Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.085403

ARTICLE

LLM Enhanced Explainable Intrusion Detection System for Generating Actionable Security Insights

Mohammed Atoum[1], **Malik Al-Essa**[1,*], **Yazeed Alsarhan**[2], **Ahmad K. Al Hwaitat**[1] **and Muhammad Imran**[3]

[1]Computer Science Department, King Abdullah II Faculty for Information Technology, The University of Jordan, Amman, Jordan
[2]Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[3]Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

*Corresponding Author: Malik Al-Essa. Email: m.alessa@ju.edu.jo

Received: 11 May 2026; Accepted: 29 July 2026; Published: 15 September 2026

ABSTRACT: With the urgent need for Intrusion Detection Systems (IDS) to protect digital infrastructure, eXplainable Artificial Intelligence (XAI) has become an important supporting layer. The integration of XAI and IDS can rank influential features that affect IDS decisions, yet these outputs often remain difficult to translate into operational security actions. In this work, we propose LEXIS (LLM-Enhanced eXplainable Intrusion detection System), an LLM-enhanced explainable IDS that converts sample-level explanations into structured report drafts that organize feature attributions into candidate response actions for analyst review, through an evidence-bounded reporting process. Given a network trace, the classifier generates a prediction for that sample, while an XAI method generates a top-k explanation set with attribution scores. Then, an LLM is prompted to produce a machine-readable incident report in a fixed JSON schema, which includes suspected causes, confidence cues, false-positive checks, and recommended response actions extracted from a predefined action catalog. To investigate the effect of adversarial attacks against both the classifier and the XAI method, we evaluate explanation stability under normalized feature-space Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) attacks using a signed-rank stability metric that separately reports feature-rank reordering through Kendall tau correlation and attribution-sign agreement between clean and attacked samples, each with bootstrap confidence intervals and permutation-test significance. The experiments are evaluated on the CICIDS2017 dataset, where the proposed model achieves high detection performance on the majority traffic classes, while minority and subtle attack classes remain more challenging. Focal-loss training with per-class threshold calibration raises macro-F1 from 0.726 to 0.776 on the imbalanced class distribution. The stability analysis shows that feature-space adversarial perturbations only partially disrupt explanations, with feature ranks retaining substantial order and attribution signs agreeing significantly above chance level. The findings suggest that coupling XAI with grounded LLM summarization can help transform raw feature attributions into standardized triage-oriented report drafts. We evaluate the framework as a benchmark-based prototype, and broader validation on diverse datasets, real Security Operations Center (SOC) logs, analyst-in-the-loop assessment, and problem-space adversarial attacks remains future work.

KEYWORDS: XAI; intrusion detection systems; SHAP; adversarial attacks; deep learning; large language models

1 Introduction

Recently, IDS has been considered a critical layer for protecting modern networks; however, the growing volume and diversity of alerts make manual triage expensive and error-prone. Machine learning (ML) and

Deep Learning (DL) techniques are proposed to be used in detecting cyber-attacks due to their powerful capabilities, but their generated decisions are considered as black-box decisions and need to be explained for security analysts. XAI techniques such as SHapley Additive exPlanations (SHAP) [1] provide local-level feature attributions that help answer *why* an alert was triggered. However, a recurring operational gap remains; ranked feature lists are rarely self-explanatory to security practitioners, particularly when features are domain-specific (e.g., protocol counters or timing statistics). As a result, security analysts still need to manually translate explanations generated by an XAI technique into decisions and actions (e.g., verify false positives, isolate hosts, block indicators), which introduces inconsistency and delays to the work. LLMs, as a new technique, that is becoming widely used, offer a promising bridge from explanations to actions, where LLMs can transform technical evidence into readable incident reports and standardized triage steps. However, the problem is that LLMs can hallucinate facts, overstate certainty, or propose unsafe actions, which may generate unsafe reports that can mislead security analysts. Therefore, what is required is that using LLMs in security triage requires restricting generation to the provided evidence, and responses are auditable and comparable across all generated alerts. Unlike prior IDS studies, in this work we (i) demonstrate an evidence-bounded LLM reporting layer that converts feature-level explanations into structured triage-oriented report drafts, (ii) constrain the generated reports using the supplied XAI evidence and a predefined action catalog, and (iii) analyze explanation stability under normalized feature-space FGSM and PGD attacks using a decomposed, statistically tested stability metric.

We emphasize at the outset that the present work is a benchmark-based prototype for structured explanation-to-action reporting, not a validated SOC triage system. All experiments are conducted offline on the refined CICIDS2017 benchmark; the framework is not evaluated with real SOC logs, human-analyst assessment, operational deployment constraints, latency profiling, or field validation. The SOC-oriented terminology used throughout (e.g., triage, response actions, analyst support) therefore describes the intended use case and the controlled benchmark setting, rather than a demonstrated operational deployment.

The contributions of this work can be summarized as follows:

1. We propose LEXIS (LLM-Enhanced eXplainable Intrusion detection System), an LLM-enhanced explainable intrusion detection framework that transforms quantitative XAI evidence into structured report drafts designed to support SOC-style triage in a controlled benchmark setting.
2. We propose a grounded LLM reasoning engine constrained by explanation evidence and a predefined action catalog. The generated report follows a fixed machine-readable schema and is designed to reduce unsupported reasoning by requiring evidence-bounded claims and catalog-based response actions.
3. We introduce a decomposed signed-rank stability analysis that reports the Kendall rank component and the sign-agreement component separately, with bootstrap confidence intervals and permutation-test significance, in order to evaluate explanation drift under normalized feature-space FGSM and PGD attacks.
4. We conduct experiments on the refined CICIDS2017 dataset to analyze IDS performance, global and local SHAP behavior, attribution drift under feature-space adversarial attacks, explanation stability, class-imbalance treatments, and representative grounded LLM-generated incident reports.

This paper is structured as follows: The related work is discussed in Section 2. Section 3 explains the proposed method. Section 4 discusses the results, and finally, Section 5 concludes the paper and outlines future directions.

2 Related Work

2.1 Machine Learning-Based Intrusion Detection Systems

ML and DL-based IDSs have grown substantially over the past few years, driven by the limitations of signature-based systems in addressing novel and polymorphic attacks. ML-based IDS adopted different techniques to detect cyber-threats, such as decision trees, support vector machines, k-nearest-neighbor algorithms [2,3]. However, these techniques had the problem of working with high-dimensional network flow data, where they failed to generalize across heterogeneous traffic environments [4]. On the other side, the use of Deep Neural Networks (DNNs) enabled an end-to-end learning of hierarchical feature representations from raw flow statistics, significantly improving the detection of cyber-threats across multi-class attack taxonomies [5]. Convolutional Neural Networks (CNNs), a technique that was originally proposed to classify images, have been applied to packet-level features encoded as images [6], while recurrent architectures, notably Long Short-Term Memory (LSTM) networks, exploit the sequential nature of network flows to capture temporal attack patterns [7]. Recently, IDS research has investigated lightweight and explainable architectures designed for deployment in resource-constrained environments, e.g., knowledge distillation techniques have been adopted to reduce computational overhead while preserving interpretability, and detection capability in consumer-device intrusion detection systems [8].

2.2 Explainable AI for Intrusion Detection

The adoption of post-hoc XAI methods in the IDS domain has been motivated by the requirement for accountability and trust in high-stakes security decisions generated by DNN models. SHAP (SHapley Additive exPlanations) [1] has emerged as the most widely adopted explainable AI (XAI) method due to its theoretical grounding in cooperative game theory and its model-agnostic applicability. In the IDS context, SHAP is typically applied at both global explanations, to identify the most influential features across the dataset, and local explanations, to explain individual decisions [9]. Complementary approaches include Integrated Gradients [10], which attributes predictions to input features by integrating gradients along a straight-line path from a baseline to the input, and LIME (Local Interpretable Model-agnostic Explanations) [11], which fits locally faithful surrogate models around individual predictions. DALEX [12] as an XAI technique that provides both global and local information, is considered as a framework for model-agnostic explanations that supports breakdown profiles and partial dependence plots, enabling comparative analysis across model families. In the literature, several works proposed applying XAI techniques to explain the decisions generated by IDSs. The work in [13] applied SHAP techniques to generate local explanations for a random forest IDS through identifying the top contributing features per attack class, finding flow-level statistics, e.g., packet duration and inter-arrival timing. Similarly, Ref. [14] demonstrated LIME-based explanations for anomaly detection in industrial control system traffic. However, these studies still have limitations related to the explanations generated by an XAI technique used, which remain in the form of ranked feature attribution vectors, quantitative outputs that are interpretable to ML practitioners but opaque to operational security analysts unfamiliar with the underlying feature engineering, which hinder the explainability of the decisions generated by these classifiers. Furthermore, explanation instability is considered as another limitation that is related to the explanations generated by an XAI technique, e.g., SHAP attributions can exhibit high variance across semantically similar samples [9], and adversarial perturbations can cause both prediction flips and attribution sign reversals, a form of semantic instability that undermines trust in explanation-driven workflows [15].

2.3 Adversarial Robustness in Intrusion Detection

Adversarial ML poses a significant threat to learning-based IDS. Fast Gradient Sign Method (FGSM) [16] and Projected Gradient Descent (PGD) [17] are canonical first-order attack algorithms that have been applied in the network security domain to generate adversarial network flow samples that evade classification. In the IDS domain, adversarial samples must satisfy problem-space constraints, where perturbed feature vectors must correspond to realizable network flows, which limits the feasible perturbation factor compared to the image domain. The work in [18] demonstrated FGSM-based evasion against DNN classifiers on network flow datasets, achieving significant accuracy degradation with minimal perturbation factor. Subsequent work explored transferability of adversarial samples across model architectures [19] and proposed adversarial training as a defense mechanism [16]. However, the impact of adversarial perturbations on model explanations, as opposed to predictions alone, has received comparatively little attention. Our work addresses this gap by evaluating explanation stability under FGSM and PGD attacks using a signed rank stability metric that captures both rank reordering and attribution sign flips.

2.4 Large Language Models for Security Analytics

Recently, LLMs have been explored as an interface layer between automated security tools and human analysts in order to generate explanations for human analysts. Their natural language generation capability makes them attractive for generating plain-language summaries of alerts generated by IDS, translating technical log entries into investigative reports, and generating structured triage recommendations [20], which makes it easier for the security analyst to understand the reason behind the generated alerts by an IDS. Early applications have focused on log parsing and threat-intelligence summarization, in which LLMs demonstrated a strong ability to synthesize information from heterogeneous sources into coherent security advisories. More recently, LLMs have been used to support SOC workflows by automating cyber-threat-intelligence analysis, extracting actionable indicators, and generating structured outputs that reduce repetitive analyst effort in analyzing explanations generated by XAI methods [21]. However, the use of LLM in security contexts may introduce hallucination risk [22]. In this case, the LLM model may fabricate indicators of compromise (IoCs) or recommend actions that are inconsistent with the available evidence generated by an IDS. This motivates the grounded reasoning design adopted in this work, where an LLM is constrained to evidence supplied by an XAI technique and instructed to select actions from a controlled catalog to mitigate hallucination. Nowadays, LLM-assisted cybersecurity systems have been used for automated alert interpretation, threat summarization, SOC support, and structured incident reporting [23]. However, LLM generation in security contexts still introduces risks related to unsupported reasoning and hallucinated outputs. This motivates the requirement for grounded reasoning frameworks that constrain generated outputs using evidence derived from explainable IDS components.

