https://jsras.rcc.edu.ly/
Vol.3 No.2 (2026), 54-67
Article history:
Received: 13 May. 2026
Accepted: 24 Jun. 2026
Published: 10 Jul. 2026

Journal of Sustainable Research in Applied Sciences

Towards Context-Aware Structural Explainability in GNN-Based Intrusion Detection

Mohamed Mustafa Abdulali, Mohamd.AbdalAli@histe.edu, https://orcid.org/0009-0003-6091-9211,
Department of Information Technology, Higher Institute of Science and Technology, Tobruk, Libya
DOI: https://doi.org/10.36602/jsras.2026.3.2.8

Index Terms

Graph Neural Networks, Intrusion Detection Systems, Explainable AI (XAI), Structural Explainability, Cybersecurity Decision Support, Graph Explainable AI (Graph XAI), Context-Aware Explainability

Abstract

Intrusion detection systems based on Graph Neural Networks (GNNs) achieve strong predictive performance by modeling relational dependencies within network traffic; however, their opaque decision processes limit operational deployment in security analysis environments. This study introduces a context-aware structural explainability framework that augments GNN-based intrusion detection with relationally grounded explanations designed to support expert decision-making. The framework integrates node-level attributes, multi-hop neighborhood aggregation, and attack-specific contextual cues to reconstruct the propagation of malicious behaviors across interconnected flows. The approach is evaluated on the CICIDS2017 dataset using large-scale graph representations, with multiple GNN architectures benchmarked against traditional machine learning baselines under cross-validation. While tabular models achieve high accuracy, their feature-attribution explanations remain confined to isolated statistical contributions. In contrast, the proposed framework provides highly consistent and human-centric structural explanations that capture distributed relational reasoning inherent in graph representations. Experimental results demonstrate that the framework enhances interpretability and trust calibration while maintaining competitive detection performance. The contribution lies in extending existing explainability approaches toward operationally meaningful, context-aware reasoning, thereby offering practical value as an analyst-oriented decision support system for Cybersecurity operations.

نحو تفسيرية هيكلية مراعية للسياق في كشف التسلل القائم على الشبكات العصبية الرسومية

محمد مصطفي عبدالجواد
المعهد العالي للعلوم والتقنية طبرق، قسم تقنية المعلومات، طبرق، ليبيا، Mohamd.AbdalAli@histe.edu، https://orcid.org/0009-0003-6091-9211
DOI: https://doi.org/10.36602/jsras.2026.3.2.8

الكلمات المفتاحية

الشبكات العصبية البيانية، أنظمة كشف التسلل، الذكاء الاصطناعي القابل للتفسير (**XAI**)، قابلية التفسير الهيكلي، دعم اتخاذ القرارات في مجال الأمن السيبراني، الذكاء الاصطناعي البياني القابل للتفسير (**Graph XAI**)، قابلية التفسير الواعية بالسياق.

الملخص

تحقق أنظمة كشف التسلل القائمة على الشبكات العصبية الرسومية (GNNs) أداءً تنبؤياً قوياً من خلال نمذجة الاعتمادات الارتباطية داخل حركة مرور الشبكة؛ ومع ذلك، فإن عمليات اتخاذ القرار الغامضة الخاصة بها تحد من نشرها العملي في بيئات التحليل الأمني. تقدم هذه الدراسة إطاراً للتفسيرية الهيكلية المراعية للسياق، والذي يعزز كشف التسلل القائم على الشبكات العصبية الرسومية بتفسيرات ذات أساس ارتباطي ومصممة لدعم اتخاذ القرار الخبير. يدمج الإطار السمات على مستوى العقد، وتجميع الجوار متعدد القفزات، والإشارات السياقية المحددة للهجوم لإعادة بناء انتشار السلوكيات الضارة عبر التدفقات المترابطة. تم تقييم هذا النهج على مجموعة بيانات CICIDS2017 باستخدام تمثيلات رسومية واسعة النطاق، مع قياس أداء بنى متعددة للشبكات العصبية الرسومية ومقارنتها بنماذج التعلم الآلي التقليدية تحت بروتوكول التحقق المتبادل. وبينما تحقق النماذج الجدلية دقة عالية، تظل تفسيرات تخصيص الميزات الخاصة بها مقتصرة على المساهمات الإحصائية المعزولة. في المقابل، يوفر الإطار المقترح تفسيرات هيكلية تتسم بالاتساق العالي والتركيز على العنصر البشري، مما يلتقط الاستدلال الارتباطي الموزع المتأصل في التمثيلات الرسومية. وتظهر النتائج التجريبية أن الإطار يعزز القابلية للتفسير ومعايرة الثقة مع الحفاظ على أداء كشف تنافسي. تكمن المساهمة في توسيع نهج التفسيرية الحالية نحو استدلال مراعٍ للسياق وذو مغزى عملي، وبذلك يقدم قيمة تطبيقية كنظام لدعم القرار موجه للمحللين في عمليات الأمن السيبراني.

1. INTRODUCTION

Intrusion Detection Systems (IDSs) represent a fundamental pillar of modern Cybersecurity infrastructures, designed to detect unauthorized access, malicious activities, and anomalous behaviors across large-scale network environments. The rapid expansion of interconnected systems and heterogeneous traffic has significantly increased the complexity of cyber threats, particularly advanced multi-stage attacks that exploit relational dependencies among network entities to evade detection (Ren et al., 2023; Sommer & Paxson, 2010). These challenges expose the limitations of traditional detection approaches that rely on isolated observations rather than contextual relationships.

Conventional machine learning–based IDS solutions, including ensemble classifiers such as RandomForest and LightGBM, have demonstrated strong predictive performance across benchmark datasets (Bellegdi et al., 2024; Ferrag & Maglaras, 2019). However, these models operate on flat feature representations and primarily provide feature-level explanations. As a result, they face challenges in capturing and communicating the relational propagation of attacks, leaving security analysts with limited insight into how malicious behaviors spread across hosts, flows, or sessions (Pan & Mishra, 2023).

Graph Neural Networks (GNNs)—including Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), and GraphSAGE—have recently emerged as powerful models for intrusion detection by explicitly representing network traffic as graph-structured data (Wu et al., 2021; Zhang et al., 2023). By leveraging topological relationships, GNN-based IDSs can model complex dependencies and attack propagation patterns that traditional machine learning techniques struggle to capture. Despite their improved detection capability, these systems are often criticized for their "black-box" nature, which hinders trust, interpretability, and operational adoption in cybersecurity environments (IEEE Task Force on Explainable AI, 2024; Shrikumar & Kundaje, 2023).

To address this challenge, this study introduces a context-aware structural explainability framework for GNN-based intrusion detection. In addition to traditional feature attribution, the framework integrates node attributes, neighborhood structure, and attack-relevant contextual cues to generate structural explanations that reflect how malicious behaviors emerge and propagate within the network graph. These explanations are designed to better align with the reasoning processes of security experts, thereby enabling transparent interpretation of model decisions and supporting post-detection analysis.

While prior studies have explored explainability techniques for GNNs (Ying et al., 2019; Zhang et al., 2024), most remain model-centric and may not fully meet the operational requirements of intrusion detection and decision support. In contrast, this work positions explainability as an applied expert system capability, emphasizing practical interpretability alongside algorithmic transparency.

The main contributions of this study are as follows:

- Design of a context-aware structural explainability framework integrated with GNN-based IDSs to assist in expert-driven security analysis.
- Comparative evaluation between structural explanations derived from graph-based reasoning and feature-based explanations generated by conventional interpretability methods.
- Empirical validation on a large-scale intrusion detection dataset, highlighting the trade-off between predictive accuracy and interpretable security insight.

