

# Anomaly-Based Intrusion Detection System for DDoS Attack with Deep Learning Techniques

Davide Agostinello<sup>a</sup>, Angelo Genovese<sup>b</sup>, Vincenzo Piuri<sup>c</sup>

*Università degli studi di Milano, Department of Computer Science*  
davideagostinello@gmail.com {angelo.genovese, vincenzo.piuri}@unimi.it

**Keywords:** Deep Learning, Intrusion Detection System, DDoS.

**Abstract:** The increasing number of connected devices is fostering a rising frequency of cyber attacks, with Distributed Denial of Service (DDoS) attacks among the most common. To counteract DDoS, companies and large organizations are increasingly deploying anomaly-based Intrusion Detection Systems (IDS), which detect attack patterns by analyzing differences in malicious network traffic against a baseline of legitimate traffic. To differentiate malicious and normal traffic, methods based on artificial intelligence and, in particular, Deep Learning (DL) are being increasingly considered, due to their ability to automatically learn feature representations for the different traffic types, without need of explicit programming or handcrafted feature extraction. In this paper, we propose a novel methodology for simulating an anomaly-based IDS based on adaptive DL by designing multiple DL models working with both binary and multi-label classification on multiple datasets with different degrees of complexity. To make the DL models adaptable to different conditions, we consider adaptive architectures obtained by automatically tuning the number of neurons for each situation. Results on publicly-available datasets confirm the validity of our proposed methodology, with DL models adapting to the different conditions by increasing the number of neurons on more complex datasets and achieving the highest accuracy in the binary classification configuration.

## 1 INTRODUCTION


The rising number of connected devices and network traffic is causing a growth in the number of cyber attacks. Cyber security is then increasingly being prioritized in a wide number of situations, from small networks at a company level to large infrastructures at a national level. Among cyber attacks, Distributed Denial of Service (DDoS) represents one of the most widespread. A DDoS attack consists in the saturation of the resources of a service through the overload of requests coming from multiple sources previously compromised by a malware. As a result of a DDoS attack, the service is no longer available to legitimate users.


To protect against cyber attacks and DDoS attacks, Intrusion Detection Systems (IDSs) are being increasingly studied and deployed. In particular, anomaly-based IDSs work by establishing a baseline of “normal” network traffic and detecting “malicious” traffic and hence possible attacks when significant differences from the baseline are detected. Recent IDSs


often rely on Artificial Intelligence and in particular Deep Learning (DL), which represents a promising solution to differentiate normal from malicious traffic, due to the ability of DL models of automatically learning feature representations and perform classification with high accuracy (Gümüşbaşı et al., 2021).

DL-based approaches for IDSs can be categorized by considering the learning paradigm and the type of classification. Learning paradigms comprise supervised instance learning, which includes methods using labeled instances and based on Deep Neural Networks (DNN) or Convolutional Neural Networks (CNN) and supervised sequence learning, which includes methods using series of data and based on Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM). Moreover, the type of classification can either be binary, by classifying traffic as either “normal” or “DDoS”, or multi-label, by also classifying different types of malicious traffic (Mittal et al., 2022).

In this paper, we propose a simulation of an anomaly-based IDS using adaptive DL models in different configurations, obtained by considering DNN- and CNN-based models (supervised instance learning) and RNN-based models (supervised sequence

<sup>a</sup>  <https://orcid.org/0009-0006-5629-0483>

<sup>b</sup>  <https://orcid.org/0000-0002-3683-4723>

<sup>c</sup>  <https://orcid.org/0000-0003-3178-8198>

learning approaches) with both a binary and a multi-label classification variant. To determine which configuration works best and in what conditions, in all the models and classification variants we consider different adaptive architectures, obtained by automatically optimizing the number of neurons in the hidden layers, adding the capability of the considered architectures to adapt to different conditions.

To the best of our knowledge, no method in the literature has performed a simulation of different DL-based models for IDSs by considering both supervised instance learning and supervised sequence learning approaches, in both a binary and multi-label classification, and using adaptive architectures with an optimized number of neurons.

The remainder of the paper is structured as follows. Section 2 presents the related works, distinguishing the different learning paradigms that can be applied. Section 3 describes our methodology. Section 4 introduces the datasets used in our experimental evaluation. Section 5 describes the experimental results. Finally, Section 6 concludes the paper.

## 2 RELATED WORKS

Anomaly-based IDSs using DL can be distinguished in *i*) supervised instance learning; *ii*) supervised sequence learning, *iii*) semi-supervised learning, and *iv*) hybrid learning (Mittal et al., 2022). We describe each of them next.

