DDoS Attacks Detection and Classification based on Deep Learning Model

W. Ferhi (wafaa.ferhi@univ-tlemcen.dz)
Tlemcen University

D. Moussaoui
Tlemcen University

M. Hadjila
Tlemcen University

A. Bouidaine
Tlemcen University

D. Bensenouci
Tlemcen University

Research Article

Keywords:

Posted Date: June 19th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3013492/v1

Additional Declarations: No competing interests reported.

DDoS Attacks Detection and Classification based on Deep Learning Model

W. Ferhi[a,1], **D. Moussaoui**[b,1], **M. Hadjila**[c,1], **A. Bouidaine**[d,1], **D. Bensenouci**[e,2]

[1]STIC Lab, Dept. of Telecom, Faculty of Technology, Tlemcen University, Post Box 230 Chetouane, Tlemcen, Algeria
[2]Dept. of Telecom, Faculty of Technology, Tlemcen University, Post Box 230 Chetouane, Tlemcen, Algeria

Received: date / Accepted: date

Abstract A Distributed Denial of Service (DDoS) attack is a massive, distributed, deliberated, and coordinated attack by multiple compromised machines to overwhelm an online service or a server. Attackers attempt to attack the availability of the service by sending voluminous dummy data to make target machine fall short of resources. Therefore, this paper proposed a deep learning model to detect and classify DDoS attacks. This work incorporates four classification techniques: binary classification, multiclass classification with label encoding, multiclass classification with one-hot encoding, and multi-label classification. Experiments were carried on CSE-CIC-IDS2018-DDoS dataset which be produced from CSE-CIC-IDS2018 dataset. Performing preprocessing data is useful to remove duplicate observations, erase missing values, convert categorical data to numerical data, and perform features scaling so that all values are within the same range. L2 regularization technique is used to achieve good generalization performance in the test dataset. According to the test results, the suggested model achieved accuracy values of 100% for binary classification, 99.87% for multi-label classification, 99.84% for multi-class classification (one-hot encoder), and 97% for multi-class classification (label encoder).

1 Introduction

The rapid technological development and digitization have brought numerous advances and conveniences to society. Different aspects of our lives have been transformed by innovations like edge computing, blockchain, robotics, and the Internet of Things (IoT), which have also increased automation and productivity [1]. However, alongside their benefits, these technologies also introduce various vulnerabilities and challenges to the digital environment. With the increasing reliance on web connectivity for businesses, the delivery of services, and critical operations, the threat landscape has expanded exponentially. Attackers constantly exploit these vulnerabilities to engage in malicious activities, posing significant risks to individuals and organizations [2].

Among the wide range of cyberattacks, DDoS (distributed denial of service) attacks have become one of the most frequent and persistent threats since their inception in 1974 [3]. DDoS attackers create a network of compromised devices, known as a botnet, where each compromised device, or bot, contributes to the flood of traffic directed at the target, making it unavailable to legitimate users. DDoS attacks can have a big effect on businesses and organizations. They can disrupt operations, lead to financial losses, and damage reputations. In some cases, DDoS attacks have even been used to cause physical damage [1].

In 2018, Google suffered a massive DDoS attack, which disrupted its services for several hours. The hackers used a technique called "Memcached Amplification," which involves using misconfigured Memcached servers to return very large responses to simple requests, overwhelming the target server. This attack was considered the largest DDoS attack on record at the time [4]. In February 2021, Amazon also suffered a major DDoS attack, which disrupted the company's services for several hours. The hackers used a technique called "Connection Flood," which involves sending a large number of connection requests to a server at the same time, overwhelming it and making it inaccessible. Amazon reacted quickly by blocking the attacker's IP address, but the incident still had a negative impact on the company's services [5].

Traditional rule-based detection techniques have been widely

[a]wafaa.ferhi@univ-tlemcen.dz
[b]djilali.moussaoui@univ-tlemcen.dz
[c]mourad.hadjila@univ-tlemcen.dz
[d]albaraa.bouidaine@univ-tlemcen.dz
[e]dounya.bensenouci@univ-tlemcen.dz

used to detect DDoS attacks. However, these methods have shown their limitations due to their lack of adaptability to new attack techniques and their high sensitivity to false positives. In addition, attackers have developed sophisticated techniques to disguise their attacks, which makes detection even more difficult. In consequence of the failure of these methods, the search for new and more intelligent detection techniques has led to the development of deep learning. DL algorithms play a crucial role in combating the ever-evolving DDoS attacks. Traditional defenses face challenges in effectively detecting and reducing these attacks due to the rapid evolution of attack techniques and the scale of attacks facilitated by botnets. By harnessing the power of artificial neural networks, deep learning algorithms analyze vast amounts of data to identify subtle and complex attack patterns. In addition, they are able to continuously adapt to new attack techniques, improving the effectiveness of the defense over time. This adaptive and robust DL approach provides a more effective and proactive defense against sophisticated DDoS attacks. The purpose of this paper is to propose different approaches to deep learning-based detection systems designed specifically for DDoS attacks. To achieve this, we use the well-known CSE-CIC-IDS2018 dataset to create a dataset specifically for DDoS attacks. The main goal is to advance a robust detection system based on DL that can accurately identify and mitigate various types of DDoS attacks.

In this work, we will evaluate the validity of DL algorithms by comparing the performance of the proposed approaches. We will assess the specialized dataset‘s capability in training and evaluating the DL models for DDoS attack detection. By leveraging the power of DL techniques, we aim to develop an adaptive detection system that can effectively detect and mitigate sophisticated DDoS attacks. The main DL algorithm we use is a deep neural network with fully connected layers. This architecture allows flexible inspection of different hyperparameters and optimizes the performance of the model for DDoS attack detection. During the training phase, the deep neural network will learn from the labeled dataset, adapting its internal parameters through an iterative optimization process. We will carefully tune the hyperparameters of the DL model to enhance its performance in accurately detecting and classifying different types of DDoS attacks. To evaluate the effectiveness of the proposed approaches, we will employ rigorous evaluation metrics such as accuracy, precision, recall, and F1-score.

The rest of the paper is organized as follows: related work will be presented in section 2. Section 3 describes the proposed solution based on deep learning to detect DDoS attacks based on CSE-CIC-IDS2018-DDoS dataset which be produced from CSE-CIC-IDS2018 dataset. Lastly, Section 4 concludes this paper and presents suggesting future research opportunities in this field.

