Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 31(10): 6045–6070.
DOI: 10.3934/era.2023308
Received: 10 April 2023
Revised: 17 August 2023
Accepted: 23 August 2023
Published: 07 September 2023

Research article

A collaborative prediction approach to defend against amplified reflection and exploitation attacks

Arvind Prasad[1,*], Shalini Chandra[1,*], Ibrahim Atoum[2,*], Naved Ahmad[2] and Yazeed Alqahhas[2]

[1] Department of Computer Science, BBA University, Lucknow, India

[2] Department of Computer Science and Information Systems, AlMaarefa University, Riyadh, Saudi Arabia

* **Correspondence:** Email: arvindbitm@gmail.com, nupur_madhur@yahoo.com, iotoum@mcst.edu.sa.

Abstract: An amplified reflection and exploitation-based distributed denial of service (DDoS) attack allows an attacker to launch a volumetric attack on the target server or network. These attacks exploit network protocols to generate amplified service responses through spoofed requests. Spoofing the source addresses allows attackers to redirect all of the service responses to the victim's device, overwhelming it and rendering it unresponsive to legitimate users. Mitigating amplified reflection and exploitation attacks requires robust defense mechanisms that are capable of promptly identifying and countering the attack traffic while maintaining the availability and integrity of the targeted systems. This paper presents a collaborative prediction approach based on machine learning to mitigate amplified reflection and exploitation attacks. The proposed approach introduces a novel feature selection technique called closeness index of features (CIF) calculation, which filters out less important features and ranks them to identify reduced feature sets. Further, by combining different machine learning classifiers, a voting-based collaborative prediction approach is employed to predict network traffic accurately. To evaluate the proposed technique's effectiveness, experiments were conducted on CICDDoS2019 datasets. The results showed impressive performance, achieving an average accuracy, precision, recall and F1 score of 99.99%, 99.65%, 99.28% and 99.46%, respectively. Furthermore, evaluations were conducted by using AUC-ROC curve analysis and the Matthews correlation coefficient (MCC) statistical rate to analyze the approach's effectiveness on class imbalance datasets. The findings demonstrated that the proposed approach outperforms recent approaches in terms of performance. Overall, the proposed approach presents a robust machine learning-based solution to defend against amplified reflection and exploitation attacks, showcasing significant improvements in prediction accuracy and effectiveness compared to existing approaches.

Keywords: cybersecurity; machine learning; reflection attack; exploitation attack; DDoS attack

1. Introduction

Communication protocols allow computer networks to serve requests from prolific clients over diverse network topologies. While low-security measures in communication protocols can enhance data communication efficiency and facilitate the smooth operation of computer networks, they also introduce significant vulnerabilities. Malicious actors can exploit these vulnerabilities to compromise the security and stability of communication systems. The reflection and exploitation attack is a significant cyberattack method that cybercriminals employ to exploit vulnerabilities in network protocols and systems to generate massive traffic volumes. This flood of malicious traffic is designed to overwhelm the target server or network, leading to various adverse effects on its stability, availability and security. These attacks can cause disruptions in online services, financial losses, reputation damage and potential data breaches. As a result, defending against such attacks and maintaining a robust cybersecurity posture is crucial for organizations to ensure the continued functioning of their digital infrastructure and protect their valuable assets.

In reflection attacks, the attacker sends requests to servers or devices configured to respond to certain types of queries or requests. These servers are often legitimate and publicly accessible. This is achieved by manipulating the packet headers to forge the source IP address. The attacker spoofs the source IP address in the requests, making it appear as if they originate from the target victim's IP address. Assuming the requests are legitimate, the servers send their responses to the victim's IP address, resulting in a flood of traffic directed at the victim. Attackers amplify the attack intensity by smartly selecting those reflector servers that can reply with a large packet size compared to the request packet [1].

In exploitation attacks, an attacker exploits the vulnerability of a network protocol to launch a high-volume attack on the victim server [2]. The attacker leverages the characteristics of the targeted network protocols or services to achieve amplification. Certain protocols, such as the domain name system (DNS) protocol, Network Time Protocol (NTP), Simple Network Management Protocol (SNMP) and others, can generate significantly larger responses than the initial request size. By exploiting these protocols' features, the attacker can amplify the volume of traffic directed at the victim. This amplification effect allows the attacker to overwhelm the victim's network infrastructure, consuming its resources and making it inaccessible to legitimate users.

The combination of reflection and exploitation techniques enables cybercriminals to launch devastating distributed denial of service (DDoS) attacks that can disrupt online services, cause financial losses and impact targeted organizations' reputations. These attacks can be challenging to mitigate due to the widespread availability of vulnerable servers and the ease of spoofing IP addresses. These attacks are generally carried out by exploiting network protocols, such as the Simple Service Discovery Protocol (SSDP), DNS protocol, Lightweight Directory Access Protocol (LDAP), Network Basic Input/Output System (NetBIOS) protocol, Simple Network Management Protocol (SNMP), Microsoft SQL Server Resolution (MCSQLR) protocol, synchronized (SYN) flood attack protocol, User Datagram Protocol (UDP) and Trivial File Transfer Protocol (TFTP).

SSDP enables devices to communicate and share information and helps the user to discover plug and play devices in the network [3]. The proliferation of Internet of Things (IoT) devices in the home or small networks has increased SSDP reflection attacks. Cybercriminals exploit the fragility of the SSDP to generate a high volume of network traffic using IoT devices, such as cameras, smart TVs,

smart cars and smart fridges, to launch amplified reflection attacks [4]. DNS servers provide the IP address of the corresponding domain name. In a DNS amplification attack, an attacker spoofs the IP with the victim's IP. Attackers craft millions of such packets and send them to the DNS reflectors; DNS reflectors send back the reply to the victim instead of the attacker [5], allowing an attacker to launch a DNS amplification attack on the victim. Cybercriminals exploit LDAP servers to launch amplified reflection attacks on the victim server. First, they send a query to an LDAP server in a way that the server sends a large response. Then, the attacker spoofs the request query, making the LDAP server send the reply to the victim [6]. The NetBIOS helps to establish communication between applications in a small network to make an application share their resources. Attackers use the IP spoofing technique to send many requests for name lookup to the NetBIOS name server. Although the request query made by the attacker is small, the server's response is detailed information about the current network and hostname configuration, which is much larger than the request query [7]. The SNMP defines a set of rules for management stations to monitor and control the network devices for the smooth execution of the network management functions requested by the network management stations [8].

Attackers send numerous spoofed queries with a forged IP of the victim to the network devices running the SNMP. The network devices send SNMP responses to the forged address i.e., the victim, in a larger volume to jam the victim's device and network. Attackers exploit the MC-SQLR protocol to launch a volumetric reflection attack. The MC-SQLR is designed to send information about all of the database instances on a Microsoft SQL Server to the clients requesting information about a Microsoft SQL (MSSQL) server database instance. Usually, portmap querying port mapper is a small request, but the reply is multiple times that of the request packet. An attacker takes advantage of it to launch an amplified portmap reflection attack. The remote procedure call (RPC) portmap helps to map the RPC service number with the network port number. The attacker makes continuous spoofed portmapper service requests using the victim's source IP. A Syn flood attack is a Transmission Control Protocol (TCP)-based exploitation attack.

An attacker exploits the TCP three-way handshake vulnerability by continuously send Syn requests to the victim server but does not acknowledge back to any Syn request. It creates a huge number of half-open connections on the server [9]. Half-open connections cause the servers to be inundated and unresponsive to legitimate traffic. Attackers send a huge number of User Datagram Protocol (UDP) packets to the random ports of the victim server. The bombardment of such a flood consumes all of the server's resources [10]. Therefore, the absence of an initial handshake in the UDP packet makes it more attractive to the attacker and enables them to launch an attack in high volume using limited resources. The TFTP enables the transference, downloading or uploading of a file without authentication [11]. The stateless nature, easy implementation and fast transmission rate help to boot diskless workstations, install an operating system or transfer large files. However, the nature of having no authentication and less security is exploited by cybercriminals to launch TFTP amplification volumetric attacks [12]. Cybercriminals send a request to download a file from the TFTP server and, while sending it, they spoof the source IP address with a victim IP address. In this way, all of the TFTP server replies are directed to the victim server. The above-discussed attacks are categorized and illustrated in Figure 1.

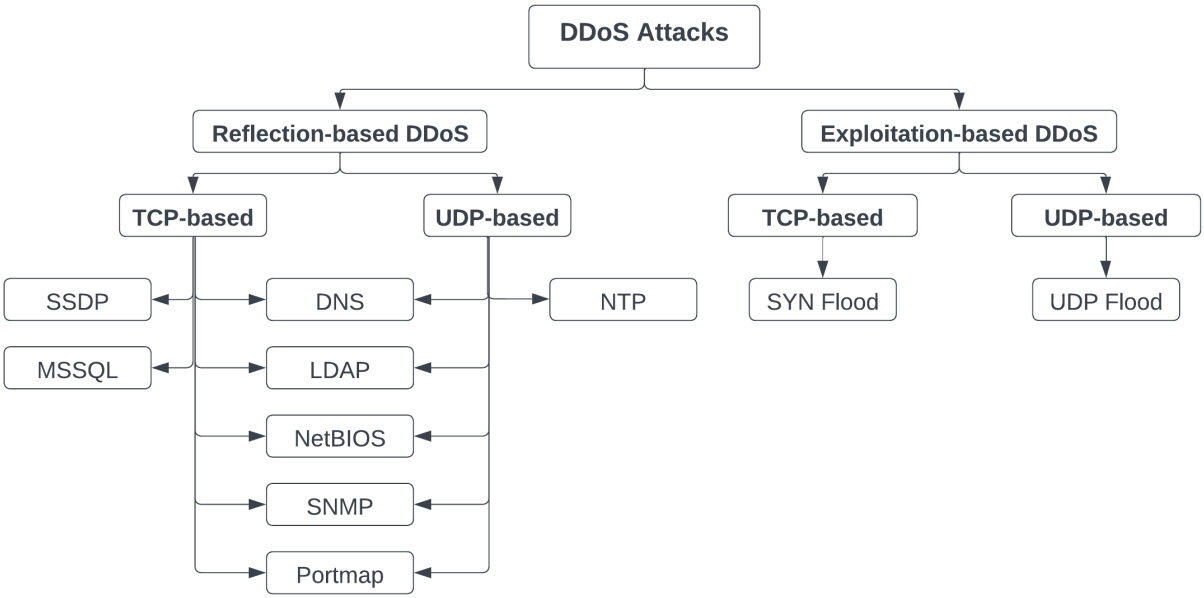

Figure 1. Reflection and exploitation-based DDoS attacks.

