International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(Research Article)

Feature-Driven Supervised Learning for Detecting DDoS Attack

Md Boktiar Hossain [1,*], Rashedur Rahman [2] and Khandoker Hoque [3]

[1] *Department of Information and Communication Engineering, University of Rajshahi, Rajshahi 6205, Bangladesh.*
[2] *Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh.*
[3] *Department of Electrical and Electronic Engineering, Brac University, Dhaka, Bangladesh.*

International Journal of Science and Research Archive, 2021, 04(01), 393-402

Publication history: Received on 21 October 2021; revised on 23 November 2021; accepted on 28 November 2021

Article DOI: https://doi.org/10.30574/ijsra.2021.4.1.0204

Abstract

Distributed Denial-of-Service (DDoS) attacks are intentional efforts to disrupt the normal traffic of a targeted server, network, or organization by overwhelming the victim or its neighboring systems with excessive network traffic. Detecting such attacks using machine-learning models is challenging due to significant variations in traffic patterns and rates. So, an automated detection approach is proposed, which reduces the feature space to minimize model overfitting and computational cost. The CICDDoS2019 dataset, including a wide range of DDoS attack scenarios, is used to train and evaluate the proposed method in a cloud-based environment. Relevant features are extracted using the Extra Trees classifier and then passed to Decision Tree, XGBoost, and Random Forest classifiers. XGBoost achieved the highest validation accuracy of 98.87% with feature selection, while Decision Tree maintained a strong baseline accuracy of 98.49% even without feature selection.

Keywords: CICDDoS2019 dataset; DDoS attacks; Data preprocessing; Feature selection; Classification

1. Introduction

The rapid advancement of network technologies and the widespread adoption of internet-enabled devices have significantly improved access to information and interpersonal communication. However, this increased connectivity has also introduced new vulnerabilities, attracting cybercriminals who exploit the availability of online services for malicious purposes. Despite the implementation of various cybersecurity policies to manage fraudulent activities, users and organizations continue to face serious challenges such as service outages, unauthorized access to sensitive data, and disruption of operations. Among the most prevalent and damaging cyber threats are Distributed Denial-of-Service (DDoS) attacks. In these attacks, adversaries exploit the typical behavior of internet-connected network devices, often targeting edge devices rather than specific servers. DDoS attacks aim to overwhelm network bandwidth or the infrastructure supplying it, rendering services inaccessible to legitimate users.

Several methodologies, including random simulation, decision theory, and game theory, have been explored to detect and mitigate such attacks [1]. However, these models often struggle to identify previously unknown or evolving attack patterns. To ensure effective defense, attacks must be detected and neutralized before reaching their intended targets [2].

DDoS attacks typically follow a two-phase approach. In the first phase, attackers build a botnet—a network of compromised devices under the control of a single entity known as a "bot-herder." In the second phase, this botnet is activated to flood the target network with malicious traffic. DDoS attacks are generally classified into two main types:

[*] Corresponding author: Md Boktiar Hossain

- Flooding attacks (volumetric attacks): These aim to saturate the target's bandwidth by generating massive amounts of traffic, thereby also exhausting system caches and resources.
- Application-layer attacks: These are more sophisticated and efficient, often requiring less bandwidth. They target specific applications or services, significantly affecting system performance and availability [3-6].

A hybrid feature selection strategy is proposed, which focuses on identifying the most relevant features to reduce training time and improve detection accuracy.

2. Related works

Freire M. M. et al. [7] proposed a DDoS detection approach that combines fuzzy logic and machine learning techniques to address quality degradation attacks. Four machine learning algorithms—Multilayer Perceptron (MLP), k-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Multinomial Naïve Bayes (MNB)—were evaluated for detecting Reduction of Quality (RoQ) attacks. A hybrid approach combining fuzzy logic, MLP, and Euclidean distance (ED) was also employed. While this fuzzy logic-based method achieved superior performance compared to MLP alone, it incurred a higher execution time. Among the evaluated algorithms, MLP yielded the best classification results for RoQ detection. The authors suggest further studies to compare this approach with additional machine learning techniques.

