electronics

Article

Denial of Service Attack Classification Using Machine Learning with Multi-Features

Furqan Rustam [1], Muhammad Faheem Mushtaq [2], Ameer Hamza [3], Muhammad Shoaib Farooq [4], Anca Delia Jurcut [1,*] and Imran Ashraf [5,*]

1 School of Computer Science, University College Dublin, D04 V1W8 Dublin, Ireland
2 Department of Computer Science, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
3 Department of Information Security, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan
4 Department of Computer Science, University of Management and Technology, Lahore 54000, Pakistan
5 Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea
* Correspondence: anca.jurcut@ucd.ie (A.D.J.); imranashraf@ynu.ac.kr (I.A.)

Abstract: The exploitation of internet networks through denial of services (DoS) attacks has experienced a continuous surge over the past few years. Despite the development of advanced intrusion detection and protection systems, network security remains a challenging problem and necessitates the development of efficient and effective defense mechanisms to detect these threats. This research proposes a machine learning-based framework to detect distributed DOS (DDoS)/DoS attacks. For this purpose, a large dataset containing the network traffic of the application layer is utilized. A novel multi-feature approach is proposed where the principal component analysis (PCA) features and singular value decomposition (SVD) features are combined to obtain higher performance. The validation of the multi-feature approach is determined by extensive experiments using several machine learning models. The performance of machine learning models is evaluated for each class of attack and results are discussed regarding the accuracy, recall, and F1 score, etc., in the context of recent state-of-the-art approaches. Experimental results confirm that using multi-feature increases the performance and RF obtains a 100% accuracy.

Keywords: cyber threats; denial of service attack; feature engineering; machine learning; network security

Citation: Rustam, F.; Mushtaq, M.F.; Hamza, A.; Farooq, M.S.; Jurcut, A.D.; Ashraf, I. Denial of Service Attack Classification Using Machine Learning with Multi-Features. *Electronics* **2022**, *11*, 3817. https://doi.org/10.3390/electronics11223817

Academic Editor: Yue Wu

Received: 26 October 2022
Accepted: 16 November 2022
Published: 20 November 2022

1. Introduction

Denial of service (DoS) attacks have become one of the most obstinate issues for many years regarding network security. Despite numerous detective and preventive schemes being contrived, the threat of distributed DoS (DDoS)/DoS attacks are still persistent, and their numbers increase every year [1]. DDoS attacks are real threats in the future, as they are increasing day by day. According to the report by [2], in the third quarter of 2022, a DDoS intelligence system detected 57,116 attacks. Most of the attacks were launched in the U.S. In the second quarter of 2022, it was 45.95%, and this ratio is 39.60% for the third quarter of 2022. Worldwide, this ratio has increased from 38.69% in Q3 2021 to 53.53% in Q3 2022. According to [3], during the first six months of 2022, malicious DDoS attacks increased by 60% as compared to the same period in 2021. These statistics show the alarming satiation for smart systems and show the need for improvements in the security systems for smart devices. In recent years, several big IT companies faced DDoS, and one of the biggest attacks was launched on AWS in February 2020 [4]. The attack was launched with huge traffic of approximately 2.3 terabits per second (Tbps). Similarly, GitHub was also targeted by the DDoS attack in 2018.

In today's world, the internet plays a vital role in several aspects of human life, such as education, transport, banking [5], hospitals, entertainment, personal use, administrations,

trade, communication [6–8], e-commerce [9], environment [10], and many more [11,12]. It has made human life easier by revolutionizing the world of communication and technology. Along with the advancement of technology, many security-related risks also emerge and rise. It is also the case with internet security where data damage, theft of resources, breaching confidentiality, social harassment, and online frauds, etc., have substantially increased [13–16].

Data availability is one of the significant challenges related to network security. DDoS is one of the leading attacks that compromises the availability of a network. It accomplishes it by redundantly requesting the targeted resources to strain services provided by the system. In a DDoS attack, requests come from multiple systems to a single target, as shown in Figure 1. Through DDoS attacks, malicious users can completely distort the availability of systems/services to authentic users. Currently, DDoS attacks are continuously growing and put network security at risk. Traditionally, DDoS attacks use a collection of compromised systems known as botnets. The major objective of this attack is to demolish the server resources such as memory, bandwidth, processor, etc., to put down the services for legitimate users. A comprehensive review of recent DDoS attacks is provided in [17], while many mitigation techniques are discussed in [18–20]. DoS attacks are classified concerning several aspects. Based on network protocol, a DoS attack is further fragmented in the transport layer and application layer of DoS attacks.

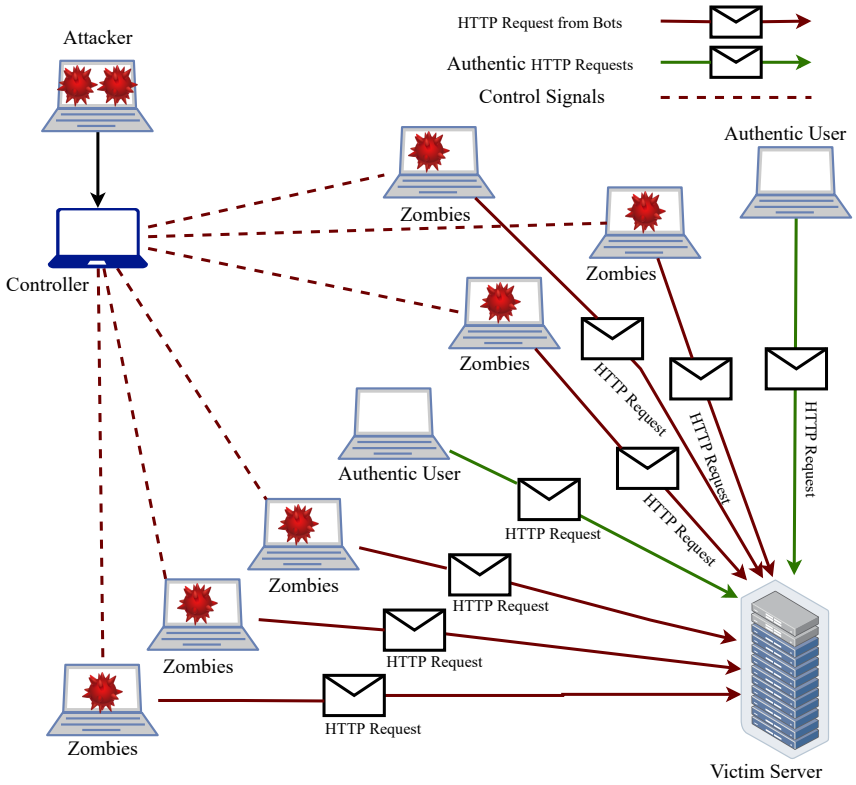

Figure 1. Schematic diagram of a DDoS attack.

This study mainly focuses on application layer DoS/DDoS attacks. In network layer DDoS attacks, attackers take advantage of partially opened connections by using transmission control protocol/user datagram protocol (TCP/UDP) protocols and send numerous forged payloads normally using internet protocol (IP) spoofing. In contrast, attackers send several requests to exhaust well-known applications such as hypertext transfer protocol (HTTP), domain name system (DNS), etc., in the application layer DDoS attack. These requests are indistinguishable from authorized user requests at the network level. These attacks have a higher rate of bogus requests as compared to legitimate requests. These

attacks are hard to detect because the connections are already established and requests seem to originate from authorized users.

This study proposes a system to detect DDoS attacks using machine learning techniques. It is very difficult to monitor network traffic manually to protect the system from attackers, so a smart protection system to detect the attacks is needed. The proposed system is significantly better than existing approaches, as it is a simple yet effective approach to achieve better results. The contributions of this research are as follows:

- An efficient and effective DDoS/DoS attack detection framework is proposed, which can perform early detection of several attacks, such as HTTP flood and stealth DDoS/DoS attacks. For experiments, the 'Application layer DDoS attack dataset' is used. A higher attack detection performance is aimed at reduced computational complexity.
- A novel feature selection approach has been developed to select highly significant features by incorporating selective features from principal component analysis (PCA) and singular value decomposition (SVD) for this purpose.
- Experiments are performed with many machine learning models such as logistic regression (LR), gradient boosting machine (GBM), random forest (RF), and extra tree classifier (ETC). Performance appraisal with state-of-the-art approaches is carried out in terms of accuracy, precision, recall, and F1 score.

This study is organized into five sections. Section 2 presents the related work. Section 3 explains the material and methods, including the dataset, machine learning methods, and the proposed methodology. Results are discussed in Section 4, while Section 5 concludes the study.

