Measurement: Sensors 32 (2024) 101037

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Enhancing DDoS attack detection with hybrid feature selection and ensemble-based classifier: A promising solution for robust cybersecurity

Md. Alamgir Hossain [a,b,*], Md. Saiful Islam [a]

[a] Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh
[b] Department of Computer Science and Engineering, Prime University, Dhaka, Bangladesh

ARTICLE INFO

Keywords:
DDoS attack detection
Hybrid feature selection to identify DDoS attacks
Ensemble-based approach to detect DDoS attacks
Ensemble random forest in cybersecurity

ABSTRACT

Distributed denial-of-service (DDoS) attacks pose a significant threat to computer networks and systems by disrupting services through the saturation of targeted systems with traffic from multiple sources. Real-time detection of these attacks has become a critical cybersecurity task. However, current DDoS attack detection methods suffer from high false positive rates and limited ability to capture the complex patterns of attack traffic. This research proposes an enhanced approach for detecting DDoS attacks using a hybrid feature selection technique in combination with an ensemble-based classifiers. The ensemble-based approach aggregates many decision trees to increase classification accuracy and reduce overfitting and model robustness. The feature selection technique uses correlation analysis, mutual information, and principal component analysis to identify the most useful characteristics for attack detection. The ensemble-based Random Forest classifier from the various ensemble-based approaches with the specified relevant features produces the best detection rates. Many datasets related to identifying DDoS attacks are used to evaluate the proposed model, and experimental findings demonstrate that it surpasses existing techniques in terms of accuracy, recall, precision, f1-score, and false positive rate, with other evaluation metrics. The proposed approach achieves almost 100 % accuracy, 100 % true positive rate, and 0 % error rate making it a promising solution for DDoS attack detection.

1. Introduction

A distributed denial of service (DDoS) attack uses a large number of compromised devices, sometimes those that are part of a botnet, to overload a targeted system or network with traffic and render it inaccessible to authorized users [1,2]. The goal of a DDoS attack is to disrupt the normal functioning of the target system or network, denying access to its intended users [3,4]. In this DDoS attack, the attacking devices may be compromised computers, routers, or IoT devices that have been infected with malware or taken over by an attacker. These devices are then directed to send a large volume of traffic to the target system or network, making it unable to respond to legitimate requests. DDoS assaults can originate from any location in the world, and since they are widespread, it may be difficult to effectively prevent or stop them. They are frequently employed by hackers or other criminals to demand money or to obstruct the work of a company, government, or organization. DDoS attacks can cause significant harm, including financial losses, reputational harm, and even legal consequences [5,6].

DDoS attacks raise significant ethical and legal concerns due to their potential to harm sensitive data and jeopardize user information. These attacks are on the rise in terms of both frequency and sophistication, which makes their identification and mitigation increasingly challenging [7]. Attackers employ a variety of techniques and technologies, and the impact of DDoS attacks extends beyond the targeted organization. For instance, an attack on a critical infrastructure provider can have a far-reaching impact, affecting other organizations, governments, and individuals. Consequently, addressing the DDoS attack problem is not only essential for individual enterprises but also for the broader community and society at large. Thus, there is a pressing need to develop effective methods and tools to detect and minimize DDoS attacks [8,9]. As DDoS attacks continue to grow in complexity, they pose challenges for mitigation. Countermeasures are difficult to implement because these attacks can target multiple network levels and originate from diverse sources [10]. Moreover, distinguishing genuine traffic from attack traffic remains a challenge. To effectively reduce the impact of DDoS attacks, innovative and collaborative approaches are required to

* Corresponding author. Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh.
E-mail addresses: alamgir.cse14.just@gmail.com (Md.A. Hossain), mdsaifulislam@iict.buet.ac.bd (Md.S. Islam).

https://doi.org/10.1016/j.measen.2024.101037
Received 24 May 2023; Received in revised form 17 October 2023; Accepted 21 January 2024
Available online 6 February 2024
2665-9174/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

address these evolving challenges [11,12].

In this research, we propose an improved method for identifying DDoS attacks by combining a novel feature selection method with ensemble-based classifier. Correlation analysis, mutual information, and principal component analysis are all integrated for the goal of selecting significant features. The random forest classifier is then used to the model from among the numerous ensemble-based machine learning techniques. The proposed approach aims to improve DDoS attack detection accuracy while decreasing false positives. We evaluate the proposed approach using various real-world datasets and show that all outperform existing techniques in terms of accuracy, recall, precision, and other evaluation metrics. Our key contributions to this research are listed below.

- Researching the most recent DDoS attack detection techniques and evaluating their advantages and disadvantages.
- Developing a combined approach for selecting the most relevant features.
- Evaluate the performance of several classical machine-learning techniques for identifying DDoS attacks with the model.
- Developing a machine-learning framework based on ensembles that combine several classifiers to increase detection accuracy.
- Evaluating the effectiveness of the proposed ensemble-based strategy against current DDoS detection techniques for different publicly available datasets.

The ability of the ensemble-based random forest strategy to integrate the predictions of many decision trees to increase the classification accuracy makes it superior to other approaches for detecting attacks involving DDoS. By using an ensemble of classifiers rather than a single classifier, the random forest technique can reduce the variance and bias of the classifier, prevent overfitting, and boost the resilience of the model [13]. Additionally, the ensemble-based methodology may more accurately detect assaults than previous techniques since it is better able to capture the complex structures of DDoS attack flow. Furthermore, the ensemble-based technique is highly suited for identifying attacks using DDoS in real-time environments due to its scalability and ability to effectively handle massive volumes of data [14]. Table 1 encompasses the abbreviated terminology employed throughout the research.

The remaining section of this paper discusses related research for detecting DDoS attacks. In the next section, every part of the proposed model development is described. The fourth section contains the results and discussion along with essential figures and tables. The fifth section of the paper addresses the conclusion of this research.

Table 1
Notations and abbreviations to increase conciseness and clarity.

Notations	Abbreviations
DDoS	Distributed Denial-of-Service Attack
HFS	Hybrid Feature Selection
ERF	Ensemble-based Random Forest
SVC	Support Vector Classification
CNN	Convolutional Neural Network
KNN	K-Nearest Neighbors
SDN	Software-Defined Network
IDS	Intrusion Detection System
GHLBO	Gradient Hybrid Leader Optimization
DSA	Deep Stacked Autoencoder
GBT	Gradient Boosted Trees
TPR	True Positive Rate
FPR	False Positive Rate
ROC	Receiver Operating Characteristic
CA	Correlation Analysis
MI	Mutual Information
PCA	Principal Component Analysis

2. Related works

Over time, a number of techniques, such as rule-based, statistical, machine-learning-based, etc., have been suggested to identify attacks involving DDoS. This section reviews the latest DDoS attack detection techniques as well as their advantages and disadvantages.

A collection of rules is developed in rule-based strategies to identify attacks using DDoS. These rules are frequently based on the traffic flow factors, such as packet rate, packet size, and protocol type. Despite being clear and easy to use, rule-based approaches might not be able to recognize innovative or sophisticated DDoS assaults that do not adhere to specified criteria. Rule-based techniques may also mistake legitimate traffic for an attack due to their high false positive rate [15,16]. Statistical methods use statistical models to identify anomalies in network traffic. These techniques look at the flow of the traffic and look for deviations from the norm. Because statistical methods can recognize both known and unknown assaults, they are preferable to rule-based systems. Statistical methods may require a large amount of training data to comprehend normal traffic patterns, which may be challenging to get. Additionally, statistical techniques may produce a large percentage of false alarms, leading to the labeling of genuine traffic as an attack [17,18].

In order to identify DDoS attacks, machine-learning approaches utilize different classifiers to understand the patterns of both legitimate and malicious traffic. The multi-scale base CNN technique presented by Cheng et al. [19] in 2020 to identify DDoS obtained 74 % accuracy, which is quite low, and a very low TPR. The same year, Sambangi and Gondi [20] introduced a method of multiple linear regression with a 75 % accuracy rate and a very high FPR. Saini et al. [21] introduced a machine learning model that utilizes the J48 classifier to identify HTTP Flood, Smurf, UDP Flood, and SIDDoS type DDoS attacks. One noteworthy limitation was the relatively small dataset employed for training and testing the model. Additionally, the research reported a significant issue with a high FPR in their model's performance.

An IDS framework that integrates a group of feature engineering methods with the use of deep neural networks was suggested by Lopes et al. [22] in 2021. Nearly 99 % accuracy was attained. Despite having an IDS framework, it can only identify DDoS attacks. For the SDN environment, Rajesh et al.'s approaches using Random forest give 97 % accuracy for DDoS attack detection [23]. To detect DDoS attacks, Dasari and Devarakonda suggested yet another ML-based model. They used different single ML-based classifiers [24]. The model with logistic regression produced the best result from the performance study, with an accuracy of 99.61 %. It is about 84 % in this case of specificity. Since it uses a single classifier, the performance varies depending on the type of DDoS attack. SVC-RF-based classifiers with a 98.8 % accuracy were proposed by Ahuja et al. Only the SDN environment is suitable for this complex model [25].

