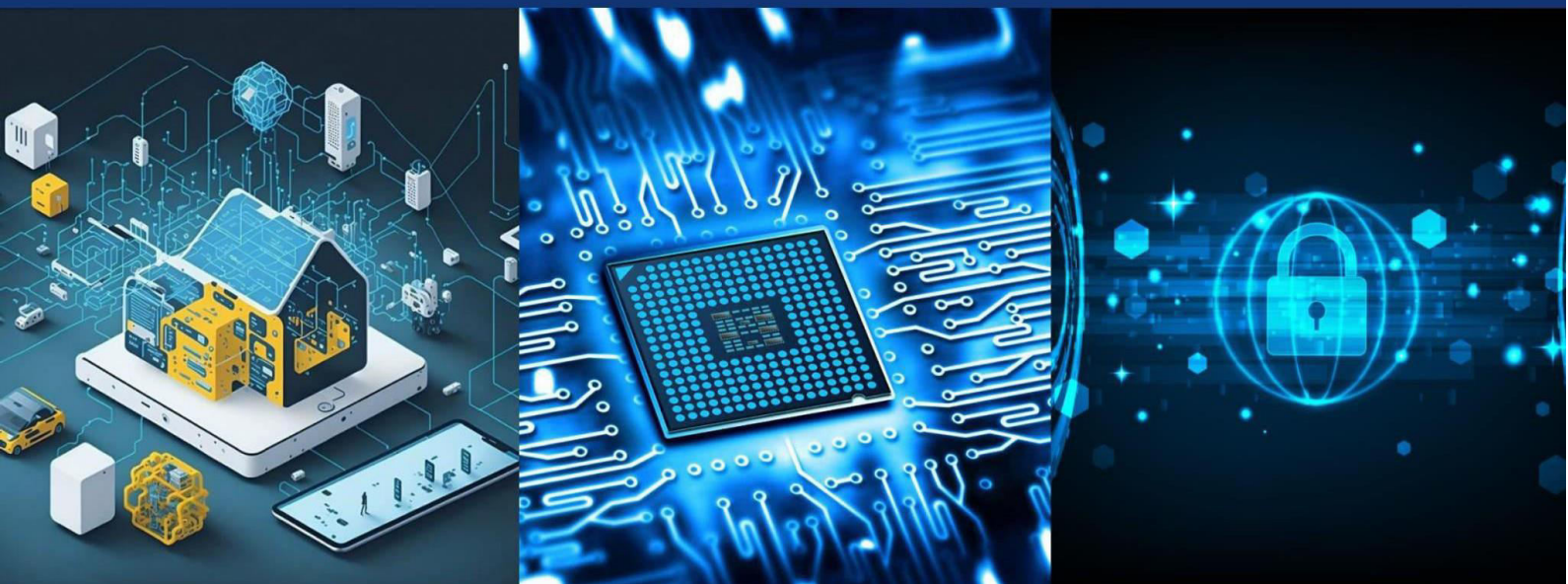
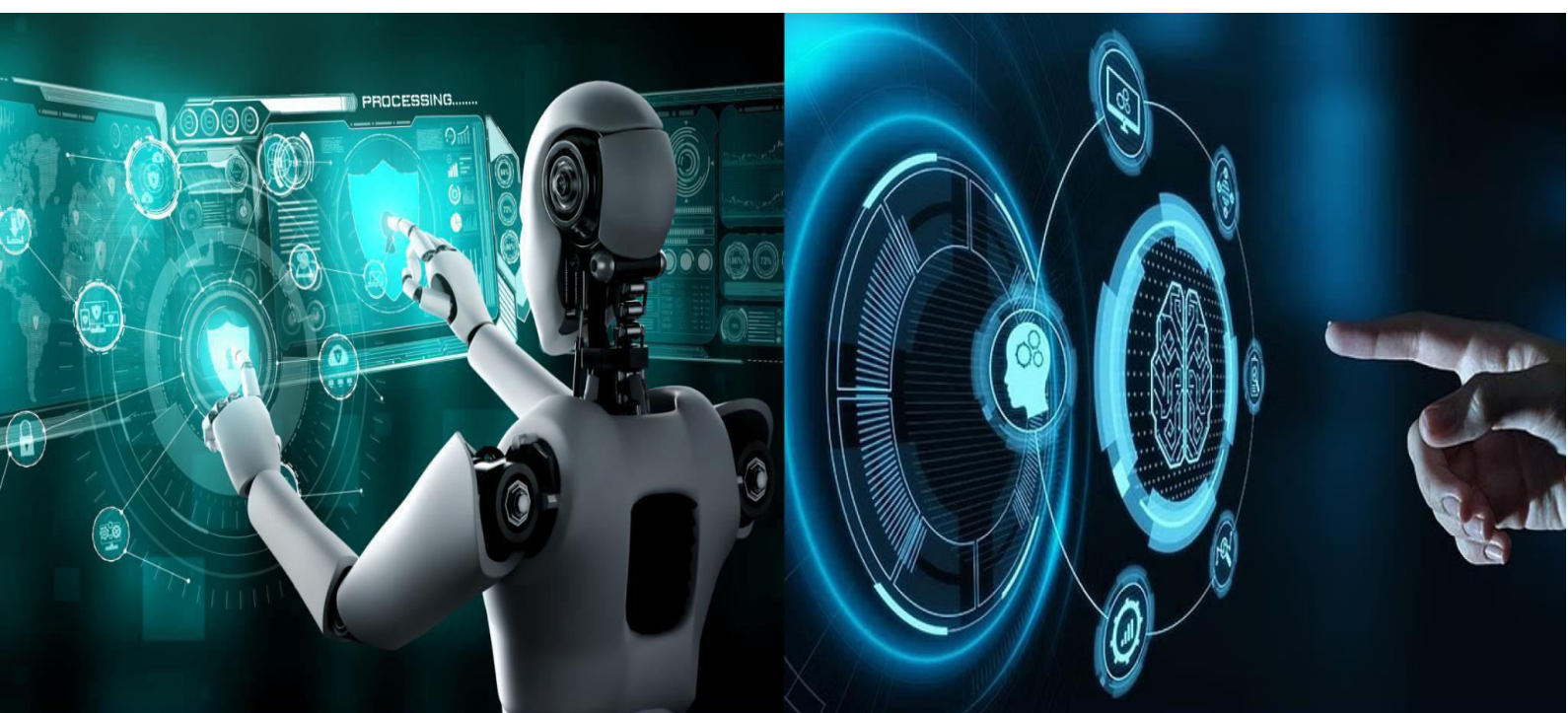




