

Cross-Framework Evaluation of DDoS Attack Detection: Assessing Machine Learning Classifiers and Feature Selection in Weka and Orange3

Waleed Abdullah Haroun Miftah ^{1*}, Sultan M. Buzid Alqaddafi ²

^{1,2} Department of Computer Science, Faculty of Science, University of Derna, Derna, Libya

**تقييم متقاطع للأنظمة في كشف هجمات حجب الخدمة الموزعة (DDoS): تقييم مصنفات تعلم الآلة
واختيار الميزات في برمجيات Weka و Orange3**

وليد عبد الله هارون مفتاح ^{1*}، سلطان القذافي ²
^{2,1} قسم علوم الحاسوب، كلية العلوم، جامعة درنة، درنة، ليبيا

*Corresponding author: Walidharoon79@gmail.com

Received: June 26, 2026

Accepted: September 15, 2026

Published: September 26, 2026

Abstract:

Predicting distributed denial of service (DDoS) attacks is a complex task due to the similarity between the attack patterns and normal traffic. The challenge lies in finding the best algorithm out of 10 common machine learning algorithms. Many previous studies exhibit a notable limitation: they often rely on a single dataset and a single experimental tool (e.g., Weka), without conducting a comprehensive comparison under unified settings. This restricts the ability to identify the most efficient algorithms in terms of accuracy and resource utilization. This study compares ten machine learning algorithms implemented in Weka and Orange3, including both shared and platform-specific ones, with and without enhancement techniques. The goal is to identify the most effective algorithm overall and to determine which platform performs better for algorithms available in both. The improved versions of the CICDDoS2019 and CICDoS2017 datasets were used for evaluation. The datasets were split into 70% & 30% and 80% & 20% ratios for training and testing, respectively. The results indicated that the J48 algorithm on CICDDoS2019, achieved the highest accuracy without using any features selection algorithm: 99.95% and took 258.14 seconds for training and 0.37 seconds for testing, while same algorithm with Information gain ratio on CICDoS2017, achieved highest accuracy: 99.3% and took 20.58 seconds for training and 0.12 seconds for testing. The contribution of this research lies in increasing the comprehensiveness of the targeted algorithms and techniques and use 2 tools within a single study, which provided a more objective comparison than previous research.

Keywords: Comparative Study, Cross-Platform Evaluation, DDoS Attack Detection, Decision Tree Classifiers, Feature Selection, Machine Learning, Network Security.

المخلص

يُعد التنبؤ بهجمات حجب الخدمة الموزعة (DDoS) مهمة معقدة نظراً للتشابه الكبير بين أنماط الهجوم وحركة المرور الطبيعية للشبكة. وتكمن الصعوبة في إيجاد الخوارزمية الأفضل من بين 10 خوارزميات شائعة لتعلم الآلة. تُظهر العديد من الدراسات السابقة قصوراً ملحوظاً، حيث تعتمد غالباً على قاعدة بيانات واحدة وأداة تجريبية واحدة (مثل: Weka)، دون إجراء مقارنة شاملة تحت ظروف إعدادات موحدة. وهذا يحد من القدرة على تحديد الخوارزميات الأكثر كفاءة من حيث الدقة واستهلاك الموارد. تقارن هذه الدراسة بين عشر خوارزميات لتعلم الآلة تم تطبيقها في برمجيات Weka و Orange3، بما في ذلك الخوارزميات المشتركة وتلك الخاصة بكل منصة، مع استخدام تقنيات التحسين وبدونها. ويتمثل الهدف في تحديد الخوارزمية الأكثر فعالية بشكل عام، وتحديد النظام الأساسي الذي يقدم أداءً أفضل للخوارزميات المتاحة في كلتا المنصتين. تم استخدام النسخ المحسنة من قواعد البيانات المعيارية CICDDoS2019 و CICDoS2017 لأغراض التقييم. وقُسمت قواعد البيانات بنسب (70% للتدريب و 30% للاختبار) ونسب (80% للتدريب و 20% للاختبار) على التوالي. أشارت النتائج إلى أن خوارزمية J48 حققت أعلى دقة على قاعدة بيانات CICDDoS2019،

دون استخدام أي خوارزمية لاختيار الميزات، بنسبة بلغت 99.95%، واستغرقت 258.14 ثانية للتدريب و 0.37 ثانية للاختبار. في المقابل، حققت الخوارزمية نفسها باستخدام تقنية "نسبة كسب المعلومات" (Information gain ratio) أعلى دقة على قاعدة بيانات CICDoS2017 بنسبة بلغت 99.3%، واستغرقت 20.58 ثانية للتدريب و 0.12 ثانية للاختبار. تكمن المساهمة العلمية لهذا البحث في تعزيز شمولية الخوارزميات والتقنيات المستهدفة واستخدام أداتين برمجيتين داخل دراسة واحدة، مما وفر مقارنة أكثر موضوعية مقارنة بالبحوث السابقة.

الكلمات المفتاحية: دراسة مقارنة، تقييم مقاطع للأنظمة، كشف هجمات DDoS، مصنفات شجرة القرار، اختيار الميزات، تعلم الآلة، أمن الشبكات.

1 INTRODUCTION

Cybersecurity has become a fundamental concern in the modern digital era due to the rapid growth of interconnected systems, cloud computing, and Internet-based services. As organizations increasingly rely on continuous online availability, computer networks are exposed to a wide range of cyber threats. Among these threats, DDoS attacks are considered one of the most disruptive and economically damaging forms of cyberattacks. A DDoS attack is characterized by the use of multiple compromised devices, typically forming a botnet, to overwhelm targeted network resources. These attacks can target different layers of the network architecture. For example, at the application layer, attackers generate a high volume of HTTP requests (e.g., GET and POST), exhausting server resources and preventing legitimate users from accessing services. This leads to significant degradation in system performance and availability. The origins of Denial of Service (DoS) attacks can be traced back to early incidents such as the 1996 Panix outage, which was caused by a single-source attack. Over time, these attacks have evolved into large-scale distributed attacks involving thousands of geographically dispersed devices. This evolution has significantly increased the complexity of detection and mitigation mechanisms. The impact of DDoS attacks extends beyond technical disruption. Organizations may experience substantial financial losses due to downtime, reputational damage, and loss of customer trust. Sectors that heavily depend on online services, such as banking, e-commerce, and government platforms, are particularly vulnerable. Traditional security mechanisms, including firewalls and signature-based intrusion detection systems, often struggle to cope with the scale and sophistication of modern DDoS attacks. To overcome these limitations, machine learning (ML) techniques have emerged as promising solutions for DDoS detection. ML-based approaches enable automated analysis of network traffic patterns and can effectively distinguish between normal and malicious activities. Their ability to adapt to evolving attack patterns makes them suitable for real-time detection in dynamic environments. Despite the growing body of research in this field, there remains a lack of comprehensive comparative studies evaluating supervised machine learning algorithms across multiple datasets and analytical frameworks. Most existing studies focus on a limited set of algorithms or a single experimental platform, which restricts the generalizability of their findings.

This study addresses this gap by conducting a systematic comparison of several supervised machine learning algorithms and preprocessing techniques using two widely adopted data mining frameworks: Weka and Orange3. Evaluating algorithms across different platforms enhances the robustness and credibility of the results. Furthermore, this study analyzes both predictive performance and computational efficiency to provide a holistic evaluation. The main contributions of this paper are summarized as follows:

- A comprehensive evaluation of supervised machine learning algorithms for DDoS detection.
- An analysis of the impact of preprocessing techniques on detection performance.
- A comparative study between Weka and Orange3 frameworks.
- Insights into the trade-offs between accuracy and computational complexity.

The results of this study provide practical guidance for researchers and practitioners in selecting suitable machine learning models for DDoS detection, thereby improving system resilience and reducing operational risks.

2 RELATED WORKS

Previous studies are extremely important, as they enable the researchers to review developments in their chosen topic. They can identify flaws or gaps upon which research can be built. The theoretical background is important, enabling anyone reading this research to Understanding the topic of the thesis and grasp the problem under study. Below are some of the most important recent studies: Recent studies have extensively explored advanced machine learning and deep learning methods for detecting DDoS attacks.

For instance, Alashhab et al. [1] introduced an ensemble online learning model aimed at strengthening DDoS detection and mitigation. Their model adapts dynamically to evolving attack behaviors and was tested in a Software Defined Network (SDN) environment using Mininet and Ryu. By employing CICIDS2019, InSDN, and a custom slow-read DDoS dataset, they demonstrated that a combination of BernoulliNB, Passive-Aggressive, Stochastic Gradient Descent (SGD), and Multi-Layer Perception (MLP) classifiers could achieve detection and legitimacy rates above 99%.

Similarly, AlSaleh et al. [2] proposed the BaysCNN model, a Bayesian-based convolutional neural network for cloud-oriented DDoS detection. With Principal Component Analysis (PCA)-based dimensionality reduction applied on the CICDDoS2019 dataset, the model reached 99.66% accuracy across 13 multiclass attack scenarios. Their extended framework, BaysFusCNN, integrated Bayesian uncertainty estimation with feature fusion, further improving accuracy to 99.79%.

Amarnath et al. [3] concentrated on detecting DDoS attacks using the NSL-KDD dataset by applying Learning Vector Quantization (LVQ) and PCA. Their findings indicated that Decision Tree (DT) performed best with an accuracy of 98.74% when combined with LVQ.

In another study, Dembele et al. [4] leveraged a Binary Grey Wolf Optimizer for feature selection in conjunction with a support vector machine (SVM) classifier on the InSDN dataset. Their approach achieved outstanding outcomes, reporting up to 100% accuracy and strong F1-scores depending on the optimization equation used.

Hasan et al. [5] explored IoT environments, employing the CICIOT2023 dataset alongside SMOTE balancing techniques and Random Forest (RF) and PCA for feature selection. Their results revealed that DT attained a near-perfect accuracy of 99.97%.

Likewise, KIKISSAGBE et al. [6] proposed a multi-step framework combining class balancing methods with feature selection and optimized classifiers on the Edge-IIoT dataset. Among various tested configurations, the combination of SMOTE with deep neural network (DNN) yielded the highest accuracy of 99.62%.

Naïem et al. [7] enhanced Gaussian Naïve Bayes (GNB) using the CICD2018 dataset and SMOTE, reporting improvements in accuracy and precision, while RF ultimately achieved the best result with 98.98%.

Similarly, Pandey and Mishra [8] investigated ensemble sampling with ADASYN and random undersampling on CICDDoS2019, applying Pearson correlation and Extra Trees for feature selection. Their results showed RF achieving 99% accuracy.

Other research emphasized hybrid models. Manisankar et al. [9] constructed datasets using CICFlowMeter and developed a hybrid RF and Gradient Boosting (GB) approach, which achieved 97% accuracy after feature selection with Chi-Square (CS) and f_{classif} .

