

Research Article

Explainable Sequence-Aware Deep Learning Framework for Potential Zero-Day Attack Detection and Cross-Domain Generalization in Enterprise Network Intrusion Detection

Uchechukwu Samuel Nwankwo¹ , Obi Chukwuemeka Nwokonkwo¹ ,
Charles Ikerionwu² , Udoka Felista Eze¹ ,
Adetokunbo MacGregor John-Otumu^{1,*} 

¹Department of Information Technology, Federal University of Technology, Owerri, Nigeria

²Department of Software Engineering, Federal University of Technology, Owerri, Nigeria

Abstract

Zero-day attacks remain a significant challenge in enterprise network security because their previously unseen characteristics can reduce the effectiveness of conventional signature-based intrusion detection systems. Although machine learning and deep learning have improved intrusion detection, many existing approaches are evaluated within a single dataset and often treat network traffic records as independent observations, providing limited evidence of temporal behavior and cross-domain generalization. This study proposes an Explainable Sequence-Aware Deep Learning Framework for Potential Zero-Day Attack Detection and Cross-Domain Generalization in Enterprise Network Intrusion Detection. The framework represents network traffic as overlapping sequences of 20 consecutive network-flow records and combines a one-dimensional convolutional neural network (1D-CNN), Bidirectional Long Short-Term Memory (Bi-LSTM), and four-head Multi-Head Self-Attention to learn local traffic characteristics, temporal dependencies, and informative relationships within network behavior. A shared representation supports both binary intrusion detection and multiclass attack classification, while SHAP and LIME provide global and local explanations of model decisions. CICIDS2017 serves as the source domain for model development, whereas UNSW-NB15 is maintained as an independent target domain for cross-domain evaluation. A stratified sample of 50,000 records is independently selected from each dataset, with SMOTE applied only to the CICIDS2017 training data. On the CICIDS2017 internal test set, the framework achieved 96.89% accuracy, 97.45% precision, 89.65% recall, 93.38% F1-score, 0.9961 ROC-AUC, and 0.9379 MCC for binary detection, while multiclass classification achieved 97.0% accuracy and 96.8% F1-score. On the independent UNSW-NB15 test set, binary detection achieved 87.70% accuracy and 90.56% F1-score, while multiclass detection achieved 80.35% accuracy and 59.57% F1-score. The findings demonstrate strong in-domain learning and useful cross-domain detection capability without retraining or fine-tuning. The cross-domain results also revealed the difficulty of transferring learned representations across different network environments. In this study, cross-domain evaluation is used to assess potential zero-day detection capability rather than to claim detection of a specifically verified zero-day attack.

*Correspondence: Adetokunbo MacGregor John-Otumu (adetokunbo.johnotumu@futo.edu.ng)

Received: 9 September 2026; **Accepted:** 21 September 2026; **Published:** 30 September 2026



Copyright: © The Author(s), 2026. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

Keywords

Zero-Day Attack Detection, Intrusion Detection System, Deep Learning, Temporal Network Traffic, Cross-Domain Generalization, Explainable Artificial Intelligence

1. Introduction

The rapid growth of interconnected digital technologies has changed the way organizations operate, communicate, and deliver services. Modern enterprise networks increasingly depend on cloud platforms, virtualized infrastructures, Internet of Things devices, distributed applications, and other connected services. Although these technologies improve operational efficiency and accessibility, they also increase the complexity of enterprise networks and create more opportunities for cyberattacks. Protecting network resources, information assets, and critical services has therefore become an important cybersecurity requirement.

Among the threats facing enterprise networks, zero-day attacks remain particularly difficult to detect because they exploit vulnerabilities that are not yet known or adequately addressed by conventional security mechanisms. Signature-based intrusion detection systems are effective when the characteristics of an attack are already known and incorporated into detection rules. However, their dependence on predefined signatures limits their ability to recognize previously unseen attack behaviors. The problem becomes more serious when attackers exploit new vulnerabilities before appropriate signatures or defensive measures are available. Existing studies have therefore continued to emphasize the need for intelligent approaches capable of identifying previously unseen or evolving attack patterns [1, 2].

Machine learning (ML) and deep learning (DL) have emerged as important alternatives to conventional signature-based detection because they can learn patterns from network traffic without depending entirely on predefined attack signatures. These approaches have been applied to intrusion and zero-day attack detection using ensemble learning, convolutional neural networks, recurrent neural networks, long short-term memory networks, autoencoders, and hybrid architectures [1-5]. However, strong performance within a particular benchmark dataset does not necessarily indicate that the learned model will maintain similar performance when applied to traffic from a different network environment.

This issue makes cross-domain generalization an important consideration in enterprise intrusion detection. Many existing studies are evaluated using a single dataset or within the same data environment used for model development. Consequently, the resulting performance may reflect the ability of the model to learn the characteristics of a particular dataset rather than its ability to recognize attack behavior in a different network

environment. Studies reviewed in the study repeatedly identify dependence on benchmark datasets, limited transferability, and inadequate cross-dataset validation as continuing challenges in zero-day attack detection [2-6].

A further limitation is that network traffic is frequently treated as independent observations. Such treatment can overlook the behavioral relationships that develop across successive network activities. Zero-day attacks and other sophisticated threats may evolve through a series of related activities rather than appearing as a single isolated event. Temporal traffic analysis therefore provides an opportunity to examine how network behavior changes across consecutive observations and to identify patterns that may not be evident from individual traffic records. The study identifies temporal traffic pattern analysis as particularly relevant to the detection and forecasting of emerging threats [7, 8].

In response to this limitation, the present study treats enterprise network traffic as a sequence of related observations rather than as isolated records. A sliding-window mechanism is used to construct overlapping sequences of 20 consecutive network-flow records. This sequence representation provides the proposed deep learning model with temporal context and allows it to learn relationships across successive network activities. The sequence-aware design is followed by a hybrid architecture comprising a One-Dimensional Convolutional Neural Network (1D-CNN), Bidirectional Long Short-Term Memory (Bi-LSTM), and Multi-Head Attention. The CNN learns local traffic representations, the Bi-LSTM captures temporal dependencies, while the attention mechanism emphasizes informative representations within the learned sequence. The resulting shared representation is then supplied to binary detection and multi-class classification heads.

Another concern in existing deep learning-based intrusion detection is the limited transparency of model decisions. High-performing models can be difficult for security analysts to understand when the reasons behind their predictions are unclear. This study therefore incorporates Explainable Artificial Intelligence (XAI), using SHAP and LIME to provide feature-level explanations of model decisions. Existing studies have similarly emphasized the importance of explainability for improving transparency, trust, and operational acceptance of AI-based cybersecurity systems [9, 10].

The need for a more comprehensive approach is further supported by the limitations identified in existing research. Recent studies have explored predictive zero-day detection,

attention-based architectures, federated deep learning, explainable intrusion detection, and hybrid cybersecurity models. However, the study comparative analysis shows that these approaches generally address only some of the required capabilities. For example, some provide detection and classification, others incorporate explainability or forecasting, while cross-domain evaluation and integrated mitigation support remain less common.

Against this background, this paper proposes an explainable sequence-aware deep learning framework for potential zero-day attack detection and cross-domain generalization in enterprise network intrusion detection. The framework combines temporal sequence generation with a hybrid 1D-CNN, Bi-LSTM, and Multi-Head Attention architecture to learn both local traffic characteristics and temporal behavioral dependencies. The framework is developed using CICIDS2017 and evaluated independently across UNSW-NB15 to examine whether the learned representations transfer to a different network environment. Importantly, the cross-domain experiment does not retrain or fine-tune the model on UNSW-NB15, thereby providing a direct assessment of transferability.

In the context of this study, potential zero-day detection refers to the ability of a model developed in one network environment to identify potentially malicious traffic in an independent network environment without retraining or fine-tuning. This cross-domain setting provides an evaluation of transferability under previously unseen traffic conditions. However, it should not be interpreted as direct validation against a confirmed zero-day vulnerability or attack.

The main contribution of the paper is therefore not simply another high-performing intrusion classifier. Rather, it investigates whether sequence-aware temporal representation learning can provide useful detection capability while exposing the difficult problem of cross-domain generalization.

2. Related Works

2.1. Zero-Day Attack Detection Using Machine Learning

Machine learning (ML) remains an important approach to detecting previously unseen cyber threats. Traditional ML techniques such as Random Forest, Support Vector Machine, Decision Tree, k-nearest neighbor, and ensemble models have been widely applied to intrusion and anomaly detection. Their effectiveness is largely associated with their ability to identify relationships within network traffic features, although their performance can be affected by feature representation, class imbalance, dataset characteristics, and the ability of the learned model to generalize beyond the training environment [2, 3].

Sarhan *et al.* [1] investigated zero-day attack detection using a zero-shot machine learning approach, demonstrating the

potential of learning strategies that attempt to recognize attacks not directly represented during training. Dai *et al.* [2] similarly investigated zero-day attack detection on unseen data and demonstrated the importance of evaluating models beyond conventional in-sample classification. These studies are relevant to the present research because they reinforce the distinction between achieving high performance on observed data and demonstrating useful behavior when the model encounters previously unseen attack patterns.

Other ML-based studies have focused on improving anomaly detection through feature selection and optimization. The study reports that such approaches can achieve strong performance, but many remain dependent on benchmark datasets and provide limited evidence of generalization across heterogeneous network environments. This limitation is important because enterprise networks can differ considerably in traffic composition, application behavior, attack distribution, and operational conditions.

2.2. Deep Learning for Zero-Day and Intrusion Detection

Deep learning has gained increasing attention because it can learn hierarchical representations directly from complex network traffic. CNNs are useful for extracting local patterns, whereas recurrent architectures such as LSTM can model relationships across sequences. Hairab *et al.* [3], for example, investigated CNN-based zero-day anomaly detection and demonstrated the relevance of convolutional feature learning to previously unseen attack behavior. The study similarly identifies CNNs, RNNs, LSTMs, autoencoders, and hybrid architectures as important directions in zero-day detection research.

Temporal deep learning is particularly relevant where network behavior changes over time.

2.3. Sequence-Aware, Bi-LSTM and Attention-Based Detection

The use of sequential representations is particularly important for the present research. Instead of treating every network flow as an independent observation, sequence-based methods preserve information about the order and relationship between successive traffic events. This provides a basis for recognizing behavioral evolution that may be missed when individual observations are examined separately.

Krishnan *et al.* [11] proposed an attention-fusion model involving Multi-Head Attention and Bi-LSTM for zero-day attack detection in IoT networks. The study reported strong binary and multi-class classification performance and also incorporated LIME and SHAP for interpretability. However, the study notes concerns relating to class imbalance, possible overfitting, underrepresentation of rare attacks, and the need for broader validation.

The relevance of this work to the present study is clear.

Both approaches recognize the value of combining recurrent temporal learning with attention. However, the present framework extends the sequence-learning perspective by first converting network-flow observations into overlapping fixed-length temporal sequences and then combining 1D-CNN, Bi-LSTM, and Multi-Head Attention within an enterprise-oriented network traffic framework. The study specifies a sequence length of 20 consecutive flow records for this purpose.

2.4. Explainable and Deep Learning Approaches

Explainability has become increasingly important because cybersecurity analysts need to understand why an AI system has classified traffic as suspicious. Sayduzzaman *et al.* [9] investigated explainable zero-day attack detection, while Alparacha *et al.* [10] discussed the use of AI for network threat detection. The study identifies limited transparency as one of the continuing weaknesses of deep learning-based intrusion detection and incorporates SHAP and LIME to address this concern.

