IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 August 2023, accepted 19 October 2023, date of publication 25 October 2023, date of current version 1 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327620

RESEARCH ARTICLE

Distributed Denial of Service Attack Detection for the Internet of Things Using Hybrid Deep Learning Model

AHMED AHMIM[1], FAIZ MAAZOUZI[1], MARWA AHMIM[2], SARRA NAMANE[2], AND IMED BEN DHAOU[3,4,5], (Senior Member, IEEE)

[1]Department of Mathematics and Computer Science, Mohamed-Cherif Messaadia University—Souk Ahras, Souk Ahras 41000, Algeria
[2]Networks and Systems Laboratory, Department of Computer Science, Badji Mokhtar-Annaba University, Annaba 23000, Algeria
[3]Department of Computer Science, Hekma School of Engineering, Computing, and Design, Dar Al-Hekma University, Jeddah 22246, Saudi Arabia
[4]Department of Computing, University of Turku, 20014 Turku, Finland
[5]Department of Technology, Higher Institute of Computer Sciences and Mathematics, University of Monastir, Monastir 5000, Tunisia

Corresponding author: Ahmed Ahmim (a.ahmim@univ-soukahras.dz)

ABSTRACT As a result of the widespread adoption of the Internet of Things, there are now hundreds of millions of connected devices, increasing the likelihood that they may be vulnerable to various types of cyberattacks. In recent years, distributed denial of service (DDoS) has emerged as one of the most destructive tools utilized by attackers. Traditional machine learning approaches are typically ineffective and unable to cope with actual traffic properties when used to identify DDoS attacks. This paper introduces a novel deep learning-based intrusion detection system, specifically designed for deployment at either the Cloud or Fog level in the IoT environment. The proposed model aims to detect all types of DDoS attacks with their specific subcategory. Our hybrid model combines different types of deep learning models, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), Deep Autoencoder, and Deep Neural Networks (DNNs). Our proposed model is made up of two main levels. The first one contains different parallel sub-neural networks trained with specific algorithms. The second level uses the output of the frozen first level combined with the initial data as input. The idea behind the combination of these various types of deep neural networks is to exploit their different properties to achieve very high performance. To evaluate our model, we used the CIC-DDoS2019 dataset, which satisfies all the constraints of an intrusion detection dataset. The results obtained demonstrate that our proposed model outperformed various well-known machine learning and deep learning models in terms of the true positive rate, accuracy, false alarm rate, average accuracy, and average detection rate.

INDEX TERMS Intrusion detection, DDoS detection, IDS, machine learning, deep learning, CNN, LSTM, autoencoder, hybrid model.

I. INTRODUCTION

The tremendous growth in the number of devices connected to the Internet, which has reached tens of billions [1], has led us to the age of the Internet of Everything. The Internet is now integrated into a wide range of devices, including simple gadgets such as smartwatches and complete smart ecosystems such as smart grids, smart transport systems, and smart cities [2]. This integration allows for the automation of many daily life tasks and brings several advantages to our lifestyle. However, connecting devices to the Internet exposes them to malicious people and things that can exploit the devices' weaknesses to hack them [3]. Moreover, the more variance and heterogeneity in devices there are, the more security failures we have, and correcting these weaknesses becomes more challenging. As a consequence, hackers have more opportunities to launch attacks that can affect more devices and cause more widespread damage [4], [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Ines Domingues.

Distributed Denial-of-Service (DDoS) attacks represent one of the most dangerous types of attack that can render online services inaccessible to legitimate users. According to current Microsoft figures, DDoS has increased by 300 percent. The top four DDOS assaults in 200 were the Amazon Web Services attack, the GitHub DDoS attack, the Dyn DDoS attack that damaged internet services, and the Mafiaboy strikes [6].

To launch such attacks, hackers use various techniques, mainly exploiting botnets, to flood the servers of the target online service with a massive volume of traffic, overwhelming their ability to respond to legitimate requests. DDoS traffic has been a real challenge in the last decade due to its high negative impact on network communication efficiency. Current defense methods focus on the service side to mitigate their damage, which makes DDoS problems inherent [7]. To deal with threats primarily coming from the Internet, various prevention techniques are deployed, including cryptography, authentication, firewalls, proxies, antivirus software, VPNs, and more. However, even though these techniques are useful, an experienced hacker can counter them and perform their malicious task. Therefore, computer security experts have introduced a new solution, called IDS (Intrusion Detection System), as a second line of defense to improve the security level [8]. An intrusion detection system is a software or hardware application that continuously monitors networks or systems to detect any violation of security policy [9].

To build an intrusion detection analyzer we can adopt two approaches, the first one focuses on normal behavior, and any deviation from the normal model is considered as an attack, this approach is well known as anomaly detection. The second approach focuses on the identification of previously recognized attacks and their similar ones, this approach is called misuse detection [10].

The algorithms used for intrusion detection can be classified into three categories: rule-based algorithms, statistics-based algorithms, and machine learning (ML)-based approaches [11]. Machine learning-based intrusion detection can also be classified into three subcategories: traditional machine learning algorithms, ensemble learning algorithms, and deep learning algorithms [12].

Currently, deep learning is one of the latest machine learning algorithms and has shown high accuracy and prediction ability in different fields, such as image classification, speech recognition, and intrusion detection [12].

The main goal of this article is to develop a high-performance intrusion detection model to detect Distributed Denial of Service (DDoS) attacks. A number of requirements must be met in order to accomplish this purpose, including the capacity to identify the various DDoS attack subcategories, high accuracy, high detection rate, low false alarm rate, capacity to distinguish between closely related attacks, capacity to handle heavy traffic, high generalization ability, and quick detection speed.

It is generally acknowledged that deep learning has a wide range of abilities, including autonomous feature extraction, scalability, high self-learning, the capacity to work with incomplete data, a wide variety of model options, the capacity to combine different models, the potential for transfer learning, a wide range of platforms and APIs, and more. The main contribution of this paper lies in the combination of different advanced deep-learning techniques, which can be summarized as follows:

- Novel preprocessing method: We provide a novel preprocessing method for the CICDDoS2019 dataset, which includes all kinds of attacks in both the training and testing subsets, even those that are quite identical.
- Hybrid architecture: We combine the convolutional neural network (CNN), Long short-term memory (LSTM), and autoencoder models in a parallel and cascade manner to create a highly complex architecture.
- Transfer learning: We apply transfer learning to improve the accuracy of traffic classification.
- Improved low-frequency traffic classification: We devise a novel training process to improve the accuracy of low-frequency traffic classification.
- Hierarchical deep learning model: We develop a new hierarchical deep learning model with both freezing and trainable levels.
- Comparative analysis: We compare our proposed model with several basic deep learning models, as well as some well-known machine learning models.

The remainder of this paper is organized as follows. Section II presents related work. In Section III, we describe the design ideas behind our proposed model, its different blocks, and the algorithms used to build it. In Section IV, we illustrate the dataset used and how it is pre-processed. Section V summarizes the experiments and comparative studies. Finally, Section 6 provides the conclusion and summarizes potential future work.

II. RELATED WORKS

Over the last decade, the availability of massive amounts of data in almost every field has led to the widespread use of machine learning and deep learning methods to address various problems. One such field is computer security, particularly intrusion detection, where machine learning and deep learning have been applied to tackle well-known issues. In this section, we present some recent works on DDoS attack detection that have utilized machine learning and deep learning techniques. We will structure the discussion based on the datasets employed for training and testing in the studies. Initially, We will start by examining the works that utilize more specialized and widely used datasets. Subsequently, we progress towards studies that employ less commonly utilized and general datasets.

A. PAPERS USING CICDDoS2019 DATASET

Zainudin et al. [13] proposed a deep learning approach (CNN, LSTM) for the detection and classification of DDoS attacks

in the IIoT environment that utilizes the feature selection technique based on XGBoost. This proposed approach achieved 99. 50 % accuracy with a latency of 0.179 ms.

Furthermore, Aydin et al. [14] designed a Long short-term memory (LSTM) DDoS attack detection and prevention system in a public cloud computing environment. The proposed solution deployed a signature-based attack detection approach. The results revealed that the LSTM-based detection model had a 99. 83 % accuracy rate.

Wei et al. [15] proposed a hybrid approach named AE-MLP to detect and classify DDoS attacks. The performance evaluation of their model reveals that the AE-MLP achieved the best performance metrics, including 98.34 % accuracy, 98.48 %recall, 97.91 % precision, and 98.18 % F1-score.

Furthermore, Rani et al. [16] were interested by the prevention and detection of DOS and DDoS attacks in device-to-device (D2D) communications. Initially, they started by evaluating the performance of several Machine Learning (ML) algorithms, namely Random Forest, Light GBM, XGBoost, and Ada Boos. The results showed that both the CICDDoS2019 and Slowloris datasets reached greater precision with Random Forest. Consequently, the authors worked on developing a technique that combines the identification of DOS and DDoS attacks in binary classification Random Forests with the binary decision. The evaluation of the suggested technique revealed good performance compared to existing solutions in terms of prevention and identification time, resources required, and battery consumption.