In 2022, Nuiaa et al. [26] suggested improved optimization techniques for the detection of DDoS attacks. The model with the KNN classifier produced a result of 89.59 %. Regarding their research, they recommended using additional methods like clustering or neural networks to increase the detection rate and reduce the false alarm rate. A model with excellent accuracy but a very high false alarm rate of 0.05 % was proposed by Nalayini and Katiravan [27]. A methodology for the detection of DDoS assaults was put out by Chavan et al. [28]. They used multiple machine learning classifiers to evaluate their model. They discovered that the model with the greatest accuracy for the logistic regression classifier was 90.4 %. This accuracy % is unacceptable in the context of the current day. Because of the heavy traffic, the model is unable to huge pick up malicious traffics such as DDoS. The FPR will thus be quite high. The same year, Elgendy et al. [29] released DTEXNet, a cutting-edge method with a 95 % accuracy rate. This is more difficult since it combines two neural network models. In addition, the accuracy should be improved in relation to the dataset's size.

In 2023, Samaan and Jeiad [30] proposed a method that uses

gradient-boosted trees techniques and achieves 93 % accuracy. The precision has to be raised. The TPR in this model is really low. Sabir [31] applied BayesNet, KNN, J48 classifiers to detect DDoS and found the J48 classifer based model produce the best result with the accuracy of 98.31 %. Some researchers have also suggested certain deep learning-based methods for different environments [32–35]. These strategies are less useful for deployment in situations with limited resources. Additionally, they have had difficulty identifying new and developing DDoS attacks when certain attack types are not covered by the training data.

Most researchers proposed a model with a single machine-learning classifier. Single machine learning classifiers often struggle to detect newly designed DDoS attacks due to several key challenges. These include the lack of training data for novel attacks, the complexity and diversity of attack features, concept drift in network behavior, the adaptability of attackers, imbalanced data, and the potential overfitting of models. Additionally, accuracy and TPR should be improved in comparison to the current methods, while FPR should be minimized. Therefore, the creation of a DDoS attack detection model that is reliable against each attack is required [36].

The proposed methodology, employing a random forest ensemble-based classifier named HFS-ERF, shows promise in providing a comprehensive and efficient solution for safeguarding networks against various DDoS attacks. To enhance the model's effectiveness, a novel feature selection approach has been introduced, which combines Correlation Analysis, Mutual Information, and Principal Component Analysis. Notably, this feature selection technique yields improved results within a single classifier-based framework. Given the ever-evolving nature of technology and the changing landscape of DDoS attacks, the model has been augmented with enhanced ensemble-based classifiers. This integration ensures the model's ability to adeptly detect newly designed and emerging types of DDoS attacks. Notably, this results in an increased TPR and a decreased FPR for the model. The model's efficacy has been rigorously evaluated using various publicly available datasets, and it consistently demonstrates excellent performance in detecting all types of DDoS attacks. Particularly, the utilization of the random forest ensemble classifier proves highly effective in achieving this objective compared to the other ensemble approaches.

3. Proposed model

The proposed approach for identifying attacks using DDoS is thoroughly described in this section. The proposed machine learning model's pipeline, including the appropriate feature selection and ensemble-based Random Forest classifier, is shown in Fig. 1.

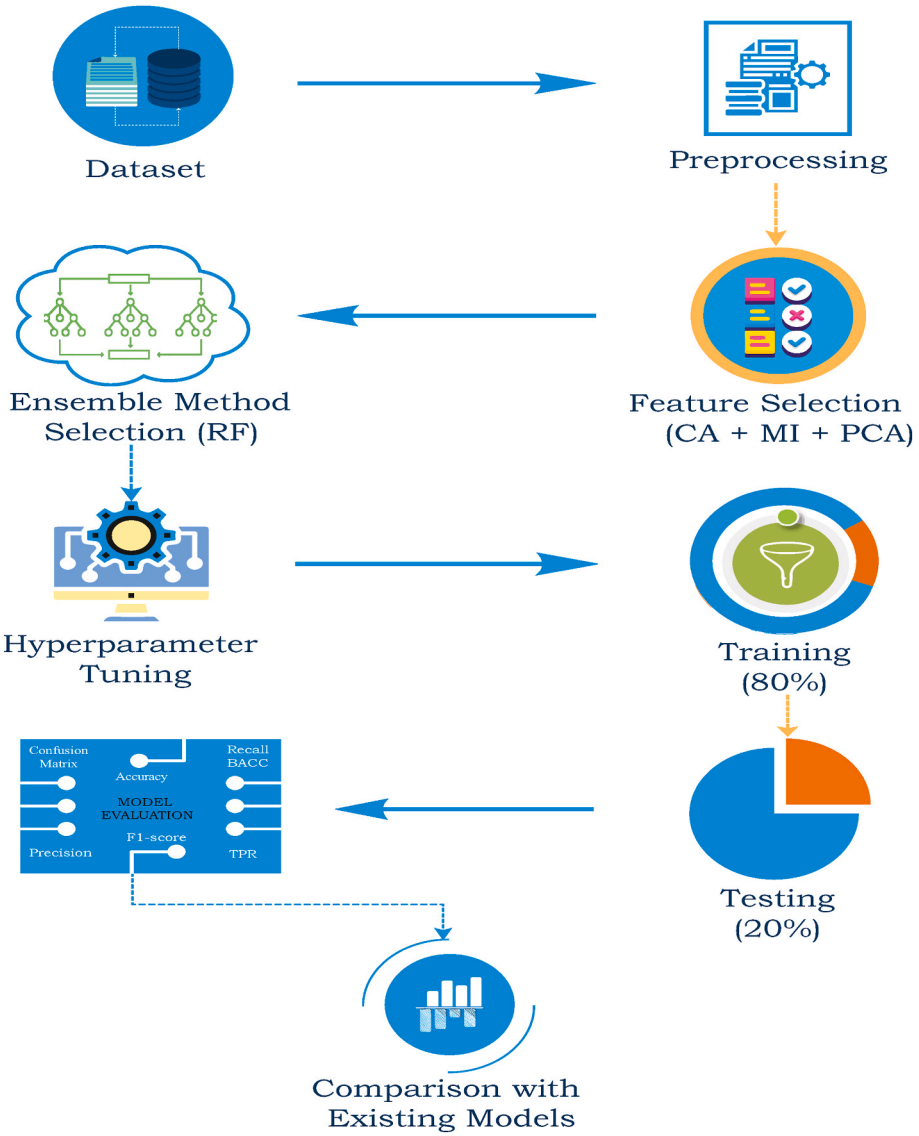

Fig. 1. Proposed model development pipeline.

3.1. Dataset

The model is trained on various datasets to learn patterns and relationships in the data, and it is then tested on separate sets of data to evaluate its performance. Several publicly available datasets are utilized to evaluate the model's performance. Below is a very short observation on each of the following.

3.1.1. CIC-DDoS2019

The CIC-DDoS2019 dataset, which closely resembles genuine real-world data, contains the most recent and benign attacks using DDoS. Additionally, it contains the outcomes of a CICFlowMeter-V3 network traffic analysis with flows labeled according to the time stamp, destination, and source IP addresses, source and destination ports, protocols, and attack. This dataset contains two forms of DDoS attacks: reflection-based and exploitation-based. TCP and UDP assaults were carried out differently by each type. Table 2 presents a comprehensive list of DDoS attack types of this dataset, along with the corresponding feature with used entries utilized in this research [37].

3.1.2. DDoS-SDN

Since the model is effective for all types of DDoS attacks and environments. So, it is trained and tested by an SDN-related dataset also. There are 23 features and 104,345 rows available in the DDoS-SDN dataset and created in 2020. Network simulation is done for malicious traffic like TCP flooding attack, UDP flooding attack, and ICMP assault as well as for normal traffic like UDP, TCP, and ICMP. The last column's class name, which indicates whether the traffic is malicious or not, is displayed. Label 1 indicates malicious traffic, while label 0 indicates benign traffic [38].

3.1.3. CSE–CIC–IDS2018

The CSE–CIC–IDS2018 [39] dataset is a publicly available dataset that contains network traffic data collected from a simulated environment for the purpose of evaluating IDSs. In our experimental setup, we exclusively utilized the '02-21-2018.csv' file, primarily due to its inclusion of both DDoS attacks and normal network traffic data. This particular dataset offers a comprehensive and balanced representation of network activities. At the outset of our investigation, we commenced with this dataset comprising a total of 1,048,575 entries, each encompassing a rich set of 80 distinct features. These features encompassed a wide range of network-related attributes, allowing us to thoroughly explore and analyze the dataset's intricacies. Its inclusion of both normal traffic and DDoS attack data provided a well-rounded and challenging environment in which to evaluate the model's performance and effectiveness.

3.1.4. APA-DDoS

As the number of connected devices grew, the main problem was identifying attacks since an intrusion detection system had been developed. Different DDoS attack types are included in the APA-DDoS attack dataset. Most of the attacks involving DDoS in the dataset are ACK and PUSH-ACK. The APA-DDoS attack dataset is being used to evaluate a model for the detection of DDoS attacks [40].

Table 2
Different attacks in CIC-DDoS2019 dataset.

Sr. No.	Attacks	Used Entries	No. of Features
1.	MSSQL	1048575	87
2.	SSDP	2611374	88
3.	LDAP	2181542	88
4.	NetBIOS	1048575	87
5.	NTP	1217007	88
6.	SNMP	1217007	88
7.	UDP Flood	3136802	88
8.	Syn Flood	1582681	88
9.	TFTP	1048575	87
10.	UDPLag	725165	88

3.1.5. DDoS-botnet

For evaluating and predicting harmful packets from DDoS botnet attacks, there is a dataset called DDoS-botnet [41]. 1927101 entries and 47 features are utilized in this research to evaluate the model. DDoS attacks sometimes involve the usage of botnets because they provide the attacker access to a big resource pool that may be utilized to produce a lot of traffic. Computers, cellphones, and other internet-connected devices that are under the attacker's control and infected with malware can all be found in a botnet. For this reason, this dataset is chosen to test the model.