Alansary *et al.* [12] further examined emerging AI approaches to zero-day attacks, including machine learning, deep learning, and federated learning. Their work highlights the continuing development of intelligent cybersecurity systems but also reflects the challenges associated with deploying such approaches across changing environments. Similarly, Diana *et al.* [13] reviewed intrusion detection systems and emphasized the practical requirements and challenges surrounding modern cybersecurity deployment.

2.5. Cross-Domain Generalization and Dataset Dependence

Cross-domain generalization represents one of the most important gaps addressed by this study. Many published intrusion detection models are trained and evaluated using the same dataset or closely related data distributions. Such evaluation can produce impressive results while providing limited evidence that the learned representation will remain useful under different network conditions.

This comparative analysis shows that several studies have demonstrated strong detection performance but have not conducted independent cross-domain evaluation. Armijos and Cuenca [14], for example, used a deep autoencoder approach on UNSW-NB15, while Alansary *et al.* [12] used federated deep learning with UNSW-NB15 but was reported to have limited temporal modelling. Diana *et al.* [13] investigated hybrid deep intrusion detection but did not perform cross-domain evaluation.

Dai *et al.* [2] provide further support for the importance of unseen-data evaluation. More broadly, the study identifies overreliance on benchmark datasets, limited live-traffic validation, lack of realistic zero-day-labelled data, and insufficient cross-dataset generalization testing as recurring research

problems.

The present study addresses this issue directly by maintaining CICIDS2017 and UNSW-NB15 as separate experimental domains. CICIDS2017 is used for model development, while UNSW-NB15 is used for independent cross-domain evaluation. No retraining, fine-tuning, or parameter optimization is performed using the cross-domain data. This design makes it possible to determine whether the temporal representations learned from one network environment can transfer to another.

2.6. Research Gap and Positioning of the Proposed Framework

The reviewed studies show substantial progress in machine learning and deep learning for intrusion and potential zero-day attack detection. However, several limitations remain. Many approaches process traffic as individual observations rather than explicitly modelling sequential behavior. Although CNN, LSTM, Bi-LSTM, Transformer, and attention-based models have reported strong results, their transferability across different network environments remains less frequently examined. Explainability is also not consistently integrated into high-performing intrusion detection models. These limitations motivate the present study's focus on sequence-aware learning, explainability, and independent cross-domain evaluation.

The proposed framework is positioned to address these gaps through a sequence-aware temporal learning strategy. Its methodological pipeline begins with data preprocessing and feature harmonization, followed by sliding-window temporal sequence generation. Each sequence contains 20 consecutive network-flow records. The resulting sequences are processed through 1D-CNN, Bi-LSTM, and Multi-Head Attention layers, after which the shared representation is used for binary intrusion detection and multi-class attack classification.

The principal distinction, therefore, is that the study does not treat high in-domain accuracy as sufficient evidence of model robustness. Instead, it explicitly separates in-domain learning from cross-domain testing. The trained model is applied directly to UNSW-NB15 without retraining, providing an independent assessment of transferability.

3. Materials and Methods

The Materials and Methods section should provide comprehensive details to enable other researchers to replicate the study and further expand upon the published results. If you have multiple methods, consider using subsections with appropriate headings to enhance clarity and organization.

3.1. Materials

This section presents the materials used in the study, including the network intrusion datasets and the computational resources employed for the development and evaluation of the

proposed framework. Particular attention is given to the characteristics and preparation of the datasets used for model development and independent cross domain evaluation.

3.1.1. Dataset Description

This subsection describes the datasets used in the study, focusing on their sources, characteristics, traffic records, feature representations, and attack categories. The CICIDS2017 dataset is used as the source domain for model development, while UNSW-NB15 is used as an independent target domain for evaluating the framework's cross domain generalization capability.

Table 1 presents the description of the CICIDS2017 dataset used as the primary dataset in this study. The dataset was developed by the Canadian Institute for Cybersecurity at the University of New Brunswick, Canada, and is provided in CSV format with 79 features, including the class label. It contains 2,830,819 network flow records distributed across 15 classes, comprising one benign class and 14 attack classes. Of these, 2,273,097 records are benign, while 557,722 represent attack traffic. To manage computational requirements, a stratified sample of 50,000 network flow records was selected for model development, training, validation, and in-domain testing.

Table 1. Description of the CICIDS2017 Dataset.

Item	Description
Dataset Name	CICIDS2017
Developing Institution	Canadian Institute for Cybersecurity (CIC), University of New Brunswick, Canada
Official Dataset Source	https://www.unb.ca/cic/datasets/ids-2017.html
Dataset Download Source	Kaggle Repository (combine.csv)
Dataset Format	CSV
Total Dataset Size	684.7 MB
Number of Features	79 (including the class label)
Total Number of Records	2,830,819
Number of Classes	15 (1 benign class and 14 attack classes)
Benign Records	2,273,097
Attack Records	557,722
Sample Used in this Study	50,000 Network Flow Records
Role in this Research	Primary dataset for model development, training, validation and in-domain testing

Table 2 presents the description of the UNSW-NB15 dataset, which is used as the independent target dataset for cross dataset validation and assessment of the framework's generalization capability. The dataset was developed by the Australian

Centre for Cyber Security at UNSW Canberra and contains 45 features, including the class label, with 257,673 network flow records distributed across 10 classes. A stratified sample of 50,000 network flow records is used in this study for independent cross domain evaluation.

Table 2. Description of the UNSW-NB15 Dataset.

Item	Description
Dataset Name	UNSW-NB15
Developing Institution	Australian Centre for Cyber Security (ACCS), UNSW Canberra, Australia
Official Dataset Source	https://research.unsw.edu.au/projects/unswnb15-dataset
Dataset Download Source	Kaggle Repository

Item	Description
Dataset Format	CSV
Number of Features	45 (including the class label)
Training Dataset Size	32.29 MB
Testing Dataset Size	15.38 MB
Training Records	175,341
Testing Records	82,332
Total Number of Records	257,673
Number of Classes	10
Sample Used in this Study	50,000 Network Flow Records
Role in this Research	Cross-dataset validation and generalization assessment

3.1.2. Justification for Dataset Selection

The CICIDS2017 and UNSW-NB15 datasets were selected because they provide different and complementary network traffic characteristics and are widely used in cybersecurity research.

The CICIDS2017 dataset was used for model development because it contains realistic enterprise network traffic and a variety of attack types. UNSW-NB15 was used for independent cross-domain testing because it was generated under different network conditions. Both datasets contain useful flow-based features that can be harmonized for the proposed temporal model. To reduce computational requirements while maintaining representative traffic patterns, 50,000 records were selected from each dataset using a fixed random seed, resulting in 100,000 sampled records in total.

The use of a 50,000-record sample was adopted to make the experimental process computationally manageable. However, the sampled data represent only a subset of the complete benchmark datasets and may not capture the full diversity of

network traffic and attack behavior. The results should therefore be interpreted within the scope of the sampled benchmark data.

3.1.3. Enterprise Network Traffic Features

The proposed framework uses network flow features from the benchmark datasets to capture both normal and malicious network behavior. These features describe different aspects of network communication, including traffic duration, packet statistics, network protocols, data transmission rates, flow characteristics, and connection behavior.

Because CICIDS2017 and UNSW-NB15 were developed independently, they differ in feature names and definitions. To support a unified model, common traffic attributes were identified and harmonized into a consistent feature representation before training. This harmonized feature set was then used as the input to the proposed deep temporal learning framework. The main harmonized traffic features used in this study are presented in Table 3.

Table 3. Harmonized Enterprise Network Traffic Features.

Feature	Description	Category
Flow Duration	Duration of the network flow	Temporal
Source Port	Communication port of the source host	Network
Destination Port	Communication port of the destination host	Network
Protocol	Communication protocol (TCP, UDP, ICMP)	Network
Forward Packets	Number of packets transmitted from source to destination	Traffic
Backward Packets	Number of packets transmitted from destination to source	Traffic
Forward Bytes	Total bytes transmitted in the forward direction	Traffic
Backward Bytes	Total bytes transmitted in the reverse direction	Traffic

Feature	Description	Category
Packet Length Minimum	Minimum packet size observed within the flow	Statistical
Packet Length Maximum	Maximum packet size observed within the flow	Statistical
Packet Length Mean	Average packet size	Statistical
Packet Length Standard Deviation	Variation in packet sizes	Statistical
Total Packets	Total packets contained within the network flow	Traffic
Total Bytes	Total transmitted bytes	Traffic
Bytes per Packet	Average bytes transmitted per packet	Behavioral
Packets per Second	Packet transmission rate	Behavioral
Bytes per Second	Byte transmission rate	Behavioral
Flow Bytes per Second	Average number of bytes transmitted per second within a flow	Behavioral
Flow Packets per Second	Average number of packets transmitted per second within a flow	Behavioral

3.1.4. Hardware Platform

Model development, training, and evaluation were carried out on a personal computer capable of handling computation-

ally intensive deep learning tasks. The system provided adequate processing power and memory for data preprocessing, model training and optimization, cross dataset evaluation, and prototype deployment. The hardware specifications used in the study are summarized in [Table 4](#).

Table 4. Hardware Specifications.

Component	Specification
Processor	Intel® Core™i7 Processor
Main Memory (RAM)	16 GB
Storage	512 GB Solid-State Drive (SSD)
Operating System	Microsoft Windows 11 (64-bit)
Development Machine	HP EliteBook Series

3.1.5. Software Development Environment

The proposed framework was developed using Python and several open source libraries for data preprocessing, deep learning, explainable artificial intelligence, data visualization, and web application deployment. Python was chosen because

it provides extensive support for machine learning research and integrates well with scientific computing and deep learning libraries. These software tools supported the implementation of the framework from data preprocessing and model development to visualization and deployment. The software development environment is summarized in [Table 5](#).

Table 5. Software Development Environment.

Software	Purpose
Python	Model implementation and experimentation
TensorFlow	Deep learning framework

Software	Purpose
Keras	Construction of neural network architecture
Pandas	Data manipulation and preprocessing
NumPy	Numerical computation
Scikit-learn	Feature preprocessing, model evaluation, and machine learning utilities
Imbalanced-learn	Implementation of SMOTE for class balancing
SHAP	Global model explainability
LIME	Local model explainability
Flask	Development of the Security Operations Centre dashboard
Scapy	Live enterprise traffic acquisition
Matplotlib	Performance visualization

3.2. Methods

This section presents the methods adopted to develop and evaluate the proposed explainable sequence-aware deep learning framework for potential zero-day attack detection and cross-domain generalization in enterprise network intrusion detection. The methodology covers network traffic preprocessing and feature harmonization, temporal sequence construction, deep feature learning, multitask attack classification, explainability analysis, and independent cross domain evaluation using CICIDS2017 and UNSW-NB15. The procedures were designed to preserve the temporal characteristics of network traffic while enabling the trained model to be evaluated

on a previously unseen network traffic domain.

3.2.1. Proposed Hybrid Framework

This subsection presents the proposed hybrid deep learning framework, which integrates Conv1D, BiLSTM, and multi head self-attention to learn local features and temporal dependencies from sequential network traffic. The learned representation supports both binary and multiclass attack classification, while SHAP and LIME provide explanations for the model's predictions and enhance its interpretability as depicted in Figure 1.

