Computers, Materials & Continua
DOI: 10.32604/cmc.2022.026552
Article

Tech Science Press

Comprehensive DDoS Attack Classification Using Machine Learning Algorithms

Olga Ussatova[1,2], Aidana Zhumabekova[1,*], Yenlik Begimbayeva[2,3], Eric T. Matson[4] and Nikita Ussatov[5]

[1]Al-Farabi Kazakh National University, Almaty, 050040, Kazakhstan
[2]Institute of Information and Computational Technologies, Almaty, 050010, Kazakhstan
[3]Satbayev University, Almaty, 050013, Kazakhstan
[4]Purdue University, West Lafayette, 47907, IN, USA
[5]Turan University, Almaty, 050013, Kazakhstan
*Corresponding Author: Aidana Zhumabekova. Email: zhumabekova2702@gmail.com
Received: 30 December 2021; Accepted: 22 February 2022

Abstract: The fast development of Internet technologies ignited the growth of techniques for information security that protect data, networks, systems, and applications from various threats. There are many types of threats. The dedicated denial of service attack (DDoS) is one of the most serious and widespread attacks on Internet resources. This attack is intended to paralyze the victim's system and cause the service to fail. This work is devoted to the classification of DDoS attacks in the special network environment called Software-Defined Networking (SDN) using machine learning algorithms. The analyzed dataset included instances of two classes: benign and malicious. As the dataset contained twenty-two features, the feature selection techniques were required for dimensionality reduction. In these experiments, the Information gain, the Chi-square, and the F-test were applied to decrease the number of features to ten. The classes were also not completely balanced, so undersampling, oversampling, and synthetic minority oversampling (SMOTE) techniques were used to balance classes equally. The previous research works observed the classification of DDoS attacks applying various feature selection techniques and one or more machine learning algorithms. Still, they did not pay much attention to classifying the combinations of feature selection and balancing methods with different machine learning algorithms. This work is devoted to the classification of datasets with eight machine learning algorithms: naïve Bayes, logistic regression, support vector machine, k-nearest neighbors, decision tree, random forest, XGBoost, and CatBoost. In the experimental results, the Information gain and F-test feature selection methods achieved better performance with all eight ML algorithms than with the Chi-square technique. Furthermore, the accuracy values of the oversampled and SMOTE datasets were higher than that of the undersampled and imbalanced datasets. Among machine learning algorithms, the accuracy of support vector machine, logistic regression, and naïve Bayes fluctuates between 0.59 and 0.75, while decision tree, random forest, XGBoost, and

CatBoost allowed achieving values around 0.99 and 1.00 with all feature selection and class balancing techniques among all the algorithms.

Keywords: Internet security; networks; systems; DDoS; software-defined networking; feature selection; class balancing; machine learning; XGBoost; CatBoost

1 Introduction

Information security is a set of technologies and management methods required to guarantee the integrity, confidentiality, and availability of information data. Information security aims to protect the information, systems, networks, and applications from accidental or deliberate threats [1]. Today's online world is full of malware attacks [2], threats [3], cyberattacks [4], and scams [5]. However, implementing effective measures to protect and preserve information from such various threats is difficult since the variety of threats and attacks that can potentially damage information and computer networks is growing every day and uses more sophisticated attack methods. Fig. 1 shows the impact of information threats on information security criteria.

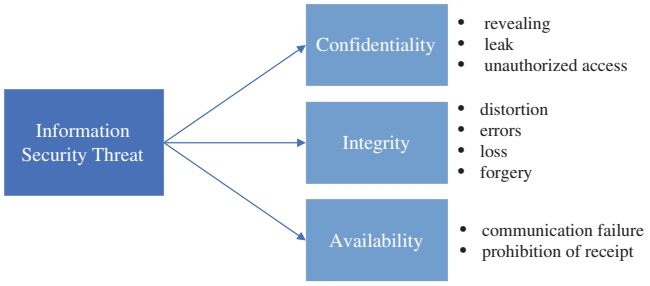

Figure 1: Influence of information threats on information security criteria

A common menace is a possibility of compromising information security in any form [6]. Various threats, attacks, and methods are used to access and damage the most important data. Phishing attacks, one of the most common threats over the years, pose a serious threat to information security, and the number of these threats is growing [7]. The main purpose of a phishing attack is to gain access to consumers' confidential personal data and financial information through various technical and social engineering methods [8]. The number of different methods and types of phishing methods is growing when the information technologies actively progress. Another popular type of software for unauthorized access to consumer information is spyware. These unwanted programs may compromise or steal the customer's information privacy. Spyware is not a direct attack by hackers but an unauthorized, covert installation on a computer. Such spyware provides remote access to the user's computer and information about its activities [9]. One of the most serious threats to attacks on Internet resources is a denial of service (DoS) attack. This type of attack does not directly damage information but disrupts services by attempting to disable the normal operation of a computer or network. The most advanced type of DoS attack is dedicated denial of services (DDoS). Unlike a DoS attack, a DDoS attack is performed from multiple devices and addresses at once, as shown in Fig. 2.

