Journal of Business Data Science Research

Combining Particle Swarm Optimization and Entropy to Detect DDoS Attacks in the Cloud Computing

Amin Rezaeipanah[a,*], Sayed Ebrahim Mousavipoor[b], Mehdi Asayeshjoo[c], and Mehdi Sadeghzadeh[d]

[a] Department of Computer Engineering, University of Rahjuyan Danesh Borazjan, Bushehr, Iran.
[b] Department of Computer Engineering, Liyan Institute of Education, Bushehr, Iran.
[c] Department of Computer Engineering, Bushehr Branch, Islamic Azad University, Bushehr, Iran.
[d] Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran.

ABSTRACT

Cloud computing is an emerging technology that is widely used to provide computing, data storage services and other remote resources over the Internet. Availability of cloud services is one of the most important concerns of cloud service providers. While cloud services are mainly transmitted over the Internet, they are prone to various attacks that may lead to the leakage of sensitive information. Distributed Denial of Service (DDoS) attack is known as one of the most important security threats to the cloud computing environment. This attack is an explicit attempt by an attacker to block or deny access to shared services or resources in a cloud environment. This paper discusses a hybrid approach to dealing with DDoS attacks in the cloud computing environment. This method highlights the importance of effective feature-based selection methods and classification models. Here, an entropy-based approach and particle swarm optimization to counter these attacks in a cloud computing environment is presented. Categorizing high-dimensional data usually requires selecting the attribute as a pre-processing step to reduce the size. However, selecting effective features is a challenging task, which in this paper uses particle swarm optimization. Here, the proposed classification model is developed based on the use of a balanced binary search tree and dictionary data structure. The simulation is based on the NSL-KDD and CICDDoS2019 datasets, which prove the superiority of the proposed method with an average detection accuracy of 99.84% over the AGA, E-SVM and AE-DNN algorithms.

KEWWORDS: Cloud Computing, Attack Detection, Entropy, Particle Swarm Optimization, DDoS Attack.

ORGINAL ARTICLE

Received 30 March 2021
Revised 12 May 2021
Accepted 08 July 2021

CITATION

Rezaeipanah, A., Mousavipoor, S. E., Asayeshjoo, M.,, and Sadeghzadehd, M., (2021). Combining Particle Swarm Optimization and Entropy to Detect DDoS Attacks in the Cloud Computing. Journal of Business Data Science Research, 1(1), 33-43.

1. Introduction

The emergence of new technologies in the field of information technology, especially in the field of web, as well as new business models, along with the production of abundant data, has necessitated changes in the fields of processing and service delivery. The new technology recently introduced is called cloud computing (Shah et al., 2021). Cloud computing is a term used for providing web hosting services that offers great potential for improving productivity and reducing costs (Rezaeipanah *et al.*, 2021). Outstanding features offered by cloud computing technology are request service, resource sharing, wide area network access, data storage, etc (SaiSindhuTheja and Shyam, 2021). Despite the numerous benefits of cloud computing, this technology faces several security issues (Osanaiye *et al.*, 2016; Bawa *et al.*, 2017; Kholidy, 2021). Various security issues in cloud service models have been investigated (Subashini *et al.*, 2011). Among security issues, availability is mentioned as the most important concern (Agrawal *et al.*, 2019), because the

* Corresponding author: amin.rezaeipanah@gmail.com

• http://www.jbdsr.com/article_683074.html

main function of cloud computing is to provide the requested services. One of the major threats to accessibility is Distributed Denial of Service (DDoS) attacks (Shah *et al.*, 2021). Therefore, recognizing and counteracting this type of security attacks on the cloud is of great importance.