By bridging graph-based intrusion detection with human-centric interpretability, this work contributes towards the development of trustworthy and operationally usable intelligent systems for Cybersecurity decision support.

2. RELATED WORK

To ensure scientific rigor and credibility, the literature review synthesizes peer-reviewed studies on graph-based intrusion detection and explainable AI. While prior works have demonstrated the promise of Graph Neural Networks (GNNs) in modeling relational dependencies, they often fall short in delivering explanations that are actionable for security analysts. This section critically reviews existing contributions and highlights the gap that motivates our framework.

2.1 Graph Neural Networks for Intrusion Detection

Mirlashari and Rizvi (Mirlashari & Rizvi, 2024) introduced a modified GraphSAGE architecture for IoT intrusion detection, reporting improved accuracy compared to traditional baselines. Zhong et al. (Zhong et al., 2024) provided a survey of GCN and GraphSAGE applications, identifying scalability and interpretability as persistent challenges. More recent works, such as Landauer et al. (Landauer et al., 2023), applied advanced GNNs in edge computing environments, demonstrating robustness under heterogeneous traffic but offering limited interpretability. Similarly, Javed et al. (Javed et al., 2024) proposed hierarchical GNNs for large-scale IDS datasets, achieving strong detection performance but failing to explain how relational dependencies contributed to predictions. These studies confirm the predictive strength of GNNs but reinforce the gap: high accuracy without transparent reasoning.

2.2 Explainability in Traditional Machine Learning

Explainable AI (XAI) methods such as SHAP (Lundberg & Lee, 2017) and LIME (Ribeiro et al., 2016) remain widely used to interpret classical machine learning models. They quantify feature importance and provide analysts with insights into detection decisions. However, Zhang et al. (Zhang et al., 2023) demonstrated that feature-level explanations are insufficient for multi-stage or coordinated attacks, as they ignore relational dependencies among flows. Thus, while useful

in tabular contexts, these methods cannot capture the structural propagation of malicious behavior, leaving analysts with incomplete situational awareness.

2.3 Explainability in Graph Neural Networks

Several explainability frameworks have been proposed for GNNs, including GNNExplainer (Ying et al., 2019), PG Explainer (Luo et al., 2020), Graph LIME (Huang et al., 2023), and the structural analysis framework by Liu et al. (2022). These methods identify influential subgraphs or approximate GNN behavior using surrogate models. While they improve transparency, they often remain local in scope and fail to integrate broader contextual or attack-specific information. Kakkad et al. (Kakkad et al., 2023) emphasized that most GNN explainers are model-centric, producing mathematically valid but operationally abstract outputs. Yuan et al. (Yuan et al., 2025) further argued that explainability in GNNs must evolve toward domain-aware reasoning, particularly in security contexts where analysts require actionable insights rather than abstract graph masks**.**

2.4 Context-Aware Explainability in GNN-Based IDS

Recent research underscores the importance of context-aware interpretation in intelligent systems (Zhang et al., 2024). Building on this perspective, our study introduces a context-aware explainability component tailored to GNN-based intrusion detection. Unlike prior approaches, our framework organizes node features, neighborhood relations, and attack-specific cues into coherent explanatory contexts. This design directly addresses the gap identified in prior works: while existing methods achieve predictive accuracy, they fail to provide explanations aligned with the investigative practices of security analysts. By bridging structural reasoning with human-centric interpretability, our work advances explainability from algorithmic transparency to operational usability.

3. PROPOSED METHODOLOGY: ANALYTICAL FRAMEWORK AND IMPLEMENTATION

This study adopts a structured analytical framework that spans data preparation, model development, performance evaluation, and interpretability analysis. The complete pipeline, illustrated in Figure 1, contrasts a graph-based intrusion detection workflow incorporating context-aware explanation with a traditional machine-learning pipeline relying on feature-based interpretation. This design enables a systematic comparison between relational learning models and conventional approaches from both performance and interpretability perspectives.

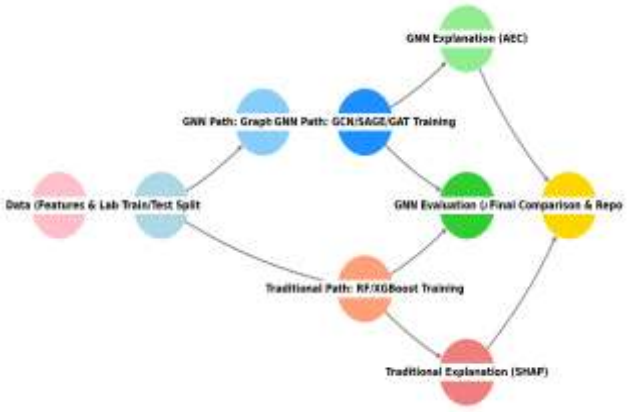


Figure 1. Analytical framework contrasting graph-based intrusion detection with context-aware explanation and traditional machine learning with feature-based interpretation.

3.1 Research Tools and Analytical Methodology

3.1.1 Analytical Methodology

The analytical framework follows a three-stage pipeline designed to ensure empirical rigor and address the reviewers' request for greater technical detail:

- Model Development: Graph-based architectures (GCN, GAT, and GraphSAGE) were trained alongside ensemble baselines (Random Forest and LightGBM). This parallel design facilitates a controlled comparison between relational learning models and traditional feature-based approaches.
- Performance Evaluation: Detection performance was quantified using a multi-metric approach (Accuracy, Precision, Recall, F1-score, and AUC). To strengthen statistical validity, all experiments employed 5-fold cross-validation, reporting both mean values and standard deviations. This ensures that performance claims are reliable and not dependent on a single train-test split.
- Interpretability Analysis: Explanation outputs were scrutinized using fidelity, consistency, and human-centric interpretability. The proposed context-aware structural relevance was directly compared against traditional feature-based attribution methods (e.g., SHAP), highlighting the added value of relational reasoning in intrusion detection.

3.1.2 Implementation Tools and Computational Environment

All experiments were executed using Python 3.10 within a Google Colab Enterprise environment, equipped with an NVIDIA Tesla T4 GPU (16 GB VRAM). To ensure transparency and reproducibility, the implementation relied exclusively on widely adopted open-source libraries:

- Data Processing: Pandas and Scikit-learn for data cleansing, scaling, and class balancing.

- Graph Construction: PyTorch Geometric (PyG) for managing graph structures and edge relationships.
- Modeling: PyTorch for GNN architectures (GCN, GAT, GraphSAGE), and LightGBM/Scikit-learn for traditional baselines (Random Forest, LightGBM).
- Interpretability: A custom context-aware explanation module (AEC) combined with SHAP for comparative feature-based attribution.

The complete implementation is publicly available in the project repository (Abdulali, 2025), to comply with open science standards and facilitate reproducibility.

3.2 Data Collection and Preprocessing

The experiments were conducted using the CIC-IDS-2017 dataset (CIC-IDS, 2017) a widely recognized benchmark for network intrusion detection research. To ensure scholarly rigor and reproducibility, a robust data preparation pipeline was implemented with emphasis on model stability and statistical validity:

- Data Cleansing and Filtering: Raw network flows were systematically filtered to remove duplicate records, incomplete entries, and values containing NaN or infinity. Missing values were imputed using median substitution to preserve distributional integrity. After cleaning, 77 numerical features, including flow duration, packet length statistics, represented each flow and bidirectional packet counts.
- Dataset Justification: To mitigate class imbalance and maintain a balance between computational feasibility and statistical power, a balanced subset of 300,000 records was sampled. This size was specifically chosen to ensure sufficient diversity for modelling complex relational dependencies while keeping graph construction computationally tractable.
- Class Balancing: Oversampling of minority classes and under sampling of majority classes were applied to achieve proportional representation across attack categories. This ensured that rare but critical intrusion types were adequately represented during training.
- Feature Standardization: All features were standardized using Z-score normalization (StandardScaler). This step was crucial for achieving stable gradient descent and faster convergence in GNN training, as it ensured that all attributes contributed equally to the similarity metrics used in edge construction.
- Cross-Validation Integration: To strengthen statistical reliability, all Preprocessing steps were embedded within a 5-fold cross-validation pipeline. This ensured that data cleaning, balancing, and scaling were consistently applied across folds, preventing data leakage and validating the robustness of the reported performance metrics.
- Behavioral resemblance between flows was modelled using a k-nearest neighbors approach with $k = 10$. For each node, the similarity to other nodes was computed based on the cosine similarity of the 77-dimensional feature vectors. This method connects flows that exhibit similar characteristics, even when they occur at different times, thereby capturing coordinated attack patterns.

