mathematics

MDPI

Article

Improvement of Distributed Denial of Service Attack Detection through Machine Learning and Data Processing

Fray L. Becerra-Suarez *, Ismael Fernández-Roman and Manuel G. Forero *

Professional School of Systems Engineering, Faculty of Engineering, Architecture and Urban Planning, Universidad Señor de Sipán, Chiclayo 14000, Lambayeque, Peru; fromanismae@uss.edu.pe
* Correspondence: bsuarezf@uss.edu.pe (F.L.B.-S.); fvargasmanuelgu@uss.edu.pe (M.G.F.)

Abstract: The early and accurate detection of Distributed Denial of Service (DDoS) attacks is a fundamental area of research to safeguard the integrity and functionality of organizations' digital ecosystems. Despite the growing importance of neural networks in recent years, the use of classical techniques remains relevant due to their interpretability, speed, resource efficiency, and satisfactory performance. This article presents the results of a comparative analysis of six machine learning techniques, namely, Random Forest (RF), Decision Tree (DT), AdaBoost (ADA), Extreme Gradient Boosting (XGB), Multilayer Perceptron (MLP), and Dense Neural Network (DNN), for classifying DDoS attacks. The CICDDoS2019 dataset was used, which underwent data preprocessing to remove outliers, and 22 features were selected using the Pearson correlation coefficient. The RF classifier achieved the best accuracy rate (99.97%), outperforming other classifiers and even previously published neural network-based techniques. These findings underscore the feasibility and effectiveness of machine learning algorithms in the field of DDoS attack detection, reaffirming their relevance as a valuable tool in advanced cyber defense.

Keywords: cybersecurity; DDoS attacks; CICDDoS2019 dataset; attack detection; data preprocessing; feature selection; outlier removal; interpretability

MSC: 68T09

Citation: Becerra-Suarez, F.L.; Fernández-Roman, I.; Forero, M.G. Improvement of Distributed Denial of Service Attack Detection through Machine Learning and Data Processing. *Mathematics* **2024**, *12*, 1294. https://doi.org/10.3390/math12091294

Academic Editors: Chi-Yao Weng, Shoko Wakamiya, Chun-Ta Li and Cheng-Ta Huang

Received: 21 March 2024
Revised: 12 April 2024
Accepted: 15 April 2024
Published: 25 April 2024

1. Introduction

The importance of cybersecurity in contemporary society is undeniable. With the increase in digital communication, cloud computing, mobile devices, and the Internet of Things (IoT), the number of possible points of attack has significantly increased [1]. Cybersecurity is responsible for protecting computer systems and networks against unauthorized intrusions, theft, damage, and service interruptions. In a landscape where cyber threats are becoming increasingly complex and sophisticated, the protection of digital assets becomes critical. Cyber attacks can have significant economic consequences, damage to reputation, and even the loss of confidential information. These attacks can target individuals, organizations, or governments, and their repercussions can be far-reaching.

In the context of contemporary cybersecurity, Distributed Denial of Service (DDoS) attacks constitute a formidable class of cyberattacks whose purpose is to overload or overwhelm access to a network or server's resources. This goal is achieved by sending large volumes of traffic to hinder its regular operation and, consequently, induce the inoperability of these systems [2–4]. As a result, the affected network or server may become inaccessible or operate extremely slowly, causing disruptions in functionality and economic losses for the affected organization. An illustrative example of this problem is CloudFlare, a network services company, which suffered a DDoS attack in early 2023, with over 71 million requests per second, marking the largest DDoS attack experienced that year [5]. Even the artificial intelligence company behind the development of ChatGPT confirmed the impact of its services by this type of attack [6].

Modern DDoS attacks exhibit notable complexity, characterized by the use of a dynamic combination of multiple attack vectors, and follow a process of continuous evolution and expansion [3,7]. These vectors include volumetric attacks that involve considerable bandwidth consumption, TCP state exhaustion attacks, and low-and-slow application layer attacks. Multivector DDoS attacks are not new, but their launch has increased by attackers, due to the wide availability of specialized tools and online rental services. These strategies are typically driven by attackers who use a network of compromised computers, commonly referred to as a botnet, further complicating the task of identifying the origin of an attack. The proliferation of these botnets has increased the difficulty of mitigating DDoS attacks and has increased the challenges in protecting critical digital infrastructures.

In the face of this growing threat of DDoS attacks, organizations and states have the option to implement different cybersecurity solutions, such as Digital Attack Map [8], Fortinet [9] and Darktrace [10], among others, which offer advanced real-time monitoring capabilities and ensure multilayered protection to detect and mitigate DDoS attacks [11]. However, it is important to note that these solutions, especially in the case of small-scale networks, tend to involve significant costs [12], largely owing to the need for continuous training and the demand for highly skilled professionals to manage these solutions, which affects their accessibility. On the other hand, other solutions based on traditional approaches, such as traffic protection systems, firewalls, and managed security services, are proving to be increasingly less effective in containing the constant flow of these attacks circulating on the network [13].

Machine learning techniques have emerged as promising tools for the effective identification and mitigation of these attacks. Classical approaches, such as Random Forest, Decision Trees, AdaBoost, Extreme Gradient Boosting, and others, remain relevant in DDoS detection because of their interpretability, speed, resource efficiency, and satisfactory performance. Additionally, they are valuable complements to more modern machine learning approaches, allowing them to play an essential role in cybersecurity and the protection of networks and systems against DDoS attacks. However, it is important to note that machine learning models often misclassify the most dangerous traffic flows owing to inadequate or poor feature selection or using datasets that are now outdated.