2 Related Work

In recent studies on intrusion detection, machine learning and deep learning based methods have been widely used. More effective intrusion detection systems are being developed in this area with the development of machine learning and deep learning algorithms, which are emerging with access to big data. In this section of the literature, we have presented some of the recent studies on DDoS attacks.

Feature selection techniques and their performance on datasets recently used in the literature were reviewed by Di Mauro et al. (2021) [5]. In this study, brief information is given on a large number of different techniques. The dataset of DDoS attacks and the selection results of different algorithms were studied using correlation maps. On the same dataset, feature selection time and training time were compared. As a result of feature extraction from the DDoS dataset, performance evaluation is presented for different types of attacks on different datasets. In this section, there will be a review of IDS system designs in the literature. It is found that many different machine learning methods are used to detect different types of attacks when examining the studies in the literature. When analyzing the performance of these methods, it was found that they have advantages and disadvantages compared to each other. In addition to designing systems using a single algorithm, it has been found that hybrid approaches are often preferred when designing. In the pre-processing stage, feature selection techniques were presented as a component of the proposed systems to reduce the workload and increase the performance of the proposed systems. A large number of algorithms were found to exist as feature extraction methods, and testing was performed to identify the method suitable for the structure of the data, and consequently the method used.

Kamalov et al. (2021) [6] applied a fusion machine learning method to develop a new IDS model. To determine the related features in the dataset, authors used the orthogonal variance decomposition technique. The given attributes are used to build a deep neural network for intrusion detection. The proposed technique achieves a detection accuracy of 100% in the accurate detection of DDoS attacks.

Wei et al. (2021) [7] proposed a hybrid model that incorporates two deep learning-based algorithms for successful detection and classification of DDoS attacks. By automatically identifying the most important feature sets, the Autoencoder component of the proposed model performs successful feature extraction. To address the performance overhead for multiple DDoS attack categories, the Multilayer Perceptron Network part of the proposed model uses compressed and reduced feature sets. Test results of the proposed method have shown that it achieves high accuracy and F1 scores above 98%.

Odumuyiwa et al. used unsupervised machine learning techniques to categorise incoming network packets at the transport layer to identify DDoS attacks in IoT networks. This study independently developed two deep learning algorithms and two clustering algorithms for mitigating DDoS attacks. Exploitation-based DDoS attacks were highlighted. These include Transmission Control Protocol SYN flood attacks and UDP lag attacks. During the experimental phase, the algorithms were trained using the Mirai, BASHLITE and CIC-DDoS2019 datasets. Results showed that auto-encoder performed best overall, with highest accuracy across all datasets [8].

Cil et al. proposed a model based on deep neural networks for the detection of DDoS attacks based on a sample of packets collected from the network traffic. The DNN model can successfully perform on the CIC-DDoS2019 dataset because it incorporates feature extraction and classification algorithms in its structure and layers that update themselves as they are trained [9].

A comprehensive and effective DDoS attack detection system for 5G and B5G was created by Amaizu et al. To identify and return the type of DDoS attack, the proposed detection system combines a composite multilayer perceptron with an effective feature extraction technique. The simulations and dataset testing showed that the proposed framework could detect DDoS attacks with a high accuracy of 99.66% and a loss of 0.011, according to the results. The outcomes of the suggested detection framework were also contrasted with those of different researchers [10].

To address the shortcomings and introduce a new taxonomy for DDoS attacks, including a new categorisation based on flow networks, Khempetch et al. presented the CIC-DDoS2019 dataset. The DNN and LSTM architecture is used for the identification of DDoS attacks [11].

Using the ResNet18 deep learning architecture, Hussain et al. proposed a method to identify DoS and DDoS attacks in IoT systems. They converted the network traffic data features into representations of images and trained ResNet18 on the resulting dataset. They trained the deep learning model on the classification of 11 types of attacks and benign traffic [12].

Convolution and LSTM-based models for detecting DDoS attacks as a function of traffic fluctuations in IoT networks were presented by Jia, et al. They merged the CIC-DDoS2019 dataset with the dataset they produced using the DDoS simulators BoNeSi and SlowHTTPPTest. They examined the appropriateness for IoT networks and assessed the model's performances [13].

Shurman et al. proposed hybrid and deep learning-based methods to detect DoS and DrDoS attacks in IoT networks by integrating the signature-based and anomaly-based methods. They trained and tested their models using the CIC-DDoS2019 dataset, and all of their implemented models contain LSTM-based networks [14].

Li et al. suggest a three-part strategy to defend against DDoS assaults on Internet of Things (IoT) devices. The first is an increase in calculation speed for entropy, the second is early detection, and the third is optimization of a detection outcome. The 1999 DARPA Intrusion Detection Evaluation Dataset, the DARPA DDoS Dataset, and the UNB CIC DDoS 2019 Evaluation Dataset were used in their experimental experiments. The suggested solution can be incorporated into real-time IoT defensive systems because to its low latency and strong performance [15].

The current DDoS datasets were reviewed and their shortcomings were discussed by Sharafaldin et al. They proposed a new dataset for DDoS attacks to evaluate IDS and IPS methods and systems, defining a new testbed by designing and implementing an attack network and a victim network. Their dataset, called CIC-DDoS2019, offers improvements over the limitations of current datasets [16].

Using the CICDDoS 2019 dataset, Javeed et al. presented an SDN-enabled architecture that utilises hybrid deep learning detection algorithms to detect cyber threats and attacks while dealing with resource-constrained IoT devices [17].

Alamri and Thayananthan have developed a scheme to detect DDoS attacks against SDN. Threshold violations are detected through bandwidth monitoring, and adaptive monitoring triggers the Extreme Gradient Boosting (XGBoost) algorithm. In this way, it is possible to determine whether the traffic is normal or malicious. They carried out performance analyses using the CIC-DDoS2019, the NSL-KDD and the CAIDA data sets [18].

To prevent DDoS attacks, De Assis et al. proposed a defence system for Software Defined Network (SDN), which is commonly used in Internet of Things (IoT). They evaluated machine learning architectures such as MLP, CNN, D-MLP and LR by classifying attacks as DDoS and normal. Simulated IP flows were collected from the CIC-DDoS2019 dataset in addition to generating SDN traffic for training and testing [19].