**Supervised Instance Learning:** this category includes methods that train the models using labeled instances, such as the ones based on DNN and CNN. In particular, we can distinguish *i*) methods based on DNN and *ii*) methods based on CNN.

Methods based on DNN include the works described in (Asad et al., 2020; Cil et al., 2021; Sbai and El boukhari, 2020), which detect DDoS attacks by training and testing the model on the CIC-IDS-2017 and CIC-DDoS-2019 datasets. The work described in (Sabeel et al., 2019) considers DNN as well as LSTM models to predict both DoS and unknown attacks, by training the models using the CIC-IDS-2017 dataset and testing them on a self-produced dataset. A DNN architecture is also considered in the work presented in (Amaizu et al., 2021), with the difference of using an ensemble of two DNNs to increase the classification accuracy.

Methods based on CNN include the approach presented in (Kim et al., 2020), which trains a network on the CIC-IDS 2018 and KDD datasets to detect DoS attacks. To improve the recognition accuracy with respect to using a single CNN, the method introduced in

(Haider et al., 2020) considers an ensemble of CNNs, while the work described in (Wang and Liu, 2020) presents a multi-level framework that first performs a coarse detection of suspicious traffic, then uses the CNN to make a fine-grained distinction between normal and malicious traffic.

Differently from the CNN-based approaches described in (Kim et al., 2020; Haider et al., 2020; Wang and Liu, 2020), which aim at increasing the detection accuracy, the methods proposed in (Doriguzzi-Corin et al., 2020; de Assis et al., 2020) have the purpose of reducing the computational complexity of CNN-based DDoS detection and allow a deployment of DL-based IDS in real-time scenarios or on devices with limited resources. In particular, the approach proposed in (de Assis et al., 2020) considers one-dimensional –rather than two-dimensional– convolutional layers. Similarly, the work presented in (Doriguzzi-Corin et al., 2020) considers a CNN with a reduced number of one-dimensional convolutional layers in combination with an ad-hoc preprocessing algorithm.

**Supervised Sequence Learning:** this category includes methods that train the models using series of data, such as the ones based on RNN, LSTM, and Gated Recurrent Units (GRU); In particular, we can distinguish *i*) methods based on LSTM and *ii*) methods based on GRU.

Methods based on LSTM include the work described in (Liang and Znati, 2019), which applies the model on raw network flow with the purpose of differentiating normal traffic and DDoS attacks. The method presented in (Ferrag et al., 2021) also considers LSTM, adding a comparison with CNNs and DNN within the applicative scenario of IoT-enabled networks.

Methods based on GRU include the paper proposed in (Assis et al., 2021), which showed that GRUs resulted in greater accuracy in detecting DDoS attacks with respect to LSTMs, despite using fewer parameters.

**Semi-Supervised Learning:** this category includes methods that train the models using unlabeled data in the pre-training stage, then fine-tune them using a combination of both labeled and unlabeled data, such as the ones based on Auto-Encoders (AE) and Support Vector Machines (SVM). For example, the work described in (Kasim, 2020) preprocesses data with a principal component analysis, then applies an AE to further reduce data dimensionality and extract the features. Lastly, it considers an SVM to perform the classification and detect DDoS attacks. Similarly, the approach proposed in (Bhardwaj et al., 2020) describes an AE for feature extraction, followed by a DNN for

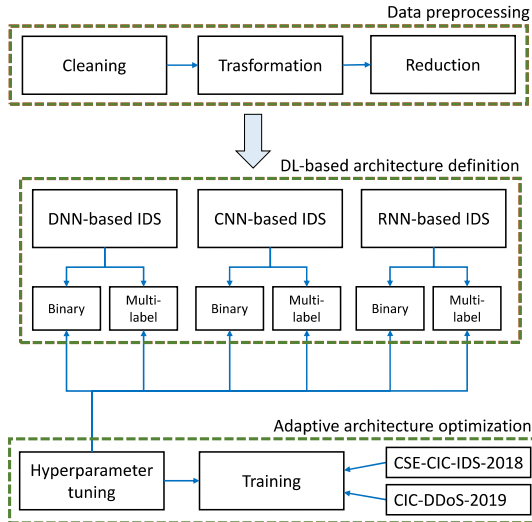


Figure 1: Outline of our methodology. After the data preprocessing step, we define six different configurations of DL-based architectures, considering DNN-, CNN-, and RNN-based models, each in a binary and a multi-label classification variant. Then, we perform the optimization by tuning the hyperparameters of each adaptive architecture and then training the DL-based models on the considered datasets for DDoS detection.

feature classification.