The exploitation of the above-discussed protocols is difficult to detect by traditional methods. The possibility of implementing complex security on these protocols is also low, as it can affect the performance of the device implementing the protocol. Therefore, fixing the vulnerabilities in these protocols or developing a modern solution is prone to be overlooked entirely. In recent years, machine learning has evolved as one of the promising solutions for analyzing a tremendous amount of network data to detect sophisticated attacks on the network and network devices. Many researchers have employed machine learning to build powerful techniques to defend against cyberattacks [13–18]. This work uses a collaborative prediction approach to detect and defect amplified reflection and exploitation attacks. The following are key contributions.

- It presents details of the network protocols an attacker exploits for reflection and exploitation attacks.
- A novel closeness index of features (CIF) technique is proposed to rank features.
- The CIF is combined with the Pearson correlation coefficient and a mutual information (MI)-based feature ranking technique to construct reduced feature sets that can give optimal performance for most of the reflection and exploitation attacks.
- A collaborative prediction approach is proposed by implementing a voting classifier to detect attacks.
- The proposed approach has experimented on various reflection and exploitation attack datasets to evaluate the model's effectiveness.
- The proposed model is further evaluated by the area under the curve (AUC) and receiver operating characteristic (ROC) curve and Matthews correlation coefficient (MCC) statistical rate.

The rest of the article is organized as follows: Section 2 discusses machine learning-based approaches closely related to the proposed work. Next, Section 3 gives a detailed description of the

proposed approach. Then, the experimental result is presented in Section 4 and the discussion is presented in Section 5. Finally, Section 6 concludes the proposed work.

2. Related works

Cyberattacks exploiting network protocols have always been a major concern for researchers and industries. Over the period, many security mechanisms have been developed to ensure the security of the network protocols. The section below discusses existing studies that propose defense mechanisms against reflection and exploitation attacks using machine learning techniques.

Thorat et al. [19] proposed TaxoDaCmachine learning, a taxonomy based on the divide and conquer approach that uses a machine learning technique to detects DDoS attacks targeted on transport layer protocol. Dividing bigger classification problems into smaller sub-problems helps the approach to perform efficiently. TaxoDaCmachine learning gives the flexibility to choose different feature sets and various machine learning classifiers to perform the classification. The extensive work on data cleaning and feature selection improves the performance of the proposed approach. Various machine learning classifiers, such as k-nearest neighbor (KNN), decision tree (DT), random forest (RF) and artificial neural network (ANN) algorithms are used at various levels to improve classification accuracy. TaxoDaCmachine learning achieved 99.9% detection accuracy when applied to the CICDDoS2019 dataset. The technique performs classification by using minimum computational cost and time. Ahmed et al. [20] implemented a machine learning technique to detect and mitigate DNS query-based DDoS attacks in software defined networking (SDN). This technique is more suitable for networks, such as military networks, that need high security. In the proposed technique, the SDN controller periodically accesses and analyzes network traffic to find the network traffic features. As a result, Dirichlet process mixture model (DPMM) outperformed the mean shift clustering method in terms of the detection accuracy of network traffic flows and hypertext transfer protocol (HTTP) and file transfer protocol (FTP) traffic flows.

Sreeram and Vuppala [21] proposed a bio-inspired bat algorithm to detect DDoS attacks based on application layer protocols, such as HTTP, DNS, VoIP or SMTP attacks. Unfortunately, DDoS attacks based on application layer protocols follow all of the communication protocols, which makes them difficult to detect. The bio-inspired proposed technique helped to achieve higher accuracy with minimal computational complexity. When experimented on the CAIDA 2007 dataset, it achieved 94.5% precision, 94% recall and 94.8% accuracy. Salman et al. [22] proposed a framework for identifying IoT devices and detecting malicious network traffic. The proposed framework has modules, such as a feature extractor to record the features of active network flow, a module to identify IoT devices to classify the devices based on statistical features of network flows, a traffic-type identification module to classify the generated traffic and an intrusion detection module to profile the normal device behavior to detect abnormal activity. During the experiment, various machine learning classifiers were employed, where RF achieved the highest accuracy, with 94.5% for device-type identification and 93.5% for traffic-type classification. The authors of [23] proposed BLCD, a broad learning-based extensive defense strategy for detecting DDoS attacks based on the SSDP. BLCD incorporates broad learning and a collection of defense strategies to detect malicious traffic, and it reduces the incoming and outgoing network traffic from a device. The defense strategies are deployed on multiple zones, such as senders, routers, service providers, victims, amplifiers and bots. The

proposed technique achieved 99.99% accuracy in detecting malicious traffic.

Ismail et al. [24] presented the weighted score selector (WSS), a lightweight ensemble machine learning approach for detecting cyberattacks in wireless sensor networks (WSNs). WSS implements MI and Kendall's correlation coefficient for identicality reduction and the extraction of an optimal subset of features. The authors employed seven conventional machine learning classifiers to create a pool and experimented with them on the WSN-DS dataset. The approach divided the original dataset into multiple balanced sub-datasets reducing the computational overhead and making the approach suitable for imbalanced datasets. Further, the most effective classifier is selected after analyzing the performance of each classifier from the pool. Kshirsagar and Kumar [25] proposed a machine learning-empowered security framework against DDoS attacks by exploiting TCP and UDP protocols. The thresholds 0.5, 0.25 and 0 were applied in IG and CR-based techniques to get reduced feature sets CRFS-1, CRFS-2 and CRFS-3. Further, CRFS-1 and CRFS-2 were combined to get a new feature set that enhanced the classification performance of the J48 classifier. Mishra et al. [26] proposed a multi-classifier algorithm-based defensive mechanism against different DDoS threats. The authors employed six machine learning classifiers on the CICDoS2019 dataset to detect DDoS attacks. The low variance features with less than a predetermined threshold were removed. Further, the tree-based feature selection approach eliminated unnecessary features and finally selected the top 25 features. The AdaBoost achieved the highest classification accuracy, while a naive bayes algorithm achieved the highest performance speed.

In summary, the existing studies have significantly contributed to the development of defense mechanisms against amplified reflection and exploitation attacks using machine learning techniques. These studies contribute valuable insights and techniques for addressing network security challenges posed by reflection and exploitation attacks, showcasing the potential of machine learning in to enhance defense mechanisms.

While existing studies have made notable advancements, it is clear that improvements are still needed in the following areas:

- **Accuracy improvement:** While the existing studies achieved high detection accuracy in many cases, there is still room for improvement. Future research should focus on developing more accurate models to minimize false positives and false negatives.
- **Feature selection techniques:** Feature selection is critical in improving the performance of machine learning models. Existing approaches may not have explored all possible relevant features or utilized advanced feature selection methods. Developing more effective feature selection techniques could enhance the overall defense mechanism.
- **Collaborative prediction:** Some studies have employed multiple machine learning classifiers to improve accuracy, but further exploration of collaborative predictive methods may yield better results. Ensemble methods or meta-learning techniques combine the strengths of different classifiers effectively.
- **Handling class imbalance data:** Dealing with imbalanced datasets is a common challenge in network security. Existing approaches have addressed this to some extent, but more robust techniques are needed to handle class imbalances and avoid bias in the model.

These identified limitations motivated the researchers to develop defense mechanisms that are both more resilient and efficient in the terms of countering the amplified reflection and exploitation DDoS

attacks. By addressing these critical gaps, the research contributes to the evolution of more robust and impactful solutions in this domain.

3. Methodology

The proposed amplified reflection and exploitation attack detection method is a machine learning-based technique. It has following stages.

Dataset enhancement: Most of the data have multiple missing values, such as null, NaN and NA. This subsection discusses the dataset used in this study and the steps involved in improving the dataset quality.

Feature selection techniques: The technique employs three different feature ranking techniques to identify the most useful features for attack detection. These techniques include the following:

- CIF: This method assesses the relevance and importance of each feature based on its proximity to the target variable or attack label.
- Pearson Correlation Coefficient-based ranking: Features are ranked based on correlation between independent and dependent features.
- MI-based ranking: MI measures the dependency between each feature and the attack label, with higher MI values indicating greater relevance.

Collaborative prediction using VotingClassifier: The selected features are input for three different machine learning classifiers: AdaBoostClassifier, LogisticRegression and BaggingClassifier. A VotingClassifier is employed to make the final prediction. This ensemble technique combines the predictions from the individual classifiers (AdaBoostClassifier, LogisticRegression and BaggingClassifier) and aggregates them by using a majority voting scheme. The VotingClassifier leverages the diversity of the individual classifiers to improve the overall predictive performance. By implementing this approach, the proposed technique aims to enhance the detection of amplified reflection and exploitation attacks.

The following subsection discusses each stage of the proposed technique in detail and the workflow is presented in Figure 2.

3.1. Dataset enhancement

It is crucial to evaluate machine learning models on modern, realistic and large datasets to ensure their real-world performance, generalizability, robustness, scalability and ethical considerations. Considering this view, the proposed approach has been evaluated on the CICDDoS2019 [27] DDoS attack dataset. It includes various modern and realistic DDoS attack traffic profiles. The CICDDoS2019 dataset has a huge amount of network traffic collected from a comprehensive testbed that combines a highly secured victim network and an attack network separated from the victim network. Therefore, the ratio of attack records is very high in each dataset compared to the ratio of benign records, which gives a realistic scenario of a high-volume DDoS attack. In most cases, attack records are more than 99.9% of total records. All datasets have a total of 88 features. Many features, such as 'Bwd PSH Flags' and 'Bwd Avg Bulk Rate' are single-value features; hence they were discarded during the experiments. The proposed technique has been applied to CICDDoS2019's

SSDP, DNS, LDAP, NetBIOS, SNMP, NTP, MSSQL, Portmap, Syn and UDP datasets. An overview of the CICDDoS2019 dataset is given in Table 1.