Wang M. et al. [8] introduced a dynamic MLP-based detection model incorporating sequential feature selection and feedback mechanisms to defend against DDoS attacks. This method effectively identified detection errors once their saliency exceeded a threshold and enabled model retraining using updated information. However, two key limitations were noted: (1) The SBS-MLP framework may fail to find globally optimal feature subsets; and (2) while the feedback system can correct false positives automatically, false negatives remain unaddressed, potentially affecting overall detection performance.

Cil A. E. et al. [9] proposed a deep neural network model based on a feed-forward architecture for DDoS detection. Although the model demonstrated effectiveness in detecting general DDoS attacks, it failed to identify more complex variants such as HTTP Flood Attacks. These attacks are particularly difficult to detect because they closely resemble legitimate traffic and specifically target application-layer components.

Shurman M. et al. [10] presented an intrusion detection framework using deep learning models to identify DoS and DDoS threats. The proposed system comprises two components: an IDS capable of monitoring suspicious network traffic across any node, and a deep learning model based on Long Short-Term Memory (LSTM), trained on multiple attack types from the CICDDoS2019 dataset. Future work aims to improve detection of other DDoS attack variants within the same dataset and validate the approach in real-world systems.

Jia Y. et al. [11] introduced FlowGuard, an IoT-oriented DDoS defense framework featuring two core components: a flow handler and a flow filter. The flow handler, leveraging LSTM and Convolutional Neural Network (CNN) models, is responsible for identifying and classifying malicious flows. The performance of these models was evaluated using both the CICDDoS2019 dataset and synthetically generated attack data, confirming their potential for accurate detection and classification.

Kshirsagar D. et al. [12] proposed a feature selection framework utilizing filter-based techniques and thresholding to enhance DDoS attack detection. The model was tested on the CICDDoS2019 dataset using the J48 decision tree classifier. Their feature reduction method achieved a reduction of original features by 56% to 82.92%, significantly improving efficiency. Future directions include exploring additional filter-based strategies to further optimize the feature set while maintaining strong detection performance.