**Hybrid Learning:** this category includes methods that combine two or more approaches, usually to leverage the capability of sequence learning approaches (e.g., LSTM) to process series of data with the capability of instance learning approaches (e.g., CNN) to detect specific patterns when trained using labeled data. For example, the works presented in (Roopak et al., 2019; Roopak et al., 2020) combine a model based on a CNN with a model based on a LSTM, then use a fully-connected layer to output the results, achieving a superior accuracy with respect to using the two models separately. Similarly, the approach described in (Elsayed et al., 2020) combines an AE to perform a feature extraction, a RNN to process the series of data, and a fully-connected layer to perform the classification.

In the literature, methods based on supervised instance learning (e.g., DNN, CNN) and supervised sequence learning (e.g., RNN) are the most studied and represent the majority of the approaches for anomaly-based IDSs. Therefore, in our work we consider DL-based models based on DNN, CNN, and RNN. However, no method in the literature has performed a simulation of different DL-based models for IDSs by considering adaptive architectures with an optimized number of neurons, using both a binary and a multi-

label classification.

### 3 PROPOSED METHODOLOGY

This section describes the proposed methodology for simulating an anomaly-based IDS for DDoS attack detection. After a preprocessing step, necessary to uniform the different datasets in a common format, we define the different configurations obtained by considering DNN- and CNN-based models (supervised instance learning) and RNN-based models (supervised sequence learning approaches) in both a binary and a multi-label classification variant. Then, in all the models and classification types we consider adaptive architectures, obtained by automatically optimizing the number of neurons in the hidden layers.

The methodology comprises three steps: *i*) data preprocessing, *ii*) DL-based architecture definition, and *iii*) adaptive architecture optimization. Fig.1 outlines our methodology: after the data preprocessing step, we define six different configurations of DL-based architectures, considering DNN-, CNN-, and RNN-based models, each in a binary and a multi-label classification variant. Then, we perform the optimization by tuning the hyperparameters of each adaptive architecture and then training the DL-based models on the considered datasets for DDoS detection.

#### 3.1 Data Preprocessing

Data preprocessing has the purpose of uniforming the different datasets into a common format, by removing inconsistent or missing values, encoding the labels in numerical values, and reducing the dimensionality, following common practices when preparing data for intrusion detection (Srikanth Yadav. and Kalpana., 2019; Alasadi and Bhaya, 2017). This step consists of three tasks: *i*) cleaning, *ii*) transformation, and *iii*) reduction.

**Cleaning:** this task involves the removal of inconsistent or missing values. First, we remove columns that do not contain values useful for model training, such as socket-related features and columns containing only zeros. Then, we remove duplicate rows and rows containing *NaNs*. Lastly, we replace infinite and null values with  $-1$ , as described in (Cil et al., 2021).

**Transformation:** this task involves the transformation of the dataset to ensure comparable numerical values across different datasets. First, we normalize the numerical values in the range  $[0, 1]$  using the min-max method. Second, we apply label encoding to categorical features, by transforming categorical values

into numerical values, considering two methods proposed in the literature:

- *Label encoder*: involves the conversion of each label into a number;
- *One Hot Encoder (OHE)*: involves the conversion of the label of each row into a vector  $\mathbf{v}$  of  $n$  columns, where  $n$  is the number of labels, with  $v_i = 1$  if the row is associated with the  $i$ -th label, and  $v_i = 0$  otherwise.

**Reduction**: this task involves reducing the dimensionality of the data to ensure a common number of features in the different datasets, reduce noise in the features, and speed up the training of DL-based models. To perform the reduction, first we apply the PCA technique, then we select the number of principal components using the MLE method (Minka, 2000).

### 3.2 DL-based Architecture Definition

In this step we define the different DL architectures that include supervised instance learning approaches such as DNN and CNN, as well as supervised sequence learning approaches such as RNN. For all models, we consider adaptive architectures, in which the hidden layers have a variable size.

In our work we consider three different architectures: *i*) DNN, *ii*) CNN, and *iii*) RNN. For each architecture, we consider two variants of the classification types, one performing a binary classification and one performing a multi-label classification.

#### 3.2.1 DNN-based IDS

As a first DL architecture, we consider a supervised instance learning approach based on a DNN, with a variable number of neurons in the hidden layer(s). We consider two variants, one for binary classification and one for multi-label classification.

**Binary Classification**: we design the DNN-based IDS for binary classification with an input layer, a hidden layer, and an output layer. We consider a Rectified Linear Unit (ReLU) as activation function in the input and hidden layers, while we use a sigmoid in the output layer.

**Multi-Label Classification**: we design the multi-label version with an additional hidden layer with respect to the binary version, and also three dropout layers placed after the input and each hidden layer to reduce the possibility of overfitting. We consider the ReLU as activation function in the input and hidden layers, and a softmax function in the output layer.