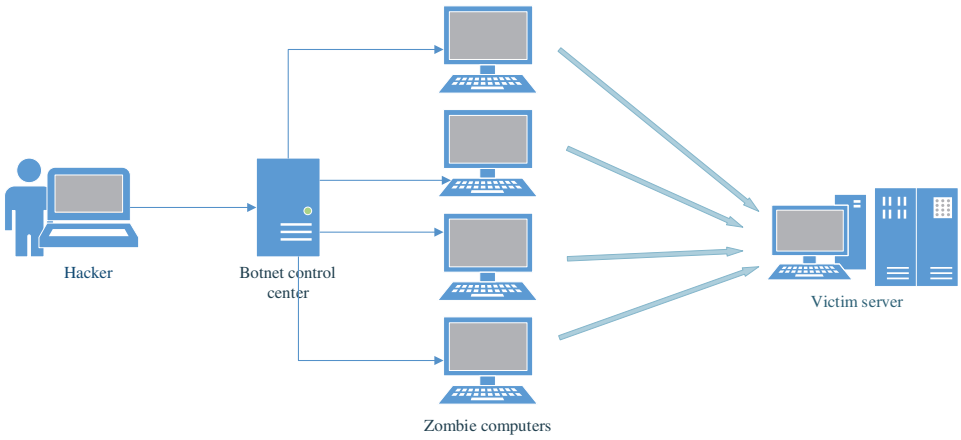

Figure 2: DDoS architecture

A DDoS attack is a hacker attack that paralyzes websites in a short period of time [10]. DDoS attacks are not intended to disrupt the victim's system but cause the service to fail [11]. The most serious malware among industrial hazards is called Flame. This program is a type of threat designed to steal and attack various valuable data. It is highly feasible because the Flame program has a variety of features, such as keyboard and network traffic control, voice recording, and screenshots. It uses a local area network (LAN) or USB storage to distribute it to different systems. Information security tools need to be strengthened to ensure information security in order to protect against various attacks, such as the threats described above. The DDoS attacks became common in the traditional and Software-Defined Networking (SDN) environments. In the traditional environment, they are directed on servers, while the attacks are executed on the controller in the SDN. The controller fails to provide services for the forwarding data packages under a DDoS attack in the SDN. At the same time, in traditional networks, the server completely stops providing services for users.

It is necessary to choose information security tools, considering the compatibility of their functions and the effectiveness of their use. Excessive reinsurance can lead to high costs. A variety of methods and tools are used to identify threats. Machine learning (ML) is one of the most important and effective methods [12]. This method is used to identify information system objects more accurately. The ML method is now popular for many real-time tasks using sophisticated algorithms. This field is closely related to computational statistics, making predictions using complex algorithms based on statistical data. ML uses data to determine the most effective algorithm based on the data's volume, quality, and nature. The development of ML algorithms for detecting Internet threats has demonstrated very effective results for classifying normal and malicious traffic in networks. Nevertheless, ML algorithms show good results with datasets that include a reasonable set of features.

Furthermore, new effective methods called neural networks started to be used to identify different kinds of threats [13–15]. Neural networks [13] are an ML discipline that mimics the way neurons work in the human brain. Neural networks [14] are specially proposed to determine the typical characteristics of system users and their statistically significant deviations. Neural networks consist of input, latent, and output nodes, and they represent the information that enters the network. Input nodes are associated with hidden nodes, and these input nodes receive the information passed to them. Each hidden node has a limit: It is activated if all aggregate inputs reach a certain value.

Despite the beginning of the use of ML and neural networks methods in the threat identification systems for the last years, the existing articles did not observe all of the aspects of DDoS attacks, all kinds of environments in which the network devices operate. Moreover, it is important to use four metrics such as accuracy, precision, recall, and F1-score to measure the efficiency of classification algorithms. Unfortunately, many research works display only one or two metrics in the experimental results section. In applying ML algorithms to DDoS threat classification, the feature selection techniques play an important role in the choice of the best features in the dataset. The list of the most popular methods includes the Information Gain (IG), Chi-square Test, F-test, Fisher's Score, Correlation Coefficient, Variance Threshold, etc. In the experimental part of this work, three feature selection techniques (IG, Chi-square, and F-test) are chosen for the evaluation. Another important problem that was not touched on in the previous research works of the DDoS threats classification is the imbalanced datasets. This problem occurs when there are unequal classes in the training dataset. This case decreases the values of evaluation metrics, and it is generally not a good situation in classification problems. Random oversampling, random undersampling, and synthetic minority oversampling (SMOTE) techniques are applied to solve this problem and make classes equal in size.

This paper is devoted to the supervised ML-based approach that is very rapid in computations and exhibits promising classification results. The rest of the paper is organized in the following way: Section 2 gives the literature review. Then the analyzed dataset, data scaling and feature selection techniques, class balancing, and ML algorithms are presented in Section 3. The experimental results, their analysis, and discussion are provided in Section 4. Finally, in Section 5, we briefly describe all the steps taken, suggest the best ML models, and outline directions for future research.

2 Literature Review

Cyber attackers usually update the software they use on a daily basis. Therefore, risk detection systems are developed daily to combat malware. To this end, there is a lot of literature research, and new research is being done to improve the performance of protection systems. In addition, there is a significant amount of research on identifying hazards using various ML methods. Therefore, this section focuses on observing ML and neural networks techniques to mitigate the existing threats. The research works devoted to the internet threats problems are shown in Tab. 1.

Table 1: Research works and their features

Study	Specifications	Advantages and drawbacks
Latchoumi et al. [16]	This paper explores the SQL injection threats, and the Support Vector Machine (SVM) is utilized to detect these kinds of threats and prevent them.	It significantly helps to identify SQL injection threats when a user logs in to a website, but it does not observe other ML algorithms. The feature selection techniques are also not provided.

