International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(14s) 2026 pp. 659-674
https://doi.org/10.70917/ijcisim-2026-4273
Article

Explainable Deep Learning Intrusion Detection Framework for Securing IoT Environment

Ravi Patni[1], Gurvinder Singh[2]

[1,2] Department of Computer Science, Guru Nanak Dev University, Amritsar, 143005, India.
Corresponding author: Ravi Patni (ravicsc.rsh@gndu.ac.in).

Abstract: In the fast-growing world of Internet of Things (IoT), devices have exploded that are not only efficient but also expose serious security vulnerabilities that can be used as vectors for more advanced cyber-attacks. Traditional IDS has the challenge of false positive rate, which could cause critical operations to be disrupted in various domains from smart medical devices (SMDs) to municipal infrastructure. Machine Learning (ML) and Deep Learning (DL) models are state-of-the art solutions to detect complex, high-dimensional and temporal network anomalies in terms of accuracy but their deployment is still hampered severely due to the fact that they lack interpretability. This paper introduces a new explainable hybrid IDS architecture for IoT environments named XABiL-IDS (Explainable Attention-based Bi LSTM-Intrusion Detection System) in response to this challenge. This study uses a robust hybrid architecture to detect attacks effectively. Global analysis using the SHAP method for determining the most relevant traffic attributes affecting the classification process in the dataset on the other hand local analysis done by LIME for providing explanation at the instance level on the prediction made regarding network flows. The key differentiating feature of this approach compared to earlier methods is the incorporation of both global and local explainability in single pipeline.

Keywords: Machine Learning (ML), Deep Learning (DL), Distributed Denial of Service (DDoS), Internet of Things (IoT), Local Interpretable Model-agonistic Explanations (LIME), SHapley Additive exPlanations (SHAP), Explainable Artificial Intelligence (XAI)

1. INTRODUCTION

Modern research on IDS for IoT and networks has seen increased use of ML and DL algorithms to detect intricate patterns of network traffic. The initial IDS research was largely based on statistics and shallow learning algorithms that were effective computationally but could not model the high-dimensional and temporal nature of network traffic [1],[2]. This drawback was overcome by using DL methods such as autoencoders, Convolutional Neural Network (CNNs), and recurrent algorithms, which led to higher accuracy due to their ability to learn feature hierarchies [3], [4], [5]. A human administrator must decide whether to ignore or blindly trust a machine when a model based on the dataset that labels a packet as malicious. This is where the discussion is transformed via LIME and SHAP. LIME provides detailed justifications at the instance level that comprehends explanations for individual and forecasts, the use of local interpretability approaches on the contrary SHAP, which is based on coalitional game theory, enables us to interrogate the model. SHAP divides the decision-making process into contributions that are accessible by humans rather than producing a straightforward Yes/No result. It tells that is it flagged as a DDoS attack because the Inter-Arrival time was too consistent and the packet length was unusually small. By integrating SHAP into IoT security, it is more than just detecting intrusions, that will be making the AI accountable. This paper proposes a novel framework that uses LIME and SHAP values not just for post-mortem analysis, but as a real-time trust filter.

In addition, recent models have considered the use of attention mechanisms and hybrid architectures to help them focus on critical traffic characteristics and temporal relationships [6], [7]. Attention-based Bi-LSTMs and multi-headed attention architectures show significant strength in detecting advanced attacks through the modeling of both spatial and temporal relationships among network flows. Despite these advancements in prediction performance, previously proposed models are predominantly considered black boxes and lack of explanatory capabilities.

In parallel, the use of XAI approaches has become increasingly common in developing IDSs that can provide some level of interpretability for their predictions [8]. Specifically, global interpretability techniques, such as SHAP, have been widely applied to determine which features play a pivotal role in recognizing attacks in a dataset. While these techniques offer insights into the behavior of an IDS model at the macroscopic level. On the other hand, cybersecurity applications require more granular explanations at the instance level. Therefore, the application of local interpretability techniques, such as LIME, has been considered to provide interpretable explanations for individual predictions generated from IDSs [8] [9]. Local interpretability techniques enable researchers to provide explanations for how a model makes its predictions for each sample. However, the sole use of local interpretability techniques to interpret models leads to inconsistencies since they fail to provide a complete picture of the global decision boundary of a model.

Although the emphasis placed on the importance of integrated explainability methods in recent research trends, many IDSs developed in previous works solely employed either SHAP or LIME without exploiting their synergies. In addition, there appears to be a gap in the development of a consistent framework that can incorporate the advantages of XAI techniques, model architecture, and semantic security considerations in IoT-based networks.

2. RELATED WORK

Recent research has demonstrated significant progress in applying Machine Learning (ML), Deep Learning (DL), and eXplainable Artificial Intelligence (XAI) techniques to Intrusion Detection Systems (IDS), particularly in IoT, cloud, and network security environments. Machine learning remains one of the most widely adopted approaches due to its ability to efficiently process structured network traffic and identify malicious activities with high accuracy. Traditional classifiers such as Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and Naïve Bayes have shown excellent performance on benchmark datasets including NSL-KDD, CICIDS2017, BoT-IoT, and TON_IoT. Survey studies further indicate that ensemble learning techniques significantly improve detection accuracy while reducing false alarm rates compared with individual classifiers [10]–[12]. However, conventional ML-based IDS still encounter challenges related to class imbalance, high-dimensional feature spaces, and reduced generalization when detecting previously unseen attacks.

Deep learning has emerged as a powerful alternative by automatically learning hierarchical feature representations from raw network traffic. Architectures such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Autoencoders have demonstrated superior capability in detecting complex and zero-day cyberattacks compared with conventional machine learning methods [13], [14]. Recent studies have also proposed hybrid intrusion detection frameworks that combine machine learning algorithms such as XGBoost with deep learning models including CNN and BiLSTM to exploit both discriminative feature learning and temporal sequence modelling. Furthermore, techniques such as the Synthetic Minority Oversampling Technique (SMOTE) are frequently incorporated to mitigate class imbalance and improve minority attack detection performance [13], [15].

Another important research direction focuses on ensemble learning, where multiple classifiers are combined to improve prediction robustness and overall detection accuracy. Optimization-based ensemble frameworks integrating Random Forest (RF), XGBoost (XGB), LightGBM (LGBM), CatBoost (CB), and Decision Tree (DT) have consistently outperformed individual machine learning algorithms across various intrusion detection datasets [10], [15]. Despite their superior classification performance, ensemble approaches generally require greater computational resources and memory, limiting their applicability in resource-constrained IoT devices and edge computing environments.

Model interpretability has become an increasingly important requirement for practical intrusion detection systems. Explainable Artificial Intelligence (XAI) techniques provide transparent explanations for model predictions, enabling security analysts to better understand attack classifications and improve trust in automated detection systems. Methods such as SHapley Additive exPlanations (SHAP), Local Interpretable Model-Agnostic Explanations (LIME), and feature attribution analysis have been widely adopted to explain the behaviour of complex machine learning and deep learning models while maintaining competitive detection performance [16], [17]. Nevertheless, integrating explainability into IDS introduces additional computational overhead, which may affect real-time deployment in latency-sensitive applications.

With the rapid expansion of cloud computing, Industrial IoT (IIoT), and edge computing, recent IDS research has increasingly focused on scalable, lightweight, and distributed security frameworks. Cloud-based intrusion detection systems require high scalability, flexibility, and adaptability to defend against evolving cyber threats in

dynamic cloud environments. Similarly, lightweight IDS architectures are essential for IoT networks, where computational power, memory, and energy resources are limited [11], [18], [19].