B. PAPERS USING CICIDS2017 DATASET

The primary goal of the paper of Elsayed et al. [17] is to choose the critical features from the original dataset while lowering the high computational complexity and increasing the classification accuracy. For that purpose, they used the Information Gain (IG) and Random Forest (RF) feature selection methods with a modified deep learning model (DL) based on the Long short-term memory Autoencoder (LSTM-Autoencoder). The evaluation of the proposed solution showed an accuracy rate of 99.50 % by the IG selection method and 98.76 % by the random forest selection method.

Moreover, Makuvaza et al. [18] proposed a new real-time DDoS attack detection method based on deep neural networks (DNN) for software-defined networks (SDN). This method achieved less time and less cost, including 97.25 % accuracy.

Agarwal et al. [19] presented a new feature selection-whale optimization algorithm in a deep neural network model. The evaluation results show a precision of 95.35 % in detecting DDoS attacks.

The limitations of traditional DDoS attack detection methods were initially discussed by Liu et al. [20], where it was noticed that these methods are based on statistical analysis or a machine learning paradigm. For the purpose of ensuring efficient and effective identification of DDoS attacks against the controller in Software Defined Networking (SDN), the authors proposed a two-level DDoS attack detection method based on information entropy and Deep Learning (DL). The research team has the conviction that the combination of these theories pledges the exploitation of both of their benefits simultaneously. The experimental results of the proposed method revealed potency with 98.98 % detection accuracy.

C. PAPERS USING OTHER DATASETS

Singh and Jayakumar [21] proposed a new model named IU-ROA to detect and mitigate DDoS attacks. The result analysis shows that the IU-ROA model achieved 96 % for DDoS Attack Detection and 90.06 % for mitigation analysis.

In addition, Selvan et al. [22] presented a FACVO-based DNFN scheme that consists of combining a feature fusion method and deep QNN to obtain constructive information. The experimental results of their model exhibited the best performance metrics, including precision values, TNR, accuracy, and TPR of 87.45%, 87.45%, 93.04%, 92.93%, and 90.88% for using the NSL-KDD dataset without attack, and 86.48%, 90.15%, 92.00%, and 89.91% for using the BoT-IoT dataset without attack.

Similarly, a hybrid method that uses deep convolutional neural networks and real network data is presented by Hussain et al. [23]. The performance of this method is evaluated using an open CDR dataset. The evaluation shows that this hybrid method achieved an accuracy detection higher than 91 % for normal and under attack cell.

From another angle, Priyadarshini and Barik [24] focused their research on the prevention and detection of DDoS attacks in fog computing environments. The authors also chose a Long short-term memory (LSTM) deep learning method that performs well with respect to sequential data. The evaluation of the proposed model achieved 98.88 % of accuracy rate.

Furthermore, Fouladi et al. [25] noticed that the separation between the forwarding devices and the control entity in Software Defined Networking (SDN) allows the network to provide a better security level compared to traditional ones. However, the authors affirm that DDoS attacks remain effective on this type of network. For this purpose, in this paper, a DDoS attack detection scheme is presented, based on a discrete wavelet transform (DWT) and an auto-encoder neural network for SDN. The experimental results showed that the proposed solution has an efficient attack detection rate with a low false alarm rate.

Similarly, Elsaeidy et al. [26] proposed a hybrid deep learning approach for replay and detection of DDoS attacks in a real-life smart city infrastructure. This approach consists of an input layer, a global average pooling (GAP) layer, a deep CNN with seven hidden layers, a deep RBM model with two hidden layers, and a softmax output layer. The performance of this hybrid approach is evaluated using real-world smart city datasets (environmental, smart river, and smart soil). The simulation results showed an improvement in the performance of this approach compared to other machine learning and deep learning models in the literature. Where it reported an accuracy equal to 98.13 % for the smart river

dataset, 99.51 % for the smart soil dataset, and 98.37 % for the environmental dataset.

Focusing on detecting Distributed Denied-of-Service (DDoS) attacks and forgetting that victim protection is still limited due to the fact that the attacks cannot be stopped remains a substantial shortcoming of these studies. To overcome this problem, Zhaou et al. [27] presented a DDoS attack flow classification system, known as SAFE. This system has the ability to distinguish the attack flows from the benign ones and, therefore, block the attackers' flows. SAFE provides a feature selection method that selects highly informative features for DDoS attack flow classification. Then, a threshold tuning method is deployed to maximize feature performance. The evaluation of the proposed classification system showed its effectiveness compared to the existing methods.

As summarized in Table 1, the majority of current research uses non-dedicated DDoS attack datasets. Even when using a dedicated dataset (such as the CICDDoS 2019 datasets), the test subset often does not encompass all attacks due to the striking similarities between certain types of DDoS attacks, making it challenging for machine learning methods to effectively differentiate between them. In order to overcome this restriction, our proposed technique is to properly classify every type of DDoS attack. To reach this goal, diverse deep-learning algorithms are combined, then a novel training technique is used.

III. THE PROPOSED MODEL

Our proposed intrusion detection system aims to detect all types of DDoS attacks originating from or passing through the Cloud or Fog levels in an IoT environment, by analyzing network traffic. As shown in Figure 1, hackers in an IoT environment can launch DDoS attacks by targeting one or more levels of the IoT infrastructure, namely, Cloud, Fog, or Edge computing. To accomplish this objective, a hacker needs to infect, hack, and take control of numerous edge nodes, and potentially fog and cloud ones. By gaining control over these nodes, they can initiate DDoS attacks targeting cloud nodes, fog nodes, or even a cluster of edge nodes.

As illustrated in Figure 1, and due to the computational and battery constraints of the Edge nodes, our approach for deploying a network intrusion detection system in an IoT environment involves the following steps. We start by gathering network traffic from different levels, namely Edge, Fog, and Cloud. Then, we pre-process this data to use it for the training process of the global intrusion detection model, which is trained on a Cloud Computing infrastructure known for its high computing performance. Finally, we deploy the trained model on both Cloud and Fog nodes to effectively monitor the network traffic in the IoT environment.

Our proposed model is designed to detect DDoS attacks initiated against Cloud, Fog, or Edge nodes, which may originate from Cloud, Fog, or Edge nodes and pass through Cloud or Fog. The sole exception not covered by our model is a direct DDoS attack that bypasses the Fog level. Such instances are rare and demand advanced equipment and extensive physical deployment to directly connect with the Edge level.

The main purpose of our model is to accurately classify benign traffic and all types of DDoS attacks, including the most similar ones. To achieve this goal and provide the correct class for each DDoS attack, we proposed a specific training algorithm and a model that combines three deep learning architectures: the Convolutional Neural Network (CNN) [43], the Long Short-Term Memory (LSTM) [44], and the Autoencoder [45].

A. INTERNAL STRUCTURE OF OUR MODEL

Due to the structural complexity of certain types of DDoS attacks, the difficulty of recognizing them, and their mutual similarity, it becomes necessary to use a hybrid multi-level deep neural network to build a model capable of recognizing all DDoS attacks.

To achieve this goal, we draw inspiration from successful models such as ResNet [46] and YOLO [47]. In these models, we observe that developers often combine various types of neural networks in parallel and cascade ways. In these models, it is apparent that developers frequently leverage a variety of neural networks at different levels to construct highly performant models. Despite the high complexity and computational intensity involved in the training stage of these models, their performance is truly remarkable. Furthermore, these models demonstrate remarkably short processing times when classifying each input.

1) THE GLOBAL MODEL

As depicted in Figure 2, our global model consists of an input layer followed by a set of frozen sub-models. These sub-models have the same structure but are trained with different subsets extracted from the initial dataset, and they serve as components of the global model. The outputs of these parallel sub-models are then used as inputs for the second level, which consists of one LSTM layer and two Conv2D layers. The third level is composed of four successive Dense layers. Finally, we have the output layer, which consists of a Softmax layer used to classify the network traffic into 13 different DDoS attack types and benign traffic.

The concept underlying our model, as depicted in Figure 2, involves the combination of various types of neural networks with the aim of extracting novel hidden features. These extracted features are then combined as inputs to achieve our objective of accurately classifying closely similar DDoS attacks. Therefore, in our approach, we utilized an autoencoder, LSTM, 2D CNN, and DNN, along with our novel training algorithm. Furthermore, during the second training stage, we incorporated a freezing mechanism for the first level, which played a significant role in achieving high performance.

TABLE 1. Related works.