(Continued)

Table 1: Continued

Study	Specifications	Advantages and drawbacks
Ucar et al. [17]	The paper proposes a model for anomalies detection in a firewall repository. The firewall logs are thoroughly analyzed and classified with such ML algorithms as naïve Bayes (NB), k-nearest neighbors (k-NN), Decision Tree (DT), and HyperPipes.	The F1-score is used to measure algorithms' performance in the experimental part. Among all algorithms, k-NN shows the best performance. Although the work provides the analysis with several ML algorithms, it does not describe or use any balancing technique.
Zargar et al. [18]	This research focuses on detecting smurf attacks with the PCA technique for dimensionality reduction and classification with the k-NN ML algorithm.	The research shows that attacks are effectively detected in 32.46 and 25.87 s in two experiments without PCA and with PCA, respectively. Thus, the presented results confirm the efficiency of the dimensionality reduction for the accuracy and computation time of intrusion detection. Nevertheless, the table of accuracy, precision, recall, and F1-score metrics is not provided in the experimental results section.
Wankhede et al. [19]	This paper is devoted to detecting DoS with the use of the Random Forest (RF) ML algorithm and the Multi-Layer Perceptron (MLP) neural network. The RF shows better results than MLP.	The list of features is provided in the paper, but the feature selection techniques are not presented. Nevertheless, the experimental results show interesting observations. Specifically, the RF demonstrates better results than MLP.

(Continued)

Table 1: Continued

Study	Specifications	Advantages and drawbacks
de Lima Filho et al. [20]	The research describes the DoS detection system. It conducts experiments on four datasets with the use of ML algorithms.	The work demonstrates the processing step, the feature selection phase, and the classification with ML models. The obtained results show an attack detection rate of 96% and high precision and accuracy values.
Khan et al. [21]	This paper describes a designed hybrid intrusion-detection model. This model uses a list of feature selection classifiers to improve detection rates and decrease false alarms.	The research uses several ML algorithms with feature selection techniques. The experimental results show that the performance of this approach is generally higher than that of each feature selection classifier working separately. It will also be better if it includes experiments with balancing techniques.

The DDoS attack consists of a large number of incoming packets that overload network resources. The server generally starts to drop the packets and becomes unavailable for other incoming legitimate packets for a definite period. As modern computer networks are commonly represented by the following list of main network devices such as hubs, switches, and routers, network management remains a challenging task. In order to overcome these difficulties, a new network approach, called Software-Defined Networking (SDN), where forwarding hardware is decoupled from the control decisions, is utilized. In this approach, the network functionalities are centralized in software-based controllers, and network devices can be programmed with an open interface. In [22], DDoS attacks in a Software-Defined Networking (SDN) environment are evaluated with the use of ML algorithms. RF, k-NN, and SVM algorithms are very efficient and show the values of accuracy, precision, recall, and F1-score above 98%. The observation of DDoS attacks in the SDN environment is also done in [23], where six characteristics of the switch flow table are extracted. A DDoS attack model is built with the application of the SVM classification algorithm. The SDN significantly simplifies network management and makes it very efficient. In the experimental results, this model achieves an accuracy of 95.24% with a small flow. [24] proposes a deep learning neural network model for detecting DDoS attacks with such performance metrics as an average delay, packet loss, packet delivery ratio, and throughput. The KDD Cup, SSE, and mixed datasets are utilized for the analysis of this model's performance. The suggested technique correspondingly shows 98.9%, 99%, and 98.1% accuracy values for the mentioned datasets. [25] uses a real-time solution to detect DDoS attacks in hardware. CAIDA

DDoS, MIT DARPA, and TUIDS datasets are used to evaluate the effectiveness of the proposed method. The experimental results demonstrate a very high accuracy of 99% for all three datasets with less than one microsecond to identify an incoming attack.

3 Methodology

This work performs the classification of DDoS attacks [26] in the SDN environment [27]. In the first step, the required dataset is chosen and thoroughly analyzed. In the second step, the categorical features in the dataset are encoded into numerical form. The optimal data scaling and feature selection techniques used for the dataset's normalization and suitable feature selection for the training model step are described in the third and fourth steps. Then the undersampling, oversampling, and SMOTE class balancing techniques are explained in detail. An important step of class balancing is realized, making classes equal in size. Finally, the principles of ML algorithms used in the experimental part are explained.

3.1 Dataset

There is a number of datasets containing information about various DDoS attacks [28,29] online, but the current work focuses on processing data for the attacks on the SDN. The corresponding dataset is shared by the following link by its authors [30]. This dataset includes benign TCP, UDP, and ICMP traffic and malicious traffic that presents the collection of TCP Syn, UDP flood, and ICMP attacks. It consists of 23 features extracted from switches. The list of the features and their descriptions are presented in Tab. 2. The dataset includes 104345 rows with 63335 benign and 40504 malicious labels (Fig. 3).

Table 2: Features of the dataset

No	Feature name	Description
1	Dt	The date and time which has been converted into a number
2	Switch	Switch number
3	Src	Source IP
4	Dst	Destination IP
5	Pktcount	Packets in a flow
6	Bytecount	Bytes in a flow
7	dur	Duration in seconds
8	dur_nsec	Duration in nanoseconds
9	tot_dur	A sum of duration in seconds and nanoseconds
10	Flows	Number of flows
11	Packetins	Number of packets in a message
12	Pktperflow	Packet count during a single flow
13	Byteperflow	Byte count during a single flow
14	Pktrate	Number of packets sent per second
15	Pairflow	Number of flow's pairs
16	Protocol	Type of protocol (TCP, UDP, ICMP)

(Continued)

Table 2: Continued

No	Feature name	Description
17	port_no	Port number
18	tx_bytes	Data transfer rate in bytes
19	rx_bytes	Data receiving rate in bytes
20	tx_kbps	Data transfer rate in kilobytes
21	rx_kbps	Data receiving rate in kilobytes
22	tot_kbps	The sum of tx_kbps and rx_kbps
23	Label	Class label which indicates whether the traffic type is benign (0) or malicious (1)

