Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

XDeep-IDS: An Explainable and GA-Optimized Deep Learning Framework for IoT Intrusion Detection

Tibebu Bekele Shana

tibebu_2121cs21@iitp.ac.in

Indian Institute of Technology Patna https://orcid.org/0009-0009-7032-1014

Krish Vaish

Indian Institute of Technology Patna

Samrat Mondal

Indian Institute of Technology Patna https://orcid.org/0000-0002-2159-3410

Research Article

Keywords: Cybersecurity, XAI, Intrusion detection, Deep Learning, High-dimensional data, Imbalanced data

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11819523/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

XDeep-IDS: An Explainable and GA-Optimized Deep Learning Framework for IoT Intrusion Detection

Tibebu Bekele Shana[1*], Krish Vaish[2] and Samrat Mondal[1]

[1]Department of Computer Science and Engineering.
[2]Department of Chemical and Biochemical Engineering.
[1,2]Indian Institute of Technology Patna, Bihta, Patna, 801106, Bihar, India.

*Corresponding author(s). E-mail(s): tibebu_2121cs21@iitp.ac.in;
Contributing authors: krish_2301cb58@iitp.ac.in; samrat@iitp.ac.in;

Abstract

With the significant growth, diversification, and interconnection of networks, there is an increasing demand for intelligent, robust intrusion detection system (IDS) solutions to defend against increasingly complex and sophisticated cyber-attacks. Despite that, existing IDS techniques suffer from issues like poor interpretability, high computing cost, high-dimensional network traffic, class imbalance, and limited detection accuracy. To address those challenges, XDeep-IDS is proposed in this paper, which is an explainable Deep-Learning (DL) based framework that combines Genetic Algorithm (GA)-based feature selection and Synthetic Minority Oversampling Technique with Edited Nearest Neighbors (SMOTE-ENN) applying for class balancing to improve minority-class detection performance. The experimental framework compares five classifiers: LightGBM (LGBM), K-nearest neighbour (KNN), Deep Neural Network (DNN), Multilayer Perceptron (MLP), and Bidirectional Long Short-Term Memory (BiLSTM). As the BiLSTM outperforms the other proposed models in general, its sequence-based representation and classification results are emphasized in the proposed framework. Five-fold cross validation is used to ensure a robust evaluation and avoid over-fitting. Additionally, the model-agnostic explainable artificial intelligence (XAI) techniques are applied to enhance the interpretability and transparency of the model. We also provide different ablation studies to verify the proposed method. The results show that XDeep-IDS can achieve accuracies of $97.43 \pm 0.21\%$ and $99.36 \pm 0.12\%$ on datasets of NF-BoT-IoT and CIC-IoT23, respectively. The proposed model demonstrates higher performance compared with state-of-the-art approaches and existing benchmark models.

Keywords: Cybersecurity, XAI, Intrusion detection, Deep Learning, High-dimensional data, Imbalanced data

1 Introduction

Today, business activities and societal functions rely more on Information and Communication Technology (ICT) platforms that offer key services, data sharing, and digital communication in different sectors. As dependence on ICT infrastructures has increased, they have become a lucrative target for cybercriminals, and the frequency and sophistication of cyberattacks has significantly increased. This has turned security and resilience of networked systems into a basic requirement for organizations across the globe. Intrusion Detection Systems (IDSs) have therefore been widely adopted as an essential component of cybersecurity frameworks, enabling the continuous monitoring of network activities and the timely identification of malicious behavior and security breaches [1–3].

Depending on their detection methods, IDS are often divided into three groups: anomaly-based, signature-based, and hybrid IDS. By matching the observed activity to known attack signatures, signature-based IDS identify known attacks. As a result, they are highly successful at identifying attacks that have previously been reported. However, their reliance on established attack patterns restricts their capacity to detect novel intrusions and zero-day exploits. On the other hand, anomaly-based IDS create a model of typical network behavior and recognize variations from this baseline as possible intrusions. Anomaly-based systems are a possible answer to the problems of contemporary cybersecurity environments because of their capacity to identify novel, unidentified, and changing threats. Hybrid IDS combine anomaly-based and signature-based detection methods to take advantage of their complementing advantages. This method makes it possible to identify both known cyberattacks and threats that have never been observed before [4–6].

Conventional IDS frequently fail to identify complex and dynamic cyberthreats. Consequently, ML models have been employed in numerous research studies to increase detection efficacy [7]. For instance, Injadat et al. [8] presented an anomaly detection framework utilizing Bayesian optimization to tune ML models such as KNN, Random Forest (RF), and Support Vector Machines (SVM). Salo et al. [9] proposed a dimensionality reduction technique that mixes Information Gain (IG) with Principal Component Analysis (PCA). Likewise, Devaraj et al. [10] added correlated feature selection to a cybersecurity model. Yet persistent class imbalance continues to be an obstacle to IDS performance; fixing this problem is essential so that models do not favor majority classes and so that IDS can reliably detect rare critical attack patterns [11].

Studies reveal several research gaps, including challenges in managing high-dimensional data due to the lack of effective feature selection techniques, persistent class imbalance issues, difficulties in optimal model selection, and insufficient focus on the efficiency of the model on IDS. Moreover, XAI techniques were not incorporated for an effective and enhanced IDS approach for better performance, transparency, and trustworthiness. These gaps reduce accurate evaluation of the real-world applicability and scalability of IDS solutions. To address the aforementioned problems, this work proposed XDeep-IDS, an enhanced approach to intrusion detection by integrating the GA for feature selection with SMOTE-ENN for class balancing. The combined GA and SMOTE-ENN framework is ensembled with the BiLSTM model to achieve accurate classification between intrusion and normal traffic. To enhance the transparency and reliability of IoT intrusion detection, we employed two model-agnostic XAI techniques, SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic

Explanations (LIME). The proposed model outperforms baseline approaches in terms of various comparison metrics. The main contributions of this study are summarized below:

- Employed a GA-based approach for feature selection to identify the most relevant and informative features.
- Utilized SMOTE-ENN-based data augmentation to effectively address class imbalance and improve the representation of minority classes.
- Evaluated multiple models (LGBM, KNN, DNN, MLP, and BiLSTM), achieving the highest accuracy for IoT intrusion detection with BiLSTM.
- Incorporated SHAP and LIME to provide clear global and local model interpretability.
- XDeep-IDS Framework: Designed an end-to-end explainable intrusion detection framework that outperforms standard models across key metrics.

The rest of the paper will be organized as follows: Section 2 reviews the literature relevant to this paper. The methodology will be outlined in Section 3, and the experimental setup will be described in Section 4. The empirical results will be discussed in Section 5. The conclusions will be drawn and future work will be described in Section 6.

2 Related work

This section presents a systematic literature review of current approaches to IDS, with the aim of providing a comprehensive overview of existing approaches, techniques, and research developments in this field. Table 1 presents comparative summary of XAI based-IDS studies including methods, datasets, contributions and limitations. Qazi et al. [12] presented HDL-NIDS, which is a DL approach involving the use of CNN and RNN to detect attacks in the network using the CICIDS-2018 dataset that includes both the normal and attack classes. However, to solve the issue of class imbalance, the authors did not mention any specific technique. The CNN automatically extracted traffic features; however, no separate feature selection technique was implemented. EAI methods like SHAP or LIME have not been applied to explain the model's predictions. For intrusion detection, Hnamte and Hussain [13] presented DCNNBiLSTM, which combines Deep CNN and BiLSTM. The Edge_IIoT and CICIDS2018 datasets were used for testing. A number of unnecessary features were eliminated during preprocessing, and SMOTE was used to solve class imbalance; nevertheless, no specific feature selection technique was used. The study did not use any XAI techniques, such as SHAP or LIME.

Wu et al. [14] introduce RTIDS, a reliable transformer-based IDS for identifying network threats. On benchmark IDS datasets, the model showed competitive detection performance against existing ML and DL methods. There was no mention of any particular balancing method, like SMOTE or ADASYN. The Transformer performs feature representation learning, but no dedicated feature selection method was implemented. The model's predictions were not explained by any XAI technique, such as SHAP or LIME. Manocchio et al. [15] proposed the FlowTransformer, which is a transformer-based IDS that used BERT and GPT-2 and was evaluated on the CSE-CIC-IDS2018, UNSW-NB15, and NSL-KDD datasets. The study achieved over 95% accuracy with different input encodings and classification heads. Balancing techniques were not utilized to overcome class imbalance. No dedicated feature selection method was implemented, and No XAI method like SHAP or LIME was applied. The study showed that Transformer models can effectively detect network intrusions while reducing

Table 1: Comparative summary of XAI based-IDS studies including methods, datasets, contributions and limitations.