Other studies addressed practical concerns in real-world networks. Sharif et al. [10] investigated the availability of DDoS tools and their impact, using CICIDS2017 with DT feature selection, where MLP with Adam optimizer reached 99.2% accuracy.

Şimşek and Atılgan [11] designed a custom dataset to assess IoT security, applying Information Gain and showing that DT achieved 99% accuracy.

Swami et al. [12] integrated rule-based and machine learning methods into an intrusion detection system, validated on the WSN-DS dataset with Information Gain and Gini Index, where DT reached 99.43% accuracy.

Valikhanli [13] addressed DDoS detection in UAV networks through an AI-driven model, employing CS, ANOVA F-value, and RF Importance (RFI) for feature selection on a custom dataset. The combination of RFI and DT yielded an accuracy of 99.51%.

Hirsi et al. [14] developed an ensemble classifier for low-rate DDoS detection in SDNs using CICDoS2017, where combining RF, K Nearest Neighbors (KNN), and SVM resulted in 98% accuracy.

Finally, Lee [15] examined dataset reduction strategies by combining CICIDS2017 and CICDDoS2019. Using hierarchical clustering and the J48 algorithm, they demonstrated superior results in binary classification with 99.9606% accuracy.

In summary, these studies provided valuable information about the research and the methods that can be used to solve the common problem among them, and the limitations therein. A review of prior studies on DDoS attack detection reveals several limitations. Many existing works rely on a single dataset—typically CICDoS2017 or CICDDoS2019—which restricts the generalizability of their findings. Moreover, several studies evaluate only a limited set of machine learning algorithms or use a single experimental platform (e.g., Weka or scikit-multiflow). To address these gaps, the present research combines two benchmark datasets (CICDoS2017 and CICDDoS2019), applies a broader range of supervised algorithms, and evaluates their performance within two distinct experimental frameworks (Weka and Orange3). This dual-platform approach provides a more comprehensive perspective on algorithmic behavior and offers additional insights into the effect of preprocessing techniques on DDoS detection performance.

3. MATERIALS & METHODS

In this section, the authors will describe the used datasets and how to properly prepare it before the training and testing process. The authors will also explain the algorithms targeted in this research. Finally, the authors will explain the evaluation criteria that will be used to evaluate the performance of algorithms to determine the best algorithm.

Research framework

The model used in this research is summarized in following: First selects the dataset to be used, selects the columns, and then applies an evaluation method (70% training and 30% testing, and 80% training and 20% testing). This results in identifying the best algorithm and the superior algorithms for each criterion. Figure (3.1) presents the model flow chart of the research methodology.

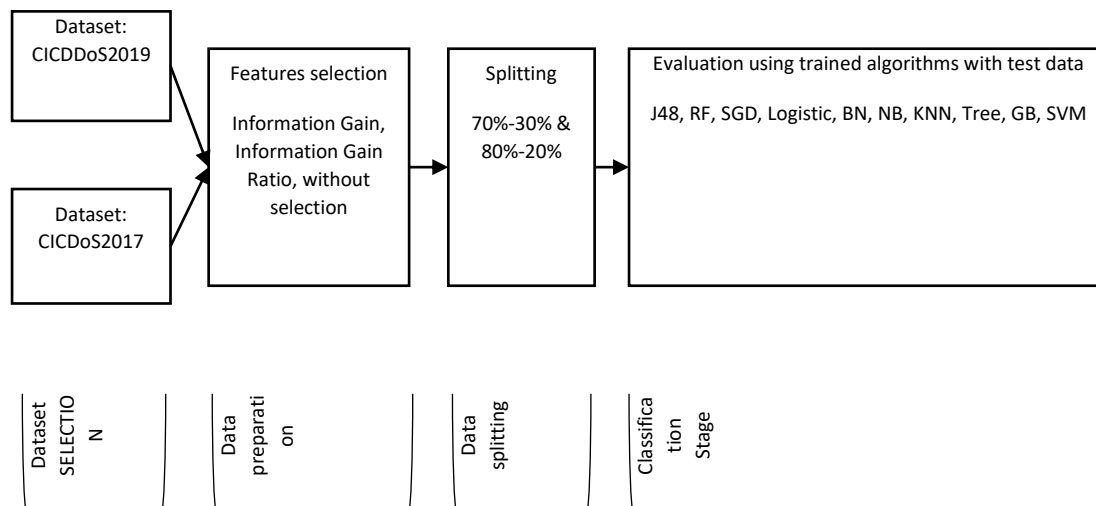


Figure 1. Model flow chart.

Datasets selection

This research employed the CICDDoS2019 dataset (79 features and 402208 instances) and CICDoS2017 dataset (78 features and 222914 instances), CICDDoS2019 dataset originally introduced by Sharafaldin et al. [16] at the Canadian Institute for Cybersecurity. The version used in this study was obtained from the publicly available repository on Kaggle [17], which provides a cleaned and reorganized edition of the dataset. Compared to the raw release, this version includes corrections for extreme class imbalance (reduced from ~4000:1 to ~9:1 attack-to-benign ratio), removal of duplicate and missing records, and conversion into optimized Parquet format for faster processing [17]. The CICDoS2017 dataset used in this study was obtained from the publicly available repository [18], which is a republished version of the original dataset created by Jazi et al. [19]. This study relies on Version 3 (V3) of the dataset, which includes preprocessing improvements such as data type corrections, removal of missing values and duplicates. The core structure of the dataset, including all 78 features and records, remains unchanged from the original release [18]. Since DoS and DDoS attacks share similar underlying patterns differing mainly in the number of sources involved successfully predicting a DoS attack can provide insights into detecting DDoS attacks as well. The datasets documentations do not disclose the preprocessing methods or the initial size of the raw data. Therefore, it is impossible to determine how records were filtered or reduced before publication. This presents a significant limitation when interpreting

comparative results. The following figure illustrates the main types more clearly it from Canadian Institute for Cybersecurity [20]:

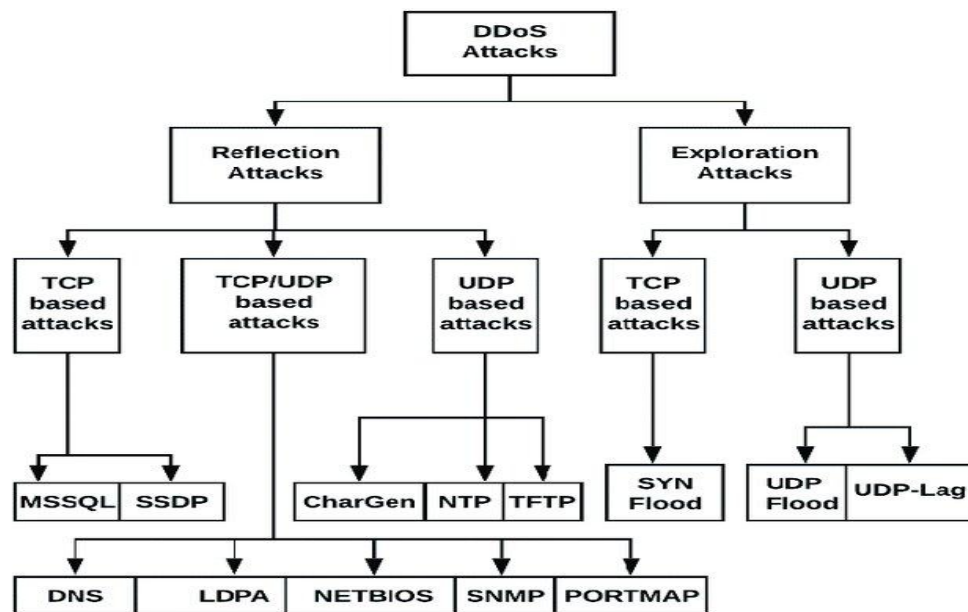


Figure 2. DDoS taxonomy in CICDDoS2019.

The types of DDoS attacks found in the CICDoS2017 dataset are shown in the following figure:

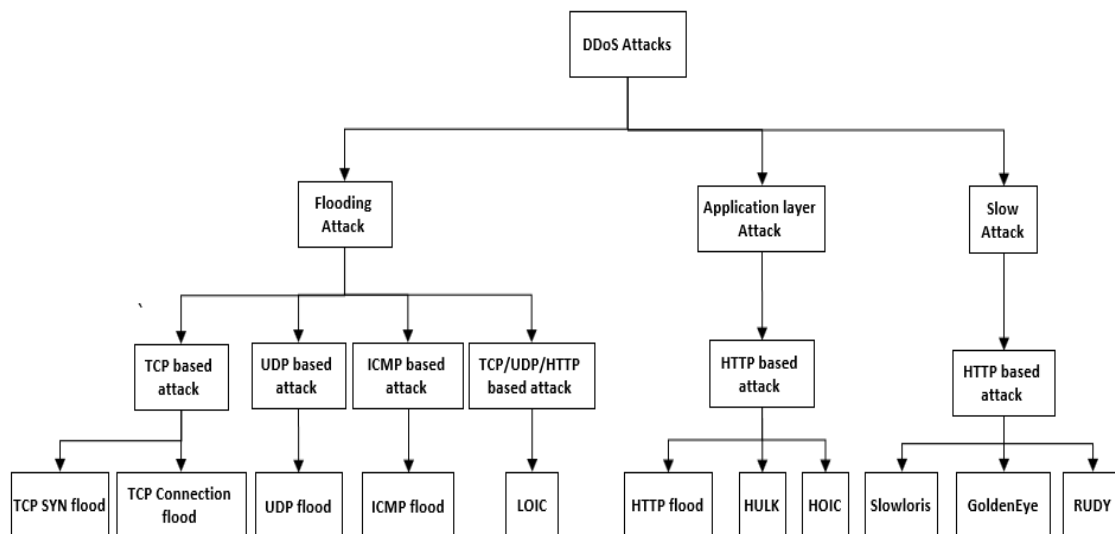


Figure 3. DDoS taxonomy in CICDoS2017.

Both datasets share the same columns in the same order, except for the Class, which exists only in CICDDoS2019. The remaining columns are common to both datasets, and they are fully described in the table below.

Table 1. Columns of CICDDoS2019 and CICDoS2017 (features).