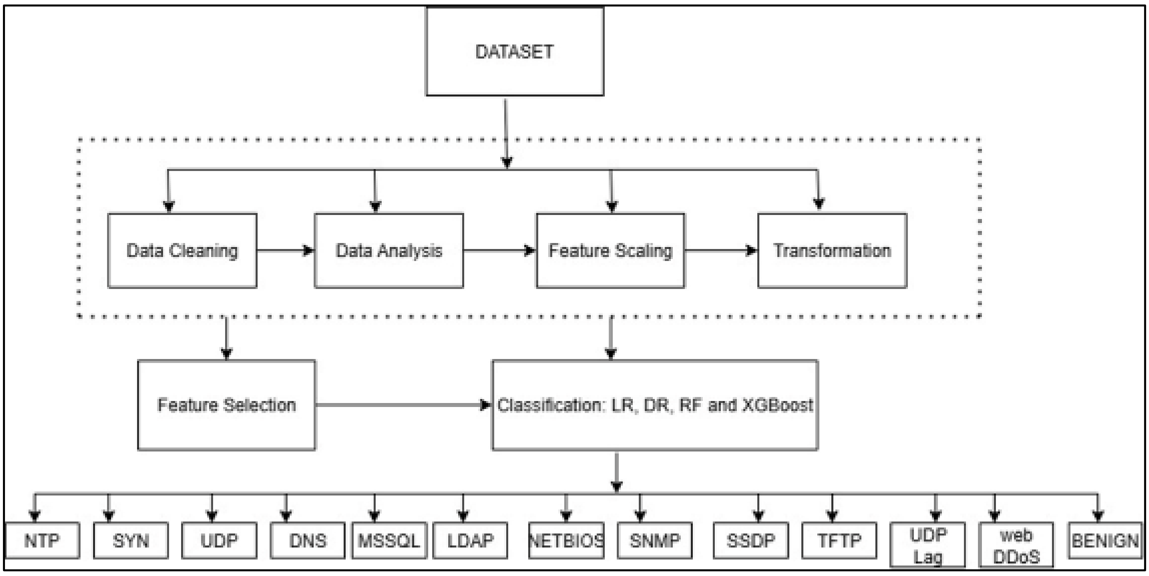

Figure 1 Flow diagram

3. Methodology

3.1. Techniques Used

The proposed model is structured into three main stages: pre-processing, feature selection, and classification. In the pre-processing phase, an exploratory analysis is conducted to better understand the dataset. Data cleansing is applied to improve the overall accuracy, followed by normalization of feature values using a standardized scale. Additionally, categorical variables are converted into numerical format to ensure compatibility with machine learning models. Next, in the feature selection stage, the most relevant features are extracted to enhance model performance. These selected features are then passed to various machine learning algorithms during the classification phase, where different types of attacks are identified. The complete workflow of the proposed approach is illustrated in Figure 1.

3.2. Pre-processing

Pre-processing is a fundamental step in any machine learning pipeline, aimed at preparing raw data for model development and training. The pre-processing phase includes:

- Data Analysis

Visual and statistical analyses are performed to understand dataset properties. Descriptive statistics (mean, standard deviation, quartiles, etc.) are calculated, and visualization tools such as histograms and box plots are used to identify outliers, feature correlations, and class imbalances.

- Data Cleaning
  - Elimination of Irrelevant Features: Features related to socket-level details are removed due to their high variability across networks, which could cause overfitting. After cleaning, 80 new features are retained for modeling.
  - Handling Missing and Noisy Data: Duplicates are removed to reduce computational overhead. Missing, infinite, and negative values are imputed using median values to preserve critical information for classification.
- Feature Scaling

To prevent scale-related training inefficiencies, numerical features are standardized using the StandardScaler, defined as:

$$S = \frac{s - \mu}{(SD)} \qquad (1)$$

Where S stands for standard Scaler, μ is the mean and SD is the standard deviation of the training sample.

- Categorical Encoding

Categorical attributes are transformed into numerical values using Label Encoding, where each category is assigned a unique integer starting from 0.

3.3. Feature Selection

After processing data, any machine learning method can use the pre-processed information of the data. The selection of features, which can be divided into three categories: filter, wrapper, and embedding, is critical for producing the best-performing model. A ranking and multivariate approach is used to select the top features, which removes features that are redundant, identical, or closely related. The filter method uses a univariate technique based on criteria to choose the independent subset of features. The machine learning model receives the selected features as input. Filter methods are less expensive. The wrapper employs a search strategy to identify potential feature subsets and uses machine learning to evaluate the selected feature subsets. The wrapper technique uses a search strategy to find suitable feature subsets and then analyses the chosen subsets of features using a machine learning approach. So, until the optimal features are obtained, this process is repeated. It is computationally demanding because of the wide variety of feature combinations it searches for. Embedded approaches manage feature selection and classification simultaneously, with the design of the machine learning algorithm handling feature selection. It chooses the best attributes according to how vital their derived features are.

- Extra Trees Classifier: It is an ensemble learning technique. To produce the classification result, it aggregates the classification results of multiple de-correlated decision trees gathered in a "forest." It is conceptually similar to a random forest classifier and differs only in how the decision trees in the forest are constructed. The extra tree forest's decision trees are built based on the original dataset. Then, at every test node, k random features are selected from the feature collection and given to the trees. A mathematical criterion must be used by every decision tree to determine which attribute is the optimal one for data splitting. This attribute selection leads to the creation of several de-correlated trees. Each attribute's standardized total decrease in the mathematical requirements for the split decision feature is calculated during the development of the forest. This mathematical criterion is known as the Gini Importance of the attribute. Each attribute is ranked in descending order as per Gini Importance, and the user chooses the best feature [9].

3.4. Classification

3.4.1. Decision Tree

Figure 2 illustrates how the decision tree's classification procedure works.