Reference	Models Used	Dataset Used	Limitation
2019 [17]	CNN with SMOTE-ENN	NSL-KDD	The reported accuracy was 83.31%. Dedicated feature selection and XAI were not used in the study.
2022 [18]	DT, RF, SVM, voting classifier	CICIDS-2017	Class-balancing techniques were not used to address class imbalance, although LIME was applied to explain predictions.
2022 [19]	Extra Trees classifier	UNSW-NB15, BoT-IoT, ToN-IoT, CSE-CIC-IDS2018	The study focused on standardized NetFlow feature representations rather than a new feature-selection method. Class imbalance was not addressed, and XAI was not applied.
2023 [12]	CNN and RNN (HDLNIDS)	CICIDS-2018	No specific class-balancing or dedicated feature-selection technique was reported. SHAP/LIME-based explanations were not incorporated.
2023 [13]	Deep CNN and BiLSTM (DCNNBiLSTM)	Edge_IIoT and CICIDS-2018	Although SMOTE was applied and irrelevant features were removed during preprocessing, no dedicated feature-selection method or XAI technique was used.
2023 [16]	Naive Bayes, SVM, RF (IIDPS)	NSL-KDD	Class imbalance was not addressed; a specific feature-selection algorithm and XAI method were not reported.
2024 [15]	FlowTransformer (BERT and GPT-2)	CSE-CIC-IDS2018, UNSW-NB15, NSL-KDD	No data-balancing technique, dedicated feature-selection method, or SHAP/LIME-based explanation was reported.
2024 [20]	Logistic Regression, SVM, DT, RF	UNSW-NB15	Correlation analysis and random sampling were used for feature selection, but no balancing technique or XAI method was reported.
2024 [21]	ADA, DNN, LightGBM, SVM, MLP, RF, KNN (XAI-IDS)	CICIDS-2017, NSL-KDD, RoEduNet-SIMARGL2021	No data-balancing technique was employed, although SHAP-based feature selection and SHAP/LIME explanations were used.
2025 [22]	Customized 1-D CNN, DNN, pre-trained TabNet	Seven ToN-IoT datasets	No balancing technique or dedicated feature-selection algorithm was reported, although LIME and SHAP were used for explanations.
2025 [23]	RF	CSE-CIC-IDS2018 & CIC-IoT-23	No data-balancing technique or separate feature-selection algorithm was reported, although SHAP was used for credibility assessment and explanations.

model size by more than 50% with suitable configurations. The proposed Internal Intrusion Detection and Protection System (IIDPS) detects system call intrusions through the use of Naïve Bayes, SVM, and RF classification algorithms [16]. The NSL-KDD dataset was used in the study. No balancing strategies were applied, so the study did not address issues of class imbalance. The suggested solution mentions no feature selection process and applies no XAI technique.

Using the UNSW-NB15 dataset, More et al. [20] assessed logistic regression, SVM, DT, and RF. No balancing technique or XAI technique, such as SHAP or LIME, was integrated in this study. Random sampling and correlation analysis were used for feature selection. For network intrusion detection, Sarhan et al. [19] presented standardized NetFlow-based feature sets with 12 and 43 features. They converted UNSW-NB15, BoT-IoT, ToN-IoT, and CSE-CIC-IDS2018 into the proposed feature format and evaluated them using an Extra Trees classifier. The study focused more on testing the effectiveness of a standard feature representation approach instead of applying a new feature selection approach, and the issue of class imbalance was not handled in the study. No application of explainable AI approaches such as SHAP or LIME took place. The 43-feature NetFlow feature set performed better than the 12-feature set for all classification tasks.

Zhang et al. [17] used SMOTE-ENN to mitigate class imbalance issues using the NSL-KDD dataset to enhance IDS. The proposed SMOTE-ENN-based CNN IDS model achieves an accuracy of 83.31%. Behera et al. proposed DDoS detection using KDDCup99 and CIC IoT-2023 datasets and used PCA for optimal feature selection and SMOTE-ENN to handle class imbalance [24]. In addition, Pramanick et al. [25] used UNSW-NB15 and NSL-KDD datasets to enhance IDS in the proposed approach and SMOTE-ENN to overcome class imbalance issues. Researchers used GA for feature selection to enhance IDS, using differing datasets, such as, KDD Cup, CIRA-CIC-DOHBrw-2020, UNSW-NB15, and Bot-IoT [26–28]. GA is the good choice for avoiding premature convergence and local optima during optimal feature selection compared to other techniques [29]. Almomani [30] evaluated different feature selection techniques on the UNSW-NB15 dataset; the GA shows good results in terms of true positive rate and false negative rate.

Alabbadi and Bajaber [22] proposed a DL-based IDS using customized 1-D CNN, DNN, and pre-trained TabNet models, evaluated on seven TON_IoT datasets. No balancing technique was used, and no dedicated feature-selection algorithm was implemented. XAI was explicitly incorporated using LIME and SHAP to identify the features influencing predictions. Patil et al. [18] presented an ensemble-based IDS using DT, RF, and SVM, followed by a voting classifier, and evaluated it on the CICIDS-2017 dataset. Balancing techniques were not used to mitigate class imbalance issues. The XAI was explicitly applied by the use of LIME to explain the predictions made. With LIME offering more responsive and explanation-friendly interpretations, the proposed voting ensemble attained 96.25% accuracy.

Hariharan et al. [31] used the NSL-KDD and Kaggle IDS datasets to assess RF, XGBoost, and LGBM for IDS. The study did not employ balancing procedures. The analysis concentrated on the top 15 significant features, and feature selection was carried out using permutation importance to find the most influential features. For both global and local explanations, XAI was widely used with SHAP, LIME, permutation importance, and CIU. These approaches were examined in the study according to stability, consistency, and explanation correctness. Arreche et al. [21] proposed XAI-IDS, an explainable IDS framework evaluating ADA, DNN, LGMB, SVM, MLP, RF, and KNN on CICIDS-2017, NSL-KDD, and RoEduNet-SIMARGL2021. No data-balancing techniques, such as SMOTE or ADASYN, were employed. Feature selection was implemented using SHAP-based feature importance, with the top 15 and top 5 features analyzed. XAI was explicitly incorporated using SHAP and LIME for global and local explanations. The models achieved near 99% accuracy on CICIDS-2017 and RoEduNet-SIMARGL2021, while XAI-based feature selection outperformed information gain and K-best in 11 of 14 model comparisons.

Wali et al. [23] proposed a credibility-aware IDS combining an RF classifier, SHAP-based credibility assessment, behavioral classifiers, and a Transformer-based payload inspection module. The framework was evaluated on the CSE-CIC-IDS2018 and CIC-IoT-23 datasets and tested against black-box adversarial attacks, including HopSkipJump and ZOO. The study did not use any data-balancing techniques, and no separate feature selection algorithm was reported. Using SHAP, XAI was specifically constructed, comparing local and global feature attributions to find predictions that were suspect. The framework maintained high clean data accuracy and improved robustness against adversarial attacks compared with existing defences. Using the Honeypot and NSL-KDD datasets, Sivamohan and Sridhar [32] presented a Bi-LSTM XAI framework for Industry 4.0 intrusion detection. Class imbalance was not taken into

consideration while using Krill Herd Optimization (KHO) for feature selection. In order to interpret the model's predictions, XAI was integrated using SHAP and LIME. The suggested architecture demonstrated efficient intrusion detection for Industry 4.0 networks, with 98.2% classification accuracy.

In order to find significant intrusion features, Arreche et al. [33] introduced XAI-based feature selection methods for network IDS. Five XAI-based feature-selection algorithms were presented in the study and contrasted with traditional methods. Feature selection is the main focus of the study, and XAI approaches are utilized to find the most significant qualities and enhance model performance. There was no indication that class-balancing strategies were applied. When compared to conventional feature selection techniques, the study demonstrates that XAI-based feature selection generally enhanced the performance of the assessed AI models. An IDS framework for SDN-based IoT networks was proposed, Tserenkhuu et al. [34], utilizing many DL models, such as CNN, with XAI-based feature selection methods and domain-constrained features. The InSDN and X-IIoTID datasets were used to test the system. Class-imbalance mitigation strategies were not used nor reported in the study. Feature selection was performed using SHAP and RF feature importance, which were found to provide consistent feature subsets across the DL models. XAI was therefore explicitly incorporated, with SHAP being particularly effective. CNN achieved 99.9% accuracy on InSDN and 98% on X-IIoTID for multiclass classification.

GA is an evolutionary optimization technique based on the principles of natural selection and biological evolution. It begins with a set of candidate solutions and assesses their quality using a predefined fitness function. Then the candidate solutions are successively improved by means of selection, crossover, and mutation in successive generations until a stopping criterion is reached. In IDS, GA is used for feature selection by exploring different combinations of features and identifying those that contribute most to classification performance. By removing irrelevant and redundant features, GA can reduce the dimensionality of the dataset and computational cost while preserving the features that are most useful for accurate intrusion detection [26, 27, 35].

3 Proposed Methodology

This section presents the overall methodology adopted in this study, as illustrated in Figure 1. The proposed framework comprises data preprocessing and scaling, GA-based feature selection, SMOTE-ENN-based class balancing, sequence construction, and DL-based classification. The NF-BoT-IoT and CIC-IoT23 datasets are used to assess the framework. Stratified five-fold cross-validation is employed to evaluate model performance. In each iteration, four folds are used for training and the remaining fold is reserved for validation. Within each iteration, preprocessing and scaling transformations are fitted using only the training data. GA-based feature selection and SMOTE-ENN class balancing are independently performed on the training data, while the learned transformations and selected features are applied to the corresponding validation fold without oversampling. Fixed-length sequences of ten observations are constructed for sequence-based experiments, while conventional feature representations are used for non-sequence experiments. Five classification models, namely LGBM, KNN, DNN, MLP, and BiLSTM, are comparatively evaluated. Their hyperparameters are optimized using random search with cross-validation restricted to the training data of each outer fold.