3. RESEARCH DESIGN

A. MOTIVATION

This section describes the overall design of the proposed study. Figure 1 shows the overall methodology. The XABiL-IDS (Explainable Attention-based Bi LSTM-Intrusion Detection System) integrates advanced ML and DL techniques with Explainable-AI (LIME and SHAP) to deliver a robust and transparent intrusion detection framework. By combining high detection accuracy with interpretability, the system addresses key limitations of traditional IDS approaches and provides a reliable solution for modern cybersecurity challenges.

The suggested XABiL-IDS will be designed that accurately detect network intrusion incidents with high interpretability. The suggested system architecture involves the combination of ML classification models along with various DL & XAI explanation techniques that allow enhancing transparency and trustworthiness of the solution. The general flow of the suggested solution consists of multiple interconnected modules such as data acquisition, data preprocessing, feature engineering, model training, predictions, and explainability layer along with decision support and visualization layers.

1. Data Acquisition and Preprocessing, the first step in the process of creating an intrusion detection system would involve the acquisition of the raw data from reliable data sources which can include both real-time monitoring applications and publicly available benchmark data sets like CICIDS. In this regard, data acquired during the process will likely include a mixture of numerical and categorical features, namely protocols used, duration of flows, packets size and characteristics of connections between clients and servers.
2. Data Preprocessing that ensures high-quality data processing, the necessary preprocessing technique must be used, which can include imputation of missing data, elimination of duplicates, categorical encoding, and normalization.
3. Feature Engineering involves extraction of relevant attributes from raw traffic datasets. Some of the features that are generated from the raw data include Packet Rate, Byte Rate, and Flow-based Statistics to capture better representations of network traffic. Feature Selection then follows by reducing the dimensionality of the extracted features as well as filtering out redundant features. Statistical Techniques such as Correlation Analysis and Model-based Techniques such as Feature Importance using Tree-based algorithms such as XGBoost, among others, are employed. This increases computational efficiency as well as increases the interpretation capability of the model.
4. Model Training and Classification in this, the models are trained on the traffic data so as to classify the traffic as either normal or attacks. The proposed model allows for several classifiers such as XGBoost for tabular data and BiLSTM Network with attention mechanism for sequential data. So the training dataset is represented as

$$, \qquad D = \{(x_i, y_i)\}_{i=1}^{N}, \qquad (1)$$

Where in equation 1 $x_i$denotes the feature vector and $y_i \in \{0,1\}$represents the corresponding class label, with 0 indicating normal instances and 1 indicating attack instances. The aim here is to develop a mapping function:

5. Feature Selection which reduces dimensionality in addition to removing redundant features. Various statistical techniques including correlation analysis, and feature importance through tree-based algorithms like XGBoost among others, are used. Not only it will reduce computation cost but also enhance the interpretability of the model.
6. Explainability Layer tackles the black box approach used by existing machine learning algorithms, this system will include an explainability layer with the use of current Explainable Artificial Intelligence or XAI. Specifically, the system will utilize SHAP and LIME.

SHAP calculates the contributions of a certain feature according to cooperative game theory. The Shapley value of a certain feature (i) is represented by the below equation:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f(S \cup \{i\}) - f(S)] \qquad (2)$$

where F is the complete set of features and S is a certain subset without feature i.

On the other hand, the technique used by LIME involves constructing an approximate interpretable model within a certain neighbourhood of a prediction in the following equation numbered 3:

$$g(x) \approx f(x) \qquad (3)$$

where g(x) is an interpretable model (such as a linear model).

These methods provide a means to visualize global and local interpretations, which help in analysing feature contributions and understanding the model's behaviour.

7. Decision Support and Visualization, in this the results obtained by the two previous modules will be visualized using an interactive visualization interface.
8. Visualization Layer helps cybersecurity experts make better decisions by offering detailed information on why a certain traffic flow falls under the category of being malicious or benign. This makes the process more trustworthy, accountable, and efficient.
9. Feedback and Model Refinement Process In order to maintain flexibility when dealing with the changing landscape of cyber security threats, the suggested IDS architecture has been designed to have a feedback loop. The security experts will be able to confirm whether the model predictions were accurate, and any corrections provided will be fed back into the model for updating or refinement.

B. PRELIMINARIES

This section discusses the basic building blocks of our model. These preliminaries provide the necessary fundamentals that would allow for the construction of the proposed XABIL IDS system. Through the use of intrusion detection methodologies, ML, DL and XAI approaches, and rigorous evaluation metrics, the basis is laid for developing an efficient IDS system. ML methods have become popular to improve IDS accuracy by allowing automatic recognition of complicated attacks. In case of a supervised learning-based IDS, the model learns from the labelled data set and predicts whether the traffic is normal or malicious, where x_i∈R^n is the input feature vector and y_i∈{0,1} represents the class labels.

The problem then becomes finding a function f(x)such that:

Some common ML models in IDS include decision trees algorithms, including eXtreme Gradient Boosting (XGBoost), and neural networks (NN) models such as Bidirectional Long Short-Term Memory (BiLSTM). XAI attempts to ensure ML models' interpretability and transparency.

Conventional ML models are considered black-box systems that offer results but do not justify their prediction process. This challenge is overcome by XAI through offering an explanation of the relationship between inputs and outputs. There are several methods employed in XAI, two of which are SHAP) and LIME are employed in this proposal. SHAP explains the contribution of each feature using cooperative game theory on the other hand LIME constructs a local surrogate model to explain each prediction.

Data Preprocessing

A. Normalization: The given expression represents the Z-score normalization (also called standardization) commonly used in statistics, machine learning, and data preprocessing.

$$x' = \frac{x-\mu}{\sigma} \qquad (3)$$

x= original data value

μ= mean of the dataset

σ= standard deviation of the dataset

x'= normalized (standardized) value

SMOTE: The equation represents the Synthetic Minority Oversampling Technique (SMOTE) used for balancing imbalanced datasets in machine learning.

$$x_{\text{new}} = x_i + \lambda(x_j - x_i) \qquad (4)$$

x_i= a minority class sample

x_j= one of the nearest minority neighbors of x_i

λ= random value between 0 and 1

x_{new}= newly generated synthetic sample

Correlation: The equation represents the Pearson Correlation Coefficient, which measures the strength and direction of the linear relationship between two variables.

$$\rho_{xy} = \frac{\mathrm{Cov}(x,y)}{\sigma_x \sigma_y} \quad (5)$$

ρ_{xy}= correlation coefficient between variables $x$and y

$\mathrm{Cov}(x,y)$= covariance between $x$and y

σ_x= standard deviation of x

σ_y= standard deviation of y

B. BiLSTM Model

The following equations represent the core operations of the Bidirectional Long Short-Term Memory (BiLSTM) architecture, widely used for sequential learning and temporal feature extraction in deep learning-based Intrusion Detection Systems (IDS).

Forget Gate

$$f_t = \sigma\big(W_f[h_{t-1}, x_t] + b_f\big) \quad (6)$$

$$f_t = \sigma\big(W_f[h_{t-1}, x_t] + b_f\big) \quad (7)$$

Input Gate

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (8)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (9)$$

Cell State Update

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (10)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (11)$$

C. Attention Mechanism

The following equation represents the Scaled Dot-Product Attention Mechanism, which is the foundational component of modern attention-based deep learning architectures used in NLP, IDS, IoT security, and trans former-enhanced neural networks.