3 Method

The proposed method is illustrated in Fig. 1 (Algorithm 1), which improves an IDS through the use of a grounded LLM layer that converts quantitative model explanations into actionable security insights. The proposed method is organized into three stages as follows:

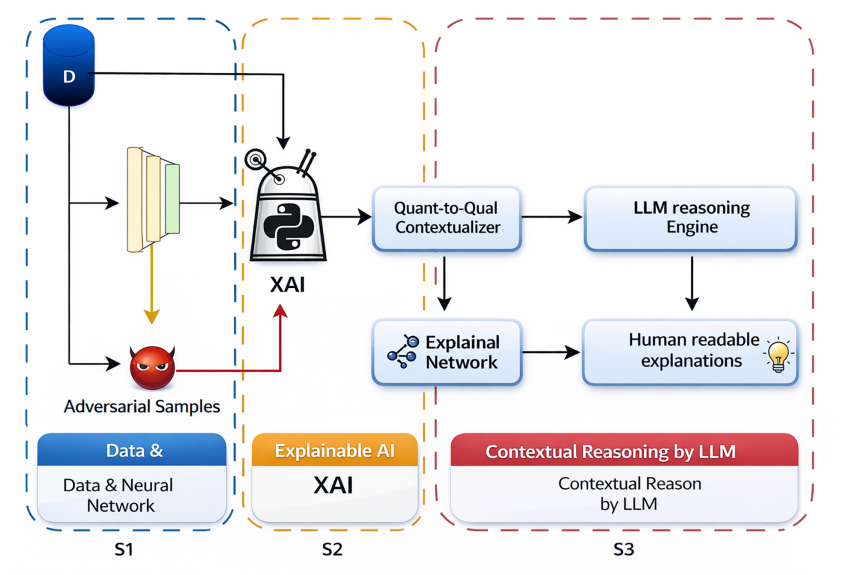


Figure 1: Overview of the proposed schema.

Algorithm 1: LEXIS

Input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$; classifier f_θ; loss $\mathcal{J}(f_\theta(x), y)$; XAI explainer $E(\cdot)$; contextualizer $C(\cdot)$; LLM $\mathcal{L}(\cdot)$;
attack factor: ϵ; PGD steps T; PGD step size α; norm $\|\cdot\|_\infty$.

Output: Explanations for each mode $\mathcal{H}^{clean}, \mathcal{H}^{fgsm}, \mathcal{H}^{pgd}$.

Function FGSM(x, y, ϵ):
  $g \leftarrow \nabla_x \mathcal{J}(f_\theta(x), y)$
  $x^{fgsm} \leftarrow \text{clip}(x + \epsilon \cdot \text{sign}(g))$
  return x^{fgsm}

Function PGD$(x, y, \epsilon, \alpha, T)$:
  $x^0 \leftarrow x + \mathcal{U}(-\epsilon, \epsilon)$
  for $t \leftarrow 1$ **to** T **do**
    $g \leftarrow \nabla_x \mathcal{J}(f_\theta(x^{t-1}), y)$
    $x^t \leftarrow x^{t-1} + \alpha \cdot \text{sign}(g)$
    $x^t \leftarrow \Pi_{\mathcal{B}_\infty(x,\epsilon)}(x^t)$ // Project to ϵ-ball
    $x^t \leftarrow \text{clip}(x^t)$
  return x^T

Function GenerateExplanation(x):
  $\hat{y} \leftarrow f_\theta(x)$
  $e \leftarrow E(f_\theta, x)$ // feature ranking
  $z \leftarrow C(x, \hat{y}, e)$ // quantitative → qualitative context pack
  $h \leftarrow \mathcal{L}(z)$ // LLM reasoning + summary
  return h

for $i \leftarrow 1$ **to** N **do**

(Continued)

Algorithm 1 (continued)

// **Mode 1: Clean**
$x^{clean} \leftarrow x_i$; $y \leftarrow y_i$
$\mathcal{H}_i^{clean} \leftarrow \text{GenerateExplanation}(x^{clean})$
// **Mode 2: FGSM**
$x^{fgsm} \leftarrow \text{FGSM}(x^{clean}, y, \epsilon)$
$\mathcal{H}_i^{fgsm} \leftarrow \text{GenerateExplanation}(x^{fgsm})$
// **Mode 3: PGD**
$x^{pgd} \leftarrow \text{PGD}(x^{clean}, y, \epsilon, \alpha, T)$
$\mathcal{H}_i^{pgd} \leftarrow \text{GenerateExplanation}(x^{pgd})$
return $\mathcal{H}^{clean}, \mathcal{H}^{fgsm}, \mathcal{H}^{pgd}$

S1: Data, IDS Model, and Adversarial Attacks Testing

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ denote a labeled IDS dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector extracted from network traffic and $y_i \in \{1, \ldots, C\}$ is the class label (e.g., benign vs. attack categories). An IDS model f_θ is trained to output class probabilities:

$$\hat{\mathbf{p}} = f_\theta(\mathbf{x}) \in [0,1]^C, \quad \hat{y} = \arg\max_c \hat{p}_c. \tag{1}$$

We distinguish two notions of robustness that are kept separate throughout this work. *Prediction robustness* refers to whether the predicted label $\hat{y}$ changes under an adversarial perturbation, whereas *explanation robustness* refers to whether the feature attributions consumed by the downstream LLM remain stable under the same perturbation. These two properties are independent: a sample whose label is unchanged may still exhibit large attribution drift, and conversely a label flip does not necessarily imply a complete change of the underlying evidence. The stability metric defined in Stage S3 (and analyzed in Section 4.3.4) measures explanation robustness specifically, and is computed under the clean-prediction convention so that it isolates pure explanation drift from the consequence of prediction flips. To evaluate not only predictive robustness but also *explanation robustness*, we generate adversarial samples $\tilde{\mathbf{x}}$ from clean inputs $\mathbf{x}$ using standard gradient-based attacks. For an input $\mathbf{x}$ with label y, define the loss $\mathcal{J}(f_\theta(\mathbf{x}), y)$. The FGSM generates:

$$\tilde{\mathbf{x}} = \mathbf{x} + \epsilon \cdot \text{sign}\left(\nabla_{\mathbf{x}} \mathcal{J}(f_\theta(\mathbf{x}), y)\right), \tag{2}$$

where ϵ controls the perturbation factor. PGD iteratively refines $\tilde{\mathbf{x}}$ within an ℓ_∞-ball:

$$\tilde{\mathbf{x}}^{(t+1)} = \Pi_{\mathcal{B}_\epsilon(\mathbf{x})}\left(\tilde{\mathbf{x}}^{(t)} + \alpha \cdot \text{sign}\left(\nabla_{\tilde{\mathbf{x}}^{(t)}} \mathcal{J}(f_\theta(\tilde{\mathbf{x}}^{(t)}), y)\right)\right), \tag{3}$$

with step size α and projection operator $\Pi_{\mathcal{B}_\epsilon(\mathbf{x})}$. In S1, both $\mathbf{x}$ (clean) and $\tilde{\mathbf{x}}$ (adversarial) are passed through f_θ, producing predictions and confidence scores used downstream.

S2: XAI

Given a sample $\mathbf{x}$ (clean or adversarial), an explainer $E(\cdot)$ produces *quantitative evidence* about the decision of f_θ. Concretely, we compute a ranked attribution vector:

$$\mathbf{a} = E(f_\theta, \mathbf{x}) \in \mathbb{R}^d, \tag{4}$$

where a_j represents the contribution of feature j to the model output (e.g., the predicted class logit or probability). Let $\mathcal{T}(\mathbf{x}) = \{\, j \mid j \in \text{TopK}(|\mathbf{a}|, k) \,\}$ denote the index set of the k features with the largest absolute

attribution for sample $\mathbf{x}$; this set is used both to form the top-k explanation set below and, in the stability analysis of Section 4.3.4, to compare clean and adversarial explanations. Note that $\mathcal{T}(\cdot)$ denotes this index set throughout, and is distinct from the scalar T used in Algorithm 1 for the number of PGD iterations. The explanation is represented as the top-k set as illustrated in the following Equation:

$$\mathcal{E}_k(\mathbf{x}) = \left\{\left(j, x_j, a_j, s_j\right)\right\}_{j\in\mathcal{T}(\mathbf{x})}, \tag{5}$$

where $s_j = \text{sign}(a_j)$ denotes whether the feature increases or decreases the predicted target class score.

S3: LLM Contextual Reasoning for Actionable Security Insights

Three components are used in Stage S3 to convert quantitative explanations into operationally meaningful outputs: (i) a quantitative-to-qualitative contextualizer, (ii) an intermediate explanation network, and (iii) a grounded LLM reasoning engine that generates human-readable and actionable results. First, in the Quant-to-Qual Contextualizer, XAI outputs contain cryptic features (e.g., protocol counters, inter-arrival statistics) and raw numeric values that are hard to be interpreted directly. We therefore define a deterministic mapping $\phi(\cdot)$ from quantitative evidence to qualitative tokens:

$$\mathcal{Q}(\mathbf{x}) = \phi\left(\mathcal{E}_k(\mathbf{x})\right), \tag{6}$$

The output $\mathcal{Q}(\mathbf{x})$ is a structured set of interpretable statements, e.g., *"high connection fan-out"*, *"burst-like packet timing"*, or *"elevated SYN-like activity"*, each explicitly tied to one or more feature tuples from $\mathcal{E}_k(\mathbf{x})$. Explainable Network Construction is used to improve consistency and auditability, through optionally building an intermediate graph representation that captures relationships among evidence items. Let $G = (V, E)$ be an Explainable Network where nodes correspond to (i) features, (ii) derived behaviors from $\mathcal{Q}(\mathbf{x})$, and (iii) hypothesis labels (e.g., scan, brute-force, DoS). Edges are created using simple domain rules (e.g., multiple timing-variance features jointly support a burstiness node) and/or similarity grouping over feature semantics.

Grounded LLM Reasoning Engine phase is where LLM receives only the following inputs: (i) the predicted class $\hat{y}$ and confidence $\hat{\mathbf{p}}$, (ii) the quantitative evidence $\mathcal{E}_k(\mathbf{x})$, (iii) the qualitative context $\mathcal{Q}(\mathbf{x})$, and (iv) the Explainable Network G (if enabled). The LLM is instructed to generate an output strictly following a predefined schema $\mathcal{S}$ (e.g., JSON) that supports SOC triage as follows:

> **Schema $\mathcal{S}$:** `alert_summary`, `predicted_label`, `top_evidence` (list), `threat_hypothesis`, `confidence_notes`, `false_positive_checks` (list), `recommended_actions` (list), and `required_logs` (list).