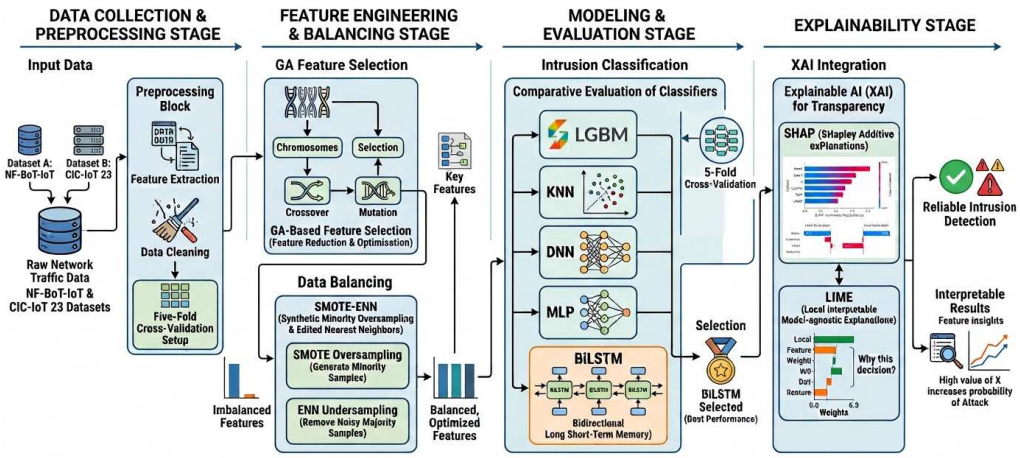


Fig. 1: End-to-end architecture of the proposed intrusion detection framework, illustrating data preprocessing, GA-based feature optimization, SMOTE-ENN class balancing, comparative classifier evaluation, and XAI integration

Among these models, BiLSTM achieves the highest overall classification performance and is therefore selected for further analysis, particularly its sequence-based representation, ablation performance, and feature-level interpretability using SHAP and LIME. The procedure is repeated five times and in each iteration one fold acts as validation set and the performance obtained in five iterations is reported the mean and standard deviation.

3.1 Data Preprocessing

The process of transforming raw data into a clean, consistent, and appropriate format for ML and DL models. Let the raw multivariate network dataset be defined as $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in R^D$ represents a D-dimensional feature vector and $y_i \in \{1, 2, \ldots, K\}$ denotes the multiclass attack label ($K > 2$). The dataset undergoes initial cleaning and feature engineering to handle missing values, infinite entries, and categorical encoding. Min-max normalization scales each continuous attribute to the interval $[0, 1]$ to prevent features with larger numeric scales from controlling model convergence:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

In the dataset, X_{ij} represents the initial value of the feature j for the i-th sample, $X_{\min}$ represents the lowest and $X_{\max}$ represents the highest value of the feature j across all samples, and X'_{ij} represents the normalized value X_{ij} scaled to a range of [0,1].

3.2 GA-Based Feature Selection

Redundant and unnecessary features are frequently present in high-dimensional network data, which raises computational complexity and reduces classification accuracy. We employ a GA to discover the optimal subset of features $c \subset D$ ($|c| = D_{\text{best}}$). A binary chromosome vector

$c \in \{0,1\}^D$ represents the feature selection state. The population evolves over successive generations using tournament selection, two-point crossover, and bit-flip mutation. The multi-objective fitness function $F(c)$ balances the macro-averaged F1-score $F1_{\text{macro}}(c)$ with the feature reduction ratio:

$$F(c) = 0.9F1_{\text{macro}}(c) + 0.1\left(1 - \frac{|c|_1}{D}\right) \tag{2}$$

Where, $F(c)$ denotes the overall fitness value of chromosome c, where a higher value indicates a better feature subset. The chromosome c is a binary vector, with $c_j = 1$ indicating that the j-th feature is selected and $c_j = 0$ otherwise. $F1_{\text{macro}}(c)$ denotes the macro-averaged F1-score achieved using the feature subset represented by c. The term $|c|_1 = \sum_{j=1}^{D} c_j$ represents the number of selected features, while D denotes the total number of available features before GA-based feature selection. The coefficients 0.9 and 0.1 assign greater importance to classification performance and a secondary importance to feature reduction, respectively. The GA was configured with a population size of 50 and a maximum of 50 generations, employing tournament selection with a tournament size of 3. Two-point crossover was applied with a probability of 0.80, while mutation was performed with a probability of 0.10 and a mutation rate of 0.02. Elitism was set to 2, and early stopping after 10 generations without improvement was employed to reduce unnecessary computational overhead. Reproducibility was ensured by using a random seed of 42.

3.3 Data Balancing via SMOTE-ENN

Minority attack classes are substantially underrepresented in network intrusion datasets due to a large class imbalance that occurs naturally. Therefore, the hybrid SMOTE-ENN sampling technique is used to address this:

1. **Synthetic Minority Over-sampling Technique (SMOTE):** Uses interpolation between k-nearest neighbors ($k = 5$) in feature space to create synthetic instances for underrepresented minority classes, producing a class-balanced dataset D_b:

$$x_{\text{new}} = x_i + \lambda(x_{zi} - x_i), \quad \lambda \sim U(0,1) \tag{3}$$

2. **Edited Nearest Neighbors (ENN):** Removes noisy, overlapping, and incorrectly labeled samples from D_b whose class labels are different from those of most of their closest neighbors. As a result, the dataset D_{new} of size $N_{\text{new}} \times D_{\text{best}}$ is clean and well-separated.

Where x_{new} a synthetic sample generated, x_i a reference minority instance, x_{zi} selected k-nearest neighbor, $(x_{zi} - x_i)$ feature difference vector, λ random interpolation weight, and $\lambda \sim U(0,1)$ uniform distribution constraint over $[0,1]$

3.4 Machine Learning and Deep Learning Models

We assess the proposed framework's efficacy using both ML and DL models, allowing for a thorough comparison of their intrusion detection capacities.

3.4.1 Light Gradient Boosting Machine (LGBM)

LGBM is an effective gradient boosting framework for regression and classification. It uses gradient-based optimization to successively construct decision trees in order to minimize a loss function. Faster training and good predicted accuracy are made possible by its leaf-wise tree growth technique and depth restrictions.

3.4.2 K-Nearest Neighbors (KNN)

KNN is a supervised machine learning algorithm that classifies samples based on the majority class among their k nearest neighbors in the feature space. Using distance-based similarity, KNN distinguishes benign network traffic from various cyberattacks in multiclass intrusion detection.

3.4.3 Multilayer Perceptron (MLP)

MLP is a feedforward neural network consisting of an input layer, one or more hidden layers, and an output layer. Through weighted transformations and nonlinear activation functions, it learns complex relationships in network traffic for multiclass classification.

3.4.4 Deep Neural Network (DNN)

A DNN is a neural network with multiple hidden layers that learns hierarchical representations and complex nonlinear relationships in network traffic. These capabilities enable it to distinguish benign activities from different cyberattacks in multiclass intrusion detection.

3.4.5 BiLSTM Network Architecture

Using a sliding window, the chosen features are arranged into overlapping sequences. To identify temporal relationships and produce bidirectional feature representations for multiclass intrusion detection, a BiLSTM network examines each sequence both forward and backward.

1. **Temporal Sequence Framing:** To capture temporal dependencies in network traffic, the feature matrix D_{new} is transformed into overlapping sequence samples using a sliding-window strategy with a fixed sequence length $L = 10$ and stride $s = 1$. The resulting input tensor is represented as

$$T \in R^{B \times L \times D_{best}} \tag{4}$$

   where B denotes the number of sequences in a batch, $L = 10$ is the sequence length, and D_{best} is the number of GA-selected features. Each sequence is assigned a corresponding target label y_{new} according to the adopted labeling strategy.

2. **BiLSTM Network Architecture:** The resulting three-dimensional input tensors are processed by a BiLSTM network. Unlike a conventional LSTM, which processes a sequence in one direction, BiLSTM captures contextual information by processing each input sequence in both forward and backward directions:

$$\vec{h}_t = \text{LSTM}_{\text{fwd}}(x_t, \vec{h}_{t-1}) \tag{5}$$

$$\overleftarrow{h}_t = \text{LSTM}_{\text{bwd}}(x_t, \overleftarrow{h}_{t+1}) \tag{6}$$

$$H_t = \left[\vec{h}_t \mid \overleftarrow{h}_t\right] \tag{7}$$

$$H \in R^{B \times L \times 2H_{\text{hidden}}} \tag{8}$$

Where x_t denotes the input feature vector at time step t, $\vec{h}_t$ and $\overleftarrow{h}_t$ represent the forward and backward hidden states, respectively, and H_{hidden} denotes the number of hidden units in each direction. The concatenated representation $H_t = \left[\vec{h}_t \mid \overleftarrow{h}_t\right]$ captures comprehensive bidirectional contextual dependencies. Finally, the complete BiLSTM output sequence is represented by the tensor $H \in R^{B \times L \times 2H_{\text{hidden}}}$, spanning a batch size B, sequence length L, and twice the unidirectional hidden unit dimension.