- Step 1: The root node (r), which has the entire dataset, is where the tree starts.
- Step 2: Use a selection attribute measure to find the optimal dividing attribute.
- Step 3: Subsets of the r should be created.
- Step 4: Generating the decision node with the best attribute is ideal.
- Step 5: Recursively make new decision trees based on the subsets of data in Step 3. Repeat this process until you reach the leaf node, which means that the node cannot be further divided.

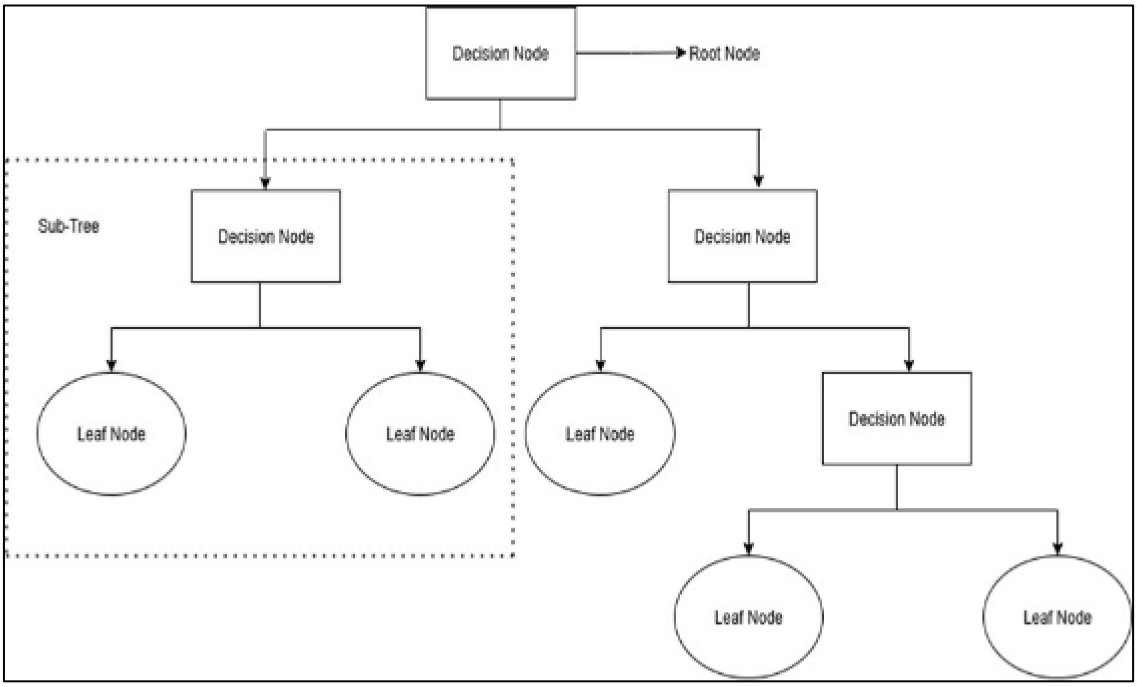

Figure 2 Decision Tree

3.4.2. Random Forest

It is a supervised machine learning algorithm used for both classification and regression problems.

It is based on the concept of ensemble learning. Instead of relying on a single decision tree, random forests predict the final result by taking a majority vote from all the trees. Increasing the number of trees generally improves accuracy and reduces the chance of overfitting. The following steps and Figure 3 explain the working process:

- Step 1: Randomly select K samples from the dataset.
- Step 2**:** Construct decision trees using the selected sample points.
- Step 3**:** Each tree provides a predicted outcome, and voting is conducted.
- Step 4**:** The outcome with the majority vote is chosen as the final result.
- Step 5**:** Evaluate the result.

