$=0.7. If no simulation is triggered, only $\mathcal{L}_{\mathrm{cls}}$ and $\mathcal{L}_{\mathrm{att}}$ contribute.

3.3 Graph Construction

NSL-KDD. We build separate graphs for the official train and test splits. Each record becomes a node with a five-way class label (Normal, DoS, Probe, R2L, U2R).

Feature processing.

1. Categorical features: discrete fields are converted to numeric form; low-cardinality fields are one-hot encoded, and high-cardinality fields are label-encoded. Unseen categories at test time are mapped to a default "unknown".
2. Numerical features: retained as-is.
3. Standardization: all features are z-score normalized using statistics fit on the training set and applied to the test set.

Graph building. For each split, we form a directed k-nearest neighbor graph (k=10) using cosine similarity over the standardized features. Each node connects to its k most similar neighbors (self excluded), keeping only edges with positive similarity. Edge weights are the cosine similarity values; the average out-degree is therefore close to k (lower if few positives exist).

Batched graphs. Large graphs are partitioned into contiguous subgraphs of approximately 3000 nodes to fit memory and enable batching; cross-partition edges are dropped and indices are relabeled. In the experiment pipeline, the test graphs are further divided into validation and test subsets of equal size.

3.4 Baseline Configurations

Graph neural networks.

- GraphSAGE: 3 layers with hidden size 128; ReLU after each layer; dropout $p = 0.5$ between layers; linear classifier on top.
- GIN: 3 layers; each layer uses a 2-layer MLP inside the GINConv; batch normalization after each layer; ReLU activations; dropout $p = 0.5$; linear classifier.
- MLP: feed-forward network with hidden sizes [256, 128, 64]; ReLU and dropout $p = 0.3$ after each hidden layer; linear output layer; ignores graph structure.

Traditional machine learning.

- Random Forest: 100 trees, `max_depth=10`, `class_weight=balanced`, `n_jobs=-1`.
- XGBoost: 100 estimators, `learning_rate=0.1`, `max_depth=6`, `subsample=0.8` (included only if the package is available).
- Logistic Regression: multinomial (softmax) with L2 regularization (C=1.0), LBFGS solver, `class_weight=balanced`, `max_iter=1000`.

3.5 Training Protocol

Data splits and batching. We use the official NSL-KDD splits available on Kaggle [nsl, 2009]. After feature encoding and graph construction, each split is formed into a single graph and then partitioned into contiguous subgraphs of $\sim$3000 nodes for memory/batching. The test split is further divided 50/50 into validation and test subsets (by number of subgraphs).

Neural models (GraphSAGE, GIN, MLP, GATv2 baseline). Training is performed per subgraph with class-weighted cross-entropy. We use Adam (lr= 0.001, betas= $(0.9, 0.999)$), StepLR (factor 0.95 every 10 epochs), early stopping on validation macro-F_1 with patience 20, and a maximum of 200 epochs. Class weights are computed from the training labels.

GATv2-NS3 (hybrid). Same optimizer/scheduler/early-stopping as above, with an additional uncertainty-guided simulation path: every 5 epochs, we compute attention-entropy, select high-uncertainty nodes via $\tau = \tau_0 + \beta \cdot \mathrm{std}(H)$ ($\tau_0 = 0.5,\ \beta = 0.3$), extract a k=2-hop subgraph (capped at 1000 nodes), and obtain per-node QoS metrics (latency, jitter, packet_loss, throughput). Simulation metrics are z-normalized and used in the simulation-consistency loss. The total objective is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_1(t)\,\mathcal{L}_{\text{sim}} + \lambda_2(t)\,\mathcal{L}_{\text{att}}, \quad \lambda_1(t) = 0.1\,e^{-0.001t},\ \lambda_2(t) = 0.01\,(1 + 0.0001t),$$

with target entropy 0.7. If no simulation is triggered in an iteration, $\mathcal{L}_{\text{sim}} = 0$.

Traditional ML (RF, XGB, LogReg). We flatten node features and labels from the graph partitions and perform one-shot training on the training set, then evaluate on validation and test sets. No learning-rate schedules are used.

Metrics and timing. For all models, we report macro-F_1, accuracy, macro-precision, and macro-recall on the held-out test partitions, along with wall-clock training time in minutes.

3.6 Experimental Setup

Practical adjustments.