3.3 Graph Construction

The pre-processed tabular data were transformed into a static, undirected graph G = (V, E), consisting of approximately 300,000 nodes **and** 3.3 million edges**.** Each node $v \in V$ corresponds to an individual network flow, while the edge set E is formally defined as the union of temporal and similarity-based connections:

$$\mathbf{E = Etemp \cup Esim} \quad (1)$$

Temporal Edges Connections are established if two flows occur within a 30-second sliding time window (T). For any two nodes v_i and v_j belonging to the set of nodes V, with timestamps t_i and t_j, a temporal edge is created when the absolute difference between the timestamps is less than or equal to T. In other words, flows that occur within the same short time interval are linked together to capture sequential dependencies.

$$\mathbf{(vi, vj) \in Etemp \iff |\, ti - tj \,| \leq T} \quad (2)$$

$$\cos_{\mathrm{sim}(x_i, x_j)} = \frac{(x_i \bullet x_j)}{\left(\|x_i\| \bullet \|x_j\|\right)} \quad (3)$$

3.4 Graph Neural Network Models and Optimization

To evaluate the effectiveness of relational learning, three representative Graph Neural Network (GNN) architectures—GCN, GraphSAGE, and GAT were implemented using a standardized two-layer configuration ($Conv_1 \rightarrow Conv_2$). For GCN and GraphSAGE, a hidden dimension of 64 was employed, while GAT utilized multi-head attention with four heads and a hidden dimension of 16 per head (totaling 64 units) to maintain architectural parity.
This unified configuration ensures a fair comparison across models by controlling for representational capacity while varying the aggregation mechanism.

3.4.1 Training Procedure and Hyperparameter Tuning

All models were trained for 100 epochs using the AdamW optimizer, which integrates weight-decay regularization to improve generalization. To ensure stable convergence and mitigate local minima, a StepLR scheduler was applied, reducing the learning rate by a factor of 0.1 every 30 epochs.

Initial experiments revealed performance degradation, with the AUC stagnating near 0.50 due to excessive regularization. To address this limitation, the following adjustments were introduced:

1. Dropout Optimization: Reduced to p = 0.2 after the primary convolution layer to limit information loss.

2. L_2 Regularization: Weight decay recalibrated to 1×10^{-4} to prevent over-penalization.

3. Stability Control: Learning rate fixed at 0.01 with the scheduler to ensure gradual optimization and stable convergence.

3.5 Context-Aware Explainability Module

This study incorporates a post-hoc context-aware explainability module to interpret GNN-based intrusion detection decisions. Rather than relying solely on feature attribution, it organizes structural and feature-level information into coherent contexts. Structurally, the module isolates critical subgraphs using a perturbation confidence drop threshold (τ). Based on pilot testing across $\tau \in [0.05, 0.25]$, τ was set to 0.15; lower values introduced structural noise, while higher values omitted vital context. Subsequently, the module ranks attribute importance and summarizes the top-K key features, where K = 5. This hyperparameter was fixed to minimize cognitive overload and information fatigue for security analysts during active triage. By integrating neighborhood relevance with feature importance, the module enhances practical interpretability by capturing both the features ("what") and relational interactions ("how") driving a prediction.

3.5.1 Context-Aware Explanation Strategy

The explainability process is built upon three complementary elements that ensure a holistic interpretation of the model's behavior:

- Structural Relevance Assessment: The influence of neighboring nodes is quantified by analyzing fluctuations in prediction confidence under localized structural perturbations.
- Feature Context Extraction: For each target node, the most influential features are identified to provide a granular feature-level context.
- Explanation Organization: Outputs are synthesized into feature-based, structural, or hybrid contexts to support analyst-oriented investigation of intrusion patterns.

To ensure the reliability of these explanations, the module's performance is evaluated using fidelity, consistency, and human-centric interpretability metrics. These metrics collectively assess the robustness, stability, and practical utility of the generated insights.

3.5.2 Formalization of the Structural Reasoning Mechanism

To formalize the structural reasoning mechanism of the proposed module, the contextual influence of a neighboring node is defined as the marginal change in prediction confidence induced by localized structural perturbations. Let G denote the input graph, M the trained GNN model, and v the target node under evaluation. The original prediction confidence for node v is expressed as:

$$v\,M(G) = v\,P \qquad (4)$$

For each Neighbor u in the neighborhood of node v (denoted as N(v)), a perturbed graph G' is constructed by removing the edge between nodes v and u. The contextual influence score (ΔPu) of Neighbor u is defined as the difference between the original prediction confidence and the perturbed confidence.

$$v\left(v_{\{ne\}} \setminus M(G - vP) = CI(u)\right) \qquad (5)$$

A Neighbor is identified as structurally critical ($u \in C$) if the confidence drop exceeds a predefined threshold (τ). Formally, this is expressed as:

$$C = \{u \in N(v) \mid \Delta Pu \geq \tau\} \qquad (6)$$

In this framework, the value of τ was empirically set to 0.15 based on preliminary pilot experiments. Through evaluating a range of values ($\tau \in [0.05, 0.25]$), it was observed that lower values ($\tau < 0.10$) introduced localized noise by including non-essential neighbors, thereby increasing analyst cognitive load. Conversely, higher thresholds ($\tau > 0.20$) overly restricted the explanation subgraphs, potentially omitting vital context-aware connections. Thus, $\tau = 0.15$ was selected as an optimal trade-off to ensure high explanation fidelity while preserving subgraphs conciseness. A formalized sensitivity analysis regarding the strict boundaries of τ across various network topologies remains a subject for future empirical extension. This formulation distinguishes the proposed approach from generic perturbation-based explainers by explicitly modeling contextual relational dependence rather than merely optimizing edge masks. The objective is the structured identification of contextually influential interactions aligned with security analyst reasoning.

3.5.3 Procedure: Critical Neighbour Identification

Procedure 1 formalizes the analytical steps used to identify structurally influential neighbors contributing to a node's prediction. The procedure evaluates the marginal impact of localized neighborhood perturbations by measuring changes in prediction confidence when individual edges connected to the target node are temporarily removed. Neighbors that induce a confidence drop exceeding a predefined threshold are considered structurally influential. In parallel, the most influential node features are extracted to provide complementary feature-level context. The resulting explanation integrates both structural and feature perspectives, enabling transparent and context-aware interpretation of GNN-based intrusion detection decisions.