A DDoS attack is a special type of denial of service (DoS) attack in which multiple distributed machines are targeted and target the victim cloud server (Somani *et al.*, 2017a). The percentage of DDoS attacks targeting cloud computing services and resources is increasing every year, according to Arbor (Praseed *et al.*, 2018). Well-known cloud-based companies such as Amazon, Sony, Google and Microsoft have been the target of DDoS attacks in recent years (Somani *et al.*, 2017b). These attacks have disrupted services, economic losses, and many short- and long-term effects on cloud service providers. In general, DDoS attacks can be classified into two categories: pervasive and semantic search (Shameli-Sendi *et al.*, 2015). In pervasive search attacks, attackers send large amounts of malicious requests to end the target cloud bandwidth (Zargar *et al.*, 2013). These attacks are easily detected by defense mechanisms due to their high rate (Praseed *et al.*, 2018). Semantic attacks, on the other hand, take advantage of the weakness of cloud computing protocols instead of depleting resources and network bandwidth. An attacker generates a small amount of malicious attacks to target a specific protocol. Such attacks, known as low-rate DDoS attacks, look like legal traffic. Therefore, it is difficult to identify this type of attack compared to pervasive search attacks (Agrawal and Tapaswi, 2017).

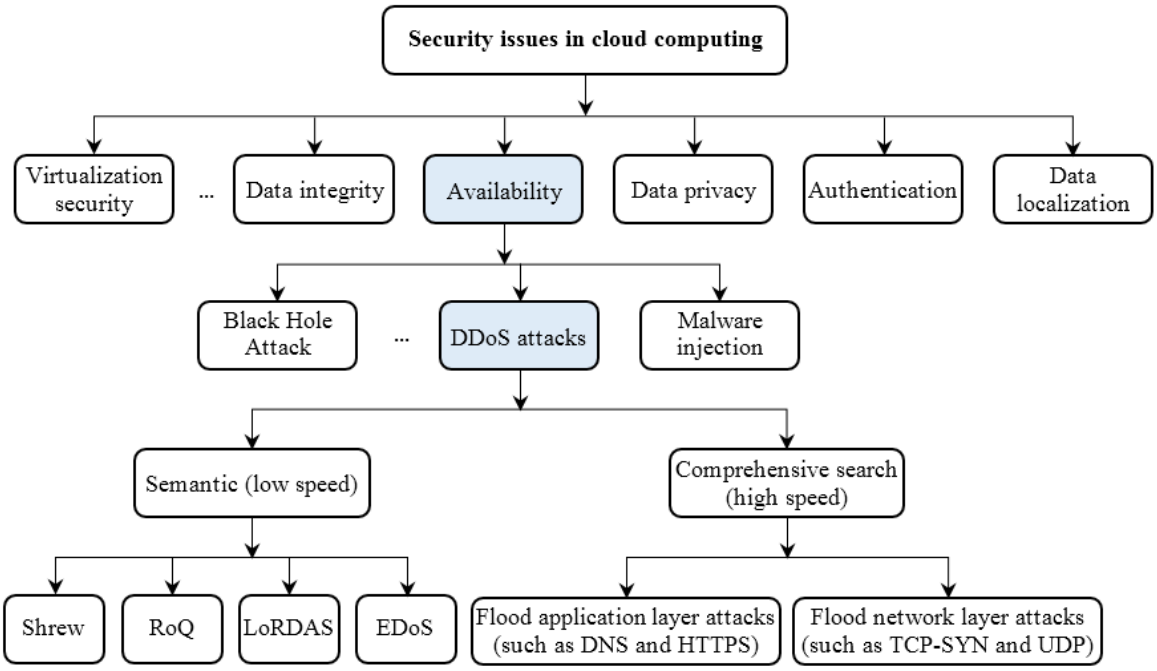

Figure 1. DDoS attack and its variants in a cloud computing environment

With technical advances, cloud platforms are becoming more powerful in terms of resources (Shidaganti *et al.*, 2020). Because of these large resources on the cloud side, it is difficult to threaten them even with DDoS attacks. An example of such a failed DDoS attack (El-Sofany and Systems, 2020) reportedly launched by an unknown group against the Amazon cloud. As a result, attackers are trying to reduce the quality of cloud services without early detection using sophisticated DDoS attacks. In 2015, a severe DDoS attack was launched against Amazon, causing $ 30,000 in daily financial losses. Figure 1 provides a complete scenario of a DDoS attack and its variants in a cloud computing environment.