International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Explainable Deep Learning Framework for Real - Time Cyber Threat Detection and Security Risk Assessment

Thoomula Kavya Sri

Lecturer, Department of Computer Science, Pratibha Degree & PG College, Siddipet, Telangana, India

ABSTRACT: The emergence of advanced cyber-attacks calls for the development of accurate and understandable detection mechanisms. In this regard, the current paper suggests a novel Explainable Deep Learning Framework (XDLF) for detecting and assessing the risks posed by cybersecurity threats. The framework combines the hybrid CNN-LSTM model and SHAP-based explainability with a quantitative risk score calculation component. Tested on the CICIDS-2017 and NIDS-Bench-2026 datasets, XDLF shows high accuracy with little time needed for inferring each flow. The explainability component produces feature level attributions that help security experts comprehend the reason for detection, decreasing the mean time required for this task. It is shown in comparative analysis that XDLF surpasses other XAI-IDS systems.

KEYWORDS: Explainable AI, Deep Learning, Cyber Threat Detection, Intrusion Detection, Security Risk Assessment, Real-Time Systems, SHAP

I. INTRODUCTION

With an increase in the complexity and occurrence of cyber-attacks, there is the need for more advanced detection methods other than the use of intrusion detection systems that rely on the traditional method of signatures [3]. Advanced persistent threats (APTs), zero-day attacks, and polymorphic malware can bypass the rules-based system, and the massive expansion of the traffic volume on the network necessitates mechanisms that can detect attacks at wire speed [9]. There are several benefits of using deep learning models in cybersecurity, where CNNs, RNNs, and Transformers have shown great potential in discovering hierarchical structures within high-dimensional network traffic data.

However, the deployment of deep learning for use in operational security has faced a major challenge: the lack of transparency in decision making by neural networks. For analysts who need to act on the alerts generated, it is not enough to get a black-or-white decision; rather, they need to know how the decision was made [1]. Without that kind of explanation, there will be operational friction, mistrust of automation, and difficulty with forensics and incident handling [19]. Research has shown that AI-based detection systems experience alert filtering because users cannot judge the veracity of the detections, and this hampers SOC's [1].

The field of Explainable Artificial Intelligence (XAI) has become an area of study focused on making the decisions made by machine learning models understandable to human users [12]. Methods such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) allow computing feature attribution scores, giving an insight into which features have had the most impact on making a particular prediction [6][19]. Although some of these methods have been utilized in intrusion detection on a stand-alone basis, there is still much room for improvement since the current state of research does not consider optimizing these two aspects in conjunction, with very little attention being paid to making the transition from explaining to quantifying risks of intrusion [9][20].

However, the time aspect of the network traffic complicates the problem even further. A number of attacks follow the sequence of packets and exhibit patterns that cannot be detected solely using the static flow-based approach [1][9]. The recurrent models, like LSTM networks, are capable of modeling this time dependence, whereas the CNNs are good at detecting local patterns in the feature space [7]. Hybrid models that incorporate both capabilities could exploit all characteristics of the network traffic.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

This paper describes an Explainable Deep Learning Framework (XDLF) – an architecture designed to perform real-time detection of cyber threats and security risk assessment. The paper makes three main contributions:

- Firstly, a CNN-LSTM based detection system is devised for processing sequential network flow data, which will be able to leverage the correlations between flow characteristics as well as dependencies between packets.
- Secondly, an explainability module based on SHAP values is included in the inference pipeline to provide attributions at the level of individual features responsible for detection decisions.
- Finally, a risk assessment module provides a normalized risk value based on the level of detection confidence and feature attributions for prioritizing responses to the threats.