3 Methodology

The objective of this paper is to present the realization of deep neural network (DNN) models for the detection of DDoS attacks using different deep learning approaches and techniques. We also evaluated the performance of these models in terms of accuracy, precision, recall and f1-score. By leveraging deep learning techniques, this work aims to improve the robustness of the network infrastructure against malicious activities. To achieve this goal, we used the CSE-CIC-IDS2018 dataset, which is a comprehensive and widely used dataset in the field of network security.

This section presents the network architecture we use in our DDoS attack detection and classification experiment.

3.1 Dataset

The chosen dataset is the CSE-CIC-IDS2018 [20]. This dataset was created as part of a collaborative project between the Communications Security Establishment (CSE) and the Canadian Cybersecurity Institute (CIC), which uses profiles to generate cybersecurity datasets in a systematic. It provides a detailed description of intrusions, as well as abstract distribution models for lower-level applications, protocols, and network entities. This dataset contains ten CSV files representing ten days of captured network flow, with over 16.2 million samples. Additionally, more than 80 features were extracted by the CICFlowMeter tool. This dataset includes six main types of intrusion attacks: distributed denial of service (DDoS), denial of service (DoS), Botnet, Brute Force, infiltration, and web attacks.

Two of the ten files contain DDoS attacks, file number four and file number five. The fourth file is 4GB in size and contains two types of traffic: benign flow (normal traffic) and a DDoS attack. Normal (benign) traffic represents 93% of the file while the remaining 7% constitutes the DDoS-LOIC-HTTP attack. The fifth file in turn is composed of normal traffic with a rate of 34% as well as two kinds of DDoS attacks, which are DDoS-HOIC and DDoS-LOIC-UDP attacks with a percentage of 65% and 1% respectively.

However, we are not going to use all Benign traffic (7372557 instances) from the fourth file. We removed 60% of the normal flow (Benign) in order to avoid a huge imbalance between attack traffic and normal traffic, which would make it more difficult for the model to converge in the training phase. Then we merged these two files into a single CSV file called “data.csv” which contains 4573789 instances. The dataset thus obtained is named CSE-CIC-IDS2018-DDoS.

3.2 Data pre-processing

When working with a dataset, it is essential to ensure the quality and organization as well as the clarity of the data before using it to train a deep learning model. Deep neural networks usually require massive amounts of data for their training, which can lead to problems: useless, redundant, erroneous or missing data NAN (Not A Number), infinity and other problems.

Data preprocessing is a crucial step to ensure that the input data to the deep learning model is clean, consistent, and relevant to the classification or prediction task at hand. It makes it possible to improve the quality of the data and to make them more coherent and to adapt them to the needs of the model to be built. Data preprocessing can also help reduce model training and testing times. By ensuring that our data is clean and ready to be used to train our model, we ensure that its performance is not compromised by missing, erroneous or redundant data. Data preprocessing is therefore an important step to ensure the reliability and efficiency of our deep learning model.

Data preprocessing steps for deep learning can include various techniques such as data normalization, removing infinity data (NaN or Inf), handling missing data, removing duplicate rows, removing unnecessary data and feature coding and so on.

3.2.1 Data cleaning

Data cleaning provides a clean and reliable dataset to perform more accurate analyzes and modeling. After analyzing the dataset, we found a percentage of 1.1% of missing values. We have cleaned all the data by removing unnecessary features ("Timestamp", "Flow ID", "Src IP", "Src Port" and "Dst IP") because their properties are related to contact information and do not represent important characteristics of attacks. Then, we removed the values of infinity, NAN and duplicate rows.

3.2.2 Data encoding

In some cases, categorical variables can be difficult to use in statistical analyzes or machine learning algorithms. For this reason, categorical variables must be transformed into an appropriate numerical form. For this, we used the method str. replace() from Pandas. This method allowed us to replace the different categorical values with integers. We then used the astype() method to convert the data type to 64-bit integers. This step was necessary so that the column could be used as an input variable in certain Machine Learning algorithms. Using these two methods, we were able to effectively transform the "Label" column into a suitable numeric variable for our analysis [21].

3.2.3 Normalization vs Standardization

Normalization puts all the quantitative variables on the same scale so that they fall within an interval between 0 and 1, which greatly facilitates machine learning [22].

Standardization is a scaling method that transforms data into a Gaussian distribution with mean equal to 0 and standard deviation equal to 1 [22].

In order to improve the performance and reliability of our DNN model and to help it converge quickly, we applied the standardization method to the training and test data with the StandardScaler function of the sklearn.preprocessing module. This method is an important step in the preprocessing of data because it makes the data comparable, which facilitates comparison between them.

3.2.4 Spliting data

In Deep Learning, model performance should never be evaluated on the same data that was used for training. When training a deep learning model, it is important to create two datasets: a training dataset and a test dataset. The training dataset is used to train the model, while the test dataset is used to evaluate the performance of the model. In general, we put 80% of the data in the training part and 20% for the test part (see Figure 1). To do all this in python, we use the train_test_split() function which belongs to the model_selection module.

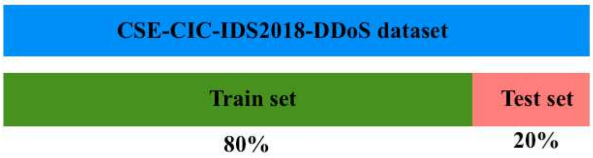


Fig. 1 Splitting the CIC-IDS2018-DDoS dataset into two parts.

3.3 Evaluation metrics

The following metrics are used to validate our created models [23].

1. **Accuracy:** Accuracy is a measure of the accuracy of a model's positive predictions. It is defined as the ratio between the true predictions and the total number of predictions made by the model.
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$
2. **Precision:** The proportion of correct predictions among the positive predictions is called precision, or positive predictive value (PPV):
$$Precision = \frac{TP}{TP+FP} \tag{2}$$
3. **Recall:** We call recall, or sensitivity, the rate of true positives, i.e. the proportion of positive examples correctly identified as such:
$$Recall = \frac{TP}{TP+FN} \tag{3}$$
4. **F-score:** We call F-score or F1-score the harmonic mean of precision and recall:
$$F-score = 2\frac{Precision \times Recall}{Precision+Recall} = \frac{2TP}{2TP+FP+FN} \tag{4}$$
5. **Specificity:** We call specificity the rate of true negatives, in other words the proportion of negative examples correctly identified as such:
$$Specificity = \frac{TN}{FP+TN} \tag{5}$$

3.4 Model creation

In order to detect DDoS attacks, we used different classification methods such as binary classification, multi-class classification with label encoding and one-hot encoding, and multi-label classification. For this, we have built a model of multilayer neural network (Multilayer Perceptron [24]) with four hidden layers of Dense type, with the first layer of size 128, the second layer of size 64, and the third and fourth layers both of size 32. The hidden layers use the ReLU activation function [25], while the output layer uses the sigmoid activation function for binary classification/multi-label classification and softmax activation function for multi-class classification. The output layer depends on the approach used. To avoid overfitting, the model was regularized using L2 regularization [26], with a lambda coefficient equal to 0.0001 applied to the node. The model is compiled with the Adamax optimizer. The choice of the loss function depends on the approach used, either categorical_crossentropy for multi-class classification or binary_crossentropy for binary classification and multi-label classification. The evaluation of the model is done using the "accuracy" metric. The model is trained with a batch size of 128 and over a number of epochs set at 30. The data is divided into 90% for training and 10% for validation. Finally, the model is evaluated on the test set.
The proposed DDoS detection system is illustrated in Figure 2.