Publication Year	Paper reference	Used Data set	Machine learning and Deep learning model	Details of used approach
	Our work	CICDDoS2019 [28]	CNN 2D, LSTM and Autoencoder	Hybrid IDS, which combine three Deep learning models and use a specific training algorithm
2022	Zainudin et al. [13]	CICDDoS2019 [28]	CNN,LSTM	A CNN-LSTM-based model for detecting and classifying DDoS attacks in an IIoT environment
2022	Aydin et al. [14]	CICDDoS2019 [28]	A signature-based attack detection approach, a Long Short-Term Memory (LSTM), deep learning	A LSTM-CLOUD system that detects and prevents DDoS attacks in a public cloud network environment
2021	Wei et al. [15]	CICDDoS2019 [28]	Autoencoder (AE), Multi-layer Perceptron Network (MLP)	A hybrid approach named AE-MLP to detect and classify DDoS attacks
2023	Rani et al. [16]	CICDDoS2019 [28], Slowloris [29]	Random Forest,Light GBM, XGBoost and Ada Boos	A DDoS attack detection system based on machine learning in D2D communications
2022	El Sayed et al. [17]	InSDN [30] , CICIDS2017 [31], CICIDS2018 [31]	Information Gain (IG), Random Forest (RF), Deep learning, Long short-term memory (LSTM) and Autoencoder	DDoS detection approach based on Information Gain (IG) and Random Forest (RF) feature selection methods in SDNs
2021	Makuvaza et al. [18]	CICIDS2017 [31]	Deep neural networks (DNN)	Real-time DDoS attack detection method for software-defined networks (SDN).
2021	Agarwal et al. [19]	CICIDS2017 [31]	FS-WOA-DNN	DDoS attack detection approach based for cloud storage service
2022	Liu et al. [20]	CICIDS2017 [31]	Information entropy detection method, deep learning and CNN	A two-level DDoS attack detection method based on information entropy and deep learning
2022	Singh and Jayakumar [21]	KDD cup 99 [32]	CNN, ROA	IU-ROA model to detect and mitigate DDoS attacks
2022	Selvan et al [22]	NSL-KDD [33], BoT-IoT [34]	Feature fusion method, data augmentation, Deep Quantum Neural Network (Deep QNN), Fractional Calculus (FC)	A DDoS attack detection system based on deep learning feature fusion method
2020	Hussain et al. [23]	open CDR by Telecom Italia in 2015 [35]	CNN, real network data	DDoS Detection for Cyber Physical Systems (CPS) on a 5G Network
2022	Priyadarshini and Barik [24]	Hogzilla [36]	A Long short-term memory (LSTM), deep learning, binary cross-entropy, a Mini-batch gradient descent (GD) algorithm	A deep learning based model to protect a Fog network from DDoS attacks
2022	Fouladi et al. [25]	MAWI [37] FIFA world cup traffic [38]	Discrete wavelet transform (DWT), auto-encoder neural network	Hybrid DDoS attack IDS that combines frequency domain analysis and ML-based approach
2021	Elsaeidy et al. [26]	Queanbeyan Smart city platform datasets [39]	a global average pooling (GAP) layer, deep CNN, deep RBM model and a softmax	The hybrid deep learning approach for replay and DDoS attack detection in a real-life smart city infrastructure
2022	Zhou et al. [27]	Mirai [40], SYN-flooding [41], LowRate [42]	A threshold tuning method and feature-based classification method	DDoS Attack Flows identification (SAFE) system

- Mirai [40]: The dataset is from IMPACT, and it contains DDoS attack traffic launched by an IoT botnet.
- SYN flooding [41]: The dataset is from the IMPACT, and it contains an SYN flooding attack and background traffic on November 5, 2009
- LowRate [42]: The dataset is from the Center for Applied Internet Data Analysis (CAIDA), and it contains a low-rate DDoS attack on August 5, 2007
- NSL-KDD [33]: It is utilized in the attack detection method, and is an extension of the KDD Cup 99 datasets.
- BoT-IoT [34]: BoT-IoT dataset: It is founded on the network system of the cyber lab in the UNSV Canberra cyber center.
- Hogzilla [36]: This dataset is extracting data from CTU-13 Botnet and the ISCX 2012 IDS datasets.
- MAWI [37]: The MAWI traffic repository archives traffic data collected from the WIDE backbone networks. The WIDE network (AS2500) is a Japanese academic network connecting universities and research institutes.
- FIFA world cup traffic [38]: The data set used in this workload characterization study is composed of the access logs collected from each of the servers used in the World Cup Web site of 1998. The access logs from each server were archived on a daily basis.
- InSDN [30]: considers the new structure of the SDN network. It was created using four virtual Machines (VMs). One VM acted as an SDN controller.
- CICIDS2017 [31]: contained network traffic of five days, generated in the period between Monday, July 3, and Friday, July 7, 2017
- CICIDS2018 [31]: represent an extension of CICIDS2017 dataset. Its traces were gathered in 10 days with a total number of 16,233,002 instances, where the size of attacks represented 17% of the entire data.
- CICDDoS2019 dataset [28]: created in 2019 using CICFlowmeter, a tool used to retrieve tangible data from a PCAP file that is generated while performing the attack. This dataset contains records for several types of DDoS attacks such as UDPLag, SYN, etc.
- Slowloris [29]: is created from scratch using two Android D2D devices (one attacking device and one victim device) in order to emulate Slowloris attacks.

2) SUB-MODELS

The structure of the sub-model is illustrated in Figure 3. It consists of two levels, with the first level containing an input layer followed in parallel by three two-dimensional convolutional neural networks (2D CNNs), one long short-term memory (LSTM), and one autoencoder. The outputs of these five parallel components are then combined and serve as the input for the second level. The second level consists of four dense components and a SoftMax output, which is used for the multi-class classification of the network traffic into the different types of DDoS attacks and benign traffic.

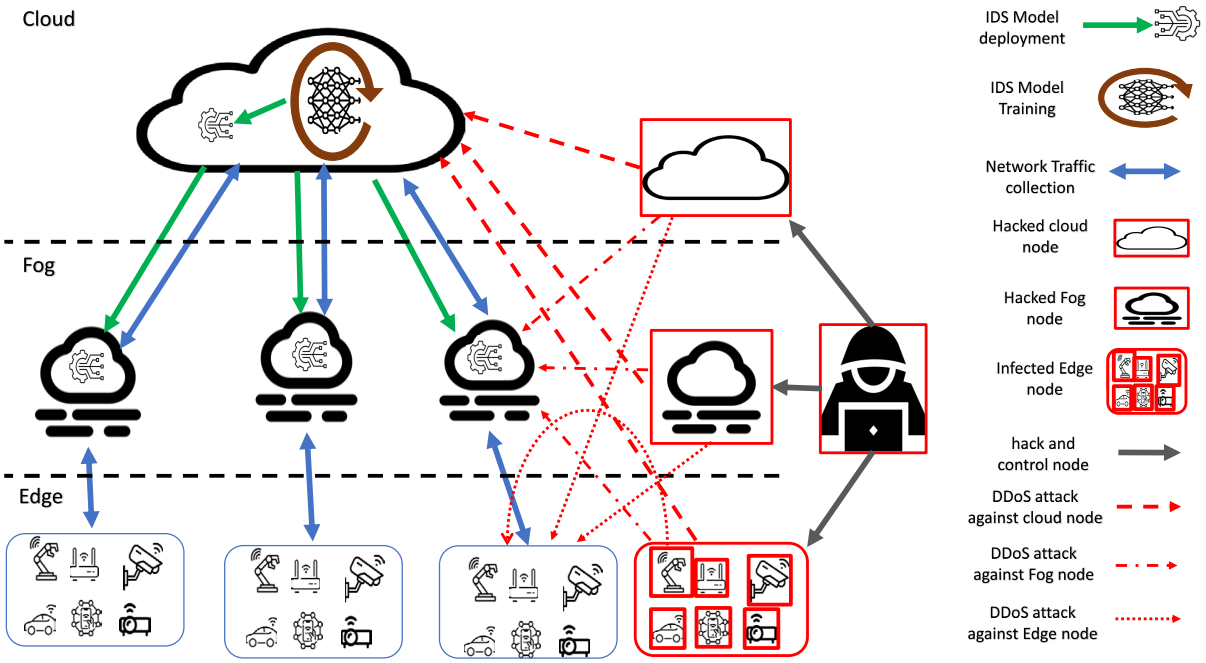


FIGURE 1. Our DDoS intrusion detection system location in the IoT environment.