For example, Sadhwani et al. [14] implemented different classifiers, such as logistic regression (LR), Random Forest (RF), Naive Bayes (NB), artificial neural networks (ANNs), and K-nearest neighbor (KNN), to classify DDoS attacks in the BOT-IOT and TON-IOT datasets. During preprocessing, missing value handling was addressed, but the number of records affected by this process was not specified. Regarding features, 15 were selected using the ExtraTreeClassifier algorithm for both datasets. Data imbalance was addressed using the SMOTE technique. Regarding the validation of classifier performance, it was observed that in the TON-IOT dataset, RF was the best algorithm for multiclass classification, with an accuracy of 100%, whereas for binary classification, NB was the best with an accuracy of 100%. In the BOT-IOT dataset, NB achieved an accuracy of 100% for both binary and multiclass classification. Liu et al. [15] also implemented RF, SVM, KNN, DT, and XGBoost, using the CSE-CIC-IDS2018 dataset, demonstrating that RF is the best classifier with an accuracy of 98.95%.

Ma et al. [16] presented a DDoS attack detection algorithm that uses feature selection along with the Random Forest (RF) algorithm, using the CIC-DDoS2019 dataset. This dataset underwent preprocessing resulting in the final selection of 24 features. All records were normalized to the range of [0, 1]. Hyperparameter selection for the RF classifier was performed using built-in functions. The experimental results showed an accuracy of 100%. It is important to note that these results may be overestimated because the model was trained using an unequal distribution of data, with 67% for training and 33% for testing.

Lv et al. [17] proposed a Decision Tree (DT)-based classifier for DDoS attack classification using the MBB-IoT dataset. Owing to the large size of the dataset, they used only 1% of the records, resulting in a total of 77 features to evaluate the classifier. Due to data imbalance, four strategies were applied to address this issue: SMOTE, K-Means SMOTE,

Gaussian Probability Distribution, and KG-SMOTE. The results of the multiclass classification showed the clear superiority of the KG-SMOTE method over the other methods, with precision, recall, F1, and AUC rates of 97.61%, 96.72%, 97.12%, and 98.15%, respectively. However, the study does not discuss the processing time required by this method, which represents a significant limitation that needs to be addressed in future research.

Hnamte et al. [13] implemented a deep neural network (DNN) consisting of three hidden layers with 128, 256, and 128 neurons, respectively. This network includes a regularization technique known as dropout with a rate of 0.1, and uses Softmax and Relu activation functions. This architecture was trained for 30 epochs, using batch sizes of 128, the Adam optimizer, and a learning rate of 0.0001. These hyperparameters were experimentally selected to calibrate the model and improve its performance. The effectiveness of this architecture was evaluated with different freely available datasets, such as InSDN, CICIDS2018 and Kaggle DDoS, achieving binary classification results of 99.98%, 100%, and 99.99%, respectively, in terms of accuracy.

Najar and Manohar [1] proposed a convolutional neural network (CNN)-based approach to detect DDoS attacks, using the CICIDS2019 dataset. From this dataset, 66 features were selected using balanced random sampling (BRS) and arbitrary selection to balance the data across all classes. A total of 672,300 records constituted the new dataset. However, the amount of duplicate, empty, and other records was not specified. This new dataset underwent min–max normalization and was used to validate the performance of the CNN architecture, whose hyperparameters were arbitrarily selected. The model exhibited performance in binary and multiple classifications, with an accuracy exceeding 99.99% and 98.64%, respectively.

Mustaphaa et al. [11] presented a method based on the Long Short-Term Memory (LSTM) model to classify Distributed Denial of Service (DDoS) attacks using the CICIDS2019 and CICIDS2017 datasets. Both datasets were merged and preprocessed, resulting in a new dataset with 67 features and 251,723 records for each benign traffic and DDoS attack class. When evaluating the performance of this model, it was compared to other machine learning classifiers such as Decision Trees (DT), Multilayer Perceptron (MLP), XGBoost, and Random Forest (RF), resulting in the proposed LSTM-based model outperforming the other classifiers with a true F1 score of 99% and a false F1 score of 98.25%. In the same context, Ahmad et al. [18] presented a hybrid model called HD-IDM that combines the GRU and LSTM classifiers. This model was evaluated using the CSE-CIC-IDS2017, CSE-CIC-IDS2018, NSL KDD, and CIC-DDoS2019 datasets, achieving a maximum accuracy of 99.91% on the CIC-DDoS2019 dataset. This result was compared to other classifiers, including the Gaussian Naive Bayes (GNB), Gradient Boosting (GB), Multilayer Perceptron (MLP), and Random Forest (RF) algorithms, demonstrating the superiority of the proposed model. Ragab et al. [19] proposed a Harris Hawks optimizer called PHHO-ODLC for feature extraction and an Attention-Based Long Short-Term Memory Bidirectional Memory Network (ABiLSTM) for classifying DDoS attacks from the BoT-IoT dataset. The accuracy rate obtained for binary classification was 99.2%, whereas for multiclass classification, it was 98.83%.

Setrita et al. [20] introduced a comprehensive approach called OptMLP-CNN, which combines advanced techniques such as SHAP feature selection, a fused architecture of Multilayer Perceptron (MLP) and a convolutional neural network (CNN), and the use of Bayesian optimization along with the ADAM optimizer. This model was evaluated using the InSDN and CICDDoS-2019 datasets. In both datasets, 20 features were selected using the SHAP method. The evaluation of the proposed model showed an accuracy rate of 99.95% and 99.98% for the CICDDoS-2019 and InSDN datasets, respectively. These results indicate exceptional performance in DDoS attack classification. However, it is important to note that these results only refer to binary classification.