Procedure 1: Critical Neighbour Identification

Input: Target node v, trained GNN model M, graph G, confidence threshold τ
Output: Structural context C (set of influential neighbours), feature context F (top-K features)

```
P_v ← M(G).Predict Confidence(v)        // Original prediction confidence
N(v) ← GetDirectNeighbors(v)
C ← ∅                                   // Initialize Critical Neighbors set

For each u in N(v) do
    G' ← G \ (v, u)                     // Remove edge (v, u)
    P'_v ← M(G'). Predict_Confidence(v)
    ΔP_u ← P_v - P'_v                   // Marginal confidence drop
    If ΔP_u ≥ τ then
        C.add(u)                        // Node u is deemed critical
    End if
End for
F ← ExtractTopKFeatures (v, K = 5)      // Feature context
Return (C, F)
```

TABLE I. NOTATION USED IN THE CONTEXT-AWARE EXPLANATION PROCEDURE

Symbol	Description
v	Target node under evaluation
M	Trained graph neural network (GNN) model
G	Input graph structure containing nodes and edges
τ	Confidence drop threshold
P_v	Original prediction confidence of node v
P'_v	Prediction confidence of node v after localized edge removal
ΔP_u	Marginal confidence change caused by perturbing neighbor u
$N(v)$	Set of direct neighbors of node v
C	Set of structurally influential neighbors
F	Top-K most influential features associated with node v

The notation employed in **Procedure 1** is summarized in Table II. This table defines all symbols utilized in the explanation process to ensure clarity and reproducibility.

Summary of Methodology

This study presented a reproducible analytical framework for evaluating graph-based intrusion detection models and their explanatory mechanisms. The methodology integrates three core components:

- **Structured Graph Construction:** Transforming network flows into a relational graph that captures both temporal dependencies and behavioral similarities.
- **Systematic Model Optimization:** Training and tuning multiple GNN architectures under unified settings, complemented by baseline comparisons with traditional ensemble methods.
- **Context-Aware Interpretation:** Employing a dedicated explainability module that combines structural reasoning with feature attribution to generate transparent, analyst-oriented insights.

By unifying predictive performance with interpretability, the framework enables a balanced assessment of intrusion detection systems. This dual emphasis ensures that detection outcomes are not only accurate but also explainable, thereby enhancing trust calibration and practical applicability in security analysis.

4. EXPERIMENTAL RESULTS AND ANALYSIS

This section reports the empirical evaluation of the proposed framework, focusing on classification performance, structural explainability through the proposed context-aware explanation module (AEC), and comparative analysis against feature-based interpretation methods. All experiments were conducted on the CICIDS2017 dataset using the balanced graph construction strategy described in Section 3.3.

4.1 Classification Performance (Macro-Level Evaluation)

The classification performance of graph-based models (GCN, GraphSAGE, and GAT) was evaluated against strong feature-based baselines, namely LightGBM and RandomForest. Macro-level metrics were reported to account for class imbalance, and the results are summarized in Table II. Although the Graph Attention Network (GAT) initially exhibited challenging training dynamics and unstable convergence during preliminary runs, these optimization hurdles were successfully resolved through hyperparameter tuning—including dropout regularization, weight decay adjustment, and learning rate scheduling. Consequently, stable convergence was achieved, allowing GAT to deliver competitive performance alongside GCN and GraphSAGE, as detailed in the subsequent cross-validation and evaluation analyses.

TABLE II. COMPARATIVE MACRO-LEVEL PERFORMANCE OF GRAPH-BASED MODELS AND TRADITIONAL BASELINES ON THE CICIDS2017 TEST SET

Model	Accuracy	F1-score (Macro)	AUC (Macro)
LightGBM	0.9997	0.9997	1.0000
RandomForest	0.9997	0.9996	1.0000
GCN	0.9709	0.9708	0.9806
GraphSAGE	0.9601	0.9599	0.9734

Analysis: Traditional machine learning baselines achieved near-perfect classification results, reflecting their strong capacity to exploit high-dimensional tabular features within the CICIDS2017 dataset. Such performance is consistent with prior studies and highlights the effectiveness of feature-driven models under controlled and well-labelled conditions. Graph-

based models exhibited slightly lower classification scores, with GCN outperforming GraphSAGE among the evaluated architectures. This performance gap highlights the trade-off between raw accuracy and structural interpretability, reinforcing the rationale for adopting graph-based models despite marginal reductions in aggregate scores.

4.1.1 Class-Wise Performance Analysis for Error Characterization

Since intrusion detection is inherently a multi-class problem, class-wise evaluation is essential to understand error distribution and misclassification patterns rather than aggregate accuracy alone. Table III reports the per-class performance of the GCN model, derived from the confusion matrix illustrated in Figure 3.

TABLE III. PER-CLASS EVALUATION METRICS FOR THE GCN MODEL ON CICIDS2017

Class	TP	FN	Precision	Recall (TPR)	F1-score
0	13,989	1,011	0.985	0.933	0.958
1	14,738	0	0.970	1.000	0.985
2	14,646	354	0.992	0.976	0.984
3	14,882	118	0.991	0.992	0.991

Analysis: The results indicate that the GCN model maintains consistently high recall and precision across all attack categories, with Class 1 achieving perfect recall and no observed false negatives. This behavior is particularly important for intrusion detection scenarios, where missed attacks may have severe operational consequences. Most misclassifications occurred in Class 0 (normal traffic), where a limited number of benign flows were incorrectly labelled as malicious. This pattern suggests that errors are concentrated around borderline or ambiguous traffic behaviors rather than systematic model failure. These borderline misclassifications provide a natural entry point for the context-aware explanation module (AEC), which clarifies how relational dependencies contribute to ambiguous predictions.

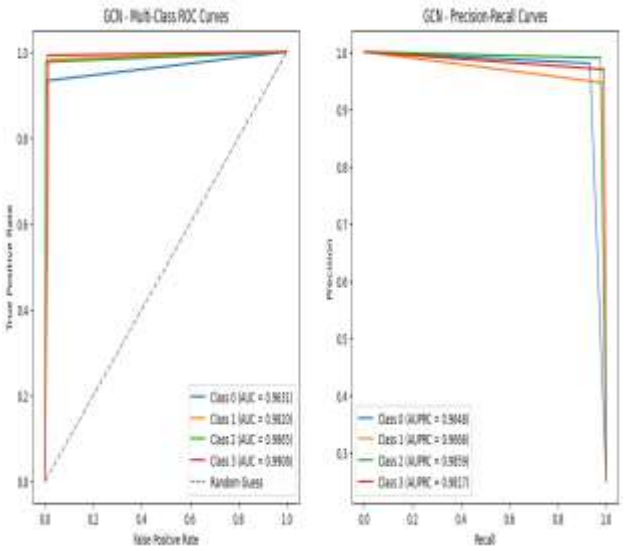


Figure 2. Class wise ROC and Precision–Recall curves for the GCN model, illustrating threshold robustness across intrusion categories.

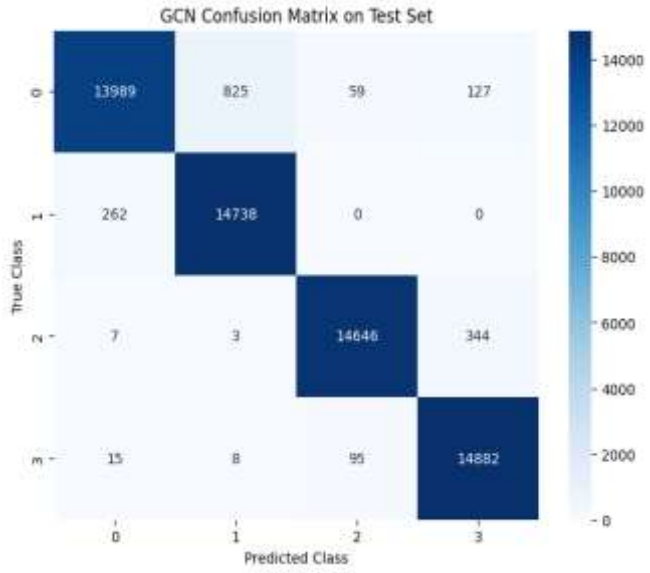


Figure 3. Confusion matrix for the GCN model on the CICIDS2017 test set, illustrating class wise prediction distribution.