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (12)$$

D. Explainability (SHAP)

The SHAP (SHapley Additive exPlanations) method is used in Explainable AI to understand how each feature contributes

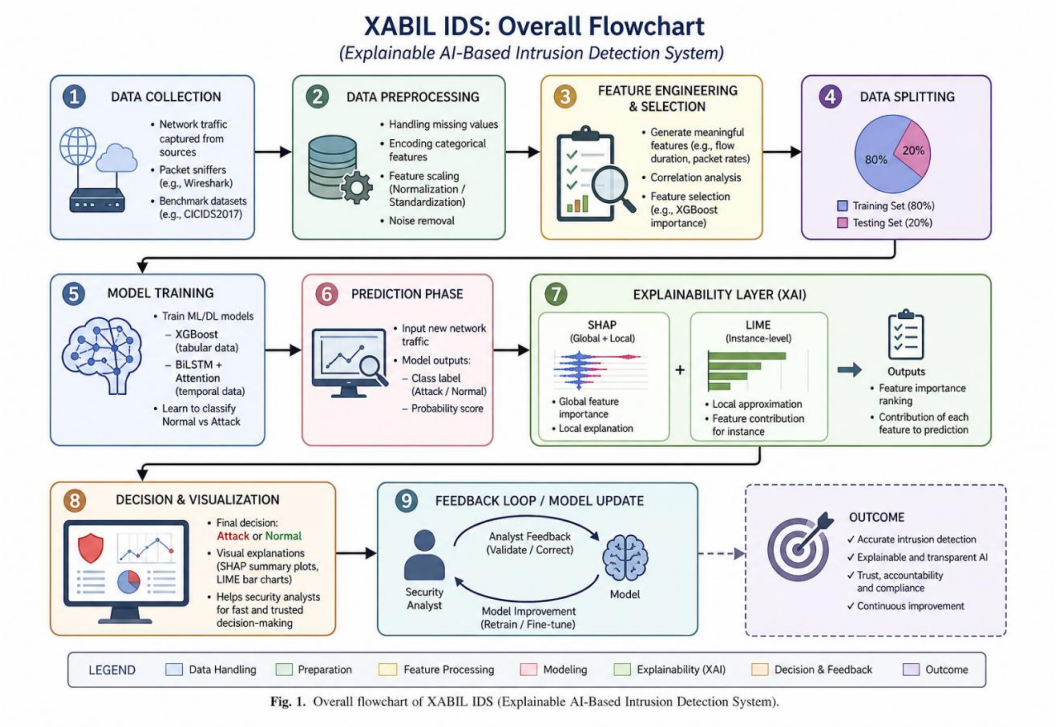

Fig. 1. Overall flowchart of XABIL IDS (Explainable AI-Based Intrusion Detection System).

FIGURE 1. Overall Flowchart of XABiL IDS used in the study

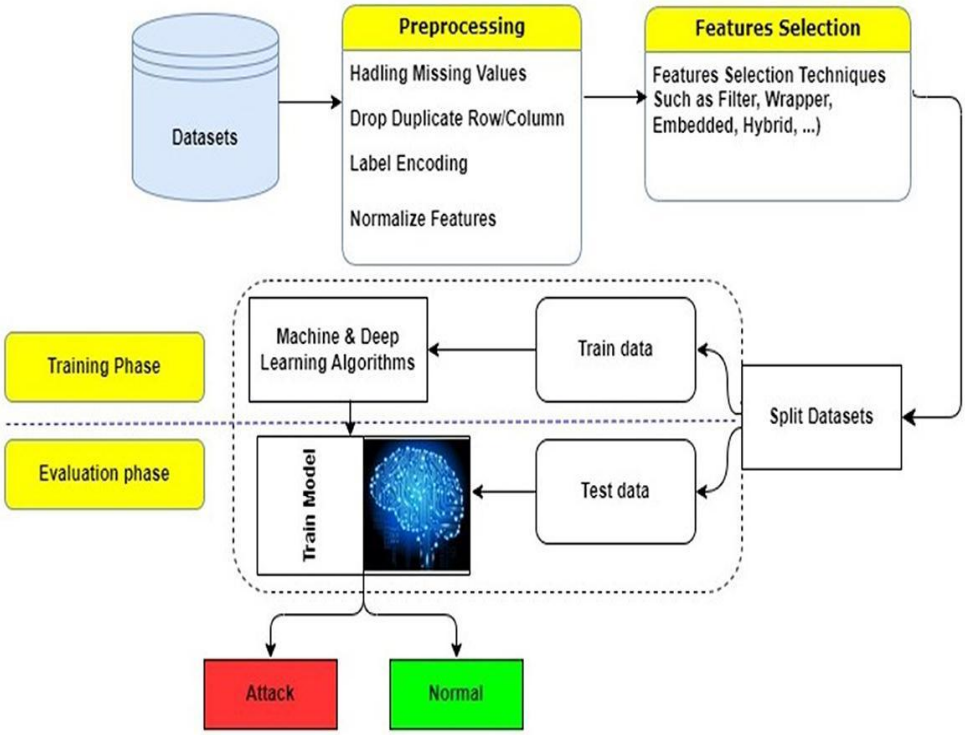

FIGURE 2. Methodology followed in the study

to a machine learning model's prediction. The SHAP explanation formula is:

$$f(x) = \phi_0 + \sum_i \phi_i$$

(13)

$f(x)\rightarrow$ Final prediction made by the model for input x

$\phi_0\rightarrow$ Base value or average prediction of the model

$\phi_i\rightarrow$ Contribution of the i^{th} feature to the prediction

$\sum\phi_i\rightarrow$ Total contribution from all features

4. PROPOSED FRAMEWORK

In this research proposal a novel model is made up named XABiL-IDF (Explainable Attention-based Bidirectional LSTM Intrusion Detection Framework). The study introduces a state-of-the-art technique that includes Explainable AI that includes SHAP & LIME, Attention-based mechanism, Bidirectional LSTM in Intrusion Detection Framework, which is based on preprocessing, hybrid deep learning techniques, and explainable AI methods. The proposed study is different from existing research where the emphasis was either on detection accuracy or explainability.

Proposed model XABiL-IDF attempts to balance both aspects by using a combination of BiLSTM, multi-head attention and SHAP/LIME methods for explanations.

A. Dataset Used

The dataset contains numerous flow-level statistical variables that are crucial for identifying anomalies or malicious activities in addition to these fundamental fields. Since the dataset provides both directional forward/backward and aggregated statistics, it is well suited for training and evaluating machine learning and deep learning-based IDS models.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Flow_ID	Src_IP	Src_Port	Dst_IP	Dst_Port	Protocol	Timestam	Flow_Dur	Tot_Fwd_	Tot_Bwd_	TotLen_Fv	TotLen_Bv	Fwd_Pkt_l	Fwd_Pkt_l	Fwd_Pkt_l	Fwd_Pkt_l	Bwd_Pkt_
2	192.168.0.	192.168.0.	10000	192.168.0.	10101	17	########	75	1	1	982	1430	982	982	982	0	1430
3	192.168.0.	222.160.1'	2179	192.168.0.	554	6	########	5310	1	2	0	0	0	0	0	0	0
4	192.168.0.	192.168.0.	52727	192.168.0.	9020	6	########	141	0	3	0	2806	0	0	0	0	1388
5	192.168.0.	192.168.0.	52964	192.168.0.	9020	6	########	151	0	2	0	2776	0	0	0	0	1388
6	192.168.0.	192.168.0.	36763	239.255.2'	1900	17	########	153	2	1	886	420	452	434	443	12.72792	420
7	192.168.0.	192.168.0.	41980	101.79.24	443	6	########	157	2	1	0	0	0	0	0	0	0
8	192.168.0.	192.168.0.	60175	210.89.16	8899	17	########	139	20	1	640	32	32	32	32	0	32
9	192.168.0.	192.168.0.	41467	58.225.75.	443	6	########	112	0	2	0	2896	0	0	0	0	1448
10	192.168.0.	192.168.0.	60132	210.89.16	8899	17	########	86	1	1	32	32	32	32	32	0	32
11	192.168.0.	111.149.1	7953	192.168.0.	554	6	########	6799	0	2	0	0	0	0	0	0	0
12	192.168.0.	192.168.0.	10102	40.100.49.	443	6	########	54	2	3	0	4076	0	0	0	0	1460
13	192.168.0.	104.118.1	443	192.168.0.	43238	6	########	165	1	1	1441	1441	1441	1441	1441	0	1441
14	192.168.0.	192.168.0.	53604	52.219.36.	443	6	########	199	2	1	2800	1400	1400	1400	1400	0	1400
15	192.168.0.	192.168.0.	10000	192.168.0.	10101	17	########	212	3	1	4290	1430	1430	1430	1430	0	1430
16	192.168.0.	192.168.0.	60079	210.89.16	8899	17	########	40	10	1	320	32	32	32	32	0	32
17	192.168.0.	192.168.0.	9020	192.168.0.	44144	6	########	60431	5	7	5680	1097	1388	1041	1136	144.4611	1097
18	192.168.0.	192.168.0.	56361	192.168.0.	10101	17	########	293	0	5	0	168	0	0	0	0	40
19	192.168.0.	192.168.0.	9020	192.168.0.	49784	6	########	120	1	1	30	1388	30	30	30	0	1388