Adeniyi et al. [21] proposed a new hybrid model called autoencoder–Multilayer Perceptron (AE-MLP) and validated its performance using the NF-UQ-NIDS-V2 dataset. In the first part of the experiment, by splitting the dataset into an 80/20 training and

testing ratio, they achieved the highest model accuracy, reaching 99.98%. The recorded training time was 4.96 s, whereas the prediction time was 0.74 s. In the second part of the experiment, the proposed AE-MLP hybrid model was compared with other deep learning models. It was found that the AE-MLP model outperformed LSTM and GRU, achieving an accuracy of 99.98%, compared to 94.98% and 97.10%, respectively. Additionally, it was compared to the MLP model, demonstrating a higher accuracy of 99.98% versus MLP's 98.63%. Finally, it was compared to a shallow machine learning model, recording a training time of 4.96 s, with an accuracy also of 99.98%.

Ramzan et al. [22] used three deep learning models, namely RNN, LSTM, and GRU, to perform binary and multiclass classification of DDoS attacks using the CICDDoS2019 dataset. A total of 20 features were selected using an additional tree classifier. The dataset was split into 70% for training and 30% for testing. The results of the binary classification showed that RNN was the most effective method, with an accuracy of 99.99%. On the other hand, the results of the multiclass classification indicated that GRU was the most effective method, with an accuracy of 98%.

In general, these previous studies on the intrusion detection problem suffer from the following limitations:

- In machine learning, data processing is considered essential to achieve good results using any machine learning model. However, in some studies, the data preprocessing was not detailed, or outliers were not taken into account [16,18,19]. In other studies [1,13,17,20,22], it was not specified how many records were affected after processing the outliers.
- A crucial aspect when working with machine learning algorithms is the appropriate selection of the hyperparameters that a model should use. In many studies, this selection was performed arbitrarily [20–22]. In other cases, the hyperparameters used were not justified, making implementation difficult, or only the author only mention the algorithm used without specifying its hyperparameters [17].
- The distribution of data between training and validation sets is a critical factor in evaluating algorithm performance. However, in [18], the proportion of data used in each group was not specified. Additionally, in other studies, it was observed that only a subset of data from the original dataset was used [14,21], which can lead to data leakage.
- Another fundamental aspect in DDoS attack detection is the response time to these events. In most of the reviewed studies, this factor was not addressed, except for [1], which only specified the inference time.

To overcome the aforementioned challenges, this study compares six different machine learning models and presents a methodology for preprocessing and feature selection using the CICDDoS2019 dataset, achieving high accuracy despite a reduction in features. Unlike previous approaches that often focus on individual machine learning techniques, this work stands out for its comprehensive approach involving a detailed comparison of six different models. We present an exhaustive methodology for preprocessing and feature selection using the CICDDoS2019 dataset. We employed techniques such as Principal Component Analysis (PCA) and Pearson correlation to study the features. Additionally, hyperparameter optimization was performed using the Tree of Parzen Estimators method. This unique approach, involving the thorough evaluation of multiple machine learning models in a single environment alongside advanced preprocessing and feature selection techniques, allowed us to achieve high accuracy in DDoS attack detection, despite the reduction in the number of features. This direct and exhaustive comparison of models, together with the use of advanced preprocessing and feature selection techniques, reinforces the premise of our work and highlights the unique contribution of this study to the field of DDoS attack detection.

2. Materials and Methods

2.1. Materials

For the development of this research, the CIC-DDoS2019 dataset, published by Sharafaldin et al. [23,24], was used, which stands out as one of the most comprehensive collections in the field of DDoS attacks. Data were collected using the TCP/UDP protocol. In total, this compilation covers 18 types of DDoS attacks, with a total of 431,371 records, and comprises 80 features. Of these, 333,540 records correspond to DDoS attacks, while 97,831 records are benign.

The classification algorithms were implemented using Python 3.10 programming language and Scikit-learn library. The calculations were performed on a system powered by an AMD Ryzen 7 3700u processor, equipped with a Radeon Vega Mobile GPU at 2.3 GHz (HP, Chiclayo, Lambayeque), backed by 24 GB of RAM, and running the 64-bit Windows 11 operating system.

2.2. Method

2.2.1. Data Preprocessing

Data preprocessing is a fundamental stage in which a series of transformations and adjustments are made to the data to improve both the data quality and the results obtained [25]. In our case, for the selected dataset, an analysis of the features with identical values in all records was performed, which were eliminated. Likewise, records with duplicate, null, positive infinite, and negative infinite values were also removed. Thus, the original database contained 80 features and 431,371 records. To simplify it, improve processing time, and reduce the required memory space for processing, a dataset analysis was carried out. This analysis revealed that 12 features contained only zero values in all records, 1 feature represented the numbering of all records, and another feature was redundant as it contained the attack vector, which was not useful for data analysis since another feature indicates whether the record corresponded to an attack or not. Therefore, these 14 features were removed, reducing the number of features from 80 to 66.

Additionally, an evaluation was conducted on records with values such as NaN (Not a Number), positive infinity (+inf), negative infinity ($-$inf), and duplicate values, totaling 12,612 records. These records were removed from the dataset, reducing the total from 431,371 to 418,759 records.

2.2.2. Feature Selection

The reduced database was analyzed using the Principal Component Analysis (PCA) algorithm to determine the data variability and establish the number of descriptors that provide significant information for detecting DDoS attacks. As shown in Figure 1, 95% of the variance is explained by 23 principal components, where the x-axis represents the number of components and the y-axis represents the cumulative explained variance.