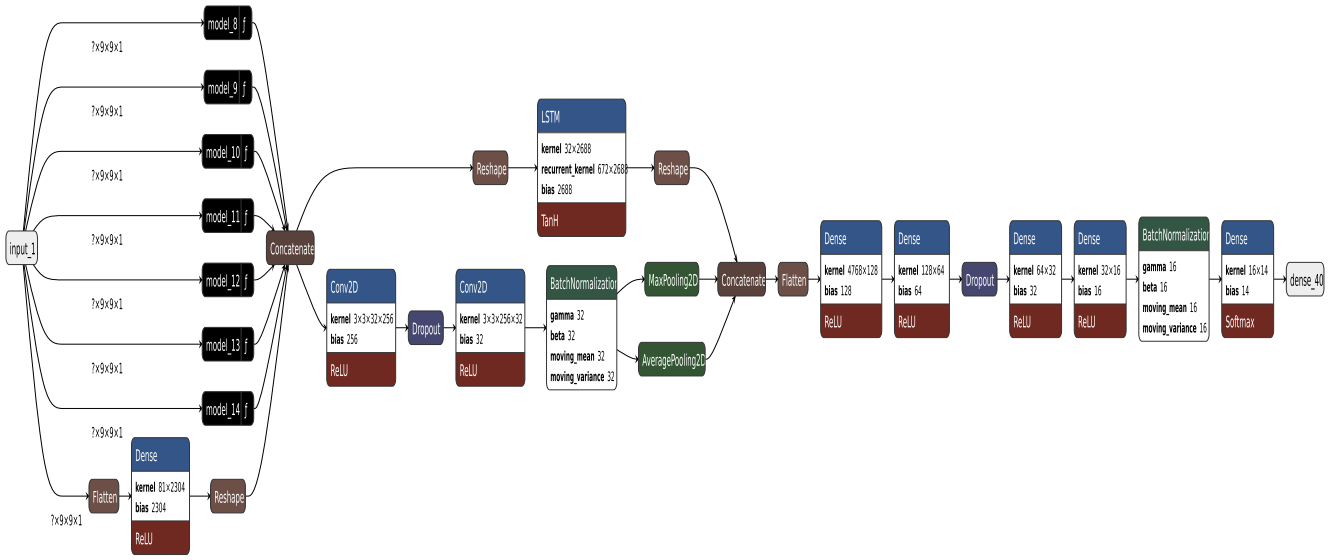


FIGURE 2. General structure of our proposed model.

B. TRAIN STEP

To obtain the global model previously depicted in Figure 2, we follow the steps outlined in Figure 4. We begin by creating the sub-models, then combine them and utilize their outputs as inputs for the second level. Finally, we use the outputs of the second level as inputs for the third level.

1) TRAIN THE SUB-MODELS

To train these sub-models, we follow the steps detailed in Algorithm 1. We initially train the first sub-model using the entire training dataset. Then we test the correctness of the classification for each row. After that, for each iteration (i), we save the trained sub-model (i-1), and we train the sub-model (i) using a training dataset (i) equal to 10 % of the training dataset (i-1) combined with all misclassified rows. We repeat these steps until we reach the threshold. This iterative process aims to provide sub-models that are able to detect specific low-frequency and closely similar attacks, which are challenging to detect.

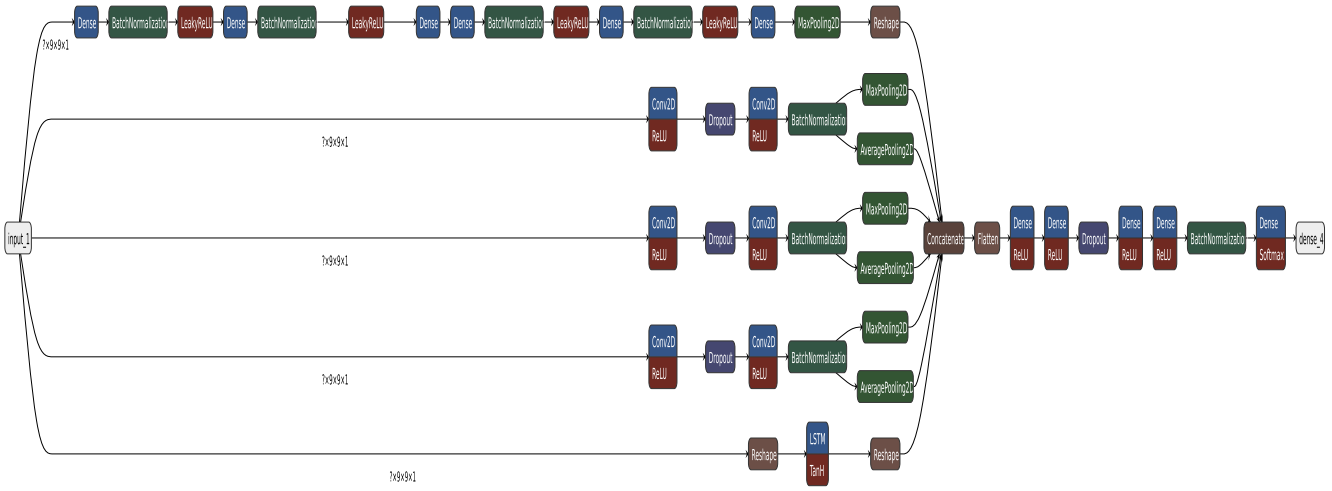


FIGURE 3. Architecture of the sub model.

2) TRAIN THE ENTIRE MODEL

After training the different sub-models, we removed their dense layers. Next, we froze them and utilized their outputs, combined with the initially reshaped input, as the input for the second level. The idea behind freezing the sub-models is to preserve their ability to detect low-frequency and similar attacks, thereby providing specific properties. This input is used as training data for the second level, which is composed of two parallel sub-models: the first one is an LSTM neural network, and the second one is a 2D CNN neural network. The output of this second level is used as the input of the third level, which is composed of a dense neural network [43].

C. TEST STEP

To evaluate our global model, we follow the process depicted in Figure 4, where each row of the test dataset is processed in parallel by all sub-models at each level. The output of a level serves as input for the next level. Finally, the Softmax output layer classifies each row as either benign or a specific type of DDoS attack.

IV. EXPERIMENTATION

In this section, we provide a comprehensive description of the dataset employed, along with the pre-processing procedure applied. Furthermore, we present the architecture of our model, detailing its corresponding hyperparameters. Additionally, we describe the performance metrics used in our experiments. Finally, we conduct a comparative study between our model and various well-known classifiers. Notably, the experiments were conducted on the Kaggle platform [48].

A. DATA SET AND DATA PRE-PROCESSING

To evaluate our model, we used the CICDDoS2019 dataset [28], a real traffic dataset specifically designed for DDoS attacks. This dataset satisfies the essential constraints necessary for a valid data set in cyber security traffic analysis, including anonymity, heterogeneity, diversity of attacks, complete interaction, complete capture, complete network configuration, available protocols, complete traffic, metadata, feature set, and labeling [49].

The CICDDoS2019 dataset in CSV format [28] is separated into 11 files: UDPLag.csv, TFTP.csv, Syn.csv, DrDoS_UDP.csv, DrDoS_SSDP.csv, DrDoS_SNMP.csv, DrDoS_NTP.csv, DrDoS_NetBIOS.csv, DrDoS_MSSQL.csv, DrDoS_LDAP.csv, and DrDoS_DNS.csv. Each file primarily consists of a specific type of attack and every row is labeled as benign or one of the 13 types of DDoS attacks namely: DrDoS_DNS, LDAP (DrDoS_LDAP), MSSQL (DrDoS_MSSQL), NetBIOS (DrDoS_NetBIOS), DrDoS_NTP, DrDoS_SNMP, DrDoS_SSDP, UDP (DrDoS_UDP), Portmap, Syn, TFTP, UDPLag, or WebDDoS. Table 2 provides a comprehensive summary of all features and their corresponding data types in the CICDDoS2019 dataset.

Algorithm 1 Create_Sub_Modellist()

```
ListsubModel ← EmpltySubModelList()
Train_data ← getTrain_data()
while Train_data.getNBRow() > Thorshold do
    A ← getInitialSubModel()
    Train(A,Train_data)
    MissData ← getMissClassified(A,Train_data)
    Train_data ← getTrain10porcent(Train_data)
    Train_data ← Concatenate(Train_data,MissData)
    A ← RemoveInputLayer(A)
    A ← RemoveOutputLayer(A)
    ListsubModel.append(A)
end while
Return  ListsubModel
```