We enforce two grounding constraints in order to reduce hallucinations and preserve traceability: **(C1) Evidence-bounded claims:** Every claim in `top_evidence` must reference a tuple from the explanation evidence set, including the feature name, value, and attribution sign or magnitude. Through this constraint, the LLM is instructed to avoid using indicators that are not present in the input. **(C2) Controlled action library:** Recommended actions and false-positive checks are selected from the proposed catalog. In the case that critical context is missing, an LLM must return `needs_context` rather than guessing. Formally, the reasoning engine in this step is defined as in the following Equation:

$$\mathcal{R}(\mathbf{x}) = \text{LLM}\left(\hat{y}, \hat{\mathbf{p}}, \mathcal{E}_k(\mathbf{x}), \mathcal{Q}(\mathbf{x}), G; \mathcal{S}, \mathcal{A}\right), \tag{7}$$

where $\mathcal{R}(\mathbf{x})$ is the human-readable report.

In this work, we use an action catalog $\mathcal{A}$ as implemented by a predefined set of SOC-oriented response actions. Each action in this report is associated with a unique identifier and a short operational description (e.g., `inspect_zeek_conn`, `rate_limit_source`), where in the report generation process, an LLM is constrained to selecting actions only from this catalog. The full action catalog $\mathcal{A}$ used in our experiments is listed in Table 1; the LLM may populate `recommended_actions` and the action-bearing `false_positive_checks` only with keys drawn from this catalog, and any out-of-catalog key triggers report rejection via `validate_soc_report` (see Section 4.3.6). This closed action vocabulary is the mechanism through which hallucination of unsafe or non-existent response actions is operationalized and prevented. In the proposed method, Quant-to-Qual contextualizer applies deterministic mapping rules that convert quantitative feature attributions into interpretable behavioral descriptors (e.g., "burst-like packet timing" or "high-volume flow"). To construct a lightweight graph structure, the Explainable Network is used, where nodes represent features, qualitative behavioral indicators, and attack hypotheses, and edges represent predefined semantic or rule-based relationships among those entities. The main aim of this integration is to improve the grounding consistency, interpretability, and reproducibility of the generated SOC reports.

Table 1: The controlled action catalog $\mathcal{A}$. The grounded LLM may only emit `recommended_actions` whose keys appear in this catalog; the `needs_context` key is the mandatory fallback when critical context is missing (constraint C2). Any key not present in $\mathcal{A}$ causes the report to be rejected by `validate_soc_report` and regenerated.

Action Key	Operational Description	Typical Trigger Evidence
`inspect_zeek_conn`	Examine Zeek/Bro connection logs for the source IP	Burst-like or high-volume flow evidence
`rate_limit_source`	Apply temporary rate limiting to the source IP	Volumetric flood indicators
`block_indicator`	Block the offending IP/indicator at the perimeter	High-confidence malicious classification
`isolate_host`	Quarantine the affected internal host	Lateral-movement/infiltration evidence
`verify_false_positive`	Run false-positive checks before escalation	Low stability score or benign-adjacent evidence
`needs_context`	Defer to analyst; do not recommend an action	Critical context missing (C2 fallback)

As $\mathcal{E}_k(\cdot)$ is computed for both clean $\mathbf{x}$ and adversarial $\mathbf{x}'$, we can quantify explanation drift. Recall that $\mathcal{T}(\mathbf{x})$, defined in Stage S2, is the set of top-k feature indices for sample $\mathbf{x}$. A simple sign-agreement stability score, computed over the top-k union rather than over set intersection, is:

$$\sigma_k(x, \tilde{x}) = \frac{\left|\left\{ j \in \mathcal{T}(x) \cup \mathcal{T}(\tilde{x}) : \operatorname{sign}(a_j) = \operatorname{sign}(\tilde{a}_j) \right\}\right|}{|\mathcal{T}(x) \cup \mathcal{T}(\tilde{x})|} \tag{8}$$

where $\sigma_k(x, \tilde{x})$ is the fraction of features in the top-k union $\mathcal{T}(x) \cup \mathcal{T}(\tilde{x})$ whose attribution sign is preserved between the clean sample x and the adversarial sample $\tilde{x}$, and a_j, $\tilde{a}_j$ denote the attribution of feature j for the clean and adversarial samples respectively. Note that σ_k is not a set-overlap measure, it quantifies directional agreement over the combined top-k set, whereas set overlap proper is reported separately as the Jaccard index Jaccard@$k = |\mathcal{T}(\mathbf{x}) \cap \mathcal{T}(\tilde{\mathbf{x}})| \,/\, |\mathcal{T}(\mathbf{x}) \cup \mathcal{T}(\tilde{\mathbf{x}})|$. The two quantities are nevertheless numerically close,

and coincide exactly whenever the features shared by the two top-k sets are also the features whose attribution sign is preserved, which is the regime observed for most classes in our experiments. The composite signed-rank stability score then combines this sign-agreement component with the Kendall rank component:

$$\mathrm{Stab}_k(x, \tilde{x}) = \frac{1}{2}\left(\tau_k(x, \tilde{x}) + \sigma_k(x, \tilde{x})\right) \tag{9}$$

where $\tau_k(x, \tilde{x})$ is the Kendall rank correlation coefficient computed over all of the feature magnitude vectors $|\mathbf{a}|$ and $|\tilde{\mathbf{a}}|$. A score of $\mathrm{Stab}_k = 1.0$ indicates perfect stability in both rank order and attribution direction. Because the two components have different ranges ($\tau_k \in [-1, 1]$, $\sigma_k \in [0, 1]$), the composite score alone is ambiguous: a value of 0.5 can arise from randomized ranks with fully preserved signs ($\tau_k \approx 0$, $\sigma_k \approx 1$) as well as from preserved ranks with fully reversed signs ($\tau_k \approx 1$, $\sigma_k \approx 0$), whereas near-random behavior in *both* components yields $\mathrm{Stab}_k \approx 0.25$. We therefore report τ_k and σ_k separately, alongside the composite score, in all stability tables.

This provides a direct measure of whether adversarial perturbations destabilize the explanation evidence that the LLM consumes, which is crucial for trustworthy deployment. To assess robustness, we evaluate explanation stability under adversarial attacks (FGSM and PGD attacks) using a signed rank stability metric Stab_k that jointly captures feature rank reordering (Kendall-τ) and attribution sign agreement between $\mathcal{E}_k(x)$ and $\mathcal{E}_k(x')$ for attacked samples x'. The proposed signed-rank stability metric incorporates two complementary properties of XAI robustness. The first one is the Kendall-τ component, which evaluates whether the relative ordering of features remains stable under adversarial attacks. On the other side, the sign-agreement component evaluates whether the explanatory direction of these features remains semantically consistent. This distinction is considered an important factor in the proposed LLM-assisted SOC setting, where feature-rank reordering may alter evidence prioritization presented to the security analyst. In contrast, attribution-sign reversals may lead to contradictory or misleading security interpretations. Therefore, jointly measuring rank consistency and sign agreement provides a more operationally meaningful assessment of explanation stability under adversarial attacks.

4 Empirical Evaluation and Discussion

4.1 Datasets

CICIDS2017 [24] is a network intrusion dataset from the Canadian Institute for Cybersecurity collected on a dedicated testbed with different hosts and operating systems. The environment separates victim and attacker segments and generates traffic that traverses the public Internet. The captured traces include common application protocols (e.g., HTTP/HTTPS, FTP, SSH, SMTP) and combine benign activity, driven by profiling agents trained on traces of real user behavior, with coordinated attack campaigns. Attacks include brute force attacks, botnet communication, Heartbleed, several variants of DoS and DDoS, infiltration, and web-related threats. Each trace in CICIDS2017 has 78 numeric features and 1 class feature extracted with the CICFlowMeter tool [24]. In our experiments, we adopt the refined CICIDS2017 release introduced by [25][1], which addresses known issues in the original dataset released by the Canadian Institute for Cybersecurity by removing artifacts, correcting errors, and repairing mislabeled samples from the original dataset. This refinement retains 72 numeric features (removing spurious attributes that can promote overfitting) and contains no categorical fields. We use two disjoint subsets of 100,000 flows each for training and testing via stratified sampling without replacement. Both splits preserve the base class distribution, containing 80% benign and 20% attack traffic. The attack traffic is organized into eight categories: *Port Scan*, *DoS Hulk*, *DDoS*,

[1] downloads.distrinet-research.be/WTMC2021

DoS GoldenEye, *DoS Slowloris*, *FTP-Patator*, *SSH-Patator*, and *DoS Slowhttptest*. CICIDS2017 dataset suffers from the problem of minority classes, where it has six out of eight classes that are under-represented in the dataset.

4.2 Implementation Details

The implementation of the proposed pipeline was developed in Python 3.9, using the Keras 2.7 API to build the neural models, with TensorFlow as the computational backend. All numerical features were pre-processed using *min–max scaling*, which linearly rescales each feature into the range $[0,1]$. This normalization mitigates scale-induced dominance among heterogeneous attributes and typically improves convergence and stability for gradient-based optimization. To optimize hyperparameters used in the DNN architecture, we employed the *Tree-structured Parzen Estimator (TPE)* algorithm, as implemented in the Hyperopt library, and a validation dataset comprising 20% of the training set (selected via stratified sampling) has been used. We set the number of optimization trials to 30, where the configuration that achieves the minimum validation loss is selected. The DNN architecture used in this work consists of three fully connected layers, with layer parameters determined through hyperparameter optimization. We used dropout and batch normalization to improve generalization and mitigate overfitting, where we applied the *ReLU* activation function in the hidden layers [26]. The output layer uses *softmax* to generate a probability distribution over classes. DNN model training was performed for up to 150 epochs, with *early stopping* (patience = 10) to restore the checkpoint with the lowest validation loss.

For interpretability, we used explanation techniques at both local and global levels. SHAP (v 0.48.0) was used to generate local feature attributions. For the LLM reasoning stage (S3), we used the `gemini-2.5-flash` model via the Google Generative AI API, configured with a temperature of 0.2 and a maximum output length of 3000 tokens to encourage low-variance and concise structured outputs. All adversarial perturbations were generated in the normalized feature space after applying min–max scaling to the refined CICIDS2017 features. The perturbation values were clipped to remain within the valid normalized range ($[0,1]$) for every feature. FGSM adversarial samples were generated using a single-step gradient update with perturbation factor ($\epsilon = 0.01$), while PGD adversarial samples were generated iteratively using projected updates constrained within an (ℓ_∞) perturbation ball, with step size ($\alpha = \epsilon/10$) and ($T = 10$) iterations. Since the refined CICIDS2017 dataset adopted in this work contains only numerical attributes, no categorical-feature perturbation constraints were required. We additionally emphasize that the generated adversarial samples are evaluated as feature-space perturbations and are not guaranteed to correspond to fully realizable packet-level network flows. The Adversarial Robustness Toolbox (ART) library was used to generate both FGSM and PGD adversarial samples.

