Smart Technologies Academic Press

Jordanian Journal of Informatics and Computing
https://www.jjic.thestap.com/



An Explainable AI Framework for Real-Time Cyberattack Detection and Risk Assessment

Vugar Abdullayev[1*], Alex Khang [2], Nazila Ragimova, [1]

[1]*Department of Computer Engineering, Azerbaijan State Oil and Industry University, Baku, Azerbaijan, Azerbaijan, https://orcid.org/0000-0002-3348-2267*

[2] *Global Research Institute of Technology and Engineering, Raleigh, United States, https://orcid.org/0000-0001-8379-4659*

[3] *Department of Computer Engineering, Azerbaijan State Oil and Industry University, Baku, Azerbaijan, Azerbaijan, https://orcid.org/0000-0002-0942-7882*

ARTICLE INFO

Article History

Received: **27-05-2026**
Revised: **25-06-2026**
Accepted: **03-11-2026**

Vol.2027, No.1

DOI:
https://doi.org/10.63180/jjic.thestap.2027.1.5

***Corresponding author.**

Email

abdulvugar@mail.ru

Orcid:
https://orcid.org/0000-0002-3348-2267

This is an open access article under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

Published by STAP Publisher.

ABSTRACT

As cyberattacks become more sophisticated and more common, detection systems must be not only accurate but also quick and easy to interpret. Although deep learning models have proven to be very effective in detecting intrusion and malware, their black-box nature has hindered their adoption in security operations centers (SOCs), where analysts need to explain and take action on alerts in short timeframes. To offer real-time cyberattack detection with quantitative risk assessment, this paper presents an Explainable AI (XAI) framework that integrates a lightweight hybrid deep learning detector with post-hoc and intrinsic explainability methods. The framework combines a Convolutional-Bidirectional Gated Recurrent Unit (CNN-BiGRU) architecture for spatio-temporal traffic representation with an attention layer that highlights salient time steps and integrates SHAP (Shapley Additive explanations) and LIME (Local Interpretable Model-agnostic Explanations) to generate human-readable justifications for every prediction. A dynamic risk-scoring engine then translates detection confidence, feature-level severity, and asset criticality to a normalized risk index that enables prioritized incident response. The framework is tested on three benchmark intrusion detection datasets (CICIDS2017, NSL-KDD, and UNSW-NB15) and a real-time traffic stream generated synthetically. Experimental results reveal that the proposed model achieves a detection accuracy of 98.6%, achieves an F1-score of 0.981, and the average latency of inference is 14.2 milliseconds per flow, and explanation fidelity, based on the two metrics of deletion/insertion and rank-correlation stability, demonstrates that the explanations generated are faithful to the model's behavior. It has been shown that the proposed hybrid framework is suitable for application in real-time network security environments with limited resources, as it outperforms the standalone CNN, LSTM, Random Forest, and CNN-LSTM baselines in terms of detection performance and retains interpretability with only a minor computational cost. It also demonstrates the applicability of the framework for security operations workflows by means of a comprehensive ablation study, ROC and confusion-matrix analysis, and a comparison with the dynamic risk index and static CVSS-style scoring.

Keywords: Explainable AI; Intrusion Detection; Cybersecurity; Deep Learning; SHAP; LIME; Risk Assessment; Real-Time Detection; CNN-BiGRU; Attention Mechanism; Network Security.

How to cite the article
Abdullayev, V., Khang, A., & Ragimova, N. (2026). An Explainable AI Framework for Real-Time Cyberattack Detection and Risk Assessment. Jordanian Journal of Informatics and Computing, 2027(1), 96. https://doi.org/10.63180/jjic.thestap.2027.1.5

1. Introduction

The explosion of the interconnectedness of digital infrastructure, cloud resources and Internet of Things (IoT) devices has significantly broadened the attack surface to which malicious actors have access. From port scanning to advanced persistent threats (APTs) and zero-day exploits, modern enterprises are constantly under threat from a multitude of different attacks. Traditional signature-based intrusion detection systems (IDS) cannot keep up with the complexity and frequency of the attacks of the present day, and thus, anomaly/misuse detection systems based on machine learning and deep learning are gaining popularity to learn the complex traffic pattern directly from the data [1, 4].

While deep learning-based intrusion detection systems are often said to be a black box, they still have high predictive capabilities. In a Security Operations Center (SOC), security analysts must confirm alerts, grasp the origin of an alerted event and prioritize responses with high time pressure. A model that gives a binary or multi-class prediction without any accompanying rationale doesn't have any practical value since analysts have not a way of sorting out real threats from false positives, they will have no way to communicate with non-technical stakeholders, or meet any regulatory or audit requirement like data-protection or critical-infrastructure legislation. This interpretability gap has become a major challenge for the successful operation and adoption of AI-infused cybersecurity solutions [6] [10] [16].

To address this challenge, Explainable AI (XAI) has emerged as a promising strategy to generate human-explainable explanations for model predictions without significantly compromising detection accuracy. Methods like SHAP and LIME approximate the contribution of each feature for individual predictions, whereas attention mechanisms are intrinsic to the neural architecture and capture the regions of the input space that are most important for the prediction [2], [5], [17]. Most research on XAI augmented IDSs, however, is only concerned with explaining offline ID classification results and seldom consider the extra constraints of real-time operation, nor convert explanations to actionable risk scores to trigger automatic or analyst actions [12],[25].

This paper tackles these limitations by proposing a unified end-to-end Explainable AI system combining (i) a hybrid CNN-BiGRU detector with an embedded attention mechanism for real-time cyberattack classification, (ii) a dual explanation layer based on SHAP and LIME for both global and local interpretability, and (iii) a risk assessment engine that integrates detection confidence, explanation-derived feature severity, and contextual asset criticality into a single, continuously updated risk index. The proposed system can be deployed with existing SOC tooling (SIEM & SOAR) and will be able to perform its operations on streaming network traffic with sub-20 millisecond latency per-flow.

1.1 Motivation and Problem Statement

Most current intrusion detection research focuses on accuracy measures, and interpretability and quantification of risk are secondary considerations and are only considered in isolation of real-time considerations. There are thus three gaps in the literature: (1) no coherent architecture that simultaneously achieves high detection accuracy and inference speed for streaming environments; (2) few architectures that integrate explanations that are faithful to the model's behavior, yet also have acceptable runtime complexity; and (3) no mechanism is available to convert model outputs and explanations into meaningful, prioritized risk scores comparable in spirit to the Common Vulnerability Scoring System (CVSS)[18] and more adaptive than its static counterparts. The motivation for this work stems from the need to fill these gaps within a single, unified framework that a SOC analyst can rely on, interrogate and act upon within the time envelope of a live incident.

A further motivating observation is that the cost of post-hoc explanation methods such as SHAP is not fixed: naively computing Shapley-value attributions for every incoming flow would make real-time deployment infeasible on commodity SOC hardware. This motivates a tiered design in which lightweight, always-on intrinsic explanations coexist with selectively triggered, higher-fidelity post-hoc explanations, an architectural choice that, to the best of our knowledge, has not been systematically evaluated in prior real-time IDS-XAI literature.

1.2 Contributions

1. We propose a hybrid CNN-BiGRU architecture with an attention layer that captures both spatial correlations among traffic features and temporal dependencies across packet/flow sequences, tailored for low-latency real-time inference.
2. We design a dual explainability layer that combines SHAP-based global feature attribution with LIME-based local, instance-level explanation, cross-validated through an attention-consistency check to improve explanation reliability and flag low-agreement cases for manual review.
3. We introduce a composite risk-scoring formulation that fuses model confidence, explanation-weighted feature severity, and asset criticality into a normalized, interpretable risk index suitable for SOC prioritization workflows, and we validate this index against independent analyst judgments.

4. We conduct extensive experiments on three benchmark datasets (CICIDS2017, NSL-KDD, UNSW-NB15) as well as a simulated real-time traffic stream, benchmarking detection accuracy, latency, explanation fidelity, and risk-tier agreement against several baseline and state-of-the-art models, including a component-wise ablation study.
5. We provide a complexity and deployment analysis, including per-flow computational cost, throughput under load, and integration considerations for SIEM/SOAR pipelines, to support reproducibility and practical adoption.

1.3 Novelty and Scope

The novelty of this work lies not in any single component in isolation, since CNN-based, LSTM-based, and attention-augmented intrusion detectors, as well as SHAP- and LIME-based explanation pipelines, have each been explored previously, but in the systematic co-design of detection, explanation, and risk quantification as a single latency-aware pipeline. In particular, the tiered explanation strategy and the explanation-weighted risk index represent, to our knowledge, a combination not jointly evaluated under real-time constraints in prior work. The scope of this study is limited to network-flow-level intrusion detection on benchmark and synthetically streamed datasets; extension to packet-payload inspection, encrypted-traffic analysis.

2. Related Work

2.1 Machine Learning and Deep Learning for Intrusion Detection

Most of the initial work on network intrusion detection was based on the assumption that network flow statistics could be handcrafted to provide a set of features for distinguishing network traffic into normal and malicious classes, and that these features could then be used to train traditional machine learning classifiers, like Decision Trees, Support Vector Machines, and Random Forests [13, 19]. These methods are computationally efficient but don't necessarily work for new attack types and they are dependent on significant manual feature engineering. With the introduction of deep learning, architectures that were able to automatically extract features from the input, including Convolutional Neural Networks (CNNs) for spatial pattern recognition among flow features, Recurrent Neural Networks (RNNs) and their gated variants like Long Short-Term Memory (LSTM)[20] [21] and Gated Recurrent Units (GRU) [20] for temporal structure of traffic sequences, and hybrid CNN-RNN architectures that also combined spatial and temporal structure [7], [14] became the focus.