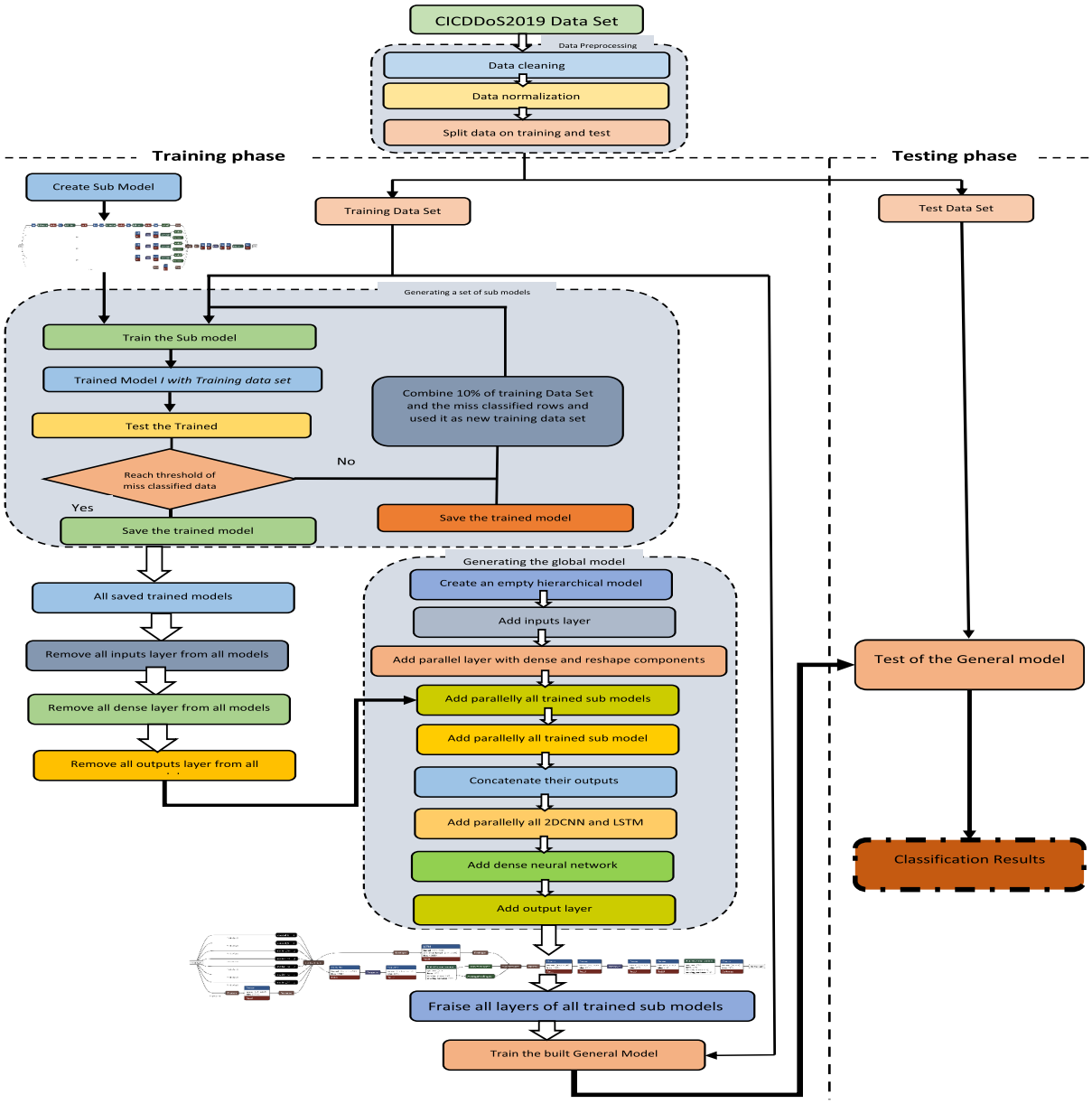


FIGURE 4. Different stage to build and test our model.

Every row in this dataset comprises 88 features, where each feature denotes a characteristic of network traffic.

To create a training and test subset, the 11 CSV files are merged into a single table. Then, the unimportant features, namely Unnamed 0, Flow ID, Source IP, Source Port, Destination IP, Timestamp, SimilarHTTP, and Inbound, are removed. After that, the redundancy is eliminated and all the features are normalized using equation (1) for each value x_i of feature j.

$$\overline{x_i(j)} = \frac{x_i(j) - min(x(j))}{max(x(j)) - min(x(j))} \tag{1}$$

Finally, the training and test subsets are extracted based on the distribution described in Table 3.

As summarized in Table 3, both the training and test datasets encompass all types of benign traffic and attack types, including even the most similar and rare ones. These comprehensive sub-datasets enable a realistic evaluation of the different models.

B. PERFORMANCE METRICS

The performance of an Intrusion Detection System (IDS) is evaluated based on its ability to accurately classify each network connection as either Benign or a specific type of attack. The Confusion Matrix, depicted in Table 4, presents the various possible scenarios and classifications.

To evaluate the performance of our proposed model, we employed two types of metrics. The first type pertains to

TABLE 2. CICDDoS2019 features and data type.

Feature Names	Data Type	Feature Names	Data Type	Feature Names	Data Type
Unnamed: 0	int64	Fwd IAT Max	float64	Avg Fwd Segment Size	float64
Flow ID	object	Fwd IAT Min	float64	Avg Bwd Segment Size	float64
Source IP	object	Bwd IAT Total	float64	Fwd Header Length.1	int64
Source Port	int64	Bwd IAT Mean	float64	Fwd Avg Bytes/Bulk	int64
Destination IP	object	Bwd IAT Std	float64	Fwd Avg Packets/Bulk	int64
Destination Port	int64	Bwd IAT Max	float64	Fwd Avg Bulk Rate	int64
Protocol	int64	Bwd IAT Min	float64	Bwd Avg Bytes/Bulk	int64
Timestamp	object	Fwd PSH Flags	int64	Bwd Avg Packets/Bulk	int64
Flow Duration	int64	Bwd PSH Flags	int64	Bwd Avg Bulk Rate	int64
Total Fwd Packets	int64	Fwd URG Flags	int64	Subflow Fwd Packets	int64
Total Backward Packets	int64	Bwd URG Flags	int64	Subflow Fwd Bytes	int64
Total Length of Fwd Packets	float64	Fwd Header Length	int64	Subflow Bwd Packets	int64
Total Length of Bwd Packets	float64	Bwd Header Length	int64	Subflow Bwd Bytes	int64
Fwd Packet Length Max	float64	Fwd Packets/s	float64	Init_Win_bytes_forward	int64
Fwd Packet Length Min	float64	Bwd Packets/s	float64	Init_Win_bytes_backward	int64
Fwd Packet Length Mean	float64	Min Packet Length	float64	act_data_pkt_fwd	int64
Fwd Packet Length Std	float64	Max Packet Length	float64	min_seg_size_forward	int64
Bwd Packet Length Max	float64	Packet Length Mean	float64	Active Mean	float64
Bwd Packet Length Min	float64	Packet Length Std	float64	Active Std	float64
Bwd Packet Length Mean	float64	Packet Length Variance	float64	Active Max	float64
Bwd Packet Length Std	float64	FIN Flag Count	int64	Active Min	float64
Flow Bytes/s	float64	SYN Flag Count	int64	Idle Mean	float64
Flow Packets/s	float64	RST Flag Count	int64	Idle Std	float64
Flow IAT Mean	float64	PSH Flag Count	int64	Idle Max	float64
Flow IAT Std	float64	ACK Flag Count	int64	Idle Min	float64
Flow IAT Max	float64	URG Flag Count	int64	SimillarHTTP	object
Flow IAT Min	float64	CWE Flag Count	int64	Inbound	int64
Fwd IAT Total	float64	ECE Flag Count	int64	Label	object
Fwd IAT Mean	float64	Down/Up Ratio	float64		
Fwd IAT Std	float64	Average Packet Size	float64		

TABLE 3. Distribution of attacks and benign rows on training and test subsets extracted from CICDDoS2019.

CATEGORY		Training	Test
BENING	BENIGN	55972	13888
DDoS ATTACK	DrDoS_DNS	7935	2065
	DrDoS_LDAP	4010	990
	DrDoS_MSSQL	3997	1003
	DrDoS_NetBIOS	4002	998
	DrDoS_NTP	7964	2036
	DrDoS_SNMP	7949	2051
	DrDoS_SSDP	8007	1993
	DrDoS_UDP	4002	998
	LDAP	4024	976
	MSSQL	4064	936
	NetBIOS	4034	966
	Portmap	7925	2075
	Syn	8083	1917
	TFTP	8027	1973
	UDP	3949	1051
	UDPLag	4950	1321
	WebDDoS	344	73
Total Attacks		93266	23422
Total		149238	37310

TABLE 4. Confusion matrix.

		Predicted class	
		Negative(Benign)	Positive(Attack)
Real class	Negative(Benign)	True negative	False positive
	Positive(Attack)	False negative	True positive

individual network connections, specifically the True Positive Rate (TPR) for each type of network connection. The second type encompasses global metrics, which include the accuracy, average detection rate, false alarm rate (FAR), and Average Accuracy. Equations 3, 4, 5, 6 summarize respectively the calculation formulas for previous metrics. Furthermore, to provide a more accurate evaluation, we utilized the ROC curve and ROC AUC (Area Under the Curve) to assess our model and compare it with other models.

$$TPR_{ConnecType} = \frac{TP_{ConnecType}}{TP_{ConnecType} + FN_{ConnecType}} \quad (2)$$

$$\text{Accuracy} = \frac{\sum^{NBclass} \text{TP}}{\sum^{NBclass} (\text{TP} + \text{FP})} \quad (3)$$