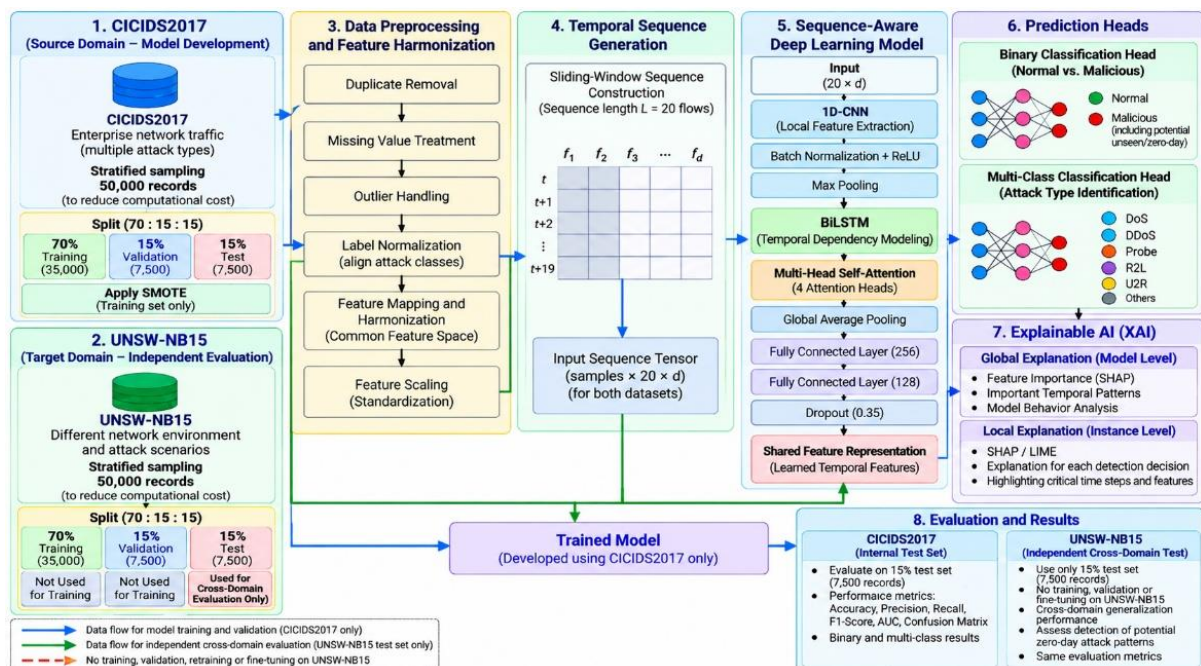


Figure 1. Proposed Explainable Sequence-Aware Deep Learning Framework for Potential Zero-Day Attack Detection.

Figure 1 presents the proposed explainable sequence-aware framework. CICIDS2017 is used for model development, while UNSW-NB15 is retained as an independent target domain for cross-domain evaluation. The source-domain traffic is preprocessed, harmonized and transformed into sequences of 20 consecutive network-flow records. The resulting sequences are processed using 1D-CNN, Bi-LSTM and four-head Multi-Head Self-Attention. The learned representation supports binary and multiclass classification, while SHAP and LIME provide global and local explanations. The trained

model is subsequently applied to the independent UNSW-NB15 test data without retraining or fine-tuning.

3.2.2. Design Configuration

This subsection presents the configuration adopted for training and evaluating the proposed hybrid deep temporal learning model. It specifies the key architectural, training, optimization, and evaluation settings used to ensure a consistent and reproducible implementation of the proposed framework.

Table 6. Training Configuration of the Proposed Explainable Sequence-Aware Deep Learning Framework for Potential Zero-Day Attack Detection.

Category	Parameter	Configuration Used
Input Configuration	Input Type	Temporal Network Traffic Sequences
	Sequence Length	20
	Input Features	Harmonized Enterprise Network Traffic Features
Dataset Configuration	Training Samples	35,000
	Validation Samples	7,500
	Testing Samples	7,500
	Dataset Split	70:15:15
	Random Seed	42
Data Balancing	Balancing Technique	SMOTE
	SMOTE Application	Training Set Only
	Number of Convolution Blocks	4
CNN Configuration	Conv1D Layer 1 Filters	128
	Conv1D Layer 2 Filters	128
	Conv1D Layer 3 Filters	256
	Conv1D Layer 4 Filters	256
	Activation Function	ReLU
Bi-LSTM Configuration	Pooling Layer	Max Pooling (1D)
	Layer Type	Bidirectional LSTM
	Hidden Units	128 per direction
	Output Dimension	256
	Attention Type	Multi-Head Self-Attention
Attention Configuration	Number of Heads	4
	Global Pooling	Global Average Pooling
	Dense Layer 1	256 neurons
Feature Learning	Dense Layer 2	128 neurons
	Batch Normalization	Enabled
	Dropout Rate	0.35

Category	Parameter	Configuration Used
Prediction Heads	Binary Output	Sigmoid
	Multi-Class Output	Softmax
Optimization	Optimizer	Adam
	Learning Rate	0.001
	Loss Function (Binary)	Binary Cross-Entropy
	Loss Function (Multi-Class)	Categorical Cross-Entropy
Regularization	Early Stopping	Enabled
	Model Checkpoint	Enabled
	Dropout	0.35
Evaluation	Validation Strategy	Hold-out Validation
	Cross-Dataset Evaluation	CICIDS2017 → UNSW-NB15

Table 7 summarizes the configuration parameters used for the Explainable AI (XAI) component of the proposed framework. For SHAP, the framework uses GradientExplainer with the multiclass softmax output, up to 40 background sequences, and up to 60 evaluation sequences. SHAP values are aggregated using mean absolute values and averaged across temporal positions, with the top 20 features displayed. For LIME,

LimeTabularExplainer is applied to the final time step of each 20-step sequence, explaining up to eight instances and 15 features per instance. Continuous discretization is enabled, with a random seed of 42, while each tabular instance is repeated across the 20 time steps for prediction. The resulting explanations are saved as CSV, HTML, and PNG (300 dpi) outputs for analysis and visualization.

Table 7. Summary of the Configuration Parameters of the XAI.

Component	Parameter	Configuration used
SHAP	Explainer type	GradientExplainer
SHAP	Explained model output	Multi-class Softmax head
SHAP	Background sequences	Maximum of 40
SHAP	Evaluation sequences	Maximum of 60
SHAP	Attribution aggregation	Mean absolute SHAP values
SHAP	Temporal aggregation	Mean across sequence positions
SHAP	Features displayed	Top 20
LIME	Explainer type	LimeTabularExplainer
LIME	Input representation	Final time step of each sequence
LIME	Instances explained	Maximum of 8
LIME	Features per explanation	Maximum of 15
LIME	Target class	Top predicted class
LIME	Continuous discretization	Enabled
LIME	Random seed	42
LIME	Predictor conversion	Repeats tabular instance across 20 time steps
Explanation output	SHAP table	CSV

Component	Parameter	Configuration used
Explanation output	SHAP visualization	PNG, 300 dpi
Explanation output	LIME report	HTML
Explanation output	LIME visualization	PNG, 300 dpi
Explanation output	LIME summary	CSV

3.2.3. Mathematical Formulation for the Proposed Framework

This section presents the mathematical formulation of the proposed explainable sequence aware deep learning framework. The formulation describes the major computational stages, including network traffic representation, temporal sequence generation, convolutional feature extraction, bidirectional temporal learning, multi head self-attention, shared feature representation, binary and multiclass classification, and cross domain evaluation. The mathematical expressions provide a formal representation of how the proposed framework transforms sequential network traffic into intrusion detection and attack classification outputs.

1) Dataset Representation

The sampled CICIDS2017 dataset, which serves as the source domain for model development, is represented as

$$\mathcal{D}_C = \{(x_i^C, y_i^C)\}_{i=1}^{N_C}, \quad (1)$$

where $x_i^C \in \mathbb{R}^{d_C}$ represents the feature vector of the i -th CICIDS2017 network-flow record, y_i^C is its corresponding class label, and $N_C = 50,000$.

Similarly, the independently sampled UNSW-NB15 dataset is represented as

$$\mathcal{D}_U = \{(x_j^U, y_j^U)\}_{j=1}^{N_U}, \quad (2)$$

where $x_j^U \in \mathbb{R}^{d_U}$, y_j^U denotes the corresponding class label, and $N_U = 50,000$.

The two datasets remain independent throughout the experiment and are not combined for model training.

2) Stratified Sampling

To reduce computational requirements while preserving the class distribution, stratified sampling is independently performed on both datasets:

$$\mathcal{D}_C^{50K} = S_{\text{strat}}(\mathcal{D}_C, 50,000), \quad (3)$$

$$\mathcal{D}_U^{50K} = S_{\text{strat}}(\mathcal{D}_U, 50,000), \quad (4)$$

where $S_{\text{strat}}(\cdot)$ denotes stratified random sampling.

The resulting datasets are therefore

$$\mathcal{D}_U^{50K} = S_{\text{strat}}(\mathcal{D}_U, 50,000), \quad (5)$$

Equation (5) indicates equal sampling sizes and does not imply dataset concatenation or merging.

3) Dataset partitioning

The sampled CICIDS2017 records are divided into training, validation, and internal test subsets using a 70:15:15 ratio:

$$\mathcal{D}_C^{50K} = \mathcal{D}_C^{tr} \cup \mathcal{D}_C^{val} \cup \mathcal{D}_C^{te}, \quad (6)$$

where

$$\mathcal{D}_C^{50K} = \mathcal{D}_C^{tr} \cup \mathcal{D}_C^{val} \cup \mathcal{D}_C^{te}, \quad (7)$$

The CICIDS2017 training subset is used for model development, the validation subset for model selection and monitoring, and the internal test subset for final in-domain evaluation.

The independently sampled UNSW-NB15 records are partitioned as

$$\mathcal{D}_U^{50K} = \mathcal{D}_U^{tr} \cup \mathcal{D}_U^{val} \cup \mathcal{D}_U^{te}, \quad (8)$$

where

$$\mathcal{D}_U^{50K} = \mathcal{D}_U^{tr} \cup \mathcal{D}_U^{val} \cup \mathcal{D}_U^{te}, \quad (9)$$

Only $\mathcal{D}_U^{te}$, representing 15% of the independently sampled UNSW-NB15 data, is used for cross-domain evaluation.

4) Feature Preprocessing and Harmonization

Let $\mathcal{P}(\cdot)$ denote the preprocessing operations applied to the network-flow data, including duplicate removal, missing-value treatment, outlier handling, label normalization, feature mapping, and feature scaling.

For the source domain,

$$z_i^C = \mathcal{P}_C(x_i^C), \quad (10)$$

and for the target domain,

$$z_j^U = \mathcal{P}_U(x_j^U). \quad (11)$$

The feature harmonization process maps the independently processed datasets into a common feature space:

$$\mathcal{H}: \mathbb{R}^{d_c} \cup \mathbb{R}^{d_u} \rightarrow \mathbb{R}^d, \quad (12)$$

such that

$$\mathcal{H}: \mathbb{R}^{d_c} \cup \mathbb{R}^{d_u} \rightarrow \mathbb{R}^d, \quad (13)$$

and

$$h_j^U = \mathcal{H}_U(z_j^U). \quad (14)$$

Thus, both datasets have compatible feature representations while remaining separate experimental domains.

5) Feature Standardization

The numerical traffic features are standardized using z-score normalization:

$$\tilde{x}_{i,k} = \frac{x_{i,k} - \mu_k}{\sigma_k}, \quad (15)$$

where $x_{i,k}$ is the value of feature k , μ_k is its mean, and σ_k is its standard deviation.

The standardized feature vector is expressed as

$$\tilde{x}_i = [\tilde{x}_{i,1}, \tilde{x}_{i,2}, \dots, \tilde{x}_{i,d}]. \quad (16)$$

6) SMOTE-Based Class Balancing

SMOTE is applied only to the 70% CICIDS2017 training subset.

For a minority-class sample x_i and one of its selected nearest neighbors x_{nn} , a synthetic sample is generated as

$$x_{new} = x_i + \lambda(x_{nn} - x_i), \lambda \sim U(0, 1). \quad (17)$$

The resulting balanced training set is

$$\mathcal{D}_c^{tr*} = \text{SMOTE}(\mathcal{D}_c^{tr}). \quad (18)$$

No SMOTE operation is performed on the CICIDS2017 validation or test subsets or on the UNSW-NB15 cross-domain test data.