- Optional subsampling: the training split can be reduced to 30,000 records for runtime control prior to graph construction.
- Graph partitioning: each split is formed as one graph, then sliced into contiguous subgraphs of ~ 3000 nodes; edges crossing partitions are dropped and indices relabeled.
- Validation creation: the test side is split 50/50 by subgraph count into validation and final test sets.

GATv2-NS3 specifics. The hybrid model uses the methodology's fixed simulation schedule (trigger every 5 epochs; $\tau_0 = 0.5$, $\beta = 0.3$; $k = 2$ hops; 1000-node cap) without hyperparameter search; per-node QoS metrics are z-normalized and used only during training.

4 Results

Table 1 reports NSL-KDD test results.

Table 1: NSL-KDD Results.

Model	Accuracy	Precision	Recall	F1
GATv2-NS3	0.8051	0.6994	0.6327	0.6492
GATv2	0.8063	0.6964	0.6411	0.6522
GraphSAGE	0.7914	0.7247	0.5930	0.6196
GIN	0.7943	0.7266	0.5847	0.6114
MLP	0.7852	0.7043	0.5858	0.6001
RandomForest	0.7426	0.7810	0.4802	0.4899
LogisticRegression	0.7957	0.7070	0.6153	0.6309
XGBoost	0.7540	0.7774	0.4891	0.4967

4.1 NSL-KDD Performance Analysis

The best macro-F_1 is achieved by GATv2 (0.6522), closely followed by GATv2-NS3 (0.6492), with nearly identical accuracy (0.8063 vs. 0.8051). Logistic Regression ranks next ($F_1 = 0.6309$), ahead of GraphSAGE (0.6196), GIN (0.6114), and MLP (0.6001). Tree-based methods show high precision but poor recall (RandomForest: precision 0.7810, recall 0.4802; XGBoost: 0.7774/0.4891), resulting in $F_1 \approx 0.49$.

Overall, macro-F_1 scores cluster around 0.60–0.65 for strong baselines. GATv2-NS3 is competitive with GATv2 but does not surpass it in this single-run evaluation; both offer a better precision–recall balance than tree ensembles, which favor precision at the expense of recall.

4.2 Multi-Class Analysis

Figure 1 summarizes per-class macro-F_1 across all models. Majority classes are well handled (Normal, DoS), while minority classes (R2L, U2R) remain challenging.

Class-wise highlights (macro-F_1):

- Normal: 0.773–0.828 (best ≈0.828; GATv2/GATv2-NS3)
- DoS: 0.843–0.878 (best ≈0.878; GIN/XGBoost)
- Probe: 0.684–0.829 (best 0.8287; GATv2. GATv2-NS3: 0.8098; LogReg: 0.8253)
- R2L: 0.100–0.622 (best 0.6215; GATv2-NS3, slightly above GATv2: 0.6195; LogReg/GraphSAGE ≈0.50)
- U2R: 0.021–0.179 (best 0.1793; GIN; GATv2-NS3: 0.1421; GATv2: 0.1402; LogReg: 0.1589)

Takeaways:

- Both GATv2 and GATv2-NS3 deliver top performance on Normal and Probe, with GATv2 slightly higher on Probe and GATv2-NS3 slightly higher on R2L.
- All methods struggle on U2R due to extreme class scarcity (support=94), with the best F_1 still below 0.18.
- Tree models (RF/XGB) show high precision but poor recall on minority classes, yielding low F_1 for R2L/U2R.

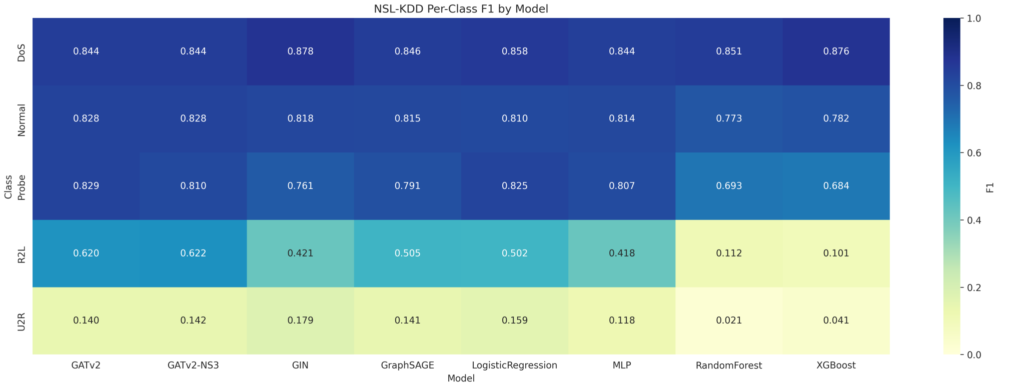

Figure 1: Per-class macro-F_1 heatmap (rows: classes; columns: models).