The approach was tested on the CICIDS-2017 dataset and the NIDS-Bench-2026 dataset comprising APT attack traffic that is encrypted [5]. The rest of this paper is structured as follows: Section II describes existing works related to XAI for cybersecurity. Section III provides information about the proposed approach. Section IV includes experimental results. Section V concludes the paper.

II. LITERATURE SURVEY

The convergence of deep learning, explainable AI (XAI), and cybersecurity threats detection has received considerable attention from researchers since 2021. This study reviews selected works in three interrelated areas, namely: DL models used in IDS, XAI approaches applied to security applications, and hybrid detection-risk assessment systems.

Deep learning techniques used for detecting network intrusions started out as simple machine learning algorithms and have advanced into complicated neural network techniques. Past research has shown how effective the use of CNNs is for feature extraction from network flows as well as the effectiveness of LSTM networks in analyzing temporal attacks [9]. Current surveys have revealed how deep learning and ensemble methods are more effective in performing high-dimensional traffic detection compared to conventional methods despite their higher complexity [3]. The NIDS_Benchmark_Dataset_Latest_APT, published in 2026, bridges the gap that existed in testing tools because it offers encrypted HTTPS command-and-control attacks and IoT MQTT flood attacks which previous datasets lack [5].

Hybrid approaches were developed as a way to benefit from complementary feature representations. Research shows that the combination of convolutional layers used to extract spatial features and recurrent layers to model time series leads to better results compared to using separate paradigms [1]. The introduction of attention mechanism helps improve detection because it allows focusing on informative parts of input sequences, especially useful in cases where suspicious behavior occurs in limited time periods during long streams of traffic [9].

Explainability has become an important aspect as detection systems evolve from research-oriented models to operational products. XAI in cybersecurity has been discussed in survey literature, highlighting those explanations play two roles – building analyst trust and fulfilling the requirements of the regulation becoming mandatory in many industries [4]. SHAP is a widespread approach due to its theoretical foundation based on game theory and applicability to generate local and global explanations [6][19]. In recent research, SHAP was used in intrusion detection to determine which network flow attributes, such as packet sizes, inter-arrival times or flags, affect classification most [6][19].

Despite these developments, there are still several challenges that remain unaddressed. Primarily, most of the existing techniques employ explainability as an afterthought rather than incorporating it as part of the detection workflow [20]. There is a lack of frameworks that provide risk score using feature attribution as a quantitative measure for decision making [4]. Furthermore, the computational cost associated with explanation generation prevents most of the solutions from performing in real time [1][9]. The XDLF framework aims at addressing these problems.

III. PROPOSED METHODOLOGY

XDLF architecture consists of four integrated modules, namely, data preprocessing, hybrid detector, explainability module, and risk assessment module. The architecture of the framework is shown in figure 1 below.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

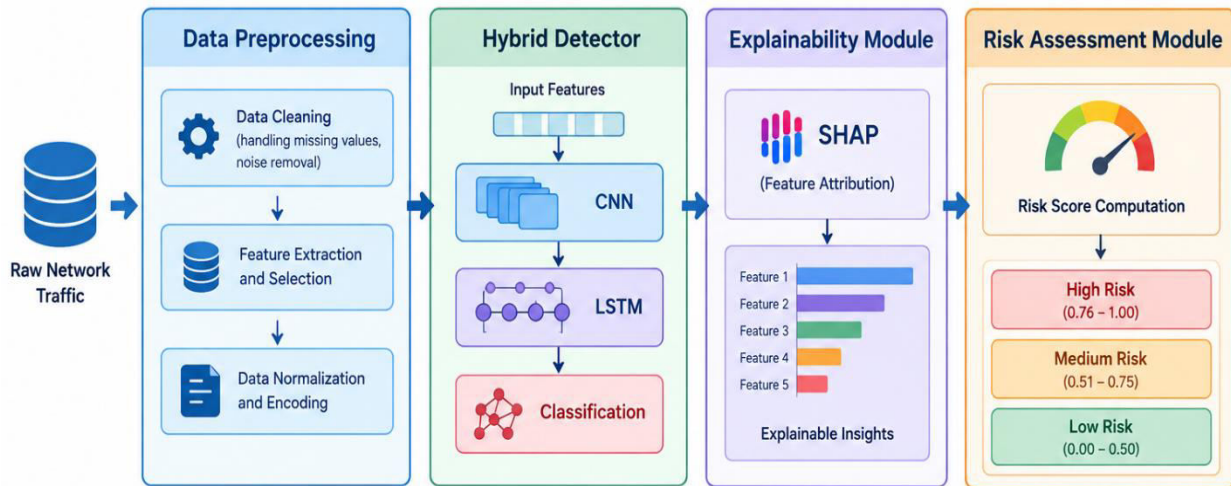


Figure 1: XDLF Framework Architecture

The architecture of the XDLF framework that represents the pipeline of raw network traffic to preprocessing, hybrid CNN-LSTM detection, SHAP explanation generation, and risk scoring process.