3.5 Multiclass Classification and XAI

The concatenated BiLSTM representations are passed through a fully connected (FC) layer with dropout regularization to mitigate overfitting. The softmax activation function converts the output logits into a probability distribution over the K intrusion classes:

$$\hat{y}_{i,k} = \frac{\exp(z_{i,k})}{\sum_{j=1}^{K} \exp(z_{i,j})} \tag{9}$$

where $\hat{y}_{i,k}$ denotes the predicted probability that the i-th input sequence belongs to class k, $z_{i,k}$ represents the corresponding output logit, and K is the total number of intrusion classes. The BiLSTM is trained by minimizing the categorical cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{B}\sum_{k=1}^{K} y_{i,k} \log(\hat{y}_{i,k}) \tag{10}$$

where B denotes the batch size, and $y_{i,k} \in \{0, 1\}$ is the one-hot encoded ground-truth indicator, equal to 1 when sequence i belongs to class k and 0 otherwise. To enhance interpretability, SHAP and LIME are applied to the trained BiLSTM model to explain global feature importance and individual predictions, respectively.

4 Experiments

4.1 Experimental Configuration

Python 3.11 served as the foundation for the experiment, and TensorFlow 2.4.1 was used to create the ML and DL models. For our research, we manage data, find features, and present the findings using programs like NumPy, pandas, matplotlib, and Scikit-Learn.

4.2 Hyper-parameter optimization

The models' hyperparameters were optimized using random search with cross-validation. The BiLSTM, MLP, KNN, LGBM, and DNN models were independently fine-tuned on both datasets to identify their respective optimal configurations. Among the evaluated models, BiLSTM achieved the highest classification performance under its optimal hyperparameter

Table 2: Hyperparameter search spaces and optimal configurations of the classification models.

Model	Hyperparameter search space	Optimal Configuration (NF-BoT-IoT)	Optimal Configuration (CIC-IoT23)
BiLSTM	Units: $\{64, 128, 256\}$	Units: 128	Units: 256
	Dropout: $\{0.2, 0.3, 0.5\}$	Dropout: 0.3	Dropout: 0.2
	Batch size: $\{32, 64, 128\}$	Batch size: 64	Batch size: 64
	Learning rate: $\{0.001, 0.0005, 0.0001\}$	Learning rate: 0.001	Learning rate: 0.0005
	Optimizer: {Adam, SGD, RMSprop}	Optimizer: Adam	Optimizer: Adam
	Epochs: $\{30, 50, 100\}$	Epochs: 50	Epochs: 50

Table 3: Input configurations for non-sequence and sequence-based BiLSTM experiments.

Dataset	Configuration	Sequence length	Input shape
NF-BoT-IoT	Without sequence	1	$(6,)$
NF-BoT-IoT	With sequence	10	$(10, 6)$
CIC-IoT23	Without sequence	1	$(25,)$
CIC-IoT23	With sequence	10	$(10, 25)$

configuration; therefore, its parameters are reported in detail. Table 2 presents the predefined hyperparameter search spaces and the corresponding optimal configurations of the evaluated classifiers, thereby supporting reproducibility. Table 3 presents the input configurations used with and without sequence during experiments, distinguishing conventional non-sequence inputs from the proposed sequence-based representation. Specifically, it shows how the GA-selected feature subsets were transformed into fixed-length sequences for BiLSTM processing. The CICIoT23 dataset is accessed from https://www.unb.ca/cic/datasets/iotdataset-2023.html. The NF-BoT-IoT dataset is accessed from https://espace.library.uq.edu.au/view/UQ:62e6d80.

4.3 Evaluation metrics

In this research work several evaluation metrics have been used to measure the performance of the detection model. Evaluation metrics include precision, recall, F1-score, alarm rate (FAR), and the ROC curve.

5 Results and Discussion

- **Dimensionality reduction**: The GA selects 25 features from the initial 47 features for the CIC-IoT23 dataset. This results in a reduction in feature dimensionality to 46.80% of its original size. This reduction in dimensionality results in a significant reduction in computational complexity. In addition, there is a corresponding reduction in processing time. In the case of the NF-BoT-IoT dataset, 6 features are selected from an initial set of 12.

Table 4: Average feature reduction achieved by GA across five-fold cross-validation

Dataset	Original Features	Average Selected Features	Feature Reduction (%)
CIC-IoT23	47	25	46.80
NF-BoT-IoT	12	6	50.00

Table 5: Multiclass classification performance metrics on the analyzed datasets using the BiLSTM model

Algorithm with Datasets	Class	Label	Precision		Recall		F1-score	
			Raw	Proposed XDeep-IDS	Raw	Proposed XDeep-IDS	Raw	Proposed XDeep-IDS
NF-BoT-IoT BiLSTM	Benign	0	93.42	93.86	64.03	94.20	75.98	94.03
	Reconnaissance	1	89.00	95.78	98.44	95.55	93.53	95.67
	DDoS	2	48.17	99.98	29.91	99.67	36.90	99.83
	DoS	3	42.12	99.99	24.50	99.76	30.98	99.88
	Theft	4	74.36	99.87	84.29	99.95	79.02	99.91
CIC-IoT23 BiLSTM	Benign	0	83.47	97.40	89.35	98.15	86.31	97.77
	DDoS	1	99.87	99.63	99.88	99.53	99.88	99.58
	DoS	2	99.60	99.56	99.66	99.63	99.63	99.59
	Malware	3	13.43	99.92	15.11	99.98	14.42	99.98
	Man-in-the-Middle	4	74.47	98.83	69.58	97.74	71.94	98.28
	Mirai Botnet	5	99.89	99.96	99.53	99.96	99.71	99.96
	Reconnaissance	6	70.19	98.77	64.04	98.90	66.97	98.83
	Vulnerability	7	86.84	99.98	89.19	100.0	88.00	99.99
	Web application	8	11.76	99.81	6.900	99.94	8.700	99.87

This results in a reduction in feature dimensionality to 50% of its original size. This feature minimization process is highly advantageous for IDS, as it allows for quicker detection using only the most discriminative features. The feature subsets selected for both datasets are shown in Table 4. The class-wise classification results obtained utilizing the proposed XDeep-IDS approach are summarized in Table 5, which shows increased performance across all classes when compared to classification on the raw dataset.

- **Class balancing**: The datasets utilized in this study show a significant class imbalance, which could lead to class bias in the models. In order to overcome class imbalance, the SMOTE-ENN method is used, which helps create synthetic data for the minority class and removes noisy data from the majority class, thus improving class separability.

- **Model comparison**: A comparative evaluation of various models, such as LGBM, KNN, DNN, MLP, and BiLSTM, is provided in Table 6. Among these models, the BiLSTM model has provided the best results with an accuracy of $97.43 \pm 0.21\%$ and an F1-score of $96.92 \pm 0.20\%$ on the NF-BoT-IoT dataset. In the same way, as presented in Table 6, the BiLSTM model has outperformed the results of other models on the CIC-IoT23 dataset with an accuracy of $99.36 \pm 0.12\%$ and an F1-score of $99.33 \pm 0.11\%$. Accordingly, the proposed model is the BiLSTM model, which is then compared with the baseline models. For the NF-BoT-IoT dataset, the BiLSTM model achieves the lowest FAR of 2.26×10^{-2},

Table 6: Performance comparison of different algorithms on NF-BoT-IoT and CIC-IoT23 datasets using mean ± standard deviation across five stratified cross-validation folds.

Dataset	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FAR
NF-BoT-IoT	LGBM	95.65 ± 0.32	95.70 ± 0.29	86.61 ± 0.47	90.66 ± 0.38	6.43×10^{-2}
	KNN	92.03 ± 0.41	92.41 ± 0.36	92.57 ± 0.33	92.48 ± 0.35	4.97×10^{-2}
	DNN	94.11 ± 0.28	94.59 ± 0.31	84.94 ± 0.52	88.79 ± 0.44	7.19×10^{-2}
	MLP	95.11 ± 0.35	95.70 ± 0.27	83.76 ± 0.49	87.54 ± 0.43	3.79×10^{-2}
	BiLSTM	97.43 ± 0.21	97.75 ± 0.18	96.09 ± 0.24	96.92 ± 0.20	2.26×10^{-2}
CIC-IoT23	LGBM	94.70 ± 0.31	94.89 ± 0.28	94.70 ± 0.34	94.78 ± 0.30	3.47×10^{-2}
	KNN	96.26 ± 0.24	96.24 ± 0.22	96.22 ± 0.26	96.23 ± 0.23	9.54×10^{-2}
	DNN	94.44 ± 0.37	95.37 ± 0.31	94.47 ± 0.39	94.92 ± 0.34	4.86×10^{-2}
	MLP	92.61 ± 0.42	92.72 ± 0.39	92.61 ± 0.44	92.66 ± 0.41	2.72×10^{-2}
	BiLSTM	99.36 ± 0.12	99.35 ± 0.10	99.31 ± 0.14	99.33 ± 0.11	1.17×10^{-2}

followed by MLP (3.79×10^{-2}) and KNN (4.97×10^{-2}), while LGBM and DNN record higher FAR values of 6.43×10^{-2} and 7.19×10^{-2}, respectively. BiLSTM also performs better than all other models on the CIC-IoT23 dataset, with the lowest FAR of 1.17×10^{-2}, followed by MLP (2.72×10^{-2}) and LGBM (3.47×10^{-2}). In contrast, DNN (4.86×10^{-2}) and KNN (9.54×10^{-2}) exhibit comparatively higher FAR values. Overall, the BiLSTM model demonstrates consistently higher performance across both datasets by effectively minimizing false positive alarms. Given that a low FAR is essential for reliable IDS, these results confirm the robustness and practical suitability of the BiLSTM model for real-world IoT security applications.