FIGURE 4. Snapshot of CICIDS-2018 Dataset

B. EXPERIMENTAL SETUP

Figure 5 represents the class distribution of the CICIDS2018 dataset being used in this research paper. It can be seen that dataset is severely imbalanced in favor of the Anomaly class compared to the Normal class. Indeed, anomalies dominate the data set, with close to 580,000 records, while the number of normal records is closer to 40,000. Thus, there is an almost 14:1 ratio of anomalies to normal. Class imbalance is one of the key issues in using ML and DL algorithms to detect intrusions, because classifiers will inevitably become biased towards the class that dominates the dataset in terms of numbers when trained. As a consequence, high classification accuracy of a model will not necessarily guarantee its ability to correctly classify instances from minority classes, which may lead to inaccurate classification in practice. There are several approaches to handling class imbalance in datasets. Resampling techniques such as oversampling the minority class or under sampling the majority class can be applied to bring some balance back. Another approach includes focusing on other metrics of classification quality besides accuracy, such as precision, recall, and F1 score. The proposed study uses Synthetic Minority Over-sampling Technique (SMOTE). To balance the distribution of the dataset, SMOTE is used as a preprocessing technique. It creates synthetic instances by interpolating feature values across nearby minority-class observations, in contrast to traditional oversampling techniques that

duplicate current minority samples. This method reduces overfitting brought on by duplicate records while improving the representation of underrepresented attack classes.

When compared to innocuous traffic, minority attack categories are often underrepresented in IDS datasets like the CICIDS datasets. Consequently, machine learning models may exhibit high overall accuracy while failing to effectively detect rare cyberattacks. By incorporating SMOTE during the preprocessing stage, the proposed model gains improved exposure to minority attack patterns, leading to enhanced classification capability, higher detection rates, and improved recall and F1-score.

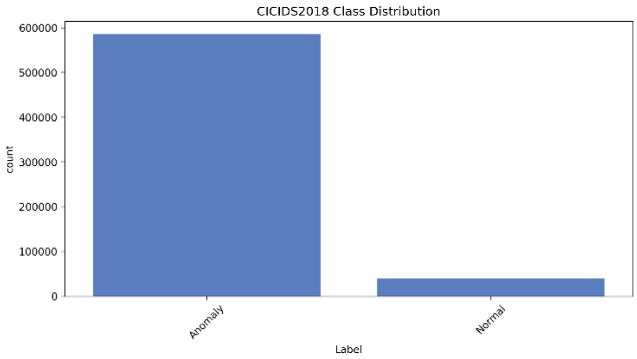

Figure 5. class distribution of the CICIDS2018 dataset

Figure 6 presents the correlation matrix of the extracted features selected from the CICIDS2018 data set. As it can be seen from the presented heat map, the matrix represents the coefficients of pairwise Pearson correlations for the selected traffic features. It can be noticed that some of the traffic characteristics have high positive correlations.

There are high correlations between Fwd Pkt Len Max, Fwd Pkt Len Mean, and Fwd Pkt Len Std, which are all packet length–related features. Likewise, there are high correlations between backward packet length characteristics, namely

Bwd Pkt Len Max, Bwd Pkt Len Mean, and Bwd Pkt Len Std. This shows that these features have high similarity in their measurements and might cause redundancy in the feature set.

Moreover, there are traffic features, such as Total Forward Packets and Total Backward Packets, which are moderately correlated with the packet length characteristics. On the contrary, some of the features have low correlations with most of the traffic characteristics, including the Protocol and Flow Duration features, indicating their independent contribution to the predictive model.

The presence of features having high correlations with each other can create multicollinearity problems for the predictive model and, thus, make it prone to overfitting.

The analysis reveals the presence of both strong positive and negative correlations among certain attributes. High interdependency is particularly evident in packet-length-based features, indicating possible redundancy in the dataset. These findings support the use of correlation analysis as a preprocessing step to find important features and reduce superfluous feature overlap. The suggested approach enhances computational efficiency and fortifies the intrusion detection model's learning capacity by eliminating redundancy and maintaining discriminative information. In order to ensure that the suggested IDS model learns from significant, non-overlapping traffic features rather than highly linked duplicate information, this figure is supplied to support feature relevance and redundancy analysis.

C.EVALUATION METRICS

There are some key performance measures that have been outlined as follows:

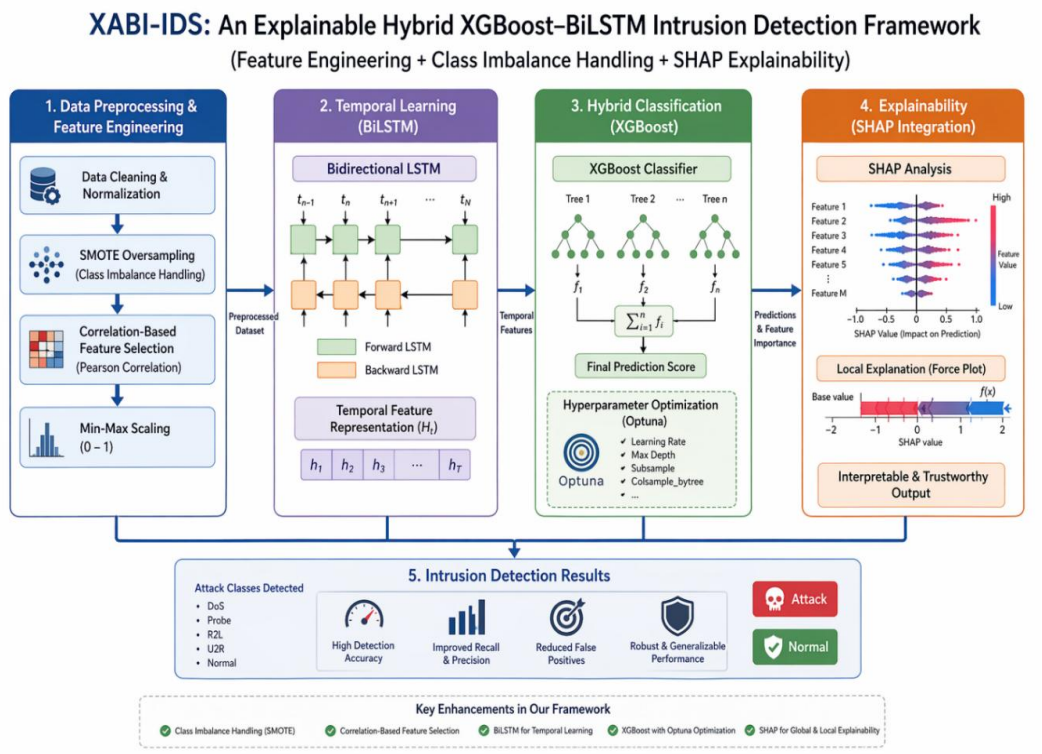

FIGURE 3. XABiL-IDS Framework

These include precision, recall, F1-score, and accuracy. These are defined as: TP refers to true positives, FP refers to false positives, TN is true negatives, and FN stands for false negatives.