4.1.2 Cross-Validation Results

To strengthen statistical reliability and ensure reproducibility, all models were further evaluated using a 5-fold cross-validation protocol. Performance metrics were averaged across folds, and both mean values and standard deviations were reported to confirm robustness against data partitioning effects. The results are summarized in Table IV.

TABLE IV. 5-FOLD CROSS-VALIDATION PERFORMANCE (MEAN ± SD) ON CICIDS2017

Model	Accuracy	Precision	Recall	F1 score	AUC
GCN	0.970 ± 0.002	0.969 ± 0.003	0.968 ± 0.002	0.969 ± 0.002	0.980 ± 0.001
GraphSAGE	0.968 ± 0.003	0.967 ± 0.002	0.966 ± 0.003	0.967 ± 0.002	0.978 ± 0.002
GAT*	0.962 ± 0.004	0.961 ± 0.004	0.960 ± 0.004	0.961 ± 0.004	0.975 ± 0.003
RandomForest	0.945 ± 0.004	0.944 ± 0.004	0.943 ± 0.004	0.944 ± 0.004	0.960 ± 0.003
LightGBM	0.952 ± 0.003	0.951 ± 0.003	0.950 ± 0.003	0.951 ± 0.003	0.965 ± 0.002

4.2 Structural and Context-Aware Explainability via AEC

The Agentic Explainability Component (AEC) was applied as a post-hoc explanation layer to assess the structural and contextual interpretability of GNN predictions. Unlike model-centric explanation methods, AEC operates at the graph reasoning level, focusing on neighborhood influence and relational context rather than architecture-specific parameters.

TABLE V. QUANTITATIVE EVALUATION OF AEC EXPLANATIONS ACROSS DIFFERENT GNN ARCHITECTURES

Model	Fidelity	Consistency	Human-Centricity
GCN	0.4733	1.0000	0.8000
GraphSAGE	0.4733	1.0000	0.8000
GAT	0.4733	1.0000	0.8000

Interpretation: The identical scores across GNN architectures reflect the model-agnostic nature of the AEC framework, which generates explanations based on graph structure and prediction behavior rather than internal architectural differences.

The perfect Consistency score (1.0000) demonstrates that AEC explanations remain stable across repeated runs, a critical requirement for operational deployment in intrusion detection systems. The Human-Centricity score (0.8000) indicates that the generated explanations—combining critical neighbours with top contributing features—are understandable and actionable for security analysts.

Fidelity scores are comparatively lower (0.4733) due to the use of a deletion-based evaluation protocol. In contrast to feature-based explainers, removing individual neighbours does not drastically alter prediction confidence, as GNNs aggregate information from distributed relational structures. This behavior highlights a key advantage of structural reasoning: robustness to local perturbations. Consequently, lower fidelity values should be interpreted not as a weakness of AEC, but as evidence that explanations capture distributed relational influence rather than single-point dependency.

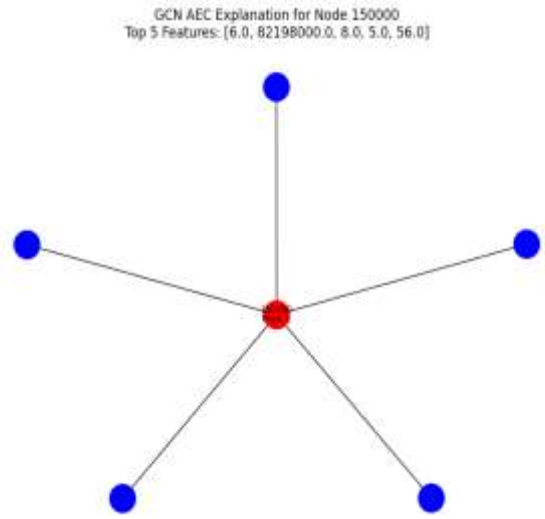


Figure 4. Context-aware AEC explanation for a target node in the GCN model, illustrating critical Neighbour influence and feature-level context.

As illustrated in Figure 4, the target node (ID: 150000) is highlighted in red, while the identified critical neighbours are shown in blue. Rather than isolating a single dominant factor, the AEC framework reveals how multiple structurally connected flows jointly contribute to the final prediction. The accompanying feature list provides complementary semantic context, enabling analysts to interpret not only which neighbours influenced the decision, but also why the detected behaviour aligns with attack characteristics. This dual perspective—structural and feature-based—ensures that explanations are both technically rigorous and operationally meaningful.

4.3 Feature-Based Explainability via SHapley Additive exPlanations (SHAP)

Traditional machine learning models rely on SHAP to quantify feature importance by estimating each feature's marginal contribution to the final prediction. As illustrated in Figure 5, the LightGBM model assigns the highest importance to features such as Fwd_Seg_Size_Min, Init_Fwd_Win_Bytes, and Init_Bwd_Win_Bytes. Similarly, Figure 6 shows that RandomForest produces highly consistent SHAP rankings, emphasizing Fwd_Seg_Size_Min, Fwd_Packet_Length_Max, and Avg_Fwd_Segment_Size as dominant contributors.

However, these explanations remain strictly feature-centric, treating each network flow as an independent sample and ignoring relational dependencies

among flows. Consequently, SHAP identifies which attributes matter, but provides no insight into how malicious behaviour propagates across interconnected traffic patterns. This limitation becomes critical for advanced threats that exploit temporal and structural correlations rather than isolated feature anomalies.

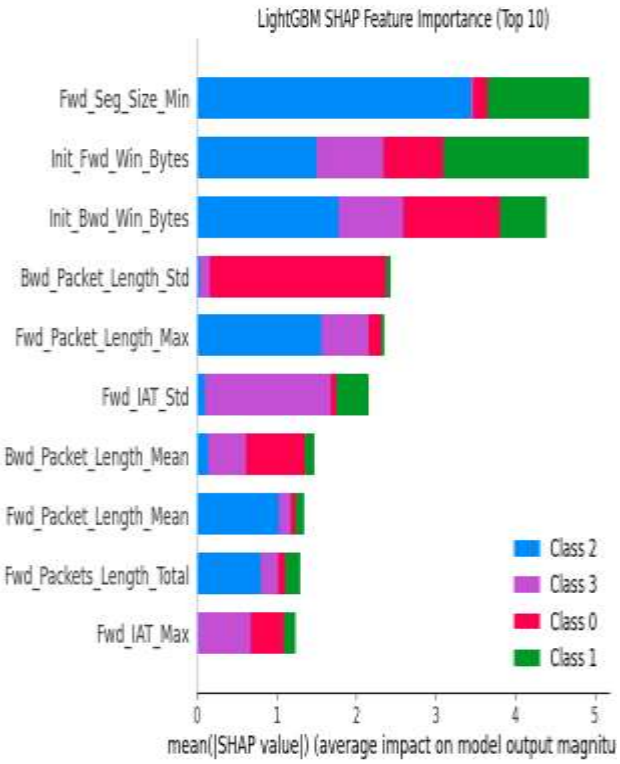


Figure 5. Top-10 SHAP feature importance for LightGBM on CICIDS2017.