The pipeline incurs three sequential per-sample costs. DNN inference on the trained three-layer model is negligible (sub-millisecond per flow on commodity hardware). The SHAP attribution step is the dominant local cost, since the explainer scales with the size of the background set and the feature count, and it is therefore the bottleneck of the explanation stage rather than the classifier. The grounded LLM reporting stage issues a single `gemini-2.5-flash` API call per alert (temperature 0.2, ≤3000 output tokens), whose end-to-end latency is network-bound and typically on the order of a few seconds per report. Importantly, in a deployment the SHAP attribution and LLM reporting stages run *after* the real-time classification decision, as a post-alert enrichment step; they therefore do not lie on the detection critical path and do not affect detection latency. The values reported here are indicative timings on our experimental setup rather than optimized production benchmarks, and a systematic latency and throughput study under operational load is left to future work.

4.3 Results and Discussion

4.3.1 Performance Analysis

Table 2 presents the per-class detection performance of the proposed model on the clean CICIDS2017 test set, revealing a generally strong ability to distinguish between benign traffic and multiple attack categories. Overall, the DNN model shows high discriminative capability across all classes, as reflected by the AUC-ROC values, which range from 0.95 to 1.00. This illustrates that the DNN model is highly effective at separating class distributions even when class-wise precision and recall vary. Specifically, the majority of traffic categories, including *Benign*, *DoS Hulk*, *PortScan*, and *DDoS*, are identified with strong performance, suggesting that the learned feature representation captures the most salient characteristics of large and well-represented classes. The CICIDS2017 dataset is considered a highly imbalanced dataset, where the normal traffic is about 79% of the dataset. The DNN model achieves perfect precision (1.00) and high recall (0.92), resulting in an F1-score of 0.96, predicting normal samples in the testing dataset. This implies that samples predicted as benign are almost always correct, although a subset of benign flows is still mis-classified as malicious due to the nature of this dataset (as a highly imbalanced dataset). From an intrusion detection perspective, this trade-off is often acceptable, since prioritizing attack detection is generally more critical than maximizing benign recall. Among the attack classes, *DoS Hulk* and *DDoS* achieve the strongest results, with recall values of 1.00, precision of 0.96 and 0.97, and F1-scores of 0.98 and 0.99, respectively. These findings show that high-volume denial-of-service patterns are highly separable from normal traffic in the dataset. Similarly, *FTP-Patator* achieves an F1-score of 0.85 with perfect recall, showing that the DNN model can also effectively capture brute-force attack behavior when class signatures are sufficiently distinctive in the dataset.

Table 2: Per-class detection performance on the clean test set.

Class	Precision	Recall	F1-Score	Support	AUC-ROC
Benign	1.00	0.92	0.96	79,288	0.99
DoS Hulk	0.96	1.00	0.98	7582	1.00
PortScan	0.64	1.00	0.78	7609	0.99
DDoS	0.97	1.00	0.99	4551	1.00
DoS GoldenEye	0.32	0.64	0.43	362	0.95
FTP-Patator	0.73	1.00	0.85	191	1.00
SSH-Patator	0.55	0.99	0.70	143	1.00
DoS slowloris	0.22	0.98	0.36	191	0.99
DoS Slowhttptest	0.33	0.93	0.49	83	0.99

However, the results showed decreased class-wise reliability for minority classes, where the minority classes are less than 1% of the samples in this CICIDS2017 dataset. Although the class *PortScan* gets perfect recall (1.00), still its precision decreases to 0.64, which yield an F1-score of 0.78, which means that the DNN model is highly sensitive to scan-related activity, so, it tends to over-predict this class, likely due to scanning behavior sharing common characteristics with other anomalous traffic patterns. A similar trend is observed for *SSH-Patator*, which reaches a recall of 0.99 but only 0.55 precision, which means effective detection of true attacks but with a non-negligible false positive rate. Furthermore, challenging classes such as *DoS GoldenEye*, *DoS slowloris*, and *DoS Slowhttptest*, whose F1-scores drop to 0.43, 0.36, and 0.49, respectively, are characterized by relatively low support in the training dataset, and markedly lower precision despite moderate-to-high recall. This suggests that the DNN model can often identify them but struggles to distinguish them cleanly from other attack categories. Finally, these results shown in Table 2 confirm

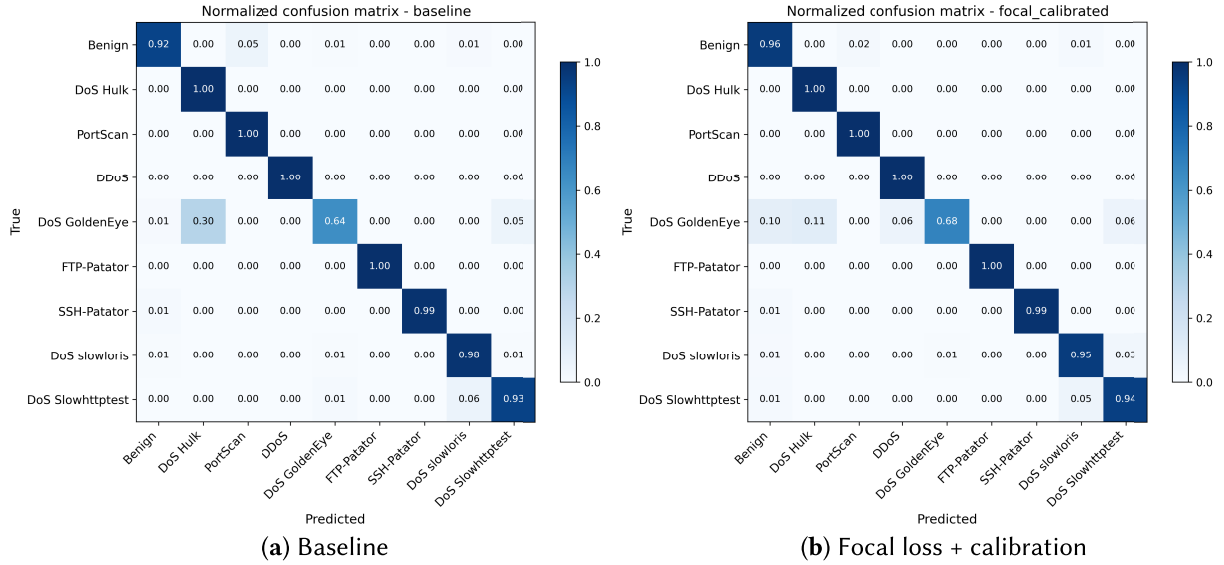
that the proposed DNN model performs very well on the majority classes in the CICIDS2017 dataset, while its performance decreases on minority and subtle attack categories, which highlights the importance of improved imbalance handling, stronger class-specific feature discrimination to improve the detection performance for these attack classes by the DNN model.

To address the reduced reliability on minority classes, we evaluated two imbalance-treatment strategies on the same architecture and training protocol: inverse-frequency class weighting and focal loss ($\gamma = 2$), each assessed both with the standard argmax decision rule and with per-class decision thresholds calibrated to maximize one-vs-rest F1 on the validation split. Table 3 reports the per-class F1-scores of all configurations, and Fig. 2 shows the corresponding confusion matrices, in which every cell is annotated with the raw flow count and the row-normalized fraction so that precision, recall, and F1 are directly derivable from the figure. Focal loss combined with threshold calibration yields the strongest overall results, improving macro-F1 from 0.726 to 0.776 and overall accuracy from 0.937 to 0.963, with clear per-class F1 gains for seven of the nine classes: *DoS GoldenEye* improves from 0.428 to 0.541, *DoS Slowhttptest* from 0.492 to 0.598, *PortScan* from 0.779 to 0.894, *SSH-Patator* from 0.705 to 0.801, and *FTP-Patator* from 0.845 to 0.936, while *Benign* and *DoS Hulk* also improve slightly and *DDoS* is essentially unchanged (0.987 to 0.988). The *PortScan* improvement is driven almost entirely by precision: the volume of benign flows misclassified as *PortScan* falls from approximately 5% to 2% of benign traffic. In contrast, inverse-frequency class weighting degrades minority-class F1 substantially (e.g., *DoS slowloris* 0.115, *DoS Slowhttptest* 0.062), because reweighting amplifies the leakage of the dominant *Benign* class into low-support attack columns; we report this negative result to show that the choice of imbalance treatment, rather than imbalance treatment per se, determines the outcome.

Two trade-offs of the focal-loss model warrant explicit discussion. First, *DoS slowloris* is the single class that regresses (F1 0.364 to 0.261): its recall remains high (0.95), but the number of benign flows misclassified as *slowloris* grows from approximately 655 to 1019, a change that is invisible in a row-normalized confusion matrix, since both correspond to roughly 1% of the benign row. Second, the error structure of *DoS GoldenEye* changes qualitatively: under the baseline, 30% of *GoldenEye* flows were misattributed to *DoS Hulk*, a semantically adjacent HTTP-flood class that still raises an alert—whereas under the focal model this confusion drops to 11%, but the fraction of *GoldenEye* flows misclassified as *Benign* rises from 1% to 10%, converting label confusion into silent misses. From an operational standpoint these are different failure modes, and miss-averse deployments may prefer the class-weighted calibrated variant, whose *GoldenEye*-to-*Benign* leakage remains at 2% in our experiments.

Table 3: Per-class F1-scores of the baseline model and the imbalance-treated variants (CW = inverse-frequency class weighting; FL = focal loss, $\gamma = 2$; +cal = per-class threshold calibration). Macro-F1 is the unweighted mean over the nine classes. The best results in bold.

Class	Baseline	CW	CW + cal	FL	FL + cal
Benign	0.960	0.893	0.936	0.965	**0.977**
DoS Hulk	0.977	0.975	0.971	0.980	**0.985**
PortScan	0.779	0.717	0.791	0.861	**0.894**
DDoS	0.987	0.959	0.984	**0.988**	**0.988**
DoS GoldenEye	0.428	0.178	0.253	0.426	**0.541**
FTP-Patator	0.845	0.628	0.674	0.709	**0.936**
SSH-Patator	0.705	0.233	0.329	0.461	**0.801**
DoS slowloris	**0.364**	0.115	0.198	0.220	0.261
DoS Slowhttptest	0.492	0.062	0.097	0.433	**0.598**
Macro-F1	0.726	0.529	0.581	0.671	**0.776**

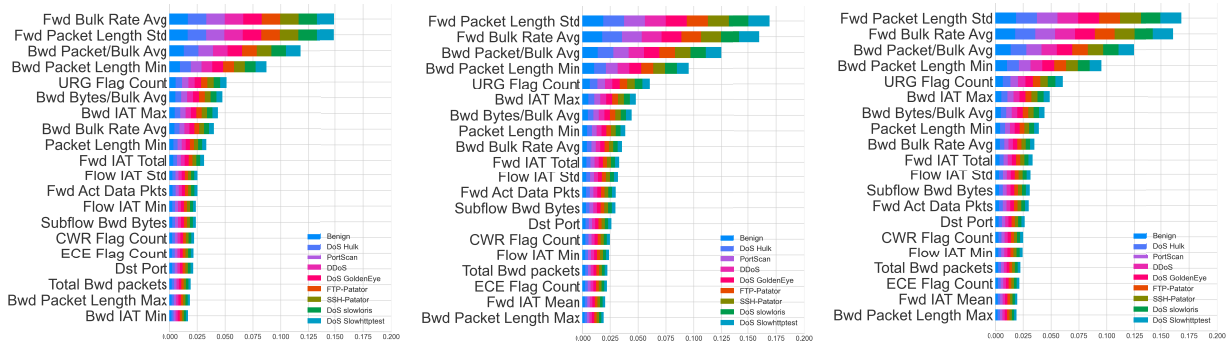


(**a**) Baseline (**b**) Focal loss + calibration

Figure 2: Confusion matrices of the baseline and the focal-loss calibrated model. Each cell shows the raw flow count and the row-normalized fraction.