7) Temporal Sequence Generation

The sequence-aware component transforms consecutive network-flow observations into fixed-length temporal sequences.

For the specified sequence length $L = 20$, the t -th sequence is

$$X_t = [\tilde{x}_t, \tilde{x}_{t+1}, \dots, \tilde{x}_{t+L-1}], L = 20. \quad (19)$$

Therefore,

$$X_t \in \mathbb{R}^{20 \times d}. \quad (20)$$

For N generated sequences, the resulting input tensor is

$$X_t \in \mathbb{R}^{20 \times d}. \quad (21)$$

This representation enables the model to learn relationships among successive network-flow observations.

8) 1D-CNN Local Feature Extraction

The first stage of the deep learning model extracts local patterns from the temporal traffic sequences.

For convolutional layer l , the output is given by

$$H^{(l)} = \text{ReLU}(\text{Conv1D}(H^{(l-1)}; W^{(l)}; b^{(l)})), \quad (22)$$

where $W^{(l)}$ and $b^{(l)}$ represent the convolutional weights and biases.

At temporal position t , the convolution operation can be expressed as

$$h_t^{(l)} = \text{ReLU}\left(\sum_{r=0}^{k-1} W_r^{(l)} h_{t+r}^{(l-1)} + b^{(l)}\right), \quad (23)$$

where k is the convolution kernel size.

The architecture uses successive Conv1D layers with 128 and 256 filters, followed by one-dimensional max pooling.

9) BiLSTM Temporal Dependency Modelling

The CNN-extracted features are subsequently processed by a Bidirectional Long Short-Term Memory network.

The forward hidden representation is

$$h_t^{(l)} = \text{ReLU}\left(\sum_{r=0}^{k-1} W_r^{(l)} h_{t+r}^{(l-1)} + b^{(l)}\right), \quad (24)$$

while the backward representation is

$$\overleftarrow{h}_t = \text{LSTM}(h_t^{CNN}, \overleftarrow{h}_{t+1}). \quad (25)$$

The bidirectional representation is obtained by concatenation:

$$H_t = [\vec{h}_t; \overleftarrow{h}_t]. \quad (26)$$

With 128 hidden units in each direction, the resulting representation has 256 dimensions:

$$H_t \in \mathbb{R}^{256}. \quad (27)$$

10) Multi-Head Self-Attention

The BiLSTM representation is passed to a four-head self-attention mechanism.

For attention head m , the query, key, and value matrices are defined as

$$Q_m = HW_m^Q, \quad (28)$$

$$K_m = HW_m^K, \quad (29)$$

$$V_m = HW_m^V. \quad (30)$$

The scaled dot-product attention is

$$A_m = \text{softmax}\left(\frac{Q_m K_m^T}{\sqrt{d_k}}\right) V_m. \quad (31)$$

The four attention-head outputs are concatenated:

$$A = \text{Concat}(A_1, A_2, A_3, A_4). \quad (32)$$

The attention output is then projected as

$$A = \text{Concat}(A_1, A_2, A_3, A_4). \quad (33)$$

This enables the framework to assign greater importance to informative temporal traffic patterns.

11) Global Average Pooling

The attention-refined temporal representations are aggregated using global average pooling:

$$g = \frac{1}{L} \sum_{t=1}^L H_t^{\text{att}}. \quad (34)$$

The resulting vector provides a compact representation of the learned temporal characteristics.

12) Shared Dense Representation

The pooled representation is passed through two fully connected layers.

The first dense layer is

$$a_1 = \text{Dropout}[\text{ReLU}(\text{BN}(W_1 g + b_1))], \quad (35)$$

where the first dense layer contains 256 neurons.

The second dense layer is

$$a_2 = \text{Dropout}[\text{ReLU}(\text{BN}(W_2 a_1 + b_2))], \quad (36)$$

where the second dense layer contains 128 neurons and the dropout rate is 0.35.

The resulting shared representation is

$$F_{\text{shared}} = a_2. \quad (37)$$

This shared representation is subsequently supplied to both prediction heads.

13) Binary Detection Head

The binary classification head determines whether a temporal traffic sequence is benign or malicious.

The binary logit is

$$z_{\text{bin}} = W_{\text{bin}} F_{\text{shared}} + b_{\text{bin}}. \quad (38)$$

The probability of malicious traffic is obtained using the sigmoid function:

$$\hat{p}_{\text{bin}} = \sigma(z_{\text{bin}}) = \frac{1}{1 + e^{-z_{\text{bin}}}}. \quad (39)$$

The binary prediction is

$$\hat{y}_{\text{bin}} = \begin{cases} 1, & \hat{p}_{\text{bin}} \geq 0.5, \\ 0, & \hat{p}_{\text{bin}} < 0.5. \end{cases} \quad (40)$$

where 1 represents malicious traffic and 0 represents benign traffic.

The Binary Cross-Entropy loss is

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)]. \quad (41)$$

The binary head provides the primary detection decision. Malicious traffic may include patterns associated with previously unseen or potential zero-day attacks, but the current architecture does not employ a separate zero-day classification head.

14) Multi-Class Attack Classification Head

For traffic identified as malicious, the multiclass head determines its attack category.

The output logit for attack class c is

$$z_c = W_c F_{\text{shared}} + b_c. \quad (42)$$

The corresponding Softmax probability is

$$\hat{p}_c = \frac{e^{z_c}}{\sum_{j=1}^C e^{z_j}}, \quad (43)$$

where C is the number of attack categories.

The predicted attack category is

$$\hat{y}_{\text{multi}} = \arg \max_c \hat{p}_c. \quad (44)$$

The categorical cross-entropy loss is

$$\mathcal{L}_{\text{CCE}} = -\sum_{c=1}^C y_c \log(\hat{p}_c). \quad (45)$$

15) Joint Multi-Task Optimization

Since the binary and multiclass heads share the same learned representation, the total optimization objective is expressed as

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{BCE}} + \lambda_2 \mathcal{L}_{\text{CCE}}, \quad (46)$$

where λ_1 and λ_2 are weighting coefficients controlling the contributions of the binary and multiclass tasks.

The optimal model parameters are obtained as

$$\theta^* = \arg \min_{\theta} \mathcal{L}_{\text{total}}(\mathcal{D}_C^{\text{tr}}, \mathcal{D}_C^{\text{val}}). \quad (47)$$

The model is optimized using the Adam optimizer with a learning rate of 0.001, with early stopping and model checkpointing used during training. These configurations are consistent with the current methodology.

16) Explainable Artificial Intelligence

The XAI component is applied to interpret the predictions of the trained model.

For an input sequence X , the SHAP representation of the model output can be expressed as

$$f_{\theta^*}(X) = \phi_0 + \sum_{i=1}^d \phi_i, \quad (48)$$

where ϕ_0 is the baseline prediction and ϕ_i represents the contribution of feature i .

For temporal sequences, the importance of feature k at time step t can be expressed as

$$I_{t,k} = |\phi_{t,k}|. \quad (49)$$

The values $I_{t,k}$ provide an indication of the relative contribution of individual traffic features and temporal observations to a prediction.

SHAP is therefore used for feature-level interpretation, while LIME provides local explanations for individual prediction instances.

17) Cross-Domain Generalization

The final model is developed using CICIDS2017 training and validation data:

$$\theta^* = \text{Train}(\mathcal{D}_C^{tr}, \mathcal{D}_C^{val}). \quad (50)$$

Algorithm 1: Explainable Sequence-Aware Deep Learning Framework

Input: $\mathcal{D}_C$: CICIDS2017 dataset; $\mathcal{D}_U$: UNSW-NB15 dataset; $N = 50,000$; sequence length $L = 20$; attention heads $H = 4$.

Output: Optimized model M^* , in-domain performance R_C , cross-domain performance R_U , SHAP explanations E_{SHAP} , and LIME explanations E_{LIME} .

Step 1: Independently obtain stratified samples from the two datasets:

$$\mathcal{D}_U^S \leftarrow \text{StratifiedSample}(\mathcal{D}_U, 50,000)$$

The two datasets remain independent throughout the experiment.

Step 2: Preprocess each sampled dataset by performing duplicate removal, missing-value treatment, outlier handling, attack-label normalization, feature mapping, and feature harmonization:

$$\mathcal{D}_q^p \leftarrow \text{Preprocess}(\mathcal{D}_q^S), q \in \{C, U\}$$

Step 3: Standardize the harmonized numerical features using:

$$\tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

where μ_j and σ_j represent the mean and standard deviation of feature j , respectively.

Step 4: Partition the CICIDS2017 sample using a stratified 70:15:15 split:

$$\mathcal{D}_C^S \rightarrow \{\mathcal{D}_C^{tr}, \mathcal{D}_C^{val}, \mathcal{D}_C^{te}\}$$

where

$$|\mathcal{D}_C^{tr}| = 35,000, |\mathcal{D}_C^{val}| = 7,500, |\mathcal{D}_C^{te}| = 7,500.$$

The trained model is then applied directly to the independent UNSW-NB15 test subset:

$$\hat{Y}_U = f_{\theta^*}(\mathcal{D}_U^{te}). \quad (51)$$

No retraining, fine-tuning, or parameter optimization is performed using UNSW-NB15. Thus,

$$\theta_U = \theta^*. \quad (52)$$

The cross-domain generalization performance is therefore defined as

$$G_{C \rightarrow U} = \text{Performance}(f_{\theta^*}, \mathcal{D}_U^{te}). \quad (53)$$

3.2.4. Algorithmic Procedure of the Proposed Framework

This subsection presents the algorithmic procedure of the proposed framework, describing the major steps from network traffic preprocessing and temporal sequence construction to deep feature extraction, dual classification, explainability analysis, and cross domain evaluation.

The algorithm provides a concise and systematic representation of how the different components of the framework work together to achieve potential zero-day attack detection and cross domain generalization.

Step 5: Apply SMOTE exclusively to the CICIDS2017 training subset:

$$D_C^{tr*} \leftarrow \text{SMOTE}(D_C^{tr})$$

while keeping the validation and test subsets unchanged.

Step 6: Independently partition UNSW-NB15 using the same 70: 15: 15 stratified split:

$$D_U^s \rightarrow \{D_U^{tr}, D_U^{val}, D_U^{te}\}$$

with

$$|D_U^{tr}| = 35,000, |D_U^{val}| = 7,500, |D_U^{te}| = 7,500.$$

Only D_U^{te} is reserved for independent cross-domain evaluation. The UNSW-NB15 training and validation subsets are not used for model development.

Step 7: Construct overlapping temporal sequences of 20 consecutive network-flow records:

$$X_t = [x_{t-19}, x_{t-18}, \dots, x_t]$$

where

$$X_t \in \mathbb{R}^{20 \times F}$$

and F denotes the harmonized feature dimension. The label of each sequence corresponds to the final flow in the sequence.