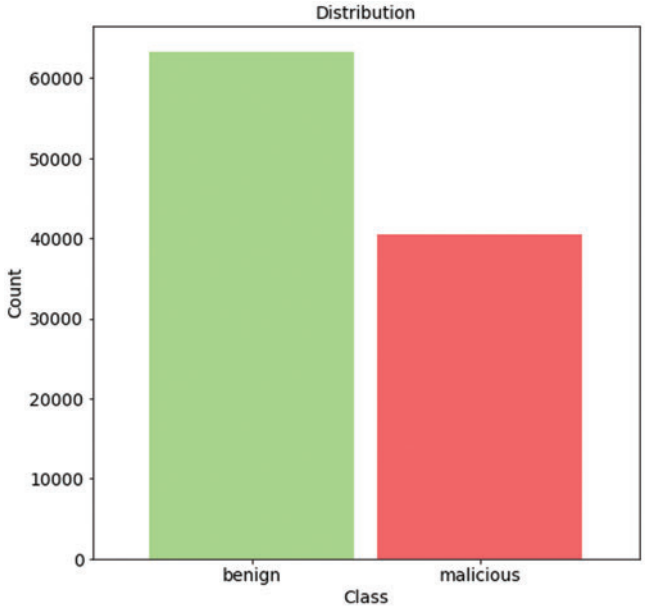

Figure 3: Benign and malicious classes

3.2 Label Encoding

In the presented dataset, some of the features like the source IP address, the destination IP address, and the protocol are categorical. It is necessary to transform them into a numerical form before applying the scaling step. Therefore, the categorical features are replaced with a numerical value between 0 and the number of classes minus 1.

3.3 Data Scaling

The values of the dataset's features are measured at different scales, and they do not contribute equally to the model fitting. Therefore, if an ML model is trained with these features unchanged, it can create a bias, making the model unprecise. Normalization techniques [31] are used to deal with this problem. Mean normalization, Min-Max normalization, and Standardization are the most frequently used scaling methods.

Mean normalization is calculated by the following formula

$$x' = \frac{x - \bar{x}}{\max(x) - \min(x)}, \tag{1}$$

where, x is an initial value, $\bar{x}$ is a mean value, and x' is a normalized value.

Min-Max normalization is a method that rescales the features to the range in [0,1]. This normalization is calculated by the formula

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{2}$$

where, x is an initial value and x' is a normalized value.

Standardization is a scaling method where the values are placed around the mean and divided by the standard deviation. It is calculated by the formula

$$x' = \frac{x - \bar{x}}{\sigma}, \tag{3}$$

where, x is an initial value, $\bar{x}$ is a mean value, x' is a normalized value, and σ is a standard deviation.

3.4 Feature Selection Techniques

Building an ML model almost rarely requires the use of all features. When redundant features are added to the model, it increases the complexity, the cost, and the running time. Therefore, feature selection techniques [32] such as the Information gain (IG), the Chi-square, and the F-test are used to overcome this problem.

The IG measures a connection between each feature in the context of the target feature. It is presented by the following formula

$$I(X; Y) = H(Y) - H(Y|X), \tag{4}$$

The Chi-square is utilized for testing the independence of two events. Having two features is necessary to get count O and expected count E. The Chi-square estimates how E and O deviate from each other.

$$x_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}, \tag{5}$$

where, O is observed values, E is expected values, and c is degrees of freedom.

The F-test is a statistical test that computes the ratio between variances values. The results of the test are effectively used for feature selection. The F-test is calculated by the formula

$$F = \frac{MST}{MSE}, \tag{6}$$

where, MST is mean square treatments, and MSE is a mean square error.

3.5 Class Balancing

A class imbalance is a term that determines that the number of elements in one class is higher than in the other class. The class imbalance is commonly a big problem in ML because it increases

the model's accuracy by straightly labeling all elements as a majority class. However, it performs weakly in classifying the other class, and the values of precision, recall, and F1-score become lower than the accuracy. Generally, the class imbalance appears in such domains as spam classification, fraud detection, disease screening, and DDoS attacks. Three effective class balancing techniques called random oversampling, random undersampling (Fig. 4), and synthetic minority oversampling (SMOTE) are used to overcome this problem [33].

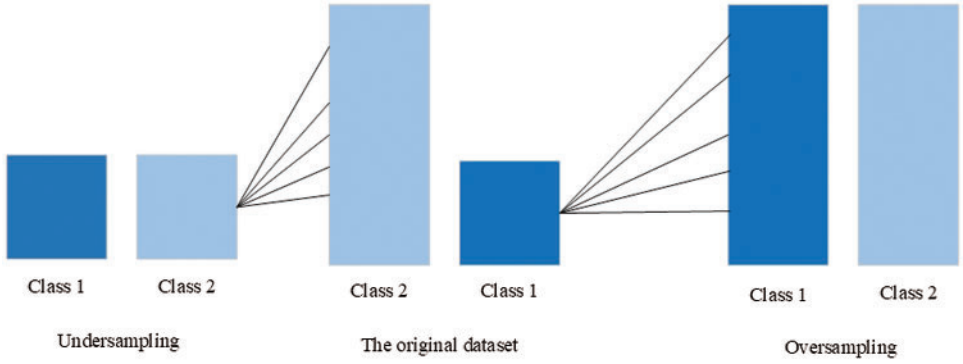

Figure 4: Class balancing: random undersampling and random oversampling

In random oversampling, elements from the minority class are randomly selected for duplication to make this class equal to the majority class. In random undersampling, the opposite operation is done. Random elements in the majority class are deleted to decrease the size and equalize it with the minority class. The disadvantage of undersampling is the loss of a large part of valuable data. In random oversampling, oppositely, the important information is kept.

SMOTE is another very efficient oversampling technique where the synthetic elements are generated for the minority class. This algorithm focuses on the feature space for generating new elements that are synthesized between the existing ones. The created elements also preserve very valuable information.