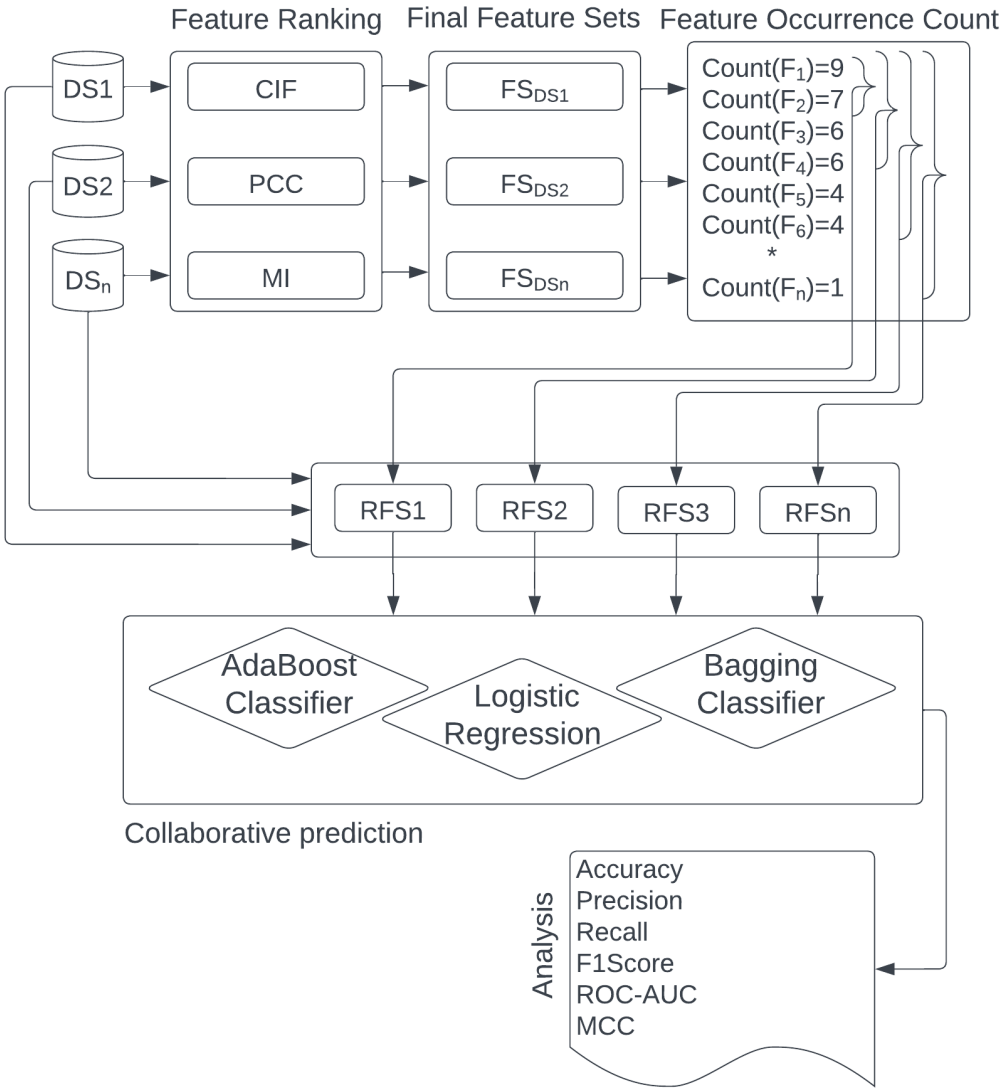

Figure 2. Workflow of the proposed approach.

Table 1. CICDDoS2019 dataset record details.

Dataset	Attack	Benign	Total	Attack %	Benign %
SSDP	2610611	763	2611374	99.97%	0.03%
DNS	5071011	3402	5074413	99.93%	0.07%
LDAP	2179930	16	2181542	99.93%	0.07%
NetBIOS	3454578	1321	3455899	99.96%	0.04%
SNMP	5159870	1507	5161377	99.97%	0.03%
NTP	1202642	14365	1217007	98.82%	1.18%
MSSQL	5772992	2794	5775786	99.95%	0.05%
Portmap	186960	4734	191694	97.53%	2.47%
Syn	4284751	35790	4320541	99.17%	0.83%
UDP	3779072	3134	3782206	99.92%	0.08%

All datasets have a significant amount of missing and infinite values. These values were imputed with zero. Deleting these values can cause a significant amount of data loss. The missing value imputation can also be done by predicting these values using machine learning models. Machine learning models can capture complex relationships between variables, allowing for more accurate imputations. Although it will improve the dataset, it can extra overload on model. Details of the missing and infinite values are given in Figure 3.

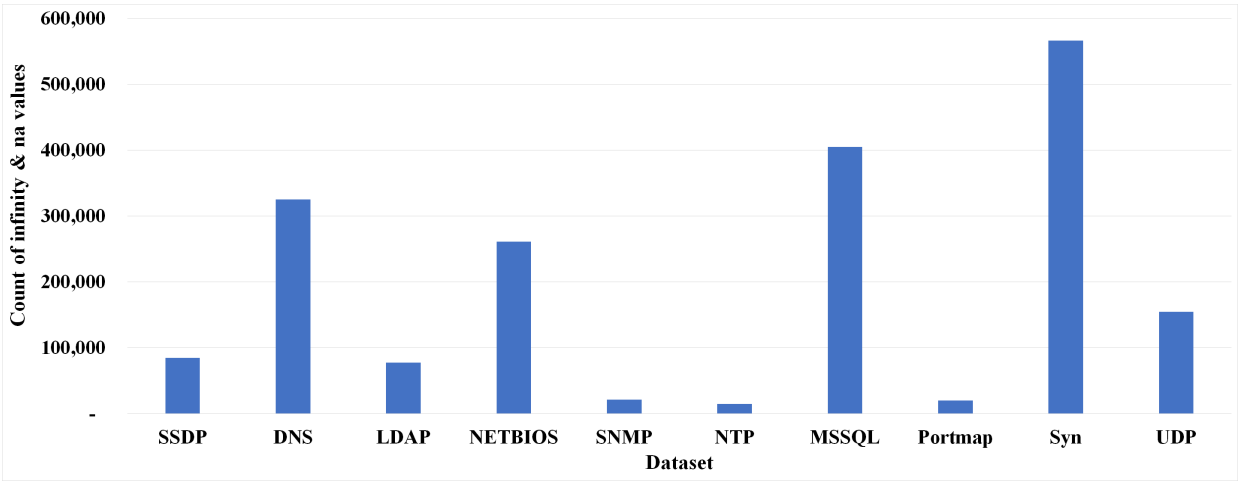

Figure 3. Details of missing and infinite values in the dataset.

3.2. Feature selection techniques

Filtering out unwanted features and selecting important features can help to improve the performance of any machine learning model. These steps become essential when the dataset size is huge and the model needs to analyze these data in real time. The proposed feature selection technique combines three feature ranking techniques: CIF, Pearson correlation coefficient and MI.

Having three different feature ranking techniques in the feature selection process offers several advantages. It allows the model to comprehensively evaluate the relevance of features from various perspectives, capture both linear and nonlinear relationships between features and the target variable, reduce the bias introduced by any single method, handle diverse data types effectively, increase robustness against noise in the data and strike a balance between feature interpretability and predictive performance.

CIF-based ranking: The CIF is determined by calculating the mean value of a feature and subtracting it from each value of that feature. The absolute value of the subtraction is divided by the mean value of the same feature. This process repeats for each feature value and a total is calculated. This total value indicates the closeness of each value of a feature to the mean value of that feature. The equation for calculating the CIF for all features is given by Eq (3.1).

$$CIF = \sum_{i=0}^{p} \sum_{j=0}^{q} \frac{|F[i][j] - mean(F[i])|}{mean(F[i])} \tag{3.1}$$

where

p is the total number of features in the dataset
q is the total number of records in i^{th} feature
F represents all features in the dataset

The CIF values calculated for each feature of the Syn flood attack dataset is shown in Figure 4.

The CIF offers insights into feature distribution and variability, benefiting decision-making and increasing data comprehension. The CIF provides a quantifiable measure of how closely individual feature values cluster around their respective means. It can help to evaluate the importance and relevance of features within a dataset. They might indicate the presence of outliers or extreme values that deviate significantly from the mean. Identifying and investigating these features could be crucial to obtaining an understanding of data quality issues or anomalies.

Pearson correlation coefficient-based ranking: The Pearson correlation coefficient is a statistical measure that quantifies the linear relationship between each feature (independent variable) and the target variable (dependent variable). Features with a higher absolute value the correlation coefficient (closer to 1 or -1) are considered as more strongly correlated with the target variable and are potentially more informative for predictive modeling tasks. On the other hand, features with correlation coefficients close to 0 are less likely to have a strong linear relationship with the target and may be less useful for prediction.

Equation (3.2) is used to calculate Pearson's correlation coefficient between independent feature I and dependent feature D.

$$P_{ID} = \frac{\sum_{i=1}^{n}(I_i - \bar{I})(D_i - \bar{D})}{\sqrt{\sum_{i=1}^{n}(I_i - \bar{I})^2}\sqrt{\sum_{i=1}^{n}(D_i - \bar{D})^2}} \tag{3.2}$$

where

I and D are two features of the dataset
n is the number of records in feature I

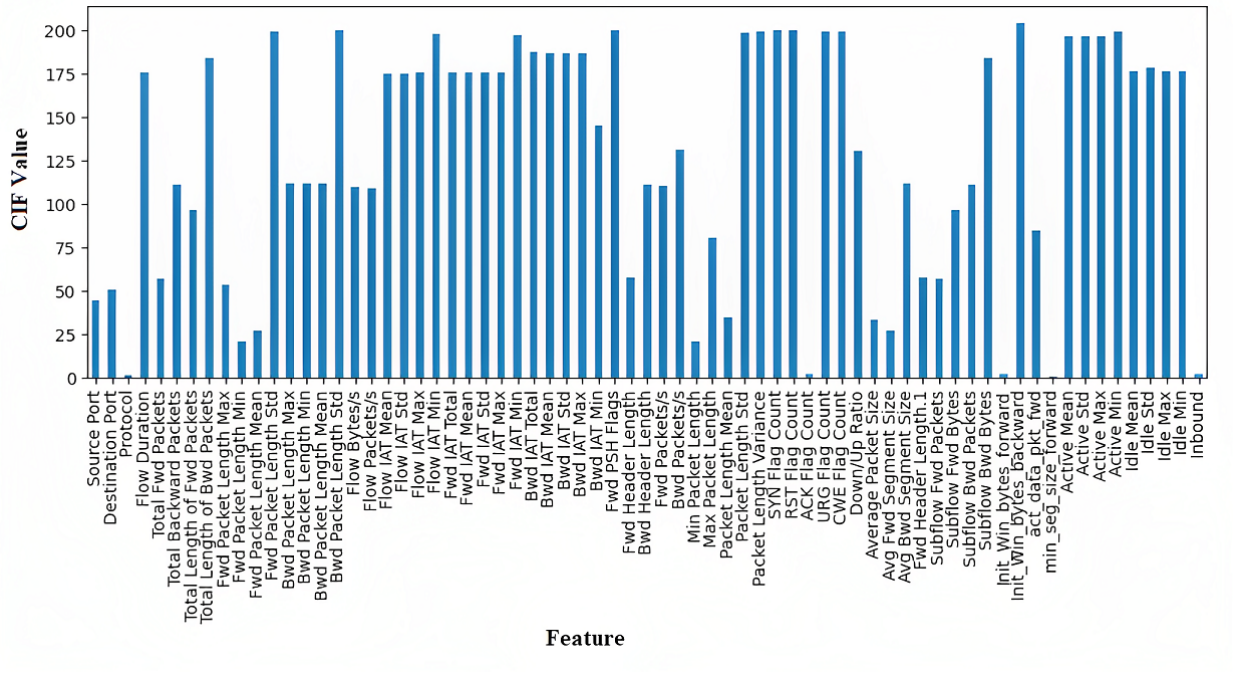

Figure 4. CIF values for Syn flood attack dataset features.