5.1 State-of-the-art model comparison

Table 7 and Table 8 present the comparison of proposed and baseline models for CIC-IoT23 and NF-BoT-IoT, respectively. The proposed model used the resulting reduced feature subset and SMOTE-ENN to achieve $99.36 \pm 0.12\%$ accuracy for the CIC-IoT23 dataset. Therefore, compared to baseline studies, the proposed strategy with the BiLSTM-based IDS shows an approximate accuracy increase of 0.26%-7.4% compared to [36] and [37], respectively. For the NF-BoT-IoT dataset, the proposed strategy achieved $97.43 \pm 0.21\%$ accuracy with an approximate accuracy increase of 9.67%-13.96% compared to [38] and [39], respectively.

For IDS, precision evaluates how accurately the system identifies malicious activities without false alarms, while recall measures its ability to detect all actual intrusions. The proposed strategy gives a precision of 99.35 ± 0.10 and a recall of $99.31 \pm 0.14\%$ for the CIC-IoT23 dataset, with an approximate increase of 2.36%–31.65% in precision compared to [40] and [36], respectively. For recall value, increase by 1.33%–8.71% compared to [41] and [36], respectively. The proposed approach, with a precision of $97.75 \pm 0.18\%$ and recall of $96.09 \pm 0.24\%$ on the NF-BoT-IoT dataset, shows an approximate increase of 14.16%–19.54% in precision compared to [38] and [39], respectively. For recall value, it increases 12.58%–14.66% compared to [42] and [43] respectively. The F1-score balances precision and recall, providing a single metric that reflects the system's effectiveness in correctly identifying intrusions

Table 7: Comparison between the proposed and baseline methods for the CIC-IoT23 dataset in a multi-class scenario.

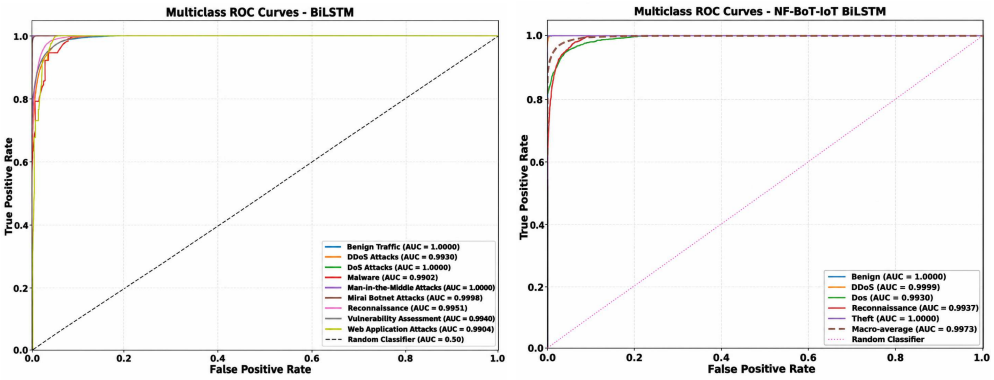
Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
RNN [44]	96.52	96.25	96.52	96.38
DifK-SAN+AO [41]	93.98	94.01	97.99	95.95
VGGIncepNet [37]	92.00	92.00	93.00	92.50
DNN [36]	99.11	67.94	90.66	77.67
MLP [40]	97.46	97.00	97.00	97.00
DL-BiLSTM [45]	93.13	91.80	93.13	92.46
XDeep-IDS (Proposed)	**99.36 $\pm$ 0.12**	**99.35 $\pm$ 0.10**	**99.31 $\pm$ 0.14**	**99.33 $\pm$ 0.11**

Table 8: Comparison of the proposed and baseline methods on the NF-BoT-IoT dataset for multiclass classification.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
DNN [39]	83.82	78.64	83.82	80.66
MLP [43]	85.00	84.00	82.00	82.98
DNN [42]	84.60	87.25	83.97	85.19
NB [38]	88.00	90.00	84.00	87.00
TS-IDS [46]	NA	NA	83.63	85.73
NEGAT+NEGSC [47]	NA	NA	83.06	82.70
EGAT-LSTM [48]	NA	NA	83.29	84.21
XDeep-IDS (Proposed)	**97.43 $\pm$ 0.21**	**97.75 $\pm$ 0.18**	**96.09 $\pm$ 0.24**	**96.92 $\pm$ 0.20**

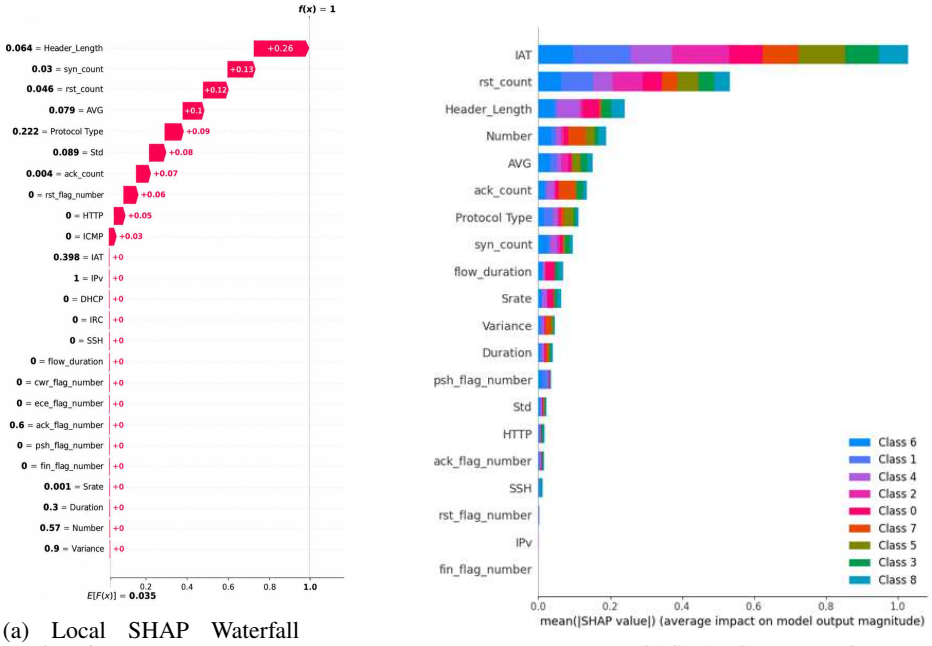
while minimizing missed detections and false alarms. The proposed method gives a 99.33 $\pm$ 0.11% F1-score value that is approximately 2.34%–21.8% higher compared with baseline techniques [40] and [36], respectively. For the NF-BoT-IoT dataset, a 96.92 $\pm$ 0.20% F1-score value of approximately 10.24%–16.77% is achieved with the proposed approach, which is approximately 10.24%–16.77% more compared to other baseline techniques [38] and [39], respectively.

Additionally, multiclass ROC curves were created to assess the classification effectiveness of the suggested BiLSTM model across several different classes in the CIC-IoT23 dataset, as shown in Figure 2a. In every assessed category, the model demonstrates exceptional discriminative power. Specifically, Benign Traffic, DoS Attacks, and Man-in-the-Middle Attacks achieved optimal classification performance with an AUC of 1.0000. Additionally, near-perfect separation was shown by sophisticated attack vectors such as malware (AUC = 0.9902), reconnaissance (AUC = 0.9951), and Mirai botnet attacks (AUC = 0.9998). As shown in Figure 2b, multiclass ROC curves were assessed on the NF-BoT-IoT dataset to further verify the robustness and generalizability of the suggested BiLSTM architecture. With an overall macro-average AUC of 0.9973, the model demonstrates exceptional discrimination across all threat categories. In particular, DDoS (0.9999), Reconnaissance (0.9937), and DoS (0.9930) showed high, consistent detection capacity with few false alarms, but benign traffic and theft attacks achieved optimal separation with an AUC of 1.0000. These findings demonstrate that the BiLSTM architecture consistently and with few false positives differentiates between typical IoT network behavior and various cyberthreats.



(a) CIC-IoT23 dataset (b) NF-BoT-IoT dataset

Fig. 2: ROC Curve for CIC-IoT23 and NF-BoT-IoT Datasets



(a) Local SHAP Waterfall Explanation (b) SHAP Summary plot

Fig. 3: SHAP analysis for CIC-IoT23 dataset

5.2 XAI Interpretability

We used two model-agnostic XAI strategies to increase transparency and confidence in IoT intrusion detection. SHAP was employed to provide global model interpretability, while LIME was used to explain individual predictions at the local level.

5.2.1 SHAP and LIME-Based XAI Analysis on the CIC-IoT23 Dataset

Figure 3a depicts the SHAP local waterfall plot for a representative CIC-IoT23 traffic instance, highlighting Header_Length, syn_count, and rst_count as the primary positive contributors to the malicious classification. These instance-level explanations validate the IDS's decision-making process, confirming that structural flow anomalies drive accurate threat detection. Figure 3b ranks feature importance, where IAT, rst_count, and Header_Length dominate, while protocol flags (psh_flag_number, fin_flag_number) contribute minimally. These patterns explain the strong performance of the utilized model, which captures temporal dependencies emphasized by these time-related features.