3.5 Multi-class classification

In this approach, we used two methods to encode the "Label" column which constitutes our target.

3.5.1 Multi-class classification - Label encoding

The "label encoding" is a simple and fast method which consists in transforming categorical variables into numerical variables [27]. This method replaces each category with a unique integer. The objective of this approach is to facilitate the training of the model.
In this case, we will apply this method to the target variable of the dataset called "Label" which contains four different classes: Benign, DDoS attacks-LOIC-HTTP, DDoS attack-HOIC and DDoS attack-LOIC-UDP. With the method of "Label Encoding", the categories Benign, DDoS attacks-LOIC-HTTP, DDoS attack-HOIC and DDoS attack-LOIC-UDP will be replaced respectively by the integers 0, 1, 2 and 3 (see Figure 3) .
Figures 4 and 5 show the "Loss" and the "Accuracy" of the model as a function of iterations during the learning process. This approach yielded an accuracy of 97.18% for test data and 97.24% for validation data. The model trained with the "Label Encoding" approach has a higher "Loss" of 9.04%

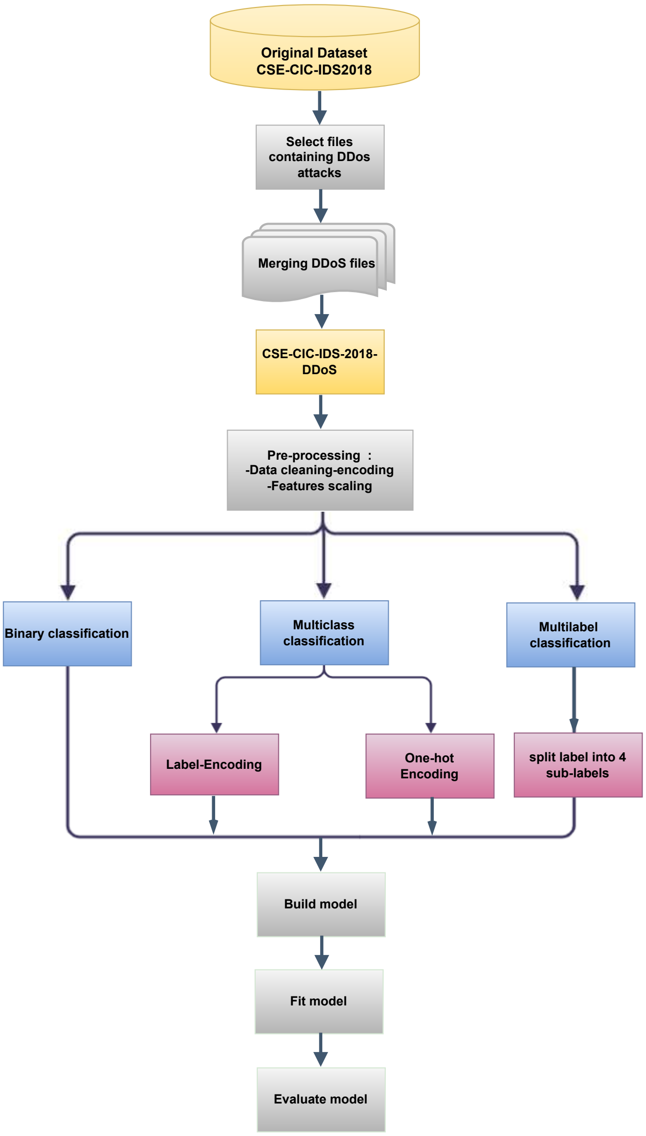


Fig. 2 Component architecture of the proposed work.

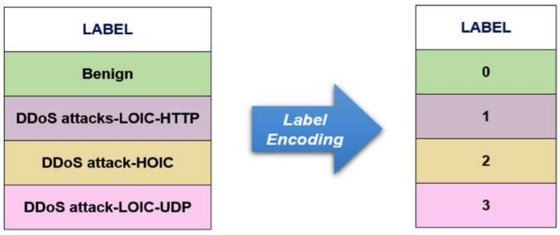


Fig. 3 Label Encoding.

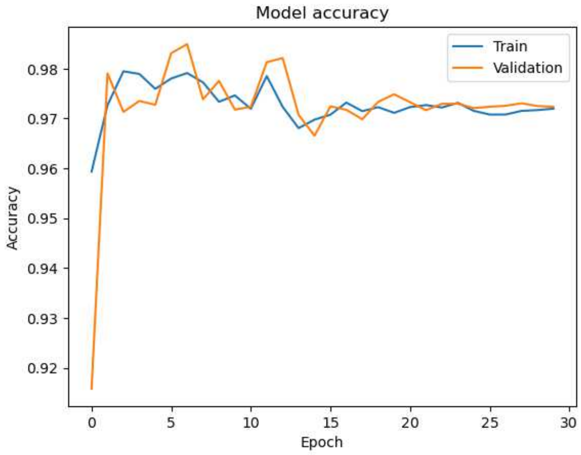


Fig. 4 Training and Validation Accuracy-Multi-class (label encoding).