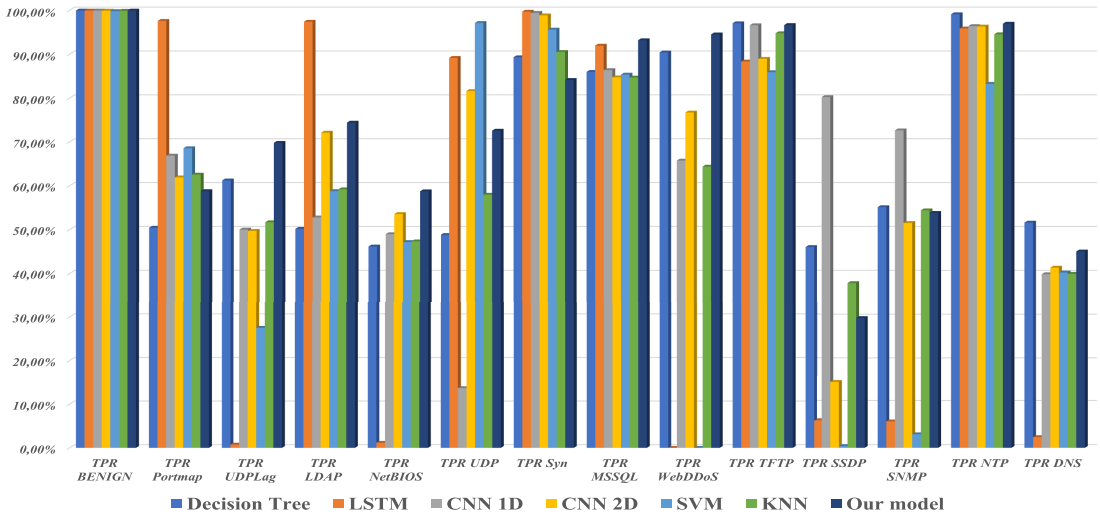
$$DR_{Average} = \frac{\sum TPR_{AttackType}}{NB_{OfAttackType}} \quad (4)$$

$$\text{FAR} = 1 - \frac{TP_{Benign}}{TP_{Benign} + FN_{Benign}} \quad (5)$$

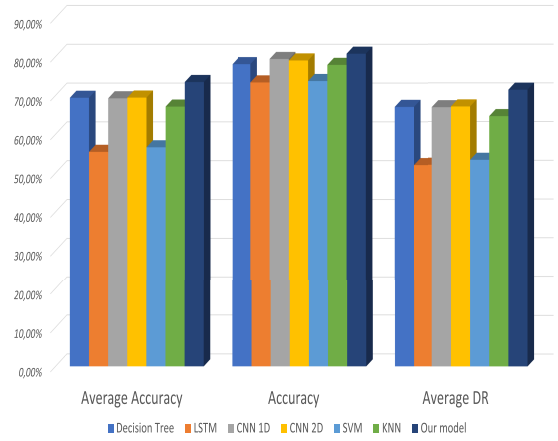
$$ACC_{Average} = \frac{1}{NBClass} \sum TPR_{ConnecType} \quad (6)$$

C. HYPERPARAMETERS USED

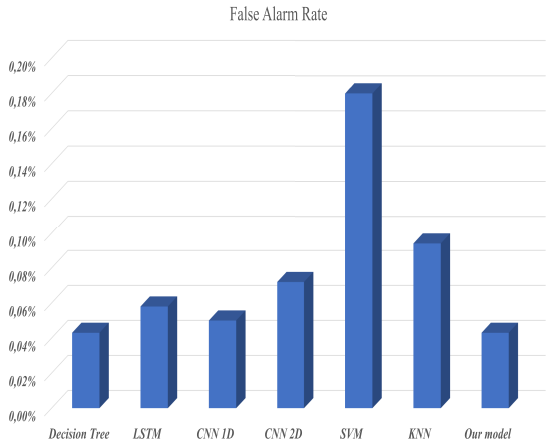
Fine-tuning is indeed a meticulous task in machine learning. In our case, we conducted numerous experiments to determine the optimal hyperparameters for training the various sub-models and the global model. We found that the choice of hyperparameters significantly impacts the performance of our model. Table 6 and Table 7 provide a summary of the best hyperparameters used to train the sub-models and the global model, respectively.



(a) TPR for each classifier



(b) Global metrics for each classifier



(c) FAR for each classifier

FIGURE 5. Global and specific Performance metrics of our model and some well-known classifiers.

TABLE 5. Hyperparameters used to train the different sub-models.

Hyperparameter	Value
Learning rate	0.001
Batch size	512
Number of epochs	700
loss function	categorical_crossentropy
Optimization algorithm	Adam

TABLE 6. Hyperparameters used to train our global model.

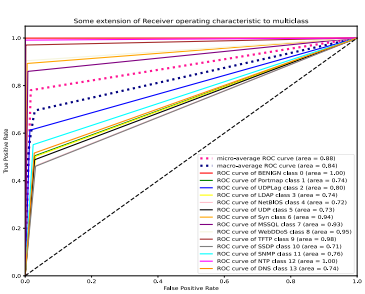
Hyperparameter	Value
Learning rate	0.001
Batch size	256
Number of epochs	300
loss function	categorical_crossentropy
Optimization algorithm	RMSprop

D. EXPERIMENTAL RESULTS

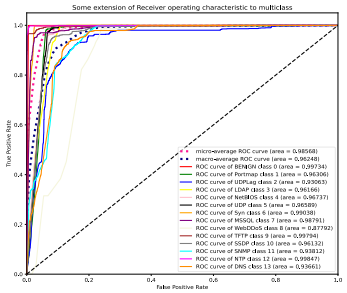
To evaluate the performance of our proposed model, we compared it to several well-known classifiers namely decision tree [50], K-nearest neighbors (KNN) [51], support vector machines (SVM) [52], long short-term memory (LSTM) [44], 1D convolutional neural network (CNN) [53], and 2D convolutional neural network (CNN) [54]. We assessed the performance using the previously detailed metrics.

Table 7 summarizes the True Positive Rate, Accuracy, Average Accuracy, Average Detection, and False Alarm Rate obtained by our model and the different classifiers used in this comparative study. As depicted in Figure 5 (a), our model provides the highest True Positive Rate (TPR) for five types of network connections, namely BENIGN, UDPLag, NetBIOS, MSSQL, and WebDDoS. Additionally, it achieves the second-highest TPR for four types of network connections, namely LDAP, TFTP, NTP, and DNS.

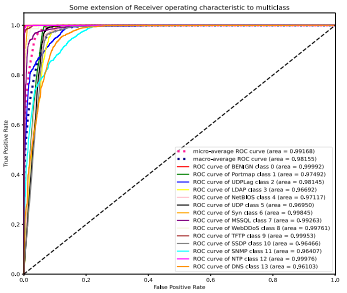
In Figures 5(b) and 5(c), we can see that our model outperforms the other models in terms of global metrics. Specifically, our model achieves the highest accuracy, the highest average accuracy, and the highest average



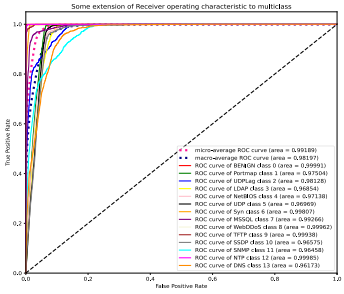
(a) Decision Tree



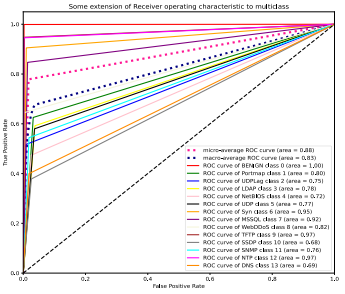
(b) LSTM



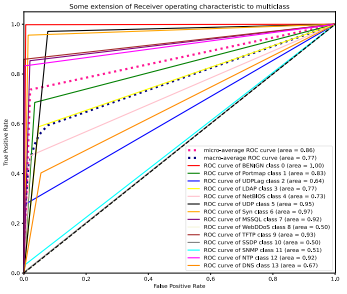
(c) CNN 1D



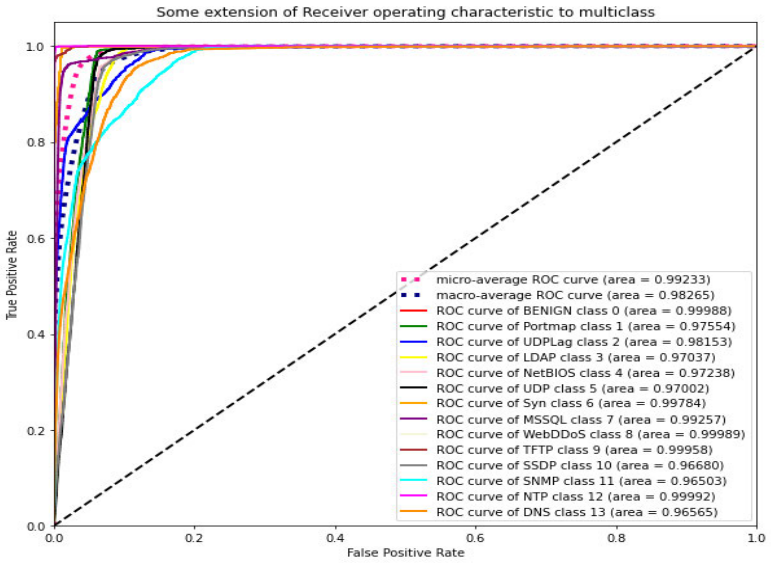
(d) CNN 2D



(e) KNN



(f) SVM



(g) Our Proposed model

FIGURE 6. **ROC curve and ROC area of our model and some well-known classifiers.**

detection rate, while also providing the lowest false alarm rate.