For error patterns, the confusion matrices (Figure 2) show residual confusion between R2L and Normal, and between U2R and other rare classes, consistent with their low F_1.

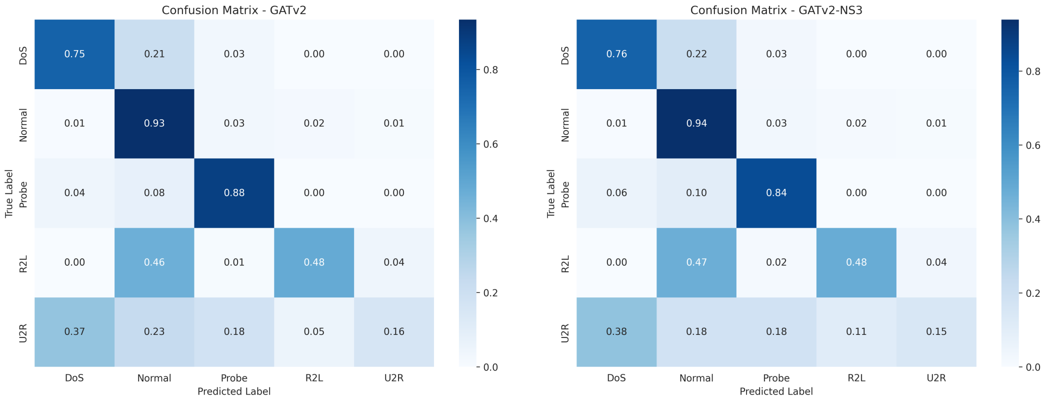

Figure 2: Confusion matrices for GATv2 and GATv2-NS3 on NSL-KDD (row-normalized).

4.3 Key Performance Patterns

On NSL-KDD, macro-F_1 concentrates around 0.60–0.65 for strong baselines. The best overall result is GATv2 ($F_1 = 0.6522$), closely followed by GATv2-NS3 ($F_1 = 0.6492$) with nearly identical accuracy (0.8063 vs. 0.8051). Logistic Regression ($F_1 = 0.6309$) outperforms GraphSAGE (0.6196), GIN (0.6114), and MLP (0.6001). Tree models show high precision but low recall (e.g., RandomForest: 0.7810/0.4802, XGBoost: 0.7774/0.4891), yielding $F_1 \approx 0.49$.

Simulation vs. baseline. GATv2-NS3 matches GATv2 on Normal/DoS and improves R2L slightly (0.6215 vs. 0.6195), but both struggle on U2R ($F_1 \approx 0.14$).

5 Discussion

5.1 Key Findings and Interpretations

Separability of prevalent classes. Majority classes (Normal, DoS) exhibit robust decision margins across architectures. Attention patterns for these classes show concentrated mass on a small subset of neighbors (low entropy), indicating that relational cues are reliable when behavioral signatures are frequent. This supports the premise that topology-aware models encode stable invariants for high-support traffic regimes.

Minority-class brittleness. Error topology concentrates on R2L and U2R. Confusion matrices reveal (i) systematic drift of R2L into Normal and (ii) diffuse U2R errors, consistent with attenuated feature contrast and severe class scarcity. The failure modes are thus attributional (insufficient discriminative representation) rather than purely statistical noise. Consequently, macro summaries obscure operational risk concentrated in rare categories.

Local effect of uncertainty-guided simulation. Self-focusing simulation modifies the geometry of hard regions by injecting QoS-derived structure but does not globally re-order models. The intervention acts as a targeted regularizer that sharpens boundaries where uncertainty is high (e.g., R2L neighborhoods), aligning with our design objective of behavior-grounded supervision. Absence of macro-F_1 gains in this setting is therefore expected: the mechanism optimizes local calibration and error localization rather than average accuracy.