2. Related Work

Several network attack detection and protection techniques have been introduced in recent years. An intrusion detection system (IDS) can be classified into signature-based, anomaly-based, and hybrid structures [21]. The first type detects anomalies by comparing the event with a built database containing signatures. The second approach observes attacks using deviations between the existing database containing the normal state and the current state. In all instances, an alert can be triggered if a matching similarity is found or a deviation is detected. Signature-based IDS are known for a low ratio of false alarms; however, it is very challenging to gather and keep possible variations of attacks [22]. However, the challenge is to write signatures for all possible attack variations. Anomaly-based detection techniques also have the potential to identify other types of attacks, but it requires more evaluation resources. Hybrid techniques try to achieve the benefits of both approaches by combining them [23,24]. Flooding attack has drawn much attention and has been highlighted in recent surveys on cloud computing, wireless network, big data, etc. [25–27].

Recently, several classification approaches have been proposed for DDoS attacks. On the protocol level, DDoS attacks can be classified into two categories, network-level and application-level DDoS flooding attacks [28]. One of the major challenges in DDoS attacks is early detection and impact mitigation. However, it requires several additional features, which are not present in the current approaches [29]. The HTTP-based approach is presented in [30] for the detection of HTTP flooding attacks through data sampling. The study is based on the CUMSUM algorithm to identify the traffic as malicious or normal. Traffic analysis is performed by using the number of requests made by the application layer and the zero-sized number of packets. Results indicate that the approach achieves a detection rate between 80% to 86% with a 20% sampling rate.

A detection system for DDoS attacks is presented in [31] that uses the D-FACE algorithm. This mechanism uses generalized information distance (GID) matrices and generalized entropy (GE) for the detection of various DDoS attacks. The proposed method requires the high involvement of internet service providers (ISP), which restricts its industrial use. The Sky-Shield system has been developed in [32] to avoid DDoS attacks at the application layer. For the detection of irregularities in traffic, it considers two hash sketches to find

divergence. In the mitigation phase, blacklisting, whitelisting, and the user filtering process are taken out for defensive mechanisms. The results are evaluated through customized datasets. Sky-Shield mainly focuses on the detection of flooding attacks at the application layer, especially HTTP protocol, so it is vulnerable to transport and network-level flooding attacks. Another probable method for DDoS flooding attack detection is using a semi-supervised based approach as in [33] to detect and protect from DDoS flooding attacks. Two different clustering algorithms are used, while the final label is managed using voting. The CICIDS 2017 data set is used for the evaluation process.

The study [34] deployed machine learning and deep learning models for DDoS attack detection. Experiments are performed using different deep learning and machine learning models such as gated recurrent unit, RF, and LR using CICDoS2017 and CICDDoS2019 datasets. Models achieved a 0.99 accuracy score on unseen test data and also perform better in the simulation network. Another study [35] performed experiments for malicious traffic detection using ensemble classifier extra boosting forest (EBF) with PCA feature selection technique. The study conducted experiments on UNSW-NB15 and IoTID20 datasets individually, as well as both combined, and achieved 0.985 and 0.984 accuracy scores, respectively. The authors investigated the performance of three supervised machine learning algorithms including KNN, RF, and NB for the DDoS attack detection in [36]. NB outperforms all used models with a significant 0.985 accuracy score.

Along the same lines, the study [37] proposed an approach to detect application-layer DDoS attacks in real-time using machine learning techniques. They deployed state-of-the-art multilayer perceptron (MLP) and RF with and without a big data approach. For both approaches, RF achieved an accuracy score of 0.999, while MLP achieved 0.990 without big data and 0.993 with the big data approach. The study [38] performed experiments for DDoS attack prediction using supervised machine learning models. For this purpose, the performance of several models is analyzed such as LR, Bayesian naive Bayes (BNB), random tree (RT), KNN, and REPTree algorithm, etc. KNN performs significantly better with a 0.998 accuracy score. Similarly, [39] investigated DDoS traffic attack detection using a hybrid SVM-RF model. They found novel features from the dataset to train the proposed model and achieved a 0.988 accuracy. An analytical summary of the discussed research works is provided in Table 1. In the literature, high-accuracy results are reported from studies that deploy complex deep learning models, and the computational cost is high. Contrarily, if simple methods are used, their computational cost is low, but such approaches show poor performance regarding attack detection accuracy. We aim to overcome this limitation by developing an approach that is significant in terms of both accuracy and computational cost.

Table 1. Summary of related work.

Ref	Year	Dataset	Models	Aim	Limitations
[34]	2021	CICDoS2017 and CICD115 DoS2019	GRU, LSTM, MLP, KNN, RF	DDoS attack detection from SDN-Based Architecture.	The proposed system for transport layer and application have different results for both, as attack detection rate at the application layer is 95%. Second, they used complex GRU models, which have high computational costs.
[35]	2021	UNSW-NB15 and IoTID20	RF, GBM, ETC	Malicious traffic detection using ensemble learning approach.	They built a stack of models, which required a high computational cost.

Table 1. *Cont.*

Ref	Year	Dataset	Models	Aim	Limitations
[36]	2020	Collect their dataset using Wireshark	KNN, RF, and NB	Detection of different types of DDoS such as ICMP flood, TCP flood, UDP flood, etc.	Proposed approach used simple models where their computational cost is lower, but their accuracy is low as compared to other approaches.
[37]	2021	Application-Layer DDoS Dataset available on public platform Kaggle	RF and MLP	Real-Time DDoS attacks detection in real-time big data.	They work on both efficiency and accuracy, but they worked with only two models, so we could not have a lot of results to obtain the significance of the approach.
[38]	2021	NB-ISCX, CTU-13, and ISOT datasets	REPTree a, J48, NB, LR	DDoS detection in SDN using machine learning and statistical methods.	This study worked with simple models, which make them efficient in terms of computational cost, but for this, they compromise on accuracy.
[39]	2021	SVM and RF, Ensemble SVM-RF	Collect their dataset for SDN networks DDoS with the help of RYU API	DDoS attack detection in SDN networks using ensemble learning.	The ensemble learning approach requires higher computational costs. The SVC model itself in the ensemble has a high computational cost.
[40]	2022	RF, SVM, DT, MLP, GRU, LSTM	Bot-IoT Dataset	DDoS attack in IoT device application layers and transport layers.	In this study, they have an imbalanced distribution of targets in binary classification.
[1]	2022	KNN, SVM, RF	Open-source dataset related to DDoS	Banking sector IoT devices DDoS attack detection.	Their proposed approach needs an improvement in accuracy.

3. Study Methodology

This section discusses the dataset, the proposed method, and techniques for DDoS attack classification.

3.1. Dataset Description

This research used the application layer DoS dataset, which was acquired from Kaggle, a familiar source for benchmark datasets [41]. The dataset contains 809,361 records, with 78 attributes about DoS attacks for the application layer. By performing network analysis, the records are classified into three classes: 'benign', which is legitimate; 'DoS slowloris', which is a DoS attack; and 'DoS hulk', which is a DDoS attack. Table 2 shows a few attributes from the application layer DoS dataset.

Table 3 shows the ratio of each class instance, which shows that the 'benign' has the highest number of records, i.e., 370,623, 'DoS slowloris' has 128,612 examples, and 'DoS hulk' contains 310,126 records. It indicates that the dataset has disparity regarding the number of samples for different classes. To solve this problem, we obtained 25,000 records from each class to make the dataset balanced. Table 4 shows sample records from the dataset.

Table 2. Description of a few attributes of the dataset.

Variable	Description
Destination Port	Destination port Number
Flow Duration	Duration of total flow
Total Forward Packet	Total Number of forwarding packet
Total Backward Packet	Total Number of Backward Packet
Length of forward Packet	Total Length of forward Packet
Flow-IAT-Max	Maximum Inter Arrival Time
Flow-IAT-Min	Minimum Inter Arrival Time
Forward IAT-Total	Total Inter Arrival Time
Flow-Packet-sec	Number of packets per second
Flow-Byte-Sec	Number of bytes per second

Table 3. Number of samples corresponding to each class.

Class Label	Total Records	Used for Experiments
BENIGN	310,126	25,000
DOS SLOWLORIS	128,612	25,000
DOS HULK	370,623	25,000

Table 4. Sample from Dataset.