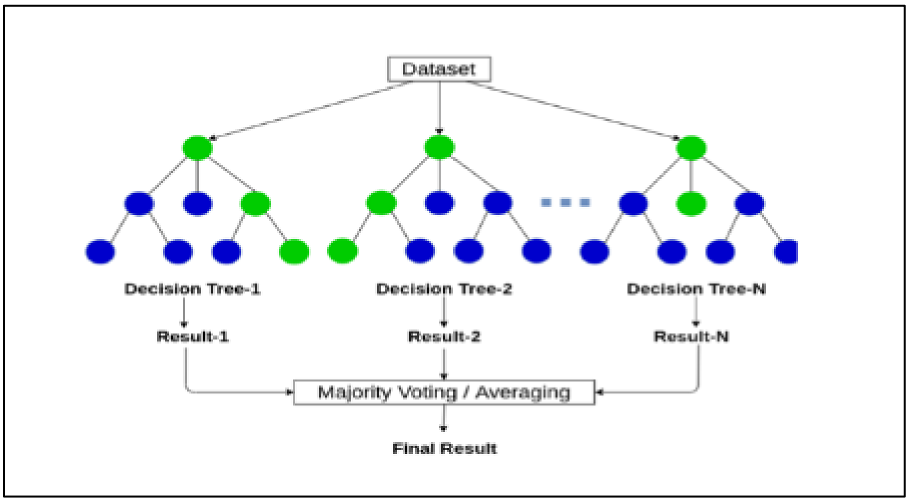

Figure 3 Random Forest

3.4.3. XGBoost

It builds decision trees sequentially, assigning initial weights to each feature. Misclassified instances from one tree are given higher weights before being passed to the next, thereby enhancing focus on harder-to-classify samples. The ensemble of these trees forms a robust predictive model [12-15]. XGBoost achieved the validation accuracy of 98.87% with feature selection. Table 1 displays the training and validation accuracy for models using feature selection, while Table 2 shows performance without it. Table 3-6 present the classification reports of all models. Among them, Random Forest consistently achieved the highest F1-scores across most DDoS classes, outperforming Decision Tree, XGBoost, and Logistic Regression. It demonstrated perfect detection of SYN attacks and strong performance for Benign traffic with a precision of 1.0 and recall of 0.1589. For most other attack types, the F1-score exceeded 0.95. While all models struggled with detecting WebDDoS, Random Forest showed superior overall detection capabilities.

3.4.4. Logistic Regression

It is a simple yet powerful linear classification algorithm that models the probability of a class belonging to a particular category using a logistic function. It is particularly valued for its efficiency, interpretability, and ability to perform well with high-dimensional data [17-22]. Despite its simplicity, Logistic Regression demonstrated strong performance in this study, achieving a weighted F1-score of 0.9880 and validation accuracy close to the tree-based models. It effectively classified most DDoS attack types, although it had limitations in detecting Benign traffic (recall: **0.0561**) and failed to identify WebDDoS attacks, similar to other models. While not as robust as Random Forest or XGBoost in handling complex non-linear patterns, its competitive scores highlight its usefulness as a lightweight baseline for DDoS detection tasks.

4. Dataset

The Canadian Cyber Security Institute's (CCI) CICDDoS2019 dataset [13] is employed to evaluate the performance of the proposed model. Compared to earlier datasets, CICDDoS2019 includes a wider variety of DDoS attack patterns and higher traffic volumes. This dataset primarily contains reflection and exploitation-based attacks.

It consists of 88 features collected over two days—one for training and one for testing. The testing data includes seven specific DDoS attacks: Port Map, SYN, UDP, MSSQL, UDP-Lag, LDAP, and NetBIOS. In contrast, the training data includes 12 DDoS attack types: NetBIOS, NTP, MSSQL, DNS, LDAP, SNMP, UDP-Lag, SSDP, UDP, SYN, TFTP, and WebDDoS. On the test day, only the Port Map attack was newly developed and observed.

5. Experimental setup

The proposed model is implemented in Python, using libraries such as pandas, NumPy, Scikit-learn, and LinearSVC. Experiments are conducted on a machine running Windows 10 with the following configuration:

- Processor: Intel Core i7-10750H @ 2.60 GHz
- RAM: 16 GB
- GPU: NVIDIA GeForce GTX 1650 Ti (4 GB)

The Day 1 subset of the dataset is used for both training and testing, ensuring a balanced evaluation of the model.