To determine the possible relationships between the 23 principal components and the highly correlated descriptors, the correlation between each pair of descriptors was calculated. In order to simplify the process and avoid using PCA, which would increase the database processing time, a new dimensionality reduction analysis was performed using the Pearson correlation coefficient (represented as r) as a similarity metric, as described in the following equation:

$$r = \frac{n\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{n\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}, \quad (1)$$

where x_i and y_i are the values of the variables x and y for the i-th data point; $\bar{x}$ and $\bar{y}$ are the means of the variables x y y. This coefficient is a measure that ranges from -1 to 1. Its primary function is to evaluate both the strength and direction of the relationship between two variables, in this case, the observed value (X) and the predicted value (Y). A value of "0" indicates no linear relationship between the two variables, while "1" denotes a perfect positive correlation where both variables increase simultaneously. On the other

hand, a value of "−1" suggests a perfect negative correlation where one variable decreases as the other increases. In Figure 2, the results obtained by applying the Pearson correlation coefficient to the 66 features of the new dataset are presented.

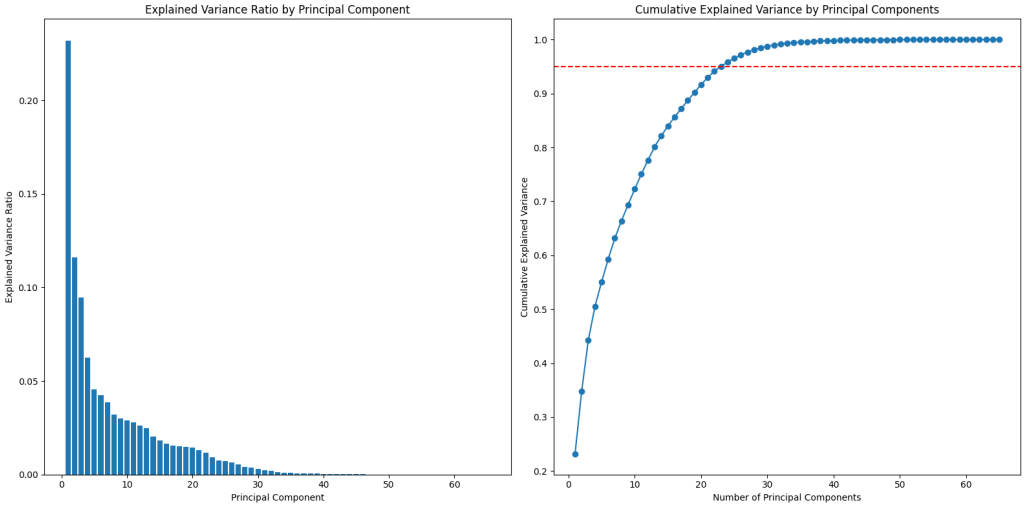

Figure 1. Cumulative variance analysis by number of components with a 95% variance threshold.

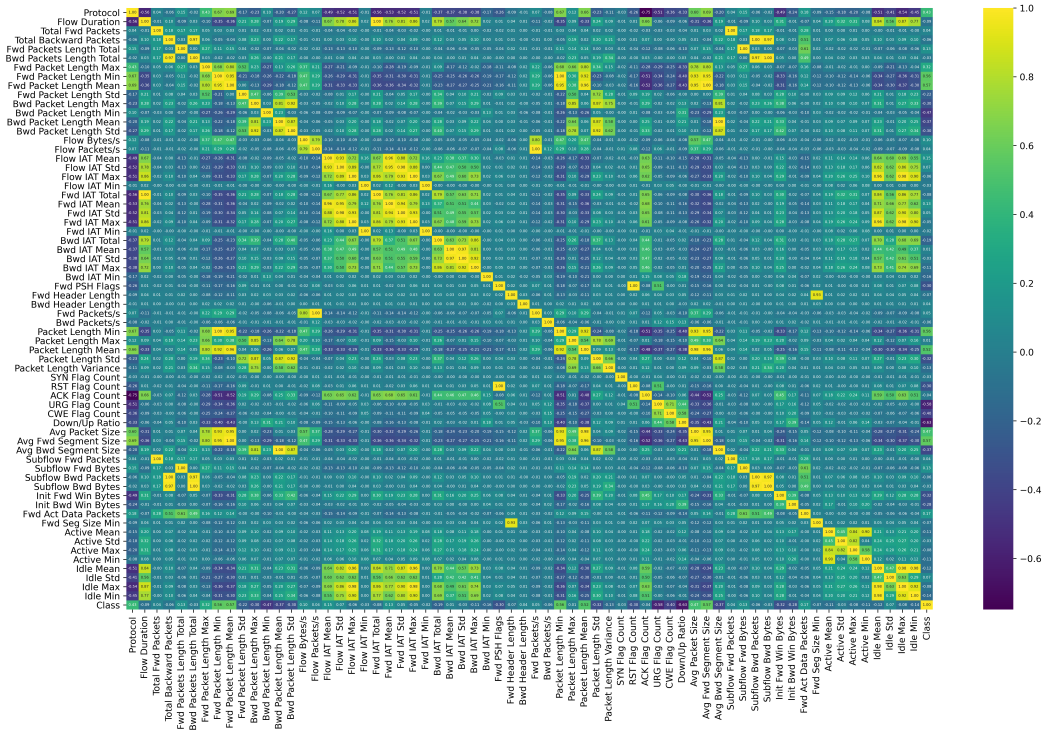

Figure 2. Correlated values between each feature of the new dataset CIC-DDoS2019.

Given that these types of DDoS attacks need to be detected quickly, our goal was to minimize the number of features to simplify detection. In this context, a feature reduction was performed, looking for those whose mutual correlation exceeded 0.9 or was below −0.9, and keeping only one of them. As a result, a new dataset was obtained consisting of 22 non-redundant features, along with their respective label, where 0 represents a benign event and 1 represents an attack. These values are detailed in Table 1.

Table 1. Final features of the CIC-DDoS2019 dataset.