Destination Port	Flow Duration	Total Forward Packet	Total Backward Packet	Length of Forward Packet	Flow-IAT-Max	Forward IAT-Total	Flow-Packet-Sec	Label
80	101,168,794	20	1	969	79,000,000.0	101,000,000.0	0.207574	DOS HULK
60711	58	1	1	0	58.0	0.0	176,182.000	BENIGN
80	68,156,881	10	1	233	26,048,320.0	68,155,459.0	0.161392	DOS SLOWLORIS
80	173,608	7	1	344	137,216.0	169,859.0	46.080826	DOS HULK
443	11,932,077	12	16	5030	11,049,290.0	882,787.0	133,390.000	BENIGN
80	71,823,654	6	2	108	25,144,732	712,895	22.3285	DOS SLOWLORIS
80	153,708	8	2	444	127,616.0	189,859.0	36.080826	DOS HULK
53	744	2	2	88	200	48.0	204,389	BENIGN
53	31,146	4	2	148	30,200	30,204	127,126	BENIGN
80	61,923,754	7	1	2108	37,133,731	61,922,996	0.1291	DOS SLOWLORIS

3.2. Feature Engineering Methods

Feature engineering is an important part of the machine learning domain that aims to find the important features from the dataset to train the machine learning models [42]. The selection of important features can enhance the performance of machine learning models [43]. This research uses feature selection and multi-features to boost the learning

models' performance. In feature selection, PCA and SVD are used to select the best feature from the dataset individually.

3.2.1. Principle Component Analysis

The PCA is a feature selection technique commonly used to select important features from the data. PCA is a very effective approach, easy to understand, and has no constraints regarding parameter selection. The main idea of PCA is to recognize the correlation in the data. It maps X-dimensional features with Y-dimensional features ($y \leq x$) to maximize certain variance. The new Y-dimensional features are orthogonal (uncorrelated) and ranked to the explained variance [44,45]. PCA reduces the redundancy of data by constructing new smaller features that have important information about the original data.

- **Step 1:** should be retained. Compute the simple mean of X-dimensional data features.

$$M = \frac{1}{n} \sum_{a=1}^{n} k_i \tag{1}$$

where k_i is a data feature and n is the total number of variables.

- **Step 2:** Enumerate the covariance matrix using the mean value.

$$C = \frac{1}{n} \sum_{a=1}^{n} (k_i - M)((k_i - M)^T) \tag{2}$$

- **Step 3:** Evaluate the feature vector and values of the covariance matrix.

$$C = F. \sum .F^T \tag{3}$$

$$\sum = diag((\mu_1, \mu_2, \ldots, \mu_n)\mu_1 \geq \mu_2 \geq \ldots \geq \mu_n \geq 0) \tag{4}$$

where F is a feature matrix having feature vector values f_i and $\sum$ arranged diagonal matrix of n feature values in descending value, and μ_i is the corresponding feature value of the covariance matrix.

- **Step 4:** Evaluate the cumulative variance contribution rate for first y-row elements by using obtained feature values and feature vector

$$\xi = \frac{\sum_{a=1}^{Y} \mu_a}{\sum_{r=1}^{n} \mu_r} \tag{5}$$

where ξ is the cumulative variance contribution rate of first y-row principle elements and ($\xi \geq 0.9$).

- **Step 5:** Perform dimension reduction based on obtained Y-row feature vector.

$$S = F_y.X \tag{6}$$

F_y is the feature matrix and acquires first y-row feature values ($y \geq x$), and S is Y-dimensional data, which is obtained after mapping with x-dimensional original data. The dimensional reduction is achieved by a linear transformation of data X to S.

3.2.2. Singular Value Decomposition

The SVD is a well-known feature selection technique that is used for dimension reduction [46,47]. SVD decomposes a matrix and exposes numerous useful and interesting properties of the original matrix [48]. The SVD of a matrix M is a decomposition of M into three matrices $M = U \sum V^T$, where U and V are orthogonal matrices and have eigenvectors columns of $M^T M$ and MM^T, respectively. The matrix $\sum$ is a diagonal matrix of positive singular entities of matrix M.

3.2.3. Multi-Features

The multi-features approach considers selected features from both PCA and SVD to make a more significant feature set. A total of 60 features are selected, the top 30 features each from PCA and SVD, for this purpose. Multi-features are defined as

$$pca_{30} = PCA(Dataset) \tag{7}$$

$$svd_{30} = SVD(Dataset) \tag{8}$$

and

$$multifeature_{60} = pca_{30} \bigcup svd_{30} \tag{9}$$

where $multifeature_{60}$ shows the multi-features obtained from *PCA* and *SVD* results.

The multi-features approach aims at improving the performance of machine learning models by selecting only those features that highly contribute to the prediction. Selecting features from two different approaches helps to optimize models' performance and improve the computational complexity as well.

3.3. Supervised Models

With the wide use of machine learning models, a large number of variations can be found in the existing literature that can be used to obtain good performance for classification. In addition, Sci-Kit [49] provides easy-to-use functions to implement such models. This research uses several machine learning classifiers to classify malicious and normal traffic, including RF, LR, GBM, and ETC. A brief description of the used models is provided here for completeness.

3.3.1. Random Forest

RF is a tree-based supervised learning model commonly used for regression and classification problems [50]. RF achieves better performance by using several decision trees. It uses the bagging technique for the training process and reduces the variance of those models that have high variance [51]. RF constitutes several bootstrap samples for each weak learner, which are used for computing the fitting process. Bootstrapping is performed by replacing several random samples from the training dataset. The RF model is very effective for servers with higher traffic because it randomly selects the data tuples [52]. By considering packet length, inter-arrival time, and average bulk rate, which are used to detect different types of network attacks such as DoS and DDoS, this reduces the impact on network bandwidth [53]. In the case of a DDoS attack, the packet size is the same if the total number of packets arrived is continuously increasing in a particular time with a very high rate—then, the RF approach can identify a DDoS attack more accurately [54]. In a DDoS attack, the attackers send a larger number of data packets to victim networks, whereas packets are increasing as compared to the normal cases. The RF algorithm can be expressed through the following equations:

$$L = mode T_1(x), T_2(x), ..., T_n(x) \tag{10}$$

$$L = mode\{\sum_{y=1}^{y} T_y(x)\} \tag{11}$$

where L is the final prediction and $T_1(x), T_2(x), ..., T_n(x)$ are decision trees participating in the prediction process.

This study implemented RF with several hyper-parameters. The 'n_estimator' has a value of 300, which indicates 300 weak learners to predict the final results. Another parameter is 'max_'epth with a value of 100, which constrains the maximum depth of each tree to 100. The third parameter is 'random_states', which is used with a value of 5, which defines the randomness of samples during the training phase.

3.3.2. Logistic Regression

LR is a statistical method commonly used to resolve classification problems [55]. To select LR parameters, the maximum likelihood estimation (MLE) framework is used [56]. The MLE framework is the probabilistic framework that is commonly used to resolve density estimation problems [57]. LR defines the relationship between the dependent and independent variables of a dataset. The logistic function is used to estimate the probability to find a correlation between dependent and one or more independent variables [58]. The logistic (sigmoid) function is a mathematical function that takes real values and generates an S-shaped logistic curve to map values between 0 and 1 [59]. The sigmoid function can be expressed as

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{12}$$

where $\sigma(z)$ is a probability estimate (output bounded 0 or 1), z is the algorithm prediction, e.g., $(mx + b)$, and e common log base is also called the Euler number.

The range of real numbers is $+\infty$ to $-\infty$, so the logistic function maps a curve that either goes to $+\infty$ or $-\infty$. If it goes to negative infinity, then the value becomes 0 otherwise 1. This study uses the 'liblinear' approach in the experimental process because it gives better outcomes on small datasets, whereas 'saga' and 'sag' perform well for big datasets. The second parameter used is 'multi_class' with an 'ovr' value because it gives optimal results for binary classification. The C is the third parameter that reduces the chance of overfitting the model.

3.3.3. Gradient Boosting Machine

GBM proves to be an impressive approach for constructing predictive models. It is a composite approach that makes a strong predictive model by combing several weak learning models to reduce the mean square error (MSE) [60]. Bagging algorithms only deal with high variance, whereas boosting algorithms consider variance and bias in both aspects and are considered potent in dealing with variance and bias trade-offs [61]. The gradient boosting algorithm recognizes the infirmities of weak learners by using a gradient in the loss function. MSE is the average value of the difference between the predicted value and actual targets from a set of considerations, such as the validation set. The loss function shows how model coefficients are capable of fitting the underlying data. The gradient boosting algorithm can be expressed as (13).

$$Loss = \sum (y_i - y_i^p)^2 \tag{13}$$

where y_i is the ith targeted value, y_i^p is the ith predicted value, and $f(y_i, y_i^p)$ is the loss function.