D. BASELINE MODELS

Figure 7 depicts the comparison of the ten most relevant attributes as identified by the Random Forest (RF) and XGBoost classifiers on the CICIDS dataset. As seen, the scores associated with each attribute demonstrate their importance to intrusion detection and classification. In case of the Random Forest (RF) classifier, Dst_Port proves to be the most informative attribute for attacks detection with its high importance score of approximately 0.18. This means that port number on the destination node is a key characteristic for detecting anomalies. The second and third positions are held by Init_Bwd_Win_Byts and Src_Port with almost equally high importance (~0.095). Bidirectional communication statistics, thus, is an important parameter. Also, certain temporal and statistical features show their significance, such as ACK_Flag_Cnt and Flow_Duration with relatively high importance. Other lower ranking attributes include Flow_IAT_Mean, Flow_IAT_Max, and Idle_Max which are also informative but with considerably lower scores.

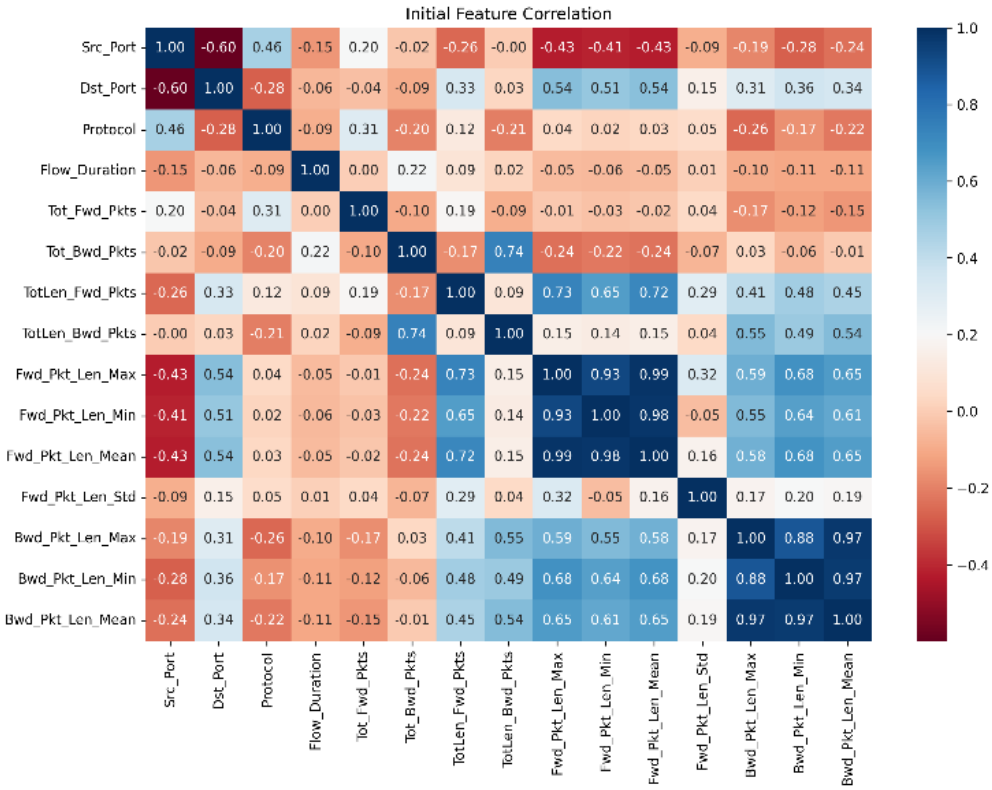

Figure 6. presents the correlation matrix of the extracted features

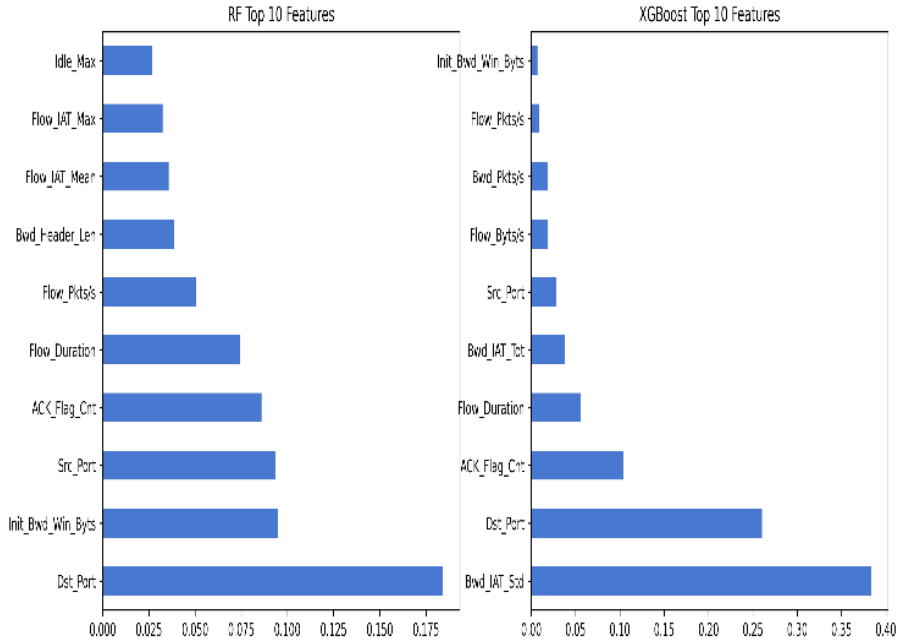

Figure 7. comparison of the ten most relevant attributes as identified by the Random Forest (RF) and XGBoost classifiers

The importance distribution in the XGBoost classifier is quite different. It can be seen that Bwd_IAT_Std takes the leading place with much higher importance score (~0.39). Variability in backward inter-arrival times appears to be an important discriminating factor for intrusions. The second position goes to Dst_Port with high importance as well

(~0.26). ACK_Flag_Cnt and Flow_Duration maintain moderate importance, while other features such as **Flow_Byts/s**, **Bwd_Pkts/s**, and **Flow_Pkts/s** contribute marginally. The comparative analysis has shown that Dst_Port, ACK_Flag_Cnt, and Flow_Duration are always rated highly by the two algorithms, thereby proving that they are highly reliable measures of detecting network intrusions.

However, while XGBoost focuses heavily on Bwd_IAT_Std, RF uses multiple variables to reach conclusions, implying that nonlinear connections can be better detected using XGBoost and RF, respectively. The conclusion from this comparison has been that the incorporation of port variables, temporal data, and flow statistics is crucial when detecting network intrusions.

Based on the results obtained, one can conclude about a very high concentration of the distribution of importance values, where the highest values correspond to Dst_Port with the importance score around 0.46. Therefore, according to the research results, destination port numbers appear to be a crucial factor for the machine learning model used to classify normal and attack traffic.