Methodological implications. First, leakage-aware preprocessing and disjoint graph construction are necessary conditions for valid claims; they suppress optimistic bias and expose true minority-class difficulty. Second, reporting per-class metrics and confusion matrices should be standard in IDS, as macro means are weak surrogates for deployment risk. Third, future research should prioritize class-cost–aware optimization, stratified uncertainty sampling, and auxiliary QoS objectives to amplify simulation signal in rare regimes.

Limitations and scope. The present study evaluates a single dataset with a controlled training pipeline and may employ a feature-aware simulation fallback. These constraints bound external validity but sharpen methodological conclusions: uncertainty cues are informative for targeted supervision, and reliable evaluation protocols materially alter claimed effectiveness. Extending to additional corpora and full ns-3 bindings will enable a stronger test of simulation-grounded learning at scale.

5.2 Comparison with Literature and Implications

Our results (macro-$F_1 \approx 0.60$–0.65) are substantially below commonly reported $> 90\%$ accuracies on NSL-KDD, aligning with recent analyses attributing inflated scores to leakage and protocol artifacts [Kus et al., 2022]. This supports the need for leakage-aware preprocessing and evaluation. While attention-based GNNs remain competitive, improvements hinge on minority classes; future work should emphasize class-rebalancing, cost-sensitive objectives, and uncertainty-aware labeling or simulation for rare attacks.

Methodological impact. Report macro-F_1 with per-class metrics and confusion matrices; adopt leakage-aware splits and avoid reusing train-fitted encoders on test without safeguards. Simulation-driven training should be evaluated not only on global metrics but also on minority-class behavior.

Practical impact. With realistic protocols, practitioners should expect macro-F_1 near 0.60–0.65 on NSL-KDD-like data unless stronger imbalance remedies are applied. Uncertainty-guided simulation offers interpretable signals and targeted analysis, though its training overhead must be budgeted.

5.3 Limitations and Future Directions

Limitations.

- *Graph fidelity:* k-NN over tabular features induces artificial topology; rare classes (R2L/U2R) remain weakly separated.
- *Simulation realism:* feature-aware fallback under-approximates full ns-3 dynamics and may mute potential gains.

Future work.

- *Stronger rare-event learning:* cost-sensitive objectives, uncertainty-driven sampling, and auxiliary QoS heads.
- *Protocol generality:* leakage-aware evaluations across modern IDS corpora with per-class reporting.
- *Higher-fidelity simulation:* full ns-3 bindings and targeted interventions on hard (R2L/U2R) regions.

Broader impact. Methodological rigor (leakage control, per-class metrics, uncertainty analysis) is essential for credible IDS claims; self-focusing offers a general template for uncertainty-driven resource allocation beyond cybersecurity.

6 Conclusion

We presented GATv2-NS3, a hybrid IDS that couples attention-based graph learning with uncertainty-guided ns-3 simulation via *Self-Focusing Simulations*. Under a leakage-aware NSL-KDD protocol, results show that realistic macro-F_1 lies near 0.60–0.65; the hybrid is competitive with GATv2, providing localized gains (notably on R2L) without altering aggregate rankings. Per-class heatmaps and confusion matrices reveal that majority classes (Normal/DoS) are reliably separated, whereas rare classes (U2R, and to a lesser extent R2L) remain the dominant failure modes—underscoring that IDS reliability hinges on rare-event handling rather than overall capacity.

Methodological contribution. Beyond aggregate scores, we contribute a leakage-aware pipeline with per-class reporting, and a principled mechanism to inject behavior-grounded (QoS) signals only where uncertainty warrants it. This clarifies why macro metrics may plateau while error structure improves locally, and establishes a template for targeted supervision in IDS.

Implications. For research, standardized leakage-aware evaluation and per-class analyses should become first-class reporting. For practice, uncertainty-guided re-simulation offers interpretable diagnostics and selective attention to hard regions, better aligning model development with operational risk.

Limitations and future work. The present study uses a single dataset/split, k-NN–induced topology, and may employ a feature-aware simulation fallback; large-scale/real-time constraints are not evaluated. Future directions include: (i) leakage-aware studies across modern IDS corpora with per-class reporting, (ii) cost-sensitive/uncertainty-driven learning for rare events (and auxiliary QoS heads), (iii) full ns-3 bindings with controlled interventions on hard classes, and (iv) budgeted triggering and subgraph scheduling for scalability.