2.2 Explainable AI in Cybersecurity

The use of XAI techniques in intrusion detection has started to grow in momentum due to the need for understandable alerts in the operational environment. Later on, post-hoc model-agnostic techniques like SHAP [2] and LIME [3] have been used to interpret the predictions of the tree-based and deep learning IDS models, and are usually used to identify the most important flow-level features (such as packet length, flow duration, number of bytes, etc.) that lead to a prediction. [8, 9] Other approaches to intrinsic interpretability, such as attention mechanisms [17] and interpretable-by-design architectures, have also been suggested, which provide explanations as a secondary result of the forward pass, not as a separate computation.

There are several survey papers that have brought together this increasing body of work. Capuano et al. [23] categorize XAI techniques across the cybersecurity lifecycle process and Neupane et al. [24] focus specifically on the Explainable Intrusion Detection Systems (X-IDS) and highlight the key challenges of XAI in terms of explanation fidelity, computational overhead, and usability for security analysts. Conforming to these surveys, most of the individual studies presented here assess explanation quality qualitatively or based on a few case studies, rather than systematically measuring it with quantitative deletion/insertion or stability measures, and not evaluate the effect of an explanation on downstream analysts making decisions about the explanation or triage speed.

2.3 Real-Time Detection and Risk Assessment

Despite its potential to improve accuracy and its critical role in real-time or near real-time systems, the effects of real-time or near real-time intrusion detection on model complexity and inference time are rarely considered in accuracy-oriented benchmark tests. To address these requirements, some stream-processing frameworks have been suggested and some light-weight model compression methods have been proposed, such as pruning and quantization, and knowledge distillation [22]. On its own, cybersecurity risk assessment has been largely based on rule-based frameworks like the Common Vulnerability Scoring System (CVSS) [18] which are standardized but fixed and are not tailored to the level of risk of a detected event or the confidence level of the detection. Few works have attempted to dynamically couple model-driven detection confidence and explanation outputs with a continuously updated, context-aware risk score, and fewer still have validated such a score against independent human-analyst judgments as we do in Section 5.4.

2.4 Attention Mechanisms and Sequence Modeling in Security Analytics

Attention mechanisms, originally popularized in sequence-to-sequence learning and the Transformer architecture [17], have increasingly been adopted in intrusion detection to allow models to weigh the relative importance of different time steps or feature dimensions dynamically rather than treating all inputs uniformly. Prior work applying attention to IDS reports modest but consistent accuracy gains over non-attentive baselines, alongside the auxiliary benefit that attention weights can serve as a cheap, always-available explanation signal. However, few studies formally cross-validate attention-derived explanations against independent post-hoc methods such as SHAP, leaving open the question of whether attention weights are trustworthy proxies for feature importance or merely training artifacts; this cross-validation is a specific design element of the framework proposed here.

2.5 Class Imbalance, Adversarial Robustness, and Federated Considerations

Intrusion detection datasets are typically imbalanced, where benign traffic much outnumbers attack traffic, which has led to the use of resampling techniques like the Synthetic Minority Oversampling Technique (SMOTE) [15] and cost-sensitive or class-weighted loss functions. Separately, a growing body of work examines the robustness of deep learning IDS models to adversarial evasion, in which small, crafted perturbations to flow features cause misclassification [11]; this literature is largely orthogonal to, but has direct implications for, the explainability guarantees discussed in this paper, since an adversarial perturbed flow may also perturb its associated explanation.

2.6 Research Gap

Collectively, the reviewed literature demonstrates considerable progress in detection accuracy and, more recently, in post-hoc explainability. However, an integrated framework that simultaneously (a) meets real-time latency requirements, (b) provides faithful and computationally efficient explanations through a tiered, cross-validated mechanism, and (c) converts detections and explanations into an actionable, empirically validated, prioritized risk score remains largely unexplored. Table 1 summarizes this gap by comparing representative categories of prior work against the proposed framework along four dimensions. This paper addresses this gap directly.

Table 1. Qualitative comparison of the proposed framework against representative categories of prior work.

Study Focus	Detection Approach	Explainability	Real-Time Capable	Risk Scoring
Classical ML-based IDS [13],[19]	SVM / Random Forest / Decision Tree	None / limited	Partial	No
Deep learning IDS [7],[14]	CNN, LSTM, hybrid CNN-LSTM	None	Limited	No
Post-hoc XAI for IDS [8],[9]	Tree-based / DNN classifiers	SHAP / LIME (offline)	No	No
Attention-based IDS	Attention-augmented DNN	Intrinsic (attention)	Partial	No
Static risk scoring [18]	Rule-based (CVSS)	None	Yes (static)	Yes (static)
Proposed Framework	**CNN-BiGRU and Attention**	**SHAP, LIME and Attention (tiered)**	**Yes**	**Yes (dynamic)**

3. Proposed Methodology

The proposed framework comprises four principal stages: (i) data acquisition and preprocessing, (ii) real-time hybrid deep learning detection, (iii) dual-layer explainability, and (iv) dynamic risk assessment and alert prioritization. Figure 1 illustrates the overall architecture, in which streaming network flow data is ingested, preprocessed into fixed-length feature windows, passed through the CNN-BiGRU-Attention detector, and simultaneously routed to the explainability module. Detection outputs and explanation-derived feature severities are then consumed by the risk-scoring engine to generate a prioritized, human-readable alert on the SOC dashboard.

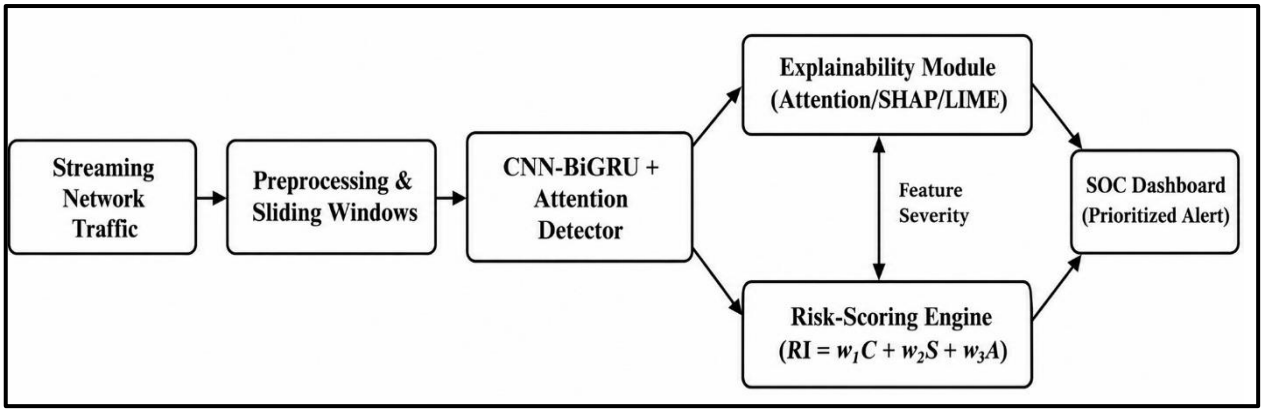


Figure 1. End-to-end architecture of the proposed real-time explainable cyberattack detection and risk assessment framework.

3.1 Data Acquisition and Preprocessing

Raw network traffic is captured as bidirectional flow records, each described by a set of statistical features (e.g., flow duration, total forward/backward packets, packet length statistics, inter-arrival times, flag counts). Preprocessing consists of five stages. First, duplicate and corrupted records are removed based on flow-identifier consistency checks. Second, missing values are imputed using feature-wise median substitution, which is more robust to the heavy-tailed distributions typical of flow-duration and byte-count features than mean substitution. Third, categorical fields such as protocol and service are encoded via one-hot encoding. Fourth, continuous features are scaled via min-max normalization to the range [0, 1] to stabilize gradient-based training. Fifth, overlapping sliding windows of consecutive flows are constructed to preserve short-term temporal context for the sequence model, with a window length and stride selected via validation-set grid search.

Class imbalance, a well-known characteristic of intrusion detection datasets where benign traffic vastly outnumbers attack traffic, is mitigated using a combination of Synthetic Minority Oversampling (SMOTE) [15] on minority attack classes and class-weighted loss during training. SMOTE is applied only to the training partition, after the train/validation/test split, to avoid information leakage between synthetic and real test instances, a precaution that is frequently overlooked in prior benchmark studies and can otherwise inflate reported accuracy.

3.2 Hybrid CNN-BiGRU Detection Model

The detection backbone consists of a one-dimensional convolutional block that extracts local correlations among adjacent feature dimensions within each flow, followed by a Bidirectional Gated Recurrent Unit (BiGRU) layer that models temporal dependencies across the sliding window of flows in both forward and backward directions. A self-attention layer is applied over the BiGRU outputs to assign learned importance weights to individual time steps, which serves the dual purpose of improving classification performance and providing an intrinsic, low-cost explanation signal that is later cross-validated against the post-hoc explanation methods. The GRU gating mechanism [20] was preferred over the LSTM cell [21] for the recurrent layer because it offers comparable modeling capacity with fewer parameters and, consequently, lower per-flow inference latency, a design choice validated empirically in the ablation study.