#### 3.2.2 CNN-based IDS

As a second DL architecture, we consider a supervised instance learning approach based on a CNN, with a variable number of neurons in the hidden layer(s). We consider two variants, one for binary classification and one for multi-label classification.

**Binary Classification**: we design the CNN-based IDS for binary classification with an input layer, a one-dimensional convolutional layer, a sub-sampling layer, and a fully-connected layer. In the convolutional layer we consider  $stride=1$  and  $padding=0$ . We used average-pooling instead of max-pooling as the sub-sampling-layer to have a more uniform sampling of the output coming from the convolutional layer. We used the sigmoid as activation function in the output layer.

**Multi-Label Classification**: we design the multi-label version by considering an input layer, two one-dimensional convolutional layers, a sub-sampling layer, two fully-connected layers, and an output layer. We used the ReLU as activation function in the convolutional layers and a softmax in the output layer.

#### 3.2.3 RNN-based IDS

As a third DL architecture, we consider a supervised sequence learning approach based on a RNN, with a variable number of neurons in the hidden layer(s). We consider two variants, one for binary classification and one for multi-label classification.

**Binary Classification**: we design the RNN-based IDS for binary classification by considering a LSTM model with an input layer, an LSTM layer, a dropout layer, and a fully-connected layer. We use a sigmoid activation function in the output layer.

**Multi-Label Classification**: we design the multi-label version starting from the same architecture as the one for binary classification and increasing the number of neurons in the LSTM layer.

### 3.3 Adaptive Architecture Optimization

In this step we perform the optimization of the considered architectures to ensure the adaptability of the DL-based models to the different conditions. For all the architectures considered in this work, we consider hidden layers with a variable size and use a hyperparameter tuning method to automatically optimize the number of neurons in the hidden layers, adding the capability of the considered architectures to adapt to different conditions. In particular, we perform the optimization of the hyperparameters by tuning the num-

Table 1: CSE-CIC-IDS-2018 dataset: size and number of rows for the 4 CSV files.

Filename	Size [MB]	N. rows
02-15-2018	358.53	1,048,575
02-16-2018	318.26	1,048,575
02-20-2018	3867.08	7,948,748
02-21-2018	313.66	1,048,575
<i>Total</i>	<i>4,858</i>	<i>11,094,473</i>

Table 2: CSE-CIC-IDS-2018 dataset: number of rows describing normal and malicious traffic.

Traffic type	Rows	
	#	%
Normal	9,176,239	82.7
Malicious	1,918,233	17.3

ber of units in hidden layers and the learning rate using the Hyperband method. We chose this method since it allows a good trade-off between times, resources and results.

After optimizing the number of neurons and tuning the learning rate, we train the models considering a binary cross entropy loss function in the case of binary classification, while we considered a categorical cross entropy in multi-label classification.

## 4 DATASETS

In this work we consider two recent datasets, CSE-CIC-IDS-2018 and CIC-DDoS-2019, both developed by the Canadian Institute for Cybersecurity (CIC) of the University of New Brunswick (CIC, 2018; CIC, 2019). We chose these two datasets since the datasets released by CIC are among the most used in the literature and are representative of real network traffic (Gümüşbaş et al., 2021).

**CSE-CIC-IDS-2018:** from the CSE-CIC-IDS-2018 dataset, we select the parts describing DDoS traffic and normal traffic. As a result, we obtain 4 files, with each file having 84 columns, 7 of which describe socket-related features (*Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol and Timestamp*). The resulting dataset describes 7 types of DDoS attacks, namely *GoldenEye, Slowloris, Hulk, SlowHTTPTest, LOIC-HTTP, HOIC, LOIC-UDP*. Table 1 presents an overview of the files in the dataset, along with the number of rows in each file, while Table 2 outlines the percentage of rows belonging to normal and malicious traffic.

**CIC-DDoS-2019:** the CIC-DDoS-2019 dataset con-

Table 3: CIC-DDoS-2019 dataset: size and number of rows for the 18 CSV files.

Filename	Size [MB]	N. rows
train/DrDoS_LDAP	874.81	2,181,542
train/DrDoS_MSSQL	1801.66	4,524,498
train/DrDoS_NetBIOS	1618.84	4,094,986
train/DrDoS_SNMP	2071.93	5,161,377
train/DrDoS_SSDP	1194.65	2,611,374
train/DrDoS_UDP	1436.27	3,136,802
train/UDPLag	150.65	370,605
train/Syn	607.79	1,582,681
train/TFTP	8871.09	20,107,827
train/DrDoS_DNS	2034.48	5,074,413
train/DrDoS_NTP	615.13	1,217,007
test/Portmap87f	74.97	191,694
test/NetBIOS	1352.76	3,455,899
test/LDAP87f	831.03	2,113,234
test/MSSQL	2275.68	5,775,786
test/UDP	1709.74	3,782,206
test/UDPLag87f	304.98	725,165
test/Syn	1790.40	4,320,541
<i>Total</i>	<i>29616.86</i>	<i>70,427,637</i>

Table 4: CIC-DDoS-2019 dataset: number of rows describing normal and malicious traffic.