4.3.2 Global SHAP Explanation Analysis

Fig. 3 shows the global SHAP explanations for the clean test set (Fig. 3a), FGSM adversarial test set (Fig. 3b), and PGD adversarial test set (Fig. 3c). What is clear in Fig. 3 is that under clean conditions, *Fwd Bulk Rate Avg* emerges as the most globally influential feature that affects the prediction of the DNN model, which exhibits a mean |SHAP| value of approximately 0.017 across all nine classes in the CICIDS2017 dataset. While *Fwd Packet Length Std* and *Bwd Packet/Bulk Avg*[2] rank second and third globally, which reflect the model's reliance on packet-size variability and backward bulk throughput as primary discriminators in the DNN model.



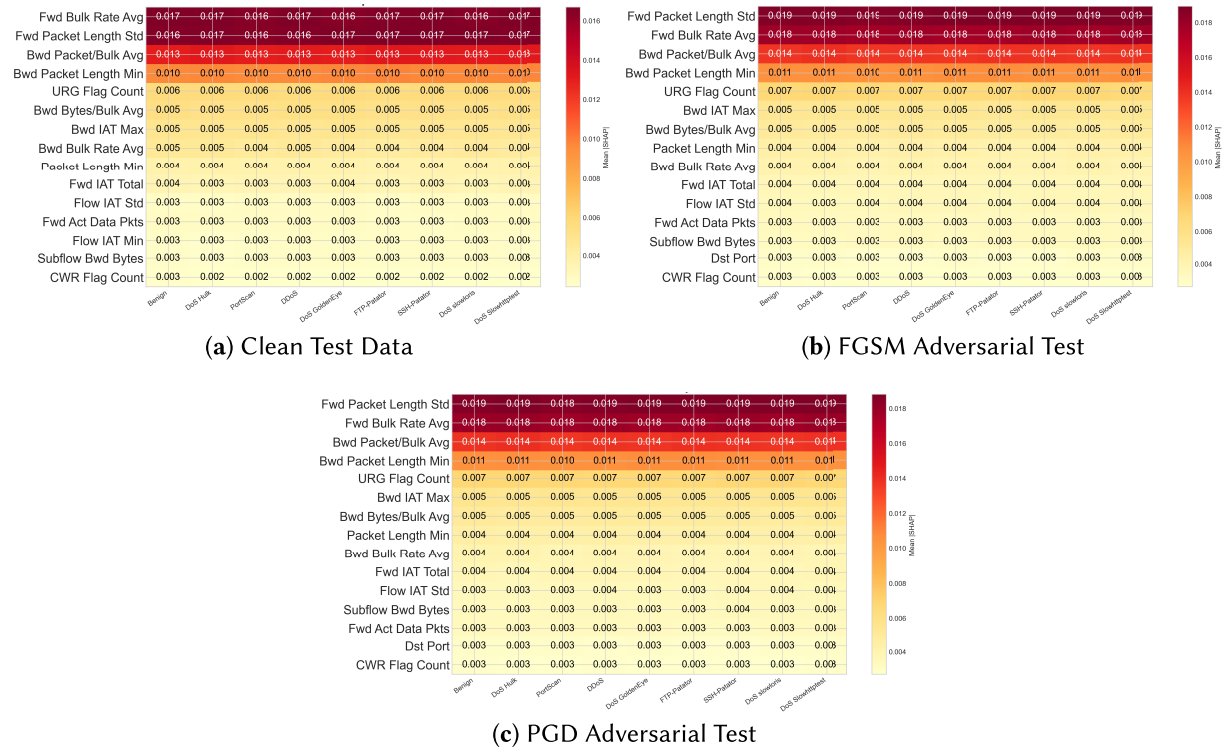
(**a**) Clean Test Data (**b**) FGSM adversarial test (**c**) PGD adversarial test

Figure 3: Global explanations generated by SHAP for the CICIDS2017 dataset.

The heatmap (Fig. 4) reveals that the top-15 global features exhibit highly uniform importance across attack classes under the clean, FGSM, and PGD conditions shown in Fig. 4a–c, respectively: which means

[2] *Bwd Packet/Bulk Avg* and *Bwd Bulk Rate Avg* are two distinct CICFlowMeter bulk-transfer statistics: the former denotes the average number of packets per bulk transfer in the backward direction (CICFlowMeter field `Bwd Avg Packets/Bulk`), while the latter denotes the average bulk transfer rate in the backward direction (`Bwd Avg Bulk Rate`). Both attributes are present in the refined CICIDS2017 release used in this work.

that |SHAP| values for the five most important features vary by less than 0.001 across all nine classes in the dataset. This homogeneity proposes that the DNN model has learned a shared set of general traffic anomaly indicators rather than class-specific distinguishing signatures. However, while this may be considered advantageous for generalization, it also implies that explanations provide limited class-discriminative information, a finding with direct implications for LLM-grounded SOC triage.



(**a**) Clean Test Data

(**b**) FGSM Adversarial Test

(**c**) PGD Adversarial Test

Figure 4: Global Heatmap explanations generated by SHAP for the CICIDS2017 dataset.

Under adversarial attacks by FGSM, the global ranking of features undergoes a notable reorganization; *Fwd Packet Length Std* displaces *Fwd Bulk Rate Avg* as the top-ranked feature across all nine classes (Fig. 3), consistent with the near-uniform cross-class importance profile noted above, while *Dst Port*, which does not appear among the top-15 global features under clean conditions (Fig. 4a), enters the top-15 for all classes under FGSM (Fig. 4b). This rank disruption is consistent with the adversarial objective: the FGSM perturbation shifts feature values in the direction of the gradient, selectively amplifying the apparent importance of features that are locally sensitive to the perturbation direction. The global heatmap under FGSM (Fig. 4b) confirms that absolute magnitudes increase slightly compared to clean conditions: the mean |SHAP| of the top-ranked feature rises from approximately 0.017 (*Fwd Bulk Rate Avg*, the top feature under clean conditions) to approximately 0.019 (*Fwd Packet Length Std*, the top feature under FGSM). Since the identity of the top-ranked feature changes between the two conditions, this comparison reflects both a reordering of the ranking and a mild inflation of attribution magnitudes, rather than a growth in a single feature's importance; the relative ordering within classes is likewise disrupted. On the other side, the PGD heatmap (Fig. 4c) mirrors the FGSM pattern closely. Under PGD, the leading positions of the global ranking are unchanged relative to FGSM, with *Bwd Packet Length Min* remaining in fourth place for the majority of classes. The PGD heatmap values are nearly identical to those of FGSM at three decimal places, suggesting that the additional iterative refinement does not substantially alter the global-level attribution profile beyond what FGSM achieves with a single gradient step.

4.3.3 Per-Class Feature Importance Comparison

Fig. 5 presents per-class grouped horizontal bar charts, comparing mean |SHAP| for the top-8 features across the three conditions. Note that while the visualization displays the top-8 features for clarity, all stability computations and evidence sets use $k = 10$ top features. A consistent pattern is observed across all attack classes: the magnitude of feature attributions is higher under adversarial conditions than under clean conditions, even when the predicted class is the same. This attribution inflation, reflecting increased feature sensitivity rather than increased feature relevance, is a manifestation of adversarial perturbations inflating gradient signals.

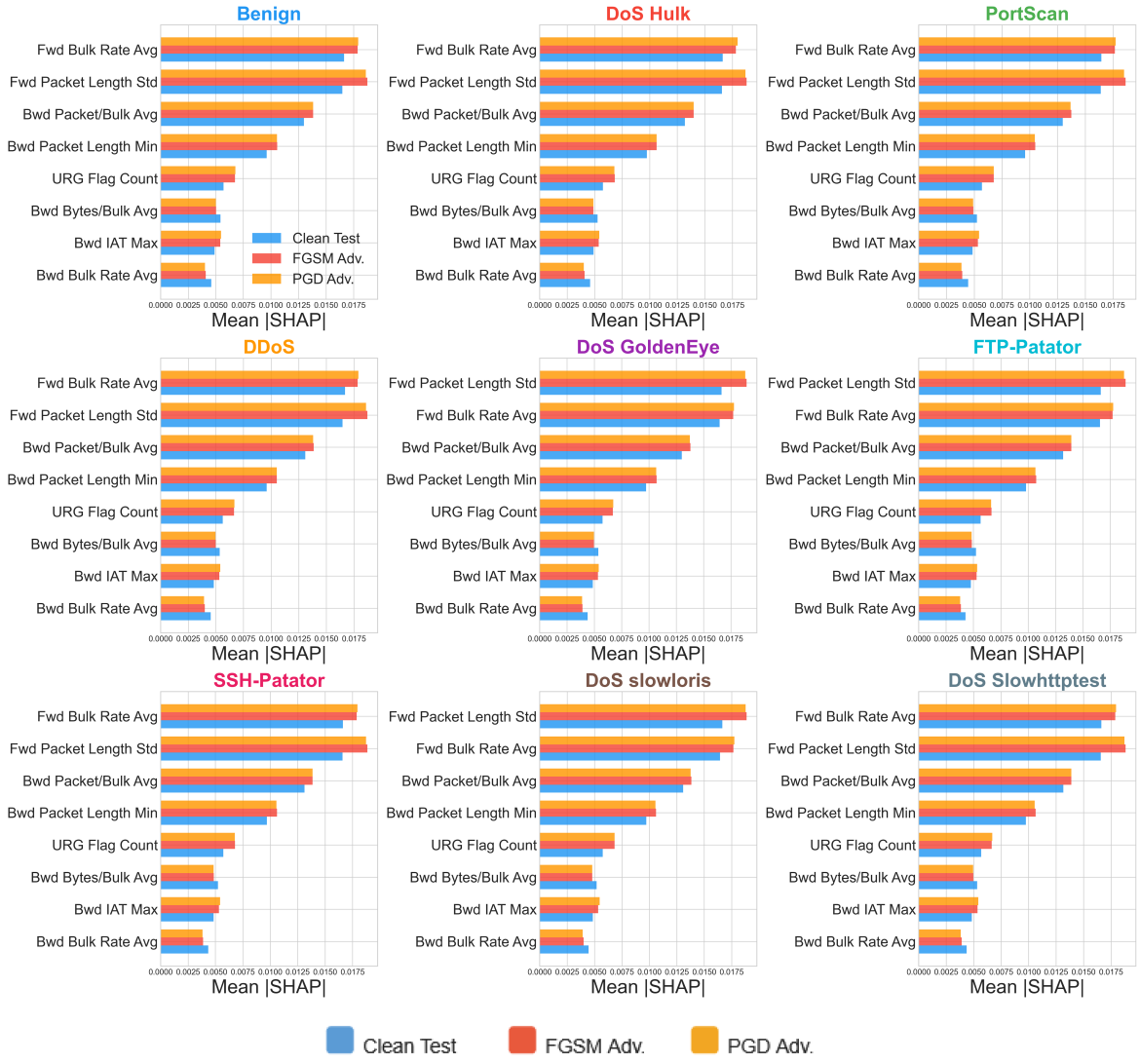
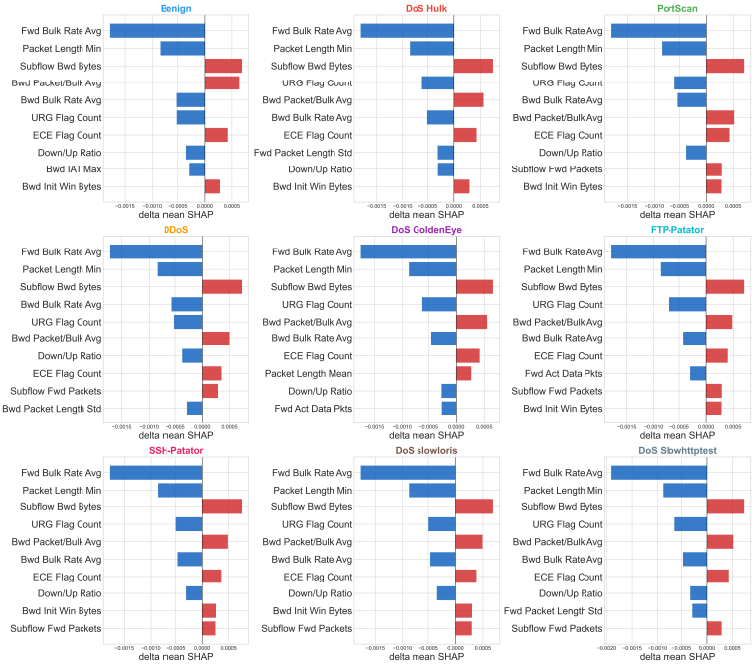