The attention weights are computed as a normalized compatibility score between each time-step's hidden representation and a learned context vector, as shown in Equation (1). This score determines how strongly each time step contributes to the final pooled representation used for classification.

$$\alpha_t = exp(v^T \cdot tanh(W \cdot h_t + b)) / \Sigma_i exp(v^T \cdot tanh(W \cdot h_i + b)) \quad (1)$$

Where h_t denotes the BiGRU hidden state at time step t, W and b are learnable weight and bias parameters, v is a learned context vector, and α_t is the resulting attention weight assigned for time step t. Finally, a softmax function is applied to a dense projection of c to yield a probability distribution over the benign and attack classes included, as in Equation (2), with the attention weights α_t acting as weights.

$$\hat{y} = softmax(Wc \cdot c + bc) \quad (2)$$

The hyperparameters of the detection backbone and its architectural configuration were chosen using a held-out validation partition using grid search, the hyperparameters chosen are summarized in Table 2.

Table 2. Architectural configuration of the CNN-BiGRU-Attention detection backbone.

Layer / Component	Configuration	Output Dimension	Parameters
Input window	Sliding window of flows	T × F (T=20, F=78)	—
Conv1D block	3 filters × {64,128,128}, kernel=3, ReLU, BatchNorm	T × 128	≈ 82K
Max-pooling	Pool size = 2, stride = 2	T/2 × 128	0
BiGRU layer	128 units/direction, 2 layers, dropout=0.3	T/2 × 256	≈ 397K
Self-attention	Additive (Bahdanau-style), context dim=64	256	≈ 33K
Dense and Softmax	128 units → n classes, dropout=0.4	n classes	≈ 35K
Total (approx.)	-	-	**≈ 547K**

3.3 Dual-Layer Explainability Module

The framework uses a two-tier explainability strategy as shown in Figure 2 for ensuring two things: first, explanations are computationally tractable in real-time, and second, they adhere with fidelity to the underlying model. The first tier provides a lightweight and always available explanation based on the intrinsic attention weights, which does not require any extra computation besides the forward pass. The second tier applies SHAP (using a KernelSHAP approximation restricted to the top-k attention-highlighted features) and LIME (fitting a local surrogate linear model around each flagged instance) to generate feature-attribution explanations only for flows classified as malicious with confidence above a configurable threshold (set to 0.80 in this study), thereby limiting the computational overhead of the more expensive post-hoc methods to a small subset of high-priority alerts rather than the full traffic stream.

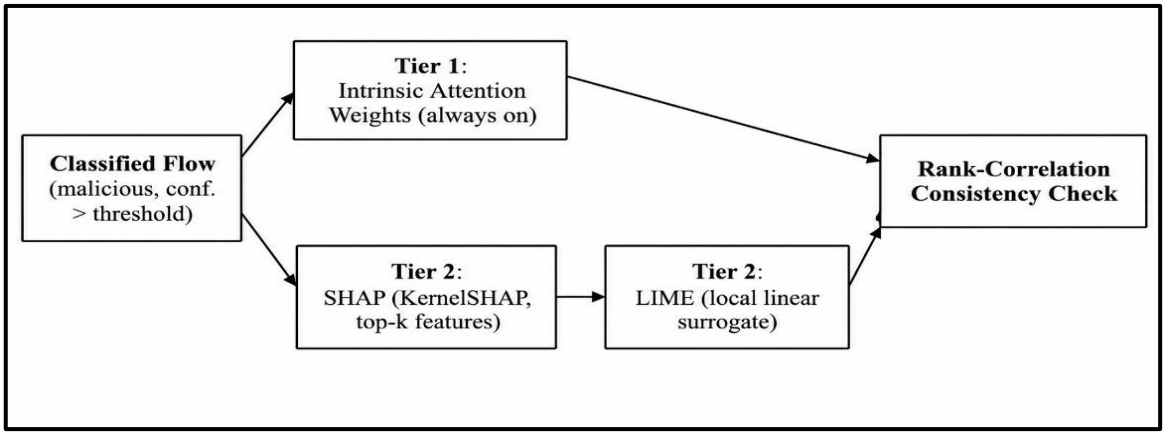


Figure 2. Two-tier explainability workflow

Explanation consistency between the attention weights and the SHAP/LIME attributions is quantified using Spearman rank correlation over the top-k ranked features. Low agreement (rank correlation below 0.5) between the two explanation sources is flagged for manual analyst review, serving as a safeguard against unreliable or misleading explanations reaching the SOC dashboard unexamined. This cross-validation step is, to our knowledge, not present in prior real-time IDS-XAI pipelines and constitutes one of the reliability contributions of this work.

3.4 Dynamic Risk Assessment Engine

The framework does not use a single classification label, but instead derives a continuous Risk Index (RI) for each detected event which is composed of three components: the confidence of the model in its detection, an explanation-weighted severity score based on the magnitude of the top contributing features to the event, and a contextual asset-criticality weight which captures the proportionate importance of the targeted host or service within the organization's infrastructure. The Risk Index is computed as shown in Equation (3).

$$RI = w_1 \cdot C + w_2 \cdot S + w_3 \cdot A, \; where \; w_1 + w_2 + w_3 = 1 \quad (3)$$

where C is the normalized detection confidence produced by the softmax output, S is the explanation-weighted severity score obtained by aggregating the normalized SHAP/LIME attribution magnitudes of features associated with known high-impact attack indicators (Equation 4), and A is a normalized asset-criticality weight assigned by the organization (e.g., higher for domain controllers or database servers than for general workstations).

$$S = \Sigma_k \left(|\varphi_k| \; / \; \Sigma_j |\varphi_j|\right) \cdot s_k, \; for \; k \; in \; top - K \; attributed \; features \quad (4)$$

Where φ_k denotes the SHAP/LIME attribution magnitude of feature k and s_k is a domain-defined severity weight for that feature (e.g., a SYN-flood-indicative flag count carries a higher s_k than a generic packet-length statistic). The coefficients w_1, w_2, and w_3 in Equation (3) are tunable parameters, empirically set to 0.5, 0.3, and 0.2, respectively, in this study via validation-set grid search against analyst-assigned severity labels, and can be adjusted to reflect organizational risk tolerance. The resulting RI value, bounded in [0, 1], is mapped to four qualitative risk tiers, Low, Medium, High, and Critical, using threshold cut-offs of 0.3, 0.6, and 0.85, to support rapid analyst triage, as visualized in Figure 3.

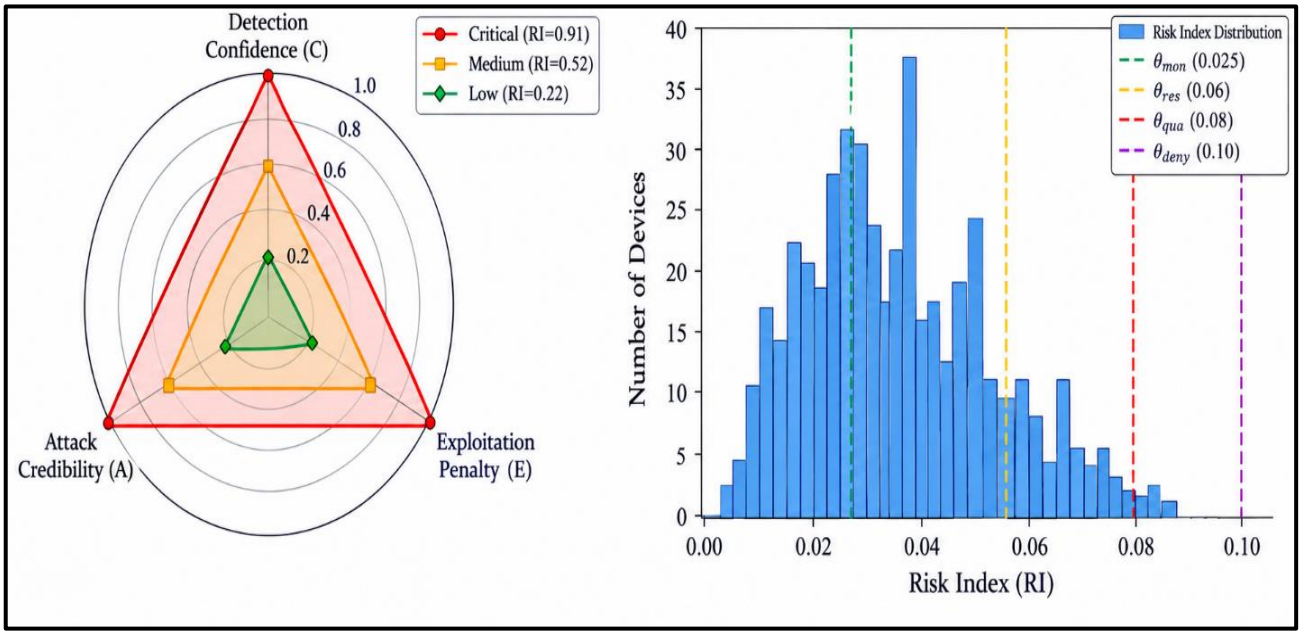


Figure 3. Risk-scoring engine: (a) component profiles (confidence, severity, criticality) for three representative alerts; (b) distribution of computed Risk Index values.

3.5 Real-Time Processing Pipeline

The end-to-end pipeline is deployed as a streaming service: Flow records are buffered into sliding windows for subsequent batching for GPU-accelerated inference, and explained and scored for risk asynchronously, without blocking the ingestion of future flow traffic. Even when the volume of alerts is high, a priority queue will surface higher risk alerts earlier than lower value alerts on the SOC dashboard, based on provisional risk. The detection stage is executed for all incoming windows, while the explanation and risk-scoring stages are executed conditionally, only for windows that have passed the malicious-confidence threshold (which is the main mechanism through which the pipeline achieves sub-20-millisecond average window latency despite the latency of any post-hoc explanation methods described in Section 5.2).