This study implements GBC with a 'max_depth' of 100 and a learning rate of 0.7, indicating that the maximum depth of each tree is 100. Another used parameter 'n_estimator' is set to 300, which shows that GBM combines 300 weak learners to make the final prediction. Similarly, 'random_states' is used with a value of 52, which defines the randomness of samples during the training phase.

3.3.4. Extra Tree Classifier

ETC is an ensemble learning approach that combines the results of multiple decision trees to obtain the classification results [62]. The difference between ETC and RF is the manner of constructing the decision trees in the forest. ETC formulates decision trees from the original training samples, and each node is provided with random samples of K features [63], whereas the selection of features is performed through random split (typically Gini index) and without replacement. This study implements ETC with different hyperparameters. The parameter 'n_estimator' is used with a value of 300, indicating the number of decision trees to predict the final results. The 'random_states' parameter is set

to 5, which defines the randomness of samples during the training phase. A complete list of the hyperparameters is given in Table 5.

Table 5. Hyperparameters for the models.

Model	Hyperparameters
RF	n_estimator = 300, max_depth = 100, random_state = 5
LR	Solver = libliner, multi_class= Ovr, C = 3.0
GBM	max_depth = 100, n_estimator= 300, random_states = 52
ETC	n_estimator = 300, random_states = 5

3.4. Proposed Methodology

We used an Intel Core-i7 11th generation system to implement the proposed method. The system contains 16 GB RAM and 1 TB SSD with the Windows operating system. We used Sci-Kit learn and TensorFlow libraries to implement machine learning and deep learning methods using Python language and a Jupyter notebook.

This study proposes a machine learning-based framework for DoS/DDoS attack detection. The proposed method for DDoS attack detection is illustrated in Figure 2. A machine learning-based early detection system is designed to combat DDoS/DoS attacks at the application layer effectively and collaboratively. In this approach, the network traffic is obtained from the benchmark dataset for testing and training purposes. In the first step of experimentation, data are preprocessed by removing null values and converting real values to integer values. Multiple feature selection techniques are applied to extract the significant features to train the model. After selecting the significant features from PCA and SVD, the multi-feature technique is applied to obtain the top 60 features extracted by applying both feature selection techniques.

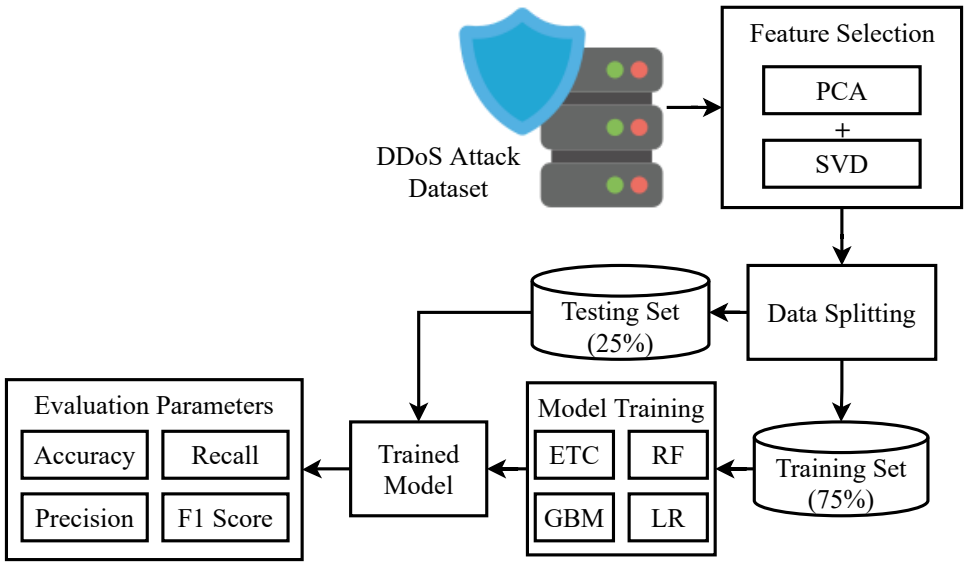

Figure 2. Flow diagram of the proposed methodology.

After the initial preprocessing, the dataset is split into 75% to 25% for training and testing. We used 75% of the total dataset for the training of models and 25% for the testing of models. The performance of the models is evaluated in terms of accuracy, precision, recall, and F1 score. Accuracy determines the correctness of the model in terms of correctly classifying attacks. The accuracy score can be computed as the total number of correct predictions divided by the total number of predictions. Similarly, other evaluation

parameters such as precision, recall, and F1 score can also be calculated using numbers of correct and wrong predictions [55,64].

4. Results and Discussion

This section discusses the results of the experiments for DoS/DDoS attack detection. For this purpose, multiple experiments are performed using the machine learning models with a dataset comprising application layer DoS/DDoS attacks.

4.1. Results without Feature Selection

In the first set of experiments, machine learning models were applied without any feature selection technique to classify normal and malicious traffic. A total of 78 attributes were used for experiments, and results are given in Table 6.

Table 6. Performance of machine learning models using all attributes.

Classifier	25,000 Samples	50,000 Samples
RF	0.993	1.000
LR	0.944	0.949
GBM	0.996	0.998
ETC	0.998	0.999

Results show that tree base ensemble models perform better as compared to linear models because the linear models only perform well when data is linearly separable. Increasing the training samples also helps to increase the overall accuracy of models. Table 6 shows that better accuracy is achieved by training the model using 50,000 data samples as compared to 25,000 data samples. By increasing training, data samples' accuracy score will become refined. Table 7 shows a performance comparison of 50,000 training samples for individual classes of attacks. RF achieves the highest precision, recall, and F1 score for all classes, while GBM and ETC perform slightly poorly as shown in Figure 3. LR shows the worst performance, especially when DDoS attacks are considered.

Table 7. Results of machine learning models for individual classes.

Classifier	Class	Precision	Recall	F1 Score
RF	Benign	1.00	1.00	1.00
	DoS Slowloris	1.00	1.00	1.00
	DoS Hulk	1.00	1.00	1.00
LR	Benign	0.95	0.99	0.97
	DoS Slowloris	0.92	0.92	0.92
	DoS Hulk	0.97	0.92	0.94
GBM	Benign	1.00	1.00	1.00
	DoS Slowloris	0.99	0.99	0.99
	DoS Hulk	0.99	0.99	0.99
ETC	Benign	1.00	1.00	1.00
	DoS Slowloris	1.00	0.99	0.99
	DoS Hulk	0.99	1.00	1.00

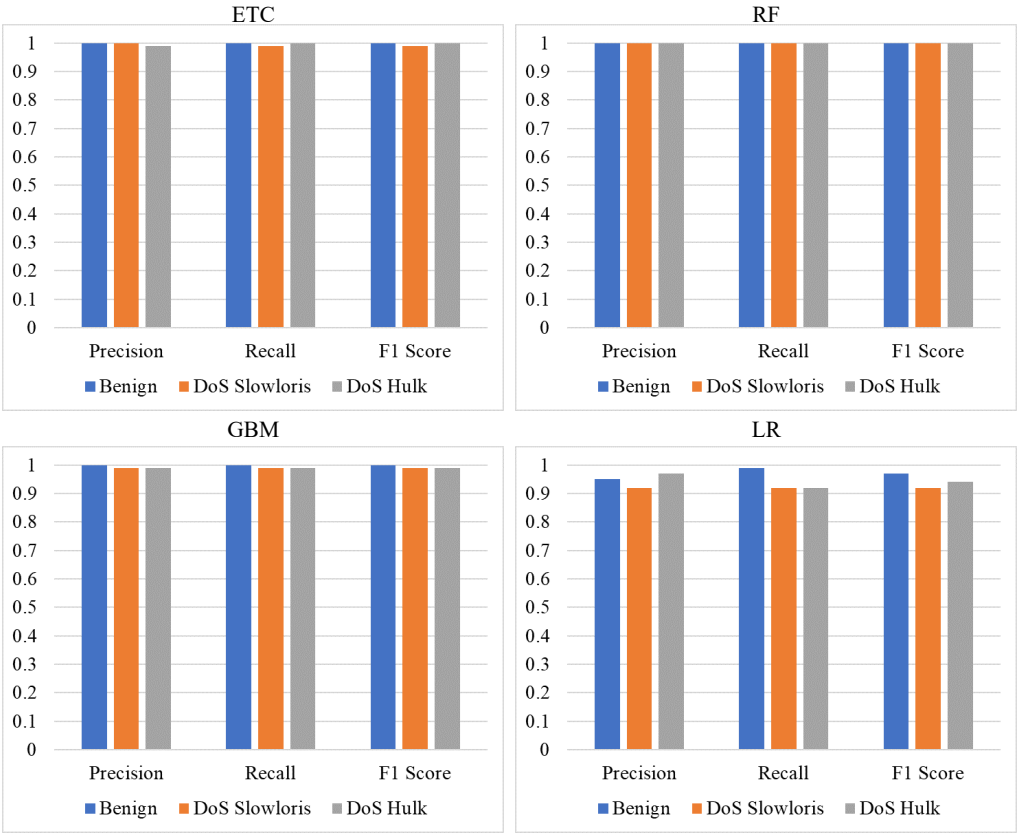

Figure 3. Performance comparison using all attributes.