#	Characteristics	Min and Max Values
1	Protocol	[0; 17]
2	Fwd Packet Length Max	[0; 32,120]
3	Fwd Packet Length Std	[0; 2221.5562]
4	Bwd Packet Length Min	[0; 1460]
5	Flow Bytes/s	[0; 2,944,000,000]
6	Bwd IAT Total	[0; 119,943,720]
7	Bwd IAT Min	[0; 249]
8	Bwd Header Length	[−2,125,437,950; 1,478,492,170]
9	Bwd Packets/s	[0; 2,000,000]
10	Packet Length Max	[0; 37,960]
11	Packet Length Variance	[0; 43,778,892]
12	SYN Flag Count	[0; 1]
13	ACK Flag Count	[0; 1]
14	URG Flag Count	[0; 1]
15	CWE Flag Count	[0; 1]
16	Down/Up Ratio	[0; 23]
17	Init Fwd Win Bytes	[−1; 65,535]
18	Init Bwd Win Bytes	[−1; 65,535]
19	Fwd Act Data Packets	[0; 18,766]
20	Active Std	[0; 21,352,442]
21	Active Max	[0; 45,536,680]
22	Idle Std	[0; 45,536,680]

2.2.3. Data Normalization

After obtaining a simplified dataset with various features (Table 1), the normalization of records was carried out using the min–max technique to confine values within a range between 0 and 1. This was performed with the aim of optimizing the performance of the classifiers used and mitigating the effect of outliers. Normalization was conducted according to the following mathematical formula:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{2}$$

where X represents the original value of the feature, X' is the normalized value, $X_{\min}$ is the minimum value of the feature, and $X_{\max}$ is the maximum value of the same. This ensures that all features in the dataset are within the same range, facilitating comparison and enhancing result interpretation in subsequent stages of the analysis.

2.2.4. Hyperparameter Tuning

The proper selection of hyperparameters is crucial to achieve optimal performance in implementing any machine learning algorithm [20]. While various studies have employed grid search methods or random hyperparameter selection, it is important to note that such approaches are computationally expensive when exploring the entire search space [26]. For this reason, it was decided to use a Bayesian search algorithm, proposed by [27], which allows for the efficient search of the optimal set of hyperparameters through an iterative process. This method, known as Tree of Parzen Estimators (TPE), has proven to be highly effective in hyperparameter optimization for machine learning models. Through TPE, the hyperparameter space can be explored systematically and efficiently, resulting in a better fit for the selected machine learning models and, ultimately, in improved performance in DDoS attack detection.

The TPE algorithm is a Bayesian method for hyperparameter optimization. It is based on constructing a Tree of Parzen Estimators to approximate the posterior distribution of the hyperparameters. The TPE algorithm operates as follows:

We define a search space containing the set of all possible values for each hyperparameter of the machine learning model.

We define the likelihood function, which measures the probability of observing the data given by a set of hyperparameters. It is expressed as

$$P(D|\theta) = \prod_i P(y_i|x_i, \theta) \tag{3}$$

where the variables are as follows:

- D: data.
- θ: hyperparameters.
- $P(D|\theta)$: likelihood of the data given the hyperparameters.
- $P(y_i|x_i, \theta)$: likelihood of the observation y_i given a point x_i in the search space and the hyperparameters θ.

Bayes' rule is used to update the probability of the hyperparameters given the data, using the following expression:

$$P(\theta|D) \propto P(D|\theta)P(\theta) \tag{4}$$

where the variables are as follows:

- $P(\theta|D)$: posterior probability of the hyperparameters given the data.
- $P(\theta)$: prior of the hyperparameters.

The Parzen Estimator is used to estimate the probability density of a distribution, defined as

$$f(x) = \sum_i K(x - x_i)/h \tag{5}$$

where the variables are as follows:

- K: kernel function.
- h: bandwidth.

The Tree of Parzen Estimators was constructed recursively by dividing the search space into subsets. In each subset, a Parzen Estimator was used to estimate the probability density.

Finally, the selection criterion was used to choose the next set of hyperparameters to evaluate. This was based on information acquisition, such as entropy reduction.

Table 2 shows a comparison of the tuning time to adjust the hyperparameters of four machine learning methods and two hyperparameter optimization methods, GridSearch and TPE. The results highlight the superior performance of the TPE method, which is why this method was chosen to search for the best set of hyperparameters in our study. TPE proved to be more efficient in exploring the hyperparameter space, resulting in significantly shorter tuning times compared to GridSearch. This efficiency in the hyperparameter search allowed for faster and more effective optimization of our machine learning models, thus contributing to improved accuracy and overall performance in DDoS attack detection.

Table 2. Comparison of the time taken for hyperparameter selection between GridSearch and TPE.

ML Classifier	GridSearch	TPE
RF	768 m 13.7 s	11 m 51 s
ADA	653 m 17.5 s	9 m 36 s
XGB	616 m 56.2 s	2 m 4 s
DT	207 m 47.6 s	25 s

2.2.5. Machine Learning Algorithms and Performance Evaluation

To identify DDoS attacks, six machine learning algorithms were selected: Random Forest (RF), Decision Tree (DT), AdaBoost (ADA), XGBoost (XGB), Multilayer Perceptron (MLP), and Deep Neural Network (DNN). These selections were based on their proven effectiveness in attack detection from previous studies. RF and DT algorithms are known for handling complex and large datasets, along with their strength in identifying patterns

in nonlinear data. ADA and XGB are ensemble methods that improve model accuracy by combining weaker models. MLP and DNN, on the other hand, are neural network models capable of learning and adapting to complex, nonlinear data patterns, making them particularly suitable for DDoS attack detection due to the diverse characteristics and behaviors these attacks can exhibit. This combination of six algorithms provided a variety of approaches and techniques, ultimately improving the system's ability to effectively identify and mitigate DDoS attacks in real time.