3.6 Machine Learning Algorithms

ML text classification has been implemented with the following algorithms: NB, SVM, Logistic regression (LR), k-NN, DT, RF, XGBoost, and CatBoost. These algorithms were chosen because they are considered advanced and widely used for data classification tasks.

An NB classifier [8] uses the Bayes' theorem as a probabilistic model for classification. An important assumption that the features are independent is used here. That is the reason this algorithm is called naïve. The Bayes formula is written below

$$p(y|X) = \frac{P(X|y) \times P(y)}{P(X)}, \tag{7}$$

where $X = (x_1, x_2, x_3, \ldots, x_n)$, and $x_1, x_2, x_3, \ldots, x_n$ is a list of features of the dataset. The expansion of the chain rule gives the following formula

$$P(y|x_1, x_2, \ldots, x_n) = \frac{P(x_1|y) \times P(x_2|y) \times \ldots \times P(x_n|y) \times P(y)}{P(x_1) \times P(x_2) \times \ldots \times P(x_n)} \tag{8}$$

As the denominator does not change for all entries in the dataset, it can be removed.

An SVM classifier [11] defines a hyperplane, dividing the input data into several classes. This hyperplane tries to separate the data in the best way. The main objective is to find the hyperplane with the maximum distance between data points of two classes (Fig. 5). The hyperplane is defined by the following formula

$$y_i(\overrightarrow{w} \times \overrightarrow{x} + b) \geq 0 \tag{9}$$

where, $\overrightarrow{x} = (x_1, x_2, \ldots, x_n)$ is an input vector; $\overrightarrow{w} = (w_1, w_2, \ldots, w_n)$ is a weight vector; y_i is an output value; b is a bias. If it is more than or equal to zero, the value belongs to a positive class. If not, it belongs to a negative class.

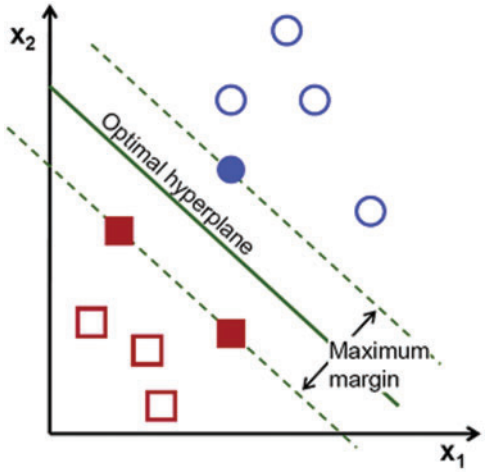

Figure 5: The hyperplane separating classes

An LR is an ML algorithm used for classification which prediction is based on the probability of an outcome by fitting data to a logistic function.

$$p(x) = \frac{1}{1 + e^{-f(x)}}, \tag{10}$$

where, $f(x) = w_0 + w_1x_1 + \ldots + w_rx_r$ is a function and $w_0, w_1, \ldots, w_r$ are the corresponding weights. The range of values $p(x)$ is between 0 and 1. It goes to class 1 if the value is close to 0. Otherwise, it goes to class 2.

A k-NN [8] is a distance ML algorithm for classifying data points. The algorithm finds the distance between an unclassified data point and all pre-classified data points. Then k points with the smallest distances are chosen, and the class is defined by the class that appeared the most times. The Euclidean distance formula commonly measures the distance between data points

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}, \tag{11}$$

where, $d(x, y)$ is the distance between two points; x_i and y_i are the feature vectors of x and y points correspondingly; n is a length of the feature vector.

A DT is a popular and widely used ML algorithm for data classification. A DT represents a structure with N nodes containing the conditions related to the features of the points in the dataset. First, the points whose feature values satisfy this condition are put to one side of the tree. Otherwise,

they are put to the other side of the tree. This process continues while propagating through the whole built tree towards its leaf nodes. An RF (Fig. 6) is an ensemble method of DTs [19]. Each DT classifies a new data point independently, and the class is defined by the largest number of votes of all trees in the ensemble.

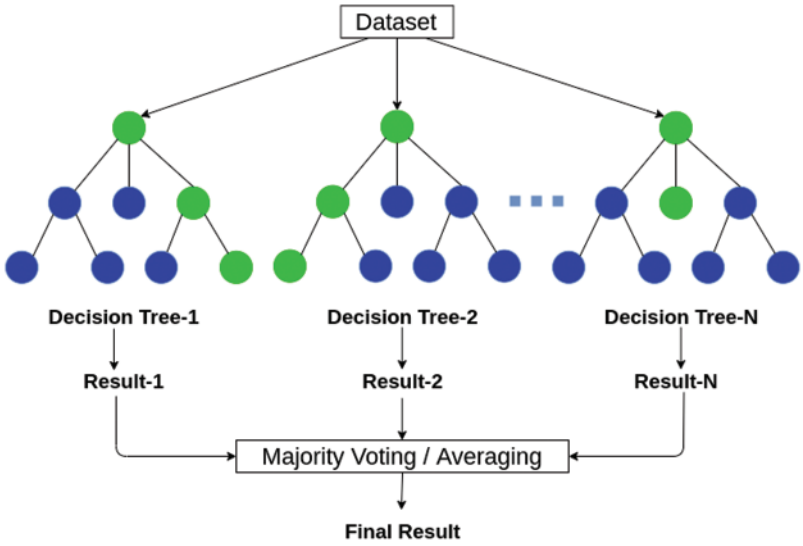

Figure 6: An RF classifier

XGBoost [33] is one of the most advanced ML algorithms released in 2014. It provides a parallel tree boosting and is significantly efficient in performance. One specification of the algorithm is that diversions of ensemble predictions are calculated at each iteration. CatBoost was developed by Yandex in 2017. This algorithm is also based on gradient boosting and focuses on categorical features in a dataset. It is also very fast and effective, allowing GPU usage during the training step.