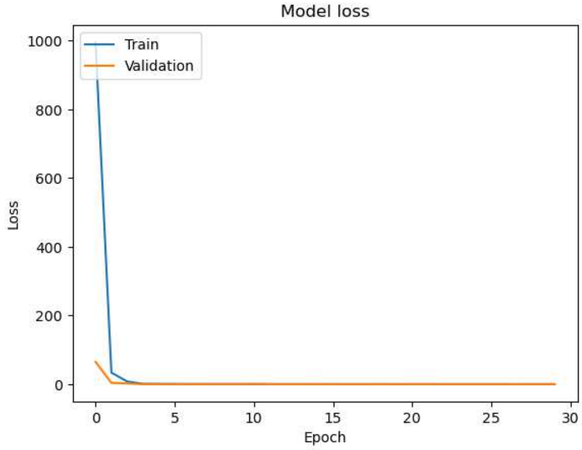


Fig. 5 Training and Validation Loss-Multi-class (label encoding).

for the test data and 8.88% for the validation data.
The confusion matrix in the case of a multi-class classification with "Label Encoding" is given by Figure 6.

3.5.2 Multi-class classification - one-hot encoding

In this approach, we will use another method called One-Hot Encoding [28]. This method consists of transforming each class of the "Label" target variable into a unique binary vector. Each vector will have a length equal to the number of different classes in the target variable, and will have a single value 1 which corresponds to the class of the target variable for this observation and the value 0 for the others. For example, if the observation belongs to the class Benign, the corresponding binary vector would be [1, 0, 0, 0] (see Figure 7).
Figures 8 and 9 shows the "Loss" and the "Accuracy" of the model according to the iterations during the training process. This approach produced very satisfactory results with an "Accuracy" of 99.87% for the test data and 99.88% for the

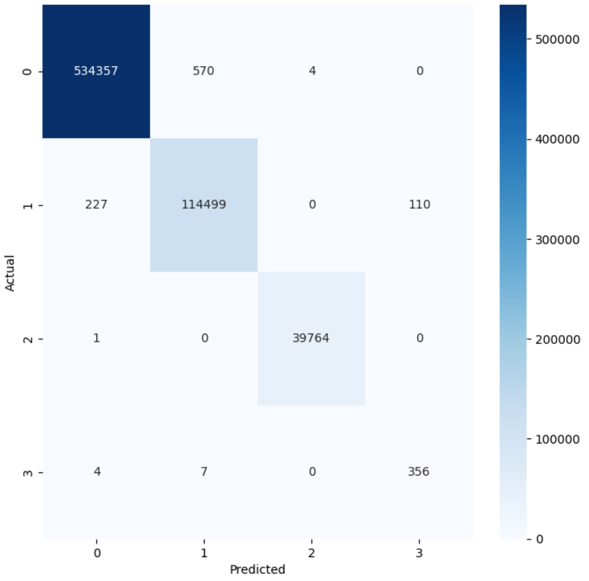


Fig. 6 Confusion matrix for multi-class (label encoding).

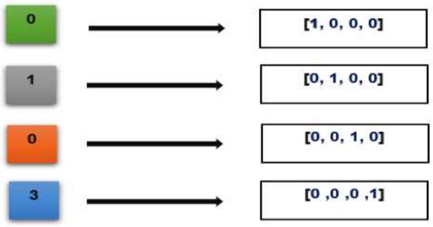


Fig. 7 One-hot Encoding.

Table 1 Evaluation Metrics for multi-class classification (one-hot encoding).

Label	Accuracy	Precision	Recall	F1score
Benign	99.90%	100%	100%	100%
DDoS LOIC-HTTP	99.50%	99%	100%	100%
DDOS HOIC	99.99%	100%	100%	100%
DDoS LOIC-UDP	98.28%	74%	98%	85%

validation data. The results of this approach also show a reliable "Loss" value for the test and validation data, with a loss of 0.45% for the test data and 0.44% for the validation data.

Table 1 displays the "Accuracy", "Precision", "Recall" and "F1-score" metrics for the 4 classes Benign, DDoS attacks-LOIC-HTTP, DDoS attacks-HOIC and DDoS attacks-LOIC-UDP in the case of multi-class classification with "one-hot encoding". The corresponding confusion matrix is given by Figure 10.

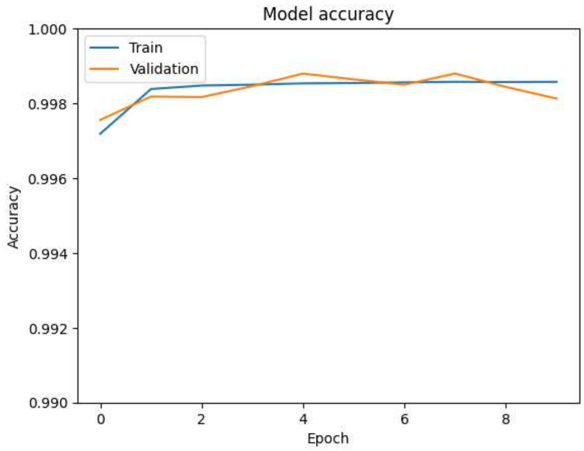


Fig. 8 Training and Validation Accuracy-Multi-class (one-hot encoding).

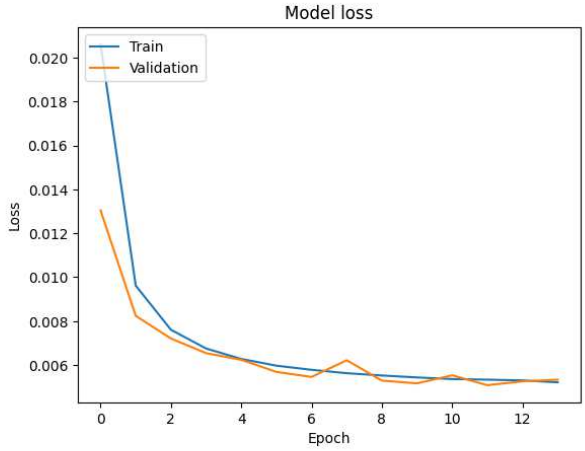


Fig. 9 Training and Validation Loss-Multi-class (one-hot encoding).

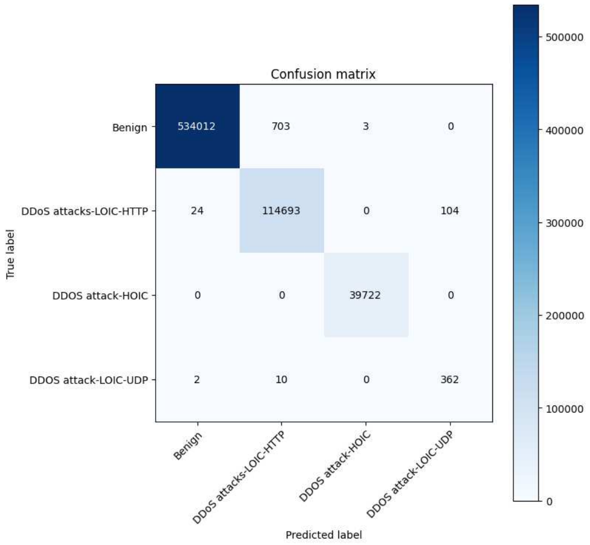


Fig. 10 Confusion matrix for multi-class (one-hot encoding).

