Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.052599

ARTICLE

Explainable AI-Based DDoS Attacks Classification Using Deep Transfer Learning

Ahmad Alzu'bi[1,*], Amjad Albashayreh[2], Abdelrahman Abuarqoub[3] and Mai A. M. Alfawair[4]

[1]Department of Computer Science, Jordan University of Science and Technology, Irbid, 22110, Jordan

[2]Department of Computer Science, The University of Jordan, Amman, 11942, Jordan

[3]Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff, CF5 2YB, UK

[4]Prince Abdullah bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University, Salt, 19117, Jordan

*Corresponding Author: Ahmad Alzu'bi. Email: agalazubi@just.edu.jo

Received: 08 April 2024 Accepted: 24 June 2024 Published: 12 September 2024

ABSTRACT

In the era of the Internet of Things (IoT), the proliferation of connected devices has raised security concerns, increasing the risk of intrusions into diverse systems. Despite the convenience and efficiency offered by IoT technology, the growing number of IoT devices escalates the likelihood of attacks, emphasizing the need for robust security tools to automatically detect and explain threats. This paper introduces a deep learning methodology for detecting and classifying distributed denial of service (DDoS) attacks, addressing a significant security concern within IoT environments. An effective procedure of deep transfer learning is applied to utilize deep learning backbones, which is then evaluated on two benchmarking datasets of DDoS attacks in terms of accuracy and time complexity. By leveraging several deep architectures, the study conducts thorough binary and multiclass experiments, each varying in the complexity of classifying attack types and demonstrating real-world scenarios. Additionally, this study employs an explainable artificial intelligence (XAI) AI technique to elucidate the contribution of extracted features in the process of attack detection. The experimental results demonstrate the effectiveness of the proposed method, achieving a recall of 99.39% by the XAI bidirectional long short-term memory (XAI-BiLSTM) model.

KEYWORDS

DDoS attack classification; deep learning; explainable AI; cybersecurity

1 Introduction

The advent of the Internet of Things (IoT) has heightened cybersecurity research by integrating numerous devices into networks to deliver complicated services. By 2025, it is estimated that there will be 75 billion smart devices, which will transform everyday life [1]. The transportation, healthcare, manufacturing, and agriculture industries, among others, are becoming increasingly dependent on IoT technology [2]. This technology is rapidly expanding and connecting a wide range of products, from businesses and residences to transportation. It is reshaping daily duties and how people conduct their personal and professional activities, potentially decreasing the demand for human labor while

increasing the intelligence of our daily lives [3]. However, the rapid expansion of IoT devices poses substantial security issues due to their interconnected nature, sensors, and massive data creation. Also, they are numerous, diverse, need little computational power, and typically operate at the periphery of computer networks. These devices frequently have fundamental vulnerabilities because of their limited computational capability, affordable design, and lack of regular security upgrades [4,5]. As a result, consumers are more vulnerable to cyberattacks. Different manufacturers' security requirements can lead to new types of attacks on IoT systems, despite numerous security measures in place. Connecting IoT devices to an unprotected network exposes them to a variety of threats, even if they are otherwise safe. Therefore, it must preserve user privacy while combating cyberattacks such as distributed denial of service (DDoS), which change over time and pose new risks on a daily basis. The complexity of the number of IoT devices and networks allows attackers to transform basic devices into destructive botnets to launch potentially damaging attacks [6]. Traditional cybersecurity techniques frequently focus on protecting against local attacks or breaches inside a limited network environment. Yet, the environment of IoT offers a new level of complexity.

IoT attacks present new challenges that surpass the capabilities of traditional security measures. These attacks can vary from minor invasions of privacy to complex, systematic attacks on interconnected networks [7]. IoT system attacks are more widespread and severe than local transmission attacks, which are limited to nodes near a small domain, causing significant damage [8,9]. The crucial necessity to secure user privacy and prevent more sophisticated assaults drives the urgency of addressing IoT security challenges. Unlike isolated breaches, IoT assaults may have far-reaching implications, affecting not only individual users but whole networks and infrastructures. As a result, the focus should be on developing robust security procedures tailored to the intricate nature of IoT systems, necessitating advanced detection techniques. Deep learning (DL) models are particularly well-suited for identifying subtle attack patterns in network traffic. To address concerns about the opaque nature of artificial intelligence (AI), explainable AI (XAI) techniques can be employed [10], improving transparency and interpretability. By integrating deep learning models with explainable AI, transparent DDoS detection systems can be developed to mitigate risks and facilitate informed decision-making and response actions. Such an approach fosters trust among stakeholders and enhances DDoS detection in dynamic and complex environments, where the impact of these attacks can be most severe.

This study aims to introduce a framework for identifying incoming DDoS attacks through deep learning models with an effective transfer learning mechanism that relies on the interpretations of attack features. We evaluate two datasets sourced predominantly from the IoT network, featuring diverse network flows. Additionally, the study seeks to elucidate the generated predictions by analyzing the data features' contribution to the decision-making process using XAI techniques. The main contribution of this study is three-fold:

- A DDoS detection framework is introduced to identify DDoS attacks in IoT environments automatically. The proposed deep learning model formulates generic discriminating descriptors of network data with various pre-trained deep backbones to scrutinize and classify potential threats in network traffic.
- Local interpretable model-agnostic explanations are provided to explain the decision-making process of deep learning models with effective model fine-tuning, which increases the transparency and reliability of the detection process.
- In addition to accuracy measurements, the performance of the DDoS detector is evaluated in terms of time complexity, which is done by rigorously conducted experiments demonstrating

a range of potential threat scenarios, i.e., binary network flows, detection of 8 attacks, and detection of 34 attacks.

The rest of this paper is organized as follows: Section 2 reviews the prior work; the methodology of the DDoS detection system is presented in Section 3; Section 4 discusses the experimental results; and Section 5 concludes this paper.

2 Related Work

This section presents recent research work dedicated to detecting DDoS attacks in IoT networks and explaining DDoS features utilizing XAI techniques.

2.1 DDoS in IoT Networks

In recent years, researchers have dedicated substantial efforts to examining attacks within IoT networks, utilizing traditional statistical methods and machine learning algorithms to differentiate between normal activity and malicious behavior. Various machine learning and DL models were utilized to consider diverse features of DDoS flows using the Canadian Institute for Cybersecurity IoT (CICIoT2023) dataset [11], leading to varying results due to the utilization of different techniques. Abbas et al. [3] presented a unique technique for detecting large IoT device threats via federated learning, which employs a deep neural network for accurate classification. Sharmin et al. [12] examined reconnaissance assaults on IoT devices utilizing time-based features and flag qualities and employed Bayesian optimization to pick a representative sample for the best flow duration range.