4 Experiments and Discussion

The experimental part utilized the Python programming language with Scikit-learn, Imbalance-learn, Matplotlib, and Seaborn libraries.

First, the categorical features of the dataset were encoded with the label encoding technique. Then all features were scaled with the Min-Max normalization. The IG, Chi-square, and F-test feature selection techniques were applied to the dataset getting the ten most important features. The dataset was balanced with undersampling, oversampling, and SMOTE techniques, randomly divided into training 70% and testing 30% parts, and classified with eight ML algorithms from Section 3.5. The hold-out split method instead of k-fold cross-validation was chosen because the classification with three feature selection techniques, four balancing methods, and eight ML algorithms requires much time. The cross-validation would take a decent time to run all the experiments and compose all the obtained data together.

The performance was evaluated by accuracy, precision, recall, and F1-score measures [26].

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{12}$$

$$precision = \frac{TP}{TP + FP}, \tag{13}$$

$$recall = \frac{TP}{TP + FN}, \tag{14}$$

$$F1\_score = 2\,\frac{precision \times recall}{precision + recall}, \tag{15}$$

where, *TP* (true positive) is a correctly classified *positive* instance; *TN* (true negative) is a correctly classified *negative* instance; *FP* (false positive) is a wrongly classified *positive* instance; *FN* (false negative) is a wrongly classified *negative* instance.

The classification of the dataset with Chi-square, IG, and F-test feature selection methods is presented in Tab. 3.

Table 3: Classification of the imbalanced and oversampled datasets

	Chi-square (Imbalanced)	IG (Imbalanced)	F-test (Imbalanced)	Chi-square (Oversampled)	IG (Oversampled)	F-test (Oversampled)
SVM						
Accuracy	0.69	0.74	0.76	0.67	0.74	0.72
Precision	0.61	0.73	0.75	0.65	0.74	0.75
Recall	0.56	0.54	0.59	0.71	0.75	0.66
F1-score	0.59	0.62	0.66	0.68	0.75	0.71
LR						
Accuracy	0.65	0.75	0.73	0.65	0.74	0.72
Precision	0.56	0.71	0.68	0.64	0.74	0.73
Recall	0.50	0.60	0.57	0.68	0.75	0.71
F1-score	0.53	0.65	0.62	0.66	0.74	0.72
NB						
Accuracy	0.59	0.63	0.60	0.62	0.64	0.64
Precision	0.46	0.55	0.48	0.64	0.66	0.64
Recall	0.30	0.21	0.27	0.58	0.59	0.64
F1-score	0.37	0.30	0.35	0.60	0.62	0.64
k-NN						
Accuracy	0.97	0.99	0.99	0.98	0.99	0.99
Precision	0.97	0.98	0.99	0.98	0.99	0.99
Recall	0.96	0.98	0.98	0.98	0.99	0.99
F1-score	0.96	0.98	0.98	0.98	0.99	0.99
DT						
Accuracy	0.99	1.00	0.99	0.99	1.00	0.99
Precision	0.99	1.00	0.99	0.99	1.00	0.99
Recall	0.99	1.00	1.00	0.99	1.00	1.00
F1-score	0.99	1.00	0.99	0.99	1.00	0.99

(Continued)

Table 3: Continued

	Chi-square (Imbalanced)	IG (Imbalanced)	F-test (Imbalanced)	Chi-square (Oversampled)	IG (Oversampled)	F-test (Oversampled)
RF						
Accuracy	0.99	1.00	1.00	0.99	1.00	1.00
Precision	0.99	1.00	1.00	0.99	1.00	1.00
Recall	0.99	1.00	1.00	0.99	1.00	1.00
F1-score	0.99	1.00	1.00	0.99	1.00	1.00
CatBoost						
Accuracy	0.99	1.00	0.99	0.99	1.00	1.00
Precision	0.99	1.00	0.99	0.99	1.00	1.00
Recall	0.99	1.00	1.00	0.99	1.00	1.00
F1-score	0.99	1.00	0.99	0.99	1.00	1.00
XGBoost						
Accuracy	0.96	0.99	0.99	0.96	0.99	0.99
Precision	0.95	0.99	0.97	0.93	0.99	0.98
Recall	0.94	0.99	0.99	0.99	0.99	0.99
F1-score	0.95	0.99	0.98	0.96	0.99	0.99

The experimental results showed that the IG and F-test feature selection methods achieved the best performance metrics with all ML algorithms. In addition, the processing of the oversampled dataset gave better results with SVM, LR, and NB ML algorithms than the imbalanced dataset. Among ML models [27], DT, RF, XGBoost, and CatBoost demonstrated significantly better results than NB, SVM, and LR. In most experiments, the accuracy, precision, recall, and F1-score values reached 0.99 and 1.00.

The classification of the dataset with Chi-square, IG, and F-test feature selection methods is presented in Tab. 4.