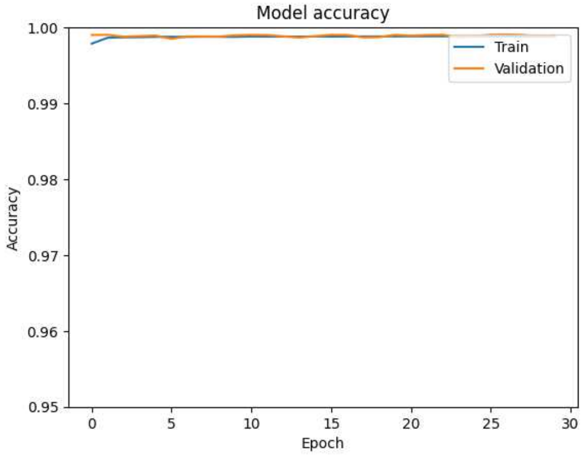


Fig. 11 Training and Validation Accuracy-binary classification.

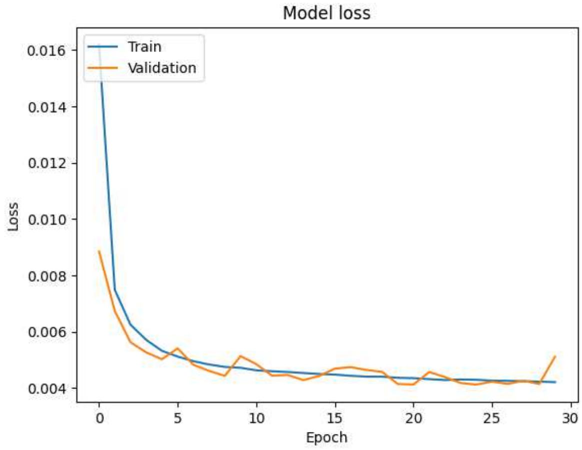


Fig. 12 Training and Validation Loss-binary classification.

3.6 Binary classification

In this approach, we have grouped the three types of DDoS attacks i.e. DDoS attacks-LOIC-HTTP, DDoS attack-HOIC and DDoS attack-LOIC-UDP into a single type named "DDoS", plus the flow "Benign". The goal of this approach is to simplify the target variable by reducing the number of classes, while maintaining an important distinction between DDoS and Benign observations.

Figures 11 and 12 show the "Loss" and the "Accuracy" of the model according to the iterations during the learning process. This approach also produced very satisfactory results with an accuracy of 99.88% for test data and 99.88% for validation data. This method also produced satisfactory results for loss, with a loss of 0.4% for test data and 0.41% for validation data.

Table 2 displays the "Accuracy", "Precision", "Recall" and "F1-score" metrics for the 2 Benign classes, DDoS in the case of binary classification. The corresponding confusion matrix is given by Figure 13.

Table 2 Evaluation Metrics for binary classification.

Label	Accuracy	Precision	Recall	F1score
Benign	100%	100%	100%	100%
DDoS	100%	100%	100%	100%

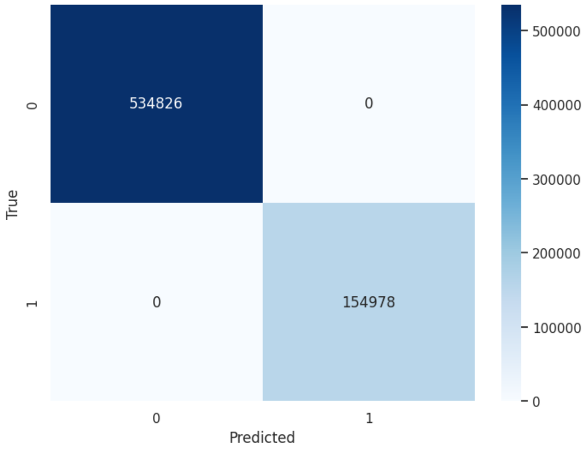
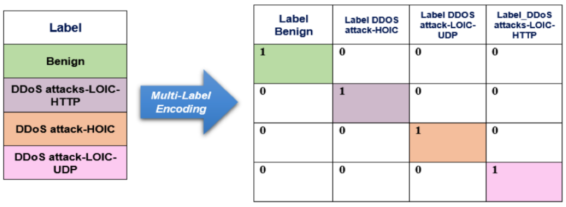


Fig. 13 Confusion matrix for binary classification.



Label Benign	Label DDOS attack-HOIC	Label DDOS attack-LOIC-UDP	Label_DDoS attacks-LOIC-HTTP
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Fig. 14 Multi-label encoding.

3.7 Multi-label classification

In this approach, we will use another encoding technique called multi-label encoding. This method consists of creating a new column for each category of the "Label" target variable, and assigning a value of 1 to the column corresponding to the category of the variable, and 0 to all the other columns. In this case, we will have four different target variables: Benign, DDoS-LOIC-HTTP, DDoS attack-HOIC and DDoS-HOIC-LOIC-UDP (see Figure 14). The objective of this approach is to allow the models to distinguish the different classes of the target variable in a finer way.

Figures 15 and 16 show "Loss" and "Accuracy" when training the model. Figures 15 and 16 show "Loss" and "Accuracy" when training the model.

The multi-label encoding approach produced perfect results with 100% accuracy for test and validation data. This method transformed the categorical variables into a multiple encoding, which makes it possible to better represent the complex relationships between the different categories.

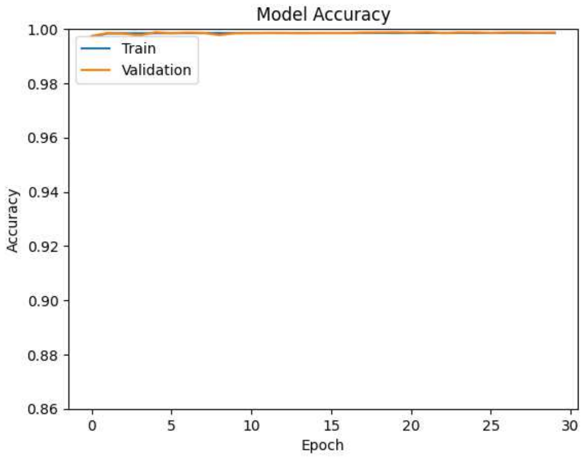


Fig. 15 Training and Validation Accuracy-multi-label classification.