Wang et al. [13] proposed a hybrid intrusion detection model that combines deep neural network (DNN) and bidirectional long short-term memory (BiLSTM) to develop a lightweight IoT intrusion identification system. The model reduces feature dimensionality, extracts nonlinear and bidirectional long-range features, and dynamically quantifies its unit structure. Khan et al. [14] investigated the utilization of supervised machine learning algorithms to identify abnormal behavior by applying the synthetic minority over-sampling technique (SMOTE). The outcomes revealed that Random Forest is the most efficient model in comparison to prior works. Additionally, the authors found that removing highly correlated features improves performance but reduces computational response time. Yaras et al. [15] developed a hybrid deep learning algorithm using convolutional neural network (CNN) and long short-term memory (LSTM) models to detect DDoS attacks. The proposed model was tested on the CICIoT2023 and telemetry of network_IoT (ToN_IoT) [16]. Table 1 provides a summary of the previous research conducted on the CICIoT2023 dataset.

Table 1: A summary of the previous research on CICIoT2023

Ref.	Technique	Task	Recall (%)
[3]	Federated learning with DNN	2-classes	99.00
[12]	Multiclass model	8-classes	88.00
		34-classes	70.00
[13]	DL-BiLSTM	8-classes	93.13

(Continued)

Table 1 (continued)

Ref.	Technique	Task	Recall (%)
[14]	Random Forest (RF)	2-classes	99.49
		8-classes	95.52
		34-classes	96.54
[15]	Hybrid deep learning	2-classes	99.99
		9-classes	99.96

2.2 DDoS Explanation

Previous studies have employed various XAI techniques to elucidate DDoS attacks, with the goal of enhancing the understanding of attack patterns, behaviors, and detection mechanisms. Gyamfi et al. [17] utilized low-cost IoT sensors for effective intrusion detection, using a network of cameras to record location information. They performed compatibility checks between datasets and features, integrating them to create a new IoT dataset using an explainable AI technique. Hasan et al. [18] developed an explainable ensemble DL based intrusion detection system (IDS) framework that improves the transparency and robustness of DL-based IDSs in IoT networks. The framework's efficacy was evaluated using the ToN_IoT dataset and extreme learning machines model. Bashaiwth et al. [19] studied the LSTM predictions on CIC datasets using local interpretable model-agnostic explanations (LIME) [20], shapley additive explanations (SHAP), Anchor, and local rule-based explanations (LORE) techniques. They conducted binary and multiclass classifications. The binary classification demonstrated high performance across all three datasets, whereas the multiclass classification showed good performance only for the first two versions. However, the LSTM model struggled to differentiate between certain attacks, resulting in poor classification performance. Hassan et al. [21] utilized machine learning techniques to detect malicious traffic data in vehicle ad hoc networks (VANETs) and proposed an IDS capable of identifying threats from 14 types of malicious attacks. Wei et al. [22] proposed a framework for detecting normal and malicious DDoS attack traffic, utilizing Kernel SHAP to understand the multilayer perceptron (MLP) classifier prediction results. Tabassun et al. [23] used XAI techniques such as SHAP, LIME, and explain like I'm 5 (ELI5) to classify DDoS attacks in IoT networks using machine learning and deep learning models. The results reveal that SHAP offers both local and global explanations, LIME provides local explanations, and ELI5 highlights important features. Antwarg et al. [24] employed Kenal SHAP to explain anomalies in the knowledge discovery in databases (KDD) dataset using autoencoder's unsupervised model. The approach explained the influence of low and high reconstruction error characteristics, proving its resilience in comparison to existing LIME explanation methods.

Senevirathna et al. [25] proposed a new framework for scaffolding attacks in security contexts, combining XAI outputs with domain knowledge to identify target features. The approach identifies essential aspects an attacker would conceal while building a model and presents an effective attack detection method. The authors utilized various models including MLP, support vector machine (SVM), RF, LSTM, gaussian naive bayes (GNB), and k-nearest neighbour (KNN), with MLP achieving an F1-score of 99.40. Arreche et al. [26] developed a framework for evaluating black-box XAI algorithms for network intrusion detection. The framework evaluates global and local scopes, examining six metrics for SHAP and LIME, including network security and AI. It is being tested with three datasets and seven AI algorithms adaptive (ADA), LSTM, DNN, light gradient boosting

machine (LGBM), MLP, RF, and KNN, serving as a baseline for network security. The authors achieved a recall score of 99.98 using the KNN model. Do et al. [27] utilized XAI algorithms such as LIME, SHAP, gradient-weighted class activation mapping (Grad-CAM), and guided backpropagation (GBP) to analyze network traffic patterns, distinguishing between malicious and benign connections. The authors empirically found that XAI algorithms like SHAP and LIME can identify complex correlations between characteristics and anomalies, enabling precise identification of benign traffic. The authors used different models such as RF and CNN and achieved a recall score equal to 99.90. Table 2 summarizes previous research that utilized explainable AI techniques to explain DDoS attacks.

Table 2: A summary of previous research utilized XAI to explain DDoS attacks

Ref.	Model	Explainable-AI	Dataset
[17]	XGBoost	TreeSHAP	IoTID20
[18]	CNN	SHAP, LIME	ToN_IoT
[19]	LSTM	LIME, SHAP, Anchor, and LORE	CICDDoS2017/2018/2019
[21]	RF	SHAP, LIME	IIoT
[22]	MLP	Kernel SHAP	CICDDoS2019
[23]	Decision tree	SHAP, ELI5, and LIME	Artificial dataset
[24]	Autoencoder	Kenal SHAP	NSL-KDD
[25]	MLP	SHAP	5GNIDD, NLS-KDD
[26]	KNN	SHAP, LIME	RoEduNetSIMARGL2021 CICIDS-2017, NSL-KDD
[27]	RF	LIME, SHAP, Grad-CAM, and GBP	MQTTset, CICIDS-2017

However, there is still a demand for handling the impact and utilization of any explained features in the training procedure, providing effective feedback for the training engine, i.e., linear regression in our study. We demonstrate in this work how several deep architectures can interpret and explain the extracted features from network traffic and update the weights accordingly. Additionally, the time complexity of model training is calculated and discussed–usually neglected in the existing approaches. We have also performed more evaluation experiments on a new dataset that has been released recently with diverse traffic data, including unknown attacks, demonstrating real-world scenarios.

3 Methodology

3.1 DDoS Evaluation Datasets

This paper examines DoS/DDoS attacks using the recent intrusion data for DDoS detection systems, primarily based on the CICIoT2023 [11] and CICDDOS2019 [28] datasets, chosen due to their diverse range of DDoS network flows, demonstrating real-world scenarios and posing significant detection challenges.

3.1.1 CICIoT2023 Dataset

The CICIoT2023 dataset [11] is a comprehensive IoT attack dataset that was released to advance the development of security analytics applications within real IoT environments. It includes 33 distinct attacks across seven categories, including DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and

Mirai, executed within an IoT network consisting of 105 devices. DDoS includes acknowledgment (ACK) fragmentation, User Datagram Protocol (UDP) flood, SlowLoris, internet control message protocol (ICMP) flood, reset finish (RSTFIN) flood, PSH acknowledgment (PSHACK) flood, hypertext transfer protocol (HTTP) flood, UDP fragmentation, transmission control protocol (TCP) flood, synchronize (SYN) flood, and SynonymousIP flood. DoS includes TCP flood, HTTP flood, SYN flood, and UDP flood. Brute Force includes Dictionary brute force, Spoofing includes address resolution protocol (ARP) spoofing and domain name system (DNS) spoofing. Recon includes Ping sweep, operating system (OS) scan, Vulnerability scan, Port scan, and Host discovery. Web-based includes structured query language (SQL) injection, Command injection, Backdoor malware, Uploading attacks, cross-site scripting (XSS), and Browser hijacking. Mirai includes generic routing encapsulation internet protocol (GRE-IP) flood, Greeth flood, and UDPPlain.