Many methods for detecting DDoS attacks have been reported in the sources, but only a few of them have been used in a real network environment and have effective performance (El-Sofany and Systems, 2020). Designing and implementing an ideal and practical defense system is really difficult (Bhardwaj *et al.*, 2021). Therefore, before presenting a defensive method, it is necessary to be aware of the characteristics that are effective in detecting this type of attack (Bhardwaj *et al.*, 2021). Due to the frequency of using different features, in this paper, a swarm optimization algorithm is used to select a subset of effective features in detecting DDoS attacks. In addition, a classification approach based on information entropy calculation is used to model attack detection. The proposed classification model is based on the use of a balanced binary search tree data structure and a dictionary. It is believed that this paper can motivate security researchers to develop effective defense solutions to prevent DDoS attacks in the cloud.

The rest of this paper is organized as follows: Section 2 discusses related works to attacks detection in cloud computing. The proposed method is presented in the section 3 and the experimental results are reported in section 4. Finally, section 5 is devoted to conclusions and suggestions.

2. Literature review

Although DDoS attacks come in many forms, they are all intended to suppress servers, firewalls, or other defined devices by sending high-rate request packets (Saied *et al.*, 2016). In this regard, cloud networks with inaccessible websites are under more pressure than other platforms (Bhardwaj *et al.*, 2021). The following section discusses intrusion detection techniques available for the cloud computing environment.

Rastegari *et al.* (2015), an intrusion detection system based on genetic algorithms and fuzzy rules is proposed. The classification model is based on the NSL-KDD dataset using the if-then set of rules on the properties selected by the genetic algorithm. In addition, condition domains in fuzzy laws are selected by genetic algorithms with the aim of minimizing the number of non-false classification samples. In this paper, correlation-based feature selection (CFS) and compatibility subset evaluation (CSE) are also analyzed, Kanakarajan and Muniasamy (2016), a method based on the greedy randomized adaptive search method with a random refrigeration classification (GAR-forest) has been proposed to detect attacks in the cloud. This method is proposed for both binary classification and multi-tag classification on the NSL-KDD dataset. This method produces a set of consistent random decision trees through information gain. The results of this method show that the accuracy of attack detection has improved compared to RF, C4.5, NB and MLP models.

Ingre and Yadav (2015), a classification model using artificial neural networks for the NSL-KDD dataset was proposed. This method is based on both two-class and five-class models. The results were analyzed based on different criteria and the results show better accuracy for this method. However, the categorization ratio still needs to be improved. The detection rate of this method is 81.2% for penetration and 79.9% for sub-attack categorization. Bamakan *et al.* (2016), an effective intrusion detection framework using a precise and adaptive optimization method is presented. This method uses time-varying chaos particle swarm optimization (TVCPSO) to simultaneously adjust parameters and select properties. Here, modeling is performed based on multiple criteria linear programming (MCLP) and support vector machine (SVM). In addition, a weighted objective function is presented that balances the maximum detection rate, the minimization of the false alarm rate, as well as the number of selected features.

Ghalehgolabi *et al.* (2017), the AGA algorithm is introduced as an intrusion detection system. AGA uses genetic algorithms and data mining techniques based on feature reduction to detect intrusion. Here, an improved genetic algorithm is proposed for the task of selecting effective traits whose parameters are controlled comparatively. For example, the jump rate parameter has a relatively high value at the beginning of the operation and decreases in the algorithm execution process, respectively. AGA is easy to implement and has low computational complexity. The experimental results of this algorithm on NSL-KDD data set provide higher accuracy in intrusion detection with false alarm and less number of features. Yang (2019), the E-SVM algorithm for dealing with DDoS attacks in the cloud computing environment is presented. This algorithm detects network traffic anomalies based on data entropy measurement and SVM model. E-SVM uses six features including source IP address, source port, destination IP address, destination port, packet type, and the number of network packets for modeling. Experimental results show that this algorithm can detect network anomalies in the large-scale data set with higher accuracy.