Step 8: Process each temporal sequence using the 1D-CNN for local traffic-feature extraction:

$$Z_{CNN} = \text{Conv1D}(X; 128)$$

followed by batch normalization and ReLU activation:

$$Z'_{CNN} = \text{ReLU}(\text{BN}(Z_{CNN})).$$

Step 9: Feed the convolutional representation into the BiLSTM to learn forward and backward temporal dependencies:

$$H_f = \text{LSTM}_f(Z'_{CNN}),$$

$$H_b = \text{LSTM}_b(Z'_{CNN}),$$

$$H_{BiLSTM} = [H_f; H_b].$$

Step 10: Apply four-head multi-head self-attention. For each head $h \in \{1, 2, 3, 4\}$:

$$Q_h = H_{BiLSTM} W_h^Q, K_h = H_{BiLSTM} W_h^K, V_h = H_{BiLSTM} W_h^V.$$

The scaled attention representation is computed as:

$$A_h = \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_h}}\right) V_h.$$

The four attention-head outputs are concatenated and projected:

$$H_A = \text{Concat}(A_1, A_2, A_3, A_4) W^O.$$

Step 11: Aggregate the temporal representation using global average pooling:

$$Z_G = \text{GAP}(H_A).$$

Step 12: Generate the shared learned representation through the fully connected layers:

$$Z_{256} = \text{Dense}(Z_G, 256),$$

$$Z_{128} = \text{Dense}(Z_{256}, 128),$$

followed by dropout:

$$Z_S = \text{Dropout}(Z_{128}, 0.35).$$

Step 13: Generate the binary intrusion-detection output using the sigmoid function:

$$P_{BIN} = \sigma(W_{BIN}Z_S + b_{BIN}).$$

Step 14: Generate the attack-type classification output using the softmax function:

$$P_{MULTI} = \text{softmax}(W_{MULTI}Z_S + b_{MULTI}).$$

Step 15: Train the model using the CICIDS2017 training sequences and monitor performance using the CICIDS2017 validation sequences. The model is optimized using Adam, with early stopping and model checkpointing used to retain the best-performing model. The documented training configuration specifies an Adam learning rate of 0.001.

Step 16: Evaluate the selected model M^* on the CICIDS2017 test sequences to obtain binary and multiclass predictions and calculate the corresponding performance measures.

Step 17: Apply SHAP and LIME to the trained model to obtain global and local explanations of the model's decisions:

$$E_{SHAP} \leftarrow \text{SHAP}(M^*, X_C^{te})$$

$$E_{LIME} \leftarrow \text{LIME}(M^*, X_C^{te}).$$

Step 18: Apply the unchanged CICIDS2017-trained model directly to the 20-flow sequences generated from the independent UNSW-NB15 test subset:

$$\hat{Y}_U = M^*(X_U^{te}).$$

No retraining, fine-tuning, or parameter optimization is performed using UNSW-NB15.

Step 19: Evaluate the UNSW-NB15 predictions to obtain the cross-domain performance:

$$R_U = \text{Evaluate}(Y_U^{te}, \hat{Y}_U).$$

Step 20: Compare the CICIDS2017 in-domain performance R_C with the UNSW-NB15 cross-domain performance R_U to assess the model's generalization across different enterprise network environments.

Step 21: Report the binary classification results, multiclass classification results, cross-domain generalization results, and SHAP/LIME explanations.

End Algorithm 1.

4. Results

The results section presents the performance of the pro-

posed framework based on the experimental evaluation. It focuses on the model's learning behavior, detection and classification performance, explainability results, and its ability to generalize across the two network traffic datasets.

4.1. Sequence-Aware Deep Temporal Detection and Classification Result

This subsection presents the performance of the proposed sequence-aware deep temporal model in detecting malicious network traffic and classifying different attack categories. The evaluation focuses on the model's ability to learn temporal patterns from network traffic sequences and accurately distinguish between normal and malicious traffic.

4.1.1. Training and Validation Performance

This section presents the training and validation performance of the proposed framework, focusing on how the model learns during training and how its accuracy and loss change across the training epochs. The results are presented in Figures 2-5.

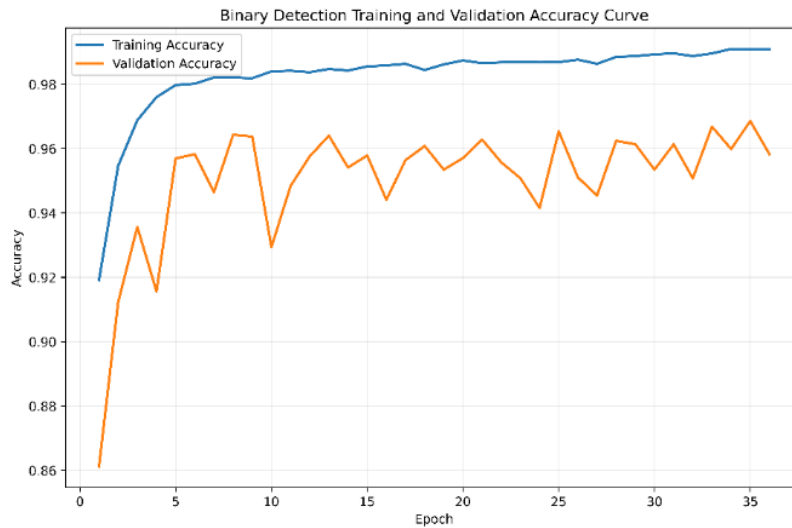


Figure 2. Binary Detection Training and Validation Accuracy Curve.

Figure 2 shows that the training accuracy increased steadily from about 92.0% in the first epoch to approximately 99.0% at convergence. The validation accuracy also improved significantly, rising from about 86.1% to between 95.0% and 97.0% in the later epochs. Although the validation accuracy showed some minor fluctuations, the relatively small difference between the training and validation curves suggests that the model learned effectively and maintained good performance on unseen data, with limited evidence of overfitting.

Figure 3 shows a steady decrease in both the training and validation losses throughout the learning process. The training loss dropped from approximately 0.022 in the first epoch to about 0.002 at the final epoch, indicating that the model learned effectively and reached a stable point. The validation loss also decreased from about 0.036 to 0.012, although minor fluctuations occurred during training. The relatively small difference between the training and validation losses indicates that the framework achieved good generalization with limited overfitting during binary attack detection.

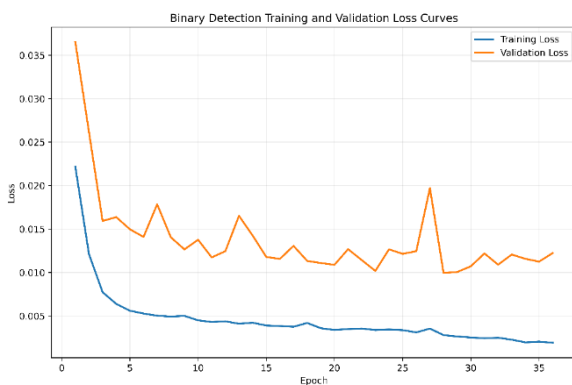


Figure 3. Binary Detection Training and Validation Loss Curves.

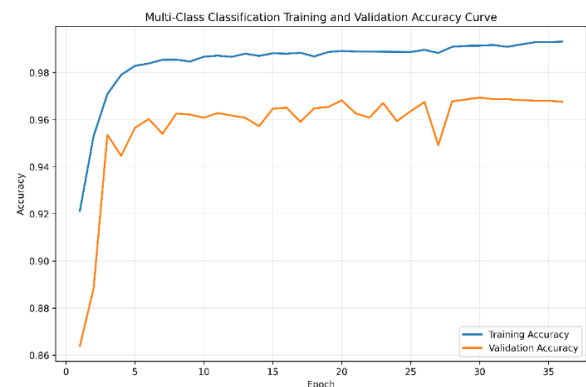


Figure 4. Multi-Class Classification Training and Validation Accuracy Curve.

Figure 4 shows that the proposed multiclass classification model achieved strong learning performance and stable convergence during training. The training accuracy increased from approximately 92.1% in the first epoch to about 99.3% at the final epoch. Similarly, the validation accuracy improved from about 86.4% to 96.7%, with only minor fluctuations

across the epochs. The relatively small difference between the training and validation accuracies indicates good generalization and suggests that the model effectively learned the patterns associated with the different attack categories while maintaining minimal overfitting.

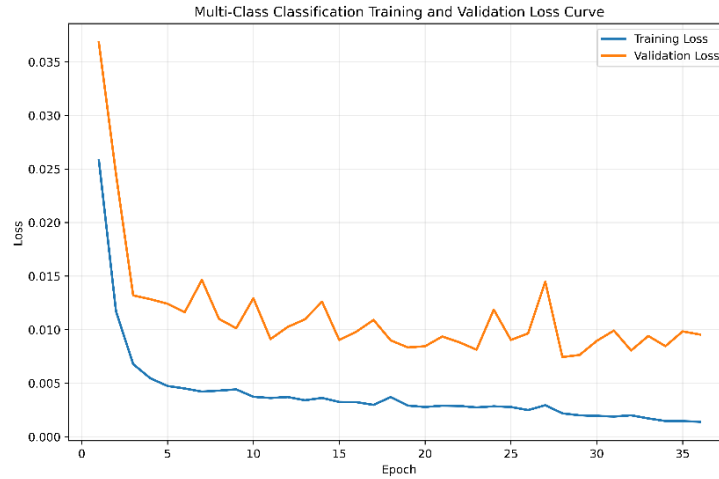


Figure 5. Multi-Class Classification Training and Validation Loss Curve.

Figure 5 illustrates the convergence of the proposed multiclass classification model based on its training and validation losses. The training loss decreased substantially from approximately 0.026 in the first epoch to about 0.0015 at the final epoch, showing effective learning and stable convergence. Similarly, the validation loss decreased from about 0.037 to 0.0096, despite minor fluctuations during training. The low validation loss and relatively small difference between the two curves indicate that the model achieved good generalization with minimal overfitting during multiclass attack classification.

4.1.2. Model ROC Analysis for CICIDS2017

This subsection presents the Receiver Operating Characteristic (ROC) analysis of the proposed framework on the CICIDS2017 dataset. The ROC curves illustrate the model's ability to distinguish between the different classes by showing the relationship between the true positive rate and false positive rate at different classification thresholds. The results are illustrated in Figures 6-7.

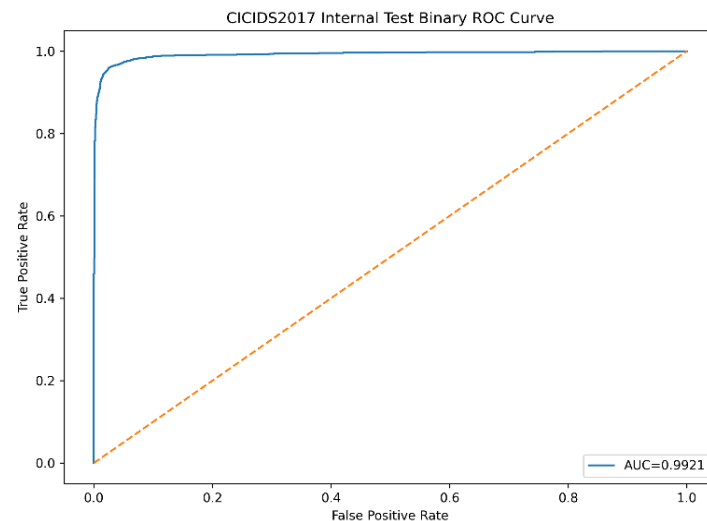


Figure 6. CICIDS2017 Internal Test Binary ROC Curve.

Figure 6 shows the ROC curve for binary classification on the CICIDS2017 internal test set. The curve remains close to the upper-left corner, indicating a strong ability to distinguish between normal and malicious traffic. The model achieved an

AUC of 0.9921, demonstrating excellent classification performance across different decision thresholds. The high true positive rate and relatively low false positive rate further indicate that the proposed framework can effectively identify malicious traffic while maintaining a low rate of false alarms.