$\bar{I}$ is mean of feature I
$\bar{D}$ is the mean of feature D

The correlation values calculated between each independent feature and dependent feature of the Syn flood attack dataset is shown in Figure 5.

MI-based ranking: The MI between two features shows how much information one feature has about another. For example, the amount of information feature F carries to correctly classify the target label L (benign or attack) is calculated by using Eq (3.3). A higher MI (F, L) value indicates higher importance of the feature [28]. Conversely, when the value of MI (F, L) is zero, feature F now has information about target feature L, which can be removed from the final feature set.

$$MI(F, L) = \sum_{i=1}^{n} \sum_{j=1}^{n} p(F_{\mathrm{i}}, L_{\mathrm{j}}) \log \left(\frac{p(F_{\mathrm{i}}, L_{\mathrm{j}})}{p(F_{\mathrm{i}}) p(L_{\mathrm{j}})} \right) \tag{3.3}$$

where

F represents all of the features in the dataset
L is the label of a record (benign or attack)
i and j are used to iterate all of the features in the dataset

Then, the MI value is calculated for each feature, which helps in the ranking of features based on the amount of information they have about the target feature. Subsequently, an i^{th} feature is selected from each group to create a feature subset. The MI value calculated for each feature of the Syn flood attack dataset is shown in Figure 6.

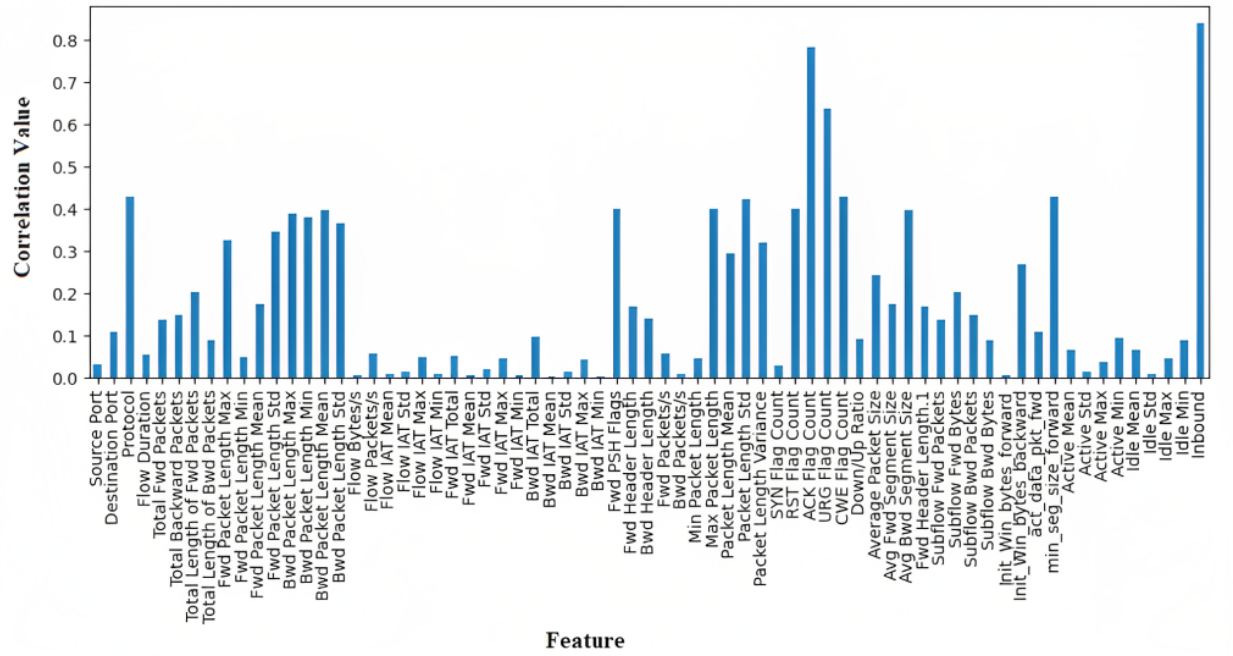

Figure 5. Correlation values of Syn flood attack dataset features.

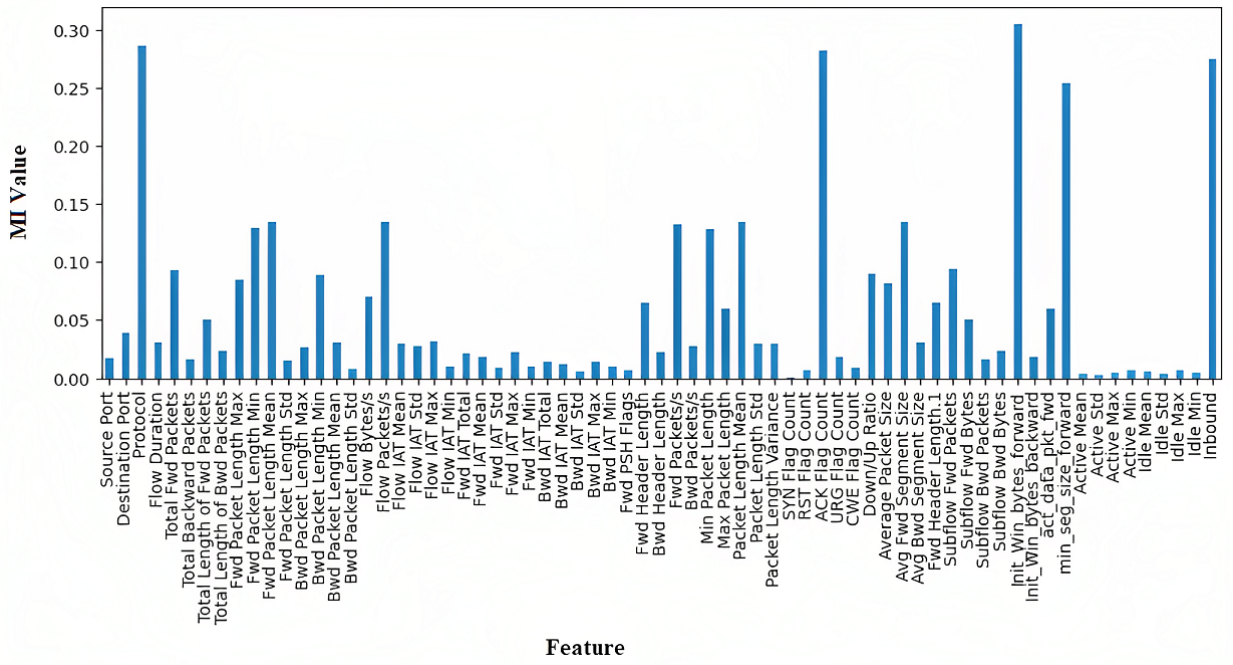

Figure 6. MI values for Syn flood attack dataset features.

Final high ranked feature identification: After ranking features using the CIF, LASSO and MI techniques, a loop is used to iterate from i=0 to the length of features. The 0th feature denotes the highest-ranked feature. At each i^{th} step, the ith feature is extracted from the CIF, LASSO and MI

feature sets and all unique values are added to the feature set. AdaBoostClassifier is implemented to find the classification accuracy of the feature set at each i^{th} step. The current classification accuracy is compared with the previous classification accuracy. At the first iteration, the current classification accuracy is compared with 0. If the current classification accuracy is higher than the previous accuracy, then all of the unique i^{th} features are included in the final feature set.

This way, a final feature set is identified for all of the datasets used in the experiment.

Reduced feature set construction: Once the final feature set is identified for all datasets, the researcher can calculate the occurrence of each feature based on all the final feature sets. A feature with the highest number of occurrences shows the highest importance of that feature for most of the dataset. Features are again ranked based on their occurrence count in descending order. A loop is used to iterate from 1 to n, where n is the total number of features. It creates an (n-1) number of final reduced feature sets. At each i^{th} step, all of the features from the 0^{th} to i^{th} position are included in the i^{th} reduced feature set.

This way, multiple reduced feature sets are constructed and experimented via the proposed collaborative prediction technique. The best-performing reduced feature set is finally selected for the proposed reflection and exploitation attack detection technique.

This approach yields reduced feature sets that might not exhibit the best performance during a particular attack, but they ensure optimal performance for all types of reflection and exploitation attacks, such as SSDP, DNS, LDAP, NetBIOS, SNMP, NTP, MSSQL, Portmap, Syn and UDP attacks.

The rankings of features can differ across techniques, with some methods assigning higher importance to certain features and lower importance to others. Relying solely on one ranking technique risks missing crucial features. To address this, the proposed feature selection technique utilizes three distinct ranking methods, reducing the chance of losing vital features and providing a more comprehensive assessment of feature importance. By combining these approaches, a more robust feature selection process is achieved, ensuring that key features are retained for subsequent analysis and modeling.

3.3. The collaborative prediction technique

The proposed collaborative prediction approach implements machine learning techniques to classify network traffic and detects reflection and exploitation attacks. Machine learning has emerged as a widely explored area in recent years. The extensive training of the machine learning model on diverse sets of network traffic allows it to detect malicious behavior of the network [29, 30]. The proposed technique implements a voting classifier to improve attack detection capabilities. The voting classifier combines various base machine learning classifiers, such as AdaBoostClassifier, LogisticRegression and BaggingClassifier to build a robust ensemble model that can achieve a higher classification accuracy on diverse datasets.