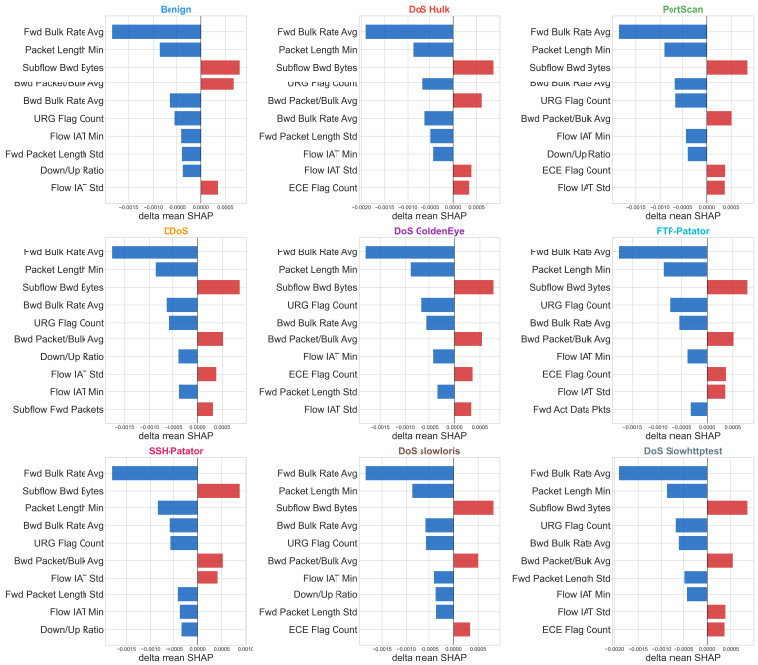
Figure 5: Top feature importance per class: Clean, FGSM, and PGD for CICIDS2017 dataset.

The Δ-SHAP analysis (Fig. 6) quantifies the signed change in mean SHAP attribution between clean and adversarial conditions for each class, comparing clean vs. FGSM in Fig. 6a and clean vs. PGD in Fig. 6b. Red bars (positive Δ) indicate features whose apparent importance was amplified by the attack; blue bars (negative Δ) indicate suppression. For benign traffic under FGSM, *Fwd Bulk Rate Avg* exhibits the largest positive Δ, with an increase of approximately +0.003 in mean SHAP magnitude, while *Flow IAT Std*

is suppressed by approximately −0.001. This directional pattern is consistent across most attack classes, suggesting that FGSM systematically redirects attribution weight from timing-related features toward bulk-rate features. PGD produces a broadly similar pattern, with *DoS slowloris* exhibiting the largest attribution drift; this is consistent with the stability analysis in Section 4.3.4, where *DoS slowloris* shows the largest FGSM-to-PGD decrease in composite stability among all nine classes (Stab$_k$ 0.629 to 0.579, Table 4).



(**a**) FGSM adversarial test



(**b**) PGD adversarial test

Figure 6: SHAP attribution shift (Δ): Clean vs. FGSM, and Clean vs. PGD for CICIDS2017 dataset.

Table 4: Explanation stability per class: Kendall rank component (τ_k), sign-agreement component (σ_k), combined score (Stab_k), and top-k Jaccard overlap ($k = 10$); mean with 95% bootstrap CI.

Class	Attack	τ_k	σ_k	Stab_k	Jaccard@10
PortScan	FGSM	0.727 [0.725, 0.729]	0.596 [0.593, 0.598]	0.661 [0.659, 0.664]	0.596 [0.593, 0.598]
Benign	FGSM	0.590 [0.589, 0.591]	0.470 [0.469, 0.471]	0.530 [0.529, 0.531]	0.476 [0.474, 0.477]
DDoS	FGSM	0.631 [0.626, 0.636]	0.512 [0.508, 0.517]	0.572 [0.567, 0.576]	0.514 [0.510, 0.519]
DoS Hulk	FGSM	0.774 [0.772, 0.776]	0.657 [0.655, 0.660]	0.716 [0.714, 0.718]	0.657 [0.655, 0.660]
FTP-Patator	FGSM	0.705 [0.691, 0.718]	0.600 [0.583, 0.616]	0.652 [0.637, 0.667]	0.600 [0.583, 0.616]
DoS slowloris	FGSM	0.681 [0.654, 0.706]	0.576 [0.550, 0.602]	0.629 [0.602, 0.654]	0.578 [0.553, 0.604]
DoS GoldenEye	FGSM	0.682 [0.660, 0.702]	0.585 [0.564, 0.605]	0.633 [0.612, 0.653]	0.587 [0.567, 0.607]
SSH-Patator	FGSM	0.648 [0.631, 0.665]	0.521 [0.502, 0.541]	0.585 [0.566, 0.603]	0.523 [0.504, 0.542]
DoS Slowhttptest	FGSM	0.626 [0.596, 0.654]	0.531 [0.500, 0.563]	0.579 [0.548, 0.607]	0.533 [0.502, 0.563]
PortScan	PGD	0.735 [0.733, 0.737]	0.607 [0.605, 0.609]	0.671 [0.669, 0.673]	0.607 [0.604, 0.610]
Benign	PGD	0.581 [0.579, 0.582]	0.463 [0.462, 0.464]	0.522 [0.520, 0.523]	0.469 [0.468, 0.470]
DDoS	PGD	0.618 [0.612, 0.623]	0.501 [0.496, 0.506]	0.559 [0.554, 0.564]	0.503 [0.498, 0.508]
DoS Hulk	PGD	0.772 [0.771, 0.774]	0.655 [0.653, 0.658]	0.714 [0.712, 0.716]	0.655 [0.653, 0.658]
FTP-Patator	PGD	0.663 [0.649, 0.677]	0.567 [0.552, 0.583]	0.615 [0.600, 0.630]	0.567 [0.551, 0.583]
DoS slowloris	PGD	0.626 [0.591, 0.661]	0.532 [0.499, 0.564]	0.579 [0.546, 0.613]	0.534 [0.502, 0.566]
DoS GoldenEye	PGD	0.669 [0.647, 0.692]	0.574 [0.553, 0.596]	0.622 [0.600, 0.643]	0.578 [0.556, 0.599]
SSH-Patator	PGD	0.624 [0.607, 0.640]	0.496 [0.478, 0.514]	0.560 [0.543, 0.577]	0.498 [0.481, 0.514]
DoS Slowhttptest	PGD	0.615 [0.590, 0.639]	0.516 [0.490, 0.542]	0.565 [0.539, 0.590]	0.517 [0.492, 0.543]

4.3.4 Explanation Stability Analysis

We recall the distinction established in Stage S1: this subsection analyzes *explanation robustness* (the stability of the attributions consumed by the LLM), which is conceptually separate from the *prediction robustness* (label flips) discussed in the local analysis of Section 4.3.5. To isolate explanation robustness, all stability quantities below are computed under the clean-prediction convention, so that a label flip does not by itself register as explanation drift. Table 4 reports the per-class explanation-stability results (with $k = 10$) for FGSM and PGD adversarial attacks, decomposed into the Kendall rank component τ_k and the sign-agreement component σ_k, each as the mean over test samples with a 95% bootstrap confidence interval ($B = 10{,}000$ resamples). Stability is computed under the clean-prediction convention: for every sample, the attribution vector of the class predicted on the clean input is compared between the clean and attacked inputs, so that the metric measures pure explanation drift rather than the consequence of prediction flips. Statistical significance is assessed with a permutation test against a feature-shuffle null that destroys the clean–adversarial feature correspondence while preserving the marginal attribution distribution (200 permutations per class).

The decomposition reveals that adversarial perturbations only partially disrupt the explanations: the composite Stab_k ranges from 0.52 to 0.72 across classes, with the Kendall component τ_k between 0.58 and 0.77 and the sign-agreement component σ_k between 0.46 and 0.66. Both components lie far above their chance levels under the feature-shuffle null ($\tau_k \approx 0$ and $\sigma_k \approx 0.05$–0.08, respectively; all $p < 0.005$), so the feature ranking retains substantial order and the attribution directions agree well above chance even under attack. The stability is also clearly heterogeneous across classes, *Benign* is the least stable ($\text{Stab}_k \approx 0.53$) while *DoS Hulk* and *PortScan* are the most stable (≈0.72 and ≈0.66), which provides class-specific guidance on how much trust an analyst should place in adversarially exposed evidence. As anticipated in Section 3, σ_k and Jaccard@10 take very similar values in Table 4, and coincide for several classes: in these cases the features

shared by the clean and adversarial top-k sets are exactly the features whose attribution sign is preserved, so directional agreement and set overlap collapse onto the same value.