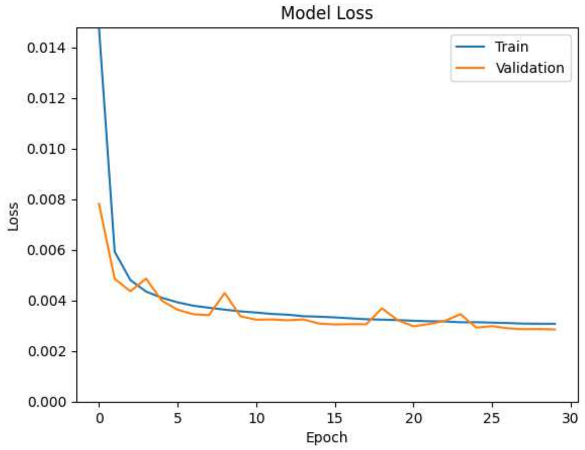


Fig. 16 Training and Validation Loss-multi-label classification.

Table 3 Evaluation Metrics for multi-label classification.

Label	Accuracy	Precision	Recall	F1score
0	99.89%	100%	100%	100%
1	99.70%	99%	100%	100%
2	99.99%	100%	100%	100%
3	97%	76%	97%	86%

For the multi-label encoding approach, the loss values of 8.15% for the test data and 11.8% for the validation data can be considered good because they are quite low, indicating that the model has learned to classify data well and that there is no overfitting.

Table 3 displays the "Accuracy", "precision", "recall" and "F1-score" metrics in the case of multi-label classification. The corresponding confusion matrix is given by Figure 17.

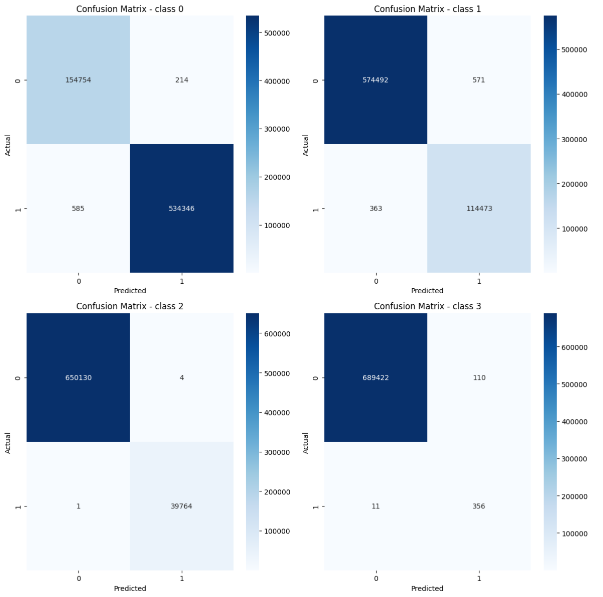


Fig. 17 Confusion matrix for multi-label classification.

Table 4 Results of the proposed approaches.

Approach	Accuracy	Precision	Recall	F1score
Binary class	100%	100%	100%	100%
Multi-class (OHE)	99.84%	93.25%	99.5%	96.25%
Multilabel class	99.87%	93.75%	99.25%	96.5%

3.8 Comparison of results

By comparing the results of the different approaches, we can see that the multi-class one-hot encoding and binary classification approaches produced very similar and very efficient results in terms of accuracy and loss, while the label encoder approach showed slightly worse performance in terms of accuracy and loss. The multi-label encoding approach produced impressive results with perfect accuracy and very low loss value. However, it is important to note that these results may vary depending on the dataset and the objectives of the analysis, and therefore it is important to choose the most suitable method.

We can also notice that the Loss values are very low for all the approaches, which indicates that the models succeeded in minimizing the prediction error. However, it can be seen that the label encoder approach produced higher Loss values than the other approaches, which confirms that this preprocessing method is not the most suitable for this dataset. On the other hand, the other approaches produced very similar and very low loss values, which shows that they are effective for this classification task.

Finally, Table 4 presents a comparison between the aforementioned approaches in terms of the "Accuracy", "Precision", "Recall" and "F1-score" metrics.

4 Conclusion

The work presented in this paper focuses on the design and development of a model using Deep Learning techniques for the detection and classification of a wide range of DDoS attacks using the CSE-CIC-IDS2018 dataset composed of ten files two of which contain DDoS attacks. These two files were merged to train and evaluate our model. Knowing that the performance of our model depends entirely on the quality of this data, we applied a preprocessing of the data such as the elimination of NAN values, infinity values, duplicate rows and useless columns which could negatively affect the performance of the model. model.

Furthermore, we found that the structure and quality of the data used to train the models played a crucial role in the performance obtained. We also investigated the effects of different hyperparameters on the performance of deep learning models, such as number of hidden layers, number of neurons in each hidden layer, activation functions, optimizer, type and regularization value, number of iterations, learning rate, etc. which have a significant impact on model performance.

In our study, we adopted different approaches to detect DDoS attacks. First, we used a binary classification that helps distinguish between normal traffic and malicious traffic. Then, we set up a multi-class classification which allows to identify several types of specific DDoS attacks. Finally, we used a multi-label classification capable of simultaneously detecting several DDoS attacks that may occur at the same time. These different approaches allowed us to evaluate the performance of each model for the detection of DDoS attacks. The results showed that binary classification achieved the highest Accuracy, followed by multi-label classification and then multi-class classification.

In perspective, we consider the detection of DDoS attacks in vehicular networks because these attacks are more serious in the vehicular ad hoc network (VANET) due to the diffusion of this attack which spreads over a large area of the network.

5 Compliance with Ethical Standards-Competing Interests-Research Data Policy and Data Availability Statements

5.1 Compliance with Ethical Standards

- The authors did not receive support from any organization for the submitted work.
- No funding was received to assist with the preparation of this manuscript.
- No funding was received for conducting this study.
- No funds, grants, or other support was received.