Traffic type	Rows	
	#	%
Normal	113,828	0.16
Malicious	70,313,809	99.84

sists of 18 files, divided into training and testing folders. Each file contains 87 feature columns: the first 80 columns contains features extracted with CICFlowMeter (Lashkari et al., 2017), while the remaining 7 columns contain socket-related features (*Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol and Timestamp*). The dataset describes 18 types of attacks, including reflection- and exploitation-based attacks: *DrDoS-LDAP, DrDoS-MSSQL, DrDoS-NetBIOS, DrDoS-NMP, DrDoS-SSDP, DrDoS-UDP, UDP-lag, WebDDoS, Syn, TFTP, DrDoS-DNS, DrDoS-NTP, Portmap, NetBIOS, LDAP, MSSQL, UDP, and UDPLag*. Table 3 presents an overview of the files in the dataset, along with the number of rows in each file. The majority of the rows in the dataset belong to malicious traffic, with > 99% of data describing attacks, as outlined in Table 4.

Table 5: Result of preprocessing on the CSE-CIC-IDS-2018 dataset.

Filename	N. rows (initial)	Duplicates		N. inf	N. NaN	N. rows (final)
		#	%			
02-15-2018.csv	1,048,575	253,000	24.13	979	547	795,028
02-16-2018.csv	1,048,575	456,701	43.55	0	0	591,874
02-20-2018.csv	7,948,748	2,511,282	31.59	2,146	1,200	5,436,266
02-21-2018.csv	1,048,575	487,179	46.46	0	0	561,396
<i>Total</i>	<i>11,094,473</i>	<i>3,708,162</i>	<i>33.42</i>	<i>3,125</i>	<i>1,747</i>	<i>7,384,564</i>

Table 6: Result of preprocessing on the CIC-DDoS-2019 dataset.

Filename	N. rows (initial)	Duplicates		N. inf	N. NaN	N. rows (final)
		#	%			
train/DrDoS_LDAP	2,181,542	2,150,051	98.56	2,480	10	31,491
train/DrDoS_MSSQL	4,524,498	4,315,627	95.38	26,715	9	208,871
train/DrDoS_NetBIOS	4,094,986	4,073,870	99.48	2,968	8	21,116
train/DrDoS_SNMP	5,161,377	5,045,899	97.76	4,135	11	115,478
train/DrDoS_SSDP	2,611,374	1,719,417	65.84	1,538	2	891,957
train/DrDoS_UDP	3,136,802	2,059,422	65.65	1,401	7	1,077,380
train/UDPLag	370,605	277,586	74.90	68	2	93,019
train/Syn	1,582,681	1,426,794	90.15	30	6	155,887
train/TFTPSv	20,107,827	15,688,113	78.02	210	22	4,419,714
train/DrDoS_DNS	5,074,413	4,958,122	97.71	10,146	22	116,291
train/DrDoS_NTP	1,217,007	90,774	7.46	225	25	1,126,233
test/Portmap87f	191,694	185,605	96.82	143	1	6,089
test/NetBIOS	3,455,899	3,444,578	99.67	397	3	11,321
test/LDAP87f	2,113,234	2,089,308	98.87	2,911	7	23,926
test/MSSQL	5,775,786	5,501,052	95.24	36,281	5	274,734
test/UDP	3,782,206	2,484,620	65.69	2,286	2	1,297,586
test/UDPLag87f	725,165	585,350	80.72	163	3	139,815
test/Syn	4,320,541	3,840,392	88.89	346	16	480,149
<i>Total</i>	<i>70,427,637</i>	<i>59,936,580</i>	<i>85.10</i>	<i>92,443</i>	<i>161</i>	<i>10,491,057</i>

## 5 EXPERIMENTS

In this section we present the experimental evaluation of our methodology by describing the output of the data preprocessing step and the accuracy results. In particular, for each DL model and variant we report the error measures on the considered datasets and the corresponding number of parameters obtained as a result of automatically adapting the number of neurons in the hidden layers.

We performed the experiments using Google Colab+, with GPU enabled and RAM set as “high”, using python with libraries Dask, Pandas, Keras, and Sci-Kit Learn.