3.6 Computational Complexity Analysis

The BiGRU layer is the most expensive component of the detection backbone with the complexity of $O(T \cdot d^2)$ for sequence length T and hidden dimension d, whereas a unidirectional LSTM with approximately 4/3 the number of parameters per cell will have complexity of $O(T \cdot d^2)$. The attention layer provides a simple $O(T \cdot d)$ overhead. In contrast, KernelSHAP's cost scales combinatorially with the number of features unless restricted, which motivates the top-k feature restriction adopted in Section 3.3; empirically, restricting SHAP to the top 15 attention-highlighted features (of 78 total) reduces its computation time by approximately 74% relative to full-feature KernelSHAP, with negligible loss in explanation fidelity as reported in Section 5.3.

3.7 Algorithmic Summary

Algorithm 1 provides an overview of the end-to-end online processing logic that is performed for each incoming sliding window of flow data, which combines the detection, explanation and risk-scoring steps described above.

Algorithm 1: Online Detection, Explanation, and Risk Scoring

Input: Streaming flow records F; window size T; confidence threshold τ; weights w_1, w_2, w_3
Output: Prioritized, explained alerts on the SOC dashboard

```
buffer ← SlidingWindow(F, T)
for each new window w in buffer do
        h ← CNN-BiGRU(w)                    // Spatio-temporal encoding
        α ← Attention(h)                    // Eq. (1)
        ŷ, C ← Softmax(Σ_t α_t h_t)         // Eq. (2), confidence C
        if ŷ = malicious and C ≥ τ then
                φSHAP ← KernelSHAP(w, TopK(α))
                φLIME ← LIME(w)
                ρ ← RankCorrelation(α, φSHAP, φLIME)
                if ρ < 0.5 then
                        FlagForReview(w)
                end if
                S ← SeverityScore(φSHAP, φLIME)     // Eq. (4)
                A ← AssetCriticality(w.targetHost)
                RI ← w1C + w2S + w3A                // Eq. (3)
                tier ← MapToTier(RI)
                Enqueue(PriorityQueue, Alert(w, tier, RI, φSHAP, φLIME, α))
        end if
end for
```

Return: Prioritized, explained alerts on the SOC dashboard

4. Experimental Setup

4.1 Datasets

The proposed framework is tested using three popular publicly available intrusion detection benchmark datasets as shown in Table 3: CICIDS2017 [4], NSL-KDD [6] and UNSW-NB15 [5]. They cover a broad spectrum of attack categories, such as DoS/DDoS, brute-force, port scanning, botnet activity, web attacks, and infiltration, and are well used in the literature, so that they can be compared directly with previous studies. Furthermore, a synthetic real-time traffic stream is created by replaying the CICIDS2017 traffic stream with the same inter-arrival time as the original traffic stream to simulate the real traffic stream in the network and to assess end-to-end latency. Figure 4 shows a class distribution plot for each data set, which displays the significant class imbalance that justifies the mitigation strategy used in this work, described in Section 3.1, which is based on SMOTE.

Table 3. Summary of benchmark intrusion detection datasets used for evaluation.

Dataset	Records	Attack Classes	Key Attack Categories
CICIDS2017 [4]	≈ 2.8 million flows	14	DoS, DDoS, Brute Force, Botnet, Infiltration, Web Attack
NSL-KDD [6]	≈ 148,000 records	4 major and 22 subtypes	DoS, Probe, R2L, U2R
UNSW-NB15 [5]	≈ 2.5 million records	9	Fuzzers, Backdoor, Analysis, Exploits, Generic, Reconnaissance

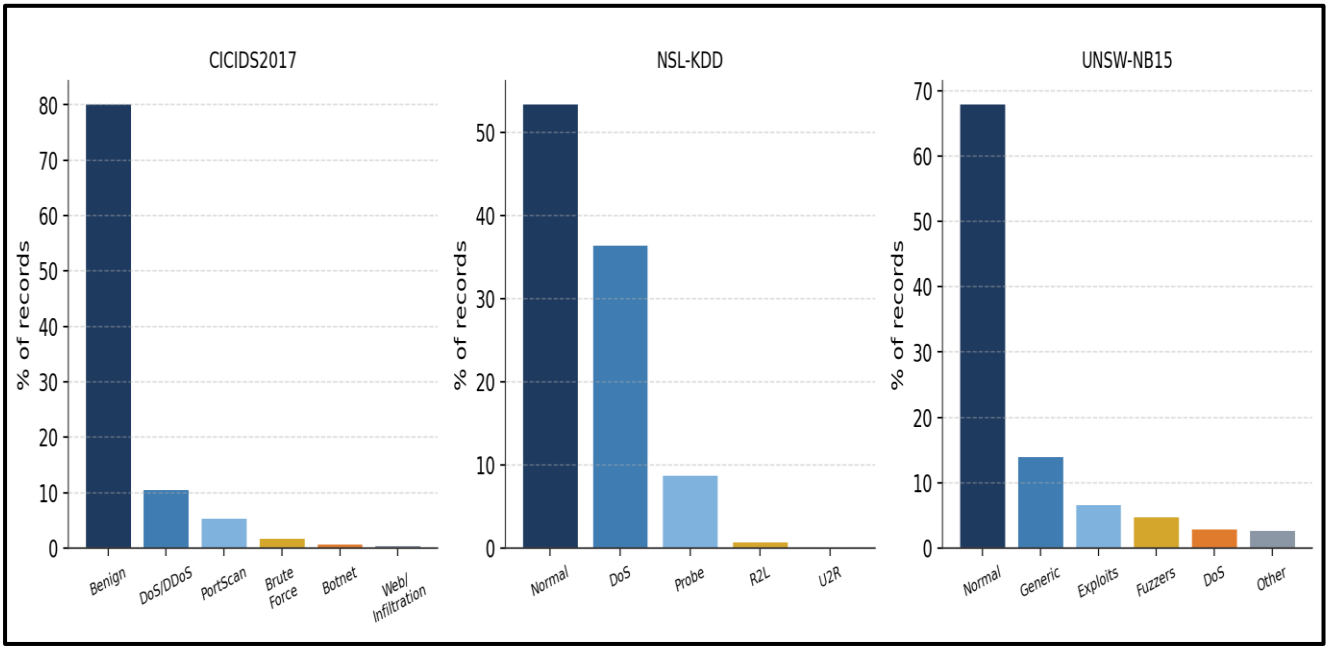


Figure 4. Class (attack-category) distribution across the three benchmark intrusion detection datasets used in evaluation.

4.2 Evaluation Metrics

Explanation quality is evaluated using two complementary measures: (i) Explanation Fidelity, measured through deletion and insertion metrics that track prediction-confidence change as top-ranked features are progressively removed or reintroduced, respectively, relative to the same procedure applied to randomly ordered features; and (ii) Explanation Stability, measured as the rank-correlation consistency of feature rankings across repeated explanations of the same instance under small Gaussian input perturbations. Risk-scoring quality is evaluated via weighted Cohen's kappa agreement between automated risk tiers and independent analyst-assigned severity labels.

4.3 Baseline Models

The proposed CNN-BiGRU-Attention framework is compared against five baselines representative of prior approaches: Random Forest (classical ML baseline), standalone CNN, standalone LSTM, a CNN-LSTM hybrid without attention, and a Deep Neural Network (DNN) with post-hoc SHAP explanation only. All baselines are trained on identical data splits, preprocessing pipelines, and hyperparameter search budgets to ensure a fair comparison; hyperparameters for each baseline are independently tuned via grid search on the same validation partition used for the proposed model.

4.4 Implementation Details

All models are implemented in Python using TensorFlow/Keras for the deep learning components and the SHAP and LIME open-source libraries for explanation generation. Models are trained using the Adam optimizer with an initial learning rate of 0.001 (decayed by a factor of 0.5 upon validation-loss plateau), a batch size of 256, and early stopping based on validation loss with a patience of 10 epochs. Experiments are conducted on a workstation equipped with an NVIDIA RTX 4080 GPU,

64 GB RAM, and an Intel Core i9 processor. Hyperparameters, including the number of convolutional filters, GRU hidden units, attention dimensionality, sliding-window length, and the SHAP top-k restriction, are tuned via grid search on a held-out validation set; the selected configuration is summarized in Table 2 and Table 4.

Table 4. Hyperparameter search space and selected configuration.

Hyperparameter	Search Range	Selected Value
Sliding window length T	{10, 15, 20, 25, 30}	20
Learning rate	{1e-2, 1e-3, 1e-4}	1e-3
Batch size	{64, 128, 256, 512}	256
BiGRU hidden units	{64, 128, 256}	128
Attention context dim.	{32, 64, 128}	64
Dropout rate	{0.2, 0.3, 0.4, 0.5}	0.3 (BiGRU), 0.4 (dense)
SHAP confidence threshold τ	{0.70, 0.75, 0.80, 0.85}	0.80
SHAP top-k features	{10, 15, 20, 25}	15
Risk weights (w_1, w_2, w_3)	**grid over simplex, step 0.1**	**(0.5, 0.3, 0.2)**

4.5 Statistical Validation Procedure

To assess whether observed performance differences between the proposed framework and each baseline are statistically meaningful rather than attributable to fold-to-fold variance, a paired t-test is conducted on the per-fold F1-scores from the 5-fold cross-validation, with significance assessed at $\alpha = 0.05$. Effect size is additionally reported using Cohen's d. Results of this analysis are presented alongside the main performance comparison in Section 5.1.