Figure 4 presents a representative LIME instance where the model classifies the traffic as "NOT 4" with an absolute probability of 1.00, while assigning negligible probabilities to class 4 (Man in-the-Middle class: MITM-ArpSpoofing and DNS Spoofing of Subcategory) and alternative categories. The local attribution breakdown shows that the model's conclusion is heavily influenced by certain conditions, particularly Header Length > 0.06, Protocol Type > 0.21, and rst_flag_number $<= 0.00$. Additionally, the corresponding feature value table provides transparent, granular validation of how specific network flow characteristics influence the classifier's output by describing the exact measurements of attributes like Header_Length (0.06), Protocol Type (0.22), ack_flag_number (0.60), and IAT (0.40).

The local SHAP force plot that explains the BiLSTM model's prediction for a particular instance from the CIC-IoT23 dataset is shown in Figure 5. The DL model's output score is driven to $f(x) = 1.00$, indicating a confident classification of malicious traffic, starting from a dataset baseline average of roughly 0.0355. Key network flow and packet attributes, namely Header_Length = 0.06371, syn_count = 0.02972, rst_count = 0.04629, AVG = 0.07911, Protocol Type = 0.2225, Std = 0.08896, ack_count = 0.003937, rst_flag_number = 0, HTTP = 0, and MP = 0, govern this positive shift.

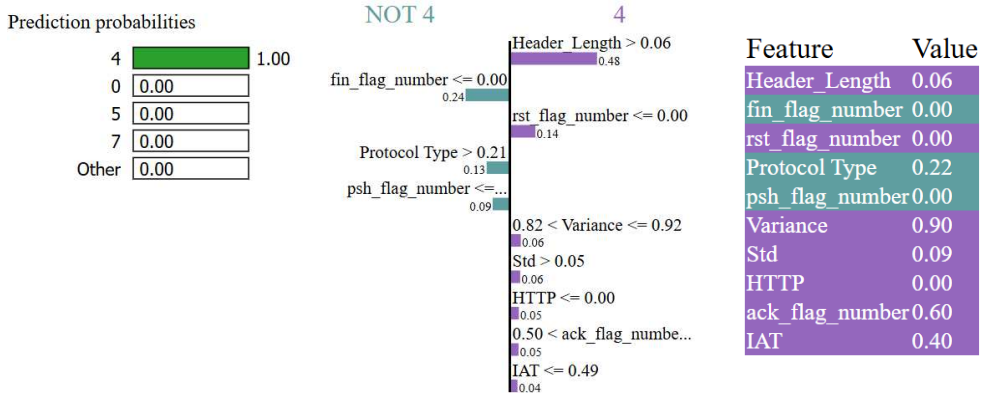


Fig. 4: LIME observations for the CIC-IoT23 dataset on the benign class

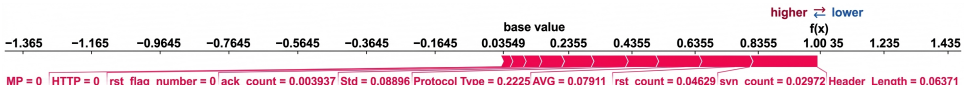
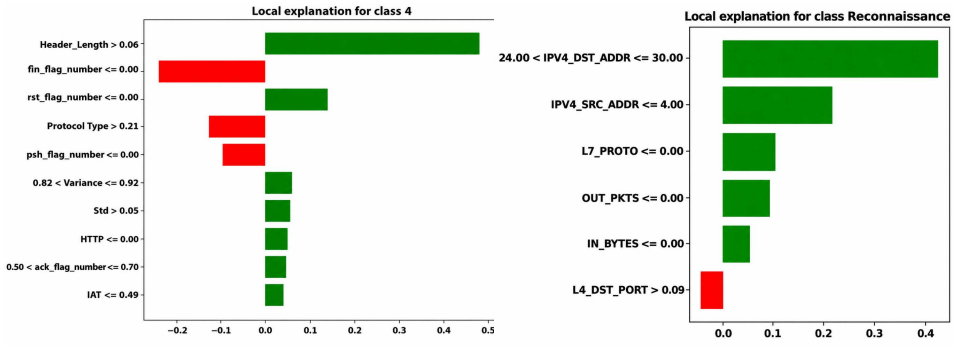


Fig. 5: SHAP Force Plot for a Local Prediction on the CIC-IoT23 dataset



(a) CIC-IoT23 dataset (b) NF-BoT-IoT dataset

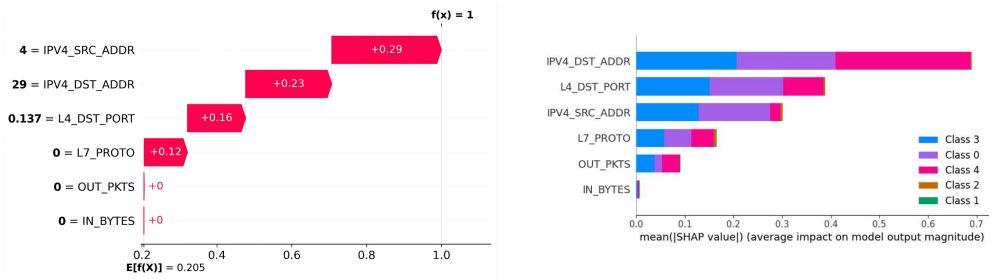
Fig. 6: LIME observations using a subplot

Figure 6a illustrates the LIME explanation for Class 4 (Man-in-the-Middle class: MITM-ArpSpoofing and DNS_Spoofing of Subcategory) on the CIC-IoT23 dataset, providing local insights into how individual network traffic features and flag thresholds influence the DL model's classification decision. The classification of this particular network flow into Class 4 is highly supported by the positive contributions (green bars), which are headed by a larger header length (Header_Length $>$ 0.06) coupled with specific variance (0.82 $<$ Variance $<=$ 0.92) and inter-arrival time (IAT $<=$ 0.49) ranges. On the other hand, opposing factors (red bars) like missing flag indications (fin_flag_number $<=$ 0.00) and protocol type thresholds (Protocol Type $>$ 0.21) serve as counter-evidence, illustrating the fine-grained, balanced logic the model uses at the instance level.

5.2.2 SHAP and LIME-Based XAI Analysis on the NF-BoT-IoT Dataset

The SHAP local waterfall plot for a typical NF-BoT-IoT traffic instance is shown in Figure 7a, where the main positive factors driving the malicious categorization are IPV4_SRC_ADDR, IPV4_DST_ADDR, and L4_DST_PORT. These instance-level explanations verify that target port characteristics and source-destination addressing patterns are important markers for precise detection of botnet threats. Mean absolute SHAP values are summarized in Figure 7b. In this case, IPV4_DST_ADDR was the most dominant attribute, followed by IPV4_SRC_ADDR and L4_DST_PORT. It is evident from these results that the chosen model uses mainly traffic attributes based on the addresses and ports in order to differentiate classes with different colors, and there are obvious differences between attack classes.

The local explanation using LIME technique is presented in Figure 8 to demonstrate how the NF-BoT-IoT classification model makes decisions internally. As indicated by the prediction probabilities, the model classifies the targeted traffic as a Reconnaissance attack with absolute certainty (1.00), while assigning negligible probabilities (0.00) to Benign, DDoS, DoS, and Theft classes. The feature contribution breakdown highlights that specific attributes most notably IPV4_DST_ADDR (29.00), IPV4_SRC_ADDR (4.00), L7_PROTO (0.00), OUT_PKTS (0.00), and IN_ BYTES (0.00) strongly drive the model toward the reconnaissance classification. In contrast, L4_DST_PORT (0.14) has a small impact that favors the alternative class ("NOT Reconnaissance"), showing the fine detail about how LIME makes



(a) Local SHAP Waterfall Explanation (b) SHAP Summary Plot

Fig. 7: SHAP analysis for the NF-BoT-IoT dataset

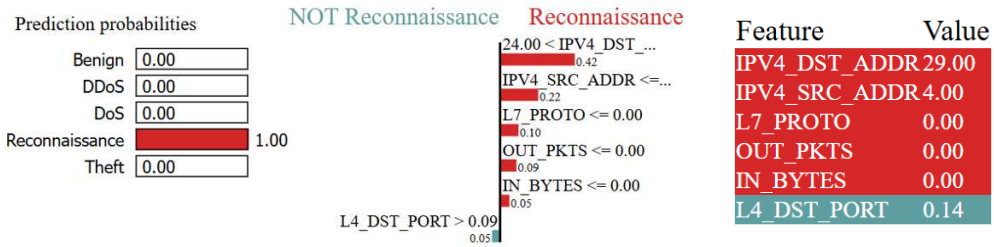


Fig. 8: LIME observations for NF-BoT-IoT dataset



Fig. 9: SHAP Force Plot for a Local Prediction on the NF-BoT-IoT dataset

clear the importance of each feature in forming a prediction. On the other hand, Figure 6b presents a LIME local feature-contribution plot for an individual reconnaissance attack sample, illustrating how specific feature-value ranges influence the DL model's prediction on the NF-BoT-IoT dataset. Green-colored bars represent supportive features like Destination IP ($24.00 < \text{IPV4_DST_ADDR} \leq 30.00$), Source IP ($\text{IPV4_SRC_ADDR} \leq 4.00$), and Protocol ($\text{L7_PROTO} \leq 0.00$) values, which help to move the algorithm in the direction of reconnaissance. On the other hand, the red bar corresponding to $\text{L4_DST_PORT} > 0.09$ carries a negative weight, opposing the Reconnaissance class.