No.	Column Name
1	Protocol
2	Flow Duration
3	Total Fwd Packets
4	Total Backward Packets
5	Fwd Packets Length Total
6	Bwd Packets Length Total
7	Fwd Packet Length Max
8	Fwd Packet Length Min
9	Fwd Packet Length Mean
10	Fwd Packet Length Std
11	Bwd Packet Length Max
12	Bwd Packet Length Min
13	Bwd Packet Length Mean
14	Bwd Packet Length Std
15	Flow Bytes/s
16	Flow Packets/s
17	Flow IAT Mean
18	Flow IAT Std
19	Flow IAT Max
20	Flow IAT Min
21	Fwd IAT Total
22	Fwd IAT Mean
23	Fwd IAT Std
24	Fwd IAT Max
25	Fwd IAT Min
26	Bwd IAT Total
27	Bwd IAT Mean
28	Bwd IAT Std
29	Bwd IAT Max
30	Bwd IAT Min
31	Fwd PSH Flags
32	Bwd PSH Flags
33	Fwd URG Flags
34	Bwd URG Flags
35	Fwd Header Length
36	Bwd Header Length
37	Fwd Packets/s
38	Bwd Packets/s
39	Packet Length Min
40	Packet Length Max
41	Packet Length Mean
42	Packet Length Std
43	Packet Length Variance
44	FIN Flag Count
45	SYN Flag Count
46	RST Flag Count
47	PSH Flag Count
48	ACK Flag Count
49	URG Flag Count
50	CWE Flag Count
51	ECE Flag Count
52	Down/Up Ratio
53	Avg Packet Size
54	Avg Fwd Segment Size

No.	Column Name
55	Avg Bwd Segment Size
56	Fwd Avg Bytes/Bulk
57	Fwd Avg Packets/Bulk
58	Fwd Avg Bulk Rate
59	Bwd Avg Bytes/Bulk
60	Bwd Avg Packets/Bulk
61	Bwd Avg Bulk Rate
62	Subflow Fwd Packets
63	Subflow Fwd Bytes
64	Subflow Bwd Packets
65	Subflow Bwd Bytes
66	Init Fwd Win Bytes
67	Init Bwd Win Bytes
68	Fwd Act Data Packets
69	Fwd Seg Size Min
70	Active Mean
71	Active Std
72	Active Max
73	Active Min
74	Idle Mean
75	Idle Std
76	Idle Max
77	Idle Min
78	Label
79	Class (only in CICDDoS2019)

The CICDDoS2019 dataset contains two classification columns:

- The Label column, which specifies the detailed attack type (e.g., UDP or specific category).
- The Class column, which distinguishes between Benign and DDoS.

Since this study does not focus on identifying the exact type of attack, the Class column was excluded, and the values of the Label column in both datasets were reformulated into two categories:

- Benign.
- DDoS.

This adjustment ensured full consistency in column naming and structure between both datasets (CICDDoS2019 and CICDoS2017), thereby facilitating fair comparison and simplifying the prediction task by focusing on the fundamental distinction between normal traffic and malicious activity. Furthermore, reframing the problem as a binary classification task aligns with the inherent design of several of the algorithms used in this study—such as SVM and Logistic—which were originally developed for binary classification and typically exhibit stronger performance in such scenarios. This contributes to a clearer analysis and more reliable comparison of algorithmic performance.

3.1 Data preparation

This section covers the initial data preparation that will be applied during the training and testing phases of the algorithms. In some experiments, no initial data preparation will be applied to the data as a baseline.

3.3.1. Feature selection algorithms

Feature selection algorithms are column evaluation methods in which the researcher specifies the number of top-performing columns to be selected. In this study, two Feature selection approaches will be utilized: Information Gain, Information Gain Ratio, and a baseline without any Feature selection. For each experiment, the selected top columns and their corresponding counts, based on the applied Feature selection method, will be presented

- Information gain

It evaluates each column based on the randomness of the data in it. That is, when a column contains, for example, 3 values that are repeated in the records, and each value is specific to only one class, and there is no other value associated with that class in the Target column, then that column will get the ideal value (1). Otherwise, it will get a percentage ranging between 0 and 1. Information Gain looks at how the entropy changes **before and after** splitting the data based on a feature:

$$E = - \sum_{i=1}^c p_i \log_2 p_i$$

$$\text{Information Gain} = \text{Entropy before split} - \text{Weighted entropy after split}$$

Where:

- *Entropy before split* is the calculation of randomness using the entire dataset.
- *Weighted entropy after split* is the calculation of randomness on the part to be evaluated in which the product of the weight of each value and its randomness in the records of the part to be evaluated is found.
- C is the number of distinct values in the column to be evaluated.
- p_i is the proportion of the value that belong to index I in the column to be evaluated, index I is the index of the value in the column to be evaluated [21].

- Information gain ratio

It improves Information Gain estimation when some columns contain a large number of values and its benefit is that it gives the largest values for the columns that we can use to generalize to new data. **Information Gain Ratio normalizing the Information Gain** with a value called **Split Information**, which measures how broadly a feature splits the data.

$$\text{Information Gain Ratio} = \frac{\text{Information Gain}}{\text{Split Information}}$$

$$\text{Split Information} = - \sum_{i=1}^u \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

Where:

- Information Gain is calculated for the column to be evaluated, then divided by the split information function, which finds the total multiplication of the weight of each value in the column to be evaluated by the logarithm of the weight of that value. [22].
- Split Information measures the diversity of values in the column to be evaluated.
- $|S_i|$ is the number of records for the column to be evaluated that belong to index I, index I is the index of the value in the column to be evaluated.
- $|S|$ is the number of records for the column to be evaluated.
- U is the number of distinct values in the column to be evaluated.

3.2 Dataset features used:

Selected features of experiments for CICDDoS2019 dataset:

Table 2. Selected features of experiments for CICDDoS2019.

Feature selection method	No of features	Final features selected
Without feature selected	78	All features
Information Gain with Orange	15	Bwd Packets/s, Init Fwd Win Bytes, Down/Up Ratio, Bwd Packet Length Max, Avg Bwd Segment Size, Bwd Packet Length Mean, Subflow Bwd Packets, Total Backward Packets, Bwd IAT Mean, Bwd IAT Max, Bwd IAT Total, Bwd Header Length, Bwd IAT Min, Avg Fwd Segment Size, Fwd Packet Length Mean

Feature selection method	No of features	Final features selected
Information Gain with Weka	15	Avg Packet Size, Fwd Packet Length Mean, Avg Fwd Segment Size, Packet Length Max, Packet Length Mean, Fwd Packets Length Total, Subflow Fwd Bytes, Packet Length Min, Fwd Packet Length Min, Fwd Packet Length Max, Flow Bytes/s, Flow IAT Max, Fwd IAT Max, Flow IAT Mean, Bwd Packets/s
Information Gain Ratio with Orange	15	URG Flag Count, Down/Up Ratio, CWE Flag Count, RST Flag Count, Fwd PSH Flags, Bwd Packets/s, Init Fwd Win Bytes, Bwd Packet Length Std, Bwd Packet Length Max, Avg Bwd Segment Size, Bwd Packet Length Mean, Subflow Bwd Packets, Total Backward Packets, Bwd Header Length, Init Bwd Win Bytes
Information Gain Ratio with Weka	15	URG Flag Count, Down/Up Ratio, CWE Flag Count, RST Flag Count, Fwd PSH Flags, Bwd Packet Length Std, Bwd Packet Length Max, Packet Length Min, Avg Bwd Segment Size, Bwd Packet Length Mean, Bwd Packets/s, Fwd Packet Length Min, Init Fwd Win Bytes, Bwd IAT Min, Bwd Header Length

Selected features of experiments for CICDoS2017 dataset:

Table 3. Selected features of experiments for CICDoS2017.

Feature selection method	No of features	Final feature selected
Without feature selected	78	All features
Information Gain with Orange	15	Idle Max, Idle Mean, Idle Min, Fwd Packets/s, Active Min, Active Max, Bwd Packets/s, Active Mean, Flow IAT Max, Flow Packets/s, Flow IAT Mean, Flow Duration, Idle Std, Flow IAT Std, Fwd IAT Std
Information Gain with Weka	15	Init Bwd Win Bytes, Flow IAT Max, Bwd Packets/s, Fwd Packets/s, Flow IAT Mean, Flow Packets/s, Flow IAT Min, Flow Duration, Idle Max, Idle Mean, Idle Min, Packet Length Variance, Packet Length Std, Fwd IAT Max, Fwd Packet Length Max
Information Gain Ratio with Orange	15	Idle Std, Active Min, Active Max, Active Mean, Active Std, Idle Max, Idle Mean, Idle Min, Fwd Packets/s, Bwd Packets/s, SYN Flag Count, Flow IAT Max, Flow Packets/s, Flow IAT Mean, Flow Duration
Information Gain Ratio with Weka	15	Idle Std, Active Min, Active Max, Active Mean, Active Std, Idle Min, Idle Max, Idle Mean, Init Bwd Win Bytes, Fwd IAT Std, Fwd IAT Max, Fwd Packet Length Max, Fwd Packet Length Min, Bwd Packets/s, Bwd IAT Mean

3.3 Data Splitting

In this paper, a 70% training/30% testing and an 80% training/20% testing breakdown were used in the experiments from which the best model will be derived. Using these ratios for training and testing ensures that the training data is not oversized and that the testing data, which is crucial for determining the model's accuracy, is not neglected. It also ensures that the algorithms are trained with sufficient data to assess the quality of that data during the learning process and to identify the best algorithm using the appropriate structure.

3.4 Experiments setup

This research used an HP Probook 650 G2 laptop running Windows 10 64-bit. The processor is an Intel(R) Core(TM) i5-6300U CPU @ 2.40 GHz. The installed RAM is 16 GB. Weka 3.9.6 and Orange3 3.40.0 were used to run the experiments using the CICDoS2017 and CICDDoS2019 datasets.

3.6.1. Experimental Platforms

Experiments were conducted using two data mining frameworks, Weka and Orange3, to evaluate the performance of supervised machine learning algorithms under different analytical environments. Both platforms were used to implement the same set of algorithms and preprocessing steps to ensure a fair and consistent

comparison. To maintain experimental consistency, the default parameter settings of both frameworks were applied. This approach allowed the evaluation to reflect the inherent performance of each algorithm and framework without bias introduced by manual parameter tuning. The following a graphical example of the operations in Weka:

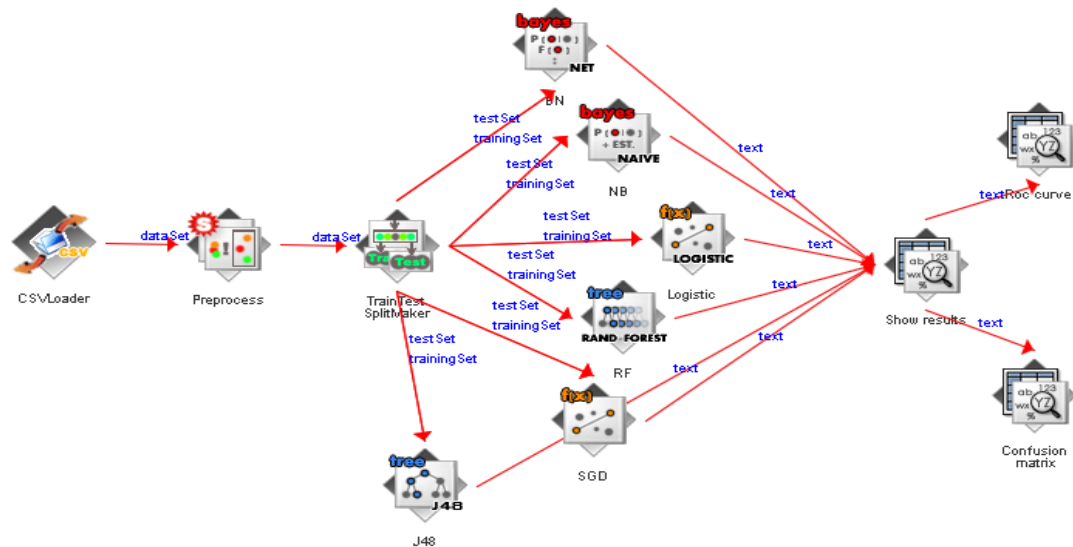


Figure 4. Experimental flowchart using Weka framework.