Table 4: Classification of the undersampled and SMOTE datasets

	Chi-square (Undersampled)	IG (Undersampled)	F-test (Undersampled)	Chi-square (SMOTE)	IG (SMOTE)	F-test (SMOTE)
SVM						
Accuracy	0.67	0.75	0.73	0.66	0.74	0.72
Precision	0.65	0.74	0.75	0.65	0.74	0.75
Recall	0.73	0.76	0.68	0.71	0.75	0.67
F1-score	0.69	0.75	0.71	0.67	0.74	0.70
LR						
Accuracy	0.65	0.74	0.73	0.65	0.74	0.72
Precision	0.64	0.74	0.73	0.64	0.74	0.73
Recall	0.69	0.75	0.72	0.68	0.74	0.71
F1-score	0.67	0.74	0.73	0.66	0.74	0.72
NB						
Accuracy	0.63	0.65	0.65	0.62	0.64	0.64

(Continued)

Table 4: Continued

	Chi-square (Undersampled)	IG (Undersampled)	F-test (Undersampled)	Chi-square (SMOTE)	IG (SMOTE)	F-test (SMOTE)
Precision	0.64	0.66	0.65	0.64	0.66	0.64
Recall	0.59	0.60	0.65	0.58	0.59	0.64
F1-score	0.61	0.63	0.65	0.61	0.62	0.64
k-NN						
Accuracy	0.97	0.98	0.98	0.98	0.99	0.99
Precision	0.97	0.98	0.98	0.98	0.98	0.99
Recall	0.97	0.99	0.98	0.98	0.99	0.99
F1-score	0.97	0.98	0.98	0.98	0.99	0.99
DT						
Accuracy	0.99	0.99	0.99	0.99	0.99	0.99
Precision	0.99	0.99	0.99	0.99	1.00	1.00
Recall	0.99	1.00	0.99	0.99	0.99	0.99
F1-score	0.99	0.99	0.99	0.99	0.99	0.99
RF						
Accuracy	0.99	1.00	0.99	0.99	0.99	0.99
Precision	0.99	1.00	0.99	0.99	1.00	0.99
Recall	0.99	1.00	0.99	0.99	0.99	0.99
F1-score	0.99	1.00	0.99	0.99	0.99	0.99
CatBoost						
Accuracy	0.99	1.00	0.99	0.99	0.99	0.99
Precision	0.99	1.00	0.99	0.99	1.00	1.00
Recall	0.99	1.00	1.00	0.99	0.99	0.99
F1-score	0.99	1.00	0.99	0.99	0.99	0.99
XGBoost						
Accuracy	0.96	0.99	0.99	0.96	0.99	0.99
Precision	0.93	0.99	0.98	0.93	0.99	0.98
Recall	0.98	0.99	0.99	0.99	0.99	0.99
F1-score	0.96	0.99	0.99	0.96	0.99	0.99

The results of the classification of undersampled and SMOTE datasets proved that the IG and F-test feature selection techniques allowed to achieve superior results than the Chi-square feature selection. DT, RF, k-NN, CatBoost, and XGBoost ML algorithms also classified these datasets better than other algorithms. The experiments support the statements that these algorithms are the most advanced in classifying Internet threats.

The obtained experimental results generally revealed that the imbalanced data classification showed the lowest performance in all three feature selection techniques compared to models trained on the oversampled, undersampled, and SMOTE models.

5 Conclusion

As the number of different threats is growing, and the earlier methods of the systems' protection are becoming less effective and more vulnerable to the various attacks, a need for more advanced methods appeared. One of these most serious threats is a DDoS attack [28] that disables the normal operation of servers, networks, and systems. This work observed DDoS attacks in the SDN network [29], where all the functionalities are centralized in software-based controllers. DDoS attacks are

especially dangerous for the SDN network [30,34], and an effective approach is required to detect them accurately and fast. ML algorithms proved to be very useful in revealing malicious traffic in these kinds of networks.

The dataset containing benign and malicious traffic instances was processed and analyzed in the experiments. The IG, Chi-square, and F-test feature selection methods retrieved the most important features. Then three balancing techniques were used to balance the classes, and eight very efficient ML algorithms (NB, SVM, LR, k-NN, DT, RF, XGBoost, and CatBoost) were applied to train the classification models. The classification performance was evaluated by accuracy, precision, recall, and F1-score measures. DT, RF, k-NN, XGBoost, and CatBoost ML algorithms showed the best results with all feature selection and class balancing techniques with the accuracy, precision, recall, and F1-score values of 0.99 and 1.00.

In future works, the ML classifiers will be tested on the datasets containing different kinds of Internet threats such as smurf, phishing, man-in-the-middle, SQL injection, password attacks, and others.

Acknowledgement: We would like to thank colleagues for their support.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. M. Balaji and T. Subbulakshmi, "Malware analysis using classification and clustering algorithms," *International Journal of e-Collaboration*, vol. 18, no. 1, pp. 1–26, 2022.

[2] A. Kamath, V. Bhatu, T. Paranjape and R. Sawant, "Malware classification and defense against adversarial attacks," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 91. Singapore: Springer, pp. 267–274, 2021.

[3] A. Aljohani and A. Bushnag, "An intrusion detection system model in a local area network using different machine learning classifiers," in *11th Int. Conf. on Advanced Computer Information Technologies (ACIT)*, Deggendorf, Germany, pp. 483–488, 2021.

[4] J. Cabrero-Holgueras and S. Pastrana, "A methodology for large-scale identification of related accounts in underground forums," *Computers & Security*, vol. 111, pp. 1–15, 2021.

[5] J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran and B. S. Bindhumadhava, "Phishing website classification and detection using machine learning," in *Int. Conf. on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, pp. 1–6, 2020.

[6] Z. Wang, K. W. Fok and V. L. L. Thing, "Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study," *Computers & Security*, vol. 113, pp. 1–15, 2022.

[7] A. Razaque, M. B. H. Frej, D. Sabyrov, A. Shaikhyn, F. Amsaad *et al.,* "Detection of phishing websites using machine learning," in *IEEE Cloud Summit*, Harrisburg, PA, USA, pp. 103–107, 2020.

[8] R. J. R. Raj, S. Srinivasulu and A. Ashutosh, "A Multi-classifier framework for detecting spam and fake spam messages in twitter," in *IEEE 9th Int. Conf. on Communication Systems and Network Technologies (CSNT)*, Gwalior, India, pp. 266–270, 2020.