The CICIoT2023 experiment investigates the utilization of IoT devices in smart home environments, with 105 devices participating in the attacks. The topology is separated into two components: a router that connects the network to the Internet with a Windows 10 desktop computer, and a Cisco switch that connects seven Raspberry Pi devices. These devices carry out assaults and criminal behaviors, exhibiting a unique feature of CICIoT2023. The Cisco switch is linked to the second component via a Gigamon Network Tap, which captures all IoT traffic and routes it to two network monitors. These monitors use Wireshark to store traffic, allowing for full-duplex, non-intrusive, and passive access to network traffic without interfering with routine operations. The device includes two networks and two monitoring ports, with one connected to attackers and the other to victims' networks. A network tap and two traffic monitors are used to monitor network traffic, with each packet saved on different computers. Wireshark monitors network activity, which is saved in pcap format. Mergecap combines pcap files for each experiment. Each assault is unique on all relevant devices, targeting rogue IoT devices in all circumstances. This technique helps assess potential risks and vulnerabilities in IoT systems. Tables 3 and 4 present data statistics for binary classification and eight-class classification tasks, respectively. Fig. 1 also depicts the data statistics for 34-classes in the multiclassification experiment.

Table 3: The data statistics for the binary classification task

Class	Training	Validation	Testing
Attack	3,487,879	387,542	968,856
Benign	84,021	9336	23,339

Table 4: The data statistics for the (8-classes) classification task

Class	Training	Validation	Testing
DDoS	2,601,180	289,020	722,550
DoS	618,159	68,685	171,711
Mirai	201,316	22,368	55,921
Benign	84,021	9336	23,339
Spoofing	37,356	4151	10,377

(Continued)

Table 4 (continued)

Class	Training	Validation	Testing
Recon	27,026	3003	7508
Web	1867	207	518
Brute force	975	108	271

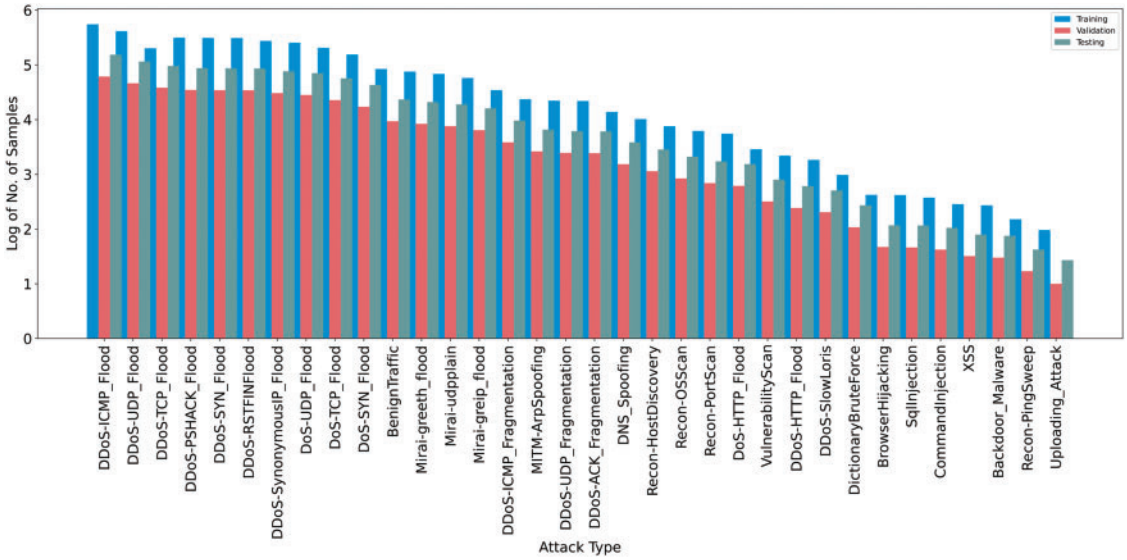

Figure 1: The data statistics for the (34-classes) classification task

3.1.2 CICDDoS2019

The CICDDoS2019 dataset [28] was proposed by the Canadian Institute of Cybersecurity as a new version of CICDDoS2017 and CICDDoS2018. The dataset was created using the B-Profile technology, which profiles abstract human interactions and generates real benign background traffic. It incorporates 25 users' abstract behavior based on HTTP, HTTPS, FTP, SSH, and email protocols to provide realistic background traffic. This dataset consists of benign and DDoS network flows that match real-world data. It contains several current DDoS reflection attacks, including such as Port Map, lightweight directory access protocol (LDAP), network basic input/output system (NetBIOS), UDP, Microsoft SQL server (MSSQL), UDP-Lag, SYN, DNS, network time protocol (NTP), simple network management protocol (SNMP) and more, making it an excellent choice for accurately reflecting the current environment, as many outdated datasets are no longer functional. Table 5 shows the statistics of the CICDDoS2019 dataset.

3.2 The Generic Framework of DDoS Detection

This paper proposes a new systematic framework for DDoS detection to classify the network flows based on the purpose of its occurrence, whether it is a natural induction or a malicious attack, this framework consists of five stages: data preparation, feature engineering, data split, deep learning

models, and LIME explanation. Fig. 2 displays the major phases adopted in the proposed DDoS detection framework, which are detailed in the following subsections.

Table 5: The CICDDoS2019 statistics

Class	Training	Validation	Testing
Attack	240,148	26,683	66,709
Benign	70,438	7827	19,566