3.2. Data preprocessing

Data preprocessing is essential for a machine-learning model since it aids in cleaning and converting raw data into a form that the model can readily interpret. Preprocessing procedures including managing missing values, eliminating outliers, scaling, and normalizing may considerably increase the accuracy and efficiency of the model. The quality of the input data directly influences the model's performance. Additionally, preprocessing helps in lowering the chance of overfitting and improving the data's interpretability. As a result, data preprocessing is a crucial stage in the machine-learning pipeline that enables models to produce predictions that are accurate and trustworthy [42].

In this model, the preprocessing steps involve eliminating duplicates, substituting NaNs for infinite and large values, deleting rows that contain NaNs, separating numerical and categorical columns, normalizing numerical columns, encoding categorical columns, and changing the target variable into a discrete variable. Using the "duplicated" function from pandas, we first examine the dataframe for duplicate entries. The method returns true for those rows if any duplications are discovered. The duplicate rows are then removed from the dataframe using the "drop_duplicates" function of the Pandas library. The "dropna" function of pandas was then used to remove any rows that had NaNs. The remaining dataframe is divided into columns for numbers and categories. The types "float64" and "int64" denote numerical columns, whereas the type "object" denotes categorical columns. After that, we used the "StandardScaler" function from the sklearn package to normalize the numerical columns. This guarantees that each feature has a unit variance and zero mean. Utilizing the "LabelEncoder" function from the sklearn package, the category columns are encoded. This changes categorical variables into numerical variables by assigning a unique integer value to each distinct value [43].

3.3. Relevant features selection

By eliminating unnecessary and redundant features, lowering the model's complexity, and enhancing its interpretability, relevant feature selection is crucial to enhancing the accuracy and effectiveness of machine learning models [44]. Using correlation analysis (CA), mutual information (MI), and principal component analysis (PCA), the relevant features in this model have been chosen.

Because it identifies strongly correlated features that could lead to model overfitting or redundancy, correlation analysis is crucial for feature selection in ensemble techniques. Using closely related characteristics can lead to several models generating similar predictions, decreasing the diversity of the ensemble [45]. Ensemble approaches integrate multiple models to increase predictive accuracy. Finding and removing highly correlated features via correlation analysis will improve the performance and stability of the ensemble technique. Correlation analysis is performed to choose appropriate features for prediction throughout our model's implementation. We first compute the correlation matrix "corr" for the features in X, and only select the features with an absolute correlation coefficient over 0.5. The correlation

coefficients meaning is the covariance between two variables, X and Y, divided by the product of their standard deviations. This can be expressed mathematically in the form of Equation (1).

$$\text{corr}(X, Y) = \text{cov}(X, Y) / (\text{std}(X) * \text{std}(Y)) \quad (1)$$

The absolute value of the correlation matrix "corr" is taken to produce the "corr_abs" variable in the implementation. This is true because the correlation coefficient's absolute value, regardless of whether it is positive or negative, indicates the strength of the linear connection between the two variables. The index of the columns (i.e., features) with an absolute correlation coefficient larger than 0.5 is then used to produce the "relevant_features_corr" variable in Equation (2). On the portion of the correlation matrix that meets the threshold, this is accomplished by using the ".index.tolist()" method:

$$\text{relevant\_features\_corr} = \text{corr\_abs}[\text{corr\_abs} > 0.5].\text{index.tolist}() \quad (2)$$

The names of the features that are highly associated with one another and may thus be redundant are listed in the relevant_features_corr list that is produced. The performance of our model can be enhanced by removing or combining these features in order to decrease the dataset's dimensionality.

Mutual information is a metric used to assess the dependency between two random variables. In the context of feature selection, it evaluates how much information a feature contributes to the target variable [46]. The top k features that are most informative about the target variable y are chosen in our implementation. Equation (3) describes the mutual information between a feature X_i and a desired variable y.

$$I(X_i, y) = H(X_i) - H(X_i|y) \quad (3)$$

Where $H(X_i)$ is the entropy of feature X_i, and $H(X_i|y)$ is the conditional entropy of feature X_i given the target variable y.

The level of uncertainty or randomness in a random variable is measured by entropy. It is calculated using the following Equation (4):

$$H(X) = -\text{sum}(p(x) * \log2(p(x))) \quad (4)$$

Where p(x) is the probability of observing the value x in the random variable X.

Conditional entropy determines the level of uncertainty in a random variable X based on the value of another random variable Y. It is computed using Equation (5) as follows:

$$H(X|Y) = -\text{sum}(p(x,y) * \log2(p(x|y))) \quad (5)$$

Where p (x,y) is the joint probability of observing the values x and y in the random variables X and Y, and p (x|y) is the conditional probability of observing the value x in X given the value y in Y.

The mutual information between each feature and the target variable y is determined using the "mutual_info_classif" function from the "sklearn.feature_selection" package. The top k features with the best mutual information scores are then chosen using the "SelectKBest" function from the same module. The features are expected from mutual information to have the strongest statistical relationship with the target variable, making them crucial for modeling and analysis. Selected by using equations (6) and (7).

$$\text{mutual\_info} = \text{SelectKBest}(\text{mutual\_info\_classif}, k = 20).\text{fit}(X, y) \quad (6)$$

$$\text{relevant\_features\_mutual} = X.\text{columns}[\text{mutual\_info.get\_support}()].\text{tolist}() \quad (7)$$

Principal Component Analysis (PCA) [47] is a critical method for feature selection as it allows us to reduce the dimensionality of the input data by identifying the most significant features that capture the most variance in the data. The fundamental formula for PCA is as follows, When given a data matrix X with n samples and m features, PCA attempts to locate a set of k orthogonal vectors u_1, u_2,…, u_k in the m-dimensional space such that the data's variance is maximized when projected onto the subspace spanned by these vectors, as shown in the equation below.

$$Y = XU_k \quad (8)$$

Where the top k eigenvectors of the X covariance matrix are represented by the matrix U_k. The columns of the Y matrix, which reflect the additional features acquired by projecting the original data onto the subspace covered by the eigenvectors, make up the main components.

In our implementation, the PCA analysis on the input data X is carried out using the "PCA" class from the "sklearn.decomposition" package [43]. The number of components is set to 20. The "pca.components_" attribute of the "PCA" object yields the matrix U_k, and the indices of the features with the greatest absolute values in each component are obtained using the "argmax()" function. The most crucial features are then obtained by mapping these indices back to the original feature names using the input data's "X.columns" property shown in equations (9) and (10).

$$\text{pca} = \text{PCA}(\text{n\_components} = 20) \quad (9)$$

$$\text{relevant\_features\_pca} = X.\text{columns}[\text{pca.components\_.argmax}(\text{axis} = 1)].\text{tolist}() \quad (10)$$

Finally, we generate a single list from the relevant features chosen from each of the three approaches - correlation analysis (CA), mutual information (MI), and principal component analysis (PCA). The final combination of the above three techniques is performed by equation (11).

$$\text{relevant\_features} = \text{list}(\text{set}().\text{union}(\text{relevant\_features\_corr}, \text{relevant\_features\_mutual}, \text{relevant\_features\_pca})) \quad (11)$$

We developed a more trustworthy and accurate collection of features that accurately capture the most crucial data by merging relevant features from multiple feature selection methods. The model's performance is improved as a result and new information about the underlying relationships between the features and the target variable is discovered. One of the key components for this model's improved DDoS detection performance is this feature selection technique.

3.4. Ensemble-based random forest classifier selection

With better detection rates and attack resistance, ensemble-based machine learning classifiers are an option for intrusion detection systems [48], particularly in the detection of DDoS attacks. These classifier integrate many separate models to increase the prediction's overall accuracy and resilience. Since there are so many different classifiers in the ensemble, DDoS attack detection may identify a variety of attack types and patterns. The proposed model employs both a variety of single classifiers and a variety of machine learning ensemble classifiers. From the various ensemble based approaches the random forest offers the best results. The ensemble-based Random Forest classifier is employed in DDoS attack detection by combining multiple decision trees to enhance accuracy and robustness. It leverages the diversity of individual trees to collectively identify attack patterns, making it particularly effective at distinguishing between normal network traffic and various DDoS attack types, ultimately bolstering the security of network environments [11]. The Random Forest ensemble classifier for the detection of DDoS attacks using the relevant features is described in short detail below.

Combining many decision trees with the ensemble-based machine learning approach known as random forest results in a final classification conclusion. The random forest builds each decision tree independently using a randomly selected subset of the training data and attributes in order to reduce overfitting and improve generalization performance. The result of the random forest method is determined by the majority vote of the different decision trees. Each decision tree in the forest casts a vote for the estimated class of the input data point, and the

predicted class with the most votes is the result [49]. Algorithm 1 describes how the Random Forest classifier works on DDoS attack-related datasets to determine normal or attack instances.