[9] V. Mahesh and S. Devi K. A., "Detection and prediction of spyware for user applications by interdisciplinary approach," in *Int. Conf. on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)*, Keonjhar, India, pp. 1–6, 2020.

[10] M. A. Haq, M. Abdul and T. AL-Harbi, "Development of PCCNN-based network intrusion detection system for EDGE computing," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1769–1788, 2021.

[11] A. Chartuni and J. Márquez, "Multi-classifier of DDoS attacks in computer networks built on neural networks," *Applied Sciences*, vol. 11, no. 22, pp. 1–15, 2021.
[12] M. Marvi, A. Arfeen and R. Uddin, "A generalized machine learning-based model for the detection of DDoS attacks," *International Journal of Network Management*, vol. 31, no. 6, pp. 1–20, 2020.
[13] Y. Tang, L. Gu and L. Wang, "Deep stacking network for intrusion detection," *Sensors*, vol. 22, no. 1, pp. 1–17, 2022.
[14] S. ur Rehman, M. Khaliq, S. I. Imtiaz, A. Rasool, M. Shafiq *et al.,* "DIDDOS: An approach for detection and identification of distributed denial of service (DDoS) cyberattacks using gated recurrent units (GRU)," *Future Generation Computer Systems*, vol. 118, pp. 453–466, 2021.
[15] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu *et al.,* "AE-MLP: A hybrid deep learning approach for DDoS detection and classification," *IEEE Access*, vol. 9, pp. 146810–146821, 2021.
[16] T. P. Latchoumi, M. S. Reddy and K. Balamurugan, "Applied machine learning predictive analytics to SQL injection attack detection and prevention," *European Journal of Molecular & Clinical Medicine*, vol. 7, no. 2, pp. 3543–3553, 2020.
[17] E. Ucar and E. Ozhan, "The analysis of firewall policy through machine learning and data mining," *Wireless Pers Commun*, vol. 96, pp. 2891–2909, 2017.
[18] G. R. Zargar and P. Kabiri, "Identification of effective network features to detect smurf attacks," in *IEEE Student Conf. on Research and Development (SCOReD)*, Serdang, Malaysia, pp. 49–52, 2009.
[19] S. Wankhede and D. Kshirsagar, "DoS attack detection using machine learning and neural network," in *Fourth Int. Conf. on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, pp. 1–5, 2018.
[20] F. S. de Lima Filho, F. A. F. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar and L. F. Silveira, "Smart detection: An online approach for DoS/DDoS attack detection using machine learning," *Security and Communication Networks*, vol. 2019, pp. 1–15, 2019.
[21] J. A. Khan and J. Nitesh, "Improving intrusion detection system based on KNN and KNN-DS with detection of U2R, R2L attack for network probe attack detection," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 5, pp. 209–212, 2016.
[22] F. Musumeci, A. C. Fidanci, F. Paolucci, F. Cugini and M. Tornatore, "Machine-learning-enabled DDoS attacks detection in P4 programmable networks," *Journal of Network and Systems Management*, vol. 30, no. 1, pp. 1–27, 2022.
[23] J. Ye, X. Cheng, J. Zhu, L. Feng and L. Song, "A DDoS attack detection method based on SVM in software defined network," *Security and Communication Networks*, vol. 2018, pp. 1–8, 2018.
[24] S. Sumathi and N. Karthikeyan, "Detection of distributed denial of service using deep learning neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 6, pp. 5943–5953, 2021.
[25] N. Hoque, H. Kashyap and D. K. Bhattacharyya, "Real-time DDoS attack detection using FPGA," *Computer Communications*, vol. 110, pp. 48–58, 2017.
[26] N. Ahuja, G. Singal, D. Mukhopadhyay and N. Kumar, "Automated DDOS attack detection in software defined networking," *Journal of Network and Computer Applications*, vol. 187, pp. 1–15, 2021.
[27] A. Banitalebi Dehkordi, M. Soltanaghaei and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *The Journal of Supercomputing*, vol. 77, no. 3, pp. 2383–2415, 2021.
[28] M. W. Nadeem, H. G. Goh, V. Ponnusamy and Y. Aun, "DDoS detection in SDN using machine learning techniques," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 771–789, 2022.
[29] A. Mishra, N. Gupta and B. B. Gupta, "Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller," *Telecommunication Systems*, vol. 77, no. 1, pp. 47–62, 2021.
[30] N. Ahuja, G. Singal and D. Mukhopadhyay, "DDOS attack SDN dataset," *Mendeley Data, V1*, https://doi.org/10.17632/jxpfjc64kr.1. 2020.
[31] H. Nugroho, N. P. Utama and K. Surendro, "Normalization and outlier removal in class center-based firefly algorithm for missing value imputation," *Journal of Big Data*, vol. 8, no. 1, pp. 1–18, 2021.

[32] R. Ahmad, R. Wazirali, Q. Bsoul, T. Abu-Ain and W. Abu-Ain, "Feature-selection and mutual-clustering approaches to improve DoS detection and maintain WSNs' lifetime," *Sensors*, vol. 21, no. 14, pp. 1–25, 2021.

[33] G. Usha, M. Narang and A. Kumar, "Detection and classification of distributed DoS attacks using machine learning," in *Computer Networks and Inventive Communication Technologies*, Coimbatore, India, pp. 985–1000, 2021.

[34] Y. Begimbayeva, O. Ussatova, R. Biyashev and S. Nyssanbayeva, "Development of an automated system model of information protection in the cross-border exchange," *Cogent Engineering*, vol. 7, no. 1, pp. 1–13, 2020.