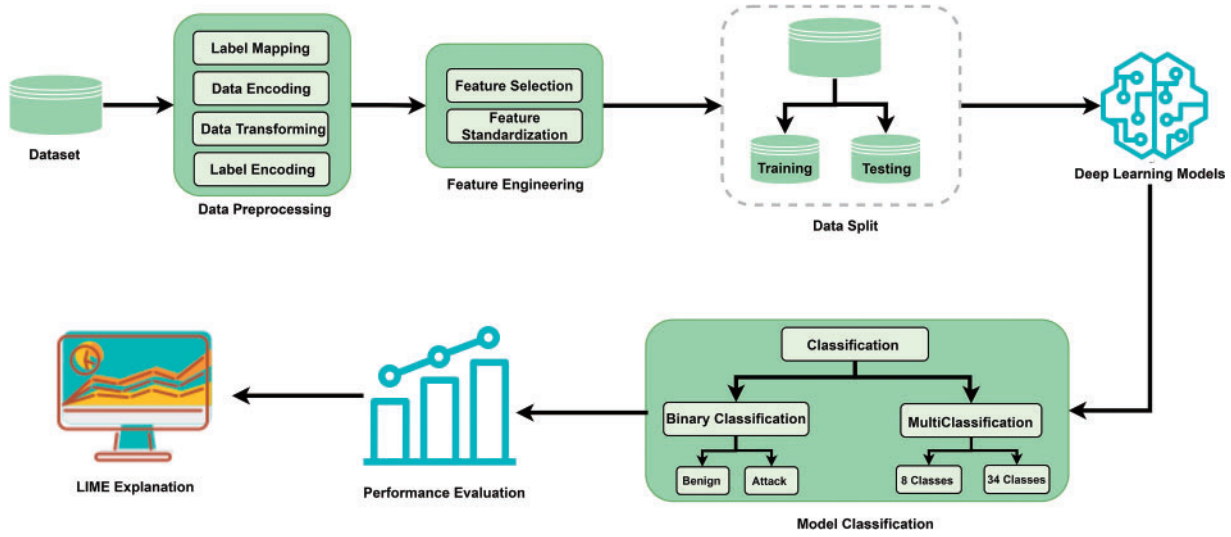

Figure 2: The generic pipeline of the DDoS detector

3.2.1 Data Preparation

To ensure the dataset is adequately prepared for training and evaluating the deep learning model, several steps were undertaken. This included extracting a subset of network flows from the original data source, which contains both benign and attack flows. All the obtained network flows have been merged and stored in one comma-separated value (CSV) file, and then label mapping has been done to map each network flow to its exact attack type. All the categorical columns have been encoded. After that, the raw data was converted into a format that the model could use. The original 34 labels were simplified by grouping related classes and assigning new labels to reduce complexity. In two experiments, the attacks were mapped into eight attacks and one general label to classify network flows to attack or begin. This approach reduced the complexity of the classification task and made the process more manageable. Also, label encoding is used to transform categorical data into a numerical representation that deep learning models can understand. Based on the mapping procedure, each unique label is allocated a unique integer.

Due to the huge amount of data, twenty-one CSV files with 4,960,973 network flows are obtained from the original data and combined in one source to train and evaluate the deep learning models. The data set is divided into three parts: 70% for training, 10% for validation, and 20% for testing. Due to the use of imbalanced data, the stratification technique is used to guarantee that models are trained and assessed on subsets of data that accurately represent the overall distribution of classes.

3.2.2 Feature Engineering

In this work, the procedure of feature engineering includes feature selection and feature standardization. The feature selection step is done manually to select the same features that were obtained by the authors of CICIoT2023 without implementing any mathematical technique. Feature standardization is used for data scaling by changing the distribution of characteristics to a standard scale with a mean of 0 and a standard deviation of 1. This paper used the standard scaler method to guarantee that numerical features contribute equally to the model's learning process and reduce the influence of differing scales on the performance of the algorithms. The standard scaler transformation for a feature $\boldsymbol{\mathcal{X}}$ is defined in Eq. (1), where $\boldsymbol{Z}$ is the standardized value, $\boldsymbol{\mathcal{X}}$ is the original feature, μ is the mean of the feature, and σ is the standard deviation of the feature.

$$\boldsymbol{Z} = \frac{\mathcal{X} - \mu}{\sigma} \tag{1}$$

3.2.3 Deep Learning Backbone Models

This paper employs various deep learning backbones in DDoS detection to provide a reliable and comprehensive approach for analyzing network traffic data. Extensive experiments were performed for the procedure of transfer learning using four different pre-trained deep learning models, which are BiLSTM [29], CNN [30], gated recurrent units (GRU) [31], and RNN [32]. Because each of these models has distinct architectural features, it is possible to thoroughly examine how well they can address the particular difficulties presented by this research. Utilizing different deep learning models validates the dataset network flows and their outcomes across multiple methodologies. As a result, organizations can improve the effectiveness and reliability of their DDoS detection systems, ultimately enhancing their defenses against cyber threats. These models are used in this paper due to their architectures which have unique capabilities that make them suitable for DDoS detection tasks compared to other deep learning and machine learning techniques. They can handle sequential data, learn hierarchical features, preserve memory and context, adapt to evolving patterns, and leverage ensemble learning. Fig. 3 shows a visual representation of the operational flow of these models at every stage, including both the training and testing phases. It details the entire process from the input of data to the ultimate decision-making.

3.2.4 Feature Explanation

To explain the model prediction and determine the feature contribution, LIME is used. LIME provides an interpretable and accurate explanation of classifier predictions by learning a local interpretable model around the prediction [20]. LIME justifies supervised learning model predictions on a variety of data formats, including text and images. It computes essential characteristics around a given instance and creates 5000 feature vector normal distribution samples. This technique works by looking for target variables for a specific number of samples and assigning weights to each row based on how close it is to the original data label. Also, it determines the important features by using feature selection techniques such as lasso and principal component analysis (PCA). This technique has been successfully implemented in XAI for image, text, and tabular data. In this paper, the LIME method is applied for DDOS network flows, which represent tabular data to determine the contribution of each feature in predicting the correct attack type. We used a LIME-based XAI procedure to analyze the contribution of each feature in the traffic data. This helped us to understand how each feature influenced the final decision and to identify any biases or incorrect decisions in the predicted classes of network attacks. A thorough investigation has demonstrated that LIME is the most effective method

for describing traffic data qualities and supporting any prediction generated by a supervised learning model, which has been also demonstrated in previous empirical studies [33].