Algorithm 1. Working process of the Ensemble-based Random Forest classifier for DDoS attack detection

i. *Initialize a set of decision trees T*
ii. *For t = 1 to T:*
  a. *Randomly sample m features from the set of input features.*
  b. *Create a new decision tree D_t by recursively splitting the data into smaller subsets based on the selected features.*
    i. *At each node of the tree, choose the feature that maximizes the information gain.*
    ii. *Stop splitting when the maximum depth of the tree is reached or when all the instances at a node belong to the same class.*
  c. *Add the decision tree D_t to the ensemble.*
iii. *For each instance x_i in the training set:*
  a. *Create a feature vector z_i by extracting the relevant features, which are chosen through techniques like correlation analysis, mutual information, and PCA.*
  b. *For each decision tree D_t in the ensemble, calculate the class prediction y_i, t using the decision path of the instance in the tree.*
  c. *Aggregate the predictions of all the decision trees to obtain the final class prediction y_i:*
    i. *If the majority of the trees predict $y_i = 1$, classify x_i as a DDoS instance.*
    ii. *Otherwise, classify x_i as a normal instance.*
iv. *Output the ensemble of decision trees.*

Table 3 lists the hyperparameters utilized by the Random Forest classifier to implement the proposed model.

4. Experimental results and analysis

In the following section, we evaluate how well the proposed DDoS attack detection model performs. Before moving into that, the environmental setup, as well as evaluation metrics, are sort of described below.

The Scikit-learn package and the Python programming language are used throughout the whole experiment for this research. Google Colaboratory, commonly referred to as "Colab", is a tool developed by Google Research that was used for the experiment. The Scikit-Learn package's StandardScaler, LabelEncoder, and other preprocessing modules are implemented. To choose relevant features, the mutual_info_classif, PCA, and SelectKBest methods are employed. The ensemble module's classifiers are BaggingClassifier, AdaBoostClassifier, RandomForestClassifier, GradientBoostingClassifier, XGBClassifier, and StackingClassifier tested. Classical classifiers include GaussianNB from naive_bayes, MLPClassifier from neural_network, KNeighborsClassifier from neighbors, LogisticRegression from linear_model, and SVC from svm are applied. The model is further assessed using the ROC curve, ROC auc score, Cohen kappa score, confusion matrix, accuracy score, precision, FPR, recall score, BCC, and f1-score of the Scikit-learn (python library) infrastructure [43].

Table 3
Hyperparameters in the Random Forest classifier.

Parameter	Values	Parameter	Values
n_estimators	10	min_impurity_decrease	0.0
bootstrap	True	warm_start	False
criterion	'gini'	max_depth	None
n_jobs	None	oob_score	False
max_features	'sqrt'	min_samples_split	2
min_samples_leaf	1	random_state	42
min_weight_fraction_leaf	0.0	verbose	0
max_leaf_nodes	None	class_weight	None

The "train_test_split" function from the scikit-learn (sklearn) package divides the whole dataset for the model into training and testing data. While 80 % of the data is used for training, the remaining 20 % is used for testing for each classifier.

4.1. Evaluation metrics

Evaluation metrics are used to assess the performance of a model. In the case of the proposed DDoS attack detection model, the evaluation metrics used include accuracy, recall, precision, f1-score, false positive rate (FPR), true positive rate (TPR), Balanced Accuracy (BACC), The AUC (Area Under the Curve), error rate, training accuracy, test accuracy, cohen's kappa etc [50,51]. These metrics provide a comprehensive view of the model's performance and can help assess its usefulness in detecting DDoS attacks with high accuracy and low false positive rates. A confusion matrix displays the number of correct and incorrect predictions made by the model, compared to the actual outcomes (or true labels) in the test data. The table is usually a square matrix, where the rows represent the actual class labels, and the columns represent the predicted class labels. The four outcomes that are possible in a binary classification problem are true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

- *True Positive (TP): The positive class is appropriately predicted by the model.*
- *False Positive (FP): The positive class is mistakenly predicted by the model.*
- *True Negative (TN): The negative class is appropriately predicted by the model.*
- *False Negative (FN): The model predicts the negative class wrongly.*

Using the values in the confusion matrix, we can compute several evaluation metrics listed in Table 4. Other metrics with proper equations are also listed.

It is possible to evaluate the model's robustness and reliability in identifying DDoS attacks in various situations and environmental factors by utilizing evaluation metrics.

4.2. Analysis of the findings

Table 5 shows the evaluation findings for the proposed DDoS attack detection model on several types of DDoS attacks using the CIC-DDoS2019 dataset. The attacks are classified as either reflection or exploitation attacks and are tested for accuracy, recall, precision, f1-

Table 4
Formulas and clarifications of measurement metrics for performance.

Metrics	Formulas	Clarifications
Accuracy	$(TP + TN)/(TP + FP + FN + TN)$	Evaluates the model's performance in terms of effectively detecting instances.
Recall	$TP/(FN + TP)$	Evaluates the model's accuracy in identifying the fraction of positive cases that are actually positive.
Precision	TP/(FP + TP)	Measures the exactness of a model's positive predictions
F1-score	2 * (recall * precision)/(recall + precision)	Determines the balance between the model's capacity for accurate positive instance identification (recall) and its capacity for reducing false positives (precision).
AUC Score	$\int TPR(FPR)\, dFPR$	AUC Curve provide insights into the model's ability to distinguish between positive and negative instances, and identify the optimal classification threshold for the model.
FPR	$FP/(TN + FP)$	Determines the percent of negative occurrences that the model misclassifies as positive
BACC	$(S_t + S_p)/2$	Balanced Accuracy measures the average of sensitivity and specificity, providing a single value that represents the overall model performance [52].
Training accuracy	Training Accuracy = (Number of Correctly Classified Instances in Training Data)/(Total Number of Instances in Training Data)	Training accuracy is a metric used to measure how well a machine learning model performs on the same data it was trained on. It indicates the proportion of correctly classified instances in the training dataset
Test accuracy	Test Accuracy = (Number of Correctly Classified Instances in Test Data)/(Total Number of Instances in Test Data)	Test accuracy is used to evaluate how well a machine learning model performs on data that it has never seen during training. It measures the proportion of correctly classified instances in a test dataset.
Error Rate	Error Rate = (Number of Misclassified Instances)/(Total Number of Instances in the Dataset)	Error rate, is a metric that quantifies the proportion of misclassified instances in a dataset. It represents the model's predictive errors as a percentage of the total instances.
Cohen's Kappa	$K = \frac{Po - Pe}{1 - Pe}$ K (Kappa) represents the Cohen's Kappa statistic.	Cohen's Kappa is a statistic that measures the level of agreement between two raters between a model's predictions and the true labels for a classification problem [52]. It accounts for the possibility of agreement occurring by chance, which makes it particularly useful for evaluating classification performance when dealing with imbalanced datasets or when simple accuracy might be misleading. Po(Observed Agreement) is the proportion of observed agreement between the model's predictions and the true labels. It is typically calculated as the ratio of the number of agreements to the total number of instances. Pe (Expected Agreement by Chance) is the proportion of agreement expected to occur by chance. It takes into account the class distribution and is calculated as the product of the marginal probabilities of each class's prevalence.

score, and AUC score. The model achieved a perfect score of 1.0 on all metrics for the reflection attacks: MSSQL, SSDP, LDAP, NetBIOS, NTP, SNMP, and TFTP. For the exploitation attacks, UDP Flood and Syn Flood, the model also achieved perfect scores of 1.0 on all metrics. For the UDPLag attack, the model achieved an accuracy of 0.9999, which is still a very high score. The recall and f1-score were both 1.0, meaning the model was able to correctly identify all instances of the attack. The high degree of performance from the table shows that the model has practical applications in real-world cybersecurity systems, where it could help to prevent or mitigate the effects of DDoS attacks.

The identical procedures, as elaborated in the section titled "Proposed Model", are employed for documenting the outcomes produced by the model when it is deployed on diverse datasets. In the pursuit of appraising the model's efficacy, only the datasets are modified. The model's performance in detecting DDoS attacks is subsequently assessed by contrasting it with alternative detection models across various datasets.

Fig. 2 displays a comparison results of the performance of various classical machine learning (ML) classifiers, including logistic regression (LR), Gaussian naive Bayes (GNB), k-nearest neighbor (KNN), artificial neural networks (ANN), and support vector machines (SVM), with the proposed ERF classifier based model for the detection of DDoS attacks. The figure is generated for the type of DrDoS_NTP under the CIC-DDoS2019 dataset. The performances of the model with the classical machine-learning classifier are varies for different datasets but the results with the ERF classifier are always high. The ERF classifier achieved a perfect score of almost 100 % for all evaluation metrics, indicating that it performed exceptionally well in detecting DDoS attacks than classical machine-learning classifiers. Another significant observation that emerges from the figure is the consistently high accuracy achieved by various classifiers. This notable performance can be largely attributed to the relevant feature selection process we employed. It becomes evident that our novel approach to identifying and utilizing relevant features plays a pivotal role in enhancing overall model accuracy.

Table 6 presents a comparison of the performance of four ensemble-based classifiers, including Random Forest, Bagging, Adaboost, and Simple Stacking, for the detection of DDoS attacks. The results are included for the DrDoS_NTP dataset under the CIC-DDoS2019 dataset. The evaluation metrics used in the comparison include Recall, F1-score, False Positive Rate (FPR), and Testing Time (in seconds). The Random Forest classifier achieved a perfect score of 1.0 for both Recall and F1-score, indicating that it performed exceptionally well in detecting DDoS attacks. It also had the lowest FPR score of 0.0, indicating that it had a very low false-positive rate. Additionally, it had a relatively low

Table 5
Evaluation findings for several DDoS attack datasets under the CIC-DDoS2019.