The three selected classifiers belong to different classifier types, providing diversity in the voting classifier. AdaBoostClassifier is an ensemble method that combines multiple weak learners to create a strong learner; LogisticRegression is a linear model for binary classification and BaggingClassifier is another ensemble method that uses bootstrap aggregation. By including classifiers from different types, the potential strengths across the ensemble can be captured. Each classifier may have its own strengths and weaknesses. By combining AdaBoostClassifier, LogisticRegression and BaggingClassifier, their individual strengths can be leveraged to improve overall performance. For

example, AdaBoostClassifier is known for its ability to handle complex relationships and outliers, LogisticRegression can work well with linearly separable data and BaggingClassifier can reduce variance and improve stability. The combination of these classifiers potentially allows us to benefit from their complementary strengths.

Ensemble methods, such as AdaBoostClassifier and BaggingClassifier, are known for their ability to reduce overfitting and improve generalization. They achieve this by aggregating predictions from multiple models. Including ensemble methods in the voting classifier increases the likelihood of obtaining more robust and generalizable predictions. LogisticRegression is a widely used classifier known for its interpretability and simplicity. It provides coefficients that indicate the impact of each feature on the target variable, making it easier to interpret and understand the model. Incorporating LogisticRegression into the voting classifier allows us to benefit from its simplicity and interpretability.

In recent years, various researchers have extensively experimented with these base machine learning classifiers to detect network attacks [31–34]. A voting classifier was constructed by implementing both 'hard' and 'soft' voting. Voting 'hard' entails opting for the prediction yielded by the maximum base classifiers, whereas 'soft' voting makes predictions based on the sum of the prediction probabilities by base classifiers. The algorithm is applied to all reduced feature sets identified during feature selection. The pseudo-code to detect amplified reflection and exploitation attacks is given in Algorithm 1.

Algorithm 1 The collaborative prediction technique

Require: Reduced feature sets (RFS): [Set1, Set2, Set3, Set4]
df: dataset
label: record classification (benign or malicious)
Ensure: Classification result for [Set1, Set2, Set3, Set4]

$Estms \Leftarrow Estms.append((AdaBoostClassifier(), LogisticRegression(), BaggingClassifier())$
$Models \Leftarrow Models.append((VotingClassifier(estimators = Estms, voting =' hard'), VotingClassifier(estimators = Estms, voting =' soft'))$
for i in range(len(RFS)) **do** ▷ Training model using Set1, Set2, Set3 and Set 4
    $nDF \Leftarrow pd.DataFrame(df[RFS[i], label]).copy()$
    $Train, Test \Leftarrow train\_test\_split(nDF, test\_size = 0.3)$
    $xTrain, yTrain \Leftarrow Train[RFS[i]], Train[label]$
    $xTest, yTest \Leftarrow Test[RFS[i]], Test[label]$
    for model in range(Models) **do** ▷ Training voting hard and voting soft model
        $model.fit(xTrain, yTrain)$
        $predict \Leftarrow model.predict(xTest)$
        $accuracy \Leftarrow metrics.accuracy\_score(yTest, predict)$
        $precision \Leftarrow metrics.precision\_score(yTest, predict)$
        $recall \Leftarrow metrics.recall\_score(yTest, predict)$
        $F1score \Leftarrow metrics.f1\_score(yTest, predict, zero\_division = 1)$
        $Store \Leftarrow accuracy, precision, recall, F1score$
    end for
end for

4. Results

This section describes the proposed feature selection technique and collaborative prediction technique that were applied to CICDDoS2019's SSDP, DNS, LDAP, NetBIOS, SNMP, NTP, MSSQL, Portmap, Syn and UDP datasets to evaluate their performance. The most important feature of all of the individual datasets of CICDDoS2019's dataset was initially identified. The identified features are shown in Table 2.

Table 2. List of identified features for all individual datasets.

Dataset	Feature sets	Accuracy
SSDP	Init_Win_bytes_backward, FlowIATStd, Protocol	0.99989
DNS	Init_Win_bytes_backward, DestinationPort, Inbound	0.99993
LDAP	Init_Win_bytes_backward, DestinationPort, act_data_pkt_fwd	0.99974
NetBIOS	Init_Win_bytes_backward, DestinationPort, Protocol	0.99993
SNMP	Init_Win_bytes_backward, DestinationPort, Inbound	0.99995
NTP	Init_Win_bytes_backward, FlowIATStd, Protocol, BwdHeaderLength, FwdIATStd, Inbound, SYNFlagCount, FlowIATMean, FwdPacketLengthMax	0.99915
MSSQL	Init_Win_bytes_forward, DestinationPort, Protocol	0.99997
Portmap	Init_Win_bytes_backward, DestinationPort, AveragePacketSize, SYNFlagCount, FlowBytes/s, MinPacketLength, ActiveStd, SourcePort, PacketLengthMean	0.99968
Syn	Init_Win_bytes_backward, FlowIATStd, Init_Win_bytes_forward	0.99938
UDP	Init_Win_bytes_backward, FlowIATStd, Inbound	0.99986

Once the best-performing feature was identified for all datasets, the total occurrence of an individual feature across all datasets was calculated. Based on the total occurrence, a ranking of the feature was assigned. Once ranked, multiple feature subsets were created by including each lower-ranked feature. This way, the final reduced feature sets were created, which are given in Table 3.

After identifying the reduced feature sets, i.e., Set1, Set2, Set3 and Set4, an experiment was conducted by using a voting classifier that implemented voting hard and soft. The voting classifier combined AdaBoostClassifier, LogisticRegression and BaggingClassifier as the base classifier. Each dataset was split into training and test data at a 70:30 ratio, where 70% of the dataset was used to train the model and 30% was used to test the model.

Evaluation metrics, such as accuracy, precision, sensitivity, F1 score and MCC were used to evaluate model performance. These evaluation indicators are commonly used to evaluate machine learning classifiers. Evaluating a model based on these performance indicators is essential. They provide a quantitative assessment of machine learning model performance, enabling the comparison, selection and monitoring of models. They play a crucial role in guiding the development and deployment of effective machine learning systems. Evaluation metrics help one to compare and choose the best model among multiple machine learning classifiers. Different models may perform differently based on the chosen metric, so these metrics provide a basis for model selection. It allows us to monitor the model's performance over time. It is important to ensure that the model maintains its effectiveness as new data

become available. At the same time, these evaluation metrics help assess the real-world impact of deploying a model. For example, precision and sensitivity metrics can help to estimate the cost and benefits of implementing a model in a specific application domain.

Table 3. Best clusters from each subcluster.

Features	Count	Group	Final feature set
Init_Win_bytes_backward	9	1	**Set1:** [All features from Group 1]
DestinationPort	6		
FlowIATStd	4		
Inbound	4	2	**Set2:** [All features from Group 1 & 2]
Protocol	4		
Init_Win_bytes_forward	2		
SYNFlagCount	2	3	**Set3:** [All features from Group 1, 2 & 3]
act_data_pkt_fwd	1		
ActiveStd	1		
AveragePacketSize	1		
BwdHeaderLength	1		
FlowBytes	1		
FlowIATMean	1	4	**Set4:** [All features from Group 1, 2, 3 & 4]
FwdIATStd	1		
FwdPacketLengthMax	1		
MinPacketLength	1		
PacketLengthMean	1		
SourcePort	1		

$$Accuracy = \frac{T_{\mathrm{p}} + T_{\mathrm{n}}}{T_{\mathrm{p}} + T_{\mathrm{n}} + F_{\mathrm{p}} + F_{\mathrm{n}}} \tag{4.1}$$

$$Precision = \frac{T_{\mathrm{p}}}{T_{\mathrm{p}} + F_{\mathrm{p}}} \tag{4.2}$$

$$Recall = \frac{T_{\mathrm{p}}}{T_{\mathrm{p}} + F_{\mathrm{n}}} \tag{4.3}$$

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.4}$$

$$MCC = \frac{(T_{\mathrm{p}} * T_{\mathrm{n}}) - (F_{\mathrm{p}} * F_{\mathrm{n}})}{\sqrt{(T_{\mathrm{p}} + F_{\mathrm{p}}) * (T_{\mathrm{p}} + F_{\mathrm{n}}) * (T_{\mathrm{n}} + F_{\mathrm{p}}) * (T_{\mathrm{n}} + F_{\mathrm{n}})}} \tag{4.5}$$

where

T_{p} represents a benign record correctly classified as benign
T_{n} represents an attack record correctly classified as attack
F_{p} represents a benign record incorrectly classified as attack

F_n represents an attack record incorrectly classified as benign

Based on the above-discussed evaluation metrics, accuracy, precision, sensitivity and F1 score can be calculated. Table 4 shows the voting hard and soft model's average performance on CICDDoS2019's SSDP, DNS, LDAP, NetBIOS, SNMP, NTP, MSSQL, Portmap, Syn and UDP datasets. The same result is depicted in Figure 7.

Table 4. Average performance of voting hard and voting soft model.