### 5.1 Data Preprocessing

We apply the data preprocessing procedure described in Section 3.1 to the CSE-CIC-IDS-2018 and CIC-DDoS-2019 datasets. In particular, in the data clean-

ing step we removed 17 columns from CSE-CIC-IDS-2018:

- 7 *socket-related features*: Flow ID, Src IP, Src Port, Dst IP, Dst Port, Protocol, Timestamp;
- 10 *features containing only zeros*: Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, CWE Flag Count, Fwd Byts/b Avg, Fwd Pkts/b Avg, Fwd Blk Rate Avg, Bwd Byts/b Avg, Bwd Pkts/b Avg, Bwd Blk Rate Avg.

and removed 20 columns from CIC-DDoS-2019:

- 8 *socket-related features*: Flow ID, SourceIP, SourcePort, DestinationIP, DestinationPort, Protocol, Timestamp, SimillarHTTP;
- 12 *features containing only zeros*: Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, FIN Flag Count, PSH Flag Count, ECE Flag Count, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk and Bwd Avg Bulk Rate.

At the end of the preprocessing phase, we reduced the CIC-IDS-2018 dataset from 11 to 7 million rows and from 84 to 56 columns, while we reduced the CIC-DDoS-2019 dataset from 70 to 10 million rows and from 87 to 57 columns. Table 5 and Table 6 outline the results of the preprocessing phase on the CIC-IDS-2018 and CIC-DDoS-2019 datasets, respectively.

## 5.2 Results

First, we train and evaluate the models on the CSE-CIC-IDS-2018 dataset with a fewer number (7) of labels, then on the CIC-DDoS-2019 dataset with more (18) labels. We evaluate the performance of the proposed models in terms of accuracy, precision, recall, and F1-score, considering 70% of data for training and 30% for testing.

Table 7 shows the accuracy results of the proposed methodology, for the two datasets considered, the two classification variants (binary and multi-label), and the three models (DNN, CNN, RNN). From the table, it is possible to observe that, in the case of binary classification, all the models exhibit a high performance on the considered datasets ( $> 99\%$  in all considered metrics). In particular, the DNN model achieves the best performance, with 99.80% and 99.95% accuracy on the CSE-CIC-IDS-2018 and CIC-DDoS-2019 datasets, respectively. However, when using the CIC-DDoS-2019 dataset the models exhibit an increased number of parameters. This is caused by the algorithm for hyperparameter tuning that increases the number of neurons in hidden layers to cope with the increased complexity of the CIC-DDoS-2019 dataset, which has more classes with respect to CSE-CIC-IDS-2018. Figure 2 and Figure 3 present the confusion matrices for the proposed models using binary classification on the CSE-CIC-IDS-2018 and CIC-DDoS-2019 datasets, respectively. The figures show the high accuracy obtained when considering binary classification.

In the case of multi-label classification, the models evaluated on the CSE-CIC-IDS-2018 dataset perform similarly to the binary classification case, at the cost of significantly increasing the number of parameters. However, when evaluating the multi-label classification models on the CIC-DDoS-2019 dataset, we can observe a decrease in performance, with  $< 80\%$  in all considered metrics. The result is caused by the combined complexity of a multi-label classification with a more complex dataset with respect to CSE-CIC-IDS-2018. However, it is worth noting that the CIC-DDoS-2019 dataset exhibits characteristics that limit the classification accuracy obtainable on the

dataset. For example, the dataset contains classes with few representative samples and attacks which have a strong semantic similarity but are classified with separate labels (Ferrag et al., 2021; Chartuni and Márquez, 2021).

## 6 CONCLUSIONS

In this paper we proposed a simulation of anomaly-based IDS using Deep Learning (DL) techniques, by considering three different models (DNN, CNN, RNN) trained and tested on two datasets with an increasing level of complexity (CSE-CIC-IDS-2018 and CIC-DDoS-2019) and considering two classification variants (binary and multi-label). To evaluate the capability of the considered architectures to adapt to different conditions, we considered adaptive architectures with a variable number of neurons in the hidden layers. Then, we automatically tuned the number of neurons in each hidden layer based on the considered dataset and the type of classification.

The results shows how, in the case of binary classification, it was possible to obtain consistently high performance ( $> 99\%$ ) in both datasets, at the cost of an increased number of parameters when considering the more complex dataset. Overall, in our experiments the DNN model achieved the best performance. However, in the case of multi-label classification, we obtained satisfactory results only on the simplest dataset, highlighting the need for more complex architectures when considering complex datasets.

Future works will consider hyperparameter tuning algorithms with an increased search space and methods based on neural architecture search to design more complex adaptive architectures.