Attacks	Type of Attacks	Accuracy	Recall	Precision	F1-score	AUC
MSSQL	Reflection (TCP)	1.0000	1.0000	1.0000	1.0000	1.0000
SSDP	Reflection (TCP)	1.0000	1.0000	1.0000	1.0000	1.0000
LDAP	Reflection (TCP/UDP)	1.0000	1.0000	1.0000	1.0000	1.0000
NetBIOS	Reflection (TCP/UDP)	1.0000	1.0000	1.0000	1.0000	1.0000
NTP	Reflection (UDP)	1.0000	1.0000	1.0000	1.0000	1.0000
SNMP	Reflection (TCP/UDP)	1.0000	1.0000	1.0000	1.0000	1.0000
UDP Flood	Exploitation (UDP)	1.0000	1.0000	1.0000	1.0000	1.0000
Syn Flood	Exploitation (TCP)	1.0000	1.0000	1.0000	1.0000	1.0000
TFTP	Reflection (UDP)	1.0000	1.0000	1.0000	1.0000	1.0000
UDPLag	Exploitation (UDP)	0.9999	1.0000	0.9867	0.9933	1.0000

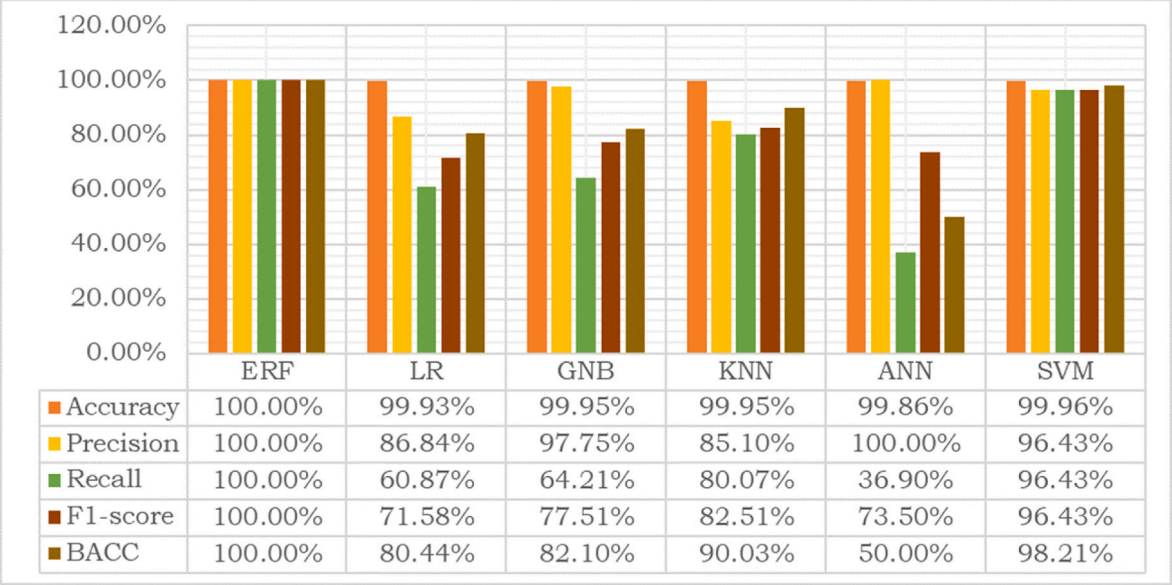

Fig. 2. Model performance with classical ML classifiers and ERF.

Table 6
Model performance using various ensemble-based ML classifiers.

Model with Ensemble Classifier	Recall	F1-score	FPR	Testing Time (seconds)
Random Forest	1.00000	1.00000	0.00000	0.16216
Bagging	0.99631	0.99815	0.00369	0.86145
Adaboost	0.99631	0.99815	0.00369	0.44962
Simple Stacking	0.99631	0.99815	0.00369	12.58364

testing time of 0.16216 seconds, making it computationally efficient. Based on the results presented in the table, the Random Forest classifier outperforms the other ensemble-based classifiers in terms of detecting DDoS attacks with high accuracy, low false-positive rate, and computational efficiency. And also higher for other datasets. Therefore, the ERF classifier in this model is the best choice for identifying DDoS attacks.

The ROC curve, shown in Fig. 3, serves as a visual representation of the crucial trade-off between a classifier system's TPR and FPR as the discrimination threshold undergoes variation during the process of identifying DDoS attacks [53]. In the figure, the TPR is plotted on the vertical axis (y-axis), while the FPR is plotted on the horizontal axis (x-axis), with the optimal classifier ideally achieving a TPR of 1 and an FPR of 0. The ROC curve in the figure showcases the performance of different classifiers on the CIC-DDoS2019 (DrDoS_NTP) dataset.

The Area Under the Curve (AUC) score associated with each ROC curve quantifies the classifier's effectiveness. A higher AUC score, approaching 1, indicates superior classifier performance [54]. Notably, the model with the Random Forest classifier exhibits the highest AUC score of 1.00000 among the evaluated classifiers. It is followed by Bagging, Gradient Boosting, Stacking, ANN, Gradient Boosting, KNN, Adaboost, SVM, LR, and GNB classifiers AUC scores. Consequently, the Random Forest classifier emerges as the top-performing choice for DDoS attack detection. Importantly, this observation holds true not only for

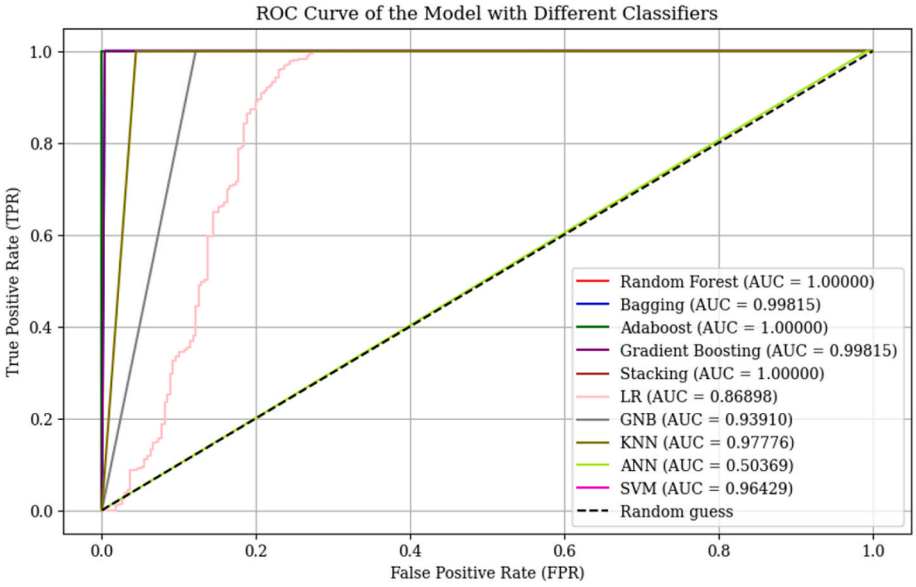

Fig. 3. ROC curve of the model for various classifier for CIC-DDoS2019 (DrDoS_NTP) dataset.

Table 7
Evaluation findings for other dataset to detect DDoS attacks.

Dataset	Accuracy	Recall	Precision	F1-score	FPR	AUC	BACC
CSE–CIC–IDS2018 [39]	1.0000	1.0000	1.0000	1.0000	0.0000	1.0000	1.0000
DDoS-SDN [38]	1.0000	1.0000	1.0000	1.0000	0.0000	1.0000	1.0000
APA-DDoS [40]	1.0000	1.0000	1.0000	1.0000	0.0000	1.0000	1.0000
Botnet-DDoS [41]	1.0000	1.0000	1.0000	1.0000	0.0000	1.0000	1.0000

the CIC-DDoS2019 (DrDoS_NTP) dataset but also for various other datasets, consistently yielding nearly identical AUC scores across experiments.

Table 7 presents the evaluation findings for the model used to detect DDoS attacks in various public datasets like CIC-IDS2018, DDoS-SDN, APA-DDoS, and Botnet-DDoS. These experimental results have been taken into account to evaluate the model's performance across a range of scenarios and datasets. The model achieves a perfect accuracy of 1.0 for all four datasets, indicating that it correctly classifies all instances in the test set. Similarly, the model's recall, precision, and f1-score are all perfect, indicating that the model correctly identifies all instances of DDoS attacks without incorrectly classifying any benign traffic as an attack. The FPR is also 0.0 for all four datasets, indicating that the model does not incorrectly classify any benign traffic as an attack. The AUC is 1.0 for all four datasets, which indicates that the model has an excellent performance in distinguishing between DDoS attacks and benign traffic. The BACC is also 1.0 for all datasets, which indicates that the model is equally good at correctly classifying both DDoS attacks and benign traffic. From the higher results for different datasets, it is clear that the model is highly robust and can accurately detect DDoS attacks even in previously unseen datasets or unknown attack scenarios.