Model	Metric	Set1	Set2	Set3	Set4
Voting Hard	Accuracy	0.99927	0.99968	0.99985	0.99997
	Precision	0.95073	0.9935	0.99662	0.99655
	Recall	0.69778	0.92096	0.94834	0.99284
	F1 score	0.79417	0.95533	0.97158	0.99468
Voting Soft	Accuracy	0.99922	0.99981	0.99994	0.99997
	Precision	0.98244	0.99707	0.99692	0.99704
	Recall	0.56901	0.93889	0.96575	0.99014
	F1 score	0.69658	0.96659	0.9809	0.99356

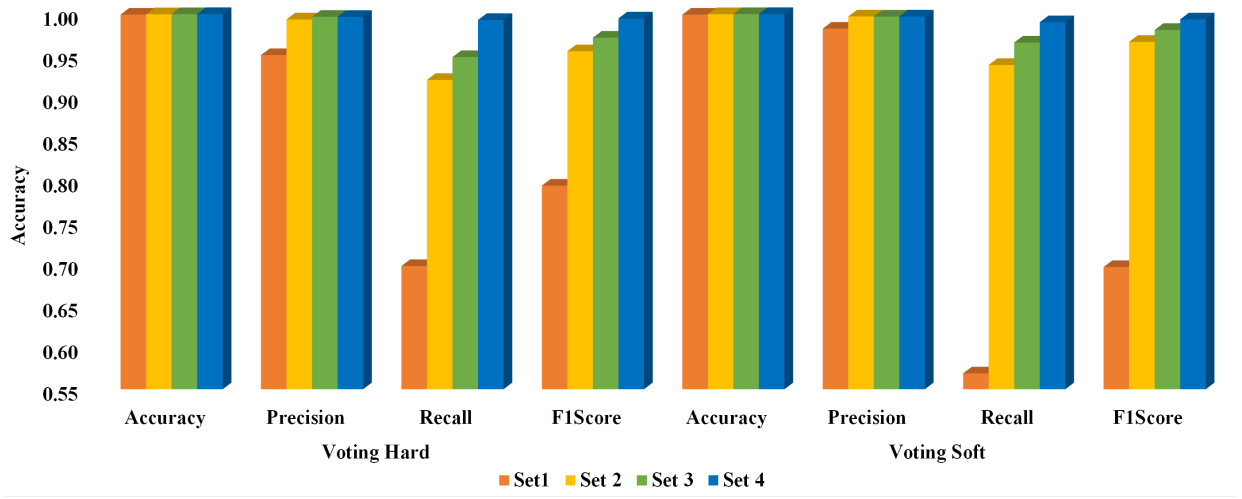

Figure 7. Average classification performance of voting hard and voting soft model.

From Figure 7, it is clear that the reduced feature set Set4 outperformed all of other feature sets. Feature set Set4 combines all of the features identified as reduced feature sets on all individual datasets. Selecting all of these features helped to the model improve classification performance across all datasets. Although the accuracy is similar for all feature sets, there is a huge difference in Recall and F1 score. An adequately built machine learning model is supposed to perform well for all evaluation metrics and selecting feature set Set4 helps to achieve this.

When comparing the hard and soft voting of the voting classifier using the best-performing feature set Set4, it was identified that voting hard achieved better performance than voting soft. Moreover, the majority voting-based voting hard technique performed better than the probability value-based voting soft technique. The results in Figure 8 also depicts this tendency.

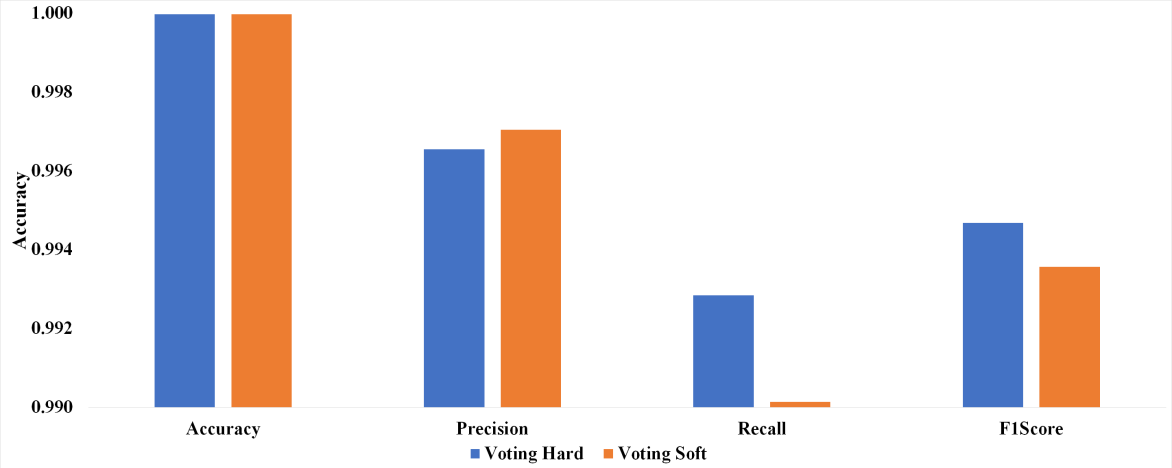

Figure 8. Comparison between voting hard and voting soft on feature set Set4.

The average time that the voting classifier (hard and soft voting classifier) took on all of the datasets was calculated. Then, the ratio of the time taken by both voting hard and voting soft was calculated and, based on that, the graph in Figure 9 was plotted. Figure 9 shows that the times taken by the two algorithms were almost similar. The experimental results demonstrated that, voting hard took less detection time on some datasets; in some cases, it was equal to that of the soft voting algorithm. However, voting soft took less detection time, in many cases, than the voting hard algorithm.

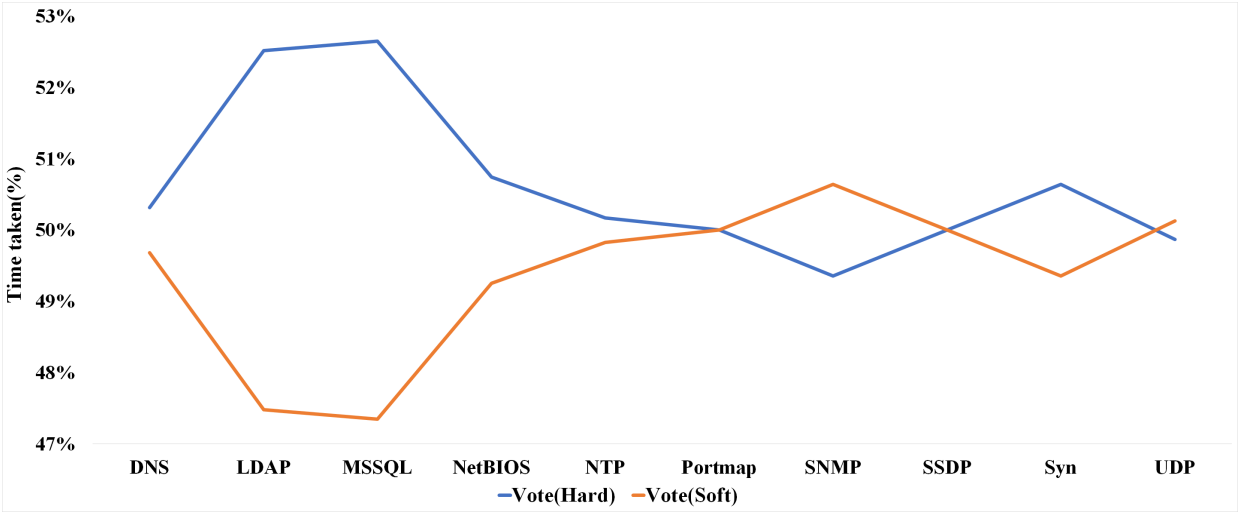

Figure 9. Average detection time of voting hard and voting soft model.

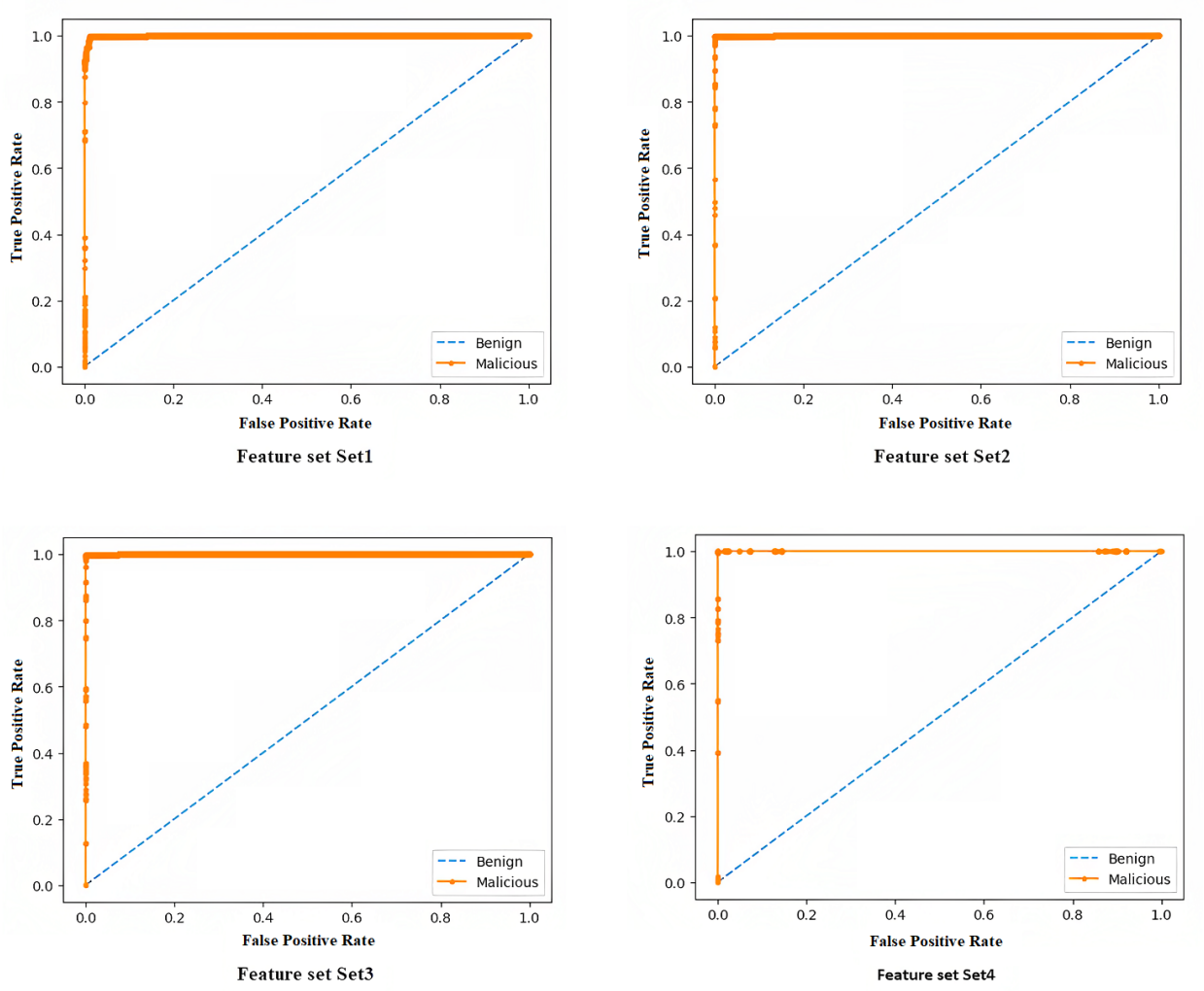

Figure 10. AUC-ROC curve analysis of the proposed model.