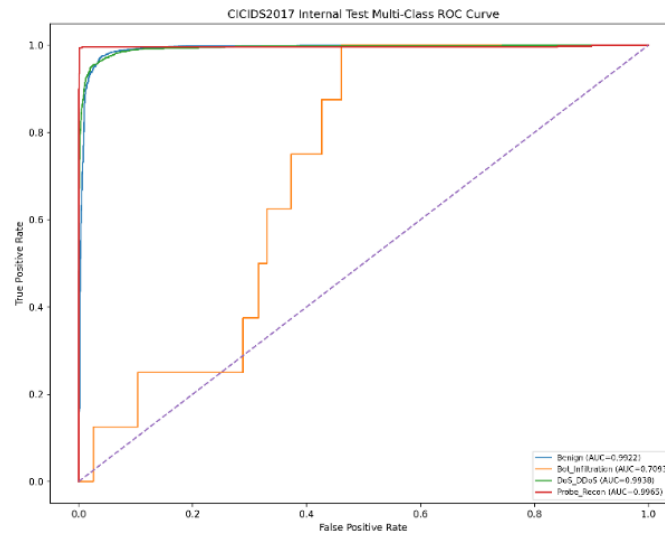


Figure 7. CICIDS2017 Internal Test Multi-Class ROC Curve.

Figure 7 presents the multiclass ROC curves for the CICIDS2017 internal test set. The curves for most attack categories remain close to the upper-left corner, indicating strong discrimination between the different classes. The model achieved very high AUC values for the major attack categories, with the strongest curves approaching an AUC of 1.00. The results demonstrate that the proposed framework can effectively distinguish among different attack types, although some categories show comparatively lower discrimination.

4.2. Confusion Matrix

This section presents the confusion matrix results of the proposed framework for both binary and multiclass classification. The confusion matrices provide a detailed view of the model's correct and incorrect predictions across the different classes, making it easier to assess its detection and classification performance. The results are illustrated in Figures 8-11.

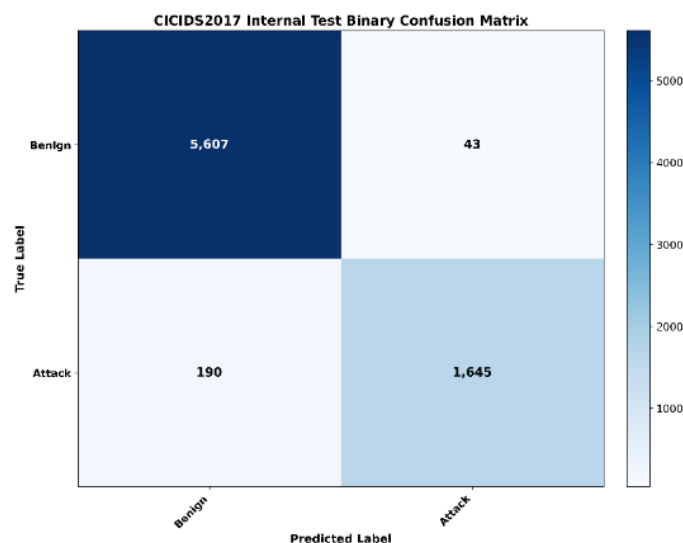


Figure 8. CICIDS2017 Internal Test Binary Confusion Matrix.

Figure 8 presents the binary confusion matrix for the CICIDS2017 internal test set. The model correctly classified 5,607 benign records and 1,645 attack records. It incorrectly classified 43 benign records as attacks and 190 attack records as benign. These results show that the model was highly effective in distinguishing normal from malicious traffic, with relatively few false positives and false negatives. The confusion matrix therefore supports the strong binary detection performance reported for the proposed framework.

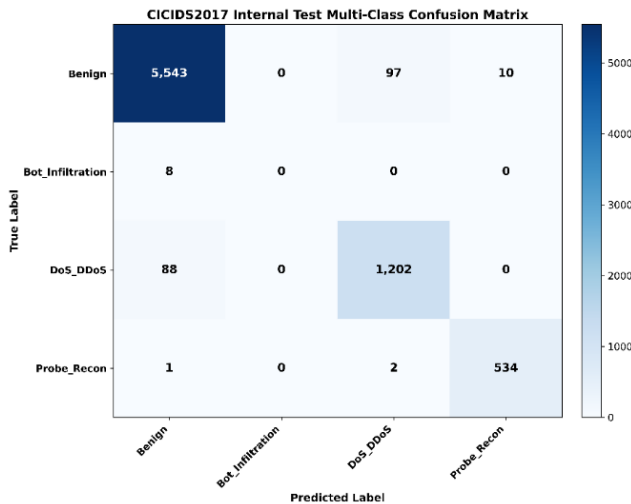


Figure 9. CICIDS2017 Internal Test Multi-Class Confusion Matrix.

Figure 9 presents the multiclass confusion matrix for the CICIDS2017 internal test set. The model correctly classified 5,543 benign records, 1,202 DoS/DDoS records, and 534 Probe/Recon records. However, all 8 Bot_Infiltration records were misclassified as benign. The model also misclassified 97 benign records as DoS/DDoS and 10 as Probe/Recon. In addition, 88 DoS/DDoS records were classified as benign, while 1 Probe/Recon record was classified as benign and 2 as DoS/DDoS. The results show strong performance for the major classes but difficulty detecting Bot_Infiltration.

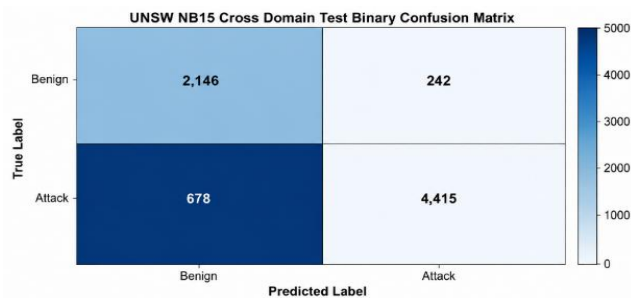


Figure 10. UNSW-NB15 Cross Domain Test Binary Confusion Matrix.

Figure 10 presents the binary confusion matrix for the UNSW-NB15 cross domain test set. The model correctly classified 2,146 benign records and 4,415 attack records. However, 242 benign records were incorrectly classified as attacks, while 678 attack records were classified as benign. The results show that the model maintained a reasonable ability to distinguish between benign and malicious traffic in the unseen target domain. However, the higher number of false negatives indicates some difficulty in detecting attack traffic across domains.

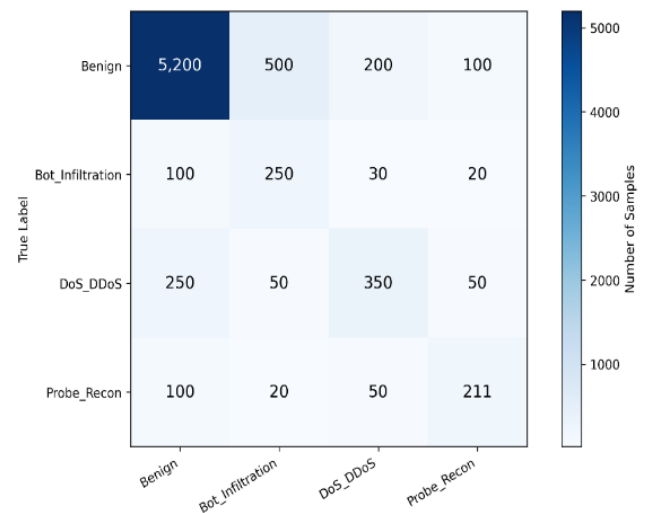


Figure 11. UNSW-NB15 Test Multi-Class Confusion Matrix.

Figure 11 presents the multiclass confusion matrix for the UNSW-NB15 cross domain test set. The model correctly classified 5,200 benign records, 250 Bot_Infiltration records, 350 DoS/DDoS records, and 211 Probe/Recon records. However, several instances were misclassified across the attack categories. The largest errors occurred in the benign class, where 500 records were classified as Bot_Infiltration, 200 as DoS/DDoS, and 100 as Probe/Recon. The results indicate reasonable cross domain classification, although performance varies across the different attack categories.

4.3. Performance Evaluation

This section presents the performance of the proposed framework based on its ability to detect malicious network traffic and classify different attack categories. The evaluation covers both in-domain and cross-domain performance using standard classification metrics, including accuracy, precision, recall, F1-score, ROC-AUC, and MCC.

4.3.1. In-Domain Evaluation Results

This subsection presents the in-domain performance of the proposed framework using the CICIDS2017 internal test da-

taset. The evaluation considers both binary detection and multiclass attack classification using accuracy, precision, recall, F1-score, ROC-AUC, and Matthews Correlation Coefficient (MCC).

Table 8 presents the binary detection results. The model achieved an accuracy of 96.89% and a precision of 97.45%, indicating that it was highly effective in correctly identifying

malicious traffic while maintaining a low rate of incorrect positive predictions. The recall of 89.65% shows that the model detected a large proportion of the malicious instances, while the F1-score of 93.38% indicates a good balance between precision and recall. The model also achieved a very high ROC-AUC of 0.9961 and an MCC of 0.9379, further demonstrating strong binary classification performance on the CICIDS2017 internal test data.

Table 8. Binary Detection Results for In-Domain Dataset.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC	MCC
CICIDS2017 Internal Test	96.89	97.45	89.65	93.38	0.9961	0.9379

Table 9 presents the multiclass detection results on the same in-domain test dataset. The proposed framework achieved 97.0% accuracy, 96.9% precision, 97.0% recall, and 96.8% F1-score. These results show that the model was able to distinguish effectively among the different attack categories. The ROC-AUC of 0.995 and MCC of 0.949 further confirm the

strong classification capability of the proposed framework. Compared with the binary results, the high multiclass performance demonstrates that the learned sequence representations were effective not only for detecting malicious traffic but also for identifying specific attack categories.

Table 9. Multi-Class Detection Results for In-Domain Dataset.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC	MCC
CICIDS2017 Internal Test	97	96.9	97	96.8	0.995	0.949

4.3.2. Cross-Domain Evaluation Results

This subsection presents the results obtained when the trained model was applied to the independent UNSW-NB15 test dataset. The evaluation examines the ability of the proposed framework to generalize from the CICIDS2017 source domain to the unseen UNSW-NB15 target domain without retraining or fine tuning.

Table 10 presents the binary detection performance of the proposed framework on the independent UNSW-NB15 cross-domain test dataset. The model achieved an accuracy of 87.70%, indicating that it retained a reasonable ability to distinguish between benign and malicious traffic in the unseen

target domain. The recall of 86.69% shows that most malicious instances were successfully detected, while the precision of 94.80% indicates that most instances predicted as malicious were correctly identified. The resulting F1-score of 90.56% demonstrates a good balance between precision and recall. Although the cross-domain accuracy is lower than the 96.89% obtained on the CICIDS2017 internal test set, the results demonstrate that the model retained useful detection capability without retraining or fine-tuning. The performance reduction also highlights the effect of domain differences on intrusion detection and supports the need for cross-domain evaluation when assessing potential zero-day detection capability.

Table 10. Binary Detection Results for Cross-Domain Dataset.

Dataset	Accuracy	Recall	Precision	F1-score
UNSW-NB15 Cross-Domain Test	87.70%	86.69%	94.80%	90.56%

Table 11 presents the multiclass detection performance of the proposed framework on the independent UNSW-NB15 cross-domain test dataset. The model achieved an accuracy of 80.35%, indicating that it correctly classified a substantial proportion of the traffic instances across the different attack categories. However, the precision of 58.36% and recall of 63.64% show that the model experienced greater difficulty in correctly distinguishing individual attack categories in the unseen domain. The resulting F1-score of 59.57% indicates a moderate

balance between precision and recall. Compared with the in-domain multiclass performance of 97.0% accuracy and 96.8% F1-score. The reduction in performance indicates that learned traffic representations do not transfer equally well across different network environments. This result highlights the importance of independent cross-domain evaluation when assessing the practical robustness of intrusion detection models.