A paired Wilcoxon signed-rank test over the per-sample Stab_k values shows that PGD is statistically significantly more disruptive than FGSM for eight of the nine classes; the exception is *PortScan*, whose mean Stab_k is marginally higher under PGD than under FGSM (0.671 vs. 0.661). The practical magnitude of this difference is nevertheless small: the per-sample effect sizes are limited (paired Cohen's $d \leq 0.35$), and the class-level mean Stab_k differences between the two attacks reported in Table 4 are at most 0.05. The per-sample test and the class-level means thus support the same conclusion drawn from the global analysis in Section 4.3.2: the iterative refinement of PGD adds only a modest amount of explanation drift over the single-step FGSM.

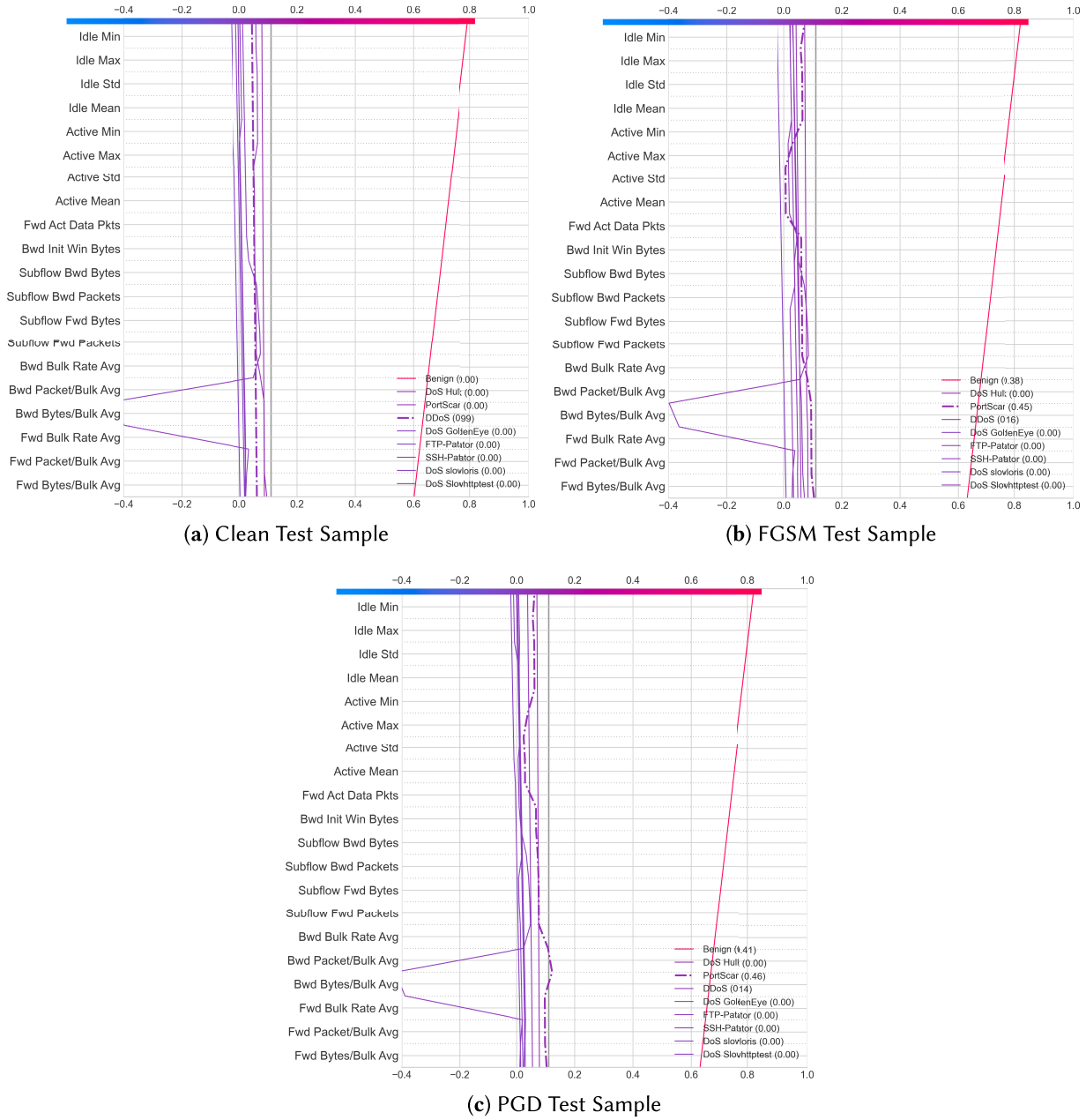
The distinction between the two components has important practical implications. Rank reordering, captured by τ_k, affects the priority of evidence items presented to an LLM or analyst, potentially causing the most operationally relevant indicators to be deprioritized. Sign reversals, captured by σ_k, are more severe: a feature that positively supports a DoS classification under clean conditions may appear to argue against that classification under adversarial conditions, causing the LLM reasoning engine to generate contradictory or misleading security reports. The measured regime, substantially preserved ranks with partial sign agreement well above chance, indicates that adversarial perturbations in this setting act primarily as a graded re-weighting of the evidence rather than a wholesale randomization of it: the evidence set consumed by the LLM remains informative under attack, but individual sign flips within the top-k union do occur and justify propagating the per-class stability scores into the generated reports as an explicit uncertainty signal.

4.3.5 Local Explanation Analysis

Fig. 7 presents the local SHAP decision plots for a representative sample under clean, FGSM, and PGD conditions. Under clean conditions (Fig. 7a), the model predicts *DDoS* with high confidence (probability 0.99), and the local explanation attributes the prediction primarily to a cluster of forward inter-arrival timing features: *Fwd IAT Total* and *Fwd IAT Max* contribute the largest positive attributions, consistent with the high-frequency, burst-like connection patterns characteristic of DDoS attacks.

Under FGSM perturbation (Fig. 7b), the predicted class shifts to *PortScan* (probability 0.45 for PortScan, 0.40 for DDoS), representing a misclassification event. This is an example of *prediction robustness* failing (the label flips); we examine below how the *explanation* reacts to the same perturbation. The local explanation is dramatically reorganised: the timing features that dominated the clean explanation are replaced by packet-level features including *Fwd Act Data Pkts* and *Bwd Bulk Rate Avg* as the primary positive attributors. Critically, attribution signs flip for several features: *Bwd Packet/Bulk Avg* transitions from a positive attributor (supporting DDoS classification) to a negative attributor (arguing against PortScan), illustrating semantic instability at the local level.

Under PGD (Fig. 7c), the prediction shifts to *PortScan* with higher confidence (probability 0.46) relative to FGSM, consistent with PGD's stronger optimization. The local SHAP bars (Fig. 7c) show that the adversarial samples induce a further reorganization of attributions relative to FGSM, with *Bwd Bulk Rate Avg* now the dominant positive attributor and the inter-arrival timing features appearing with reduced magnitude. The reorganization of features across clean, FGSM, and PGD, provides a case study of how adversarial attacks can incrementally change the evidence landscape that is available to an LLM reasoning engine.



(**a**) Clean Test Sample (**b**) FGSM Test Sample

(**c**) PGD Test Sample

Figure 7: Local explanations generated by SHAP for a sample selected from the CICIDS testing set. (**a**) Represents the top-20 features affecting the prediction of the model for the original sample, (**b**) represents the top-20 features affecting the prediction of the model for the FGSM sample, and (**c**) represents the top-20 features affecting the prediction of the model for the PGD sample.

4.3.6 Example LLM Report

An example of the structured incident report generated by LLM reasoning engine (As reported in Section 3, Stage S3) for the DDoS-classified clean sample analyzed in Section 4.3.5 is illustrated in Table 5. The proposed report is grounded in $\mathcal{E}_k(\mathbf{x})$ and constrains all recommended actions to the predefined catalog $\mathcal{A}$ as described previously in Section 3. The report strictly follows the schema $\mathcal{S}$ and exposes eight top-level fields as reported previously, where the `alert_summary` field provides a single-sentence triage headline

incorporating the predicted class by the DNN model and the dominant evidence signal. This enables rapid analyst prioritization without requiring inspection of the full JSON object by the security analyst. Moreover, the `predicted_label` field contains the full class name as returned by the DNN model, which is "`DDoS`" in this sample.

Each entry in `top_evidence` carries six subfields: the raw `feature` name, its min-max normalized `value`, the SHAP attribution `shap`, its `sign` (+1 or −1 indicating whether the feature increases or decreases the predicted class score), a `qualitative_token` produced by the deterministic mapping $\phi(\cdot)$, and a one-sentence `interpretation`; the abridged listing in Table 5 omits the `value` and `interpretation` subfields for space. This structure directly implements constraint C1 in order to force the LLM to allow only to reference the features present in $\mathcal{E}_k(\mathbf{x})$ and every interpretation sentence is anchored to the associated feature tuple. In this example, `Fwd IAT Total` (SHAP = 0.048, sign = +1) is mapped to the qualitative token "`burst-like inter-arrival timing`", which the interpretation links explicitly to the high-frequency flood pattern that characterizes volumetric DDoS traffic. Similarly, `Fwd Bulk Rate Avg` (SHAP = 0.040, sign = +1) is contextualized as "`high-volume flow`", which is consistent with the global SHAP analysis in Section 4.3.2 where this feature is ranked first under clean conditions.

The `threat_hypothesis` field asks the LLM to synthesize the evidence into a single coherent attack report rather than simply repeating feature attributions to make it easier for the security analyst to understand the report. For the DDoS sample that is provided in Table 5, the engine identifies a high-frequency burst pattern and infers a coordinated source flood, a hypothesis consistent with the local SHAP decision plot in Fig. 7, in which forward inter-arrival timing features collectively dominate the positive-attribution side. The `confidence_notes` field serves a dual purpose: first, it records the model's softmax confidence (0.99 for the clean sample) and, second, the adversarial stability context derived from the Stab_k scores defined in Stage S3. So, for this sample, the field records the sample-level Stab_k values computed for this specific flow under both attacks, which the analyst reads alongside the per-class stability context (τ_k, σ_k, and Stab_k; Table 4), flagging that, while the feature ranking is substantially preserved under perturbation, individual attribution signs within the top-k union may flip, even though the clean-condition prediction is highly confident.

The field of `false_positive_checks` provides the actionable verification steps that a security analyst should perform before escalating the alert (one such step is shown in the abridged listing of Table 5). These are generated by the LLM from the qualitative context $\mathcal{Q}(\mathbf{x})$ and the Explainable Network G, then they are phrased as concrete, operationally testable questions. Such checks mitigate false-positive escalation rates by directing security analyst attention to the most plausible benign explanations for the observed traffic pattern. The `recommended_actions` field implements constraint reported in C2: it is a plain list of string keys drawn exclusively from the predefined $\mathcal{A}$ catalog (Table 1). In this example the two actions are `inspect_zeek_conn` ("Examine Zeek/Bro connection logs for this source IP") and `rate_limit_source` ("Apply temporary rate limiting to the source IP"). These keys are validated programmatically by `validate_soc_report` against $\mathcal{A}$ before the report is accepted; so, any hallucinated action key not present in the catalog causes the report to be rejected and re-requested with an explicit violation message. The `required_logs` field complements the action list by specifying the log sources an analyst must retrieve to execute the recommended actions (`zeek_conn.log`), making the report self-contained for SOC analysts.