The following is a graphical example of the operations in Orange3:

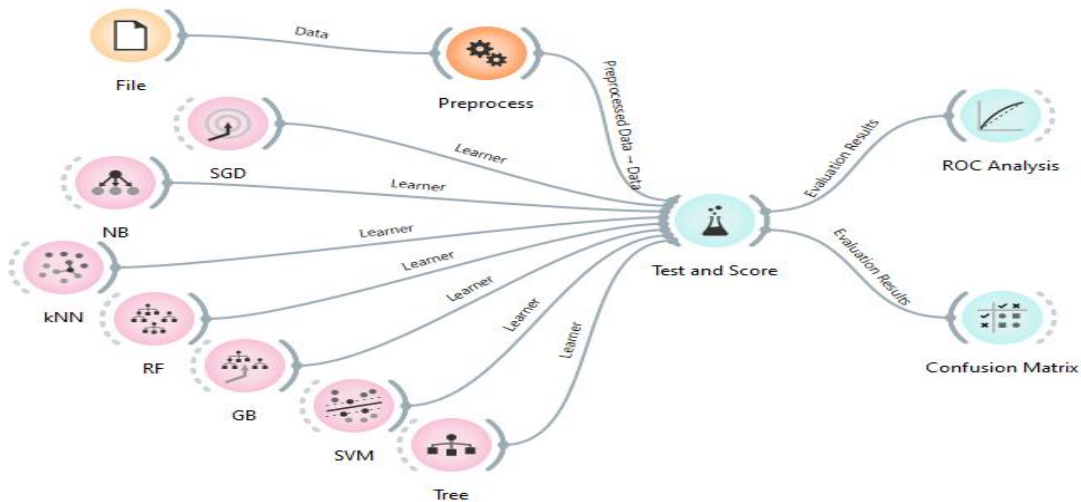


Figure 5. Workflow of experimental setup in Orange3.

3.6.2. Evaluation Phase

Machine learning is a branch of Artificial Intelligence that focuses on developing models and algorithms that let computers learn from data without being explicitly programmed for every task. In simple words, machine learning teaches the systems to think and understand like humans by learning from the data. The following are the targeted algorithms for the two frameworks:

Tree & J48 – RF – GB – SGD – NB – BN – Logistic – SVM – KNN

At this stage, 24 experiments were carried out using the two datasets, selected features, and two data partitions. Instead of repeating the presentation of results, the first 3 experiments carried out were mentioned, and then the results of all experiments were summarized in Table 10.

4. RESULTS

The results for each dataset will be divided into two parts: Orange3 and Weka, both of which contain the results of splitting the dataset into 70% training and 30% testing, and 80% training and 20% testing. The results are presented in following 8 tables:

Results of 70% for training & 30% for testing with Orange3 (CICDDoS2019 dataset)

Table 4. Results of the 24 experiments that were carried out.

EN	Model	Train	Test	AUC	CA	F1	Prec	Recall	Features Selection Algorithm
1	Tree	10.533	0.031	97.7%	98.3%	98.3%	98.3%	98.3%	Without feature selection
1	GB	58.788	1.254	99.6%	98.8%	98.8%	98.8%	98.8%	Without feature selection
1	RF	15.154	1.299	100%	99.9%	99.9%	99.9%	99.9%	Without feature selection
1	SGD	4.211	1.478	99.5%	99.5%	99.5%	99.5%	99.5%	Without feature selection
1	NB	5.651	2.372	99.4%	96.7%	96.8%	96.9%	96.7%	Without feature selection
1	SVM	119.156	21.697	75.2%	68.9%	69.3%	69.9%	68.9%	Without feature selection
1	kNN	2.02	989.774	99.7%	99.6%	99.6%	99.6%	99.6%	Without feature selection
2	Tree	1.891	0.017	99.4%	99%	99%	99%	99%	Information gain
2	SGD	1.287	0.179	87.8%	92.8%	92.6%	92.7%	92.8%	Information gain
2	GB	6.881	0.212	99.5%	96.4%	96.3%	96.5%	96.4%	Information gain
2	NB	1.096	0.256	97.3%	91.7%	91.7%	91.7%	91.7%	Information gain
2	RF	4.914	0.543	100%	99.9%	99.9%	99.9%	99.9%	Information gain
2	SVM	62.233	11.419	45.7%	77.3%	74.4%	74.4%	77.3%	Information gain
2	kNN	5.027	893.112	99.8%	99.7%	99.7%	99.7%	99.7%	Information gain
3	Tree	0.993	0.015	97.7%	98.3%	98.3%	98.3%	98.3%	Information gain ratio
3	NB	0.689	0.171	98.6%	97.9%	97.9%	98%	97.9%	Information gain ratio
3	SGD	1.42	0.216	98%	98.9%	98.9%	98.9%	98.9%	Information gain ratio
3	GB	4.992	0.294	98.6%	98.8%	98.8%	98.8%	98.8%	Information gain ratio
3	RF	2.418	0.341	99%	99.4%	99.4%	99.4%	99.4%	Information gain ratio
3	SVM	63.128	11.327	1.8%	89%	87.8%	90.1%	89%	Information gain ratio
3	kNN	0.489	517.329	98.7%	99.3%	99.3%	99.3%	99.3%	Information gain ratio

Results of 70% for training & 30% for testing with Orange3 (CICDoS2017 dataset)

Table 5. Results of the 24 experiments that were carried out.

EN	Model	Train	Test	AUC	CA	F1	Prec	Recall	Features Selection Algorithm
4	Tree	6.91	0.014	88.5%	98.5%	98.4%	98.5%	98.5%	Without feature selection
4	NB	1.748	0.319	95.4%	90.6%	92.2%	95.3%	90.6%	Without feature selection
4	SGD	1.656	0.47	88.7%	97.4%	97.4%	97.4%	97.4%	Without feature selection
4	RF	12.266	1.471	99.1%	99.7%	99.7%	99.7%	99.7%	Without feature selection
4	GB	54.579	4.871	97.4%	97.9%	97.7%	97.9%	97.9%	Without feature selection
4	SVM	70.925	19.768	38.8%	42.2%	54.5%	86.5%	42.2%	Without feature selection
4	kNN	4.076	585.329	97.4%	98.9%	98.9%	98.9%	98.9%	Without feature selection
5	Tree	4.278	0.026	87.1%	98%	97.9%	98%	98%	Information gain
5	SGD	0.5	0.08	60.6%	94.2%	92.9%	92.8%	94.2%	Information gain

EN	Model	Train	Test	AUC	CA	F1	Prec	Recall	Features Selection Algorithm
5	NB	0.648	0.127	91.7%	88.4%	90.5%	94.5%	88.4%	Information gain
5	GB	22.739	0.202	94.9%	95.7%	94.7%	95.9%	95.7%	Information gain
5	RF	7.416	0.418	97.3%	98.8%	98.8%	98.8%	98.8%	Information gain
5	kNN	2.255	7.21	96.2%	98.4%	98.4%	98.4%	98.4%	Information gain
5	SVM	65.656	14.834	22.4%	16.6%	23%	76.9%	16.6%	Information gain
6	Tree	2.15	0.032	87.7%	98.2%	98.1%	98.2%	98.2%	Information gain ratio
6	SGD	0.773	0.228	60.6%	94.9%	93.4%	94.6%	94.9%	Information gain ratio
6	NB	0.681	0.231	92.7%	90.3%	91.8%	94.5%	90.3%	Information gain ratio
6	RF	10.319	0.619	98.1%	99.4%	99.4%	99.4%	99.4%	Information gain ratio
6	GB	20.957	1.571	96.1%	97.4%	97.2%	97.4%	97.4%	Information gain ratio
6	SVM	52.068	12.192	17.3%	16.1%	23.1%	73.9%	16.1%	Information gain ratio
6	kNN	1.05	348.955	95.9%	98.3%	98.2%	98.2%	98.3%	Information gain ratio

Results of 70% for training & 30% for testing with Weka (CICDDoS2019 dataset)

Table 6. Results of the 24 experiments that were carried out.

EN	Model	Train	Test	AUC	CA	F1	Prec	Recall	Features Selection Algorithm
7	Logistic	65.46	0.2	99.9%	99.57%	99.6%	99.6%	99.6%	Without feature selection
7	SGD	87.45	0.25	99.6%	99.59%	99.6%	99.6%	99.6%	Without feature selection
7	J48	258.14	0.37	100%	99.95%	100%	100%	100%	Without feature selection
7	RF	58.87	1.13	100%	99.92%	99.9%	99.9%	99.9%	Without feature selection
7	BN	69.38	1.27	99.9%	98.82%	98.8%	98.9%	98.8%	Without feature selection
7	NB	6.65	3.78	96.5%	90.96%	90.6%	90.9%	91%	Without feature selection
8	J48	31.25	0.18	99.9%	99.72%	99.7%	99.7%	99.7%	Information gain
8	Logistic	16.05	0.25	98.9%	95.87%	95.9%	96.1%	95.9%	Information gain
8	SGD	72.63	0.25	95.7%	95.73%	95.8%	96%	95.7%	Information gain
8	RF	26.98	0.53	99.9%	99.73%	99.7%	99.7%	99.7%	Information gain
8	BN	22.13	0.6	99.9%	97.58%	97.6%	97.8%	97.6%	Information gain
8	NB	1.69	1.8	89.4%	85.81%	86.7%	90.7%	85.8%	Information gain
9	J48	26.72	0.19	100%	99.93%	99.9%	99.9%	99.9%	Information gain ratio
9	SGD	75.61	0.22	97.4%	98.58%	98.6%	98.6%	98.6%	Information gain ratio
9	Logistic	22.03	0.25	99.1%	98.43%	98.4%	98.4%	98.4%	Information gain ratio
9	RF	21.58	0.42	100%	99.92%	99.9%	99.9%	99.9%	Information gain ratio
9	BN	7.62	0.51	99.9%	99.17%	99.2%	99.2%	99.2%	Information gain ratio
9	NB	1.66	1.87	98.5%	92.69%	92.4%	92.9%	92.7%	Information gain ratio

Results of 70% for training & 30% for testing with Weka (CICDoS2017 dataset)

Table 7. Results of the 24 experiments that were carried out.