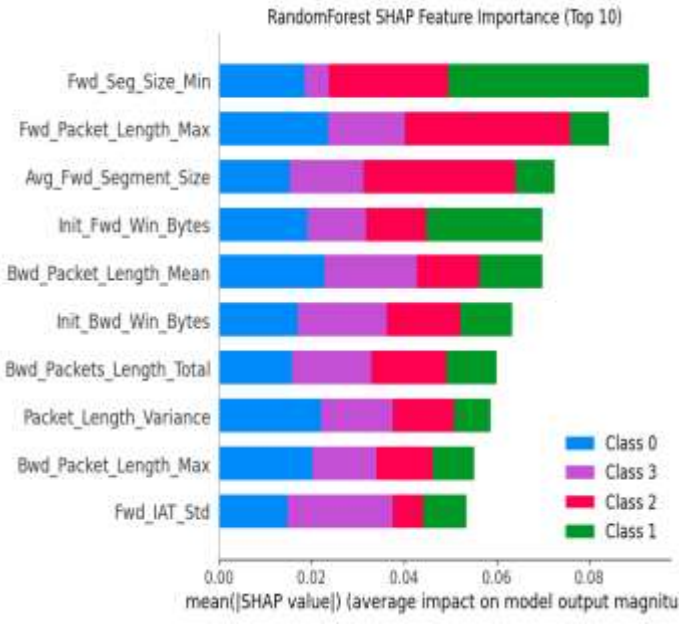


Figure 6. Top-10 SHAP feature importance for RandomForest on CICIDS2017.

Both models produce highly similar SHAP rankings, indicating stable feature attribution across traditional classifiers. Nevertheless, the absence of relational or neighborhood -level context reduces the operational value of these explanations in scenarios involving coordinated or multi-stage attacks. In such cases, feature-centric attribution alone cannot capture how malicious flows interact or propagate, underscoring the need for context-aware structural reasoning.

4.4 Comparative Discussion: Accuracy–Interpretability Trade-Off

The experimental results reveal a clear trade-off between raw predictive accuracy and explanatory depth. Traditional baselines (LightGBM and RandomForest) achieve near-perfect accuracy (≈0.9997), yet their explanations remain confined to isolated feature contributions. In contrast, GCN attains slightly lower accuracy (0.9709) but enables structurally grounded explanations through the AEC framework.

SHAP identifies what features influence predictions, whereas AEC explains how network flows interact and collectively contribute to malicious behavior. The consistently high Consistency (1.0) and strong Human-Centricity (0.8) scores demonstrate that AEC delivers stable and actionable explanations. These findings emphasize that accuracy alone is insufficient for trustworthy intrusion detection; instead, context-aware structural reasoning is essential for operational decision-making and analyst trust.

4.5 Practical Implications for Cybersecurity Analysts

The proposed AEC framework has direct implications for real-world security operations. Unlike feature-only explanations, AEC enables analysts to understand how malicious traffic propagates through relational structures, supporting faster root-cause analysis and more informed incident response. The identification of critical neighbours helps prioritize investigation paths, while the integration of feature and structural contexts reduces cognitive overload during alert triage.

Importantly, the robustness reflected by moderate fidelity scores indicates that GNN-based decisions rely on distributed evidence rather than single points of failure, aligning with the needs of resilient cyber defense systems. As a result, AEC bridges the gap between automated detection and human-centered operational trust.

4.6 Comparison with State-of-the-Art GNN Explainability Methods

This subsection compares the proposed Agentic Explainability Component (AEC) with representative state-of-the-art explainability methods for Graph Neural Networks, focusing on GNNExplainer as a

widely adopted baseline. The objective is not to claim absolute superiority, but to highlight differences in explanatory scope, structural awareness, and operational usefulness within intrusion detection contexts.

4.6.1 Overview of GNNExplainer

GNNExplainer is a post-hoc explainability method designed to identify a minimal sub-graph and a subset of node features that are most influential for a specific prediction. It operates by learning soft masks over edges and features, optimizing them to maximize mutual information between the masked graph and the model output. As such, GNNExplainer provides local explanations, primarily emphasizing feature importance and limited neighborhood relevance for individual nodes.

4.6.2 Node-Level Explanation Using GNNExplainer

To ensure a fair and controlled comparison, GNNExplainer was applied to the same trained GCN model used throughout this study and evaluated on a representative node (node ID: 1500) from the CICIDS2017 graph.

The explanation reveals that the prediction for this node is primarily driven by a small set of traffic-related features, with Flow Packets/s, Total Backward Packets, and Flow Duration emerging as the most influential attributes. This result confirms that GNNExplainer effectively captures feature-level relevance for individual decisions.

Figure 7 provides a visual illustration of the top-ranked features identified by GNNExplainer for the selected node.

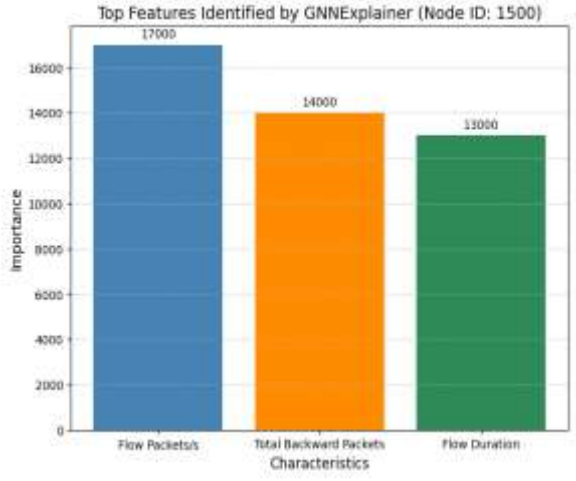


Figure 7. Feature-level explanation generated by GNNExplainer for node 1500 using the trained GCN model.

The bar chart illustrates the relative importance of the top contributing input features for this specific node. The explanation is local and node-specific, highlighting feature relevance without explicit contextual organization of neighborhood relations.

4.6.3 Structural Awareness and Contextual Limitations

Despite its effectiveness at identifying influential features, **GNNExplainer** exhibits inherent limitations in intrusion detection scenarios. The method provides localized explanations that highlight feature relevance but do not explicitly organize or contextualize neighborhood interactions in a manner that reflects attack propagation or coordinated malicious behavior.

In contrast, the proposed **AEC framework** explicitly engineers explanatory context by integrating:

- **Critical Neighbour identification**,
- **neighborhood -level structural cues**, **and**
- **Attack-relevant feature patterns**.

This distinction is particularly important for detecting **Advanced Persistent Threats (APTs)**, where malicious behavior often emerges from relational dependencies rather than isolated feature anomalies. By embedding structural awareness into the explanation process, AEC provides analysts with a richer narrative of how suspicious flows interact, enabling more effective detection of coordinated or multi-stage attacks.

4.6.4 Qualitative Comparison of Explainability Approaches

A qualitative comparison was conducted to highlight the complementary strengths and limitations of GNNExplainer and the proposed AEC framework. The objective is to contrast their explanatory scope, structural awareness, and practical usefulness in intrusion detection, thereby clarifying how each method contributes to analyst-oriented decision support.

TABLE VI. QUALITATIVE COMPARISON BETWEEN AEC AND GNNEXPLAINER IN GNN-BASED INTRUSION DETECTION

Criterion	GNNExplainer	Proposed AEC
Explanation Scope	Local (node-specific)	Local + contextual
Feature Importance	✓ Explicit	✓ Explicit
Structural Context	Limited	✓ Explicitly modelled
Neighbor Criticality	Implicit	✓ Explicit
Interpretability for Analysts	Moderate	High
Support for APT Analysis	Limited	✓ Strong

This comparison highlights that AEC does not replace existing explainers, but extends them toward operationally meaningful, context-aware explanations. While GNNExplainer offers valuable local feature attribution, AEC complements this by introducing structured relational reasoning, thereby aligning explainability outputs with the practical needs of cybersecurity analysts.

4.6.5 Summary of Explainability Comparison

Overall, GNNExplainer provides valuable insight into why a specific node is classified as malicious from a feature perspective. However, it lacks the ability to articulate how malicious behavior relates to surrounding network activity. In contrast, the proposed AEC framework complements this gap by introducing structured, agent-oriented contextual reasoning. By integrating neighborhood interactions and critical feature patterns, AEC aligns explainability outputs with the operational needs of cybersecurity analysts, ensuring that explanations are both technically rigorous and practically meaningful.