4.2. Performance Using Multi-Features

In the second set of experiments, feature selection techniques such as PCA and SVD were used to select important features from data samples. By applying feature selection techniques, learning models improved their results. These techniques also improve the performance of all learning models, as shown in Tables 8 and 9. The use of multi-features from PCA and SVD boosted the accuracy score of tree-based ensemble models. The results of LR also improved.

Table 8. Models' performance using multi-features from principal component analysis (PCA) and singular vector decomposition (SVD).

Classifier	Features	25 k Samples	50 k Samples
RF	Multi-features	0.998	1.000
LR	Multi-features	0.950	0.960
GBM	Multi-features	0.999	1.000
ETC	Multi-features	1.000	1.000

Table 8 shows that the accuracy with all features is lower than PCA + SVD because PCA and SVD, when used together, provide prominent features, which help the model's training. It also shows that feature engineering techniques are valuable for machine learning and support results optimization. Experimental results show that the ensemble models achieve the highest accuracy because ensemble models use several decision trees and often perform better than a single decision tree in the classification process. As mentioned above, this study has multiple target classes, including 'benign', 'DDoS', and 'DoS', and individual class accuracy varies, as shown in Table 9. LR shows the worst accuracy regarding DDoS attack detection, with an accuracy score of 0.92, which is the lowest among all models.

Table 9. Classification report of each learning model with (PCA + SVD).

Classifier	Class	Precision	Recall	F1 Score
RF	Benign	1.00	1.00	1.00
	DoS Slowloris	1.00	1.00	1.00
	DoS Hulk	1.00	1.00	1.00
LR	Benign	0.99	0.99	0.99
	DoS Slowloris	0.92	0.97	0.94
	DoS Hulk	0.97	0.92	0.94
GBM	Benign	1.00	1.00	1.00
	DoS Slowloris	1.00	1.00	1.00
	DoS Hulk	1.00	1.00	1.00
ETC	Benign	1.00	1.00	1.00
	DoS Slowloris	1.00	1.00	1.00
	DoS Hulk	1.00	1.00	1.00

Figure 4 shows that all machine learning models achieve the best results when trained using multi-features from PCA and SVD. PCA and SVD provide appropriate features that play a significant role to optimize the performance of machine learning models. As compared to other models, RF persistently shows better performance for all classes of network attacks.

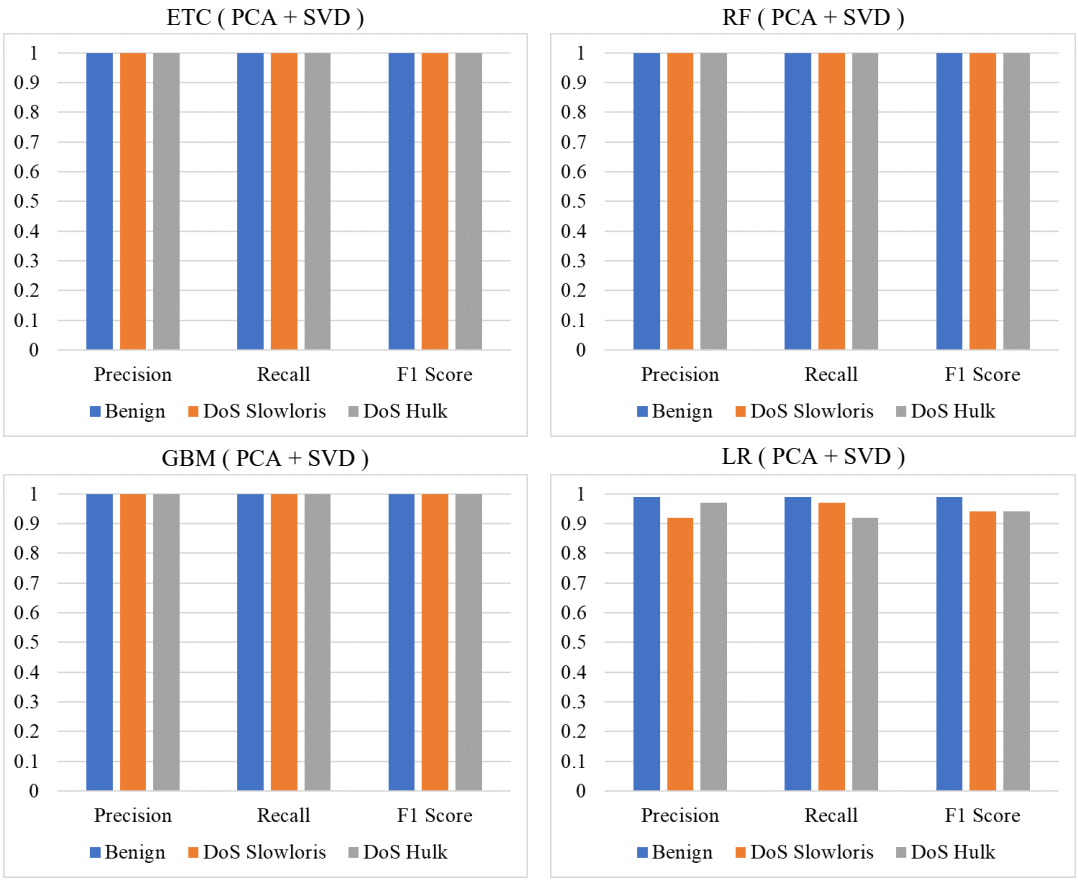

Figure 4. Comparison of performance when PCA and SVD are used.

4.3. Performance Comparison Using Different Features

Besides the experiments without any feature selection and multi-features, experiments were performed using all the features and individual features from PCA and SVD to compare the performance with the proposed multi-features. Table 10 shows the importance of multi-features for attack detection, as it shows the best performance among all the feature methods used for experiments.

Table 10. Accuracy report of each learning model using different features.

Model	PCA		SVD		All Features		Multi-Features	
	25 k	50 k	25 k	50 k	25 k	50 k	25 k	50 k
RF	0.991	0.993	0.940	0.948	0.993	1.000	0.998	1.000
LR	0.950	0.951	0.464	0.468	0.944	0.949	0.950	0.960
GBM	0.993	0.990	0.938	0.944	0.996	0.998	0.999	1.000
ETC	0.996	0.996	0.948	0.949	0.998	0.999	1.000	1.000

Results using PCA and SVD features alone show that SVD shows poor results compared to PCA, especially for LR. Using all features also shows better performance; however, this performance is further improved when multi-features are used, as shown in Figure 5.

It can be observed that PCA gives better results as compared to SVD because PCA recognizes correlations in data and reduces the redundancy of data by constructing new smaller features. Applying multi-features from SVD and PCA, all learning models improved their results, and tree-based ensemble learning models obtained 100% accuracy.