The exam of the different ROC curves and ROC-AUC values provided in Figure 6 shows that our model provides a high discriminative ability for the classes BENIGN, Syn, MSSQL, WebDDoS, TFTP, SSDP, NTP, and DNS compared to the other classes. The ROC curve for BENIGN, Syn, MSSQL, WebDDoS, TFTP, SSDP, NTP, and DNS exhibits a steep rise from the bottom-left corner towards the top-left corner, denoting a high True Positive Rate (TPR) and a relatively low False Positive Rate (FPR). The ROC-AUC values for BENIGN, Syn, MSSQL, WebDDoS, TFTP, SSDP, NTP, and DNS are all close to 1, which confirms the effectiveness of our model in accurately distinguishing

TABLE 7. Performances of our model and some well-known classifiers.

Metric \ Model		Decision Tree	LSTM	CNN 1D	CNN 2D	SVM	KNN	Our model
Metrics relative to each Type of connection	TPR BENIGN	99,96%	99,94%	99,95%	99,93%	99,82%	99,91%	99,96%
	TPR Portmap	50,46%	97,59%	66,89%	61,93%	68,58%	62,55%	58,80%
	TPR UDPLag	61,24%	0,76%	50,04%	49,74%	27,56%	51,70%	69,80%
	TPR LDAP	50,20%	97,41%	52,80%	72,13%	58,80%	59,21%	74,42%
	TPR NetBIOS	46,18%	1,17%	48,98%	53,56%	47,20%	47,35%	58,76%
	TPR UDP	48,80%	89,17%	13,71%	81,60%	97,12%	57,98%	72,57%
	TPR Syn	89,31%	99,69%	99,43%	98,85%	95,62%	90,51%	84,14%
	TPR MSSQL	85,97%	91,96%	86,39%	84,73%	85,35%	84,68%	93,19%
	TPR WebDDoS	90,41%	0,00%	65,75%	76,71%	0,00%	64,38%	94,52%
	TPR TFTP	97,06%	88,34%	96,60%	88,95%	85,91%	94,78%	96,66%
	TPR SSDP	46,06%	6,27%	80,23%	14,95%	0,40%	37,83%	29,85%
	TPR SNMP	55,14%	6,00%	72,65%	51,54%	3,07%	54,41%	53,83%
	TPR NTP	99,12%	95,87%	96,42%	96,32%	83,25%	94,55%	96,96%
	TPR DNS	51,62%	2,42%	39,86%	41,36%	40,24%	39,95%	45,04%
Global metrics	Average Accuracy	69,40%	55,47%	69,26%	69,45%	56,64%	67,13%	73,46%
	Accuracy	78,06%	73,35%	79,39%	79,02%	73,72%	77,87%	80,75%
	Average DR	67,04%	52,05%	66,90%	67,10%	53,32%	64,61%	71,42%
	FAR	0,04%	0,06%	0,05%	0,07%	0,18%	0,09%	0,04%

TABLE 8. Times performance.

Model \ Time criteria	Traning Time	Test Time	Time to classify one row
Decision Tree	5.094S	0.019S	0.0005 ms
LSTM	1319.393S	5.094S	0.1365 ms
CNN 1D	675.016s	3.281s	0.0879 ms
CNN 2D	683.195s	5.272s	0.1413 ms
SVM	428.241 s	211.207s	5.6609 ms
KNN	0.049s	30.594s	0.8200 ms
Our Model	442800s	144.040s	3.8606 ms

network connections belonging to these classes from the rest. On the other hand, the ROC curves for the classes Portmap, UDPLag, LDAP, NetBIOS, UDP, and SNMP demonstrate more moderate increases. Additionally, the ROC-AUC values for Portmap, UDPLag, LDAP, NetBIOS, UDP, and SNMP range between 96% and 98%.

These results imply that the model's ability to discriminate these classes may be relatively lower compared to the other classes of network connection. When comparing our model to other models, particularly in terms of the average Micro and Macro ROC Curve and ROC-AUC values, we observe that our model shows the highest average Micro and Macro ROC Curve. Furthermore, our model gives the highest or nearly the highest steepness from the bottom-left corner towards the top-left corner across the different classes when considering ROC curves. Additionally, our model achieves the highest average Micro and Macro ROC-AUC values, as well as the highest or close to the highest ROC-AUC values for the individual classes. These findings provide unequivocal evidence of the exceptional performance of our model.

After comparing the performance of your model with other well-known classifiers, it is common to evaluate and compare several processing time factors, including training time, test time, and one-row processing time. These metrics help in assessing the efficiency and effectiveness of different classifiers. Furthermore, the utilization of time processing comparison assists us in assessing the feasibility of deploying our model within the specific location outlined in our architecture. It is worth noting that these results were obtained using the Kaggle environment. As indicated in Table 8, our model's training time is significantly higher due to its complexity. However, the extended training time does not impact the model's performance since it runs only once on a high-performance machine with powerful GPUs located in the cloud, which substantially reduces the training time. Moreover, to meet the needs of updating, we will apply transfer learning. In this approach, the trained model will serve as the base model, and we will continue the training process using the new training data combined with samples from the old dataset. Utilizing transfer learning in an offline manner ensures the retraining and updating of our model at minimal cost, especially in terms of training time.

Regarding the test time, it remains reasonable despite the complexity of our model. This characteristic enables its deployment on both cloud and Fog sites. The test time mentioned in the previous statement was obtained for a dataset consisting of 37,310 rows. This indicates that the average time required to process one row is only 3.8606 milliseconds. This rapid processing time allows real-time detection of DDoS attacks.

V. CONCLUSION

In this paper, we present a novel hybrid deep learning model that combines various types of deep neural networks,

including CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), Deep Autoencoder, and DNN (Deep Neural Network). Our model operates on two levels: the first level comprises parallel sub-neural networks trained using our newly developed training algorithm, while the second level combines the initial data with the outputs from the first level. Through the integration of diverse deep neural network types, our objective is to leverage their diverse characteristics and create a high-performing model.

The results obtained using the CIC-DDoS2019 dataset demonstrate the superiority of our model, achieving an average detection rate of 71.42%, a global accuracy of 80.75%, an average accuracy of 73.46%, and a false alarm rate of 0.04%. These results outperform several well-known classifiers and deep learning models.

Moreover, these results were obtained on a dataset that includes various types of DDoS attacks, encompassing both the most prevalent and similar ones. In our future work, we will explore hyperparameter optimization and feature selection to further enhance the results.

REFERENCES

[1] L. S. Vailshery. *IoT Connected Devices Worldwide 2019-2030*. Accessed: Nov. 1, 2022. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide

[2] I. Ben Dhaou, M. Ebrahimi, M. Ben Ammar, G. Bouattour, and O. Kanoun, "Edge devices for Internet of Medical things: Technologies, techniques, and implementation," *Electronics*, vol. 10, no. 17, p. 2104, Aug. 2021.

[3] I. Ahmim, N. Ghoualmi-Zine, A. Ahmim, and M. Ahmim, "Security analysis on 'three-factor authentication protocol using physical unclonable function for IoV,'" *Int. J. Inf. Secur.*, vol. 21, no. 5, pp. 1019–1026, Oct. 2022.

[4] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[5] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[6] (Feb. 17, 2023). *Top 5 Most Famous DDoS Attacks | Microsoft 365*. [Online]. Available: https://www.microsoft.com/en-us/microsoft365lifehacks/privacyandsafety/top5mostfamousddosattacks

[7] X. Chen, L. Xiao, W. Feng, N. Ge, and X. Wang, "DDoS defense for IoT: A Stackelberg game model-enabled collaborative framework," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9659–9674, Jun. 2022.

[8] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A novel hierarchical intrusion detection system based on decision tree and rules-based models," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2019, pp. 228–233.

[9] X. Liu and J. Liu, "Malicious traffic detection combined deep neural network with hierarchical attention mechanism," *Sci. Rep.*, vol. 11, no. 1, p. 12363, Jun. 2021.

[10] L. Chen, S. Gao, and B. Liu, "An improved density peaks clustering algorithm based on grid screening and mutual neighborhood degree for network anomaly detection," *Sci. Rep.*, vol. 12, no. 1, p. 1409, Jan. 2022.

[11] D. Gümüsbas, T. Yildirim, A. Genovese, and F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," *IEEE Syst. J.*, vol. 15, no. 2, pp. 1717–1731, Jun. 2021.

[12] C. Zhang, D. Jia, L. Wang, W. Wang, F. Liu, and A. Yang, "Comparative research on network intrusion detection methods based on machine learning," *Comput. Secur.*, vol. 121, Oct. 2022, Art. no. 102861.

[13] A. Zainudin, L. A. C. Ahakonye, R. Akter, D.-S. Kim, and J.-M. Lee, "An efficient hybrid-DNN for DDoS detection and classification in software-defined IIoT networks," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8491–8504, May 2023.

[14] H. Aydın, Z. Orman, and M. A. Aydın, "A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment," *Comput. Secur.*, vol. 118, Jul. 2022, Art. no. 102725.