6. Results

This section presents the performance evaluation of the proposed approach using Decision Tree, Random Forest, and XGBoost, with and without feature selection.

- Validation Accuracy:
    - XGBoost with feature selection achieved the highest validation accuracy of 98.87%.
    - Decision Tree without feature selection reached 98.49%, indicating strong baseline performance.
- Performance Tables:
    - Table 1 presents model accuracy with feature selection.
    - Table 2 presents model accuracy without feature selection.
- Classification Reports: Shown in table 3-6, these highlight precision, recall, and F1-scores for each classifier:
- Random Forest achieved the highest F1-score overall, accurately detecting most attacks except WebDDoS.

Table 1 Accuracy with feature selection

Model	Training Accuracy	Validation Accuracy
Decision Tree	100%	99.07%
Random Forest	100%	99.09%
XGBoost	100%	98.87%
Logistic Reg	100%	98.88%

Table 2 Accuracy without feature selection

Model	Training Accuracy	Validation Accuracy
Decision Tree	100%	98.49%
Random Forest	99%	93.69%
XGBoost	94%	87.10%
Logistic Reg	98%	94.65%

Table 3 Logistic Regression

Class	Precision	Recall	F1-Score	Support
BENIGN	1.0000	0.0561	0.1062	214
DrDoS_DNS	0.9755	0.9800	0.9777	14,233
DrDoS_LDAP	0.9712	0.9900	0.9805	14,774
DrDoS_MSSQL	1.0000	0.9899	0.9949	14,404
DrDoS_NTP	1.0000	0.9699	0.9847	9,140
DrDoS_NetBIOS	1.0000	0.9900	0.9950	14,664
DrDoS_SNMP	0.9720	0.9900	0.9809	14,791
DrDoS_SSDP	1.0000	1.0000	1.0000	14,795
DrDoS_UDP	0.9867	1.0000	0.9933	14,648
Syn	1.0000	1.0000	1.0000	10,371
UDP-lag	1.0000	0.9800	0.9899	2,495
WebDDoS	-	-	-	4
accuracy			0.9888	124,533
macro avg	0.9088	0.8288	0.8336	124,533
weighted avg	0.9889	0.9887	0.9880	124,533

Table 4 Decision Tree

Class	Precision	Recall	F1-Score	Support
BENIGN	1.0000	0.1495	0.2602	214
DrDoS_DNS	0.9771	0.9900	0.9835	14,233
DrDoS_LDAP	0.9804	0.9900	0.9852	14,774
DrDoS_MSSQL	1.0000	0.9899	0.9949	14,404
DrDoS_NTP	1.0000	0.9699	0.9847	9,140
DrDoS_NetBIOS	1.0000	0.9900	0.9950	14,664
DrDoS_SNMP	0.9721	0.9930	0.9824	14,791
DrDoS_SSDP	1.0000	1.0000	1.0000	14,795
DrDoS_UDP	0.9921	1.0000	0.9960	14,648
Syn	1.0000	1.0000	1.0000	10,371
UDP-lag	1.0000	0.9948	0.9974	2,495
WebDDoS	0.0000	0.0000	0.0000	4
accuracy			0.9907	124533
macro avg	0.9101	0.8389	0.8483	124533
weighted avg	0.9908	0.9907	0.9901	124533

Table 5 Random Forest

Class	Precision	Recall	F1-Score	Support
BENIGN	1.0000	0.1589	0.2742	214
DrDoS_DNS	0.9773	0.9900	0.9836	14,233
DrDoS_LDAP	0.9804	0.9900	0.9852	14,774
DrDoS_MSSQL	1.0000	0.9899	0.9949	14,404
DrDoS_NTP	1.0000	0.9720	0.9858	9,140
DrDoS_NetBIOS	1.0000	0.9900	0.9950	14,664
DrDoS_SNMP	0.9733	0.9930	0.9830	14,791
DrDoS_SSDP	1.0000	1.0000	1.0000	14,795
DrDoS_UDP	0.9924	1.0000	0.9962	14,648
Syn	1.0000	1.0000	1.0000	10,371
UDP-lag	1.0000	0.9968	0.9984	2,495
WebDDoS	0.0000	0.0000	0.0000	4
accuracy			0.9909	124533
macro avg	0.9103	0.8400	0.8497	124533
weighted avg	0.9910	0.9909	0.9904	124533