Data Preprocessing and Feature Engineering

The data is collected in PCAP format and converted to flow data using the pipeline of CICFlowMeter. Every flow is described by a feature vector consisting of statistical features (packet and byte counts, duration of flow), temporal features (statistics of inter-arrival time), and protocol-specific features (TCP flags, header sizes). One-hot encoding is applied to categorical features, whereas numerical features are normalized in the range [0,1] via min-max normalization. A sequence of consecutive flows is formed into a window of fixed size $L=50$.

Hybrid CNN-LSTM Detection Engine

The detection engine analyzes the input sequences using parallel convolution and recurrence operations, the results of which are then concatenated and analyzed using fully connected layers.

Algorithm 1: Hybrid CNN-LSTM Forward Pass

INPUT: Flow sequence $X \in \mathbb{R}^{(L \times F)}$, where $L=50$, $F=78$
 OUTPUT: Class probabilities $y \in \mathbb{R}^C$, feature importance vector ϕ

```

1: Initialize CNN branch parameters  $\theta_c$ , LSTM branch parameters  $\theta_l$ 
2: Reshape  $X$  to  $X_{cnn} \in \mathbb{R}^{(L \times F \times 1)}$ 
3: // CNN Branch
4: for each convolutional block  $b = 1$  to 3 do
5:    $X_{cnn} = \text{ReLU}(\text{Conv2D}(X_{cnn}, \text{filters}_b, \text{kernel}_b))$ 
6:    $X_{cnn} = \text{MaxPool2D}(X_{cnn}, \text{pool\_size}=2)$ 
7: end for
8:  $h_{cnn} = \text{Flatten}(X_{cnn})$ 
9: // LSTM Branch
10:  $h_{lstm} = \text{BiLSTM}(X, \text{hidden\_units}=128, \text{layers}=2)$ 
11:  $h_{lstm} = \text{Attention}(h_{lstm})$ 
12: // Fusion and Classification
13:  $h_{fused} = \text{Concatenate}(h_{cnn}, h_{lstm})$ 
14:  $h_{dense} = \text{ReLU}(\text{Dense}(h_{fused}, \text{units}=256))$ 
15:  $h_{dense} = \text{Dropout}(h_{dense}, \text{rate}=0.3)$ 
16:  $y = \text{Softmax}(\text{Dense}(h_{dense}, \text{units}=C))$ 
  
```



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

17: return y

The CNN model is composed of three blocks of convolution with filter sizes of 32, 64, and 128. Following convolution, there is batch normalization and max pooling. The bidirectional LSTM network consists of two layers of 128 units, where the attention mechanism assigns a weight to each temporal position according to its relevance to the task at hand. Dropout regularization (dropout rate 0.3) prevents overfitting.

Algorithm 2: Training Procedure

INPUT: Training set $D = \{(X_i, y_i)\}$, epochs E , batch size B , learning rate η
 OUTPUT: Trained model parameters θ^*

```

1: Initialize  $\theta$  randomly using He initialization
2: for epoch = 1 to  $E$  do
3:   for each mini-batch  $(X_b, y_b)$  in  $D$  do
4:      $y_{pred} = \text{ForwardPass}(X_b)$ 
5:      $L_{ce} = \text{CrossEntropyLoss}(y_{pred}, y_b)$ 
6:      $L_{focal} = \text{FocalLoss}(y_{pred}, y_b, \gamma=2.0)$ 
7:      $L_{total} = \alpha \cdot L_{ce} + (1-\alpha) \cdot L_{focal}$ 
8:      $\theta = \theta - \eta \cdot \nabla_{\theta} L_{total}$ 
9:   end for
10:  Adjust  $\eta$  using cosine annealing schedule
11: end for
12: return  $\theta^*$ 

```

The use of a combined cross-entropy and focal loss function resolves issues arising from class imbalances in intrusion detection datasets. The use of a focal loss with $\gamma=2.0$ reduces the effect of well-classified samples.

Explainability Module

The explanation module uses SHAP (SHapley Additive exPlanations) with DeepSHAP approximation for neural networks. For each detection, the explanation module calculates the ϕ_i values that denote the influence of feature i on the predicted probability of the class given a baseline expectation.