[15] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: A hybrid deep learning approach for DDoS detection and classification," *IEEE Access*, vol. 9, pp. 146810–146821, 2021.

[16] S. V. J. Rani, I. Ioannou, P. Nagaradjane, C. Christophorou, V. Vassiliou, S. Charan, S. Prakash, N. Parekh, and A. Pitsillides, "Detection of DDoS attacks in D2D communications using machine learning approach," *Comput. Commun.*, vol. 198, pp. 32–51, Jan. 2023.

[17] M. S. E. Sayed, N.-A. Le-Khac, M. A. Azer, and A. D. Jurcut, "A flow-based anomaly detection approach with feature selection method against DDoS attacks in SDNs," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 4, pp. 1862–1880, Dec. 2022.

[18] A. Makuvaza, D. S. Jat, and A. M. Gamundani, "Deep neural network (DNN) solution for real-time detection of distributed denial of service (DDoS) attacks in software defined networks (SDNs)," *Social Netw. Comput. Sci.*, vol. 2, pp. 1–10, Feb. 2021.

[19] A. Agarwal, M. Khari, and R. Singh, "Detection of DDoS attack using deep learning model in cloud storage application," *Wireless Pers. Commun.*, vol. 127, no. 1, pp. 419–439, Nov. 2022.

[20] Y. Liu, T. Zhi, M. Shen, L. Wang, Y. Li, and M. Wan, "Software-defined DDoS detection with information entropy analysis and optimized deep learning," *Future Gener. Comput. Syst.*, vol. 129, pp. 99–114, Jan. 2022.

[21] S. Singh and S. K. V. Jayakumar, "DDoS attack detection in SDN: Optimized deep convolutional neural network with optimal feature set," *Wireless Pers. Commun.*, vol. 125, no. 3, pp. 2781–2797, Aug. 2022.

[22] G. S. R. E. Selvan, R. Ganeshan, I. D. J. Jingle, and J. P. Ananth, "FACVO-DNFN: Deep learning-based feature fusion and distributed denial of service attack detection in cloud computing," *Knowl.-Based Syst.*, vol. 261, Feb. 2023, Art. no. 110132.

[23] B. Hussain, Q. Du, B. Sun, and Z. Han, "Deep learning-based DDoS-attack detection for cyber–physical system over 5G network," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 860–870, Feb. 2021.

[24] R. Priyadarshini and R. K. Barik, "A deep learning based intelligent framework to mitigate DDoS attack in fog environment," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 3, pp. 825–831, 2022.

[25] R. F. Fouladi, O. Ermiş, and E. Anarim, "A DDoS attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN," *Comput. Netw.*, vol. 214, Sep. 2022, Art. no. 109140.

[26] A. A. Elsaeidy, A. Jamalipour, and K. S. Munasinghe, "A hybrid deep learning approach for replay and DDoS attack detection in a smart city," *IEEE Access*, vol. 9, pp. 154864–154875, 2021.

[27] L. Zhou, Y. Zhu, T. Zong, and Y. Xiang, "A feature selection-based method for DDoS attack flow classification," *Future Gener. Comput. Syst.*, vol. 132, pp. 67–79, Jul. 2022.

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

[29] B. L. Dalmazo, V. M. Deolindo, and J. C. Nobre, "Public dataset for evaluating Port Scan and Slowloris attacks," Harvard Dataverse, V1, 2019, doi: 10.7910/DVN/ZJOT5G.

[30] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, vol. 8, pp. 165263–165284, 2020.

[31] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, vol. 1, Jan. 2018, pp. 108–116.

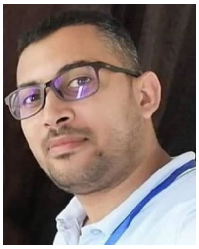
[32] UCI KDD. (1999). *The Third International Knowledge Discovery and Data Mining Tools Competition Dataset KDD Cup 1999 Data*. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[33] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. for Secur. Defense Appl.*, Jul. 2009, pp. 1–6.

[34] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[35] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of Milan and the province of trentino," *Scientific Data*, vol. 2, no. 1, pp. 1–15, Oct. 2015.

[36] P. A. A. Resende and A. C. Drummond, "HTTP and contact-based features for botnet detection," *Secur. PRIVACY*, vol. 1, no. 5, p. e41, Sep. 2018.
[37] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho, "Seven years and one day: Sketching the evolution of internet traffic," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 711–719.
[38] M. Arlitt and T. Jin, "A workload characterization study of the 1998 world cup web site," *IEEE Netw.*, vol. 14, no. 3, pp. 30–37, 2000.
[39] A. Elsaeidy. (Nov. 2021). *Queanbeyan Smart City Platform Datasets*. [Online]. Available: https://github.com/asmaa-elsaeidy/QBN-Smart-CityDataset.git
[40] T. G. Palla and S. Tayeb, "Intelligent Mirai malware detection for IoT nodes," *Electronics*, vol. 10, no. 11, p. 1241, May 2021.
[41] (2021). *Information Marketplace for Policy and Analysis of Cyber-Risk & Trust*. Accessed: Mar. 10, 2023. [Online]. Available: https://www.impactcybertrust.org/dataset_view?idDataset=742
[42] CAIDA. (2007). *Caida DDoS Attack Dataset*. Accessed: Mar. 10, 2023. [Online]. Available: https://www.caida.org/data/passive/ddos20070804_dataset.xml
[43] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[45] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proc. ICML Workshop Unsupervised Transf. Learn.*, 2012, pp. 37–49.
[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[47] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[48] *Kaggle Com*. Accessed: May 1, 2023. [Online]. Available: https://www.kaggle.com
[49] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset," in *Proc. Int. Conf. Inf. Sci. Secur. (ICISS)*, Dec. 2016, pp. 1–6.
[50] L Breiman, J Friedman, R Olshen, and C Stone, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.
[51] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.
[52] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, Apr. 1995.
[53] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Mar. 1998.
[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.



AHMED AHMIM received the bachelor's, master's, and Ph.D. degrees in computer science from Badji Mokhtar-Annaba University, Algeria, in 2007, 2009, and 2014, respectively. From May 2015 to September 2019, he was an Assistant Professor with the Department of Mathematics and Computer Science, University of Larbi Tebessi—Tebessa, Algeria. Since October 2019, he has been a Senior Lecturer with the Department of Mathematics and Computer Science, Souk Ahras University, Algeria. His research interests include the IoT, computer security, network security, machine learning, deep learning, federated learning, and intrusion detection systems. He has served as a reviewer for various journals, including Elsevier, IEEE, Springer, and Wiley. Additionally, he has been actively involved in organizing international conferences, serving as an organizing committee member in various capacities, such as the track chair, the co-chair, the publicity chair, and a proceedings editor.

FAIZ MAAZOUZI received the bachelor's, master's, and Ph.D. degrees in computer science from Badji Mokhtar-Annaba University, Algeria, in June 2007, June 2009, and March 2014, respectively. Since June 2014, he has been an Assistant Professor with the Department of Mathematics and Computer Science, Souk Ahras University, Algeria. His research interests include data mining, signal processing, and machine learning. He has served as an organizing committee member [the Track Chair, the Co-Chair, and a technical program committee (TPC) member] in numerous international conferences.

MARWA AHMIM received the Ph.D. degree in computer science from Badji Mokhtar-Annaba University, Algeria, in 2016. Currently, she is an Associate Professor with the Department of Computer Science, Badji Mokhtar-Annaba University. Her current research interests include network security, security metrics, blockchain, cryptography, and the Internet of Things security.

SARRA NAMANE received the Ph.D. degree in computer science from Badji Mokhtar-Annaba University, in 2018. She is currently an Associate Professor with the Department of Computer Science, Badji Mokhtar-Annaba University. Her research interests include network security, grid computing security, cloud computing security, access control techniques, blockchain, and the Internet of Things security.

IMED BEN DHAOU (Senior Member, IEEE) received the Ph.D. degree from the Royal Institute of Technology, Sweden.

He is currently an associate professor and a docent in embedded systems for IoT. Since 2021, he has been with the Department of Computer Science, Dar Al-Hekma University. He has authored and coauthored more than 110 journals and conference papers, book chapters, and technical reports.

In recognition of his commitment to scientific inquiry and innovation, he is a valued member of Sigma Xi, The Scientific Research Honor Society. He received numerous awards, including the Best Paper Award from the 1997 Finnish Symposium on Signal Processing, a travel grant from the Ph.D. forum at DAC (Los Angeles, 2000), a Publication Award from Qassim University, and Dr. Hussein Mohammed Al-Sayyed Award for Research. Since September 2014, he has been an Editor of *Microelectronics Journal* (Elsevier). He served as a Guest Editor for four special issues of the ISI journals (*Electronics*, *Journal of Cloud Computing*, *Analogue Integrated Circuits and Signal Processing*, and *Microprocessors and Microsystems*). He has chaired or served on the TPC for various conferences in his core areas of expertise.