Table 6 XGBoost

Class	Precision	Recall	F1-Score	Support
BENIGN	1.0000	0.1589	0.2742	214
DrDoS_DNS	0.9773	0.9900	0.9836	14,233
DrDoS_LDAP	0.9804	0.9900	0.9852	14,774
DrDoS_MSSQL	1.0000	0.9899	0.9949	14,404
DrDoS_NTP	1.0000	0.9720	0.9858	9,140
DrDoS_NetBIOS	1.0000	0.9900	0.9950	14,664
DrDoS_SNMP	0.9733	0.9930	0.9830	14,791
DrDoS_SSDP	1.0000	1.0000	1.0000	14,795
DrDoS_UDP	0.9924	1.0000	0.9962	14,648
Syn	1.0000	1.0000	1.0000	10,371
UDP-lag	1.0000	0.9968	0.9984	2,495
WebDDoS	0.0000	0.0000	0.0000	4
BENIGN			0.9909	124533
DrDoS_DNS	0.9103	0.8400	0.8497	124533
DrDoS_LDAP	0.9910	0.9909	0.9904	124533
accuracy			0.9887	124533
macro avg	0.9088	0.8300	0.8357	124533
weighted avg	0.9889	0.9887	0.9881	124533

7. Conclusion

This study evaluated the performance of four supervised machine learning classifiers—Decision Tree, Random Forest, XGBoost, and Logistic Regression—for detecting DDoS attacks using the CICDDoS2019 dataset. The workflow involved data preprocessing, feature selection, and classification. XGBoost achieved the highest validation accuracy of 98.87% with feature selection, while Decision Tree maintained a strong baseline accuracy of 98.49% even without feature selection. Random Forest outperformed all other models in terms of overall F1-score and class-wise precision and recall, particularly excelling in detecting SYN and Benign traffic. Although all models failed to identify WebDDoS attacks, they consistently achieved F1-scores above 0.95 across major DDoS types. These findings confirm that ensemble methods, particularly Random Forest and XGBoost, are well-suited for high-precision DDoS detection. Future work may explore deep learning models and hybrid approaches to improve detection of rare or evolving threats like WebDDoS, and incorporate blockchain for secure blacklisting of malicious sources.

References

[1] M. Kim, “Supervised learning-based DDoS attacks detection: Tuning hyperparameters,” *ETRI Journal*, vol. 41, no. 5, pp. 560–573, 2019.

[2] R. K. Batchu and H. Seetha, “A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning,” *Computer Networks*, vol. 200, p. 108498, 2021.

[3] P. S. Saini, S. Behal and S. Bhatia, "Detection of DDoS Attacks using Machine Learning Algorithms," 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2020, pp. 16-21, doi: 10.23919/INDIACom49435.2020.9083716.

[4] G. Kaur and P. Gupta, "Hybrid Approach for detecting DDOS Attacks in Software Defined Networks," 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 2019, pp. 1-6, doi: 10.1109/IC3.2019.8844944

[5] A. R. Wani, Q. P. Rana, U. Saxena and N. Pandey, "Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques," 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 870-875, doi: 10.1109/AICAI.2019.8701238

[6] J. A. Pérez-Díaz, I. A. Valdovinos, K. -K. R. Choo and D. Zhu, "A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning," in IEEE Access, vol. 8, pp. 155859-155872, 2020, doi: 10.1109/ACCESS.2020.3019330

[7] V. de Miranda Rios, P. R. Inácio, D. Magoni, and M. M. Freire, “Detection of reduction-of-quality DDoS attacks using fuzzy logic and machine learning algorithms,” *Computer Networks*, vol. 186, p. 107792, 2021.