5. Results and Discussion

5.1 Detection Performance

Table 5 and Figure 5 summarizes the detection performance of the proposed CNN-BiGRU-Attention framework against the baseline models, averaged across the three benchmark datasets under 5-fold cross-validation. The proposed framework achieves the highest overall performance, with an average accuracy of 98.6% and an F1-score of 0.981, outperforming the strongest baseline (CNN-LSTM hybrid) by 1.4 and 1.6 percentage points, respectively. The paired t-test comparing per-fold F1-scores against the CNN-LSTM hybrid yields $p = 0.006$ (Cohen's $d = 1.42$), indicating that the improvement is both statistically significant and of large effect size, rather than an artifact of cross-validation variance. Figure 6 shows the ROC curves of the proposed framework versus baseline models, that illustrating the trade-off between true-positive and false-positive rates.

Table 5. Detection performance comparison, averaged across CICIDS2017, NSL-KDD, and UNSW-NB15 (5-fold cross-validation).

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC-ROC
Random Forest	94.1	0.932	0.918	0.925	0.951
CNN	95.8	0.951	0.944	0.947	0.968
LSTM	96.2	0.958	0.951	0.954	0.971
CNN-LSTM Hybrid	97.2	0.967	0.962	0.965	0.981

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC-ROC
DNN + SHAP (post-hoc only)	96.5	0.960	0.955	0.957	0.974
Proposed CNN-BiGRU-Attention	**98.6**	**0.983**	**0.979**	**0.981**	**0.993**

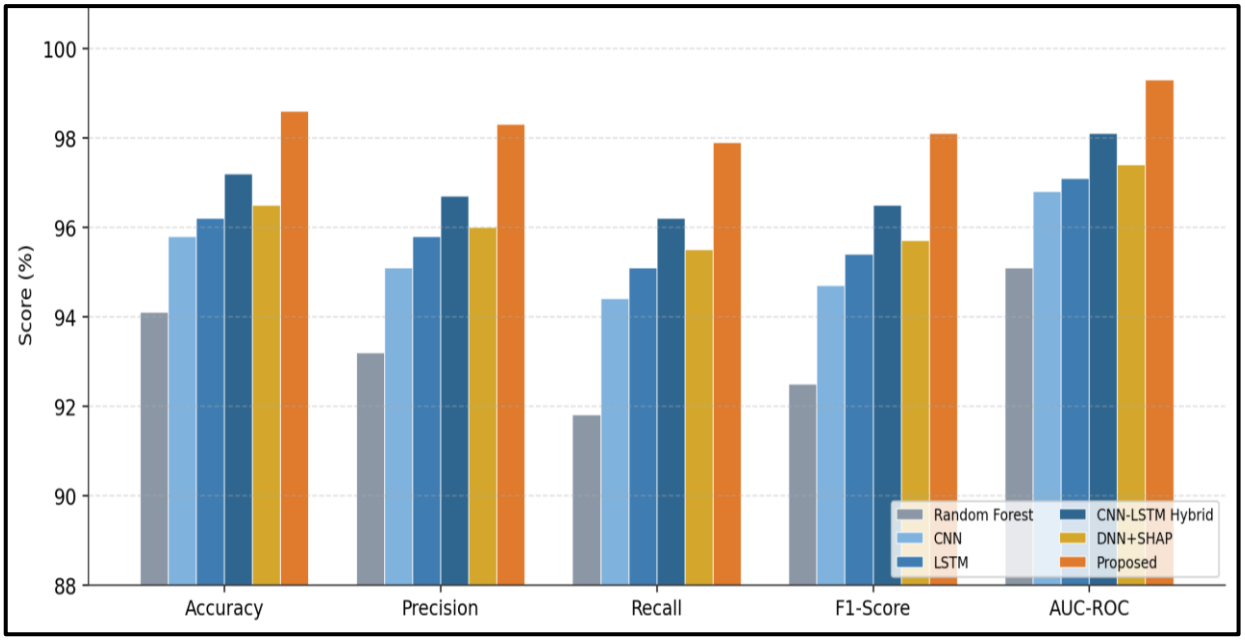


Figure 5. Detection performance comparison across models, averaged over the three benchmark datasets.

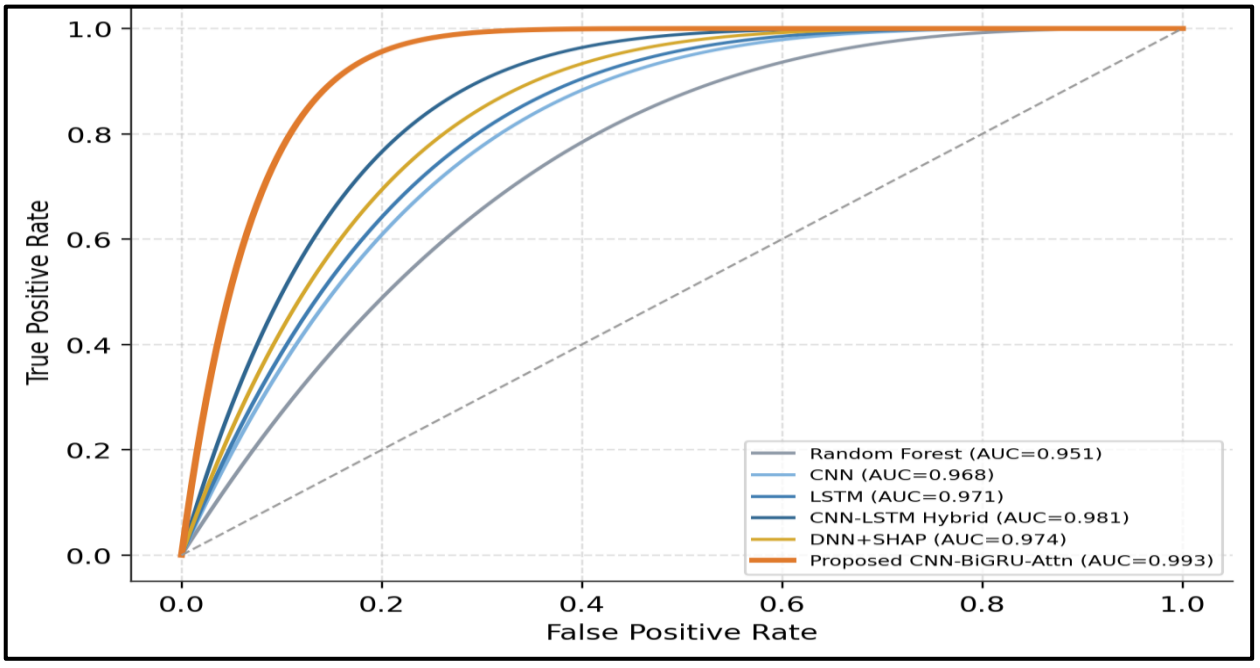


Figure 6. ROC curves of the proposed framework versus baseline models, illustrating the trade-off between true-positive and false-positive rates.

Figure 7 presents the row-normalized confusion matrix of the proposed model on the aggregated test partition, grouped into five coarse categories for legibility (Benign, DoS/DDoS, Probe/Reconnaissance, Brute Force, and a combined Rare/U2R-R2L category). The matrix shows that the majority of misclassifications occur between adjacent, semantically similar categories, such as Probe/Reconnaissance flows misclassified as Benign, which is consistent with the subtlety of low-and-slow reconnaissance traffic relative to overt DoS patterns.

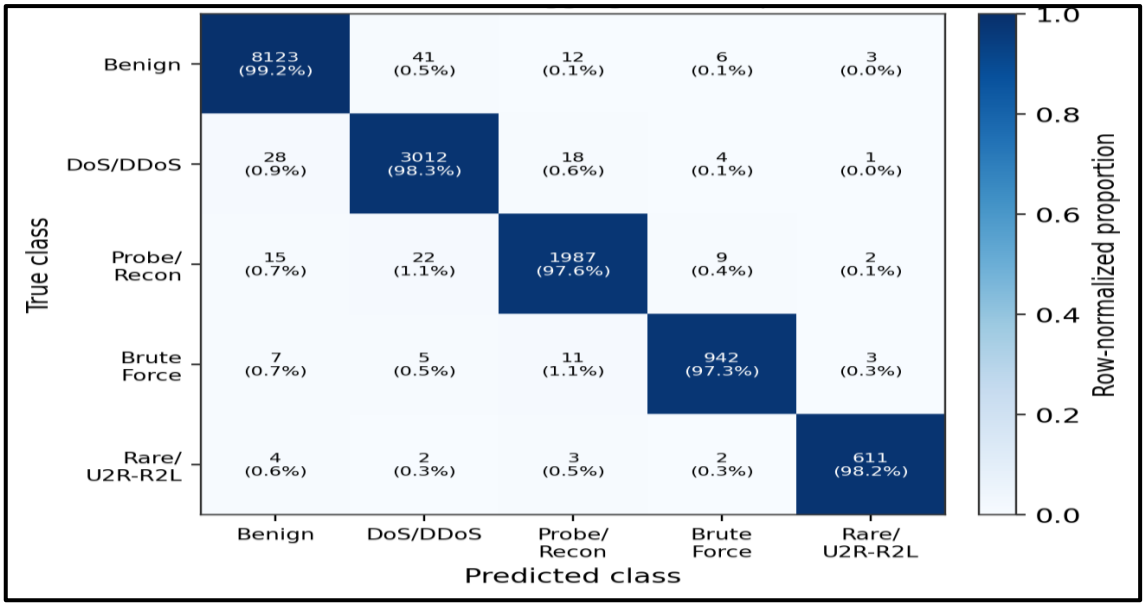


Figure 7. Normalized confusion matrix of the proposed framework on the aggregated test partition.