Figure 9 shows the local SHAP force plot that explains the prediction of the DL model for a particular instance of the NF-BoT-IoT dataset. With a base average value from the dataset of approximately 0.2049, the model output value is made to converge to $f(x) = 1.00$, signifying a high confidence level for malicious IoT traffic. This positive shift is governed by key network flow attributes, specifically $\text{IPV4_SRC_ADDR} = 4$, $\text{IPV4_DST_ADDR} = 29$, $\text{L4_DST_PORT} = 0.1374$, and $\text{L7_PROTO} = 0$, thereby providing transparent, instance-level insights into how individual IP addresses, destination ports, and protocol configurations influence the neural network's decision-making process.

Table 9: Ablation Study of the Major Components of the Proposed IDS on the NF-BoT-IoT Dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (s)	Prediction Time (s)
With Raw Data	77.80 ± 0.27	58.45 ± 0.31	57.67 ± 0.42	57.88 ± 0.29	371.09 ± 9.42	2.13 ± 0.38
With GA	83.90 ± 0.40	72.29 ± 0.29	62.40 ± 0.41	67.34 ± 0.17	191.25 ± 4.17	1.32 ± 0.21
With SMOTE-ENN	95.59 ± 0.33	95.42 ± 0.40	95.20 ± 0.31	95.21 ± 0.26	603.26 ± 6.35	3.21 ± 0.41
XDeep-IDS	$\mathbf{97.43 \pm 0.21}$	$\mathbf{97.75 \pm 0.18}$	$\mathbf{96.09 \pm 0.24}$	$\mathbf{96.92 \pm 0.20}$	286.56 ± 3.28	1.62 ± 0.36

Table 10: Ablation study of the major components of the proposed IDS on the CIC-IoT-23 dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (s)	Prediction Time (s)
With Raw Data	98.95 ± 0.08	70.35 ± 0.24	67.75 ± 0.31	69.05 ± 0.27	421.55 ± 1.43	2.37 ± 0.36
With GA	99.02 ± 0.06	72.14 ± 0.21	67.67 ± 0.28	69.92 ± 0.24	215.53 ± 0.39	1.54 ± 0.44
With SMOTE-ENN	98.43 ± 0.15	97.68 ± 0.18	98.75 ± 0.23	98.21 ± 0.16	711.65 ± 0.35	3.41 ± 0.44
XDeep-IDS	$\mathbf{99.36 \pm 0.12}$	$\mathbf{99.35 \pm 0.10}$	$\mathbf{99.31 \pm 0.14}$	$\mathbf{99.33 \pm 0.11}$	351.88 ± 0.37	1.87 ± 0.29

Table 11: Ablation study of BiLSTM sequence length on the NF-BoT-IoT dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (s)	Prediction Time (s)
Without Sequence ($L = 1$)	95.82 ± 0.20	95.81 ± 0.32	95.82 ± 0.21	95.81 ± 0.33	245.41 ± 0.22	1.48 ± 0.41
With Sequence ($L = 10$)	$\mathbf{97.43 \pm 0.21}$	$\mathbf{97.75 \pm 0.18}$	$\mathbf{96.09 \pm 0.24}$	$\mathbf{96.92 \pm 0.20}$	286.56 ± 3.28	1.62 ± 0.36

Table 12: Ablation study of BiLSTM sequence length on the CIC-IoT23 dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (s)	Prediction Time (s)
Without Sequence ($L = 1$)	87.67 ± 0.47	88.34 ± 0.36	87.67 ± 0.41	87.60 ± 0.39	264.52 ± 0.22	1.59 ± 0.51
With Sequence ($L = 10$)	$\mathbf{99.36 \pm 0.12}$	$\mathbf{99.35 \pm 0.10}$	$\mathbf{99.31 \pm 0.14}$	$\mathbf{99.33 \pm 0.11}$	351.88 ± 0.37	1.87 ± 0.29

5.3 Ablation Study

The results of the ablation study carried out on the NF-BoT-IoT and CIC-IoT-23 datasets are presented in Table 9 and Table 10, respectively, demonstrates the individual and collective effects of feature selection and data balancing in the detection of intrusions. In both datasets, the models trained on the raw data demonstrate biased classification with high accuracy and poor precision, recall, and F1 measures due to the presence of large class imbalance and redundancy in the datasets. In contrast, the SMOTE-ENN-only approach achieves lower accuracy than the proposed method while requiring considerably more computational time. Adding the GA-based feature selection strategy results in an improvement in the accuracy and precision of the classifiers by discarding the irrelevant features, but the recall and F1 measures remain poor. In contrast, the proposed GA with the SMOTE-ENN framework achieves consistently higher and well-balanced performance, reaching approximately $99.33 \pm 0.11\%$ F1-score on CIC-IoT-23 and $96.92 \pm 0.20\%$ on NF-BoT-IoT. Although this improvement incurs higher computational cost during training and inference, the results demonstrate that the joint application of optimized feature selection and hybrid resampling is essential for reliable, noise-resilient, and highly accurate intrusion detection in imbalanced IoT traffic scenarios.

To investigate the contribution of sequence modeling in the proposed framework, two BiLSTM configurations were evaluated on both the NF-BoT-IoT and CIC-IoT23 datasets, as presented in Table 11 and Table 12, respectively. In the first configuration, GA + SMOTE-ENN + BiLSTM without sequence modeling was implemented with a sequence length of 1, where each network-flow record was treated as an independent input. In the second configuration, GA + SMOTE-ENN + BiLSTM with sequence modeling was implemented using a sequence length of 10, where each input consisted of 10 consecutive flow records. In both configurations, the same GA-selected features and SMOTE-ENN-balanced datasets were used to ensure a fair comparison. This experimental design enables the effect of explicit sequence modeling on BiLSTM-based intrusion detection performance to be evaluated independently of feature selection and class balancing. BiLSTM was evaluated under independent-record and fixed-length sequence configurations to investigate whether modeling relationships among consecutive flow records improves intrusion-detection performance. As a result, the proposed approach XDeep-IDS effectively utilizes sequence length to deliver higher detection accuracy with reasonable computational overhead.

5.4 Limitation of the study

This study employed GA for feature selection and SMOTE-ENN to address class imbalance, using the CIC-IoT23 and NF-BoT-IoT datasets for evaluation. Although these approaches produced promising results, they may not fully address the challenges associated with developing robust IDS. Future studies could therefore investigate alternative class-balancing techniques, including variants of Generative Adversarial Networks (GANs), as well as advanced feature-selection and feature-fusion approaches. Besides, applying the discussed algorithms to more modern and diversified datasets from IoT cybersecurity would enable one to conduct a more generalized assessment and enhance IDS performance even further.

6 Conclusion

Network data characteristics pose significant challenges to IDS, particularly due to their high dimensionality, class imbalance, noise, and outliers. This study proposed XDeep-IDS, which effectively addressed these challenges through a comprehensive approach combining GA for feature selection and SMOTE-ENN for class imbalance resolution, resulting in utilized BiLSTM-based models that demonstrated higher performance across multiple evaluation metrics compared to the state-of-the-art, especially in multi-class attack classification scenarios. In addition, to enhance the transparency and reliability of IoT intrusion detection, we employed two model-agnostic XAI techniques, SHAP and LIME. These advancements aim to further improve IDS capabilities in response to evolving security challenges in modern network environments, particularly in scenarios requiring robust and adaptive solutions for emerging attack patterns. The proposed framework achieved mean accuracies of $97.43 \pm 0.21\%$ and $99.36 \pm 0.12\%$ on the NF-BoT-IoT and CIC-IoT23 datasets, respectively, demonstrating its effectiveness in multiclass intrusion detection. To further improve the effectiveness and efficiency of IDS, future research will explore various balancing techniques and federated learning approaches integrating with fusion-based feature selection and incorporate more recent datasets for comprehensive analysis.

Author Contributions Tibebu Bekele Shana: Conceptualization, methodology, implementation, writing - original draft, writing - review, and editing. Krish Vaish: Implementation, investigation, writing - original draft, writing - review, and editing. Samrat Mondal: Conceptualization, supervision, and review.

Funding: This work was not supported by any funding agency.

Data Availability: The datasets used in this study, CICIoT23 and NF-BoT-IoT, are publicly available at https://www.unb.ca/cic/datasets/iotdataset-2023.html and https://espace.library.uq.edu.au/view/UQ:62e6d80., respectively.

Declaration of competing interest

Ethics approval None
Consent for publication None
Code availability On request
Competing interests The authors declare no competing interests.