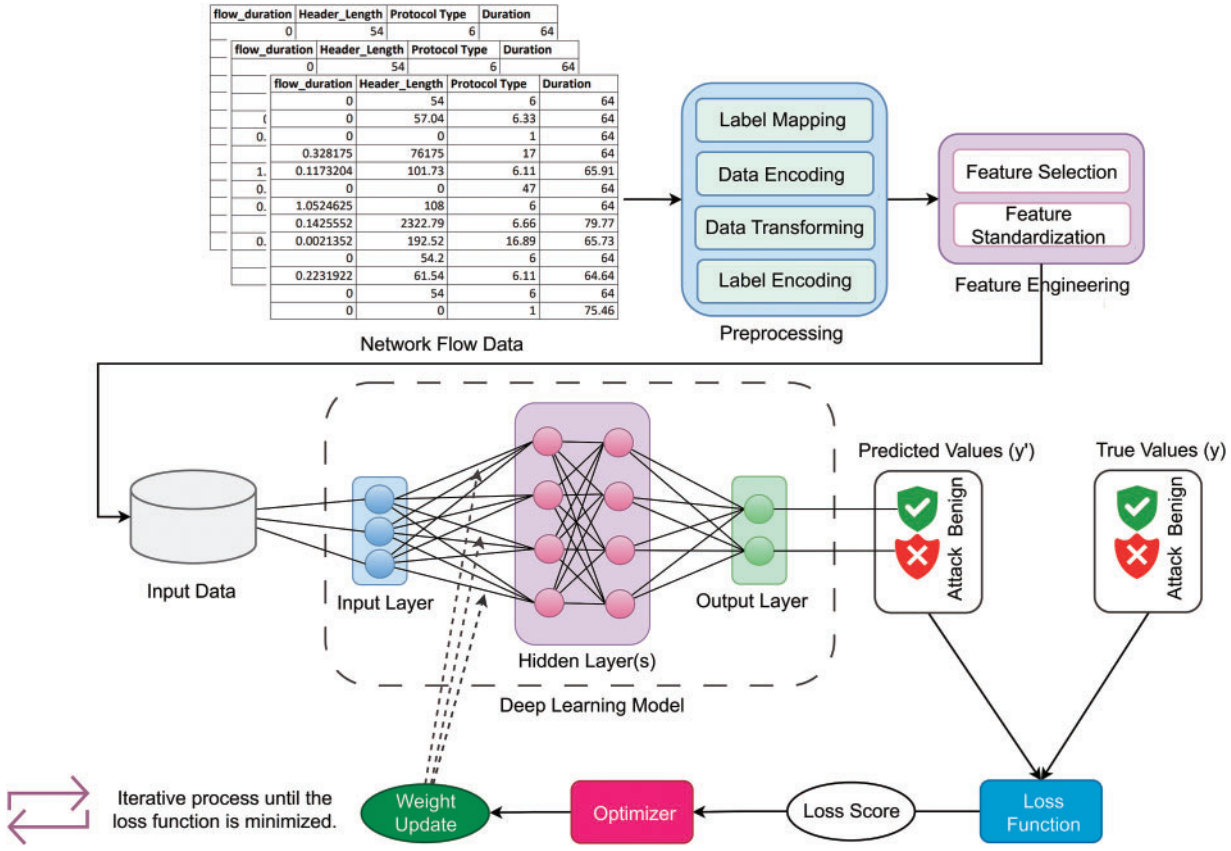

Figure 3: Visual representation of neural network learning process

4 Experimental Results and Discussion

4.1 Experimental Setup

The experiments were conducted on a professional cloud processing platform equipped with powerful GPUs and CPUs. Several Python and ML libraries were used for implementing the proposed DDoS classifier, e.g., TensorFlow and Keras. The specifications of the client machine used to initiate and control the experiments include Nvidia T4 Tensor Core GPU, Intel core i7 of 3.4 GHz, and RAM of 12 gigabytes (GB). The experiments were performed using identical hyperparameters under the same configuration. The models' performance is assessed based on training time complexity and final prediction results. The binary classification experiment uses binary cross entropy as a loss function, and the sigmoid as an activation function for the output layer. Also, the Adam optimizer is used with a learning rate equal to 0.001. A batch size of 128 is used during five epochs, with an early stopping equal to 3. On the other hand, in the multiclassification tasks, categorical cross-entropy is used as a loss function, and SoftMax is applied with a batch size of 64 for ten epochs. Table 6 shows the hyperparameters utilized in each experiment.

Table 6: The list of hyper-parameters used in all training experiments

Hyper-Parameter	Binary Classification	Multi Classification
Loss function	Binary cross-entropy	Categorical cross-entropy
Activation function	Sigmoid	SoftMax
Optimizer	Adam	Adam
Epochs	5	10
Learning rate	0.001	0.001
Batch size	128	64
Early stopping	True	True

Four distinct standard metrics are used to evaluate the performance of deep learning models, which are accuracy, recall, precision, and F1-score. The metrics are calculated in terms of macro average and weighted average. The weighted averaging metrics tend to the majority class and affect the final decision in the prediction process; therefore, to ensure the validity of the performance results for the used models, we measured their performance using macro averaging, which is not affected by the majority class since it handles all the labels equally by distributing equal weights for each label.

This study focuses on the recall metric, a key metric in DDoS attack detection, to evaluate a model's ability to accurately identify all DDoS network flows, aiming to maximize true positives and minimize false negatives in network flow classification. High recall is essential in DDoS classification due to the significant cost and damage caused by false negatives. It is also important in this work because of the imbalanced class distribution in the dataset. The model often accurately identifies the majority class but fails to identify the infrequent minority class. However, recall is not affected by the imbalanced distribution since a high recall score ensures accurate classification.

4.2 Classification Results of DDoS Attacks

A thorough analysis of DDoS attacks is presented using various deep learning models. Three experiments are conducted based on the attack type. The first experiment is conducted to detect and classify the network flows based on their general nature, whether they are benign or attack flows. The second experiment presents more detailed results by classifying the detected attacks into eight types. The third experiment provided more details by classifying them into 34 different attack types. In the context of DDoS detection, true positive (TP) represents the correctly classified attacks, true negative (TN) represents the correctly classified non-attacks, false positive (FP) defines non-attacks incorrectly classified as attacks, and false negative (FN) defines attacks incorrectly detected as non-attacks. The primary target for this study is achieving a high recall since the recall represents the ratio of correctly classified DDoS attacks.

Recall estimates the proportion of positive samples identified by a model among all positive samples, aiming to capture as many attacks as possible while minimizing missed attacks. Optimizing for recall ensures the model effectively detects most DDoS attacks, even if it indicates tolerating some false positives that represent normal traffic inaccurately classified as attacks. However, four different deep learning models are used in each experiment, BiLSTM, GRU, RNN, and CNN. Tables 7–9 demonstrate the detailed results of the deep learning models in each experiment. In the binary classification experiment, the BiLSTM model outperformed all the other models with an accuracy of

99.40, precision of 99.44, recall of 99.39, and F1-score of 99.41 in the weighted averaging measurement. Regarding the macro-averaging, it achieved a precision of 91.50, a recall of 96.21, and an F1-score of 93.72.

Table 7: Summary of the binary classification results

Model	Val Acc	Test Acc	Weighted Average			Macro Average		
			P	R	F	P	R	F
BiLSTM	**99.40**	**99.39**	**99.44**	**99.39**	**99.41**	91.50	**96.21**	**93.72**
GRU	99.37	**99.39**	99.42	**99.39**	99.40	91.91	95.56	93.66
RNN	99.33	99.35	99.36	99.35	99.35	**92.92**	93.04	92.98
CNN	99.35	99.35	99.39	99.35	99.36	91.19	95.49	93.23

Table 8: Summary of multiclassification results (8-classes)

Model	Val Acc	Test Acc	Weighted Average			Macro Average		
			P	R	F	P	R	F
BiLSTM	**99.03**	**99.04**	**99.07**	**99.04**	**98.97**	90.87	**66.79**	**69.62**
GRU	99.00	**98.98**	99.03	98.98	98.91	**92.53**	66.07	68.87
RNN	96.86	96.86	96.91	96.86	96.72	79.42	63.36	66.47
CNN	99.07	99.00	99.12	99.00	98.92	89.32	65.32	68.35

Table 9: Summary of multiclassification results (34-classes)

Model	Val Acc	Test Acc	Weighted Average			Macro Average		
			P	R	F	P	R	F
BiLSTM	**98.44**	**98.43**	98.35	**98.43**	**98.23**	71.18	**65.17**	**65.35**
GRU	96.95	98.13	98.10	98.13	97.94	71.88	64.68	64.73
RNN	94.64	95.98	96.04	95.98	95.75	68.95	60.90	61.11
CNN	98.32	97.87	**98.90**	97.87	98.17	**76.11**	64.28	64.41