Fig. 4 offers an in-depth view of the confusion matrix generated from our analysis of the tested datasets. These matrices are exclusively associated with the test data, constituting 20 % of the entire dataset. The results unveiled by these matrices are undeniably impressive, as the model consistently demonstrates near-perfect classification across all four key metrics: TP, TN, FP, and FN. This remarkable accuracy serves as strong evidence of the model's exceptional performance in

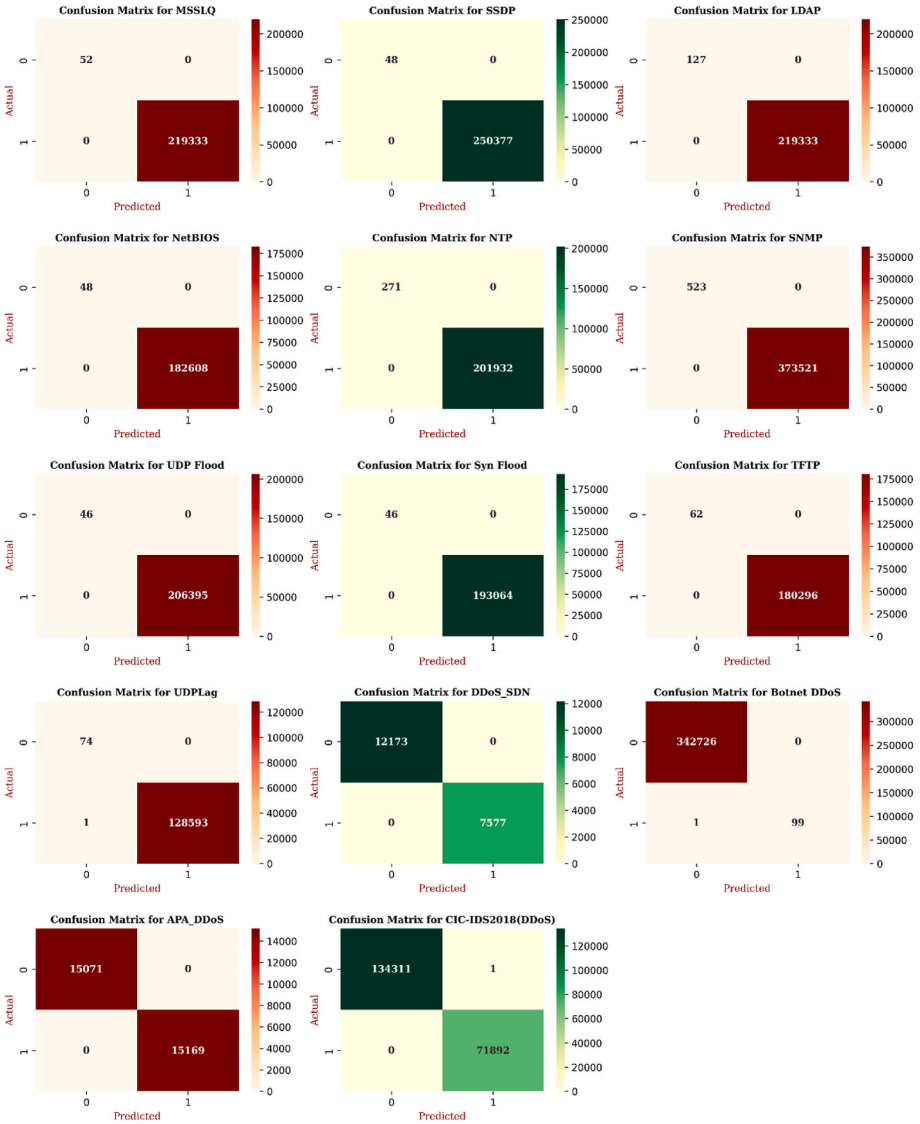

Fig. 4. Confusion matrix for all tested datasets.

Table 8
Results of other metrics for the model for various datasets.

Datasets (Type of DDoS Attacks)	Training Accuracy	Test Accuracy	Error Rate	Cohen's Kappa	
				Observed Accuracy (Po)	Expected Accuracy (Pe)
APA-DDoS	1.00000	1.00000	0.00000	1.00000	0.37583
CSE–CIC–IDS2018	1.00000	1.00000	0.00000	1.00000	0.54348
CIC-DDoS2019 (MSSQL)	1.00000	1.00000	0.00000	1.00000	0.99882
CIC-DDoS2019 (SSDP)	1.00000	1.00000	0.00000	1.00000	0.99962
CIC-DDoS2019 (LDAP)	1.00000	1.00000	0.00000	1.00000	0.99884
CIC-DDoS2019 (NetBIOS)	1.00000	1.00000	0.00000	1.00000	0.99947
CIC-DDoS2019 (NTP)	1.00000	1.00000	0.00000	1.00000	0.99732
CIC-DDoS2019 (SNMP)	1.00000	1.00000	0.00000	1.00000	0.99982
CIC-DDoS2019 (UDP Flood)	1.00000	1.00000	0.00000	1.00000	0.99955
CIC-DDoS2019 (Syn Flood)	1.00000	1.00000	0.00000	1.00000	0.99952
CIC-DDoS2019 (TFTP)	1.00000	1.00000	0.00000	1.00000	0.99931
CIC-DDoS2019 (UDPLag)	1.00000	0.99999	0.00001	0.99999	0.99884
DDoS-SDN	0.99999	1.00000	0.00000	1.00000	0.52708
Botnet DDoS	1.00000	1.00000	0.00000	1.00000	0.99942

Based on the results of the table for various datasets, the HFS-ERF model performs very well in detecting DDoS attacks, as shown by the high accuracy, low error rate, and high values of Po and Pe for all datasets.

distinguishing between different classes. The precision with which it identifies TP, TN, FP, and FN instances underscores its robustness. Given these nearly flawless outcomes, it is evident that the model's predictions, when coupled with the confusion matrices, consistently yield highly favorable results across various evaluation metrics.

The comprehensive evaluation of these metrics highlights the model's outstanding ability to detect DDoS attacks when applied to a diverse range of publicly available datasets. Its consistent, high-level performance across these datasets underscores its reliability and effectiveness in identifying and mitigating such attacks.

The findings of more evaluation metrics for the HFS-ERF model on various datasets for the detection of attacks with DDoS are presented in Table 8. These additional results have been included in the table to evaluate the model's effectiveness across several datasets and to see how it performs under various DDoS attack scenarios and environmental factors. The model's training accuracy is perfect (1.0) for all datasets, indicating that the model has learned the training data well. The test accuracy is also very high, indicating that the model is performing well on the unseen test data [55]. The error rate is almost zero for most of the datasets, which is a good indication of the model's performance.

The observed accuracy (Po) and expected accuracy (Pe) are calculated using Cohen's Kappa, a statistical measure of inter-rater agreement. Po represents the observed agreement between the model's predictions and the actual values, while Pe represents the expected agreement by chance. In general, a higher value of Po indicates better model performance. The model achieved very high observed accuracy (Po) values ranging from 0.99999 to 1.0 for all datasets, indicating excellent performance in detecting DDoS attacks. Moreover, the expected accuracy (Pe) values are relatively high, indicating that the model's performance is not due to chance alone.

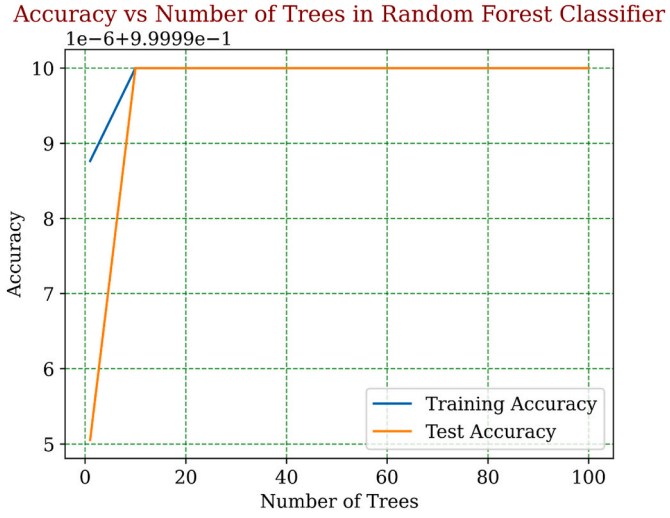

Fig. 5. Effectiveness of the model for different no. of trees.

Fig. 5 depicts the relationship between the number of trees in a Random Forest Classifier and the corresponding training and test accuracy scores. The figure presented in this analysis is derived from the test data of the CIC-DDoS2019 (NTP) dataset. It's worth noting that similar results have been observed when applying the Random Forest Classifier in this model to other publicly available datasets. The X-axis represents the number of trees (decision trees) in the Random Forest Classifier. The Y-axis represents the accuracy of the model. The training accuracy line shows how accurately the model fits the training data as the number of trees in the forest increases. In this case, the training accuracy starts at an extremely high value (close to 1) and remains consistently high as the number of trees increases. The test accuracy demonstrates how well the model generalizes to unseen data as the number of trees changes. The ensemble-based Random Forest classifier's optimal estimator is 10 since this number of trees results in the model producing the best training and test accuracy. Like the training accuracy, the test accuracy is also very high, and it remains consistently high regardless of the number of trees in the forest. The high and consistent accuracy scores indicate that the model is capable of learning and generalizing effectively to make accurate predictions. The lack of a significant difference between training and test accuracy (overfitting) suggests that the model generalizes well and is not likely to suffer from overfitting issues [56], even with a large number of trees. These consistent patterns of accuracy across different datasets are indicative of the model's robust performance. This consistency underscores the model's ability to generalize effectively, making it a strong model for real-world security, especially in the context of detecting various DDoS attacks.