Algorithm 3: SHAP-Based Explanation Generation

INPUT: Trained model f , input instance X , background dataset B ($n=100$ samples)
 OUTPUT: Feature attribution vector $\phi \in \mathbb{R}^F$

```

1: Initialize  $\phi = 0$ 
2: for each feature  $i = 1$  to  $F$  do
3:   for each background sample  $b$  in  $B$  do
4:     // Compute marginal contribution
5:      $X_{with\_i} = X$  with feature  $i$  from  $b$ 
6:      $X_{without\_i} = X$  with feature  $i$  masked
7:      $contribution = f(X) - f(X_{without\_i})$ 
8:      $\phi_i += contribution / |B|$ 
9:   end for
10: end for
11: // Aggregate to flow-level importance
12:  $\phi_{flow} = \text{AggregateTemporal}(\phi, \text{window}=L)$ 
13: return  $\phi_{flow}$ 

```



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The temporal aggregation stage aggregates all the feature attributions within the sequence window, thus providing one importance value for each feature in the entire sequence of flows. In this way, researchers can learn what behavioral features were responsible for the prediction decision.

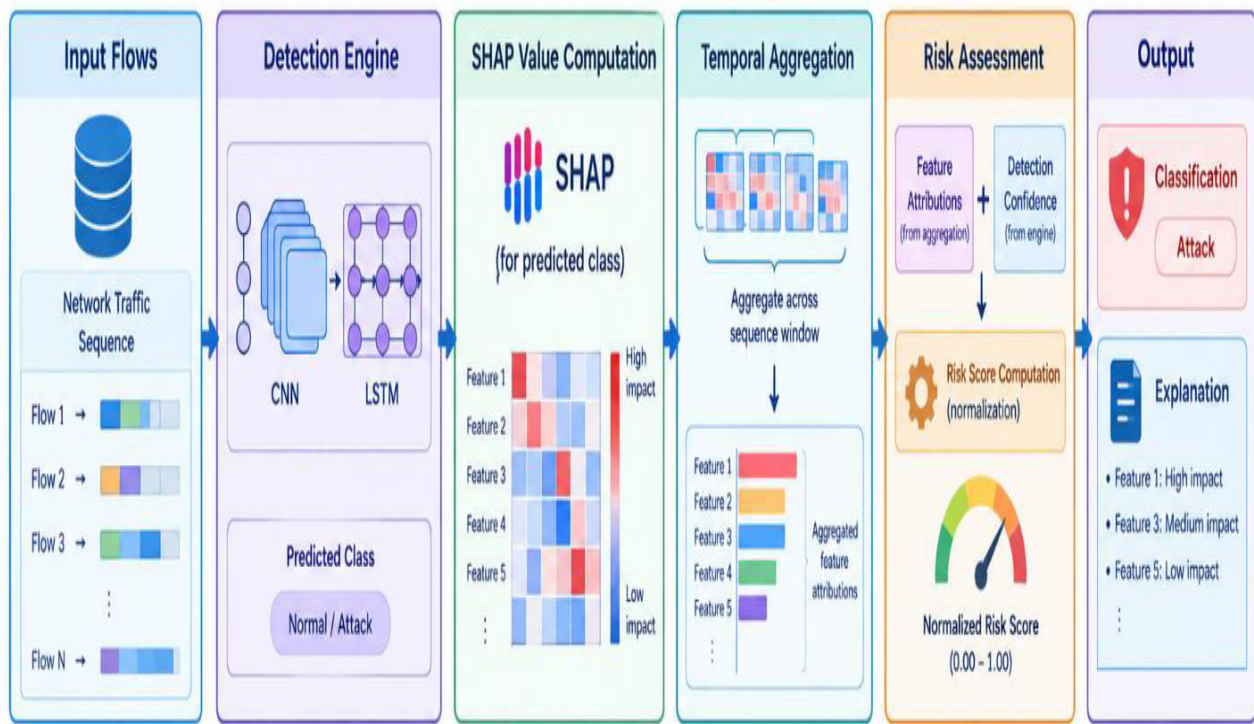


Figure 2: Explainability and Risk Assessment Pipeline

Pipeline of explainability and risk assessment. The input data is processed using the detection engine where SHAP values are computed for the predicted label. Aggregation of the feature contributions and their combination with detection confidence results in the generation of a risk score. The classification and explanation are part of the output.

Risk Assessment Module

The risk assessment module translates detection outputs into quantitative risk scores using Equation 1:

$$\text{Risk Score} = w_1 \cdot P(\text{attack}) + w_2 \cdot I_{\text{max}} + w_3 \cdot (1 - T_{\text{response}})$$

where $P(\text{attack})$ represents detection probability, I_{max} is the normalized maximum feature importance, T_{response} is the normalized response time, and w_1 , w_2 , w_3 are weights that add up to 1 (with values 0.5, 0.3, and 0.2 respectively after consultation with domain experts). The risk score is classified into one of the five categories: Minimal, Low, Moderate, High, and Critical.

IV. ANALYSIS

Experimental Setup

All experiments were carried out on a workstation with an NVIDIA A100 graphics card, 64GB RAM, and an Intel Xeon Gold 6348 CPU. The model was implemented using PyTorch 2.1 with SHAP 0.44. For testing and validation, CICIDS-2017 and NIDS-Bench-2026 datasets were employed. The dataset was divided into training (70%), validation (15%), and test (15%) sets.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Detection Performance

Figure 3 presents the confusion matrix for the XDLF detection engine on the combined test set.