In the second experiment, it achieved an accuracy of 99.04, a weighted precision of 99.07, a weighted recall of 99.04, a weighted F1-score of 98.97, a macro precision of 90.87, a macro recall of 66.79, and a macro F1-score of 69.62. Also, in the 34-class experiment, it got an accuracy of 98.43, a weighted precision of 98.35, a weighted recall of 98.43, a weighted F1-score of 98.23, a macro precision of 71.18, a macro recall of 65.17, and a macro F1-score of 65.35. The second and third experiments showed a decrease in macro recall due to the complexity of identifying the exact type of DDoS attack, as the model was burdened by the detailed attack type. This paper validated the findings obtained from the CICIoT2023 dataset by evaluating the models using a high-quality dataset collected from real-world scenarios. This dataset is particularly challenging as it serves as a testbed for assessing the

algorithms' capability to detect network attack flows. We used the CICDDoS2019 dataset to evaluate the selected models under identical settings, employing consistent hyperparameters for comparison.

The Bi-LSTM model exhibited exceptional performance, achieving an accuracy of 99.82. Additionally, it demonstrated a recall rate of 99.82, a precision score of 99.82, and an F1-score of 99.82. These results highlight the robustness and high quality of the model's performance across various metrics. Table 10 demonstrates the detailed results of the deep learning models. After validating the results, the models are compared based on their training time complexity. As shown in Table 11, the CNN model outperforms the other models in detecting DDoS attacks, demonstrating superior efficiency with minimal training time.

Table 10: Summary of the classification results on CICDDOS2019

Model	Val Acc	Test Acc	Weighted Average			Macro Average		
			P	R	F	P	R	F
BiLSTM	**99.77**	**99.82**	**99.82**	**99.82**	**99.82**	**99.65**	**99.84**	**99.74**
GRU	99.69	99.73	99.73	99.73	99.73	99.49	99.73	99.61
RNN	99.70	99.77	99.77	99.77	99.77	99.56	99.77	99.67
CNN	99.72	99.71	99.71	99.71	99.71	99.43	99.74	99.58

Table 11: A summary of training time complexity

Experiment	Model	Step	Epoch	Training time (minutes)
2-classes	BiLSTM	6 ms	174 s	14.5
	GRU	5 ms	135 s	11.25
	RNN	**3 ms**	**96 s**	**8**
	CNN	8 ms	229 s	19
8-classes	BiLSTM	12 ms	655 s	109
	GRU	7 ms	374	62
	RNN	5 ms	269 s	44.8
	CNN	**4 ms**	**225 s**	**26.25**
34-classes	BiLSTM	9 ms	530 s	80.3
	GRU	7 ms	414 s	69
	RNN	6 ms	321 s	53
	CNN	**8 ms**	**229 s**	**19**
CICDDoS2019	BiLSTM	7 ms	17 s	1.41
	GRU	6 ms	15 s	1.25
	RNN	5 ms	12 s	1
	CNN	**4 ms**	**10 s**	**<1**

BiLSTM and GRU models need additional training time because of their complex architectures and increased processing demands. As the number of classes increases, there will be more training time

for all models. Nevertheless, the importance differs per model. For example, with the BiLSTM model, training time increases dramatically as the number of classes increases, but the CNN model has rather steady training times across varied class distributions.

Training time varied throughout the experiments, with RNN taking the least time in the 2-classes experiment, followed by GRU and BiLSTM, and CNN taking the longest. However, the model architecture plays a significant role in determining training time complexity. Variations in model parameter values, such as step size and epoch, can also influence training time complexity. This demonstrates the tradeoff between high performance and accurate results, which necessitates an extensive amount of time for training the model. However, the results indicate that incorporating explainable AI techniques into network security systems allows organizations to understand complex decisions made by AI models, improving the transparency and reliability of automated defense mechanisms. This builds trust in AI-powered security solutions. Understanding how AI models differentiate between normal network traffic and malicious attacks enables proactive response strategies, reducing DDoS attacks and downtime. This reduces the risk of misidentifying events as attacks and unauthorized resource usage. It also enhances the system's ability to detect insider threats and abnormal activities.

4.3 The Impact of Feature Explanation on DDoS Detection

The LIME-based XAI technique is employed to explain the contribution of each feature in the decision-making process during the prediction phase. The XAI technique utilizes tabular data through four steps: generating perturbed instances, predicting using a black-box model, fitting an interpretable model, and explaining the prediction. In the first step, it produces perturbed instances from the original data, randomly perturbing certain attributes while maintaining others constant. Then, the black-box model is used to predict labels for perturbed instances, allowing for a better understanding of how changes in feature values affect the model's predictions. Then, an interpretable model, in our case linear regression, is applied to the altered instances and their accompanying predictions, serving as a surrogate of the black-box model's behavior at the closest distance to the original data. LIME does not directly deal with optimization but focuses on finding an interpretable model that best explains the model's predictions in the local neighborhood of a data point. It uses a sampling-based approach to generate perturbed instances of input data, reducing the likelihood of getting stuck at local optima.

Finally, the XAI model examines interpretable model coefficients or decision rules, providing explanations for the most important elements driving model prediction in a given instance. The outcome of this process is evaluating the contribution of each feature by assigning weights for each feature to determine the power of its effect on the prediction result and determine whether this effect is negative or positive. Fig. 4 shows six instances from the binary classification task, with 0 indicating benign network flow and 1 indicating an attack. In the first three instances, features such as information assurance technical (IAT), Max, Total Size, DNS, and Rate have a positive effect by helping the model correctly classify the instance into its correct class label. On the other hand, features such as header length have a positive impact on the left three instances while harming the right three instances.

The contribution of features varies based on their value, target class label, and situation. Fig. 5 shows the feature explanations for the 8-class task with one instance per class. The 0 refers to Benign, 1 refers to Brute-Force, 2 indicates DDoS, 3 refers to DoS, 4 represents Mirai, 5 refers to Recon, 6 indicates Spoofing, and 7 refers to Web. As can be observed from the figure, the prediction probabilities for 1 and 7 are 82% and 85%, respectively, indicating the model's confusion in class label prediction. The vast majority of the features have a positive impact on the final result, which reflects how the

model accurately classifies network flows into their specific attack type. However, some features have a negative impact, such as the HTTP feature.