EN	Model	Train	Test	AUC	CA	F1	Prec	Recall	Features Selection Algorithm
10	SGD	66.94	0.28	84.9%	97.6%	97.5%	97.5%	97.6%	Without feature selection
10	J48	238.21	0.48	99.3%	99.73%	99.7%	99.7%	99.7%	Without feature selection
10	Logistic	63.59	0.92	98.1%	98.05%	98%	98%	98.1%	Without feature selection
10	BN	53.51	1.02	98.7%	98.79%	98.8%	98.8%	98.8%	Without feature selection
10	RF	50.92	1.08	98.9%	99.69%	99.7%	99.7%	99.7%	Without feature selection
10	NB	6.38	2.96	92.7%	48.49%	59.3%	94%	48.5%	Without feature selection
11	Logistic	6.75	0.31	85.7%	94.04%	92.5%	92.6%	94%	Information gain
11	SGD	31.92	0.39	50.2%	93.52%	90.4%	92.5%	93.5%	Information gain
11	BN	10.65	0.56	99%	97.38%	97.5%	97.8%	97.4%	Information gain
11	J48	27.11	0.62	99.3%	99.72%	99.7%	99.7%	99.7%	Information gain
11	RF	26.02	0.93	98.9%	99.7%	99.7%	99.7%	99.7%	Information gain
11	NB	1.51	1.03	87.5%	92.33%	93%	94%	92.3%	Information gain
12	Logistic	7.07	0.13	86.2%	94.61%	93.6%	93.7%	94.6%	Information gain ratio
12	SGD	39.24	0.14	59.2%	94.3%	92.7%	93.3%	94.3%	Information gain ratio
12	BN	7.03	0.27	98.8%	97.68%	97.7%	97.8%	97.7%	Information gain ratio
12	J48	26.23	0.29	99.4%	99.72%	99.7%	99.7%	99.7%	Information gain ratio
12	RF	22.48	0.42	98.9%	99.72%	99.7%	99.7%	99.7%	Information gain ratio
12	NB	1.15	0.5	86.7%	92.59%	92.5%	92.4%	92.6%	Information gain ratio

Results of 80% for training & 20% for testing with Orange3 (CICDDoS2019 dataset)

Table 8. Results of the 24 experiments that were carried out.

EN	Model	Train	Test	AUC	CA	F1	Prec	Recall	Features Selection Algorithm
13	Tree	12.431	0.026	97.7%	98.3%	98.3%	98.3%	98.3%	Without feature selection
13	GB	67.777	0.82	99.6%	98.9%	98.9%	98.9%	98.9%	Without feature selection
13	RF	17.109	0.98	100%	99.9%	99.9%	99.9%	99.9%	Without feature selection
13	SGD	4.761	1.161	99.6%	99.6%	99.6%	99.6%	99.6%	Without feature selection
13	NB	7.465	1.244	99.4%	96.7%	96.8%	96.9%	96.7%	Without feature selection
13	SVM	139.267	14.925	97.7%	76.9%	68.2%	80.9%	76.9%	Without feature selection
13	kNN	2.11	759.553	99.7%	99.6%	99.6%	99.6%	99.6%	Without feature selection
14	Tree	2.189	0.015	99.4%	99%	99%	99%	99%	Information gain
14	SGD	1.795	0.134	88.1%	92.6%	92.4%	92.5%	92.6%	Information gain
14	NB	1.268	0.197	97.3%	91.7%	91.7%	91.7%	91.7%	Information gain
14	GB	10.764	0.255	99.6%	96.4%	96.3%	96.4%	96.4%	Information gain
14	RF	5.123	0.361	100%	99.9%	99.9%	99.9%	99.9%	Information gain
14	SVM	94.294	10.958	42.9%	76.6%	73.7%	73.5%	76.6%	Information gain
14	kNN	5.78	779.237	99.8%	99.7%	99.7%	99.7%	99.7%	Information gain
15	Tree	1.094	0.007	97.7%	98.3%	98.3%	98.3%	98.3%	Information gain ratio
15	NB	0.758	0.131	98.6%	97.9%	97.9%	97.9%	97.9%	Information gain ratio
15	SGD	1.645	0.17	98%	98.9%	98.9%	98.9%	98.9%	Information gain ratio
15	RF	2.794	0.282	99%	99.4%	99.4%	99.4%	99.4%	Information gain ratio
15	GB	7.502	0.303	98.6%	98.8%	98.8%	98.8%	98.8%	Information gain ratio
15	SVM	84.873	9.741	5.3%	90.4%	89.6%	91.3%	90.4%	Information gain ratio
15	kNN	0.573	400.848	98.9%	99.3%	99.3%	99.3%	99.3%	Information gain ratio

Results of 80% for training & 20% for testing with Orange3 (CICDoS2017 dataset)

Table 9. Results of the 24 experiments that were carried out.

EN	Model	Train	Test	AUC	CA	F1	Prec	Recall	Features Selection Algorithm
16	Tree	2.363	0.004	88.6%	98.5%	98.4%	98.5%	98.5%	Without feature selection
16	SGD	1.679	0.158	88.2%	97.9%	97.9%	97.8%	97.9%	Without feature selection
16	GB	24.858	0.191	97.2%	97.9%	97.7%	97.9%	97.9%	Without feature selection
16	RF	10.501	0.24	99%	99.7%	99.7%	99.7%	99.7%	Without feature selection
16	NB	1.695	0.277	95.5%	90.6%	92.2%	95.3%	90.6%	Without feature selection
16	SVM	40.563	3.065	46.5%	46.1%	58.4%	86.5%	46.1%	Without feature selection
16	kNN	0.603	91.464	97.2%	98.9%	98.9%	98.9%	98.9%	Without feature selection
17	Tree	14.562	0.036	87.3%	98.1%	97.9%	98%	98.1%	Information gain
17	NB	1.184	0.212	91.8%	88.4%	90.6%	94.5%	88.4%	Information gain
17	GB	31.421	0.557	94.7%	95.5%	94.3%	95.6%	95.5%	Information gain
17	RF	18.378	1.527	97.1%	98.8%	98.8%	98.8%	98.8%	Information gain
17	SGD	4.616	1.631	52%	93.8%	91.2%	92.8%	93.8%	Information gain
17	kNN	8.09	8.899	96.2%	98.4%	98.4%	98.4%	98.4%	Information gain
17	SVM	72.475	9.475	18.5%	16.6%	23.1%	76.5%	16.6%	Information gain
18	Tree	2.146	0.009	87.9%	98.2%	98.1%	98.3%	98.2%	Information gain ratio
18	SGD	1.769	0.15	66.7%	95.1%	94.2%	94.4%	95.1%	Information gain ratio
18	NB	1.752	0.158	92.7%	90.3%	91.8%	94.5%	90.3%	Information gain ratio
18	GB	23.951	0.201	96.1%	97.4%	97.1%	97.3%	97.4%	Information gain ratio
18	RF	19.663	0.511	98%	99.3%	99.3%	99.3%	99.3%	Information gain ratio
18	SVM	79.421	9.92	16.9%	16.2%	23.2%	73.8%	16.2%	Information gain ratio
18	kNN	0.52	228.008	95.9%	98.2%	98.2%	98.2%	98.2%	Information gain ratio

Results of 80% for training & 20% for testing with Weka (CICDDoS2019 dataset)

Table 10. Results of the 24 experiments that were carried out.

EN	Model	Train	Test	AUC	CA	F1	Prec	Recall	Features Selection Algorithm
19	Logistic	59.3	0.18	99.9%	99.55%	99.5%	99.6%	99.5%	Without feature selection
19	SGD	98.77	0.19	99.5%	99.57%	99.6%	99.6%	99.6%	Without feature selection
19	J48	264.05	0.21	100%	99.95%	99.9%	99.9%	99.9%	Without feature selection
19	RF	61.49	0.45	100%	99.93%	99.9%	99.9%	99.9%	Without feature selection
19	BN	63.45	0.87	99.9%	98.79%	98.8%	98.8%	98.8%	Without feature selection
19	NB	5.77	2.43	96.4%	90.93%	90.5%	90.9%	90.9%	Without feature selection
20	J48	30.94	0.11	99.9%	99.72%	99.7%	99.7%	99.7%	Information gain
20	Logistic	20.33	0.14	98.8%	95.81%	95.9%	96.1%	95.8%	Information gain
20	SGD	73.17	0.16	95.7%	95.65%	95.7%	95.9%	95.6%	Information gain
20	RF	27.08	0.3	99.9%	99.73%	99.7%	99.7%	99.7%	Information gain
20	BN	20.19	0.42	99.9%	97.56%	97.6%	97.8%	97.6%	Information gain
20	NB	1.84	0.77	89.1%	85.63%	86.5%	90.6%	85.6%	Information gain
21	Logistic	17.87	0.15	99.1%	98.43%	98.4%	98.4%	98.4%	Information gain ratio
21	SGD	85.81	0.17	97.4%	98.58%	98.6%	98.6%	98.6%	Information gain ratio
21	BN	7.41	0.28	99.9%	99.18%	99.2%	99.2%	99.2%	Information gain ratio
21	J48	34.1	0.28	100%	99.91%	99.9%	99.9%	99.9%	Information gain ratio
21	RF	21.35	0.29	100%	99.91%	99.9%	99.9%	99.9%	Information gain ratio
21	NB	1.77	2.13	98.5%	92.63%	92.3%	92.8%	92.6%	Information gain ratio

Results of 80% for training & 20% for testing with Weka (CICDoS2017 dataset)

Table 11. Results of the 24 experiments that were carried out.