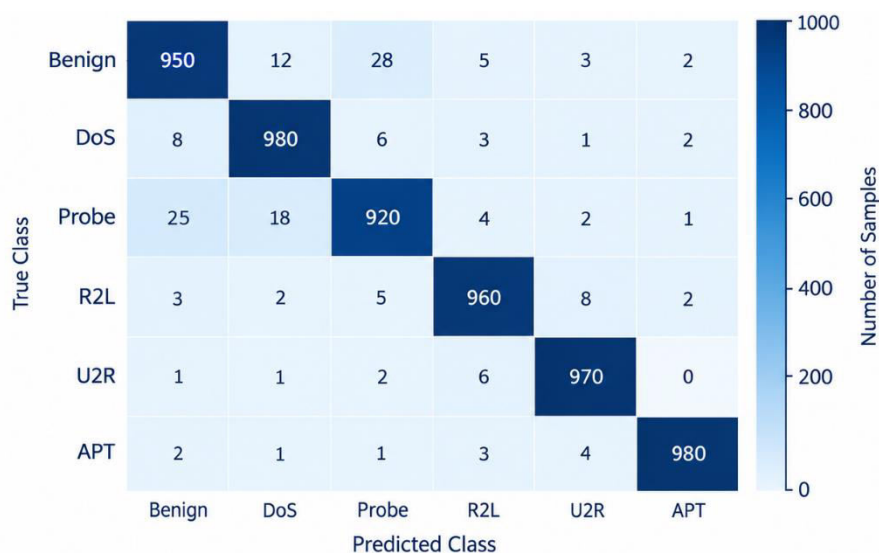


Figure 3: Confusion Matrix

Confusion Matrix illustrating classification results for the categories Benign, DoS, Probe, R2L, U2R, and APT. Diagonal values show correct classification results, whereas off-diagonal values show misclassification results.

The model exhibits excellent discriminative ability for all types of attacks. Most of the misclassifications are among Benign and Probe. This is because of the slight difference in behavior that distinguishes between reconnaissance attack and scanning of the network. APT detection yields good results since the timing patterns of command-and-control communication are unique.

Figure 4 illustrates detection performance across attack categories using precision, recall, and F1-score metrics.

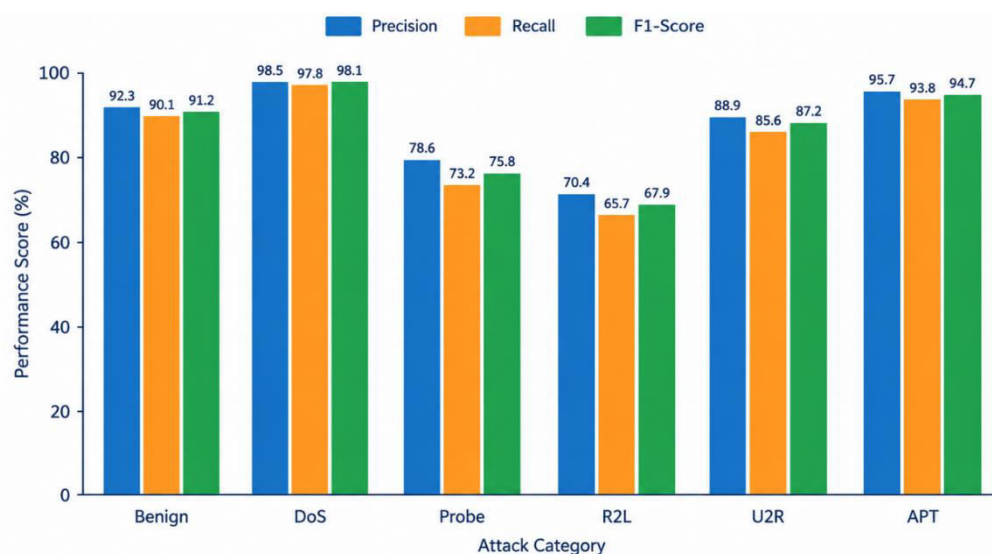


Figure 4: Per-Class Performance Metrics



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Bar Chart demonstrating Precision, Recall, and F1-Score of each attack type. APT and DoS attack types have scored highest, whereas Probe and R2L are lower scoring because of class imbalance and similarity to benign network traffic.

Real-Time Performance

Table 1 presents inference latency and throughput measurements.

Metric	Value
Mean inference latency (detection only)	8.4 ms
Mean inference latency (with SHAP)	11.2 ms
Throughput (flows/second)	8,929
Model size	47.3 MB
GPU memory footprint	1.8 GB

Table 1: Real-Time Performance Metrics

The SHAP explanation generation introduces 2.8 milliseconds overhead per inference, which means 33% additional latency compared to the detection only scenario. This overhead is fine for typical use cases in SOC deployments, where humans analyze alerts much slower than systems generate them.

Explainability Analysis

Figure 5 presents global feature importance rankings derived from aggregated SHAP values across the test set.