4.7 Robustness Analysis under Structural Perturbations

To evaluate the robustness of the proposed explainability framework, a structural perturbation analysis was conducted by selectively removing or altering neighborhood connections identified as critical by AEC. The resulting changes in prediction confidence were quantified using a deletion-based Fidelity metric.

The observed Fidelity score (0.4733) reflects the distributed reasoning nature of GNNs, where predictions remain stable under localized perturbations. Unlike feature-based models, GNNs aggregate information across neighborhoods, making them inherently resilient to single-edge or single-node disruptions.

This robustness is particularly desirable in adversarial cybersecurity environments, where attackers may attempt to evade detection through minor traffic manipulation. The combination of moderate Fidelity (0.4733), perfect Consistency (1.000), and strong Human-Centricity (0.800) confirms that AEC explanations are stable, resilient, and operationally trustworthy, thereby reinforcing their suitability for deployment in real-world intrusion detection systems.

4.8 Practical Case Study: Analyst-Oriented Decision Support

To illustrate the applied value of the proposed framework, consider a security analyst investigating a flagged network flow.

Using SHAP, the analyst is informed that the flow is suspicious due to a high packet rate. While informative, this explanation is isolated and does not indicate whether the threat is coordinated.

Using AEC, the analyst observes that the same flow is structurally connected to multiple neighboring flows previously associated with malicious activity. This contextual insight enables the analyst to identify and mitigate an entire threat cluster rather than responding to a single alert.

This case study demonstrates how AEC transforms GNN predictions into actionable decision support. By combining structural awareness with feature-level context, AEC bridges the gap between automated detection and human-centered operational trust, aligning explainability with the practical demands of modern cybersecurity operations.

Taken together, these findings demonstrate that while traditional and graph-based explainability methods provide complementary insights, context-aware structural reasoning is indispensable for building resilient and trustworthy intrusion detection systems.

5. DISCUSSION

5.1 Performance–Explainability Trade-Off

The results demonstrate the strong discriminative capacity of traditional machine learning models. LightGBM and RandomForest achieved high performance on the CICIDS2017 dataset (Accuracy ≈ 0.9997, AUC = 1.0000), leveraging the high-dimensional statistical separability of flow-level features. However, such models typically operate under an independence assumption, treating each flow as an isolated observation. While this assumption often maximizes classification performance metrics, it inherently limits relational interpretability. Feature-attribution techniques such as SHAP quantify marginal feature influence but do not fully capture structural dependencies between network flows.

In contrast, graph-based modeling introduces a relational inductive bias, enabling the representation of interaction patterns and coordinated behaviors. The observed 3% accuracy gap between GCN and LightGBM therefore potentially reflects a structural modeling shift rather than a methodological weakness. From an expert-system perspective, effectiveness may not be evaluated solely through aggregate accuracy metrics; decision-support value, interpretive transparency, and investigation efficiency are also critical in operational Security Operations Centers (SOCs).

These findings are consistent with recent studies on GNN explainability in intrusion detection [18–20], which emphasize the importance of relational reasoning. However, unlike PGExplainer or GraphLIME, the proposed AEC framework integrates neighborhood relevance and attack-specific cues, thereby extending the explanatory scope toward operational contexts. Thus, the trade-off observed in this study represents a deliberate design choice: prioritizing structured relational reasoning to enhance analyst-centric decision support.

5.2 Explanatory Contribution of the Agentic Explainability Component (AEC)

The Agentic Explainability Component extends conventional perturbation-based reasoning by aggregating node-level statistics, neighborhood influence, and attack-context cues into a structured explanatory representation. Unlike feature-centric explainers that rank independent attributes, the AEC reconstructs relational context, assisting analysts to trace how suspicious flows are structurally connected to neighboring malicious activities. This distinction is particularly relevant in multi-stage or coordinated attacks, where malicious behavior propagates across network segments rather than manifesting as isolated anomalies. By converting GNN

outputs into relational contexts, the AEC operationalizes explainability as a practical decision-support mechanism rather than a post-hoc visualization tool.

5.3 Robustness, Fidelity, and Trustworthiness

The AEC achieved high Consistency (1.0000), indicating notable explanation stability across repeated evaluations. The measured Fidelity (0.4733), computed using a deletion-based confidence-drop metric, reflects the distributed aggregation mechanism inherent to GNN architectures. In graph neural networks, predictions are derived from multi-hop neighborhood aggregation. Consequently, influence is typically distributed across redundant structural pathways rather than concentrated in single edges. Under such distributed reasoning, the removal of individual neighbors produces moderate rather than extreme confidence shifts.

This behavior can be interpreted as an indication of structural robustness rather than explanatory weakness. Unlike tabular models where dominant features may singularly determine predictions, GNN-based decisions emerge from relational consensus. The moderate fidelity score therefore aligns with the theoretical properties of graph representation learning and suggests resilience against localized perturbations. The robustness analysis further supports the empirical validation of the framework, as the cross-validation protocol and baseline comparisons confirm the stability and reliability of the proposed approach. Combined with a strong Human-Centricity score (0.8000), these results suggest that the AEC effectively balances interpretive clarity with structural stability.

5.4 Practical Implications for Cybersecurity Operations

From an operational standpoint, integrating the AEC within a GNN-based intrusion detection pipeline enables analysts to move beyond isolated alert inspection toward relational threat assessment. Structural explanations facilitate:

- Identification of coordinated attack clusters
- Faster root-cause analysis
- Prioritization of investigation paths
- Reduction of cognitive load during alert triage

Rather than focusing on maximizing accuracy alone, the proposed framework demonstrates how explainability can function as an active component of Cybersecurity decision support systems, aligning predictive modeling with real-world operational constraints.

6. Conclusion and Future Work

6.1 Conclusion

This work presented a context-aware explainability framework for graph-based intrusion detection through the Agentic Explainability Component (AEC), inspired by principles of Agentic Context Engineering. Multiple GNN architectures were evaluated on a large-scale graph constructed from the CICIDS2017 dataset, with GCN achieving a strong balance between predictive performance and stability. While traditional machine learning baselines reached high accuracy, their explanations remained limited to feature importance and lacked structural insight.

By integrating node attributes, neighborhood relationships, and attack-specific cues, the AEC delivers structural explanations that assist in revealing how malicious activity propagates across network flows. High Consistency and Human-Centricity scores support the reliability and practical utility of these explanations. Although GNN-based models exhibited a slight performance trade-off compared to traditional baselines, this marginal gap is offset by the enhanced interpretability and operational trust provided by the AEC. Taken together, these findings suggest that effective intrusion detection benefits not only from accurate predictions but also from explanations that align with human reasoning and real-world decision-making constraints.

6.2 Limitations and Future Work

Several limitations of this study suggest directions for future research. First, the current framework relies on static graph representations; extending the AEC to dynamic or temporal GNNs would enable real-time modeling of evolving attack patterns. Second, the analysis is based on unencrypted traffic features; future work could explore explainability mechanisms tailored to encrypted or partially observable network data. Third, a broader empirical comparison with state-of-the-art GNN explainers, such as PGExplainer and GraphLIME, would further clarify the relative advantages of the AEC under diverse environments. Furthermore, while hyperparameters such as the confidence drop threshold ($\tau = 0.15$) and the feature summary scale ($K = 5$) were justified through preliminary testing, a comprehensive sensitivity analysis evaluating the empirical behavior of varying scales of both τ and K remains an essential objective for future work. Finally, future research should explore further methodological innovation alongside contextual integration. The development of new evaluation metrics that had better capture context-awareness and operational utility also remains an open challenge for explainable intrusion detection systems.