## Acknowledgements

This work was supported in part by the EC under projects EdgeAI (101097300), GLACIATION (101070141), and MARSAL (101017171), and by the Italian MUR under project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. We also thank the NVIDIA Corporation for the GPU donated. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor Italian MUR can be held responsible for them.

Table 7: Accuracy results of the proposed anomaly-based IDS using DL.

Dataset	Class. variant	Model	N. param.	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
CSE-CIC-IDS-2018	Binary	DNN	2,977	99.80	99.80	99.80	99.80
		CNN	8,065	99.33	99.33	99.34	99.33
		RNN	12,705	99.78	99.78	99.78	99.78
	Multi-label	DNN	11,608	99.79	99.79	99.79	99.78
		CNN	12,616	99.78	99.78	99.78	99.77
		RNN	25,832	98.59	98.59	98.57	98.53
CIC-DDoS-2019	Binary	DNN	5,185	99.95	99.95	99.95	99.95
		CNN	9,537	99.88	99.88	99.89	99.88
		RNN	12,705	99.80	99.80	99.81	99.80
	Multi-label	DNN	17,183	77.41	77.41	77.52	71.78
		CNN	19,135	77.29	77.29	77.56	71.65
		RNN	71,687	73.52	73.52	64.52	66.55

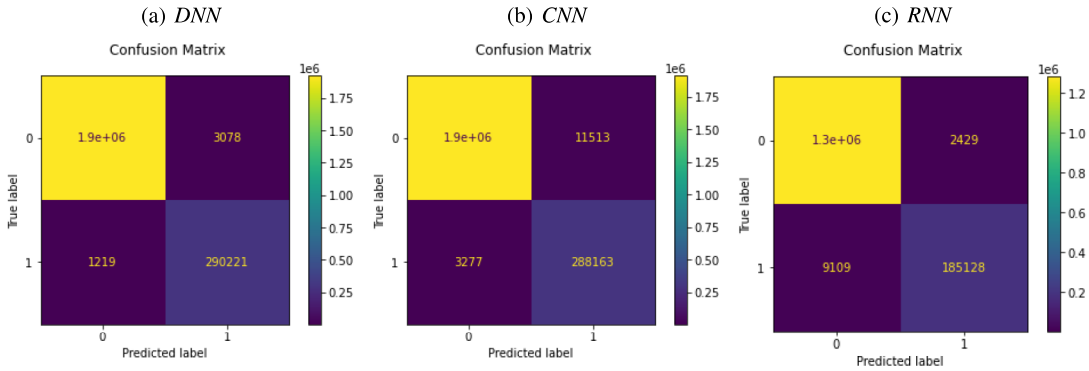


Figure 2: Confusion matrices for the proposed DNN, CNN, and RNN models, using binary classification on the CSE-CIC-IDS-2018 dataset.

## REFERENCES

- Alasadi, S. A. and Bhaya, W. S. (2017). Review of data pre-processing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12(16):4102–4107.
- Amaizu, G. C., Nwakanma, C. I., Bhardwaj, S., Lee, J., and Kim, D.-S. (2021). Composite and efficient DDoS attack detection framework for B5G networks. *Computer Networks*, 188:107871.
- Asad, M., Asim, M., Javed, T., Beg, M. O., Mujtaba, H., and Abbas, S. (2020). Deepdetect: detection of distributed denial of service attacks using deep learning. *The Computer Journal*, 63(7):983–994.
- Assis, M. V., Carvalho, L. F., Lloret, J., and Proença Jr., M. L. (2021). A GRU deep learning system against attacks in software defined networks. *Journal of Network and Computer Applications*, 177:102942.
- Bhardwaj, A., Mangat, V., and Vig, R. (2020). Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of DDoS attacks in cloud. *IEEE Access*, 8:181916–181929.
- Chartuni, A. and Márquez, J. (2021). Multi-classifier of DDoS attacks in computer networks built on neural networks. *Applied Sciences*, 11(22):10609.
- CIC (2018). Cse-cic-ids-2018. <https://www.unb.ca/cic/datasets/ids-2018.html>.
- CIC (2019). Cic-ddos-2019. <https://www.unb.ca/cic/datasets/ddos-2019.html>.
- Cil, A. E., Yildiz, K., and Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. *Expert Systems with Applications*, 169:114520.
- de Assis, M. V., Carvalho, L. F., Rodrigues, J. J., Lloret, J., and Proença Jr, M. L. (2020). Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. *Computers & Electrical Engineering*, 86:106738.
- Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martinez-del Rincon, J., and Siracusa, D. (2020). LUCID: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE Trans. on Network and Service Management*, 17(2):876–889.