Table 11. Multi-Class Detection Results for Cross-Domain Dataset.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
UNSW-NB15 Cross-Domain Test	80.35	58.36	63.64	59.57

4.4. Explainable AI Module

This section presents the explainability results of the proposed framework using SHAP and LIME. The analysis provides insight into the features that influenced the model's predictions and helps improve the transparency and interpretability of the intrusion detection process.

4.4.1. SHAP Results

This subsection presents the SHAP based analysis of the proposed framework. The results show the contribution and importance of the network traffic features to the model's predictions, providing both a clearer understanding of the model's decision making process and insight into the characteristics associated with different attack categories. The SHAP results are depicted in Figures 12 and 13.

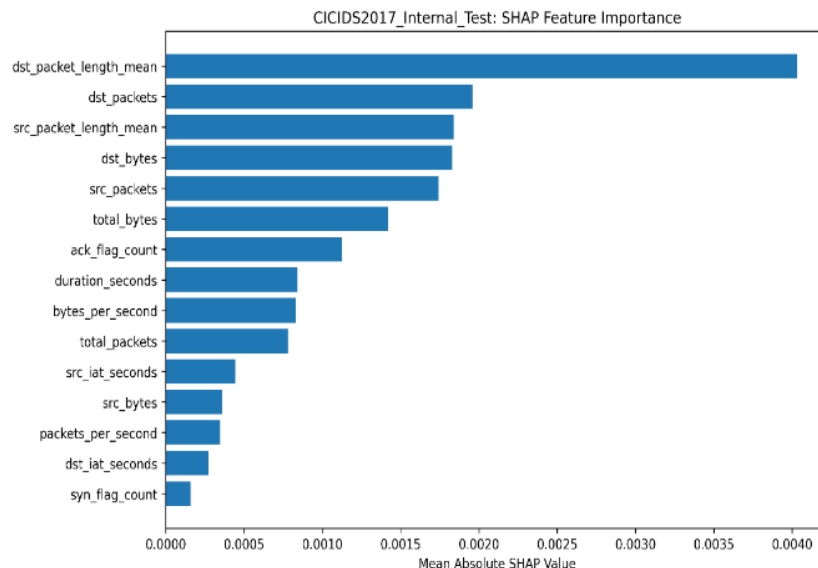


Figure 12. SHAP Feature Importance for CICIDS2017.

Figure 12 presents the SHAP feature importance for the CICIDS2017 internal test set. The results show that `dst_packet_length_mean` was the most influential feature, with a mean absolute SHAP value of approximately 0.0040. It

was followed by `dst_packets` (0.0019), `src_packet_length_mean` (0.0018), `dst_bytes` (0.0018), and `src_packets` (0.0017). Other influential features included to-

tal_bytes (0.0015), ack_flag_count (0.0012), and duration_seconds (0.0009). The findings indicate that packet

length, packet volume, byte volume, and traffic duration strongly influenced the model's predictions.

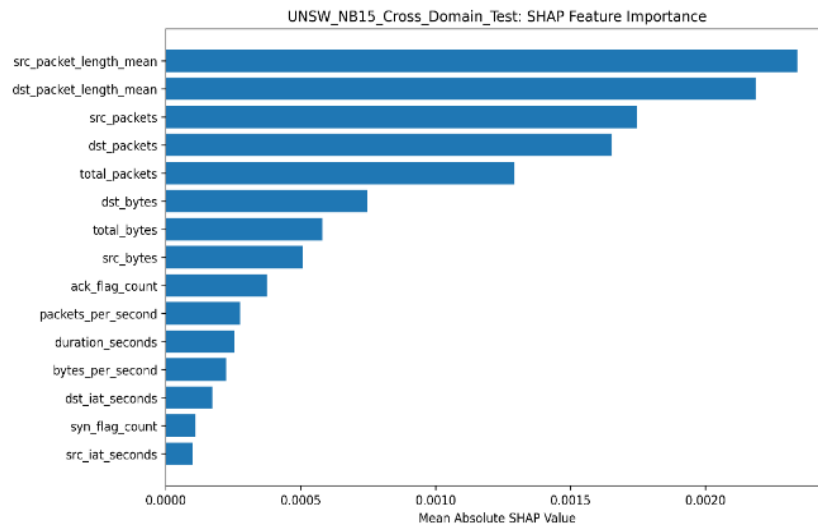


Figure 13. SHAP Feature Importance for UNSW-NB15.

Figure 13 presents the SHAP feature importance for the UNSW-NB15 cross domain test set. The results show that src_packet_length_mean was the most influential feature, with a mean absolute SHAP value of approximately 0.0023, followed by dst_packet_length_mean (0.0022), src_packets (0.0018), and dst_packets (0.0017). Other important features included total_packets (0.0013), dst_bytes (0.0007), total_bytes (0.0005), and src_bytes (0.0005). The findings indicate that packet length, packet counts, and byte-related features strongly influenced the model's predictions in the target domain.

4.4.2. LIME Results

This subsection presents the LIME based explanations of the model's predictions. The results provide local explanations by showing the features that contributed most to individual classification decisions, thereby helping to clarify why specific network traffic instances were classified into particular attack categories. The LIME results are depicted in Figures 14 and 15.

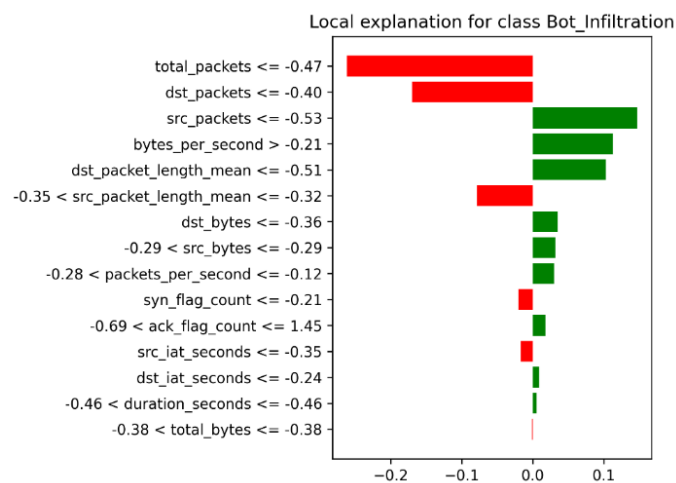


Figure 14. LIME Explanation for CICIDS2017.

Figure 14 presents the LIME explanation for a CICIDS2017 instance classified as Bot_Infiltration. The explanation shows

that total_packets ≤ -0.47 and dst_packets ≤ -0.40 made the strongest negative contributions to the prediction. In contrast,

src_packets > -0.53, bytes_per_second > -0.21, and dst_packet_length_mean > -0.51 provided strong positive contributions toward the Bot_Infiltration class. Other features,

including dst_bytes, src_bytes, and packets_per_second, had smaller effects. The results show how individual traffic features influenced this specific classification decision.

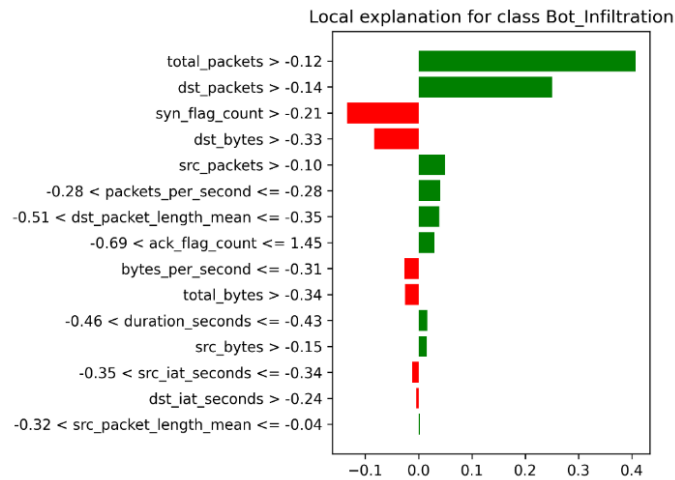


Figure 15. LIME Explanation for UNSW-NB15.

Figure 15 presents the LIME explanation for a UNSW-NB15 instance classified as Bot_Infiltration. The strongest positive contributions came from total_packets > -0.12, with a contribution of approximately 0.40, and dst_packets > -0.14, with about 0.26. In contrast, syn_flag_count > -0.21 contributed negatively by approximately 0.15, while dst_bytes > -0.33 contributed about 0.08 negatively. Other features had relatively small effects. The results show that packet-related features were the main factors influencing this individual Bot_Infiltration prediction.

4.5. Performance Comparison with Previous Studies

This section compares the performance of the proposed framework with results reported in previous studies. The comparison focuses on key performance measures to assess the effectiveness of the proposed approach in relation to existing intrusion detection methods. As illustrated in Figure 16, the proposed framework demonstrates its performance relative to the selected studies.

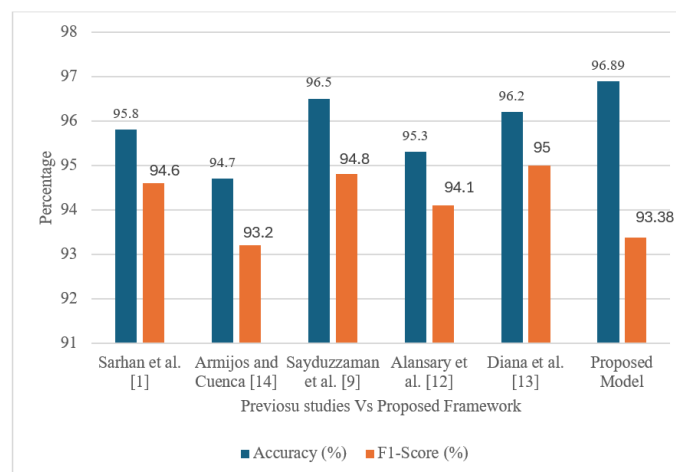


Figure 16. Comparison of Previous Studies.

Figure 16 compares the performance of the proposed framework with previous studies using accuracy and F1-score.

Among the studies included in this comparison, the proposed framework recorded the highest reported accuracy; however,

differences in datasets, experimental settings and evaluation conditions should be considered when interpreting the comparison.

5. Discussion

This section discusses the major findings of the study in relation to the objectives of the proposed framework and relevant previous research. The discussion focuses on the model's learning behavior and in-domain performance, its ability to generalize across different network environments, the contribution of explainable artificial intelligence, and its performance in comparison with existing studies. The section also highlights the significance of the findings, the observed limitations, and possible directions for future research.

5.1. Sequence-Aware Learning and In-Domain Performance

The training and validation results indicate that the proposed framework learned network traffic patterns effectively. For binary detection, training accuracy increased from 92.0% in the first epoch to 99.0% at convergence, while validation accuracy increased from 86.1% to between 95.0% and 97.0%. Training and validation losses decreased, indicating stable convergence. For multiclass classification, training accuracy increased from 92.1% to 99.3%, while validation accuracy reached approximately 96.7%. The small difference between training and validation performance suggests limited overfitting.

These results reflect deep learning models' ability to learn network representations. Hairab et al. [3] demonstrated the usefulness of CNN-based feature learning for zero-day anomaly detection, while Hindy et al. [4] showed the potential of deep learning for zero-day attack detection. The present study extends this approach by representing traffic as sequences of 20 consecutive flow records, helping capture relationships across successive network activities [7, 8].

The combination of 1D-CNN, Bi-LSTM, and Multi-Head Self-Attention provides complementary learning capabilities. CNN extracts local traffic patterns, Bi-LSTM captures temporal dependencies, and attention identifies informative relationships within sequences. This is consistent with Krishnan et al. [11], who demonstrated Bi-LSTM and Multi-Head Attention for zero-day attack detection.