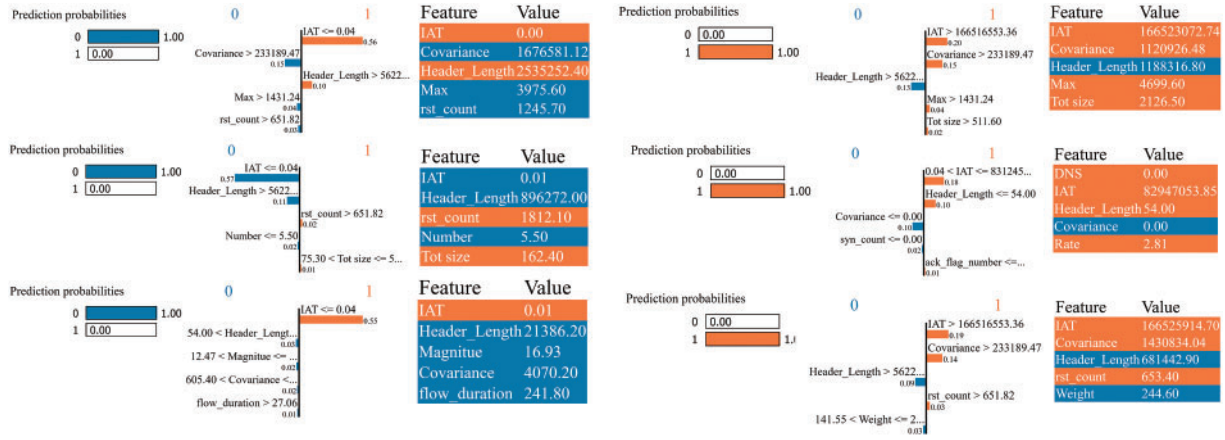

Figure 4: The feature explanation for six instances in the binary classification task

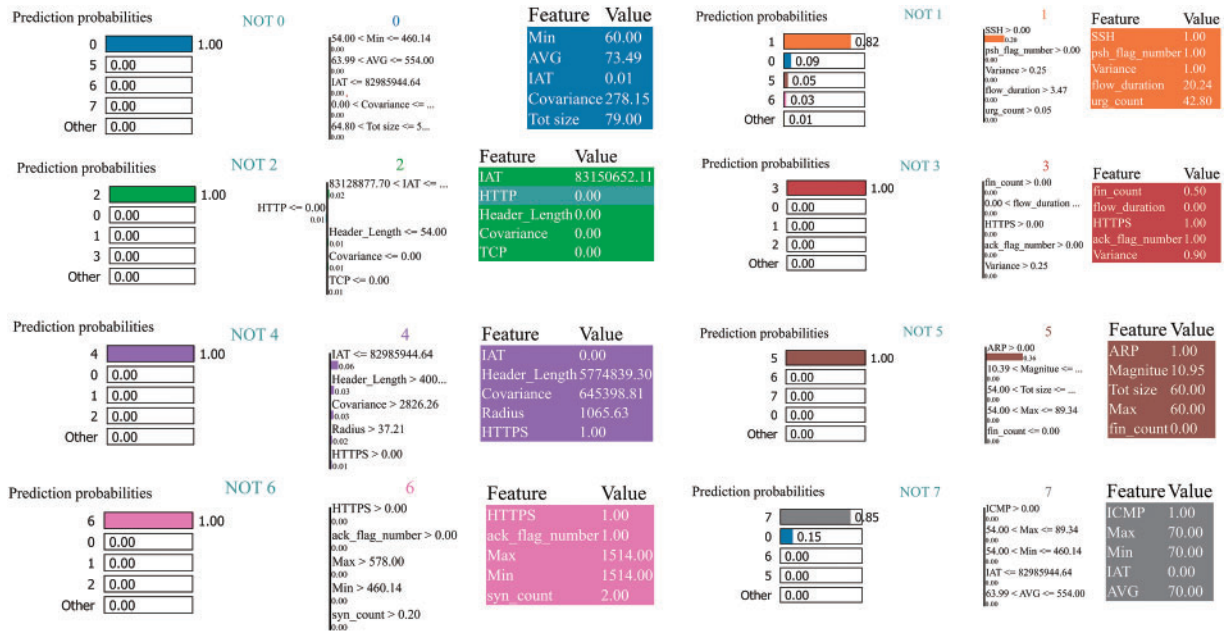

Figure 5: The feature explanation for eight instances in the multiclassification task

5 Conclusion

This paper presents an efficient approach for automatically detecting DDoS attacks in IoT environments. The proposed model investigates and combines various deep learning algorithms and uses XAI to interpret the model predictions. Three different experiments were conducted with varying levels of attack-type complexity to test the system. In all experiments, BiLSTM outperformed the other models, with recalls of 99.39%, 66.79%, and 65.17%, respectively. On the other hand, the CNN

model outperformed the other models in detecting DDoS attacks, demonstrating superior efficiency with minimal training time. The CNN model is highly parallelizable, which means it can efficiently perform parallel computations on the used GPU. This model reduces parameters through parameter sharing, resulting in faster and more stable training. This demonstrates the tradeoff between high performance and accurate results, which necessitates an extensive amount of time for training the model. The primary limitation of this study lies in the benchmarking dataset, which is quite large. This posed challenges in extracting and interpreting the entire traffic data thoroughly, potentially impacting the model's generalization ability. Although the evaluation was performed on another recent DDoS dataset, further processing could be applied to filter the traffic DDoS categories, thereby enhancing the quality of learnable features. Many future directions may include developing a federated learning framework for detecting IoT network attacks and creating reliable datasets to improve the accuracy of deep learning models for detecting DDoS attacks.

Acknowledgement: The authors would like to thank the anonymous reviewers for their constructive feedback and insightful comments, which helped us in improving the quality of the manuscript.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Conceptualization, Ahmad Alzu'bi, Amjad Albashayreh, and Abdelrahman Abuarqoub; methodology, Ahmad Alzu'bi, Amjad Albashayreh, Abdelrahman Abuarqoub, and Mai A. M. Alfawair; formal analysis, Ahmad Alzu'bi, and Amjad Albashayreh; investigation, Abdelrahman Abuarqoub, and Mai A. M. Alfawair; data curation, Amjad Albashayreh; writing—original draft preparation, Ahmad Alzu'bi, and Amjad Albashayreh; writing—review and editing, Abdelrahman Abuarqoub, and Mai A. M. Alfawair; visualization, Ahmad Alzu'bi, and Amjad Albashayreh; supervision, Ahmad Alzu'bi; project administration, Abdelrahman Abuarqoub, and Mai A. M. Alfawair; correspondence author, Ahmad Alzu'bi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The CICIoT2023 dataset is publicly available on https://www.unb.ca/cic/datasets/iotdataset-2023.html, accessed on 8 April 2024. The CICDDoS2019 dataset is publicly available on https://www.unb.ca/cic/datasets/ddos-2019.html, accessed on 5 May 2024.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. Alam, "A reliable communication framework and its use in Internet of Things (IoT)," vol. 10, pp. 450–456, 2018. doi: 10.36227/techrxiv.12657158.v1.

[2] Z. A. El Houda, B. Brik, and S. -M. Senouci, "A novel IoT-based explainable deep learning framework for intrusion detection systems," *IEEE Internet Things Mag.*, vol. 5, no. 2, pp. 20–23, Jun. 2022. doi: 10.1109/IOTM.005.2200028.

[3] S. Abbas *et al.*, "A novel federated edge learning approach for detecting cyberattacks in IoT infrastructures," *IEEE Access*, vol. 11, pp. 112189–112198, 2023.