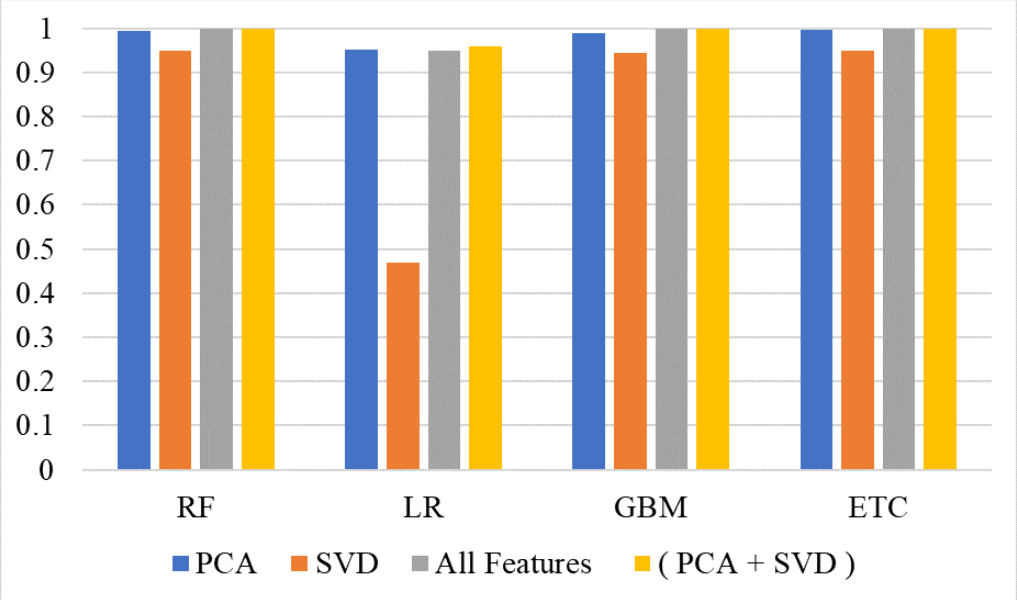

Figure 5. Models' performance comparison using all features techniques.

4.4. K-Fold Cross Validation

A 10-fold cross-validation was performed to show the significance of the proposed multi-features approach. Results of the 10-fold cross-validation with both 50 k and 25 k samples are given in Table 11. Results show that tree-based models show significant accuracy with 10-fold cross-validation with (+/− 0.00) standard deviation, while LR had a 0.96 accuracy score. LR performs better when the number of features is higher as compared to the number of samples, which is not the case with the current dataset.

Table 11. Results using 10-fold cross-validation.

Model	25 k	50 k
RF	1.00 (±0.00)	1.00 (±0.00)
LR	0.96 (±0.00)	0.96 (±0.00)
GBC	1.00 (±0.00)	1.00 (±0.00)
ETC	1.00 (±0.00)	1.00 (±0.00)

4.5. Performance of Deep Learning Models

Deep learning models are also deployed in this study for performance appraisal. In this regard, state-of-the-art CNN-LSTM [65], LSTM [66], CNN [67], and recurrent neural network (RNN) were deployed for DDoS attack detection. Performance analysis was carried out using the multi-features for deep learning models' training using 25 k and 50 k samples. Results using 25 k samples with multi-features are given in Table 12.

Table 12. Performance of deep learning models using multi-features with 25 k samples.

Model	Accuracy	Precision	Recall	F1 Score
CNN	1.00	1.00	1.00	1.00
LSTM	0.991	0.99	0.99	0.99
CNN-LSTM	0.993	0.99	0.99	0.99
RNN	0.991	0.99	0.99	0.99

Results for 50 k samples with multi-features are shown in Table 13. Results show that models perform well on 50 k samples as CNN achieved the highest accuracy score of 0.994 with 50 k samples and RNN shows poor performance as compared to other deep learning models with a 0.961 accuracy score. RNN performs poorly as a simple architecture of recurrent applications which does not perform well on large datasets. However, decreasing the size of the dataset increases its performance as shown in Table 12. With 25 k samples, CNN achieves the highest 1.00 accuracy using the multi-features approach while all other models show a 0.99 accuracy score on average.

Table 13. Performance of deep learning models using multi-features with 50 k samples.

Model	Accuracy	Precision	Recall	F1 Score
CNN	0.994	0.99	0.99	0.99
LSTM	0.990	0.99	0.99	0.99
CNN-LSTM	0.993	0.99	0.99	0.99
RNN	0.961	0.96	0.96	0.96

4.6. Computational Complexity of Multi-Features

One objective of this study is to obtain high DDoS attack detection accuracy with reduced computational complexity. Using all 78 features requires a substantial amount of time for models' training, which can be reduced using the multi-features approach without compromising the performance of models. Table 14 shows the computation time for each model using 25 k and 50 k samples used with the multi-features technique and all 78 features. It can be observed that the computational time using all 78 features was significantly high as compared to the proposed multi-features approach. In addition, despite using the low number of features, the provided accuracy was higher than using all 78 features.

Table 14. Computational time for all models.

Model	Multi-Features		Multi-Features	
	25 k	**50 k**	**25 k**	**50 k**
RF	74.00 s	172.00 s	94.20	184.6
LR	38.01 s	71.32 s	47.40	88.1
GBC	5588.12 s	5788.12 s	7023.00 s	8231.0 s
ETC	18.00 s	49.23 s	23.45 s	56.2 s
CNN	338.25 s	575.90 s	431.40 s	723.48 s
LSTM	800.54 s	1105.25 s	1003.00 s	1416.51 s
CNN-LSTM	2109.94 s	2266.99 s	2731.71 s	2915.45 s
RNN	581.69 s	698.00 s	734.55 s	897.04 s

4.7. Comparison with Existing Studies

A performance comparison was carried out with state-of-the-art studies to show the significance of the proposed multi-features approach for DDoS attack detection. For comparison, the selected models from the state-of-the-art studies were implemented and tested using the dataset used in this study. For performance comparison, we selected the most recent studies that have worked on similar tasks. We implemented the models as per their given implementation details and performed experiments in the same environment that was used for the proposed approach. The performance comparison results are shown in Table 15. The proposed approach is significantly better than existing approaches because it has simple architecture and shows robust results. We used state-of-the-art methods in combination with feature selection to obtain high accuracy. As a result, results are good in terms of both accuracy and efficiency, as most of the models achieved 100% accuracy scores using multi-features.

Table 15. Comparison with other studies.

Study	Year	Model	Accuracy
[36]	2020	NB	0.985
[34]	2021	LSTM, GRU, CNN, MLP, RF, LR, KNN	0.998, 0.998, 0.991, 0.998, 0.963, 0.997, 0.999
[37]	2021	MLP, RF	Without big data (0.990, 0.999) With big data (0.993, 0.999)
[38]	2021	LR, BNB, RT, KNN, REPTree	0.996, 0.998, 0.993, 0.998, 0.998
[39]	2021	SVM with RF	0.988
This study	2022	RF, LR, GBM, ETC, CNN, LSTM, CNN-LSTM, RNN using (PCA+SVD)	1.00, 0.960, 1.00, 1.00, 0.998, 0.991, 0.993, 0.991

Studies on DDoS attack detection predominantly work on optimizing the models to boost their performance. On the other hand, this study primarily focuses on selecting the optimal features for increased attack detection accuracy. The study [36] used NB for DDoS attack classification, and [34] used both deep learning and machine learning models, including LSTM, GRU, CNN, MLP, RF, LR, and KNN. The study [39] worked on an ensemble model and combined SVM and RF to achieve better performance. In comparison with other studies, the proposed approach outperforms previous studies and obtains superior results.

4.8. Discussion

The purpose of the system is to detect a DDoS attack using machine learning. With increasing DDoS attacks recently, an automated approach is needed to mitigate the risk of such attacks. The proposed approach has the potential to be used in real-time, as it is simple and effective and provides robust results. It can be implemented on network interfaces to detect DDoS attacks. Compared to the existing complicated approach that requires high computation resources, the proposed approach is more effective regarding the need for computation resources. Additionally, the provided attack detection accuracy is high, which provides more security against DDoS attacks. The use of a smaller yet effective hybrid feature set makes it work faster and better.

An SSL DDoS attack targets the SSL handshake protocol either by sending worthless data to the SSL server, which results in connection issues for legitimate users, or by abusing the SSL handshake protocol itself. An HTTP flood DDoS attack also uses HTTP post and obtains a request to access the server. Since the attacks on application layers are targeted in this study, attacks on protocols such as HTTP can also be covered by the proposed approach.

5. Conclusions

The internet network has persistently been exploited by DoS/DDoS attacks, and the number of these attacks have substantially increased over the past few years. Despite the available advanced and sophisticated attack detection approaches, network security remains a challenging problem today. This study proposes a machine learning-based framework to detect DDoS/DoS attacks at the application layer by using multi-features. For multi-features, the best features from PCA and SVD are extracted to obtain superior performance. Extensive experiments are carried out using LR, RF, GB, ETC, as well as, CNN, LSTM, CNN-LSTM, and RNN models using different dataset sizes and different features. Experimental results indicate that PCA features tend to show better performance as compared to SVD features. Despite the results being better with using all features, multi-features help optimize the models' performance. The RF model outperforms all other models by obtaining 100% accuracy when used with multi-features. On average, the tree-based ensemble models show better performance than linear models. This study shows excellent results, yet has several limitations. First, the approach is tested on a single dataset and requires further experiments regarding attacks on other layers. Second, the impact of dataset size is not investigated. Increasing the size might produce better results for deep learning models. Third, the influence of feature set size is also not analyzed, which we intend to perform in future work.