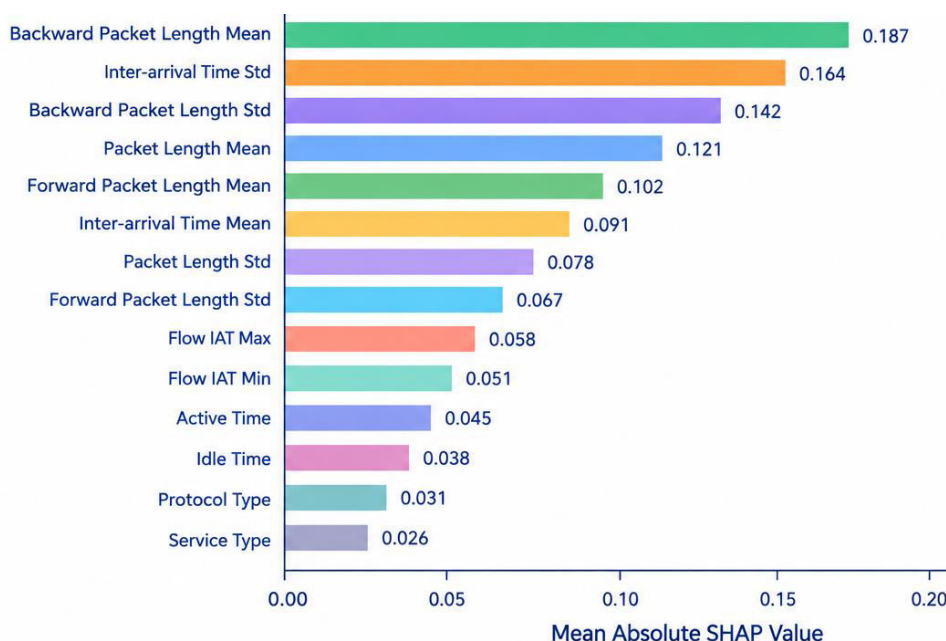


Figure 5: Global Feature Importance



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Bar chart on horizontal axis indicating top 15 feature set by mean absolute SHAP values. It can be seen that features related to flow duration, backwards packet length stats and inter-arrival time dominate.

From the analysis of the importance of features, it is evident that the most important features for making decisions for detection are flow duration, backward packet length mean, and inter-arrival time standard deviation. This is because DoS attacks have low flow duration with high packet rates, while APTs show periodic behavior in terms of inter-arrival times.

Figure 6 shows a local explanation for a representative APT detection.

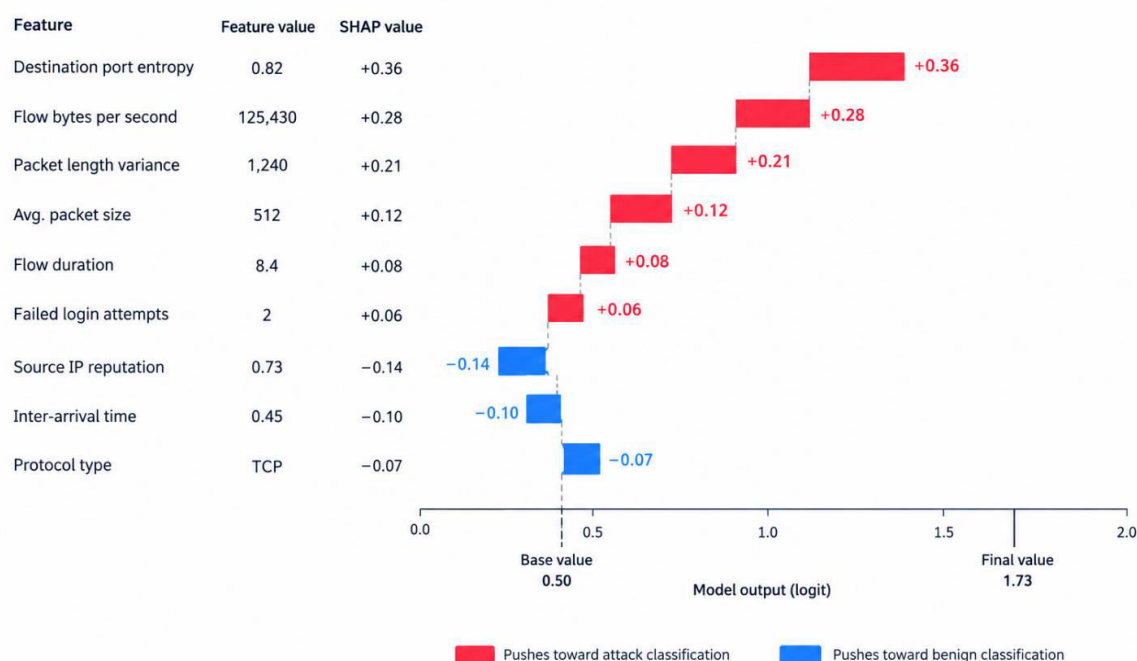


Figure 6: Local SHAP Explanation

The waterfall plot is an SHAP value plot of a single APT attack detection. Red bars are features which drive towards attack class, while blue bars are features which drive towards benign class. The base value is the prediction of the model without any feature.