The second feature that has an essential impact is the Init_Bwd_Win_Byts (~0.28). It should be noted that this particular parameter emphasizes the significance of backward window size when evaluating flow control actions. Furthermore, Bwd_Pkts/s (~0.18) and Src_Port (~0.08) demonstrate moderate levels of influence on the classification of packets in terms of their nature. Finally, other features such as Flow_Duration, Bwd_IAT_Tot, Bwd_Header_Len, Idle_Max, Bwd_IAT_Max, and Flow_IAT_Max appear to be rather unimportant according to the results, implying the proper filtering of unnecessary information. In comparison with other traditional ML models such as Random Forest (RF) and XGBoost, the use of attention mechanisms proves to be more selective and focused on the important data.

E. EXPERIMENTAL RESULTS

Figure 9 gives the confusion matrices for the XGBoost and BiLSTM models tested against the CICIDS dataset in binary classification. The confusion matrices give an elaborate account of the performance of each model concerning the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

With respect to the XGBoost model, the confusion matrix shows that there are 7,935 correctly identified TNs while 80 normal samples are wrongly classified as attacks (FP).

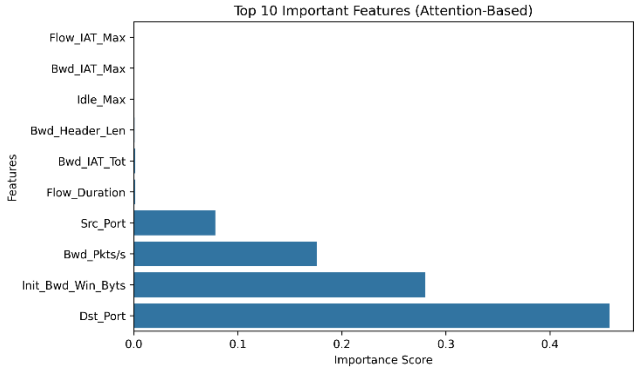

Figure 8. presents the top ten most important extracted features that are attention based

With regards to the attacks, the confusion matrix gives 117,082 as the number of correctly identified TP, while there are 60 attack samples wrongly classified as normal (FN). The confusion matrix shows a remarkable detection capability since the value of FP and FN is very low. Thus, precision and recall of the model are both high.

On the other hand, the BiLSTM model performs poorly compared to the XGBoost model with respect to the detection capabilities and misclassification rate. In that case, the number of correctly classified TNs is 4,951, whereas the wrongly classified TNs as attacks is 3,064. Also, for the attacks, 113,537 TP were correctly classified as attacks, but there were 3,605 samples wrongly classified as normal.

In terms of comparative analysis, it can be seen that the XGBoost method proves to be much superior to BiLSTM in all important parameters. The first method has near-perfect classification with zero false positive and false negative errors; meanwhile, BiLSTM shows a trade-off between these two indicators. Thus, a higher number of false positive predictions from BiLSTM may result in an increase in the number of false alerts, while a high number of false

negatives indicates missed attacks. So, to conclude tree-based ensemble approaches, such as XGBoost, prove to be the most efficient, while other approaches, including BiLSTM, require additional improvements.

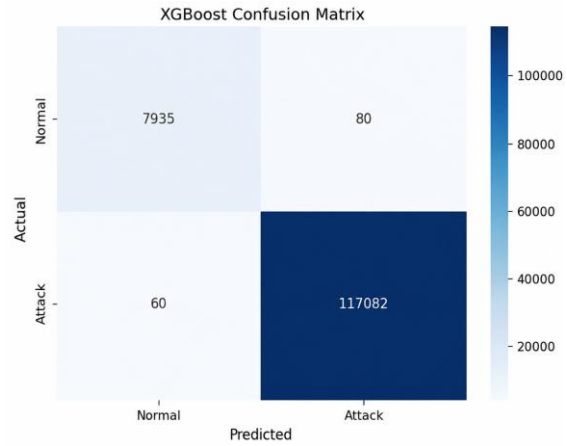

Figure 9. gives the confusion matrices for the XGBoost model

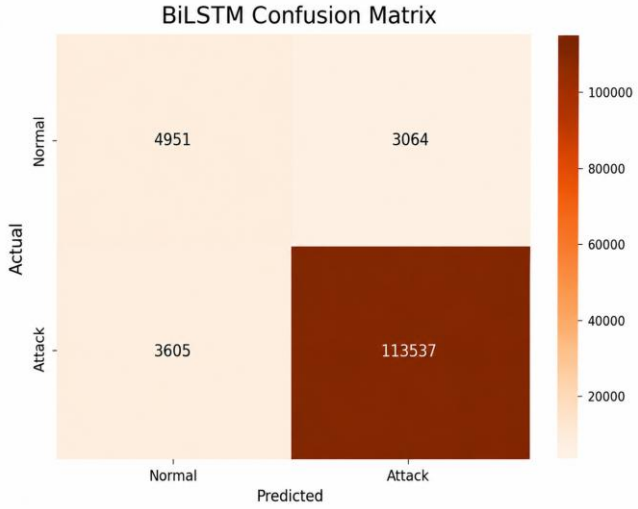

Figure 10. represents the confusion matrices for BiLSTM model

Figure 11 shows the comparison between ROC curves of the three models XGBoost, Dense Neural Network (DNN), and BiLSTM trained on the CICIDS dataset. In the ROC curve, the TPR vs FPR trade-off is observed, whereas AUC stands for a numerical measurement of classification accuracy.

XGBoost is able to reach almost perfect results by obtaining an AUC of 0.9999. As one can see from the figure, the ROC curve of XGBoost nearly reaches the top left corner of the graph, meaning very good discriminative ability, where nearly all true cases are detected as such with practically zero false detections. Such performance level proves the efficiency of using gradient boosting in structured intrusion detection.

The DNN is characterized by the same high accuracy level as the XGBoost with an AUC of 0.9972. The difference is that the DNN's ROC curve does not touch the optimal point; however, it is still close enough to it. This means that nonlinear structures are well detected by the DNN but somewhat less accurately compared to XGBoost. On the other hand, the BiLSTM model performs considerably worse, with an AUC score of 0.4843, below the baseline level of accuracy expected from random guessing (AUC score = 0.5). The ROC plot is near the diagonal line of reference, implying that the model lacks the ability to discriminate between the normal and malicious samples.

In summary, the results of the comparison analysis indicate that the XGBoost algorithm consistently achieves superior performance than the other two models. Although the Deep Neural Network is still among the top performers,

the BiLSTM model lags behind significantly, suggesting that sequence models are not well-suited for tabular IDS datasets.

Figure 12 presents a comparative evaluation of three models—XGBoost, Dense Neural Network (DNN), and BiLSTM—based on four key performance metrics: Accuracy, Precision, Recall, and F1-score, computed on the CICIDS dataset. From the above analysis, the XGBoost algorithm can be regarded as having performed optimally on the dataset since the values attained by all the evaluation criteria are extremely high.

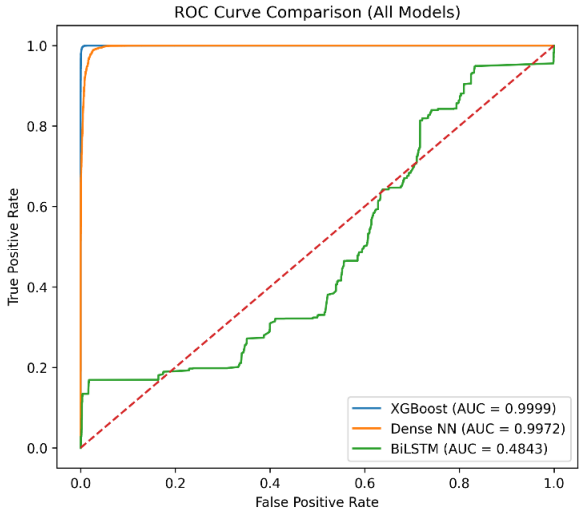

Figure 11 shows the comparison between ROC curves of the three models XGBoost, Dense Neural Network (DNN), and BiLSTM

The accuracy, precision, recall, and F1-score for the classification algorithm almost attain 1.0 in value which shows the efficiency of the technique when applied for intrusion detection. All the evaluation criteria have almost similar results, making it an optimal algorithm for classification purposes.