Author Contributions: Conceptualization, F.R., A.H. and M.F.M.; data curation, A.H.; formal analysis, M.F.M.; funding acquisition, A.D.J.; investigation, F.R. and A.D.J.; methodology, F.R. and M.S.F.; project administration, M.F.M. and M.S.F.; resources, A.D.J.; software, F.R.; supervision, A.D.J. and I.A.; validation, A.D.J. and I.A.; visualization, M.S.F.; writing—original draft, A.H. and F.R.; writing—review and editing, I.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the School of Computer Science, University College Dublin, Dublin, Ireland.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that there is no conflict of interests.

Abbreviations

The following abbreviations are used in this paper.

Acronym	Detailed
DDoS	Distributed Denial of Services
DNS	Domain Name System
DT	Decision Tree
ETC	Extra Trees Classifier
GBM	Gradient Boosting Machine
GE	Generalized Entropy
GID	Generalized Information Distance
GRU	Gated Recurrent Unit
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
KNN	K-Nearest Neighbour
LR	Logistic Regression
LSTM	Long Short Term Memory
MLP	Multilayer Perceptron
NB	Naive Bayes
PCA	Principal Component Analysis
RF	Random Forest
RNN	Recurrent Neural Networks
SVD	Singular Value Decomposition
SVM	Support Vector Machine

References

1. Islam, U.; Muhammad, A.; Mansoor, R.; Hossain, M.S.; Ahmad, I.; Eldin, E.T.; Khan, J.A.; Rehman, A.U.; Shafiq, M. Detection of distributed denial of service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models. *Sustainability* **2022**, *14*, 8374. [CrossRef]
2. DDoS Attacks in Q3 2022. Available online: https://securelist.com/ddos-report-q3-2022/107860/#:~:text=The%20number%20of%20DDoS%20attacks%20in%20Q3%202022%20fell%20again,further%2027.29%20percent%2C%20to%2057%2C116 (accessed on 25 October 2022).
3. Hacktivism and DDOS Attacks Rise Dramatically in 2022. Available online: https://www.govtech.com/blogs/lohrmann-on-cybersecurity/hacktivism-and-ddos-attacks-rise-dramatically-in-2022 (accessed on 25 October 2022).
4. Famous DDoS Attacks | The Largest DDoS Attacks of All Time. Available online: https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/ (accessed on 25 October 2022).
5. Ramalingam, H.; Venkatesan, V.P. Conceptual analysis of Internet of Things use cases in Banking domain. In Proceedings of the TENCON 2019-2019 IEEE Region 10 Conference (TENCON), Kochi, India, 17–20 October 2019; pp. 2034–2039.
6. George, A.; Ravindran, A.; Mendieta, M.; Tabkhi, H. Mez: An adaptive messaging system for latency-sensitive multi-camera machine vision at the iot edge. *IEEE Access* **2021**, *9*, 21457–21473. [CrossRef]
7. George, A.; Ravindran, A. Distributed middleware for edge vision systems. In Proceedings of the 2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT), Charlotte, NC, USA, 6–9 October 2019; pp. 193–194.
8. Mendieta, M.; Neff, C.; Lingerfelt, D.; Beam, C.; George, A.; Rogers, S.; Ravindran, A.; Tabkhi, H. A Novel Application/Infrastructure Co-design Approach for Real-time Edge Video Analytics. In Proceedings of the 2019 SoutheastCon, Atlanta, GA, USA, 10–13 March 2019; pp. 1–7.
9. Xanthidis, D.; Nicholas, D. Evaluating internet usage and ecommerce growth in Greece. In *Proceedings of the Aslib Proceedings*; Emerald Group Publishing Limited: Bingley, UK, 2004.
10. Ch, A.; Ch, R.; Gadamsetty, S.; Iwendi, C.; Gadekallu, T.R.; Dhaou, I.B. ECDSA-Based Water Bodies Prediction from Satellite Images with UNet. *Water* **2022**, *14*, 2234. [CrossRef]
11. Liu, J.; Zhang, W.; Ma, T.; Tang, Z.; Xie, Y.; Gui, W.; Niyoyita, J.P. Toward security monitoring of industrial Cyber-Physical systems via hierarchically distributed intrusion detection. *Expert Syst. Appl.* **2020**, *158*, 113578. [CrossRef]
12. Fallows, D. *The Internet and Daily Life*; Pew Internet & American Life Project: Washington, DC, USA, 2004.
13. Gupta, M.; Abdelsalam, M.; Khorsandroo, S.; Mittal, S. Security and Privacy in Smart Farming: Challenges and Opportunities. *IEEE Access* **2020**, *8*, 34564–34584. [CrossRef]
14. Alqahtani, A.S. Security threats and countermeasures in software defined network using efficient and secure trusted routing mechanism. *Comput. Commun.* **2020**, *153*, 336–341. [CrossRef]

15. Al-Ghamdi, A.; Al-Sulami, A.; Aljahdali, A.O. On the security and confidentiality of quantum key distribution. *Secur. Priv.* **2020**, *3*, 1–14. [CrossRef]
16. Gunduz, M.Z.; Das, R. Cyber-security on smart grid: Threats and potential solutions. *Comput. Netw.* **2020**, *169*, 107094. [CrossRef]
17. Jaafar, G.A.; Abdullah, S.M.; Ismail, S. Review of Recent Detection Methods for HTTP DDoS Attack. *J. Comput. Netw. Commun.* **2019**, *2019*, 1283472. [CrossRef]
18. Rahman, O.; Quraishi, M.A.G.; Lung, C.H. DDoS attacks detection and mitigation in SDN using machine learning. *Proc. 2019 IEEE World Congr. Serv. Serv.* **2019**, *2642-939X*, 184–189. [CrossRef]
19. Amjad, A.; Alyas, T.; Farooq, U.; Tariq, M. Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm. *ICST Trans. Scalable Inf. Syst.* **2018**, *6*, 159834. [CrossRef]
20. Sreeram, I.; Vuppala, V.P.K. HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm. *Appl. Comput. Inform.* **2019**, *15*, 59–66. [CrossRef]
21. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowl.-Based Syst.* **2020**, *189*, 105124. [CrossRef]
22. Masdari, M.; Khezri, H. A survey and taxonomy of the fuzzy signature-based Intrusion Detection Systems. *Appl. Soft Comput. J.* **2020**, *92*, 106301. [CrossRef]
23. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Network anomaly detection: Methods, systems and tools. *IEEE Commun. Surv. Tutorials* **2013**, *16*, 303–336. [CrossRef]
24. Meng, W.; Li, W.; Su, C.; Zhou, J.; Lu, R. Enhancing trust management for wireless intrusion detection via traffic sampling in the era of big data. *IEEE Access* **2017**, *6*, 7234–7243. [CrossRef]
25. Singh, K.; Singh, P.; Kumar, K. Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges. *Comput. Secur.* **2017**, *65*, 344–372. [CrossRef]
26. O'Ree, A.J.; Obaidat, M.S. Security enhancements for UDDI. *Secur. Commun. Netw.* **2011**, *4*, 871–887. [CrossRef]
27. Zargar, S.T.; Joshi, J.; Tipper, D. A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 2046–2069. [CrossRef]
28. Kesavamoorthy, R.; Alaguvathana, P.; Suganya, R.; Vigneshwaran, P. Classification of DDoS attacks—A survey. *Test Eng. Manag.* **2020**, *83*, 12926–12932.
29. Chang, R.K. Defending against flooding-based distributed denial-of-service attacks: A tutorial. *IEEE Commun. Mag.* **2002**, *40*, 42–51. [CrossRef]
30. Jazi, H.H.; Gonzalez, H.; Stakhanova, N.; Ghorbani, A.A. Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. *Comput. Netw.* **2017**, *121*, 25–36. [CrossRef]
31. Behal, S.; Kumar, K.; Sachdeva, M. D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events. *J. Netw. Comput. Appl.* **2018**, *111*, 49–63. [CrossRef]
32. Wang, C.; Miu, T.T.; Luo, X.; Wang, J. SkyShield: A sketch-based defense system against application layer DDoS attacks. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 559–573. [CrossRef]
33. Aamir, M.; Zaidi, S.M.A. Clustering based semi-supervised machine learning for DDoS attack classification. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *33*, 436–446 [CrossRef]
34. Yungaicela-Naula, N.M.; Vargas-Rosales, C.; Perez-Diaz, J.A. SDN-Based Architecture for Transport and Application Layer DDoS Attack Detection by Using Machine and Deep Learning. *IEEE Access* **2021**, *9*, 108495–108512. [CrossRef]
35. Pubudu, R.D.; Indrasiri, L.; Lee, E.; Rupapara, V.; Rustam, F.; Ashraf, I. Malicious Traffic Detection in IoT and Local Networks Using Stacked Ensemble Classifier. *Comput. Mater. Contin.* **2022**, *71*, 489–515. [CrossRef]
36. Priya, S.S.; Sivaram, M.; Yuvaraj, D.; Jayanthiladevi, A. Machine learning based DDoS detection. In Proceedings of the 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 5–7 March 2020; pp. 234–237.
37. Awan, M.J.; Farooq, U.; Babar, H.M.A.; Yasin, A.; Nobanee, H.; Hussain, M.; Hakeem, O.; Zain, A.M. Real-Time DDoS Attack Detection System Using Big Data Approach. *Sustainability* **2021**, *13*, 743. [CrossRef]
38. Dehkordi, A.B.; Soltanaghaei, M.; Boroujeni, F.Z. The DDoS attacks detection through machine learning and statistical methods in SDN. *J. Supercomput.* **2021**, *77*, 2383–2415. [CrossRef]
39. Ahuja, N.; Singal, G.; Mukhopadhyay, D.; Kumar, N. Automated DDOS attack detection in software defined networking. *J. Netw. Comput. Appl.* **2021**, *187*, 103108. [CrossRef]
40. Almaraz-Rivera, J.G.; Perez-Diaz, J.A.; Cantoral-Ceballos, J.A. Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models. *Sensors* **2022**, *22*, 3367. [CrossRef] [PubMed]
41. Application Layer DoS Attack Dataset. Available online: https://www.kaggle.com/hamzasamiullah/ml-analysis-application-layer-dos-attack-dataset (accessed on 30 August 2021).
42. Vickers, N.J. Animal communication: When i'm calling you, will you answer too? *Curr. Biol.* **2017**, *27*, R713–R715. [CrossRef] [PubMed]
43. Bahassine, S.; Madani, A.; Al-Sarem, M.; Kissi, M. Feature selection using an improved Chi-square for Arabic text classification. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *32*, 225–231. [CrossRef]
44. Reddy, G.T.; Reddy, M.P.K.; Lakshmanna, K.; Kaluri, R.; Rajput, D.S.; Srivastava, G.; Baker, T. Analysis of Dimensionality Reduction Techniques on Big Data. *IEEE Access* **2020**, *8*, 54776–54788. [CrossRef]