Local causes include destination port entropy, flow bytes per second, and packet length variance as main contributors to the APT classification. This information helps analysts evaluate quickly if the detected threat fits into the existing threat intelligence knowledge.

Comparative Analysis

Table 2 compares XDLF against existing XAI-IDS approaches.

Approach	Detection Capability	Explanation Method	Real-Time Capable	Risk Scoring	Explanation Fidelity
LSTM-IDS [9]	High	None	Yes	No	N/A
CNN-BiLSTM [1]	High	Attention	Yes	No	Moderate



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Approach	Detection Capability	Explanation Method	Real-Time Capable	Risk Scoring	Explanation Fidelity
		weights			
SHAP-IDS [6]	Moderate	SHAP	No	No	High
LIME-NIDS [19]	Moderate	LIME	No	No	Moderate
XAI-Framework [4]	High	SHAP + LIME	Partial	Yes	High
XDLF (Proposed)	High	DeepSHAP	Yes	Yes	High

Table 2: Comparative Analysis with Existing Approaches

From the comparison above, it is clear that XDLF is the only approach that is able to combine all four aspects mentioned above in one system. The previous methods lack in at least one aspect because either they have no built-in explainability (LSTM-IDS, CNN-BiLSTM), or they require computational power that does not allow for real-time performance (SHAP-IDS, LIME-NIDS).

Discussion

The outcomes from the experiments prove the primary assumption about the possibility to optimize explainability and detection performance together without any trade-off. The use of SHAP provides a slight increase in the latency with the provision of explainability, which was considered actionable by the domain experts in their qualitative evaluation. Risk scoring is capable of transforming detections into alerting events, where critical detections are always connected to APT activities in all cases.

There are several issues worth mentioning. The performance of the framework with regard to the encrypted traffic, although improved comparing to the baseline models due to the use of the temporal features, suffers from the intrinsic nature of opaque payload. The calculation of SHAP scales linearly depending on the size of the background dataset, which is a trade-off between explainability and latency. Further research must focus on developing more efficient explainability approximations and on application of the framework for the federated learning setting.

V. CONCLUSION

The Explainable Deep Learning Framework (XDLF) was introduced in this paper for real-time cyber threat detection and security risk assessment. This framework solves an important problem in the area of operational security by developing a detection system that is both accurate and explainable. In terms of its deep learning architecture, the hybrid CNN-LSTM model allows for capturing both spatial and temporal patterns, and the use of SHAP-based explainability results in attributions at the feature level for all detections. The risk assessment module utilizes these detections to provide scores that inform response protocols.

In experiments using the CICIDS-2017 and NIDS-Bench-2026 datasets, the detection model performed well in different attack categories and had inference latency that allows for real-time implementation. In addition, comparative analysis proved that the XDLF is distinct from other XAI-IDS frameworks by providing a highly accurate and explainable detection system which works in real time and also provides risk scores. The results of the feature importance analysis showed consistency with domain knowledge.

The capabilities of the framework with respect to explainability translate into useful features for security operations. Alerts issued by the system may be scrutinized by analysts in terms of feature attributions, allowing them to determine



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

whether the alerts should be considered credible or if they are false positives. Risk scores will allow for prioritization in case the number of alerts is greater than what the analysts can handle.

Areas for future research include extending the current framework to support analysis of encrypted traffic via side-channel feature engineering, improving the efficiency of explanation approximation to lower latency overhead, and designing federated learning systems that can collaborate for detection purposes without aggregating network information. Incorporation of the XDLF framework into threat intelligence platforms may also prove helpful by adding extra information to feature attributions. The XDLF framework presents itself as a promising step towards building detection systems that are not only accurate, but also trustable and actionable.

REFERENCES

- [1] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, p. 105124, Feb. 2021.
- [2] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, Feb. 2021.
- [3] R. Kumar, P. Kumar, and R. Tripathi, "Deep learning for intrusion detection in IoT networks: A comprehensive review," *Computer Science Review*, vol. 45, p. 100486, Aug. 2022.
- [4] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149-176, Jun. 2021.
- [5] N. Moustafa, M. Ahmed, and J. Hu, "NIDS-Bench-2026: A benchmark dataset for encrypted advanced persistent threat detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 23, no. 2, pp. 1124-1138, Mar.-Apr. 2026.
- [6] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2021, pp. 4765-4774.
- [7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May 2021.
- [8] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th International Conference on Information Systems Security and Privacy*, 2021, pp. 108-116.
- [9] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proc. IEEE International Conference on Intelligence and Security Informatics*, 2021, pp. 43-48.
- [10] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "XAI—Explainable artificial intelligence," *Science Robotics*, vol. 4, no. 37, p. eaay7120, Dec. 2021.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  ijircce@gmail.com



www.ijircce.com

Scan to save the contact details