Table 5: Example structured incident report produced by the grounded LLM stage for a DDoS-classified sample (clean sample). The listing is abridged: only the first `top_evidence` entry is shown, and its `value` and `interpretation` subfields are omitted for space.

```
{
  "alert_summary":
    "High-confidence DDoS alert driven by burst-like
     timing behavior.",

  "predicted_label":
    "DDoS",

  "top_evidence": [
   {
     "feature": "Fwd IAT Total",
      "shap": 0.04821,
      "sign": 1,
      "qualitative_token":
        "burst-like inter-arrival timing"
    },
    { "...": "..." }
  ],

  "threat_hypothesis":
    "High-frequency burst pattern consistent with
     volumetric DDoS activity.",

  "confidence_notes":
    "Model confidence 0.99.
     Stab_k(FGSM) = 0.505,
     Stab_k(PGD) = 0.503.",

  "false_positive_checks": [
    "Verify source IP is not a legitimate
     load balancer."
  ],

  "recommended_actions": [
    "inspect_zeek_conn",
    "rate_limit_source"
  ],

  "required_logs": [
      "zeek_conn.log"
  ]
}
```

As described in Section 4.3.5, when the same sample is passed through the pipeline under both FGSM and PGD attacks, the predicted class shifts to PortScan. In those cases, the LLM report correctly updates `predicted_label` and `top_evidence` to reflect the new prediction and its corresponding $\mathcal{E}_k(\mathbf{x})$ explanations, while the `confidence_notes` field flags the per-class stability scores and the partial

sign agreement between the clean and adversarial explanations. This illustrates the pipeline's transparency property: rather than silently generating a confident but potentially misleading report based on adversarially distorted evidence, the framework surfaces the explanation drift explicitly in the structured output, allowing the receiving analyst to appropriately downweight the triage confidence.

Although the present work demonstrates the structure of the grounded LLM report through the representative case study, a large-scale quantitative evaluation of LLM-generated reports is not included in the current version. We therefore stress that hallucination mitigation in this work is enforced *structurally*, through the evidence-bounding constraint C1 and the catalog-restriction constraint C2, both checked programmatically by `validate_soc_report`, but that its effectiveness has not yet been measured quantitatively; consequently we do not report or claim a measured hallucination rate, and the qualitative grounding analysis presented here should not be read as a validation of the LLM reporting stage. Future work will conduct a systematic report-generation evaluation using hallucination rate, evidence faithfulness, schema compliance, action-catalog validity, predicted-label consistency, regeneration frequency, and human analyst usefulness ratings. This evaluation will also compare grounded LLM reports against deterministic non-LLM template reports in order to quantify the additional value of LLM-based contextual reasoning.

5 Conclusion

This work proposed an LLM-enhanced explainable IDS framework that mitigates the operational gap between raw model explanations and actionable security guidance to enhance the capability of a security analyst taking actions. The proposed method integrates a DNN classifier, a post-hoc XAI method (SHAP), and a grounded LLM reasoning engine that converts sample-level feature attributions into structured, machine-readable incident reports conforming to a fixed JSON schema. Through constraining the LLM to evidence supplied by the XAI stage and restricting recommended actions to a predefined catalog, the proposed method provides an auditable report structure designed to support SOC-style triage in a controlled benchmark setting. Experiments on the refined CICIDS2017 dataset show strong performance on majority traffic classes, while minority and subtle attack classes remain more challenging. Although the present work demonstrates the structure of grounded LLM-generated reports, large-scale quantitative evaluation of hallucination rate, evidence faithfulness, grounding quality, and human analyst usefulness remains future work. To address the highly imbalanced class distribution, we evaluated class weighting and focal loss with per-class threshold calibration: focal-loss training with calibration raises macro-F1 from 0.726 to 0.776 and improves per-class F1 for seven of the nine classes, including most of the minority attack classes, while one class (*DoS slowloris*) regresses due to a precision effect that we report transparently and that remains an open case for future imbalance treatments. A central finding of this work concerns the robustness under adversarial attacks. Decomposing the signed-rank stability metric Stab_k into its Kendall rank component (τ_k) and sign-agreement component (σ_k), with bootstrap confidence intervals and permutation-test significance, we found that both FGSM and PGD attacks only partially disrupt SHAP explanations: feature ranks retain substantial order (τ_k between 0.58 and 0.77) and attribution signs agree well above chance level (σ_k between 0.46 and 0.66; all $p < 0.005$ against a feature-shuffle null), with composite stability between 0.52 and 0.72 and clear heterogeneity across the nine traffic classes. This result has implications for LLM-grounded security reasoning: the evidence set consumed by the LLM remains informative under attack, but individual attribution-sign flips within the top-k union do occur, so the per-class stability scores are propagated into the generated reports as an explicit uncertainty signal rather than treated as a binary trust decision.

Several directions remain open for future work: First, adversarially robust XAI methods, such as smoothed or certified attribution techniques, should be investigated as a replacement for standard SHAP in high-threat environments. Second, the proposed method should be evaluated on more recent and diverse

datasets. A further limitation is that the present work demonstrates the structure of grounded LLM-generated reports through a representative case study, but does not include large-scale quantitative evaluation of hallucination rate, evidence faithfulness, grounding quality, or human analyst usefulness under real SOC conditions.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Mohammed Atoum: conceptualization, methodology, software, formal analysis, investigation, writing—original draft, writing—review & editing. Malik AL-Essa: conceptualization, methodology, software, formal analysis, investigation, writing—original draft, writing—review & editing, project administration. Yazeed Alsarhan: conceptualization, supervision, writing—review & editing, resources. Ahmad K. Al Hwaitat: conceptualization, supervision, writing—review & editing, resources. Muhammad Imran: supervision, writing—review & editing. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The dataset analyzed during the current study is publicly available. The code supporting the findings of this study will be made available upon acceptance of the manuscript.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst. 2017;30:4768–77.
2. Al-Essa M, Alsharo M, Alnsour Y, Ali WA, Almomani O. Transforming data representation: a comparative analysis of tabular and image-based approaches with XAI. J Comput Cogn Eng. 2026;5(2):333–40.
3. Al-Essa M, Ali WA, Alsharo M, Atoum MS, Imran M. COBRA: counterfactual oversampling framework for imbalanced structured data classification. Expert Syst Appl. 2025;304:130764.
4. Lu H, Ma Z, Li X, Bi S, He X, Wang K. TrafficHD: efficient hyperdimensional computing for real-time network traffic analytics. In: Proceedings of the 61st ACM/IEEE Design Automation Conference; 2024 Jun 23–27; San Francisco, CA, USA. p. 1–6.
5. AL-Essa M, Andresini G, Appice A, Malerba D. Striving for simplicity in deep neural models trained for malware detection. In: Meo R, Silvestri F, editors. Machine learning and principles and practice of knowledge discovery in databases. Cham, Switzerland: Springer Nature; 2025. p. 529–40.
6. Kim H, Yoon Y. An ensemble of text convolutional neural networks and multi-head attention layers for classifying threats in network packets. Electronics. 2023;12(20):4253. doi:10.3390/electronics12204253.
7. Muhuri PS, Chatterjee P, Yuan X, Roy K, Esterline A. Using a long short-term memory recurrent neural network (LSTM-RNN) to classify network attacks. Information. 2020;11(5):243. doi:10.3390/info11050243.
8. Moslemi A, Briskina A, Dang Z, Li J. A survey on knowledge distillation: recent advancements. Mach Learn Appl. 2024;18(6):100605. doi:10.1016/j.mlwa.2024.100605.
9. AL-Essa M, Andresini G, Appice A, Malerba D. An XAI-based adversarial training approach for cyber-threat detection. In: Proceedings of the 2022 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech); 2022 Sep 12–15; Falerna, Italy. p. 1–8.
10. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning; 2017 Aug 6–11; Sydney, Australia. p. 3319–28.

11. Ribeiro MT, Singh S, Guestrin C. "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; 2016 Aug 13–17; San Francisco, CA, USA. p. 1135–44.
12. Biecek P. DALEX: explainers for complex predictive models in R. J Mach Learn Res. 2018;19(84):1–5.
13. Hossain MA, Ishtiaq W, Islam MS. A comparative analysis of ensemble-based machine learning approaches with explainable AI for multi-class intrusion detection in drone networks. Secur Priv. 2026;9(1):e70164. doi:10.1002/spy2.70164.
14. Oswal S. From detection to diagnosis: TCAE–DBSCAN with LIME for interpretable ICS anomaly analysis. Int J Appl Math. 2025;38(1s):828–39.
15. Al-Essa M, Qatawneh M, Al-Shamayleh AS, Abualghanam O, Almobaideen W. From hardening to understanding: adversarial training vs. CF-Aug for explainable cyber-threat detection system. Comput Mater Contin. 2026;87(3):76608. doi:10.32604/cmc.2026.076608.
16. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings; 2015 May 7–9; San Diego, CA, USA. p. 1–11.
17. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: Proceedings of the 6th International Conference on Learning Representations, ICLR 2018; 2018 Apr 30–May 3; Vancouver, BC, Canada.
18. Rai MHA, Noor Y, Faisal M, Nawazish MF. Adversarial robustness of deep learning-based intrusion detection systems against AI-powered cyber attacks. Spectr Eng Sci. 2025;3(11):899–922.
19. Wu L, Zhu Z, Tai C. Understanding and enhancing the transferability of adversarial examples. arXiv:1802.09707. 2018.
20. Hassanin M, Moustafa N. A comprehensive overview of large language models (LLMs) for cyber defences: opportunities and directions. arXiv:2405.14487. 2024.
21. Tseng P, Yeh Z, Dai X, Liu P. Using LLMs to automate threat intelligence analysis workflows in security operation centers. arXiv:2407.13093. 2024.
22. Haque MA, Siddique S, Rahman MM, Hasan AR, Das LR, Kamal M, et al. SOK: exploring hallucinations and security risks in AI-assisted software development with insights for LLM deployment. In: Proceedings of the 2025 Sixth International Conference on Intelligent Data Science Technologies and Applications (IDSTA); 2025 Sep 1–4; Varna, Bulgaria. p. 57–64.
23. Alqahtani H, Kumar G. Large language models for cybersecurity intelligence: a systematic review of emerging threats, defensive capabilities, and security evaluation frameworks. Comput Mater Contin. 2026;87(3):77367.
24. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Int Conf Inf Syst Secur Priv. 2018;1(2018):108–16. doi:10.5220/0006639801080116.
25. Engelen G, Rimmer V, Joosen W. Troubleshooting an intrusion detection dataset: the CICIDS2017 case study. In: Proceedings of the 6th IEEE European Symposium on Security and Privacy Workshops, EuroS&PW 2021; 2021 Sep 6–10; Vienna, Austria. p. 7–12.
26. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics; 2011 Apr 11–13; Fort Lauderdale, FL, USA. p. 315–23.