4.3. Comparing the performance with the existing models

Table 9 presents a comparison of different machine learning models for the detection of DDoS attacks, based on four evaluation metrics. Each model's performance is reported for each metric, and the proposed HFS-ERF model outperforms all other models in terms of all four metrics. For comparing the performance of the, very recently published models in 2023 are listed. In most of the models, the researchers used the CIC-DDoS2019 dataset. Our model provides better results for any publicly available datasets. Based on the results in the table, we can conclude that the proposed HFS-ERF model is the best-performing model for detecting DDoS attacks. In addition, it is effective for any type of DDoS attack

Table 9
Comparison of the proposed model with the very recent published models.

Model with Reference	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
GHLBO-based DSA [35]	91.40	90.90	*	*
GBT [30]	93.62	93.63	96.28	94.41
CNN-DSFO [57]	98.85	93.42	*	91.65
LSTM [33]	98.00	97.00	98.00	97.00
J48 [31]	98.31	98.30	98.30	98.30
MLP [34]	98.99	*	*	*
DCNN [32]	99.77	99.77	99.77	99.77
Proposed HFS-ERF	100.0	100.0	100.0	100.0

(*) Means not mentioned.

detection for any environment.

It is clear that the proposed DDoS detection model is effective based on the outcomes of the various evaluation metrics for the datasets of diverse environments stated above. It may also be used to identify attacks using DDoS from botnets and IoT devices. For currently available models to detect DDoS attacks, the Proposed HFS-ERF model also offers improved results.

5. Conclusion and future direction

The advanced approach presented for DDoS attack detection, employing a hybrid feature selection method and an ensemble-based Random Forest machine learning classifier, has showcased exceptional performance compared to existing techniques. The fusion of various feature selection methods and ensemble-based classifiers has yielded remarkable results, with near-perfect accuracy and outstanding performance across a range of evaluation metrics, making it a highly promising solution for real-world DDoS attack detection. The model's consistent excellence in handling diverse DDoS datasets, including Botnet DDoS, APA-DDoS, DDoS-SDN, and others, underscores its versatility and effectiveness. As DDoS attacks continue to evolve in complexity and frequency, the demand for innovative and efficient approaches to identify and mitigate these threats has become imperative. The proposed model represents a significant leap in this domain and holds the potential to deliver substantial benefits to cybersecurity practitioners and organizations. Its applicability in real-time scenarios and its capability to effectively mitigate DDoS attack impacts ensure the uninterrupted availability and functionality of vital systems and services.

The methodologies and principles outlined in this research can be extended to identify and mitigate other network threats beyond DDoS attacks. Investigating its applicability to various cybersecurity challenges is a promising direction. Developing a comprehensive framework that not only detects attacks but also initiates adaptive responses and countermeasures in real-time can enhance network security. This proactive approach will be pivotal in addressing evolving cyber threats.

Availability of data and materials

The datasets used in this research are publicly available. We provided all relevant information in the datasets section.

Finding

The authors did not receive any funds for this research.

CRediT authorship contribution statement

Md. Alamgir Hossain: Conceptualization, research implementation, experiment design, data analysis, manuscript drafting, research design, literature review, objective refinement, and methodology shaping. **Md. Saiful Islam:** Research initiation, project supervision, comprehensive review, guidance, academic standard assurance, data interpretation, manuscript finalization, and quality enhancement.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All the dataset are cited properly in the paper.

References

[1] A. Cheema, M. Tariq, A. Hafiz, M.M. Khan, F. Ahmad, M. Anwar, Prevention Techniques against Distributed Denial of Service Attacks in Heterogeneous Networks: A Systematic Review vol. 2022, Security and Communication Networks, May 2022, pp. 1–15, https://doi.org/10.1155/2022/8379532.
[2] K.B. Adedeji, A.M. Abu-Mahfouz, A.M. Kurien, DDoS attack and detection methods in internet-enabled networks: concept, research perspectives, and challenges, JSAN 12 (4) (Jul. 2023) 51, https://doi.org/10.3390/jsan12040051.
[3] A. Aljuhani, Machine learning approaches for combating distributed denial of service attacks in modern networking environments, IEEE Access 9 (Jan. 2021) 42236–42264, https://doi.org/10.1109/ACCESS.2021.3062909.
[4] M.J. Pasha, K.P. Rao, A. MallaReddy, V. Bande, LRDADF: an AI enabled framework for detecting low-rate DDoS attacks in cloud computing environments, Measurement: Sensors 28 (Aug. 2023) 100828, https://doi.org/10.1016/j.measen.2023.100828.
[5] D. Kozhuharova, A. Kirov, Z. Al-Shargabi, Ethics in cybersecurity. What are the challenges we need to Be aware of and how to handle them? in: J. Kołodziej, M. Repetto, A. Duzha (Eds.), Cybersecurity of Digital Service Chains, Lecture Notes in Computer Science, vol. 13300 Springer International Publishing, Cham, 2022, pp. 202–221, https://doi.org/10.1007/978-3-031-04036-8_9, vol. 13300.
[6] R. Uddin, S.A.P. Kumar, V. Chamola, Denial of service attacks in edge computing layers: taxonomy, vulnerabilities, threats and solutions, Ad Hoc Netw. 152 (Jan. 2024) 103322, https://doi.org/10.1016/j.adhoc.2023.103322.
[7] G. Sujatha, Y. Kanchhal, G. George, An advanced approach for detection of distributed denial of service (DDoS) attacks using machine learning techniques, in: 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), IEEE, Trichy, India, Oct. 2022, pp. 821–827, https://doi.org/10.1109/ICOSEC54921.2022.9951944.
[8] Azure Network Security Team, "2022 in review: DDoS attack trends and insights," Microsoft Security. Accessed: May 3, 2023. [Online]. Available: https://www.microsoft.com/en-us/security/blog/2023/02/21/2022-in-review-ddos-attack-trends-and-insights/.
[9] R. Chaganti, B. Bhushan, V. Ravi, A survey on Blockchain solutions in DDoS attacks mitigation: techniques, open challenges and future directions, Comput. Commun. 197 (Jan. 2023) 96–112, https://doi.org/10.1016/j.comcom.2022.10.026.
[10] L.F. Eliyan, R. Di Pietro, DoS and DDoS attacks in Software Defined Networks: a survey of existing solutions and research challenges, Future Generat. Comput. Syst. 122 (Sep. 2021) 149–171, https://doi.org/10.1016/j.future.2021.03.011.
[11] S. Das, A.M. Mahfouz, D. Venugopal, S. Shiva, DDoS intrusion detection through machine learning ensemble, in: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), IEEE, Sofia, Bulgaria, Jul. 2019, pp. 471–477, https://doi.org/10.1109/QRS-C.2019.00090.
[12] R. Abu Bakar, X. Huang, M.S. Javed, S. Hussain, M.F. Majeed, An intelligent agent-based detection system for DDoS attacks using automatic feature extraction and selection, Sensors 23 (6) (Mar. 2023) 3333, https://doi.org/10.3390/s23063333.
[13] S. Bharathidason, C. Jothi Venkataeswaran, Improving classification accuracy based on random forest model with uncorrelated high performing trees, Int. J. Crit. Account. 101 (13) (Sep. 2014) 26–30, https://doi.org/10.5120/17749-8829.
[14] J. Cui, M. Wang, Y. Luo, H. Zhong, DDoS detection and defense mechanism based on cognitive-inspired computing in SDN, Future Generat. Comput. Syst. 97 (Aug. 2019) 275–283, https://doi.org/10.1016/j.future.2019.02.037.
[15] X. Liu, J. Ren, H. He, B. Zhang, Q. Wang, Z. Zheng, All-Packets-Based Multi-Rate DDoS Attack Detection Method in ISP Layer, vol. 2022, Security and Communication Networks, May 2022, pp. 1–18, https://doi.org/10.1155/2022/7551107.
[16] F. Musumeci, A.C. Fidanci, F. Paolucci, F. Cugini, M. Tornatore, Machine-learning-Enabled DDoS attacks detection in P4 programmable networks, J. Netw. Syst. Manag. 30 (1) (Jan. 2022) 21, https://doi.org/10.1007/s10922-021-09633-5.
[17] Seong Soo Kim, A.L.N. Reddy, Statistical techniques for detecting traffic anomalies through packet header data, IEEE/ACM Trans. Netw. 16 (3) (Jun. 2008) 562–575, https://doi.org/10.1109/TNET.2007.902685.
[18] R. Hajtmanek, M. Kontšek, J. Smieško, J. Uramová, One-parameter statistical methods to recognize DDoS attacks, Symmetry 14 (11) (Nov. 2022) 2388, https://doi.org/10.3390/sym14112388.