AUC-ROC curve analysis: All sub-datasets of the CICIDS2019 dataset are class-imbalanced datasets. Averaging revealed that 99.79% of the records are malicious and only 0.21% are benign, as shown in Table 1. Any machine learning model, even achieving an accuracy of 99%, cannot be concluded as an effective model. The AUC-ROC curve analysis measures the performance of a machine-learning model [35] and it confirmed that the model was performing as expected, even on class-imbalanced datasets. AUC-ROC evaluates how well the model performs on both classes, not favoring the majority class. In class-imbalanced datasets, one class (majority class) has significantly more samples than the other class (minority class). While accuracy can be misleading in class-imbalanced scenarios, AUC-ROC curve analysis offers a more robust and informative way to assess the performance of machine learning models. It ensures that the imbalanced nature of the data does not skew the model's performance and provides a clearer picture of a model's ability to handle such challenges.

The AUC-ROC graph in Figure 10 was calculated and plotted to evaluate the model's effectiveness. During the experiment, the model achieved higher AUC-ROC on all of the feature sets except for

feature set Set1.

MCC analysis: The MCC is a dependable and elegant way to assess the classification performance of a machine learning model [36]. The calculation is based on all four values in the confusion matrix: true positive, true negative, false positive and false negative. Achieving higher values for accuracy, precision, recall or the F1 score on balanced datasets does not always guarantee the machine learning model's efficiency, especially on imbalanced datasets. It is particularly useful for imbalanced datasets and provides a balanced measure of the model's performance.

The MCC ranges from -1 to +1, where +1 indicates a perfect prediction, 0 represents a random prediction and -1 denotes a complete disagreement between the prediction and the true labels. However, a higher MCC value indicates a more informative and truthful score, and it ensures the model's superiority, especially on imbalanced datasets. The MCC value was calculated for all four reduced feature sets to analyze the model's performance. The model achieved the highest MCC value on reduced set Set4, which shows the model's effectiveness. The comparative analysis based on the MCC value is represented in Figure 11 for all four sets.

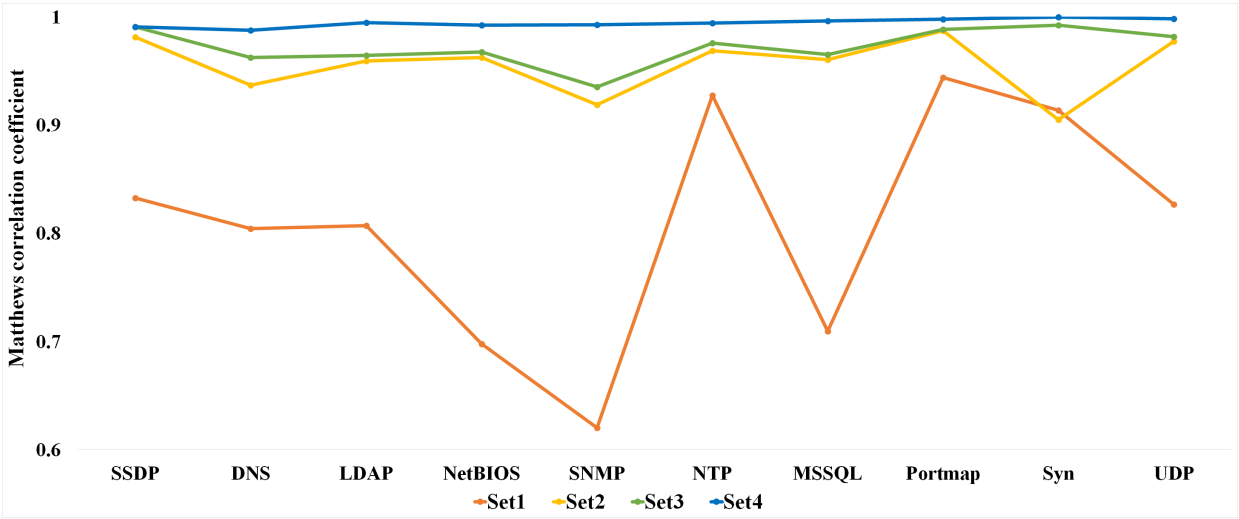

Figure 11. MCC analysis of the model.

In all of the experiments conducted, regardless of using hard or soft voting models, the classification performance was consistently superior when using feature set Set4 compared to Sets 1, 2, or 3. The AUC-ROC curve and MCC analysis also confirmed that Set4 had a significant role in improving the performance of both algorithms. Furthermore, the highest classification accuracy overall was achieved when the voting classifier was implemented with a hard-voting technique using reduced feature set Set4, as shown in Table 5.

Table 5. Performance of voting hard model on various datasets using feature set Set4.

Dataset	Accuracy	Precision	Recall	F1 score
SSDP	0.99999	0.99535	0.98618	0.99074
DNS	0.99998	0.99489	0.98085	0.98782
LDAP	0.99999	1	0.9898	0.99487
NetBIOS	0.99999	0.99024	0.9951	0.99267
SNMP	1	1	0.98569	0.99279
NTP	0.99988	0.99433	0.99527	0.9948
MSSQL	1	0.99394	1	0.99696
Portmap	0.99989	0.99788	0.99788	0.99788
Syn	1	0.9999	0.99981	0.99986
UDP	1	0.99894	0.99788	0.99841
Average	0.99997	0.99655	0.99284	0.99468

5. Discussion

The results of comparative analysis of the proposed approach with state-of-the-art techniques is given in Table 6. It shows that the proposed approach achieved high accuracy compared to most studies. The extensive work on feature selection significantly promoted the selection of only those features that contributed to the machine learning model to improve the classification accuracy. Reducing the feature size improved the computational efficiency, making the model perform the detection at high speed. Detection speed plays a major role, especially when the model is dedicated to defending against cyberattacks in high-speed networks with the possibility of volumetric attacks. In various studies [37, 38], authors have experimentally shown that ensemble classifiers perform better than base classifiers. One machine learning classifier achieving significant classification accuracy on a dataset cannot guarantee that the same classifier will achieve the same accuracy on any dataset. The experiments have shown that combining boosting, bagging and base classifiers in the voting classifier improved the collaborative predictive model. The proposed collaborative approach ensures that, even if one machine learning classifier performs poorly, the other two classifiers will not significantly degrade the final prediction.

The AUC-ROC curve analysis in Figure 10 shows that the model correctly distinguished between benign and attack records when applied to the reduced feature set Set4. The higher AUC-ROC curve analysis graph in Figure 10 shows the predictive power of the classifier. In the case of the reduced feature set Set4, the significantly high MCC values on all datasets, as shown in Figure 11, confirm the prediction capability of the proposed approach on all datasets. It shows the model's effectiveness on diverse datasets, including imbalanced datasets where records from a particular class are numerous.

Table 6. Comparative analysis of the proposed approach with state-of-the-art techniques.

Study	Application area	Feature selection	Techniques	Accuracy
[24]	Computer network	Information gain and correlation	J48 classifier	**99.99%**
[39]	D2D	Features specific to D2D network	RF, LightGBM, XGBoost, & AdaBoost	99.5%
[40]	Computer network	Yes	Machine and Deep learning	99.94%
[41]	IoT	Feature selection RFR	SVM, KNN, DT, NB, RF, & LR	99%
[42]	Computer network	Sparse autoencoder & normalization	DNN	98%
[43]	Computer network	Ensemble of feature selection	DNN	99.6%
[44]	Computer network	Yes	CuDNNLSTM and CuDNNGRU	99.74%
Proposed	Computer network	CIF, mean-value & MI	AdaBoost, LR, & Bagging	**99.99%**

6. Conclusions and future work

This research article addressed the critical issue of amplified reflection and exploitation attacks, which pose significant threats to network security. By exploiting network protocols and reflector servers, these attacks can lead to severe disruptions and downtime, rendering the network unresponsive to legitimate users. To counter these sophisticated attacks, a novel machine learning-based approach was developed by leveraging the CIF technique for feature selection, effectively filtering out less important features and identifying reduced feature sets. Combined with a collaborative prediction approach using a voting classifier, we were able to predict network traffic and detect potential attacks accurately. Additionally, the proposed collaborative prediction approach, implemented using a voting classifier, harnesses the diverse strengths of AdaBoostClassifier, LogisticRegression and BaggingClassifier algorithms, resulting in improved detection accuracy and reliability. The experimental evaluations conducted on the CICDDoS2019 datasets showcased impressive results, with the average values of the accuracy, precision, recall and F1 score exceeding 99%. Furthermore, the use of AUC-ROC curve analysis and MCC statistical rate demonstrated the superiority of the approach, surpassing the performance of existing methods, particularly on class-imbalanced datasets. These findings solidify the efficacy and reliability of the machine learning-based solution in terms of defending against amplified reflection and exploitation attacks. The key contributions of this research lie in the introduction of the CIF technique for feature selection, which effectively filters out less important features and ranks them to identify the reduced feature sets. This technique aids in improving the efficiency and effectiveness of the subsequent predictive stage. Additionally, the collaborative prediction approach, implemented using a voting classifier, harnesses the diverse strengths of AdaBoostClassifier, LogisticRegression and BaggingClassifier algorithms, resulting in improved detection accuracy and reliability.

In conclusion, the proposed research presents a comprehensive and effective machine learning-

based defense mechanism for mitigating amplified reflection and exploitation attacks. The research findings provide network administrators and security practitioners with a valuable tool to safeguard critical network infrastructures, ensuring the availability and integrity of services even in the presence of sophisticated DDoS attacks. Future research endeavors should focus on optimizing the detection speed aspect while continuing to enhance the overall performance and adaptability of the proposed approach to evolving cyber threats. By continuously enhancing and evolving defense mechanisms, staying one step ahead of attackers is possible, ensuring the security and stability of networked systems in the ever-changing cybersecurity landscape.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgements

Ibrahim Atoum would like to express his gratitude to AlMaarefa University, Riyadh, Saudi Arabia for publication support.

Conflict of interest

The authors declare no conflict of interest.