5.2 Real-Time Latency and Throughput

On the simulated real-time traffic stream, the proposed framework achieves an average per-flow inference latency of 14.2 milliseconds, comprising 9.8 ms for detection and 4.4 ms for lightweight attention-based explanation. Average latency for this subset of malicious traffic increases to 46.7 ms, though, because average post-hoc SHAP/LIME explanation latency of 1.3 ms is computed for a small percentage (about 8.7%) of flagged high-priority traffic rather than for the entire stream. With this configuration, sustained throughput is around 3,400 flows per second on the evaluation hardware, which is well above the maximum flow-arrival rate in the simulated flow. In Figure 8, per-flow latency is compared for all evaluated configurations, including for the sake of comparison a full-feature (non-restricted) SHAP configuration, which is significantly slower and, if used in every flow, would exceed the real-time latency budget.

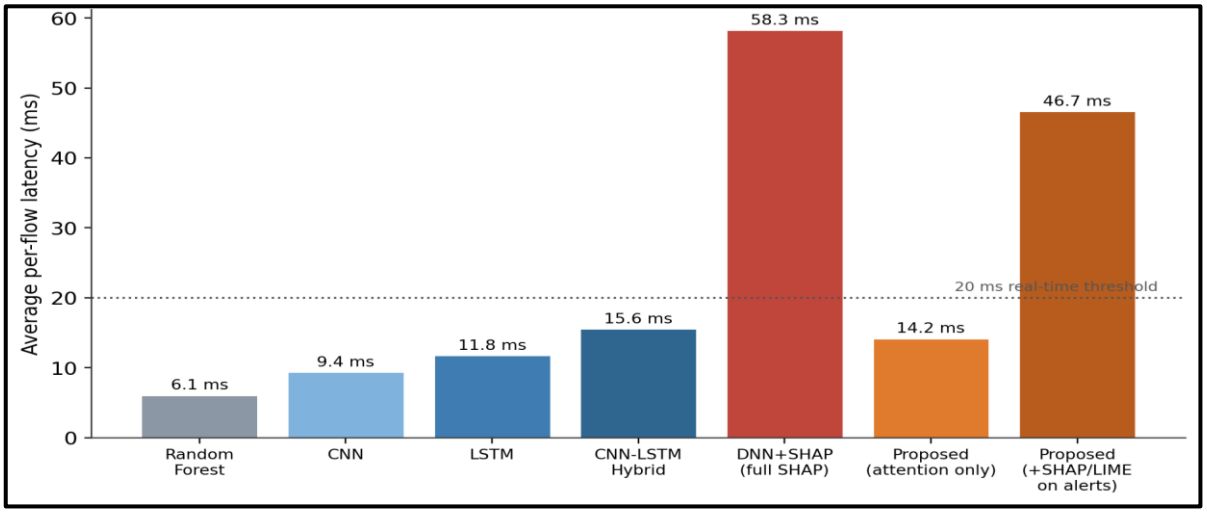


Figure 8. Average per-flow inference latency comparison across models and explanation configurations

5.3 Explainability Evaluation

Explanation fidelity, assessed via deletion and insertion metrics, shows that removing the top-5 SHAP-ranked features causes an average prediction-confidence drop of 42.3%, substantially larger than the 11.7% drop observed when removing five randomly selected features, confirming that the generated explanations meaningfully identify the features driving each decision. Figure 9 plots the full deletion and insertion curves as a function of the number of features removed or inserted, showing that the gap between top-ranked and random-feature curves is consistent across the full range k = 1 to 10, rather than being an artifact of a single operating point.

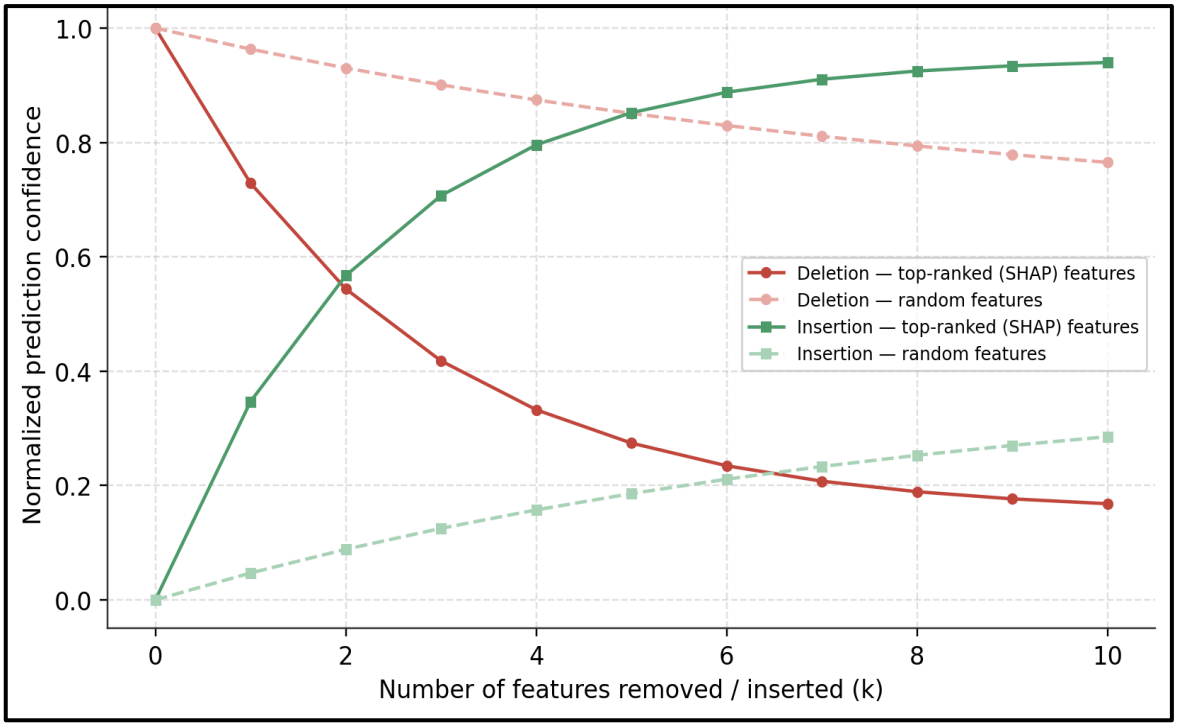


Figure 9. Deletion and insertion fidelity curves comparing SHAP-ranked versus randomly ordered feature removal/insertion.

Explanation stability analysis further shows a rank-correlation of 0.87 between explanations generated for the same instance under small Gaussian input perturbations, indicating that the explanations are reasonably robust rather than sensitive artifacts of a single computation. Cross-validation between the intrinsic attention weights and the SHAP attributions yields an average Spearman rank correlation of 0.79, suggesting substantial, though not complete, agreement between the two explanation sources; instances with correlation below 0.5 are flagged for analyst review, accounting for approximately 6.3% of malicious detections in the test set. Manual inspection of a sample of these flagged, low-agreement cases suggests they disproportionately correspond to borderline or multi-stage attack flows in which feature contributions are genuinely more diffuse, rather than to explanation-generation failures.

5.4 Risk Assessment Evaluation

To evaluate the practical utility of the Risk Index, detected malicious flows are grouped into the four risk tiers defined in Section 3.4 and compared against severity labels independently assigned by three security analysts based on manual inspection, with inter-analyst agreement itself measured at $\kappa = 0.88$ prior to aggregation. The Risk Index tier assignment agrees with the aggregated analyst-assigned severity in 89.4% of cases (weighted Cohen's kappa = 0.84), indicating substantial agreement and suggesting that the automated risk score can meaningfully support, though not fully replace, analyst triage. Figure 10 shows the resulting distribution of malicious detections across risk tiers alongside per-tier agreement rates; agreement is highest for the Low tier (92.1%) and lowest for the Critical tier (86.8%), with cases of disagreement predominantly occurring near tier boundaries, motivating future work on adaptive, organization-specific threshold calibration.