[19] J. Cheng, Y. Liu, X. Tang, V.S. Sheng, M. Li, J. Li, DDoS attack detection via multi-scale convolutional neural network, Comput. Mater. Continua (CMC) 62 (3) (2020) 1317–1333, https://doi.org/10.32604/cmc.2020.06177.
[20] S. Sambangi, L. Gondi, A machine learning approach for DDoS (distributed denial of service) attack detection using multiple linear regression, in: The 14th International Conference on Interdisciplinarity in Engineering—INTER-ENG 2020, MDPI, Dec. 2020, p. 51, https://doi.org/10.3390/proceedings2020063051.
[21] P.S. Saini, S. Behal, S. Bhatia, Detection of DDoS attacks using machine learning algorithms, in: 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, New Delhi, India, Mar. 2020, pp. 16–21, https://doi.org/10.23919/INDIACom49435.2020.9083716.
[22] I. Ortet Lopes, D. Zou, F.A. Ruambo, S. Akbar, B. Yuan, Towards effective detection of recent DDoS attacks: a deep learning approach, Secur. Commun. Network. 2021 (Nov. 2021) 1–14, https://doi.org/10.1155/2021/5710028.
[23] S. Rajesh, M. Clement, S. S. B., A. S. S. H, J. Johnson, Real-Time DDoS Attack Detection Based on Machine Learning Algorithms, Sep. 27, 2021, https://doi.org/10.2139/ssrn.3974241. Rochester, NY.
[24] K.B. Dasari, N. Devarakonda, Detection of different DDoS attacks using machine learning classification algorithms, ISI 26 (5) (Oct. 2021) 461–468, https://doi.org/10.18280/isi.260505.
[25] N. Ahuja, G. Singal, D. Mukhopadhyay, N. Kumar, Automated DDOS attack detection in software defined networking, J. Netw. Comput. Appl. 187 (Aug. 2021) 103108, https://doi.org/10.1016/j.jnca.2021.103108.
[26] R.R. Nuiaa, S. Manickam, A.H. Alsaeedi, E.S. Alomari, A new proactive feature selection model based on the enhanced optimization algorithms to detect DRDoS attacks, Int. J. Energy a Clean Environ. (IJECE) 12 (2) (Apr. 2022) 1869, https://doi.org/10.11591/ijece.v12i2.pp1869-1880.
[27] C. M Nalayini, J. Katiravan, Detection of DDoS Attack Using Machine Learning Algorithms, Rochester, NY, Jul. 26, 2022. May 12, 2023. [Online]. Available: https://papers.ssrn.com/abstract=4173187.
[28] N. Chavan, M. Kukreja, G. Jagwani, N. Nishad, N. Deb, DDoS attack detection and botnet prevention using machine learning, in: 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, Coimbatore, India, Mar. 2022, pp. 1159–1163, https://doi.org/10.1109/ICACCS54159.2022.9785247.
[29] S. Elgendy, M. Attawiya, O. Haridy, A. Farag, P. Branco, "DTEXNet: Artificial Intelligence-Based Combination Scheme for DDoS Attacks Detection," Presented at the the 35th Canadian Conference on Artificial Intelligence, Canadian AI, Toronto, Mar. 2022.
[30] S.S. Samaan, H.A. Jeiad, Feature-based real-time distributed denial of service detection in SDN using machine learning and Spark, Bulletin EEI 12 (4) (Aug. 2023) 2302–2312, https://doi.org/10.11591/eei.v12i4.4711.
[31] M.Y. Sabir, DDoS Attacks Detection Using Machine Learning, Gandhi Institute of Technology and Management, Visakhapatnam, India, 2023. Apr. 29, 2023. [Online]. Available: http://hdl.handle.net/1828/15046.
[32] V. Hnamte, J. Hussain, An efficient DDoS attack detection mechanism in SDN environment, in: Review, preprint, Jan. 2023, https://doi.org/10.21203/rs.3.rs-2393388/v2.
[33] D. Kumar, R.K. Pateriya, R.K. Gupta, V. Dehalwar, A. Sharma, DDoS detection using deep learning, Procedia Comput. Sci. 218 (2023) 2420–2429, https://doi.org/10.1016/j.procs.2023.01.217.
[34] S. Ahmed, et al., Effective and efficient DDoS attack detection using deep learning algorithm, multi-layer perceptron, Future Internet 15 (2) (Feb. 2023) 76, https://doi.org/10.3390/fi15020076.
[35] S. Balasubramaniam, et al., Optimization enabled deep learning-based DDoS attack detection in cloud computing, Int. J. Intell. Syst. 2023 (Feb. 2023) 1–16, https://doi.org/10.1155/2023/2039217.
[36] M.A. Hossain, M.S. Islam, Ensuring network security with a robust intrusion detection system using ensemble-based machine learning, Array (Jul. 2023) 100306, https://doi.org/10.1016/j.array.2023.100306.
[37] I. Sharafaldin, A.H. Lashkari, S. Hakak, A.A. Ghorbani, Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy, in: 2019 International Carnahan Conference on Security Technology (ICCST), IEEE, CHENNAI, India, Oct. 2019, pp. 1–8, https://doi.org/10.1109/CCST.2019.8888419.
[38] N. Ahuja, DDOS attack SDN Dataset, Mendeley, Sep. 27 (2020), https://doi.org/10.17632/JXPFJC64KR.1.
[39] I. Sharafaldin, A. Habibi Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: *Proceedings Of the 4th International Conference On Information Systems Security And Privacy*, Funchal, SCITEPRESS - Science and Technology Publications, Madeira, Portugal, 2018, pp. 108–116, https://doi.org/10.5220/0006639801080116.
[40] Y. R. Kumbam, "APA-DDoS Dataset." Accessed: Feb. 05, 2023. [Online]. Available: https://www.kaggle.com/datasets/yashwanthkumbam/apaddos-dataset..
[41] DDoS Botnet Attack on IOT Devices." Accessed: Oct. 02, 2023. [Online]. Available: https://www.kaggle.com/datasets/siddharthm1698/ddos-botnet-attack-on-iot-devices.
[42] S. Chernykh, A. Stepnov, O. Lukyanova, Data preprocessing for machine learning in seismology, in: CEUR Workshop Proceedings (CEUR-WS.Org), Khabarovsk, Russia, Sep. 2021.
[43] P, et al., Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[44] P. Araujo, et al., Impact of feature selection methods on the classification of DDoS attacks using XGBoost, JCIS 36 (1) (2021) 200–214, https://doi.org/10.14209/jcis.2021.22.
[45] S. Jiang, L. Wang, Efficient feature selection based on correlation measure between continuous and discrete features, Inf. Process. Lett. 116 (2) (Feb. 2016) 203–215, https://doi.org/10.1016/j.ipl.2015.07.005.
[46] F. Macedo, R. Valadas, E. Carrasquinha, M.R. Oliveira, A. Pacheco, Feature selection using decomposed mutual information maximization, Neurocomputing 513 (Nov. 2022) 215–232, https://doi.org/10.1016/j.neucom.2022.09.101.
[47] E. Odhiambo Omuya, G. Onyango Okeyo, M. Waema Kimwele, Feature selection for classification using principal component analysis and information gain, Expert Syst. Appl. 174 (Jul. 2021) 114765, https://doi.org/10.1016/j.eswa.2021.114765.
[48] M.A. Hossain, Enhanced ensemble-based distributed denial-of-service (DDoS) attack detection with novel feature selection: a robust cybersecurity approach, Artificial Intelligence Evolution 4 (2) (Aug. 2023) 165–186, https://doi.org/10.37256/aie.4220233337.
[49] N.S. Chauhan, Random Forest® — a powerful ensemble learning algorithm. Accessed: Feb. 27, 2023. [Online]. Available:https://www.kdnuggets.com/2020/01/random-forest-powerful-ensemble-learning-algorithm.html.
[50] H. M, S. M.N, A review on evaluation metrics for data classification evaluations, IJDKP 5 (2) (Mar. 2015) 1–11, https://doi.org/10.5121/ijdkp.2015.5201.
[51] Ž.Đ. Vujovic, Classification model evaluation metrics, Int. J. Adv. Comput. Sci. Appl. 12 (6) (2021), https://doi.org/10.14569/IJACSA.2021.0120670.
[52] I.M. De Diego, A.R. Redondo, R.R. Fernández, J. Navarro, J.M. Moguerza, General performance score for classification problems, Appl. Intell. 52 (10) (Aug. 2022) 12049–12063, https://doi.org/10.1007/s10489-021-03041-7.
[53] M.S. Akter, et al., Exploring the vulnerabilities of machine learning and quantum machine learning to adversarial attacks using a malware dataset: a comparative analysis, 2023 IEEE International Conference on Software Services Engineering (SSE). arXiv (May 31, 2023), pp. 222–231, Accessed: Sep. 11, 2023. [Online]. Available: http://arxiv.org/abs/2305.19593.
[54] C. Gigliarano, S. Figini, P. Muliere, Making classifier performance comparisons when ROC curves intersect, Comput. Stat. Data Anal. 77 (Sep. 2014) 300–312, https://doi.org/10.1016/j.csda.2014.03.008.
[55] A.D. Jadhav, V. Pellakuri, Highly accurate and efficient two phase-intrusion detection system (TP-IDS) using distributed processing of HADOOP and machine learning techniques, J Big Data 8 (1) (Dec. 2021) 131, https://doi.org/10.1186/s40537-021-00521-y.
[56] I.H. Sarker, CyberLearning: effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks, Internet of Things 14 (Jun. 2021) 100393, https://doi.org/10.1016/j.iot.2021.100393.
[57] Dr V.G. Krishnan, S. Hemamalini, P. Cheraku, K.H. Priya, S. Ganesan, Dr R. Balamanigandan, Attack detection using DL based feature selection with improved convolutional neural network, IJEER 11 (2) (May 2023) 308–314, https://doi.org/10.37391/ijeer.110209.