On the same note, the Dense Neural Network (DNN) model has performed similarly to the XGBoost algorithm since all the values attained are closer to 1.0. Even though the values are slightly lower than the former, they still indicate that the neural network has been able to capture the nonlinearity in the dataset.

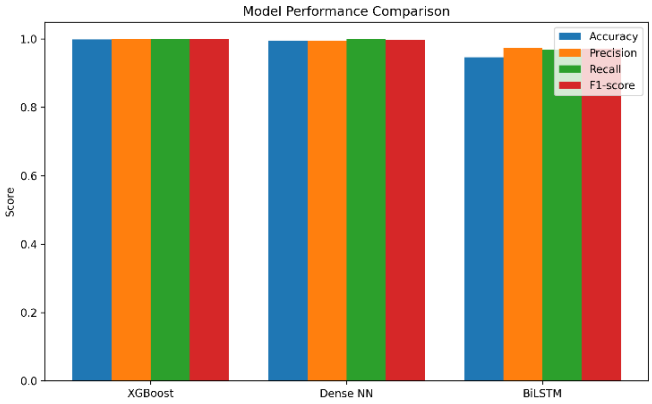

Figure 12. presents a comparative evaluation of three models: XGBoost, Dense Neural Network (DNN), and BiLSTM based on four key performance metrics: Accuracy, Precision, Recall, and F1-score

On the other hand, the BiLSTM model has performed relatively poorly compared to XGBoost and DNN. The accuracy value for this model stands at 0.95. In addition, the precision, recall, and F1-score values attained by this model have slightly lower values than XGBoost and DNN models. In general, through the comparative analysis, it can be found that the XGBoost algorithm performs better than all other machine learning algorithms, followed by DNN. Though the BiLSTM is an efficient algorithm, yet it comes third in terms of performance as it performs slightly

worse than others due to the relatively poor performance of classification. From this, it is concluded that tree-based and feedforward neural network algorithms are ideal for structured datasets while BiLSTM needs optimization.

MODEL ENHANCEMENT USING SHAP AND LIME

Global Feature Importance using SHAP: The left subfigure shows the global feature importance computed through SHAP (Shapley Additive Explanations). Features have been sorted according to their average absolute impact on the model output. In this case, Dst_Port turns out to be the feature with maximum importance as destination port value is the most effective in distinguishing benign and malicious traffic. Following Dst_Port, there are two more important features, Flow_Duration and Src_Port, which play an important role in contributing to the model performance. Other important features include Init_Bwd_Win_Byts and ACK_Flag_Cnt, whereas temporal and statistical features such as Bwd_IAT_Tot, Flow_IAT_Mean, and Pkt_Len_Max are of lesser importance in comparison.

Local Explanation based on LIME: The right side of the figure shows the results of LIME, which is Local Interpretable Model Agnostic Explanations to explain a particular instance locally. In the graph, the bars indicate how each feature contributed to the predicted output of the classifier, wherein the green bar implies a push towards an attack label, while the red bar implies a push towards the normal label. The most relevant local feature would be when Init_Bwd_Win_Byts > 1869 since this strongly impacts the prediction towards normal. Likewise, Bwd_IAT_Tot > 119 and ACK Flag Cnt also negatively contribute to normal behavior. On the other hand, Flow_IAT_Mean ≤ 73 and Flow_IAT_Min ≤ 72 are indicative of attacks, meaning shorter inter-arrival times suggest abnormal behavior.

From the analysis, there is a crucial difference in the interpretability of global versus local models: While SHAP highlights the dominance of Dst_Port in global interpretation, the LIME analysis suggests that its significance in the decision process for a particular observation might be relatively lower. Features such as temporal variables (inter-arrival time), which have little significance in global interpretability, become important in making decisions in local instances. The model behaves differently in different contexts, where the importance of features differs for each observation. This figure indicates that while global interpretation methods like SHAP give a broad insight into how the model works, the local explanation method LIME is necessary for analysing the specific decision process for observations.

F. DISCUSSION

However, there exists an inherent trade-off between complexity and interpretability of a model. Although XGBoost performs better, it lacks interpretability and hence requires external explainers like SHAP, whereas BiLSTM with attention allows for partial interpretability based on attention weights but not at the feature level. With the use of interpretability, XABiL is able to overcome the challenge and achieve good performance along with interpretable models. This aspect is essential in intrusion detection applications, especially in the field of cybersecurity.

From a practical standpoint, some of key implications of the analysis are Firstly port and flow duration information need to be considered first because of their importance globally. Secondly instance-level information about traffic should not be overlooked because it may provide useful information for detecting attacks. Lastly XAI tools can help security analysts in justifying the decision of the model.

5. CONCLUSION AND FUTURE WORK

First of all, the model is mainly validated against the CICIDS dataset. While commonly used as a testing dataset for intrusion detection algorithms, this particular dataset cannot reflect the dynamics and variations inherent in the real-world network environment. Namely, network traffic is highly dynamic, frequently encrypted, and constantly exposed to innovative approaches and techniques employed by cybercriminals. At the same time, datasets used for benchmarking are usually produced in laboratory settings under strictly defined conditions. Hence, patterns learned from benchmark datasets do not apply equally well in live systems and, especially, in zero-day attacks.

Secondly, the feature space used in CICIDS is perfectly structured and tailored to ML and DL algorithms. Nevertheless, in real-life conditions, network traffic may not be that well-formulated and may require substantial processing before analysis. Therefore, an approach that involves high quality and carefully pre-selected features may not be feasible without feature engineering and, thus, may be less convenient for practical implementation in a live environment.

Another aspect to pay attention to is the computational complexity of explanation approaches. Namely, SHAP is consistent and solid; however, being computation-intensive, this algorithm cannot be applied in large-scale and

complicated datasets. Similarly, LIME, based on several perturbations of the input data. This makes both methods less suitable for real-time or high-throughput intrusion detection systems, where decisions must be made within milliseconds.

Moreover, there is a natural trade-off between complexity and explainability. Even though the proposed architecture enhances detection capabilities, it also raises complexity levels. The results from SHAP and LIME are valuable, but they can also be ambiguous and need to be interpreted using some expertise in the field. For cybersecurity experts who lack experience in machine learning, the interpretation of SHAP and LIME results might be challenging.

Finally, the present study is concentrated mainly on evaluating the performance of the proposed approach through traditional criteria like accuracy, precision, recall,