EN	Model	Train	Test	AUC	CA	F1	Prec	Recall	Features Selection Algorithm
22	Logistic	72.42	0.14	98.1%	98.06%	98%	98%	98.1%	Without feature selection
22	J48	238.47	0.15	99.1%	99.72%	99.7%	99.7%	99.7%	Without feature selection
22	SGD	67.83	0.16	84.9%	97.63%	97.5%	97.5%	97.6%	Without feature selection
22	RF	58.78	0.39	98.8%	99.69%	99.7%	99.7%	99.7%	Without feature selection
22	BN	47.06	0.84	98.7%	98.85%	98.9%	98.9%	98.9%	Without feature selection
22	NB	5.17	2.25	92.6%	48.24%	59.1%	94.1%	48.2%	Without feature selection
23	J48	25.53	0.08	99.2%	99.7%	99.7%	99.7%	99.7%	Information gain
23	SGD	38.12	0.08	50.5%	93.61%	90.6%	93.1%	93.6%	Information gain
23	Logistic	6.25	0.14	85.8%	94.07%	92.6%	92.7%	94.1%	Information gain
23	BN	10.11	0.35	99%	97.34%	97.5%	97.8%	97.3%	Information gain
23	RF	24.93	0.46	98.7%	99.69%	99.7%	99.7%	99.7%	Information gain
23	NB	1.35	0.6	87.2%	92.12%	92.9%	94%	92.1%	Information gain
24	Logistic	7.05	0.09	86.2%	94.67%	93.7%	93.7%	94.7%	Information gain ratio
24	SGD	37.16	0.1	59.1%	94.35%	92.7%	93.4%	94.3%	Information gain ratio
24	J48	20.58	0.12	99.3%	99.74%	99.7%	99.7%	99.7%	Information gain ratio
24	BN	8.5	0.27	98.8%	97.71%	97.8%	97.8%	97.7%	Information gain ratio
24	RF	22.53	0.51	98.8%	99.73%	99.7%	99.7%	99.7%	Information gain ratio
24	NB	1.08	0.71	86.9%	92.34%	92.3%	92.3%	92.3%	Information gain ratio

The following table summarizes the best algorithms in all criteria with the feature selection (FS) algorithm used whether it is Information gain (IG) or Information gain ratio (IGR) or without using it:

Table 12. Summary of best algorithms in all criteria.

Split	Tool	Dataset	Feature selection	Train	Test	AUC	CA	F1	Precision	Recall
70%-30%	Orange3	CICDDoS 2019	without FS	KNN	Tree	RF	RF	RF	RF	RF
			IG	NB	Tree	RF	RF	RF	RF	RF
			IGR	KNN	Tree	RF	RF	RF	RF	RF
		CICDoS 2017	without FS	SGD	Tree	RF	RF	RF	RF	RF
			IG	SGD	Tree	RF	RF	RF	RF	RF
			IGR	NB	Tree	RF	RF	RF	RF	RF
	Weka	CICDDoS 2019	without FS	NB	Logistic	J48, RF	J48	J48	J48	J48
			IG	NB	Logistic	RF	J48, RF	J48, RF	J48, RF	J48, RF
			IGR	NB	J48	J48, RF	J48, RF	J48, RF	J48, RF	J48, RF
		CICDoS 2017	without FS	NB	SGD	J48	J48	J48, RF	J48, RF	J48, RF
			IG	NB	Logistic	J48	J48	J48, RF	J48, RF	J48, RF
			IGR	NB	Logistic	J48	J48	J48, RF	J48, RF	J48, RF
80%-20%	Orange3	CICDDoS 2019	without FS	KNN	Tree	RF	RF	RF	RF	RF
			IG	NB	Tree	RF	RF	RF	RF	RF
			IGR	KNN	Tree	RF	RF	RF	RF	RF
		CICDoS 2017	without FS	KNN	Tree	RF	RF	RF	RF	RF
			IG	NB	Tree	RF	RF	RF	RF	RF
			IGR	KNN	Tree	RF	RF	RF	RF	RF
	Weka	CICDDoS 2019	without FS	NB	Logistic	RF	J48, RF	J48, RF	J48, RF	J48, RF

Split	Tool	Dataset	Feature selection	Train	Test	AUC	CA	F1	Precision	Recall
			IG	NB	J48	BN, J48, RF	RF	J48, RF	J48, RF	J48, RF
			IGR	NB	Logistic	J48	J48, RF	J48, RF	J48, RF	J48, RF
		CICDoS 2017	without FS	NB	Logistic	J48	J48	J48, RF	J48, RF	J48, RF
			IG	NB	SGD	J48	J48	J48, RF	J48, RF	J48, RF
			IGR	NB	Logistic	J48	J48	J48, RF	J48, RF	J48, RF

5. DISCUSSION

5.1. Best algorithms

Based on the results of experiments conducted using the CICDDoS2019 and CICDoS2017 datasets with Weka and Orange3 tools, the best-performing algorithms were identified as follows:

- Fastest Training: KNN (0.489 seconds) using Orange3 on CICDDoS2019 with Information Gain Ratio — Experiment No. (3).
- Fastest Testing: Tree (0.004 seconds) using Orange3 on CICDoS2017 without feature selection — Experiment No. (16).
- Best Overall Accuracy (AUC, CA, F1, Precision, and Recall): J48 using Weka on CICDDoS2019, without feature selection, and using 70% training size — Experiment No. (7).
- Best with Weka: J48 — detailed in Experiment No. (7).
- Best with Orange3: RF — detailed in Experiment No. (2).
- Best with CICDDoS2019: J48 — Experiment No. (7).
- Best with CICDoS2017: J48 — Experiment No. (24).
- Best at 70% Training Size: J48 — Experiment No. (7).
- Best at 80% Training Size: J48 — Experiment No. (19).
- Best Without Feature Selection: J48 — Experiment No. (7).
- Best with Information Gain: RF — Experiment No. (2).
- Best with Information Gain Ratio: J48 — Experiment No. (9).

General Conclusion: Omitting feature selection increases accuracy but requires more time. Information Gain and Information Gain Ratio provide faster training and testing, with Information Gain Ratio delivering superior accuracy in most experiments compared to Information Gain.

5.2. Average performance

Average performance of all algorithms with Weka

Table 13. Average performance of all algorithms with Weka.

Algorithm	Train	Test	AUC	CA	F1	Prec	Recall
J48	101.7775	0.256667	99.6167%	99.7925%	99.775%	99.775%	99.775%
RF	35.25083	0.575833	99.4%	99.78%	99.7667%	99.7667%	99.7667%
BN	23.69318	0.605	99.3667%	98.2375%	98.2833%	98.3833%	98.25%
Logistic	30.3475	0.241667	94.65%	96.7633%	96.3417%	96.4083%	96.7667%
SGD	64.55417	0.199167	81.175%	96.5592%	95.775%	96.3%	96.55%
NB	3.001667	1.735833	91.8333%	83.73%	85.675%	92.4667%	83.7167%

4.3. Average performance of algorithms showed the following results:

- ✓ The best algorithm in terms of training time is NB.
- ✓ The best algorithm in terms of testing time is SGD.
- ✓ The best algorithm in terms of AUC, CA, F1, Prec, and Recall is J48.

Below is a representation of average performance of algorithms in charts:

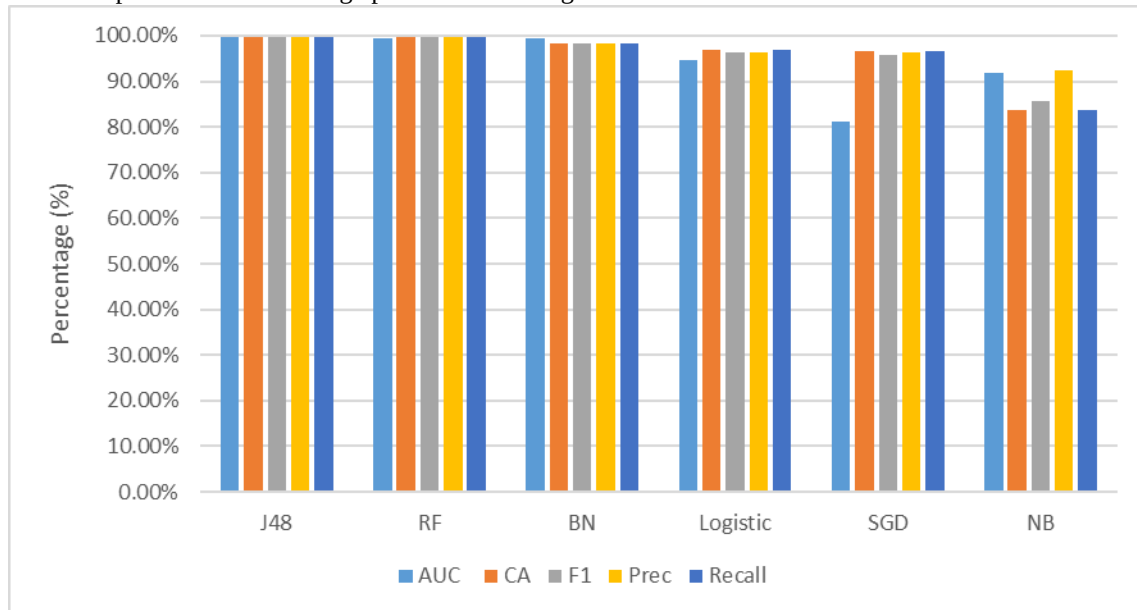


Figure 6. Average performance of all algorithms

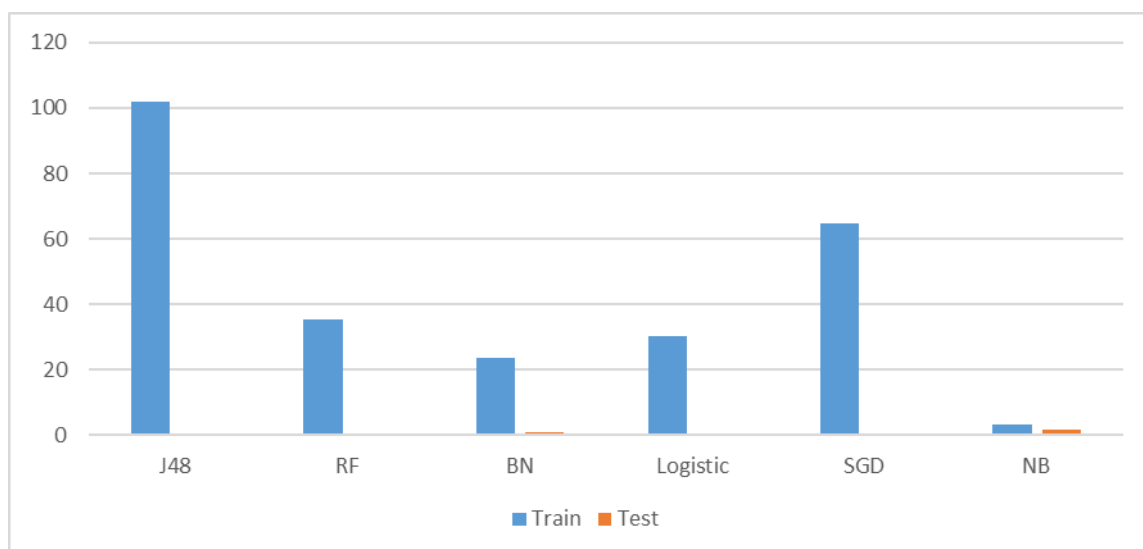


Figure 7. Average of training time and Testing time.

Average performance of all algorithms with Orange3

Table 14. Average performance of all algorithms with Orange3.