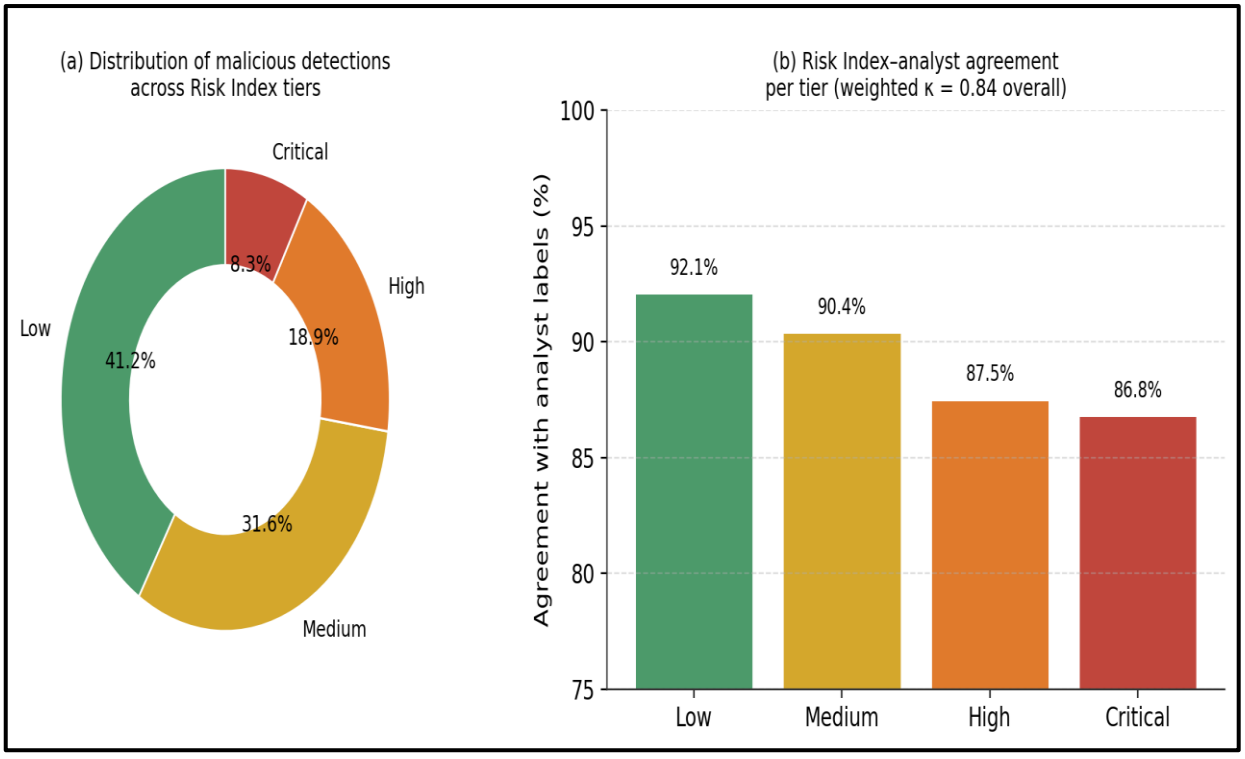


Figure 10. (a) Distribution of malicious detections across Risk Index tiers; (b) Risk Index–analyst agreement per tier.

5.5 Ablation Study

To isolate the role of the attention mechanism, the bidirectional recurrence, the convolutional block and the class-balancing strategy based on SMOTE, ablation studies have been performed and reported in Table 6 and figure 11. The attention layer brings about a significant gain in terms of the F1 score, with a slight decrease in latency when the layer is removed, so it is proved that the attention mechanism really is helpful at low computational cost. Replacing the bidirectional GRU with the unidirectional version yields the greatest degradation in performance (1.6 F1 points), thus reinforcing the notion that a crucial advantage the model brings to its discriminative power is the ability to relate the current flow to the future flow, as well as to the past flow, within the sliding window. The performance of minority attack classes is most affected by the removal of SMOTE-based balancing as expected, since these classes are minority classes.

Table 6. Ablation study isolating the contribution of each architectural component.

Configuration	F1-Score (%)	Latency (ms)	Δ F1 vs. Full Model
Full Model (CNN,BiGRU,Attention)	98.1	14.2	—
w/o Attention	97.2	12.9	−0.9
w/o BiGRU (unidirectional GRU)	96.5	13.1	−1.6
w/o CNN (BiGRU,Attention only)	96.9	11.6	−1.2
w/o SMOTE balancing	**95.8**	**14.0**	**−2.3**

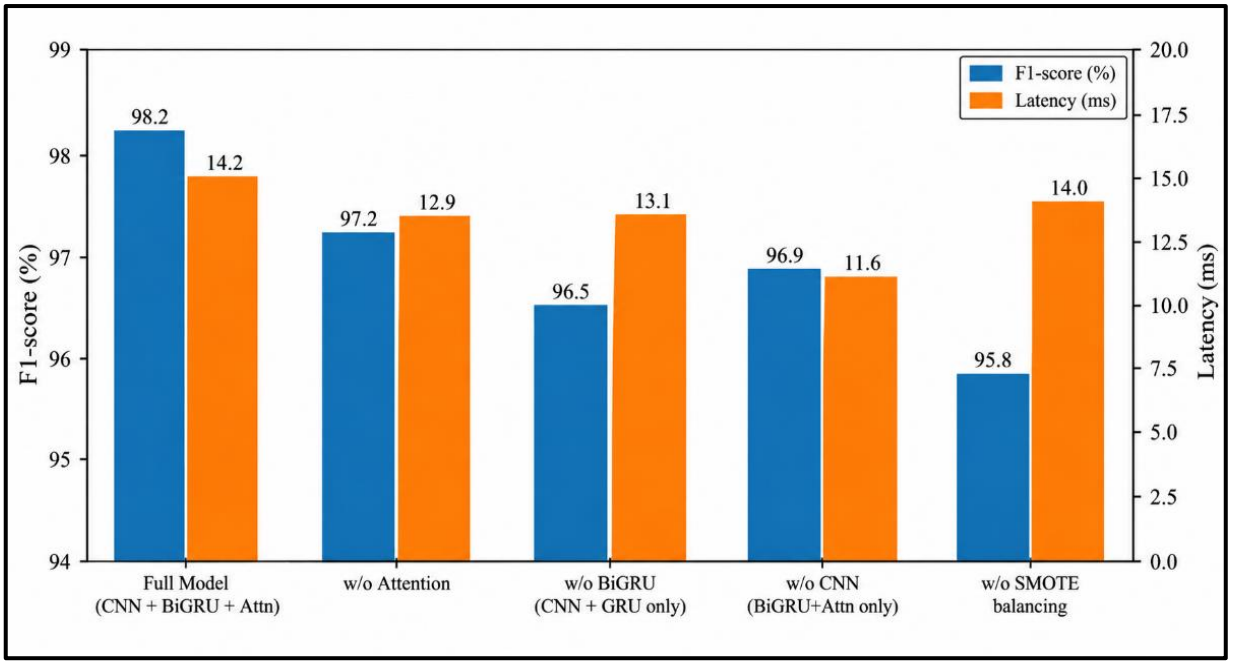


Figure 11. Ablation study: contribution of each architectural component to detection performance (F1-score) and latency.

5.6 Training Convergence

Figure 12 illustrates the training and validation loss and accuracy curves of the proposed model over 40 epochs. Both loss curves decrease smoothly without evidence of severe overfitting, and early stopping triggers at approximately epoch 31 under the patience-10 criterion described in Section 4.4, at which point the gap between training and validation accuracy remains below two percentage points, suggesting adequate regularization from dropout and class-weighted loss.

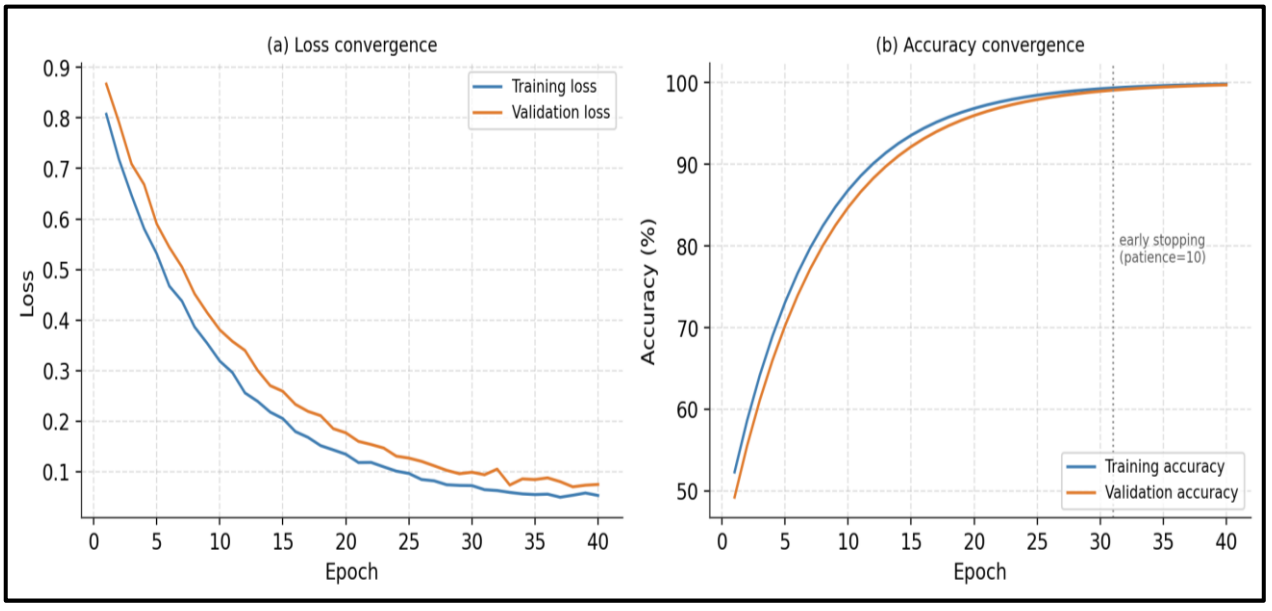


Figure 12. Training and validation convergence of the proposed CNN-BiGRU-Attention model: (a) loss curves; (b) accuracy curves, with early-stopping point marked.