References

[1] Smys, S., Basar, A., Wang, H., *et al.*: Hybrid intrusion detection system for internet of things (iot). Journal of ISMAC **2**(04), 190–199 (2020)

[2] Jin, D., Lu, Y., Qin, J., Cheng, Z., Mao, Z.: Swiftids: Real-time intrusion detection system based on lightgbm and parallel intrusion detection mechanism. Computers & Security **97**, 101984 (2020)

[3] Pang, H., Jin, F., Chen, M., Jiang, Y., Yuan, Y.: A multistage intrusion detection method for alleviating class overlapping problem. Neural Computing and Applications **37**(6), 4853–4867 (2025)

[4] Medhi, J., Liu, R., Wang, Q., Chen, X.: A lightweight and efficient intrusion detection system (ids) for unmanned aerial vehicles. Neural Computing and Applications **37**(20), 15819–15836 (2025)

[5] Zhou, Y., Cheng, G., Jiang, S., Dai, M.: Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer networks **174**, 107247 (2020)

[6] Abdulganiyu, O.H., Ait Tchakoucht, T., Saheed, Y.K.: A systematic literature review for network intrusion detection system (ids) oh abdulganiyu et al. International journal of information security **22**(5), 1125–1162 (2023)

[7] Liu, J., Gao, Y., Hu, F.: A fast network intrusion detection system using adaptive synthetic oversampling and lightgbm. Computers & Security **106**, 102289 (2021)

[8] Injadat, M., Moubayed, A., Nassif, A.B., Shami, A.: Multi-stage optimized machine learning framework for network intrusion detection. IEEE Transactions on Network and Service Management **18**(2), 1803–1816 (2020)

[9] Salo, F., Nassif, A.B., Essex, A.: Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection. Computer networks **148**, 164–175 (2019)

[10] Sarker, I.H.: Cyberlearning: Effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks. Internet of Things **14**, 100393 (2021)

[11] Talukder, M.A., Islam, M.M., Uddin, M.A., Hasan, K.F., Sharmin, S., Alyami, S.A., Moni, M.A.: Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. Journal of big data **11**(1), 33 (2024)

[12] Qazi, E.U.H., Faheem, M.H., Zia, T.: Hdlnids: Hybrid deep-learning-based network intrusion detection system. Applied Sciences **13**(8), 4921 (2023)

[13] Hnamte, V., Hussain, J.: Dcnnbilstm: An efficient hybrid deep learning-based intrusion detection system. Telematics and Informatics Reports **10**, 100053 (2023)

[14] Wu, Z., Zhang, H., Wang, P., Sun, Z.: Rtids: A robust transformer-based approach for intrusion detection system. IEEE Access **10**, 64375–64387 (2022)

[15] Manocchio, L.D., Layeghy, S., Lo, W.W., Kulatilleke, G.K., Sarhan, M., Portmann, M.: Flowtransformer: A transformer framework for flow-based network intrusion detection systems. Expert Systems with Applications **241**, 122564 (2024)

[16] Kiran, A., Prakash, S.W., Kumar, B.A., Sameeratmaja, T., Charan, U.S.S.R., *et al.*: Intrusion detection system using machine learning. In: 2023 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–4 (2023). IEEE

[17] Zhang, X., Ran, J., Mi, J.: An intrusion detection system based on convolutional neural network for imbalanced network traffic. In: 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), pp. 456–460 (2019)

[18] Patil, S., Varadarajan, V., Mazhar, S.M., Sahibzada, A., Ahmed, N., Sinha, O., Kumar, S., Shaw, K., Kotecha, K.: Explainable artificial intelligence for intrusion detection system. Electronics **11**(19), 3079 (2022) https://doi.org/10.3390/electronics11193079

[19] Sarhan, M., Layeghy, S., Portmann, M.: Towards a standard feature set for network intrusion detection system datasets. Mobile Networks and Applications **27**(1), 357–370 (2022) https://doi.org/10.1007/s11036-021-01843-0

[20] More, S., Idrissi, M., Mahmoud, H., Asyhari, A.T.: Enhanced intrusion detection systems performance with unsw-nb15 data analysis. Algorithms **17**(2), 64 (2024) https://doi.org/10.3390/a17020064

[21] Arreche, O., Guntur, T., Abdallah, M.: Xai-ids: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems. Applied Sciences **14**(10), 4170 (2024) https://doi.org/10.3390/app14104170

[22] Alabbadi, A., Bajaber, F.: An intrusion detection system over the iot data streams using explainable artificial intelligence (xai). Sensors **25**(3), 847 (2025) https://doi.org/10.3390/s25030847

[23] Wali, S., Farrukh, Y.A., Khan, I.: Explainable ai and random forest based reliable intrusion detection system. Computers & Security **157**, 104542 (2025) https://doi.org/10.1016/j.cose.2025.104542

[24] Behera, A., Sagar Sahoo, K., Kumara Mishra, T., Nayyar, A., Bilal, M.: Enhancing ddos detection in sdiot through effective feature selection with smote-enn. PloS one **19**(10), 0309682 (2024)

[25] Pramanick, N., Srivastava, S., Mathew, J., Agarwal, M.: Enhanced ids using bba and smote-enn for imbalanced data for cybersecurity. SN Computer Science **5**(7), 875 (2024)

[26] Gharaee, H., Hosseinvand, H.: A new feature selection ids based on genetic algorithm and svm. In: 2016 8th International Symposium on Telecommunications (IST), pp. 139–144 (2016). IEEE

[27] Halim, Z., Yousaf, M.N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., Ahmad, I., Hanif, M.: An effective genetic algorithm-based feature selection method for intrusion detection systems. Computers & Security **110**, 102448 (2021)

[28] Tahir, M., Tubaishat, A., Al-Obeidat, F., Shah, B., Halim, Z., Waqas, M.: A novel binary chaotic genetic algorithm for feature selection and its utility in affective computing and healthcare. Neural Computing and Applications **34**(14), 11453–11474 (2022)

[29] Xu, H., Fu, Y., Fang, C., Cao, Q., Su, J., Wei, S.: An improved binary whale optimization algorithm for feature selection of network intrusion detection. In: 2018 IEEE 4th International Symposium on Wireless Systems Within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), pp. 10–15 (2018). IEEE

[30] Almomani, O.: A feature selection model for network intrusion detection system based on pso, gwo, ffa and ga algorithms. Symmetry **12**(6), 1046 (2020)

[31] Hariharan, S., Rejimol Robinson, R.R., Prasad, R.R., Thomas, C., Balakrishnan, N.: Xai for intrusion detection system: comparing explanations based on global and local scope. Journal of Computer Virology and Hacking Techniques **19**(2), 217–239 (2023) https://doi.org/10.1007/s11416-022-00441-2

[32] Sivamohan, S., Sridhar, S.S.: An optimized model for network intrusion detection

systems in industry 4.0 using xai based bi-lstm framework. Neural Computing and Applications **35**(15), 11459–11475 (2023) https://doi.org/10.1007/s00521-023-08319-0

[33] Arreche, O., Guntur, T., Abdallah, M.: Xai-based feature selection for improved network intrusion detection systems. arXiv preprint arXiv:2410.10050 (2024) https://doi.org/10.48550/arXiv.2410.10050

[34] Tserenkhuu, M., Hossain, M.D., Taenaka, Y., Kadobayashi, Y.: Intrusion detection system framework for sdn-based iot networks using deep learning approaches with xai-based feature selection techniques and domain-constrained features. IEEE Access **13**, 136864–136880 (2025) https://doi.org/10.1109/ACCESS.2025.3595595

[35] Deb, K.: An introduction to genetic algorithms. Sadhana **24**(4), 293–315 (1999)

[36] Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., Ghorbani, A.A.: CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors **23**(13), 5941 (2023)

[37] Chen, J., Xiao, J., Xu, J.: Vggincepnet: Enhancing network intrusion detection and network security through non-image-to-image conversion and deep learning. Electronics **13**(18), 3639 (2024)

[38] Khanday, S.A., Fatima, H., Rakesh, N.: Implementation of intrusion detection model for ddos attacks in lightweight iot networks. Expert Systems with Applications **215**, 119330 (2023)

[39] Vishwakarma, M., Kesswani, N.: Dids: A deep neural network based real-time intrusion detection system for iot. Decision Analytics Journal **5**, 100142 (2022)

[40] Gheni, H.Q., Al-Yaseen, W.L.: Two-step data clustering for improved intrusion detection system using ciciot2023 dataset. e-Prime - Advances in Electrical Engineering, Electronics and Energy **9**, 100673 (2024)

[41] Sharma, M., Sharma, M., Sharma, P., Gurpur, S., Taqui, S.N., *et al.*: Security analysis of iot networks using diffusion kernel spike attention network with atomic optimization based intrusion detection system. In: 2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI), pp. 146–152 (2024). IEEE

[42] Deng, P., Huang, Y.: Edge-featured multi-hop attention graph neural network for intrusion detection system. Computers & Security **148**, 104132 (2025)

[43] Alotaibi, F.A., Mishra, S.: Cyber security intrusion detection and bot data collection using deep learning in the iot. International Journal of Advanced Computer Science and Applications **15**(3) (2024)

[44] Abbas, S., Bouazzi, I., Ojo, S., Al Hejaili, A., Sampedro, G.A., Almadhor, A., Gregus, M.: Evaluating deep learning variants for cyber-attacks detection and multi-class

classification in iot networks. PeerJ Computer Science **10**, 1793 (2024)

[45] Wang, Z., Chen, H., Yang, S., Luo, X., Li, D., Wang, J.: A lightweight intrusion detection method for iot based on deep learning and dynamic quantization. PeerJ Computer Science **9**, 1569 (2023)

[46] Nguyen, H., Kashef, R.: Ts-ids: Traffic-aware self-supervised learning for iot network intrusion detection. Knowledge-Based Systems **279**, 110966 (2023)

[47] Xu, R., Wu, G., Wang, W., Gao, X., He, A., Zhang, Z.: Applying self-supervised learning to network intrusion detection for network flows with graph neural network. Computer Networks **248**, 110495 (2024)

[48] Zhang, L., Tan, L., Shi, H., Sun, H., Zhang, W.: Malicious traffic classification for iot based on graph attention network and long short-term memory network. In: 2023 24st Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 54–59 (2023). IEEE

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.