Overall, uncertainty-guided, simulation-grounded learning is a viable path to more reliable IDS, shifting attention from inflated aggregates to behaviorally meaningful improvements where it matters most.

Acknowledgments

We thank the organizers of Agents4Science 2025 for enabling transparent AI-driven scientific discovery. We acknowledge NSL-KDD maintainers, NS-3 simulation community, PyTorch/PyTorch Geometric and scikit-learn development teams. We appreciate the computational resources, open-source software community, and human collaborators who provided guidance throughout this AI-human research partnership. Finally, we thank the network security and graph ML communities for their foundational contributions.

References

Nsl-kdd dataset. `https://www.kaggle.com/datasets/hassan06/nslkdd`, 2009. Accessed: 2024.

Giuseppina Andresini, Annalisa Appice, and Donato Malerba. Insomnia: Towards concept-drift robustness in network intrusion detection. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, pages 111–122, 2021.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

Evan Caville, Wai Weng Lo, Siamak Layeghy, and Marius Portmann. Anomal-e: A self-supervised network intrusion detection system based on graph neural networks. In *IEEE International Conference on Communications (ICC)*. IEEE, 2022.

Communications Security Establishment and Canadian Institute for Cybersecurity. Cse-cic-ids2018 on aws. `https://www.unb.ca/cic/datasets/ids-2018.html`, 2018.

Salwa Elsayed, Khalil Mohamed, and Mohamed Ashraf Madkour. A comparative study of using deep learning algorithms in network intrusion detection. *IEEE Access*, 12:58851–58870, 2024.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1050–1059. PMLR, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pages 1321–1330. PMLR, 2017.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1024–1034, 2017.

Tom R Henderson and George F Riley. *ns-3 Network Simulator in Practice*. CreateSpace Independent Publishing Platform, 2020.

Sydney M Kasongo and Yanxia Sun. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *Journal of Big Data*, 7(1):1–20, 2020.

Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796, 2019.

Dominik Kus, Eric Wagner, Jan Pennekamp, Konrad Wolsing, Ina Berenice Fink, Markus Dahlmanns, Klaus Wehrle, and Martin Henze. A false sense of security? revisiting the state of machine learning-based industrial intrusion detection. *arXiv preprint arXiv:2205.09199*, 2022.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Joffrey L. Leevy and Taghi M. Khoshgoftaar. A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data. *Journal of Big Data*, 7, 2020. URL `https://api.semanticscholar.org/CorpusID:227155174`.

Gabriel Maciá-Fernández, José Camacho, Roberto Magán-Carrión, Pedro García-Teodoro, and Roberto Therón. UGR'16: A new dataset for the evaluation of cyclostationarity-based network idss. *Computers & Security*, 73:411–424, 2018.

Omid Madani, Sai Ankith Averineni, and Shashidhar Gandham. A dataset of networks of computing hosts. In *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics*, pages 100–104. ACM, 2022.

Nour Moustafa. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustainable Cities and Society*, 72:102994, 2021.

Nour Moustafa and Jill Slay. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6. IEEE, 2015.

Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *4th International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116, 2018.

Stanford Network Analysis Project (SNAP). Cisco secure workload networks of computing hosts. `https://snap.stanford.edu/data/cisco-networks.html`, 2024. Accessed: 2024.

Jacopo Talpini, Fabio Sartori, and Marco Savi. Enhancing trustworthiness in ml-based network intrusion detection with uncertainty quantification. *Journal of Reliable Intelligent Environments*, 10(4):501–520, 2024.

Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6. IEEE, 2009.

Münteha Tüzün and Pelin Angin. Network intrusion detection with incremental active learning. pages 344–353, 04 2024. ISBN 978-3-031-57941-7. doi: 10.1007/978-3-031-57942-4_33.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Limin Yang, Wenbo Guo, Qingying Hao, Arridhana Ciptadi, Ali Ahmadzadeh, Xinyu Xing, and Gang Wang. CADE: Detecting and explaining concept drift samples for security applications. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2327–2344, 2021.

Li Zhipeng, Zheng Qin, Kai Huang, Xiao Yang, and Shuxiong Ye. Intrusion detection using convolutional neural networks for representation learning. pages 858–866, 10 2017. ISBN 978-3-319-70138-7. doi: 10.1007/978-3-319-70139-4_87.