[8] M. Wang, Y. Lu, and J. Qin, “A dynamic MLP-based DDoS attack detection method using feature selection and feedback,” *Computers & Security*, vol. 88, p. 101645, 2020.

[9] A. E. Cil, K. Yildiz, and A. Buldu, “Detection of DDoS attacks with feed-forward based deep neural network model,” *Expert Systems with Applications*, vol. 169, p. 114520, 2021.

[10] M. M. Shurman, R. M. Khrais, A. A. Yateem, et al., “DoS and DDoS attack detection using deep learning and IDS,” *International Arab Journal of Information Technology*, vol. 17, no. 4A, pp. 655–661, 2020.

[11] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, “FlowGuard: An intelligent edge defense mechanism against IoT DDoS attacks,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9552–9562, 2020.

[12] D. Kshirsagar and S. Kumar, “A feature reduction-based reflected and exploited DDoS attacks detection system,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 393–405, 2022.

[13] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[14] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, “Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy,” in *Proc. Int. Carnahan Conf. Security Technology (ICCST)*, 2019, pp. 1–8. IEEE.

[15] Podder, Prajoy, Aditya Khamparia, M. Rubaiyat Hossain Mondal, Mohammad Atikur Rahman, and Subrato Bharati. "Forecasting the Spread of COVID-19 and ICU Requirements." iJOE 17, no. 05 (2021): 81.

[16] S. Bharati, M. A. Rahman and P. Podder, "Breast Cancer Prediction Applying Different Classification Algorithm with Comparative Analysis using WEKA," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, 2018, pp. 581-584, doi: 10.1109/CEEICT.2018.8628084.

[17] Azam, S., Huda, A. F., Shams, K., Ansari, P., Hasan, M. M., & Mohamed, M. K. (2015). Anti-inflammatory and anti-oxidant study of ethanolic extract of Mimosa pudica. Journal of Young Pharmacists, 7(3), 234.

[18] Mondal, K. K. (2015). Potential investigation of anti-inflammatory activity and phytochemical investigations of ethanolic extract of Glycosmis pentaphylla leaves. American Journal of Biomedical Research, 3(1), 6-8.

[19] Rahaman, Md Zahedur, Shammi Akhter, Md Rakibul Islam, Suriya Begum, Kallol Kanti Mondal, Md Mottakin, Md Shahadat Hossain, Sulata Bayen, and Munny Das. "Assessment of thrombolytic, antioxidant and analgesic properties of a medicinal plant of Asteraceae family growing in Bangladesh." Discovery Phytomedicine 7, no. 1 (2020): 47-52.

[20] Ahammed, Md Salim, Md Mottakin, Md Kayes Mahmud, Mst Monira Khanom, Md Eleas Kobir, Shammi Akhter, Md Shahadat Hossain et al. "A Study on Hevea Brasiliensis for evaluation of phytochemical and pharmacological properties in Swiss Albino Mice." Discovery Phytomedicine 7, no. 2 (2020): 72-75.

[21] Ansari, P., Shofiul, A. J., Sumonto, S., Kallol, K. M., Tasnim, T., & Sanjeeda, S. B. (2015). Potential investigation of anti-inflammatory and anti-oxidative properties of ethanolic extract of Ixora nigricans leaves. IJPR, 5(4), 104.

[22] Bharati, S., Robel, M.R.A., Rahman, M.A., Podder, P., Gandhi, N. (2021). Comparative Performance Exploration and Prediction of Fibrosis, Malign Lymph, Metastases, Normal Lymphogram Using Machine Learning Method. In: Abraham, A., Panda, M., Pradhan, S., Garcia-Hernandez, L., Ma, K. (eds) Innovations in Bio-Inspired Computing and Applications. IBICA 2019. Advances in Intelligent Systems and Computing, vol 1180. Springer, Cham. https://doi.org/10.1007/978-3-030-49339-4_8