All analyses of these classifiers were carried out using the simplified and normalized dataset. This dataset was randomly split into 80% for hyperparameter tuning using cross-validation, with k = 5. This splitting strategy ensured the use of all training data over the five iterations, thus reducing potential biases and data leakage. The remaining 20% was reserved as a test set.

The performance evaluation of the classifiers was conducted by analyzing various metrics, including accuracy, precision, recall, F1 score, and the area under the curve (AUC), whose mathematical expressions are presented in the corresponding equations:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{9}$$

$$\text{AUC} = 1 - \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN}} \tag{10}$$

where TP, FP, TN, and FN represent the number of true positives, false positives, true negatives, and false negatives, respectively.

3. Results

To determine the optimal hyperparameter values, an iterative process was conducted. Initially, 10% of the training dataset was randomly selected, and arbitrary values were assigned to the hyperparameters. Subsequently, the performance of the model was evaluated using the TPE algorithm. If satisfactory results were not achieved, additional adjustments were made to the hyperparameter values to improve the model's performance. Once an accuracy greater than 99% was achieved for all models, fine-tuning was performed by reducing the search range for the hyperparameter values, detailed in Table 3, and the full training set was used to find the optimal hyperparameters.

Table 3. Range of hyperparameters used for tuning DDoS attack classification models.

ML Classifier	Space Values
RF	max_depth = range(10, 16) n_estimators = range(35, 46) criterion = ["gini", "entropy"] max_features = range(0.01, 1)
DT	criterion = ["gini", "entropy"] splitter =["best", "random"] max_depth = range(1, 10) min_samples_split = range(2, 30) min_samples_leaf = range(1, 15)

Table 3. *Cont.*

ML Classifier	Space Values
ADA	learning_rate= range(0, 1) n_estimators= range(20, 75) algorithm = [“SAMME, “SAMME.R”]
XGB	n_estimators = range(50, 100) max_depth = range(1, 10) learning_rate = range(0, 1) gamma = range(0.0, 1.0) min_child_weight = range(1, 10))
MLP	hidden_layer_sizes = [(32,), (64,), (128,)] activation = [“relu”, “tanh”, “logistic”] alpha = [0.0001, 0.01] solver = [“adam”]
DNN	layers = [[64, 32], [128, 64], [256, 128]] activation= [“relu”, “tanh”] dropout_rate = range(0.0, 0.5) optimizer = [“adam”, “rmsprop”] batch_size=[32, 64, 128] epochs = [10, 20, 30, 40, 50]

In Table 4, the optimal hyperparameter configurations obtained from the search space for the different machine learning models are presented, along with the accuracy and training time. As can be seen, the RandomForest and Xtreme Gradient Boosting techniques achieved the highest accuracy of 99.95%, while Decision Tree closely followed with 99.88%. However, the training time for the Decision Tree method was much shorter, at only 25 s, followed by Xtreme Gradient Boosting with 2 min and 4 s. On the other hand, the MLP and DNN methods achieved slightly lower accuracies of 99.66% and 99.70%, respectively, but required considerably longer training times, with 2 h, 14 min, and 40 s for MLP and 50 min and 14 s for DNN.

Table 4. Optimal hyperparameter configurations, accuracy, and time for DDoS attack classification models.

ML Classifier	Best Hyperparameters	Accuracy	Training Time
RF	{‘criterion’: ‘entropy’, ‘max_depth’: 12, ‘max_features’: 0.9145, ‘n_estimators’: 43}	99.95%	11 m 51 s%
DT	{‘criterion’: ‘gini’, ‘min_samples_split’: 24, ‘max_depth’: 7, ‘min_samples_leaf’: 10, ‘splitter’: ‘best’}	99.88%	25 s%
ADA	{‘algorithm’: ‘SAMME.R’, ‘learning_rate’: 0.55, ‘n_estimators’: 68}	99.59%	9 m 36 s%
XGB	{‘gamma’: 0.21, ‘learning_rate’: 0.65, ‘max_depth’: 8, ‘min_child_weight’: 1, ‘n_estimators’: 80}	99.95%	2 m 4 s%
MLP	{‘activation’: ‘relu’, ‘alpha’: 0.0001, ‘hidden_layer_sizes’: (32,), ‘solver’: ‘adam’}	99.66%	2 h 14 m 40 s%
DNN	{‘activation’: ‘tanh’, ‘batch_size’: 128, ‘dropout_rate’: 0.070, ‘epochs’: 50, ‘layers’: (256, 128), ‘optimizer’: ‘adam’}	99.70%	50 m 14 s%

Once the models were tuned, they were evaluated using the test set that had not been previously used. The results obtained are presented in the confusion matrix for each classifier shown in Table 5, where it can be observed that all models correctly classified a

significant number of instances. The high number of true positives (TP) and true negatives (TN) shows that the models efficiently detected normal instances and DDoS attacks, while the number of false positives and false negatives is very low, indicating the classifiers' adequate performance using the simplified database.

Table 5. Confusion matrix results for binary classification.

ML Classifier	Benign	Attack	
RF	18,876	13	Benign
	12	64,851	Attack
DT	18,771	66	Benign
	28	64,787	Attack
ADA	18,699	190	Benign
	147	64,716	Attack
XGB	18,874	15	Benign
	22	64,841	Attack
MLP	18,759	130	Benign
	163	64,700	Attack
DNN	18,833	56	Benign
	194	64,669	Attack

With these results, the Random Forest (RF) classifier exhibited the best performance compared to the other classifiers (Table 6). The model's accuracy, reaching 99.97%, reflects an exceptional level in predicting both normal and attack events. Additionally, the F1 score of 99.98%, which balances precision and recall, highlights the model's ability to classify attacks and minimize false negatives accurately. This underscores the model's ability to achieve high precision and effectively manage false alarms. Furthermore, the AUC score of 99.96% emphasizes the model's strong discriminatory power to effectively discern benign and malicious network traffic.