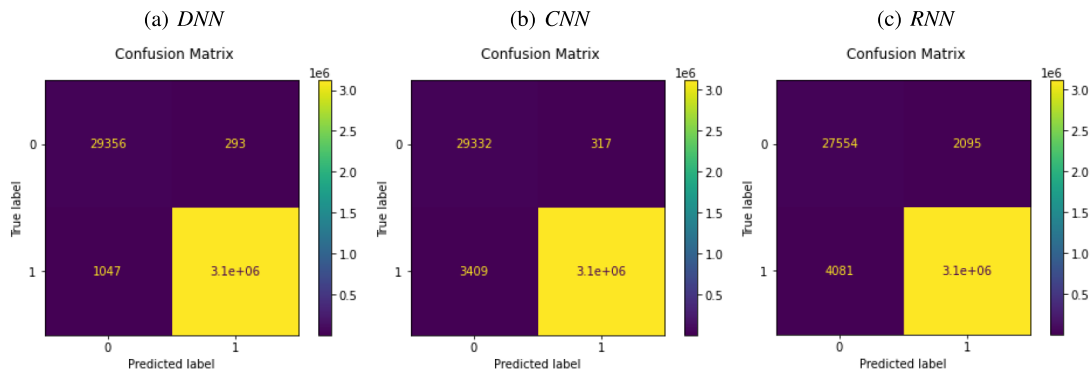


Figure 3: Confusion matrices for the proposed DNN, CNN, and RNN models, using binary classification on the CIC-DDoS-2019 dataset.

- Elsayed, M. S., Le-Khac, N.-A., Dev, S., and Jurcut, A. D. (2020). DDoSNet: A deep-learning model for detecting network attacks. In *Proc. of the 2020 IEEE 21st Int. Symp. on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 391–396.
- Ferrag, M. A., Shu, L., Djallel, H., and Choo, K.-K. R. (2021). Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. *Electronics*, 10(11).
- Gümüşbaşı, D., Yıldırım, T., Genovese, A., and Scotti, F. (2021). A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. *IEEE Systems Journal*, 15(2):1717–1731.
- Haider, S., Akhunzada, A., Mustafa, I., Patel, T. B., Fernandez, A., Choo, K.-K. R., and Iqbal, J. (2020). A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks. *IEEE Access*, 8:53972–53983.
- Kasim, Ö. (2020). An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks. *Computer Networks*, 180:107390.
- Kim, J., Kim, J., Kim, H., Shim, M., and Choi, E. (2020). CNN-based network intrusion detection against denial-of-service attacks. *Electronics*, 9(6):916.
- Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., Ghorbani, A. A., et al. (2017). Characterization of tor traffic using time based features. In *Proc. of the 3rd Int. Conf. on Information Systems Security and Privacy (ICISSP)*, pages 253–262.
- Liang, X. and Znati, T. (2019). A long short-term memory enabled framework for DDoS detection. In *Proc. of the 2019 IEEE Global Communications Conf. (GLOBECOM)*, pages 1–6.
- Minka, T. (2000). Automatic choice of dimensionality for PCA. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13.
- Mittal, M., Kumar, K., and Behal, S. (2022). Deep learning approaches for detecting DDoS attacks: A systematic review. *Soft Computing*, pages 1–37.
- Roopak, M., Tian, G. Y., and Chambers, J. (2020). An intrusion detection system against DDoS attacks in IoT networks. In *Proc. of the 2020 10th Annual Computing and Communication Workshop and Conf. (CCWC)*, pages 0562–0567.
- Roopak, M., Yun Tian, G., and Chambers, J. (2019). Deep learning models for cyber security in IoT networks. In *Proc. of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conf. (CCWC)*, pages 0452–0457.
- Sabeel, U., Heydari, S. S., Mohanka, H., Bendhaou, Y., Elgazzar, K., and El-Khatib, K. (2019). Evaluation of deep learning in detecting unknown network attacks. In *Proc. of the 2019 Int. Conf. on Smart Applications, Communications and Networking (SmartNets)*, pages 1–6.
- Sbai, O. and El boukhari, M. (2020). Data flooding intrusion detection system for MANETs using deep learning approach. In *Proc. of the 13th Int. Conf. on Intelligent Systems: Theories and Applications (SITA)*. Association for Computing Machinery.
- Srikanth Yadav, M. and Kalpana, R. (2019). Data preprocessing for intrusion detection system using encoding and normalization approaches. In *Proc. of the 2019 11th Int. Conf. on Advanced Computing (ICoAC)*, pages 265–269.
- Wang, L. and Liu, Y. (2020). A DDoS attack detection method based on information entropy and deep learning in SDN. In *Proc. of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conf. (ITNEC)*, pages 1084–1088.