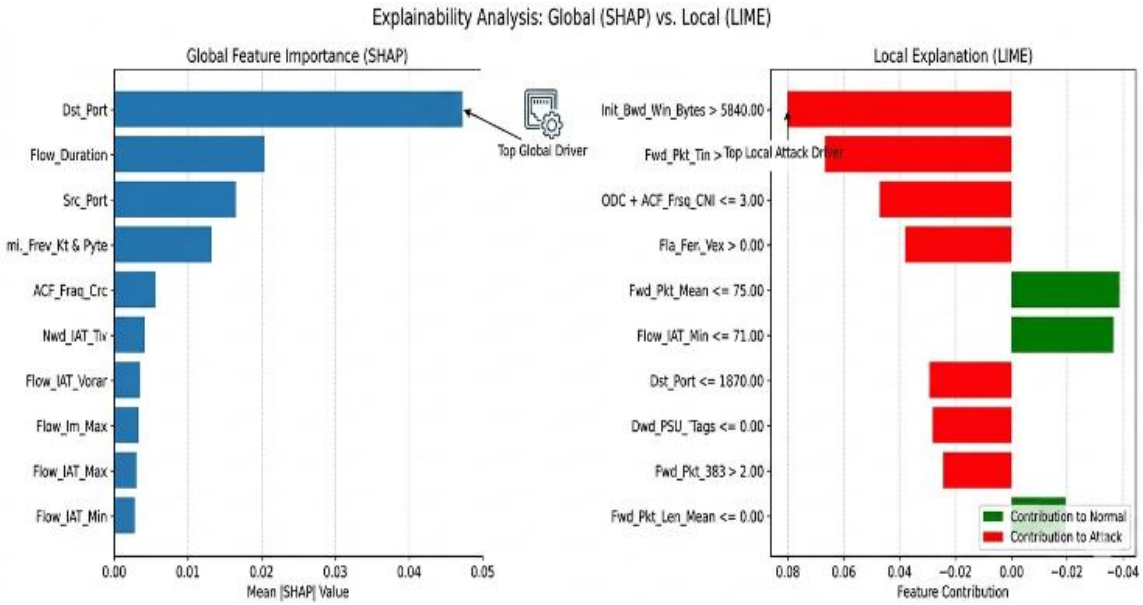

Figure 13. shows the comparison and applicability of SHAP and LIME

and F1-score. Although the results show that the model performs well in terms of classification, it does not provide any information about operational parameters like detection speed, scaling capabilities, and robustness against adversarial attacks.Through traditional criteria like accuracy, precision, recall, and F1-score. Although the results show that the model performs well in terms of classification, it does not provide any information about operational parameters like detection speed, scaling capabilities, and robustness against adversarial attacks.

Directions for further research that could alleviate the aforementioned issues have been presented. First of all, there exists an evident need for increasing the model generalization capacity. Such an objective can be accomplished through training and testing the model on a variety of different data sets, such as traffic captures from the real world and datasets from other domains. Specifically, employing datasets containing encrypted traffic, data collected in IoT networks, or cloud infrastructures will yield greater insight into the problem. Secondly, there is a possibility to optimize explainability algorithms for use in time-sensitive settings. Further studies could consider developing an approximation-based version of SHAP, which would speed up computations without deteriorating the explanation quality. Moreover, one can try implementing a faster alternative to LIME or designing an algorithm that would incorporate both explanations and would work within limited timeframes. Lastly, there is the option to implement online learning schemes into the algorithm itself. In real-world situations, the environment does not remain constant, and any model may gradually become obsolete. Hence, it would be advantageous if the model could adjust itself to the new reality and update its parameters accordingly.

Moreover, future research can be devoted to designing self-explainable models that limit the need for post hoc interpretability techniques. Attention-based models can be improved to allow for better human-understandable explanations. This will increase the model's utility for security analysts and avoid using resource-intensive methods. Scalability and deployability issues should be addressed in future work as well. Scaling the model to work in distributed settings, in the context of edge computing, or by utilizing specialized hardware such as GPUs and TPUs can make it

applicable to larger networks. The model's efficiency and performance in terms of latency and throughput should also be considered.

Finally, adversarial attacks should be taken into account when designing future systems. Attackers might try to circumvent the intrusion detection system by tampering with the features. Defense mechanisms can be developed to counteract such adversarial threats.

References

1. M. Learning, "Intrusion Detection in Internet of Things Systems : A Review on Design Approaches Leveraging Multi-Access Edge," 2022.
2. M. B. Bankó et al., "Advancements in Machine Learning-Based Intrusion Detection in IoT : Research Trends and Challenges," 2025.
3. Z. Xu, Y. Wu, S. Wang, J. Gao, T. Qiu, and Z. Wang, "Deep Learning-based Intrusion Detection Systems: A Survey," J. ACM, vol. 1, no. 1, pp. 1–38, 2025.
4. A. Kumar, B. Arnob, R. R. Chowdhury, N. A. Chaiti, S. Saha, and A. Roy, "A comprehensive systematic review of intrusion detection systems : emerging techniques , challenges , and future research directions," vol. 4, pp. 73–104, 2025.
5. I. Bibers, O. Arreche, and W. Alayed, "Ensemble-IDS : An Ensemble Learning Framework for Enhancing AI-Based Network Intrusion Detection Tasks," pp. 1–37, 2025.
6. K. Ileri, "Comparative analysis of CatBoost , LightGBM , XGBoost , RF , and DT methods optimised with PSO to estimate the number of k - barriers for intrusion detection in wireless sensor networks," Int. J. Mach. Learn. Cybern., vol. 16, no. 9, pp. 6937–6956, 2025, doi: 10.1007/s13042-025-02654-5.
7. V. Z. Mohale and I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI : enhancing transparency and interpretability," no. May, pp. 1–23, 2025, doi: 10.3389/fcomp.2025.1520741.
8. Y. Hosain, "XAI-XGBoost : an innovative explainable intrusion detection approach for securing internet of medical things systems," pp. 1–17, 2025.
9. S. M. Othman, A. Y. Al-mutawkkil, and M. Alnashi, "Survey of Intrusion Detection Techniques in Cloud Computing," vol. 2, no. 4, pp. 363–374, 2024.
10. A. Ghaffari, N. Jelodari, S. Pouralish, N. Derakhshanfard, and B. Arasteh, "Securing Internet of Things using machine and deep learning methods: A survey," Cluster Computing, vol. 27, pp. 9065–9089, 2024. doi: 10.1007/s10586-024-04509-0.
11. M. M. Rahman, S. A. Shakil, and M. R. Mustakim, "A survey on intrusion detection system in IoT networks," Cyber Security and Applications, vol. 3, Art. no. 100082, 2025. doi: 10.1016/j.csa.2024.100082.
12. L. Tonke, D. K. Yadav, and M. Bargadiya, "A Systematic Review of Machine Learning-Based Models for IoT Network Security in Intrusion Detection Systems," Journal of Information Systems Engineering and Management, vol. 9, no. 4s, 2024.doi: 10.52783/jisem.v9i4s.14758.
13. Q. A. Al-Haija and A. Droos, "A comprehensive survey on deep learning-based intrusion detection systems in Internet of Things (IoT)," Expert Systems, vol. 42, no. 2, e13726, 2025. doi: 10.1111/exsy.13726.
14. Z. Xu, Y. Wu, S. Wang, J. Gao, T. Qiu, Z. Wang, H. Wan, and X. Zhao, "Deep Learning-based Intrusion Detection Systems: A Survey," arXiv, 2025.
15. M. Alrashdi et al., "Landscape of learning techniques for intrusion detection system in IoT: A systematic literature review," Computers & Electrical Engineering, vol. 122, Art. no. 109725, 2025. doi: 10.1016/j.compeleceng.2024.109725.
16. X. Zhang et al., "A comprehensive survey on intrusion detection algorithms," Computers & Electrical Engineering, vol. 121, Art. no. 109863, 2025. doi: 10.1016/j.compeleceng.2024.109863.
17. A. Komal and S. Li, "Intrusion Detection in the Internet of Things: A Comprehensive Review of Techniques, Architectures, Datasets, and Emerging Trends," Sensors, vol. 26, no. 11, Art. no. 3405, 2026. doi: 10.3390/s26113405.
18. A. Amouri, M. M. Al Rahhal, Y. Bazi, I. Butun, and I. Mahgoub, "Enhancing Intrusion Detection in IoT Environments: An Advanced Ensemble Approach Using Kolmogorov-Arnold Networks," arXiv, 2024.
19. M. A. Akif, I. Butun, A. Williams, and I. Mahgoub, "Hybrid Machine Learning Models for Intrusion Detection in IoT: Leveraging a Real-World IoT Dataset," arXiv, 2025.