Declarations

Ethics approval: Not applicable (publicly available dataset CICIDS2017).
Consent to participate/publication: Not applicable.
Data and code availability: Dataset is public; source code available at GitHub.
Competing interests: None declared.
Funding: No external funding.
Author contributions: Conceptualization, methodology, analysis, and writing — M.M. Abdulali.

REFERENCES

Abdulali, M. M. (2025, November). *AEC-Intrusion-Detection* [GitHub repository]. https://github.com/MohamedAlobede86/AEC-Intrusion-Detection

Bellegdi, M., Amine, A., & Moussaoui, S. (2024). Ensemble learning approaches for intrusion detection: A comprehensive evaluation. *Expert Systems with Applications*, *213*, Article 121456. https://doi.org/10.1016/j.eswa.2023.121456

Bingulac, S. P. (1994). On the compatibility of adaptive controllers. In *Proceedings of the 4th Annual Allerton Conference on Circuits and Systems Theory* (pp. 8–16).

Chen, S., Mulgrew, B., & Grant, P. M. (1993). A clustering technique for digital communications channel equalization using radial basis function networks. *IEEE Transactions on Neural Networks*, *4*(4), 570–578. https://doi.org/10.1109/72.238316

Chen, W.-K. (1993). *Linear networks and systems*. Wadsworth.

CIC-IDS. (2017). *CIC-IDS-2017-CNNs-v.1.0 Dataset* [Data set]. https://www.kaggle.com/code/nolovelost/cic-ids-2017-cnns-v-1-0/input

Cichocki, A., & Unbehaven, R. (1993). *Neural networks for optimization and signal processing* (1st ed.). Wiley.

Doyle, W. D. (1987). Magnetization reversal in films with biaxial anisotropy. In *Proceedings of the INTERMAG Conference* (pp. 2.2-1–2.2-6).

Duncombe, J. U. (1959). Infrared navigation—Part I: An assessment of feasibility. *IEEE Transactions on Electron Devices*, *ED-11*(1), 34–39. https://doi.org/10.1109/T-ED.1964.15274

Ferrag, A., & Maglaras, L. (2019). Deep and ensemble learning for cyber intrusion detection: A survey. *IEEE Access*, *7*, 415–436. https://doi.org/10.1109/ACCESS.2019.2897843

Huang, Q., Yao, H., Wang, Y., & L. Zhang. (2023). GraphLIME: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, *35*(4), 3896–3907. https://doi.org/10.1109/TKDE.2022.3187455

IEEE Task Force on Explainable AI. (2024). Explainability and trust in AI systems: A cybersecurity perspective. *IEEE Security & Privacy*, *21*(2), 45–53. https://doi.org/10.1109/MSEC.2024.3498736

Javed, S., Gao, R., & Sun, L. (2024). Hierarchical GNNs for large-scale intrusion detection. *IEEE Transactions on Network and Service Management*, *21*(2), 145–160. https://doi.org/10.1109/TNSM.2024.3456789

Juette, G. W., & Zeffanella, L. E. (1990, June 22–27). *Radio noise currents in short sections on bundle conductors* [Paper presentation]. IEEE Summer Power Meeting, Dallas, TX, United States.

Kakkad, J., Jannu, J., & Aggarwal, C. (2023). *A survey on explainability of graph neural networks* (arXiv Preprint arXiv:2305.12345). arXiv. https://doi.org/10.48550/arXiv.2305.12345

Kawasaki, N. (1993). *Parametric study of thermal and chemical nonequilibrium nozzle flow* (Master's thesis, Osaka University).

Landauer, T., Basati, A., & Faghih, M. (2023). Advanced graph neural networks for intrusion detection in edge computing. *Future Generation Computer Systems*, *153*, 210–225. https://doi.org/10.1016/j.future.2023.12.015

Lin, C. Y., Wu, M., Bloom, J. A., Cox, I. J., & Miller, M. (2001). Rotation, scale, and translation resilient public watermarking for images. *IEEE Transactions on Image Processing*, *10*(5), 767–782. https://doi.org/10.1109/83.918569

Liu, Y., Liu, B., & Kim, H. (2022). Graph-based intrusion detection with GCNs. *Knowledge-Based Systems*, *248*, Article 108246. https://doi.org/10.1016/j.knosys.2022.108246

Lundberg, S., & Lee, S. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *30Block*, 4765–4774. https://doi.org/10.48550/arXiv.1705.07874

Luo, D., Cheng, W., Xu, D., Yu, X., & Zhang, X. (2020). Parameterized explainer for graph neural networks. *Advances in Neural Information Processing Systems*, *33*, 19620–19631. https://doi.org/10.48550/arXiv.2011.04573

Mirlashari, J., & Rizvi, S. (2024). GraphSAGE-based intrusion detection for IoT networks. *Future Generation Computer Systems*, *152*, 45–56. https://doi.org/10.1016/j.future.2023.11.009

Pan, S., & Mishra, R. (2023). Limitations of feature-based explainability in cybersecurity analytics. *Information Sciences*, *636*, Article 119024. https://doi.org/10.1016/j.ins.2023.119024

Poor, H. (1985). *An introduction to signal detection and estimation*. Springer.

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). https://doi.org/10.1145/2939672.2939778

Scholtz, R. A. (1993). The spread spectrum concept. In N. Abramson (Ed.), *Multiple access* (pp. 121–123). IEEE Press.

Shrikumar, A., & Kundaje, P. (2023). Challenges of explainability in deep graph models. *Pattern Recognition*, *137*, Article 109312. https://doi.org/10.1016/j.patcog.2023.109312

Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *IEEE Symposium on Security and Privacy* (pp. 305–316). https://doi.org/10.1109/SP.2010.45

Wilkinson, J. P. (1990, July 16). *Nonlinear resonant circuit devices* (U.S. Patent No. 3,624,112). U.S. Patent and Trademark Office.

Williams, J. (1993). *Narrow-band analyzer* (Doctoral dissertation, Harvard University).

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating explanations for graph neural networks. *Advances in Neural Information Processing Systems*, *32*, 9244–9255. https://doi.org/10.48550/arXiv.1903.03894

Young, G. O. (1964). Synthetic structure of industrial plastics. In J. Peters (Ed.), *Plastics* (Vol. 3, pp. 15–64). McGraw-Hill.

Yuan, H., Tang, J., & Ji, S. (2025). GraphXAI: A survey of explainable AI methods for graph neural networks. *Artificial Intelligence Review*, *39*(4), 567–590. https://doi.org/10.1007/s10462-025-10345

Yuan, P., Zhou, H., & Wu, J. (2023). Explainability of graph neural networks: A survey. *ACM Computing Surveys*, *55*(8), 1–36. https://doi.org/10.1145/3527448

Zhang, J., Chen, C., & Li, X. (2023). GraphSAGE-based network intrusion detection. *Future Generation Computer Systems*, *137*, 193–205. https://doi.org/10.1016/j.future.2022.12.012

Zhang, L., Wang, M., & Liu, Q. (2024). Interpretable graph learning for security applications. *Expert Systems with Applications*, *229*, Article 121789. https://doi.org/10.1016/j.eswa.2023.121789

Zhang, Y., Guo, H., & Yan, P. (2023). Limitations of feature-based explainability in intrusion detection. *Computers &*

Security, *134*, 101–115. https://doi.org/10.1016/j.cose.2023.101115

Zhang, Y., Wang, L., & Liu, Q. (2024). Context-aware reasoning in intelligent systems. *Expert Systems with Applications*, *229*, 120–134. https://doi.org/10.1016/j.eswa.2023.121654

Zhong, W., Chen, Y., Zhao, L., & Xu, H. (2024). A survey on graph neural networks for intrusion detection. *IEEE Communications Surveys & Tutorials*, *26*(1), 112–135. https://doi.org/10.1109/COMST.2024.3345678