5.2 Competing Interests

- The authors have no relevant financial or non-financial interests to disclose.
- The authors have no competing interests to declare that are relevant to the content of this article.
- All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.
- The authors have no financial or proprietary interests in any material discussed in this article.

5.3 Research Data Policy & Data Availability Statements

CSE-CIC-IDS2018-DDoS dataset is available at the following link:

https://www.kaggle.com/datasets/wafaaferhi/cse-cic-ids2018-ddos

Original CSE-CIC-IDS2018 dataset is available at the following links:

https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv

https://www.unb.ca/cic/datasets/ids-2018.html

References

1. GUPTA, Brij B. et DAHIYA, Amrita. Distributed Denial of Service (DDoS) Attacks: Classification, Attacks, Challenges and Countermeasures. CRC press, 2021.
2. FENIL, Edwin et MOHAN KUMAR, P. Survey on DDoS defense mechanisms. Concurrency and Computation: Practice and Experience, 2020, vol. 32, no 4, p. e5114.
3. SINGH, Rajeev et SHARMA, T. P. Present Status of Distributed Denial of service (DDoS) attacks in internet world. International Journal of Mathematical, Engineering and Management Sciences, 2019, vol. 4, no 4, p. 1008.
4. SNEHI, Manish et BHANDARI, Abhinav. Vulnerability retrospection of security solutions for software-defined Cyber–Physical System against DDoS and IoT-DDoS attacks.Computer Science Review, 2021, vol. 40, p. 100371.
5. DI MAURO, Mario, GALATRO, Giovanni, FORTINO, Giancarlo,et al. Supervised feature selection techniques in network intrusion detection: A critical review. Engineering Applications of Artificial Intelligence, 2021, vol. 101, p. 104216.
6. KAMALOV, Firuz, MOUSSA, Sherif, EL KHATIB, Ziad, et al.Orthogonal variance-based feature selection for intrusion detection systems. In : 2021 International

Symposium on Networks, Computers and Communications (ISNCC). IEEE, 2021. p. 1-5.

7. WEI, Yuanyuan, JANG-JACCARD, Julian, SABRINA, Fariza, et al. Ae-mlp: A hybrid deep learning approach for ddos detection and classification. IEEE Access, 2021, vol. 9, p. 146810-146821.
8. ODUMUYIWA, Victor et ALABI, Rukayat. DDOS detection on Internet of things using unsupervised algorithms. Journal of Cyber Security and Mobility, 2021, p. 569–592-569–592.
9. CIL, Abdullah Emir, YILDIZ, Kazim, et BULDU, Ali. Detection of DDoS attacks with feed forward based deep neural network model. Expert Systems with Applications, 2021, vol. 169, p. 114520.
10. AMAIZU, Gabriel Chukwunonso, NWAKANMA, Cosmas Ifeanyi, BHARDWAJ, Sanjay, et al. Composite and efficient DDoS attack detection framework for B5G networks. Computer Networks, 2021, vol. 188, p. 107871.
11. KHEMPETCH, Thapanarath et WUTTIDITTACHOTTI, Pongpisit. DDoS attack detection using deep learning. IAES International Journal of Artificial Intelligence, 2021, vol. 10, no 2, p. 382.
12. HUSSAIN, Faisal, ABBAS, Syed Ghazanfar, HUSNAIN, Muhammad, et al. IoT DoS and DDoS attack detection using ResNet. In : 2020 IEEE 23rd International Multitopic Conference (INMIC). IEEE, 2020. p. 1-6.
13. JIA, Yizhen, ZHONG, Fangtian, ALRAWAIS, Arwa, et al.Flowguard: An intelligent edge defense mechanism against IoT DDoS attacks. IEEE Internet of Things Journal, 2020, vol. 7, no 10, p. 9552-9562.
14. SHURMAN, Mohammad M., KHRAIS, Rami M., YATEEM, Abdulrahman A., et al. DoS and DDoS attack detection using deep learning and IDS. Int. Arab J. Inf. Technol., 2020, vol. 17, no 4A, p. 655-661.
15. LI, Jiabin, LIU, Ming, XUE, Zhi, et al. RTVD: A realtime volumetric detection scheme for DDoS in the Internet of Things.IEEE Access, 2020, vol. 8, p. 36191-36201.
16. SHARAFALDIN, Iman, LASHKARI, Arash Habibi, HAKAK, Saqib, et al. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In : 2019 International Carnahan Conference on Security Technology (ICCST). IEEE, 2019. p. 1-8.
17. JAVEED, Danish, GAO, Tianhan, et KHAN, Muhammad Taimoor. SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT. Electronics, 2021, vol. 10, no 8, p. 918.
18. ALAMRI, Hassan A. et THAYANANTHAN, Vijey. Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks.IEEE Access, 2020, vol. 8, p. 194269-194288.
19. ASSIS, Marcos VO, CARVALHO, Luiz F., LLORET, Jaime, et al.A GRU deep learning system against attacks in software defined networks. Journal of Network and Computer Applications, 2021, vol. 177, p. 102942.
20. RAVIKUMAR, Dharshini. Towards Enhancement of Machine Learning Techniques Using CSE-CIC-IDS2018 Cybersecurity Dataset. Rochester Institute of Technology, 2021.
21. JIA, Bin-Bin et ZHANG, Min-Ling. Multi-dimensional classification via sparse label encoding. In : International Conference on Machine Learning. PMLR, 2021. p. 4917-4926.
22. RASHKA, S. et MIRDZHALILI, V. Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2.Birmingham, Mumbai. Packt, 2020.
23. TARIQ, Muhammad Imran, MEMON, Nisar Ahmed, AHMED, Shakeel, et al. A review of deep learning security and privacy defensive techniques. Mobile Information Systems, 2020, vol. 2020, p. 1-18.
24. MOONS, Bert, BANKMAN, Daniel, et VERHELST, Marian. Embedded deep learning. Embedded Deep Learning, 2019.
25. MICHELUCCI, Umberto. Applied Deep Learning with TensorFlow 2: Learn to Implement Advanced Deep Learning Techniques with Python. Apress, 2022.
26. GREENGARD, Samuel. AI Rewrites Coding. Communications of the ACM, 2023, vol. 66, no 4, p. 12-14.
27. RASCHKA, Sebastian, PATTERSON, Joshua, et NOLET, Corey. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information, 2020, vol. 11, no 4, p. 193.
28. SILAPARASETTY, N. Machine learning concepts with Python and the Jupyter Notebook Environment. Apress: Berkeley, CA, USA, 2020.