Algorithm	Train	Test	AUC	CA	F1	Prec	Recall
RF	10.50458	0.716	98.8833%	99.5083%	99.5083%	99.5083%	99.5083%
kNN	2.716083	467.4765	97.95%	99.025%	99.0167%	99.0167%	99.025%
Tree	5.128333	0.019333	93.0583%	98.3917%	98.3333%	98.3917%	98.3917%
GB	27.93408	0.89425	97.6583%	97.4917%	97.2167%	97.5167%	97.4917%
SGD	2.176	0.504583	82.3167%	96.3%	95.7417%	95.9917%	96.3%
NB	2.052917	0.474583	95.8667%	92.6%	93.4917%	95.1417%	92.6%
SVM	78.67158	12.44342	35.75%	52.7417%	55.6917%	79.5167%	52.7417%

4.4. Average performance of algorithms showed the following results:

- ✓ The best algorithm in terms of training time is NB.
- ✓ The best algorithm in terms of testing time is Tree.
- ✓ The best algorithm in terms of AUC, CA, F1, Prec, and Recall is RF.

Below is a representation of average performance of algorithms in charts:

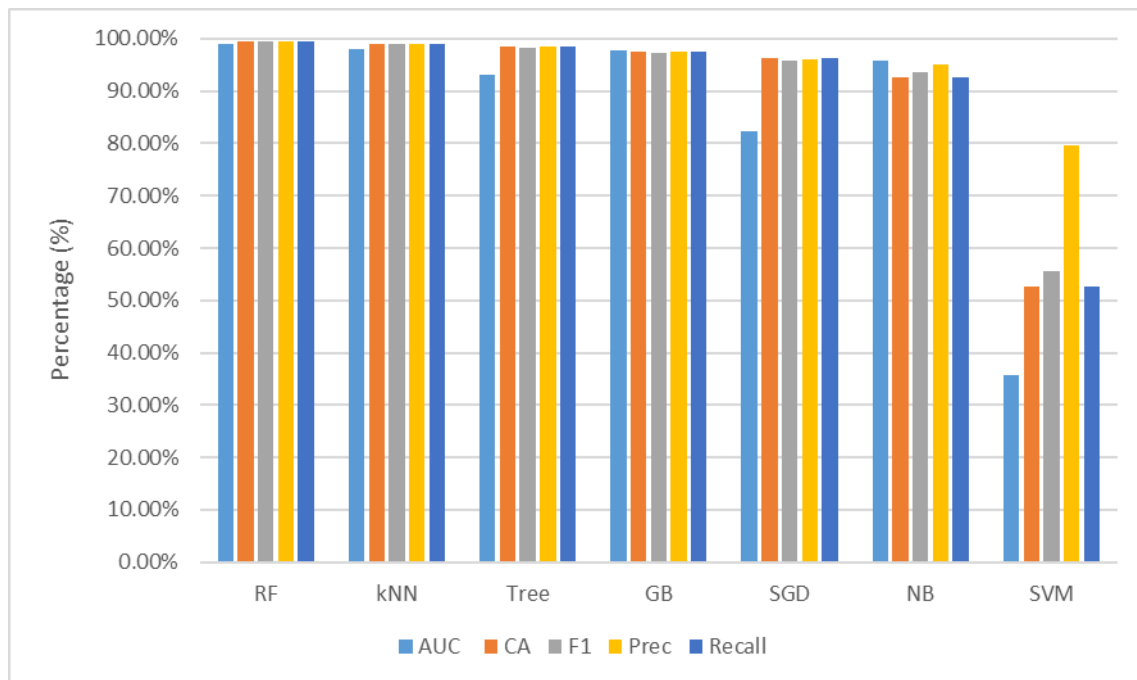


Figure 8. Average performance of all algorithms

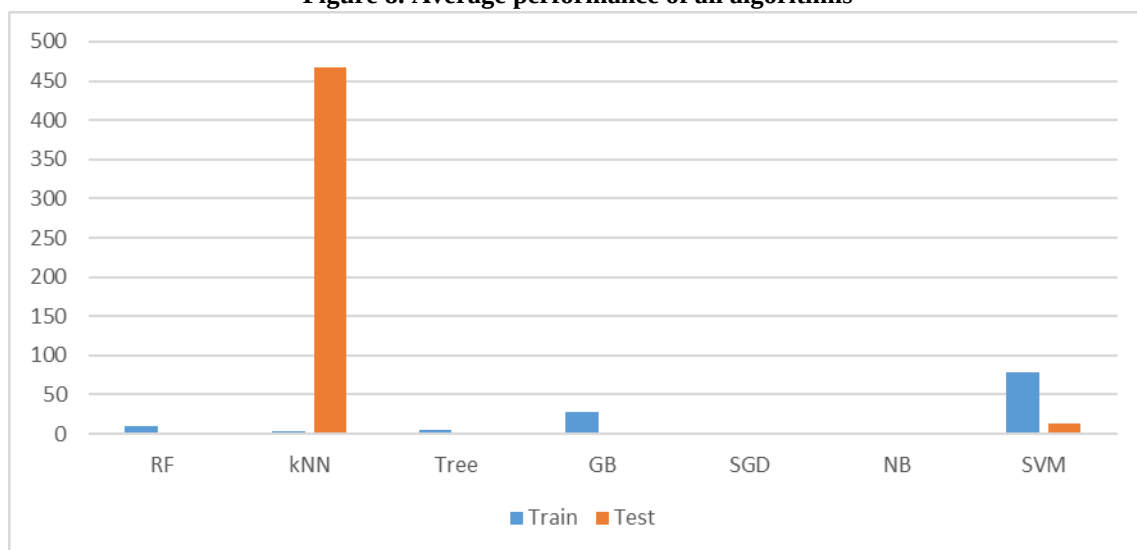


Figure 9. Average of training time and Testing time.

The J48 algorithm was nominated as the best algorithm in this paper using the study sample because it achieved the best average performance in accuracy with good training and testing speed. The following Roc curve and confusion matrix of J48 with CICDDoS2019 without using any feature selection:

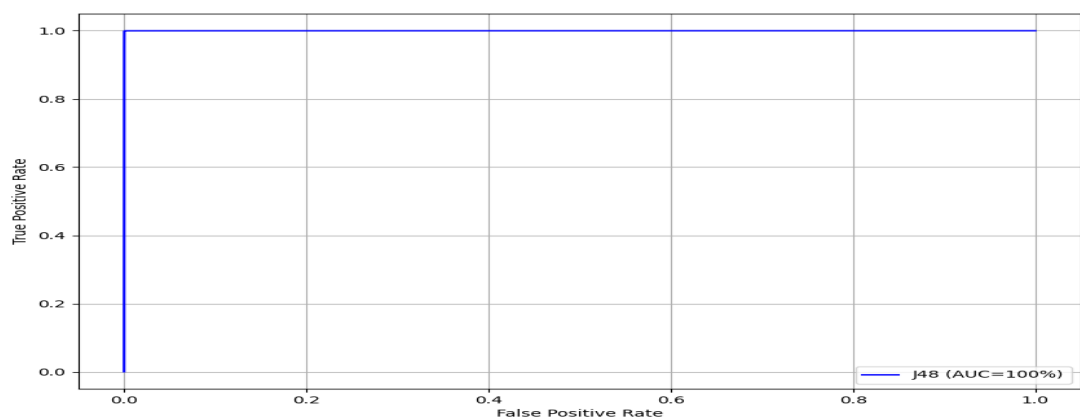


Figure 10. Average of training time and Testing time.

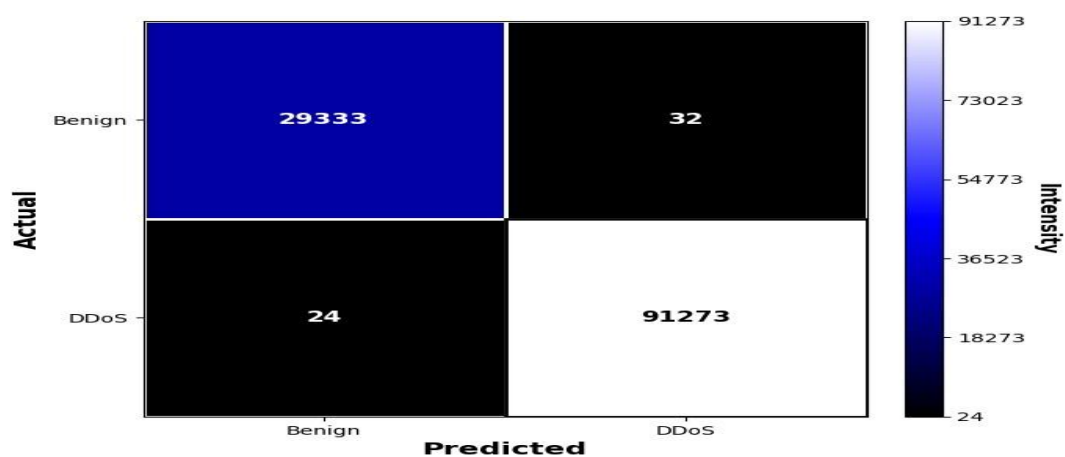


Figure 11. Average of training time and Testing time.

The following Roc curve and confusion matrix of J48 with CICDoS2017 without using any feature selection:

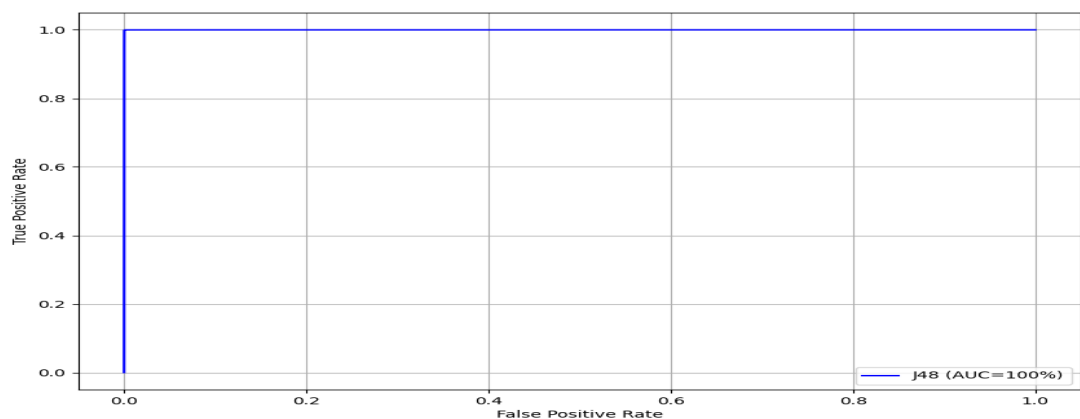


Figure 12. Average of training time and Testing time.