[4] A. Langiu, C. A. Boano, M. Schuß, and K. Römer, "UpKit: An open-source, portable, and lightweight update framework for constrained IoT devices," in *2019 IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, 2019, pp. 2101–2112.

[5] D. Canavese, L. Mannella, L. Regano, and C. Basile, "Security at the edge for resource-limited IoT devices," *Sensors*, vol. 24, no. 2, 2024, Art. no. 590. doi: 10.3390/s24020590.
[6] A. Tabassum and W. Lebda, "Security framework for IoT devices against cyber-attacks," arXiv preprint arXiv:1912.01712, 2019.
[7] A. Alzu'bi, A. Alomar, S. Alkhaza'leh, A. Abuarqoub, and M. Hammoudeh, "A review of privacy and security of edge computing in smart healthcare systems: Issues, challenges, and research directions," *Tsinghua Sci. Technol.*, vol. 29, no. 4, pp. 1152–1180, Aug. 2024. doi: 10.26599/TST.2023.9010080.
[8] T. -L. Nguyen, H. Kao, T. -T. Nguyen, M. -F. Horng, and C. -S. Shieh, "Unknown DDoS attack detection with fuzzy C-means clustering and spatial location constraint prototype loss," *Comput. Mater. Contin.*, vol. 78, pp. 1–10, 2024. doi: 10.32604/cmc.2024.047387.
[9] R. Vishwakarma and A. K. Jain, "A survey of DDoS attacking techniques and defence mechanisms in the IoT network," *Telecommun. Syst.*, vol. 73, no. 1, pp. 3–25, 2019. doi: 10.1007/s11235-019-00599-z.
[10] S. K. Jagatheesaperumal, Q. -V. Pham, R. Ruby, Z. Yang, C. Xu and Z. Zhang, "Explainable AI over the Internet of Things (IoT): Overview, state-of-the-art and future directions," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 2106–2136, 2022. doi: 10.1109/OJCOMS.2022.3215676.
[11] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, 2023, Art. no. 5941. doi: 10.3390/s23135941.
[12] N. Sharmin and C. Kiekintveld, "Enhancing IoT device security: Predicting and analyzing reconnaissance attacks using flags and time-based attributes," in *2023 10th Int. Conf. Internet of Things: Syst., Manage. Secur. (IOTSMS)*, San Antonio, TX, USA, IEEE, 2023, pp. 23–30.
[13] Z. Wang, H. Chen, S. Yang, X. Luo, D. Li and J. Wang, "A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization," *PeerJ Comput. Sci.*, vol. 9, 2023, Art. no. e1569.
[14] M. M. Khan and M. Alkhathami, "Anomaly detection in IoT-based healthcare: Machine learning for enhanced security," *Sci. Rep.*, vol. 14, no. 1, 2024, Art. no. 5872.
[15] S. Yaras and M. Dener, "IoT-based intrusion detection system using new hybrid deep learning algorithm," *Electronics*, vol. 13, no. 6, 2024, Art. no. 1053.
[16] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustain. Cities Soc.*, vol. 72, 2021, Art. no. 102994. doi: 10.1016/j.scs.2021.102994.
[17] E. O. Gyamfi *et al.*, "A model-agnostic XAI approach for developing low-cost IoT intrusion detection dataset," *J. Inform. Secur. Cyber. Res.*, vol. 6, no. 2, pp. 74–88, 2023. doi: 10.26735/LPAO2070.
[18] M. K. Hasan *et al.*, "An explainable ensemble deep learning approach for intrusion detection in industrial Internet of Things," *IEEE Access*, vol. 11, pp. 115047–115061, 2023.
[19] A. Bashaiwth, H. Binsalleeh, and B. AsSadhan, "An explanation of the LSTM model used for DDoS attacks classification," *Appl. Sci.*, vol. 13, no. 15, 2023, Art. no. 8820.
[20] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Dis. Data Min.*, San Diego, CA, USA, 2016, pp. 1135–1144.
[21] F. Hassan, J. Yu, Z. S. Syed, A. H. Magsi, and N. Ahmed, "Developing transparent IDS for VANETs using LIME and SHAP: An empirical study," *Comput. Mater. Contin.*, vol. 77, no. 3, pp. 3185–3208, 2023. doi: 10.32604/cmc.2023.044650.
[22] Y. Wei, J. Jang-Jaccard, A. Singh, F. Sabrina, and S. Camtepe, "Classification and explanation of distributed Denial-of-Service (DDoS) attack detection using machine learning and shapley additive explanation (SHAP) methods," arXiv preprint arXiv:2306.17190, 2023.
[23] S. Tabassum, N. Parvin, N. Hossain, A. Tasnim, R. Rahman and M. I. Hossain, "IoT network attack detection using XAI and reliability analysis," in *2022 25th Int. Conf. Comput. Inform. Technol. (ICCIT)*, Cox's Bazar, Bangladesh, 2022, pp. 176–181.
[24] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using shapley additive explanations," *Expert Syst. Appl.*, vol. 186, 2021, Art. no. 115736.

[25] T. Senevirathna, B. Siniarski, M. Liyanage, and S. Wang, "Deceiving post-hoc explainable AI (XAI) methods in network intrusion detection," in *2024 IEEE 21st Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, IEEE, 2024, pp. 107–112.

[26] O. Arreche, T. R. Guntur, J. W. Roberts, and M. Abdallah, "E-XAI: Evaluating black-box explainable AI frameworks for network intrusion detection," *IEEE Access*, vol. 12, pp. 23954–23988, 2024. doi: 10.1109/ACCESS.2024.3365140.

[27] U. Do, L. Lahesoo, R. M. Carnier, and K. Fukuda, "Evaluation of XAI algorithms in IoT traffic anomaly detection," in *2024 Int. Conf. Artif. Intell. Inform. Commun. (ICAIIC)*, Osaka, Japan, IEEE, 2024, pp. 669–674.

[28] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *2019 Int. Carnahan Conf. Secur. Technol. (ICCST)*, Chennai, India, IEEE, Oct. 2019, pp. 1–8.

[29] M. Méndez, M. G. Merayo, and M. Núñez, "Long-term traffic flow forecasting using a hybrid CNN-BiLSTM model," *Eng. Appl. Artif. Intell.*, vol. 121, 2023, Art. no. 106041.

[30] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A survey of CNN-based network intrusion detection," *Appl. Sci.*, vol. 12, no. 16, 2022, Art. no. 8162.

[31] M. V. O. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença Jr, "A GRU deep learning system against attacks in software defined networks," *J. Netw. Comput. Appl.*, vol. 177, 2021, Art. no. 102942.

[32] K. Kim, J. -H. Lee, H. -K. Lim, S. W. Oh, and Y. -H. Han, "Deep RNN-based network traffic classification scheme in edge computing system," *Comput. Sci. Inf. Syst.*, vol. 19, no. 1, pp. 165–184, 2022. doi: 10.2298/CSIS200424038K.

[33] P. Gohel, P. Singh, and M. Mohanty, "Explainable AI: Current status and future directions," arXiv preprint arXiv:2107.07045, 2021.