References

1. Y. Jia, F. Zhong, A. Alrawais, B. Gong, X. Cheng, FlowGuard: An intelligent edge defense mechanism against IoT DDoS attacks, *IEEE Internet Things J.*, **7** (2020), 9552–9562. https://doi.org/10.1109/JIOT.2020.2993782
2. A. Prasad, S. Chandra, Machine learning to combat cyberattack: a survey of datasets and challenges, *J. Def. Model. Simul. Appl. Methodol. Technol.*, **2022** (2022). https://doi.org/10.1177/15485129221094881
3. H. Wang, H. He, W. Zhang, W. Liu, P. Liu, A. Javadpour, Using honeypots to model botnet attacks on the internet of medical things, *Comput. Electr. Eng.*, **102** (2022), 108212. https://doi.org/10.1016/j.compeleceng.2022.108212
4. Y. Lee, H. Chae, K. Lee, Countermeasures against large-scale reflection DDoS attacks using exploit IoT devices, *Automatika*, **62** (2021), 127–136. https://doi.org/10.1080/00051144.2021.1885587
5. M. Anagnostopoulos, S. Lagos, G. Kambourakis, Large-scale empirical evaluation of DNS and SSDP amplification attacks, *J. Inf. Secur. Appl.*, **66** (2022), 103168. https://doi.org/10.1016/j.jisa.2022.103168
6. K. B. Dasari, N. Devarakonda, Detection of different DDoS attacks using machine learning classification algorithms, *Ing. Des Syst. d Inf.*, **26** (2021), 461–468. https://doi.org/10.18280/isi.260505

7. C. Rossow, Amplification hell: Revisiting network protocols for DDoS abuse, in*NDSS*, (2021), 1–15.
8. J. D. Case, M. Fedor, M. L. Schoffstall, J. Davin, Simple network management protocol (SNMP), 1989.
9. D. Kshirsagar, S. Sawant, A. Rathod, S. Wathore, CPU load analysis & minimization for TCP SYN flood detection, *Procedia Comput. Sci.*, **85** (2016), 626–633. https://doi.org/10.1016/j.procs.2016.05.230
10. S. Muthurajkumar, A. Geetha, S. Aravind, H. Barakath Meharajnisa, UDP flooding attack detection using entropy in software-defined networking, in *Proceedings of International Conference on Communication and Computational Technologies*, Springer, (2023), 549–560. https://doi.org/10.1007/978-981-19-3951-8_42
11. N. N. Mohamed, Y. Mohd Yussoff, M. A. Mat Isa, H. Hashim, Extending hybrid approach to secure Trivial File Transfer Protocol in M2M communication: a comparative analysis, *Telecommun. Syst.*, **70** (2019), 511–523. https://doi.org/10.1007/s11235-018-0522-5
12. H. Aydın, Z. Orman, M. A. Aydın, A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment, *Comput. Secur.*, **118** (2022), 102725. https://doi.org/10.1016/j.cose.2022.102725
13. S. Pundir, M. S. Obaidat, M. Wazid, A. K. Das, D. P. Singh, J. Rodrigues, MADP-IIME: malware attack detection protocol in IoT-enabled industrial multimedia environment using machine learning approach, *Multimedia Syst.*, **29** (2023), 1785–1797. https://doi.org/10.1007/s00530-020-00743-9
14. M. Gallagher, N. Pitropakis, C. Chrysoulas, P. Papadopoulos, A. Mylonas, S. Katsikas, Investigating machine learning attacks on financial time series models, *Comput. Secur.*, **123** (2022), 102933. https://doi.org/10.1016/j.cose.2022.102933
15. A. Prasad, S. Chandra, VMFCVD: An optimized framework to combat volumetric DDoS attacks using machine learning, *Arabian J. Sci. Eng.*, **47** (2022), 9965–9983. https://doi.org/10.1007/s13369-021-06484-9
16. C. S. Kalutharage, X. Liu, C. Chrysoulas, N. Pitropakis, P. Papadopoulos, Explainable AI-based DDOS attack identification method for IoT networks, *Computers*, **12** (2023), 32. https://doi.org/10.3390/computers12020032
17. A. Prasad, S. Chandra, BotDefender: A collaborative defense framework against botnet attacks using network traffic analysis and machine learning, *Arabian J. Sci. Eng.*, (2023). https://doi.org/10.1007/s13369-023-08016-z
18. M. Bhattacharya, S. Roy, A. K. Das, S. Chattopadhyay, S. Banerjee, A. Mitra, DDoS attack resisting authentication protocol for mobile based online social network applications, *J. Inf. Secur. Appl.*, **65** (2022), 103115. https://doi.org/10.1016/j.jisa.2022.103115
19. O. Thorat, N. Parekh, R. Mangrulkar, TaxoDaCmachine learning: Taxonomy based Divide and Conquer using machine learning approach for DDoS attack classification, *Int. J. Inf. Manage. Data Insights*, **1** (2021), 100048. https://doi.org/10.1016/j.jjimei.2021.100048

20. M. E. Ahmed, H. Kim, M. Park, Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking, in *IEEE Military Communications Conference (MILCOM)*, (2017), 11–16. https://doi.org/10.1109/MILCOM.2017.8170802

21. I. Sreeram, V. P. K. Vuppala, HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm, *Appl. Comput. Inf.*, **15** (2019), 59–66. https://doi.org/10.1016/j.aci.2017.10.003

22. O. Salman, I. H. Elhajj, A. Chehab, A. Kayssi, A machine learning based framework for IoT device identification and abnormal traffic detection, *Trans. Emerging Telecommun. Technol.*, **33** (2022). https://doi.org/10.1002/ett.3743

23. X. Liu, L. Zheng, S. Helal, W. Zhang, C. Jia, J. Zhou, A broad learning-based comprehensive defence against SSDP reflection attacks in IoTs, *Digital Commun. Networks*, **2022** (2022). https://doi.org/10.1016/j.dcan.2022.02.008

24. S. Ismail, Z. El Mrabet, H. Reza, An ensemble-based machine learning approach for cyber-attacks detection in wireless sensor networks, *Appl. Sci.*, **13** (2022), 30. https://doi.org/10.3390/app13010030

25. D. Kshirsagar, S. Kumar, A feature reduction based reflected and exploited DDoS attacks detection system, *J. Ambient Intell. Hum. Comput.*, **13** (2022), 393–405. https://doi.org/10.1007/s12652-021-02907-5

26. A. Mishra, N. Gupta, B. B. Gupta, Defensive mechanism against DDoS attack based on feature selection and multi-classifier algorithms, *Telecommun. Syst.*, **82** (2023), 229–244. https://doi.org/10.1007/s11235-022-00981-4

27. I. Sharafaldin, A. H. Lashkari, S. Hakak, A. A. Ghorbani, Developing realistic Distributed Denial of Service (DDoS) attack dataset and taxonomy, in *International Carnahan Conference on Security Technology (ICCST)*, (2019), 1–8. https://doi.org/10.1109/CCST.2019.8888419

28. A. Prasad, S. Chandra, Defending ARP spoofing-based MitM attack using machine learning and device profiling, in *2019 International Carnahan Conference on Security Technology (ICCST)*, (2022), 978–982. https://doi.org/10.1109/ICCCIS56430.2022.10037723

29. D. Tang, L. Tang, R. Dai, J. Chen, X. Li, J. Rodrigues, MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost, *Future Gener. Comput. Syst.*, **106** (2020), 347–359. https://doi.org/10.1016/j.future.2019.12.034

30. B. Sabir, M. A. Babar, R. Gaire, A. Abuadbba, Reliability and robustness analysis of machine learning based phishing URL detectors, *arXiv preprint*, (2022), arXiv:2005.08454. https://doi.org/10.48550/arXiv.2005.08454

31. S. A. Khanday, H. Fatima, N. Rakesh, Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks, *Expert Syst. Appl.*, **215** (2023), 119330. https://doi.org/10.1016/j.eswa.2022.119330

32. M. M. Alani, E. Damiani, XRecon: An explainbale IoT reconnaissance attack detection system based on ensemble learning, *Sensors*, **23** (2023), 5298. https://doi.org/10.3390/s23115298

33. R. Verma, S. Chandra, RepuTE: A soft voting ensemble learning framework for reputation-based attack detection in fog-IoT milieu, *Eng. Appl. Artif. Intell.*, **118** (2023), 105670. https://doi.org/10.1016/j.engappai.2022.105670
34. S. Pokhrel, R. Abbas, B. Aryal, IoT security: botnet detection in IoT using machine learning, *arXiv preprint*, (2021), arXiv:2104.02231. https://doi.org/10.48550/arXiv.2104.02231
35. A. P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, *Pattern Recognit.*, **30** (1997), 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2
36. D. Chicco, G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, *BMC Genomics*, **21** (2020), 6. https://doi.org/10.1186/s12864-019-6413-7
37. Md. M. Rashid, J. Kamruzzaman, M. Ahmed, N. Islam, S. Wibowo, S. Gordon, Performance enhancement of intrusion detection system using bagging ensemble technique with feature selection, in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, (2020), 1–5. https://doi.org/10.1109/CSDE50874.2020.9411608
38. Md. Raihan-Al-Masud, H. A. Mustafa, Network intrusion detection system using voting ensemble machine learning, in *2019 IEEE International Conference on Telecommunications and Photonics (ICTP)*, (2019), 1–4. https://doi.org/10.1109/ICTP48844.2019.9041736
39. S. V. J. Rani, I. Ioannou, P. Nagaradjane, C. Christophorou, V. Vassiliou, S. Charan, et al., Detection of DDoS attacks in D2D communications using machine learning approach, *Comput. Commun.*, **198** (2023), 32–51. https://doi.org/10.1016/j.comcom.2022.11.013
40. S. ur Rehman, M. Khaliq, S. I. Imtiaz, A. Rasool, M. Shafiq, A. R. Javed, et al., DIDDOS: An approach for detection and identification of Distributed Denial of Service (DDoS) cyberattacks using Gated Recurrent Units (GRU), *Future Gener. Comput. Syst.*, **118** (2021), 453–466. https://doi.org/10.1016/j.future.2021.01.022
41. R. J. Alzahrani, A. Alzahrani, Security analysis of DDoS attacks using machine learning algorithms in networks traffic, *Electronics*, **10** (2021), 2919. https://doi.org/10.3390/electronics10232919
42. S. Sindian, S. Sindian, An enhanced deep autoencoder-based approach for DDoS attack detection, *Wseas Trans. Syst. Control*, **15** (2020), 716–724. https://doi.org/10.37394/23203.2020.15.72
43. 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. Netw.*, 2021 (2021), 1–14. https://doi.org/10.1155/2021/5710028
44. D. Javeed, T. Gao, M. T. Khan, SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT, *Electronics*, **10** (2021), 918. https://doi.org/10.3390/electronics10080918

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)