Table 6. Performance of ML algorithms for binary classification of DDoS attacks.

ML Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC (%)
RF	99.97	99.98	99.80	99.98	99.96
DT	99.89	99.90	99.96	99.23	99.80
ADA	99.60	99.70	99.77	99.74	99.38
XGB	99.96	99.98	99.97	99.97	99.94
MLP	99.65	99.80	99.75	99.77	99.53
DNN	99.70	99.91	99.70	99.80	99.70

As observed in the results, the RF model demonstrated the best performance, with an accuracy of 99.97% and an F1 score of 99.98%, using the simplified database obtained in this work, compared to models from other studies detailed in Table 7. This table provides a comprehensive evaluation of previous studies that used the CICDDoS2019 dataset and different machine learning techniques, including deep learning (DL) and the technique proposed in this work. As can be seen, the results obtained here surpass those achieved in previous works, except for those proposed in [16,22], with a difference of just 0.02 in accuracy and 0.01 in the F1 score. It is worth noting that in [16], the number of features is 24, in ours it is 22, while in the research of [22] it is 20 features. Another aspect to consider

is that both studies used a random split for training and testing, which can face the issue of data leakage, compared to our study, where cross-validation (k = 5) was applied to ensure the representativeness of all data during the training phase.

Table 7. Comparison of the results obtained in other work.

Ref.	Approach	Features	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
[1]	CNN	66	98.64	99.0	99.0	99.0
[11]	LSTM	67	-	-	-	99.0
[16]	RF	24	99.99	99.99	99.99	99.99
[18]	Hybrid GRU and LSTM	-	99.91	99.62	99.43	99.52
[20]	OptMLP-CNN	20	99.95	99.90	99.98	99.93
[22]	RNN	20	99.99	99.99	99.99	99.99
	LSTM		99.99	99.0	99.0	99.0
	GRU		99.99	99.0	100	100
Our study	RF (Our approach)	22	99.97	99.98	99.80	99.98

The integration of machine learning models into network security systems could significantly improve real-time DDoS attack detection and mitigation, thus strengthening the protection of online resources. The adaptability of these models to new traffic patterns allows them to cope with the increasing sophistication and variety of DDoS attacks, providing a more robust defense in ever-changing cybersecurity environments. This adaptability makes them an invaluable tool to complement and enhance existing cybersecurity frameworks, which often require continuous updates to keep up with these threats.

4. Conclusions

This study demonstrated the effectiveness of applying different machine learning algorithms, such as RF, DT, ADA, XGB, MLP, and DNN, in the detection and classification of Distributed Denial of Service (DDoS) attacks. The results highlighted the superiority of certain algorithms, such as RF, in terms of accuracy and F1 score in classifying DDoS attacks in a new simplified dataset obtained from CICDDoS2019.

Furthermore, the importance of data preprocessing and feature selection to improve the effectiveness of machine learning models in detecting DDoS attacks was shown. The use of techniques such as feature selection through the Tree of Parzen Estimators (TPE) algorithm and data normalization within a specific range significantly contributed to the accuracy of the models.

The research underscores the importance of applying machine learning techniques in cybersecurity, especially in contexts where speed and accuracy are paramount. Although deep learning-based studies show promising results, this study demonstrates that not everything is about deep learning. Simple, easy-to-interpret, and implementable models can be used to address DDoS attacks. This comparative analysis, coupled with careful algorithm selection, empowers organizations to anticipate and effectively counter DDoS threats, thus protecting the integrity of their systems and data in an ever-evolving digital environment.

Given the increasingly sophisticated evolution of DDoS attacks, one of the limitations of this work is the ability to adjust the model to new types of attacks. Therefore, additional research should be conducted focused on the adaptability of the model to the evolution of these attacks. This could involve exploring more advanced machine learning techniques or incorporating dynamic model update mechanisms to ensure their effectiveness in detecting emerging attacks

In future research, the exploration of hybrid models emerges as a very promising strategy to improve the accuracy and efficiency of DDoS attack detection. While the results obtained with the CIC-DDoS2019 dataset are favorable, it is important to recognize that their generalization could be limited. Therefore, it is recommended to broaden the methodological diversity to evaluate a wider range of more recent datasets, algorithms, and techniques. It is also necessary to consider the analysis of the response time in real-time DDoS attack situations, which is a critical aspect for cybersecurity. Addressing these limitations in future research will not only strengthen the validity and applicability of the results but will also contribute to significantly improving defense strategies against DDoS attacks in increasingly challenging environments.

Author Contributions: Conceptualization, M.G.F. and F.L.B.-S.; supervision, M.G.F.; methodology, validation, and formal analysis, M.G.F. and F.L.B.-S.; software and data curation, all authors; writing, review, editing, and visualization, M.G.F. and F.L.B.-S.; project administration and funding acquisition, M.G.F. and F.L.B.-S. All authors have read and approved the final version of the manuscript for publication.