45. Gao, J.; Chai, S.; Zhang, B.; Xia, Y. Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis. *Energies* **2019**, *12*, 1223. [CrossRef]
46. Al-Saffar, Z.A.; Yildirim, T. A Novel Approach to Improving Brain Image Classification Using Mutual Information-Accelerated Singular Value Decomposition. *IEEE Access* **2020**, *8*, 52575–52587. [CrossRef]
47. Punithavalli, S.S. PCA and SVD based Feature Reduction for Cardiac Arrhythmia Classification. *Int. J. Eng. Res. Technol. (IJERT)* **2014**, *3*, 1544–1551. [CrossRef]
48. Zare, M.; Eftekhari, M.; Aghamollaei, G. Supervised feature selection via matrix factorization based on singular value decomposition. *Chemom. Intell. Lab. Syst.* **2019**, *185*, 105–113. [CrossRef]
49. Varoquaux, G.; Buitinck, L.; Louppe, G.; Grisel, O.; Pedregosa, F.; Mueller, A. Scikit-learn. *GetMobile Mob. Comput. Commun.* **2015**, *19*, 29–33. [CrossRef]
50. Rustam, F.; Mehmood, A.; Ahmad, M.; Ullah, S.; Khan, D.M.; Choi, G.S. Classification of Shopify App User Reviews Using Novel Multi Text Features. *IEEE Access* **2020**, *8*, 30234–30244. [CrossRef]
51. Archer, K.J.; Kimes, R.V. Empirical characterization of random forest variable importance measures. *Comput. Stat. Data Anal.* **2008**, *52*, 2249–2260. [CrossRef]
52. Ham, J.S.; Chen, Y.; Crawford, M.M.; Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501. [CrossRef]
53. Idhammad, M.; Afdel, K.; Belouch, M. Detection System of HTTP DDoS Attacks in a Cloud Environment Based on Information Theoretic Entropy and Random Forest. *Secur. Commun. Netw.* **2018**, *2018*, 1263123. [CrossRef]
54. Hosseini, S.; Azizi, M. The hybrid technique for DDoS detection with supervised learning algorithms. *Comput. Netw.* **2019**, *158*, 35–45. doi: 10.1016/j.comnet.2019.04.027. [CrossRef]
55. Rustam, F.; Ashraf, I.; Mehmood, A.; Ullah, S.; Choi, G.S. Tweets classification on the base of sentiments for US airline companies. *Entropy* **2019**, *21*, 1078. [CrossRef]
56. Austin, P.C.; Merlo, J. Intermediate and advanced topics in multilevel logistic regression analysis. *Stat. Med.* **2017**, *36*, 3257–3277. doi: 10.1002/sim.7336. [CrossRef] [PubMed]
57. Lee, S. Logistic Regression Procedure Using Penalized Maximum Likelihood Estimation for Differential Item Functioning. *J. Educ. Meas.* **2019**, *57*, 1–15. [CrossRef]
58. Rymarczyk, T.; Kozłowski, E.; Kłosowski, G.; Niderla, K. Logistic regression for machine learning in process tomography. *Sensors* **2019**, *19*, 1–19. [CrossRef]
59. Khairunnahar, L.; Hasib, M.A.; Rezanur, R.H.B.; Islam, M.R.; Hosain, M.K. Classification of malignant and benign tissue with logistic regression. *Inform. Med. Unlocked* **2019**, *16*, 100189. [CrossRef]
60. Lian, J.; Zhao, Q.F. Prediction of heparin dose during continuous renal replacement therapy surgery by using the gradient boosting regression model. In Proceedings of the 6th International Conference on Control, Decision and Information Technologies, CoDIT, Paris, France, 23–26 April 2019; pp. 182–186. [CrossRef]
61. Alqahtani, M.; Gumaei, A.; Mathkour, H.; Ismail, M.M.B. A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks. *Sensors* **2019**, *19*, 4383. [CrossRef] [PubMed]
62. Rahman, S.S.M.M.; Rahman, M.H.; Sarker, K.; Rahman, M.S.; Ahsan, N.; Sarker, M.M. Supervised Ensemble Machine Learning Aided Performance Evaluation of Sentiment Classification. *J. Phys. Conf. Ser.* **2018**, *1060*, 012036. [CrossRef]
63. Leghari, M.; Memon, S.; Sahito, F.; Chandio, A.A.; Leghari, M. Biometric verification enhancement with ensemble learning classifiers. In Proceedings of the 5th International Multi-Topic ICT Conference: Technologies For Future Generations, IMTIC, Jamshoro, Pakistan, 25–17 Apil 2018; pp. 1–6. [CrossRef]
64. Confusion Matrix, Accuracy, Precision, Recall, F1 Score. Available online: https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd (accessed on 25 October 2022).
65. Jamil, R.; Ashraf, I.; Rustam, F.; Saad, E.; Mehmood, A.; Choi, G.S. Detecting sarcasm in multi-domain datasets using convolutional neural networks and long short term memory network model. *PeerJ Comput. Sci.* **2021**, *7*, e645. [CrossRef] [PubMed]
66. Rupapara, V.; Rustam, F.; Amaar, A.; Washington, P.B.; Lee, E.; Ashraf, I. Deepfake tweets classification using stacked Bi-LSTM and words embedding. *PeerJ Comput. Sci.* **2021**, *7*, e745. [CrossRef]
67. Siddiqui, H.U.R.; Shahzad, H.F.; Saleem, A.A.; Khan Khakwani, A.B.; Rustam, F.; Lee, E.; Ashraf, I.; Dudley, S. Respiration Based Non-Invasive Approach for Emotion Recognition Using Impulse Radio Ultra Wide Band Radar and Machine Learning. *Sensors* **2021**, *21*, 8336. [CrossRef] [PubMed]