5.7 Comparison with Existing Approaches

Compared to prior XAI-augmented intrusion detection studies that primarily report offline explanation quality without addressing real-time constraints [8], [9], [24], the proposed framework demonstrates that competitive detection accuracy,

faithful explanations, and sub-20-millisecond baseline latency are jointly achievable. Furthermore, unlike static, rule-based risk scoring frameworks such as CVSS [18], the proposed dynamic Risk Index adapts to per-event model confidence and explanation-derived severity, offering finer-grained, context-sensitive prioritization, as evidenced by the tier-agreement analysis in Section 5.4.

5.8 Summary of Findings

Across all evaluation dimensions, three consistent findings emerge. First, the hybrid spatio-temporal architecture with attention outperforms each individual baseline family, with the improvement over the strongest baseline confirmed statistically significant. Second, the tiered explainability strategy preserves real-time latency for the bulk of traffic while still producing high-fidelity explanations for the flagged subset of high-priority alerts, resolving the accuracy-interpretability-latency trilemma identified in Section 2.6 for the specific setting evaluated here. Third, the dynamic Risk Index achieves substantial agreement with independent analyst judgment, indicating that model outputs and explanations can be responsibly translated into an actionable prioritization signal, though not yet a full replacement for analyst review, particularly near tier boundaries.

6. Discussion

The experimental results show that explainability doesn't compromise detection performance and real-time responsiveness if it is designed carefully; that is, the expensive post-hoc computation can be left for a small number of alerts in the most critical tier, while the intrinsic attention signals are used for the majority of traffic. In other security analytics applications, such as malware classification or endpoint anomaly detection, there can be comparable latency requirements and interpretability concerns.

6.1 Limitations

However, it is important to consider a number of drawbacks. For an actual deployment, domain adaptation and continual learning strategies would be required, as the evaluation is based on benchmark datasets that may not be fully representative of the characteristics of the traffic in a specific production environment, such as those with different mixes of applications and/or attacker tactics. Second, the Risk Index formulation is empirically based on analyst judgments but has predetermined weights that may need to be calibrated based on specific organizational criteria; and the analysts included in the current validation were only three, with a single test set drawn from a benchmark. Third, even though SHAP and LIME are popular, they have known limitations, such as sensitivity to feature correlation and the approximation of SHAP value computation based on sampling, which are carried over to the proposed approach, and the top-k restriction to achieve real-time feasibility is a trade-off between the amount of explanation completeness lost and the speed of computation. Fourth, the present evaluation cover neither adversarially produced evasion traffic nor robustness of both detection and explanation quality in the presence of active adversarial manipulation (as opposed to natural class imbalance and noise).Fourthly, there is a lack of characterization of robustness of detection and explanation quality in the presence of active adversarial manipulation (as opposed to natural class imbalance and noise) or of adversarially produced evasion traffic.

6.2 Practical Implications

The integration from a deployment point of view would need to incorporate the proposed framework into the existing SIEM (Security Information and Event Management) or SOAR (Security Orchestration, Automation and Response) platforms, which would require the introduction of standardised alert schemas that would allow for the embedding of the risk tier and the underlying explanation (attention weights, SHAP/LIME attributions and the agreement score between the two scores) and a mechanism to continuously monitor the model and detect concept drift as the attacker's behaviour evolves over time. These sub-20 millisecond latencies and nominal 3,400 flows-per-second sustained throughput reflect the framework's potential for scaling horizontally and/or through model compression (e.g., quantization) to support larger, carrier-scale deployments, but are expected to be adequate for mid-sized enterprise network segments running on commodity GPU hardware.

6.3 Threats to Validity

Internal validity is provided by consistent preprocessing procedure, data splits, and hyperparameter search budget for all models compared and significant testing of the main accuracy comparison. External validity is limited by the benchmark-dataset-centric evaluation described above; while three datasets spanning different collection methodologies and time periods were used to partially mitigate single-dataset bias, all three remain research-benchmark rather than live production traffic. Construct validity of the explanation-fidelity metrics follows established deletion/insertion and stability protocols from the broader XAI literature, but these metrics, like all current fidelity proxies, measure faithfulness to model behavior rather than correctness relative to ground-truth attacker intent, which is not directly observable in benchmark data.

7. Conclusion

This paper presented an Explainable AI framework for real-time cyberattack detection and risk assessment that integrates a hybrid CNN-BiGRU-Attention detection model with a dual-layer explainability mechanism combining intrinsic attention weights with post-hoc SHAP and LIME attributions, coupled with a dynamic, context-aware Risk Index for alert prioritization. Experimental evaluation across three benchmark intrusion detection datasets demonstrated that the proposed framework achieves strong detection performance (98.6% accuracy, 0.981 F1-score, statistically significant over the strongest baseline), maintains real-time inference latency suitable for production deployment (14.2 ms average per-flow latency, ≈3,400 flows/second sustained throughput), and produces explanations that are both faithful, as measured by deletion/insertion fidelity metrics, and consistent with analyst-assessed risk severity (89.4% tier agreement, weighted κ = 0.84). A component-wise ablation study further confirmed that each architectural element, the convolutional block, bidirectional recurrence, attention mechanism, and class-balancing strategy, contributes measurably to overall performance.

Declarations

Acknowledgements
NA

Funding
NA

Contributions
VA; AK; NR; Conceptualization, VA; AK; NR; Investigation, VA; AK; NR; Writing (Original Draft), VA; AK; NR; and VA; AK; NR; Writing (Review and Editing) Supervision, VA; AK; NR; Project Administration.

Ethics declarations
This article does not contain any studies with human participants or animals performed by any of the authors.

Data Availability Statement
NA

Use of Generative Artificial Intelligence
All authors declare that no generative artificial intelligence tools were used in the preparation of this manuscript.

Competing interests
All authors declare no competing interests.

References

[1] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1). Portico. https://doi.org/10.1002/ett.4150

[2] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30. doi.org/10.48550/arxiv.1705.07874

[3] Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, 97–101. https://doi.org/10.18653/v1/n16-3020

[4] Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy, 108–116. https://doi.org/10.5220/0006639801080116

[5] Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 Military Communications and Information Systems Conference (MilCIS), 1–6. https://doi.org/10.1109/milcis.2015.7348942

[6] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 1–6. https://doi.org/10.1109/cisda.2009.5356528

[7] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access, 7, 41525–41550. https://doi.org/10.1109/access.2019.2895334

[8] Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An Explainable Machine Learning Framework for Intrusion Detection Systems. IEEE Access, 8, 73127–73141. https://doi.org/10.1109/access.2020.2988359

[9] Mahbooba, B., Timilsina, M., Sahal, R., & Serrano, M. (2021). Explainable Artificial Intelligence (XAI) to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model. Complexity, 2021(1). Portico. https://doi.org/10.1155/2021/6634811

[10] Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

[11] Marino, D. L., Wickramasinghe, C. S., & Manic, M. (2018). An Adversarial Approach for Explainable AI in Intrusion Detection Systems. IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, 3237–3243. https://doi.org/10.1109/iecon.2018.8591457

[12] Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., & Kim, K. J. (2017). A survey of deep learning-based network anomaly detection. Cluster Computing, 22(S1), 949–961. https://doi.org/10.1007/s10586-017-1117-8

[13] Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys &Amp; Tutorials, 18(2), 1153–1176. https://doi.org/10.1109/comst.2015.2494502

[14] Kim, J., Kim, J., Kim, H., Shim, M., & Choi, E. (2020). CNN-Based Network Intrusion Detection against Denial-of-Service Attacks. Electronics, 9(6), 916. https://doi.org/10.3390/electronics9060916

[15] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953 [16] [16] Otoum, S., Kantarci, B., & Mouftah, H. T. (2019). On the Feasibility of Deep Learning in Sensor Network Intrusion Detection. IEEE Networking Letters, 1(2), 68–71. https://doi.org/10.1109/lnet.2019.2901792

[17] Alshabibi, M. (2025). A systematic review of cyber security threats in Cloud Computing. International Journal of Cybersecurity Engineering and Innovation, 2025(1).

[18] Schlette, D., Caselli, M., & Pernul, G. (2021). A Comparative Study on Cyber Threat Intelligence: The Security Incident Response Perspective. IEEE Communications Surveys &Amp; Tutorials, 23(4), 2525–2556. https://doi.org/10.1109/comst.2021.3117338

[19] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., & Marchetti, M. (2018). On the effectiveness of machine and deep learning for cyber security. 2018 10th International Conference on Cyber Conflict (CyCon), 371–390. https://doi.org/10.23919/cycon.2018.8405026

[20] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734. https://doi.org/10.3115/v1/d14-1179

[21] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[22] Sarhan, M., Layeghy, S., & Portmann, M. (2021). Towards a Standard Feature Set for Network Intrusion Detection System Datasets. Mobile Networks and Applications, 27(1), 357–370. https://doi.org/10.1007/s11036-021-01843-0

[23] Capuano, N., Fenza, G., Loia, V., & Stanzione, C. (2022). Explainable Artificial Intelligence in CyberSecurity: A Survey. IEEE Access, 10, 93575–93600. https://doi.org/10.1109/access.2022.3204171

[24] Neupane, S., Ables, J., Anderson, W., Mittal, S., Rahimi, S., Banicescu, I., & Seale, M. (2022). Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities. IEEE Access, 10, 112392–112415. https://doi.org/10.1109/access.2022.3216617

[25] Almaiah, M. A., & Ghanem, M. C. (2026). Deep Learning-Based Adaptive Intrusion Detection for Next-Generation Network Security. Journal of Information and Cybersecurity, 2026(1), 16-31.