Funding: This research received no external funding, and the APC was funded by Universidad Señor de Sipán (Peru).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Najar, A.A.; Manohar, S. Cyber-Secure SDN: A CNN-Based Approach for Efficient Detection and Mitigation of DDoS attacks. *Comput. Secur.* **2024**, *139*, 103716. [CrossRef]
2. Bravo, S.; Mauricio, D. Systematic review of aspects of DDoS attacks detection. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *14*, 155–168. [CrossRef]
3. Li, Q.; Huang, H.; Li, R.; Lv, J.; Yuan, Z.; Ma, L.; Han, Y.; Jiang, Y. A comprehensive survey on DDoS defense systems: New trends and challenges. *Comput. Netw.* **2023**, *233*, 109895. [CrossRef]
4. Behal, S.; Kumar, K.; Sachdeva, M. Characterizing DDoS attacks and flash events: Review, research gaps and future directions. *Comput. Sci. Rev.* **2017**, *25*, 101–114. [CrossRef]
5. The Cloudflare Blog. Available online: http://blog.cloudflare.com/cloudflare-mitigates-record-breaking-71-million-request-per-second-ddos-attack/ (accessed on 20 January 2024).
6. OpenAI Status. Available online: https://status.openai.com/history (accessed on 4 February 2024).
7. Bahashwan, A.A.; Anbar, M.; Manickam, S.; Al-Amiedy, T.A.; Aladaileh, M.A.; Hasbullah, I.H. A Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-Defined Networking. *Sensors* **2023**, *23*, 4441. [CrossRef] [PubMed]
8. Digital Attack Map. Available online: https://www.digitalattackmap.com/ (accessed on 20 August 2023).
9. Fortinet Threat Map. Available online: https://threatmap.fortiguard.com/ (accessed on 20 August 2023).
10. Darktrace. Available online: https://es.darktrace.com/ (accessed on 8 February 2024).
11. Mustapha, A.; Khatoun, R.; Zeadally, S.; Chbib, F.; Fadlallah, A.; Fahs, W.; Attar, A.E. Detecting DDoS attacks using adversarial neural network. *Comput. Secur.* **2023**, *127*, 103117. [CrossRef]
12. Dayal, N.; Srivastava, S. Analyzing effective mitigation of DDoS attack with software defined networking. *Comput. Secur.* **2023**, *130*, 103269. [CrossRef]
13. Hnamte, V.; Najar, A.A.; Nhung-Nguyen, H.; Hussain, J.; Sugali, M.N. DDoS attack detection and mitigation using deep neural network in SDN environment. *Comput. Secur.* **2024**, *138*, 103661. [CrossRef]
14. Sadhwani, S.; Manibalan, B.; Muthalagu, R.; Pawar, P. A Lightweight Model for DDoS Attack Detection Using Machine Learning Techniques. *Appl. Sci.* **2023**, *13*, 9937. [CrossRef]
15. Liu, Z.; Wang, Y.; Feng, F.; Liu, Y.; Li, Z.; Shan, Y. A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks. *Sensors* **2023**, *23*, 6176. [CrossRef] [PubMed]
16. Ma, R.; Wang, Q.; Bu, X.; Chen, X. Real-Time Detection of DDoS Attacks Based on Random Forest in SDN. *Appl. Sci.* **2023**, *13*, 7872. [CrossRef]
17. Lv, H.; Du, Y.; Zhou, X.; Ni, W.; Ma, X. A Data Enhancement Algorithm for DDoS Attacks Using IoT. *Sensors* **2023**, *23*, 7496. [CrossRef] [PubMed]
18. Ahmad, I.; Imran, M.; Qayyum, Q.; Ramzan, M.S.; Alassafi, M.O. An Optimized Hybrid Deep Intrusion Detection Model (HD-IDM) for Enhancing Network Security. *Mathematics* **2023**, *11*, 4501. [CrossRef]

19. Ragab, M.; Alshammari, S.M.; Maghrabi, L.A.; Alsalman, D.; Althaqafi, T.; AL-Ghamdi, A.A.-M. Robust DDoS Attack Detection Using Piecewise Harris Hawks Optimizer with Deep Learning for a Secure Internet of Things Environment. *Mathematics* **2023**, *11*, 4448. [CrossRef]
20. Setitra, M.A.; Fan, M.; Agbley, B.L.Y.; Bensalem, Z.E.A. Optimized MLP-CNN Model to Enhance Detecting DDoS Attacks in SDN Environment. *Network* **2023**, *3*, 538–562. [CrossRef]
21. Adeniyi, O.; Sadiq, A.S.; Pillai, P.; Aljaidi, M.; Kaiwartya, O. Securing Mobile Edge Computing Using Hybrid Deep Learning Method. *Computers* **2024**, *13*, 25. [CrossRef]
22. Ramzan, M.; Shoaib, M.; Altaf, A.; Arshad, S.; Iqbal, F.; Castilla, A.K.; Ashraf, I. Distributed Denial of Service Attack Detection in Network Traffic Using Deep Learning Algorithm. *Sensors* **2023**, *23*, 8642. [CrossRef]
23. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In Proceedings of the International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–8 October 2019.
24. Talukder, M.A.; Uddin, M.A. CIC-DDoS2019 Dataset. 2023, Version 1. Available online: https://data.mendeley.com/datasets/ssnc74xm6r/1 (accessed on 5 January 2023).
25. Frye, M.; Mohren, J.; Schmitt, R.H. Benchmarking of Data Preprocessing Methods for Machine Learning-Applications in Production. *Procedia CIRP* **2021**, *104*, 50–55. [CrossRef]
26. Zhang, J.; Wang, Q.; Shen, W. Hyper-parameter optimization of multiple machine learning algorithms for molecular property prediction using hyperopt library. *Chin. J. Chem. Eng.* **2022**, *52*, 115–125. [CrossRef]
27. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégel, B. Algorithms for Hyper-Parameter Optimization. In Advances in Neural Information Processing Systems, Curran Associates. 2011. Available online: https://papers.nips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html (accessed on 11 January 2023).