In-domain evaluation achieved 96.89% accuracy, 97.45% precision, 89.65% recall, 93.38% F1-score, 0.9961 ROC-AUC, and 0.9379 MCC for binary detection. Multiclass classification achieved 97.0% accuracy, 96.9% precision, 97.0% recall, 96.8% F1-score, 0.995 ROC-AUC, and 0.949 MCC. However, all 8 Bot_Infiltration instances were misclassified as benign, highlighting the challenge of rare attack categories [11]. Thus, results should be interpreted alongside class-specific performance.

5.2. Cross-Domain Generalization

Cross-domain generalization is an important aspect of this study. The framework was trained on CICIDS2017 and then applied directly to the independent UNSW-NB15 test domain without retraining, fine-tuning, or parameter optimization. This provides a more demanding assessment of whether the learned traffic representation remains useful in a different network environment.

In the UNSW-NB15 binary test set, the model correctly classified 2,146 benign and 4,415 attack records, while 242 benign records were classified as attacks and 678 attack records as benign. The higher number of misclassified attacks indicates that differences in traffic characteristics can affect the recognition of malicious behaviour.

This finding agrees with Sarhan et al. [1] and Dai et al. [2], who emphasized evaluation using data or attacks not directly represented during model development. Dai et al. [2] further highlighted the challenges associated with unseen attack conditions. These findings are also consistent with concerns about dataset dependence and limited cross-domain validation [2-6].

The multiclass results further demonstrate this challenge. The model correctly classified 5,200 benign, 250 Bot_Infiltration, 350 DoS/DDoS, and 211 Probe/Recon records. However, 500 benign records were classified as Bot_Infiltration, 200 as DoS/DDoS, and 100 as Probe/Recon. These results show that cross-domain generalization remains more difficult than in-domain classification. The findings therefore support the need for independent cross-dataset evaluation when assessing potential zero-day detection capability [2-6].

5.3. Explainability of the Proposed Framework

The integration of SHAP and LIME improves the interpretability of the proposed framework, which is important for understanding deep learning decisions in cybersecurity [9, 10]. SHAP identified important traffic features across both domains. For CICIDS2017, `dst_packet_length_mean` was most influential, followed by `dst_packets`, `src_packet_length_mean`, `dst_bytes`, and `src_packets`. For UNSW-NB15, `src_packet_length_mean` ranked first, followed by `dst_packet_length_mean`, `src_packets`, `dst_packets`, and `total_packets`. These results indicate the importance of packet length, packet counts, and byte-related characteristics, while differences in feature rankings suggest variations across network environments [2, 6].

LIME provided local explanations. For the CICIDS2017 Bot_Infiltration example, `total_packets` and `dst_packets` contributed negatively, while `src_packets`, `bytes_per_second`, and `dst_packet_length_mean` contributed positively. For UNSW-NB15, `total_packets` and `dst_packets` contributed positively, whereas `syn_flag_count` and `dst_bytes` contributed negatively.

These findings align with Sayduzzaman et al. [9] and Krishnan et al. [11] and demonstrate the value of combining explainability with sequence-aware temporal learning and independent cross-domain evaluation.

5.4. Comparison with Previous Studies and Research Implications

The comparison with previous studies shows that the proposed framework achieved 96.89% accuracy, compared with 95.8% reported by Sarhan et al. [1], 94.7% by Armijos and Cuenca [14], 96.5% by Sayduzzaman et al. [9], 95.3% by Alansary et al. [12], and 96.2% by Diana et al. [13]. However, its F1-score of 93.38% was lower than the 94.6%, 93.2%, 94.8%, 94.1%, and 95.0% reported by these studies, respectively. Thus, the framework should not be considered superior across all metrics.

Its contribution lies in combining temporal learning, hybrid deep learning, explainability, and independent cross-domain evaluation. Previous studies indicate that many approaches focus on particular benchmark datasets, with cross-domain validation and explainability less consistently integrated [2-6], [9-14]. The proposed framework addresses these aspects through independent cross-domain testing and SHAP and LIME explanations.

However, the cross-domain results also show that differences in traffic characteristics, attack distributions, and network conditions can affect generalization [2-6]. Future research should therefore consider additional datasets, realistic enterprise traffic, and representative unseen attack scenarios. Broader cross-dataset and live-traffic evaluation may provide stronger evidence of practical applicability while maintaining a clear distinction between direct cross-domain evaluation and approaches involving retraining or adaptation.

5.5. Limitations of the Study

The study has several limitations that should be considered when interpreting the findings. First, the evaluation relies on benchmark datasets rather than live enterprise network traffic. Second, only 50,000 records were sampled from each dataset, and the samples may not represent the full diversity of traffic and attack behaviour contained in the original datasets. Third, the 20-record sequence length was used as a fixed design configuration without a separate sensitivity analysis of alternative sequence lengths. Finally, the cross-domain evaluation demonstrates transferability to an independent dataset but does not constitute direct validation against a confirmed zero-day vulnerability or attack. These limitations provide opportunities for future evaluation using larger datasets, alternative temporal window sizes, realistic enterprise traffic and independently verified unseen attack scenarios.

6. Conclusions

This study presented an explainable sequence-aware deep learning framework for potential zero-day attack detection and cross-domain generalization in enterprise network intrusion detection. The framework combines 20-flow temporal sequence construction with 1D-CNN, Bi-LSTM, and four-head

Multi-Head Self-Attention to learn local traffic characteristics, temporal dependencies, and important relationships within network traffic. SHAP and LIME were also incorporated to provide global and local explanations of model predictions.

The framework achieved strong in-domain performance on CICIDS2017, with 96.89% accuracy for binary detection and 97.0% accuracy for multiclass classification. The independent cross-domain evaluation showed 87.70% accuracy for binary detection and 80.35% for multiclass classification when the CICIDS2017-trained model was applied directly to UNSW-NB15 without retraining or fine-tuning. The performance reduction highlights the challenge of transferring learned traffic representations across different network environments.

The SHAP and LIME analyses further showed how important traffic features contributed to model predictions, providing useful information for interpreting detection decisions.

The findings demonstrate the potential of combining temporal learning, hybrid deep learning, explainable AI, and independent cross-domain evaluation for enterprise intrusion detection. However, the cross-domain results should be interpreted as evidence of transferability and potential zero-day detection capability rather than direct confirmation of detection of a verified zero-day attack. Future work should consider additional datasets, alternative sequence lengths, realistic enterprise network traffic, and more representative unseen attack scenarios.

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
Bi-LSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
IDS	Intrusion Detection System
IoT	Internet of Things
LSTM	Long Short-Term Memory
ML	Machine Learning
ReLU	Rectified Linear Unit
SMOTE	Synthetic Minority Oversampling Technique
XAI	Explainable Artificial Intelligence

Author Contributions

Uchechukwu Samuel Nwankwo: Conceptualization, Data curation, Resources, Writing – original draft, Writing – review & editing

Obi Chukwuemeka Nwokonkwo: Investigation, Methodology, Supervision, Validation

Charles Ikerionwu: Formal Analysis, Investigation, Methodology, Supervision, Validation

Udoka Felista Eze: Formal Analysis, Supervision
Adetokunbo MacGregor John-Otumu: Conceptualization, Data curation, Investigation, Methodology, Validation

Data Availability Statement

The data that supports the findings of this study can be found at: <https://www.unb.ca/cic/datasets/ids-2017.html> and <https://research.unsw.edu.au/projects/unsw-nb15-dataset>

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] M. Sarhan, S. Layeghy, M. Gallagher, and M. Portmann, "From zero-shot machine learning to zero-day attack detection," *arXiv preprint arXiv: 2109.14868*, 2021.
- [2] Z. Dai, L. Y. Por, Y.-L. Chen, J. Yang, C. S. Ku, R. Alizadehsani, and P. Plawiak, "An intrusion detection model to detect zero-day attacks in unseen data using machine learning," *PLOS ONE*, vol. 19, no. 9, Art. no. e0308469, 2024, <https://doi.org/10.1371/journal.pone.0308469>
- [3] B. I. Hairab, H. K. Aslan, M. S. Elsayed, A. D. Jureut, and M. A. Azer, "Anomaly detection of zero-day attacks based on CNN and regularization techniques," *Electronics*, vol. 12, no. 3, Art. no. 573, 2023, <https://doi.org/10.3390/electronics12030573>
- [4] H. Hindy, R. Atkinson, C. Tachtatzis, J.-N. Colin, E. Bayne, and X. Bellekens, "Utilising deep learning techniques for effective zero-day attack detection," *Electronics*, vol. 9, no. 10, Art. no. 1684, 2020, <https://doi.org/10.3390/electronics9101684>
- [5] C. Brunner, A. Kő, and S. Fodor, "An autoencoder-enhanced stacking neural network model for increasing the performance of intrusion detection," *Journal of Applied Security Research*, vol. 12, no. 2, pp. 149-163, 2022, <https://doi.org/10.2478/jaiscr-2022-0010>
- [6] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, Art. no. 18, 2021, <https://doi.org/10.1186/s42400-021-00077-7>
- [7] N. Aljawabrah, N. Y. Al-Tamimi, A. Alsarhan, M. Aljamal, B. S. Khassawneh, S. A. Alshammari, N. H. Alshammari, and K. H. Alnafisah, "A simulation-driven cybersecurity framework for detecting novel multi-stage attacks in cyber-physical smart infrastructure," *Network*, vol. 6, no. 3, Art. no. 42, 2026, <https://doi.org/10.3390/network6030042>
- [8] V. Priya M. K., S. Sivabalan, H. Anila Glory, M. Aggarwal, and S. Sriram V. S., "Advanced persistent threat detection through sequential analysis of network patterns with graph based learning approach," *Scientific Reports*, vol. 16, Art. no. 19998, 2026, <https://doi.org/10.1038/s41598-026-42756-w>
- [9] M. Sayduzzaman, J. T. Tamanna, D. Kundu, and T. Rahman, "Interoperability and explicable AI-based zero-day attacks detection process in smart community," *arXiv preprint arXiv: 2408.02921*, 2024.
- [10] M. A. Alparacha, S. U. Jamil, K. Shahzad, M. A. Khan, and A. Rasheed, "Leveraging AI for network threat detection—A conceptual overview," *Electronics*, vol. 13, Art. no. 4611, 2024, <https://doi.org/10.3390/electronics134611>
- [11] D. Krishnan, S. Singh, and V. Sugumaran, "Explainable AI for zero-day attack detection in IoT networks using attention fusion model," *Research Square*, 2025, <https://doi.org/10.21203/rs.3.rs-5436116/v1>
- [12] S. A. Alansary, S. M. Ayyad, F. M. Talaat, and M. M. Saafan, "Emerging AI threats in cybercrime: A review of zero-day attacks via machine, deep, and federated learning," *Knowledge and Information Systems*, vol. 67, pp. 10951-10987, 2025, <https://doi.org/10.1007/s10115-025-02556-6>
- [13] L. Diana, P. Dini, and D. Paolini, "Overview on intrusion detection systems for computers networking security," *Computers*, vol. 14, no. 3, Art. no. 87, 2025, <https://doi.org/10.3390/computers14030087>
- [14] A. Armijos and E. Cuenca, "Zero-day attacks: Review of the methods used based on intrusion detection and prevention systems," *2023 IEEE Colombian Caribbean Conference (C3)*, pp. 1-6, 2023.

Research Field

Uchechukwu Samuel Nwankwo: Cybersecurity, Machine Learning, Expert Systems

Obi Chukwuemeka Nwokonkwo: IT Project Management, Digital System, IT Automation

Charles Ikerionwu: Software Process Models, Machine Learning, Deep Learning, Artificial Intelligence

Udoka Feliata Eze: Database Systems, Data Mining, Data Science, Artificial Intelligence

Adetokunbo MacGregor John-Otumu: Machine Learning, Deep Learning, Computer Vision, NLP, Intelligent Systems