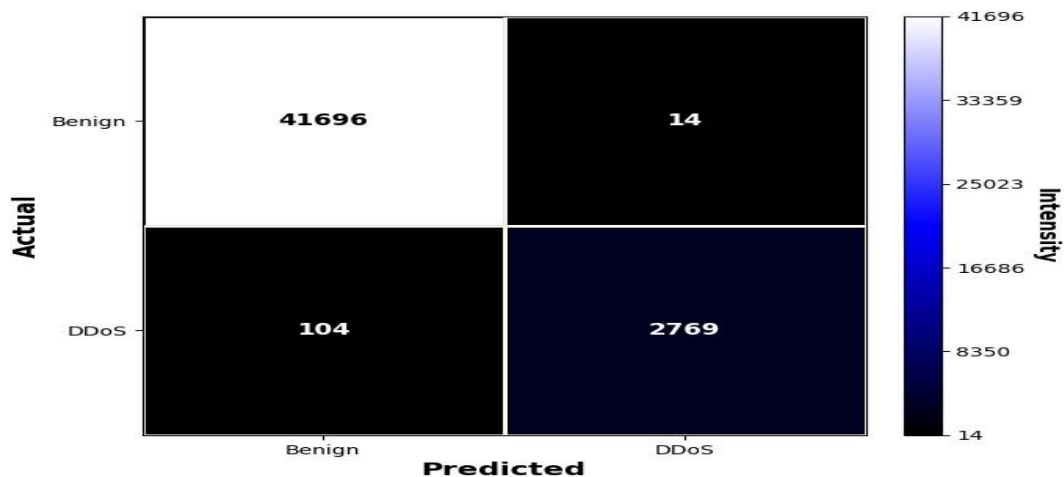


Figure 13. Average of training time and Testing time.

5. Compare with related works

Here the researcher compared average accuracy of the best algorithm with the previous studies that used the same or one of the same datasets because the data used in training the algorithm is mostly what affects the algorithm's ability and for this reason the study only cared about previous studies that used some or both of the datasets in this comparison.

Table 15. Results comparison with existing works.

Authors	Dataset	Algorithm	Accuracy
AlSaleh el al. [2]	CICDDoS2019 with PCA for Feature selection	BaysCNN and BaysFusCNN	99.66%, 99.79%
Pandey & Mishra [8]	CICDDoS2019	RF + Pearson correlation coefficient	99%
Hirsi el al. [14]	CICDoS2017	Ensemble model using RF, KNN and SVM	98%
This study	CICDDoS2019	J48	99.95%
	CICDoS2017	J48 + Information gain ratio	99.3%

As is clear from the table, the candidate algorithm in the study outperforms the rest of the algorithms in this comparison.

6. CONCLUSION

This study presented a comprehensive evaluation of supervised machine learning algorithms for DDoS attack detection using two benchmark datasets, CICDoS2017 and CICDDoS2019, across two analytical frameworks, Weka and Orange3. Unlike many previous studies that rely on a single dataset or platform, this work provides a broader and more robust comparative analysis through 24 experimental scenarios. The experimental results demonstrated that tree-based algorithms, particularly J48, achieved the best overall performance across multiple evaluation metrics, including Accuracy, AUC, Precision, Recall, and F1-score. J48 consistently outperformed other models, achieving 99.95% accuracy on CICDDoS2019 without using any features selection algorithm and 99.3% on CICDoS2017 using Information gain ratio. In terms of computational performance, NB achieved the fastest training time, while tree-based models demonstrated efficient testing performance, making them suitable for real-time detection scenarios. The results also revealed that preprocessing techniques do not universally improve performance and their effectiveness depends on the compatibility between the dataset and the learning algorithm. Additionally, the study showed that dataset size significantly affects model performance, with larger datasets leading to better predictive accuracy. The comparison between Weka and Orange3 highlighted that framework selection influences both accuracy and computational efficiency, where Weka provided more consistent results while Orange3 offered faster execution in certain cases. Overall, this study contributes a large-scale experimental comparison based on 24 experimental scenarios and provides practical insights for selecting efficient and reliable machine learning models for DDoS

detection, emphasizing the effectiveness of decision tree-based approaches in achieving a balance between accuracy and computational cost.

7. LIMITATION

Despite the promising results, several limitations should be considered. First, the performance of the models depends on the quality and correctness of the labeled datasets, which may contain inaccuracies affecting the results. Second, computational performance is influenced by the hardware environment, meaning that execution time may vary depending on system capabilities. Finally, the study is limited to two datasets, which may not fully represent all types and variations of DDoS attacks.

8. FUTURE WORK

Future research can extend this work by incorporating additional datasets that cover a wider range of attack types and network conditions. Evaluating the proposed models in real-time environments using streaming data would further validate their practical applicability. Moreover, exploring hybrid and ensemble learning approaches may improve detection accuracy and system robustness.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that they have no conflict of interest.

References

1. Alashhab, A. A., Zahid, M. S., Isyaku, B., Elnour, A. A., Nagmeldin, W., Abdelmaboud, A., ... & Maiwada, U. (2024). Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model. *IEEE Access*.
2. AlSaleh, I., Al-Samawi, A., & Nissirat, L. (2024). Novel machine learning approach for DDoS cloud detection: Bayesian-based CNN and data fusion enhancements. *Sensors*, 24(5), 1418.
3. Amarnath, J. L., Chandramouli, H., & Shah, P. G. (2024). Design and implementation of machine learning methodologies for DDoS attack detection and cloud security. *International Journal of Communication Networks and Information Security*, 16(3), 448–455.
4. Dembele, A., Mwangi, E., Ronoh, K. K., & Ataro, E. O. (2024). A novel approach for detection of DDoS attacks in software-defined networks based on Grey Wolf Optimizer and Support Vector Machine.
5. Hasan, K., Hossain, K. S., Apurbo, G. M., Islam, M. D., & Alam, M. S. (2024). Real-time DDoS detection in software-defined networks using machine learning (Doctoral dissertation, Brac University).
6. Kikissagbe, B. R., Adda, M., Célicourt, P., Haman, I. T., & Najjar, A. (2024). Machine learning for DoS attack detection in IoT systems. *Procedia Computer Science*, 241, 195–202.
7. Naiem, S., Khedr, A. E., Idrees, A. M., & Marie, M. I. (2023). Enhancing the efficiency of Gaussian naïve Bayes machine learning classifier in the detection of DDoS in cloud computing. *IEEE Access*, 11, 124597–124608.
8. Pandey, N., & Mishra, P. K. (2023). Detection of DDoS attack in IoT traffic using ensemble machine learning techniques. *Networks & Heterogeneous Media*, 18(4).
9. Manisankar, S., Hariharan, M. N. J., Ashwin, M., & Gowthami, N. (2024, March). Detection and mitigation of DDoS attack using hybrid machine learning approach in SDN. In 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 505–509). IEEE.
10. Sharif, D. M., Beitollahi, H., & Fazeli, M. (2023). Detection of application-layer DDoS attacks produced by various freely accessible toolkits using machine learning. *IEEE Access*, 11, 51810–51819.
11. Şimşek, M. M., & Atılğan, E. (2024). DoS and DDoS attacks on Internet of Things and their detection by machine learning algorithms. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 15(2), 341–353.
12. Swami, S., Singh, P., & Chauhan, S. S. (2024, May). Design and analysis of an integrated rule-based and machine learning system for DoS attack detection in wireless sensor networks. In 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE) (pp. 1046–1052). IEEE.
13. Valikhanli, O. (2024). UAV networks DoS attacks detection using artificial intelligence based on weighted machine learning. *Results in Control and Optimization*, 16, 100457.

14. Hirs, A., Audah, L., Salh, A., Sahar, N. M., Alhartomi, M. A., & Ahmed, S. (2024, December). Detecting low-rate DDoS attacks in SDN using ensemble machine learning techniques. In 2024 IEEE 22nd Student Conference on Research and Development (SCOREd) (pp. 299–304). IEEE.
15. Lee, Y. H. (2024). Data mining techniques for effective detection of distributed denial-of-service attacks (Master's thesis, UTAR).
16. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In 2019 International Carnahan Conference on Security Technology (ICCST) (pp. 1–8). IEEE.
17. D'hooge, L. (2022a). CICDDoS2019 dataset [Data set]. Kaggle. Retrieved September 1, 2025, from <https://www.kaggle.com/datasets/dhoogla/cicddos2019>
18. D'hooge, L. (2022b). CICDoS2017 dataset [Data set]. Kaggle. Retrieved September 1, 2025, from <https://www.kaggle.com/datasets/dhoogla/cicdos2017>
19. Jazi, H. H., Gonzalez, H., Stakhanova, N., & Ghorbani, A. A. (2017). Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. *Computer Networks*, 121, 25–36.
20. Canadian Institute for Cybersecurity. (n.d.). DDoS evaluation dataset (CIC-DDoS2019). University of New Brunswick. Retrieved August 22, 2025, from <https://www.unb.ca/cic/datasets/ddos-2019.html>
21. Zhou, V. (2022, September 16). A simple explanation of information gain and entropy. Retrieved June 29, 2025, from <https://victorzhou.com/blog/information-gain/>
22. Natarajan, M. (2023, October 31). Entropy metrics in decision trees: Diving deep into the decisions. Medium. <https://medium.com/@megha.natarajan/entropy-metrics-in-decision-trees-diving-deep-into-the-decisions-64355237a193>
23. Orange3. (n.d.). Tree. Retrieved from <https://orange3.readthedocs.io/projects/orange-visual-programming/en/master/widgets/model/tree.html>
24. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. Retrieved from <https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf>
25. GeeksforGeeks. (2025a). Gradient boosting in ML. Retrieved from <https://www.geeksforgeeks.org/machine-learning/ml-gradient-boosting/>
26. GeeksforGeeks. (2025b). ML – Stochastic gradient descent (SGD). Retrieved from <https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/>
27. GeeksforGeeks. (2025c, August 25). Naive Bayes classifiers. <https://www.geeksforgeeks.org/machine-learning/naive-bayes-classifiers/>
28. Bouckaert, R. R. (2004, September 1). Bayesian network classifiers in Weka [PDF manual]. University of Waikato. Retrieved from <https://weka.sourceforge.io/manuals/weka.bn.pdf>
29. GeeksforGeeks. (2025d). Logistic regression in machine learning. <https://www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/>
30. Liu, Z., Wang, Y., Feng, F., Liu, Y., Li, Z., & Shan, Y. (2023). A DDoS detection method based on feature engineering and machine learning in software-defined networks. *Sensors*, 23(13), 6176.
31. Jayaswal, V. (2020, September 14). Performance metrics: Confusion matrix, precision, recall and F1-score. Towards Data Science. Retrieved September 9, 2025, from <https://towardsdatascience.com/performance-metrics-confusion-matrix-precision-recall-and-f1-score-a8fe076a2262/>
32. Brownlee, J. (2020, December 10). Make better predictions with boosting, bagging and blending ensembles in Weka. *Machine Learning Mastery*. <https://machinelearningmastery.com/improve-machine-learning-results-with-boosting-bagging-and-blending-ensemble-methods-in-weka/>

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of **AJAPAS** and/or the editor(s). **AJAPAS** and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.