(Agrawal *et al.*, 2021), a defense mechanism using software-centric networks (SDNs) is introduced to detect DDoS attacks in a cloud computing environment. This discusses a new mechanism that not only detects and reduces DDoS attacks, but also tracks the location of attack sources. Here, attacks are identified using information entropy changes, and attack sources are tracked using a packet marking scheme. This method detects attack sources between 14.45 milliseconds to 10.02 seconds and provides 97.6% accuracy. Sharma *et al.* (2020), an entropy-based approach to detect real-time DDoS attacks using the Hadoop framework has been proposed. In this paper, the entropy of source addresses is used as a measure of DDoS and real-time analysis of large volumes of traffic. The results of this method show that DDoS attacks can be detected more accurately in real-time. Bhardwaj *et al.* (2020), hyperband tuned deep neural network (DNN) with well-posed stacked Sparse autoencoder (AE) is proposed to detect of DDoS attacks in cloud. AE and DNN are optimized for detection of DDoS attacks by tuning the parameters using appropriately designed techniques.

3. Proposed method

The purpose of this paper is to identify and reduce DDoS attacks in the cloud computing environment. Here, a hybrid algorithm is proposed to detect these attacks based on effective features that separate DDoS attack traffic from actual traffic. Using and accessing appropriate raw data in data mining to discover knowledge is referred to as data preparation or processing. The importance of data processing is due to the fact that; Lack of quality data equals lack of quality in exploration results, and poor input leads to poor output. Therefore, in the first step of the proposed method, the input data is pre-processed with the aim of improving the quality.

One of the problems is the implementation of forecasting systems and the recognition of high information and high number of input features. The presence of irrelevant and redundant features in the data set negatively affects the performance of learning models and also increases computational complexity. In this paper, a particle swarm optimization algorithm is used to reduce computational complexity. This algorithm selects a subset of effective features for use in the classification model with the aim of reducing the size of the data. In addition, the technique of calculating the entropy of information and creating a balanced binary search tree is used to create the classification model. This tree becomes a grammar, so that in this mapping the root node is considered as a symbol and each node is considered as a variable in the grammar. Figure 2 shows the flowchart of the proposed method.

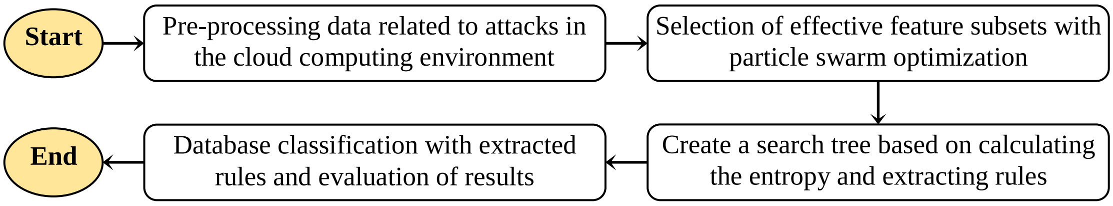
Start
Pre-processing data related to attacks in the cloud computing environment
Selection of effective feature subsets with particle swarm optimization
Create a search tree based on calculating the entropy and extracting rules
Database classification with extracted rules and evaluation of results
End

Figure 2. Flowchart of the proposed method

3.1 Data processing

In this paper, the database information used includes the information of the network input records, so that the record label (attack and non-attack) is also available. In general, preprocessing of data is required to create any classification model. Here, preprocessing is applied to improve the quality of the actual data. This stage of the research method consists of two parts: preparation and normalization. In the preparation phase, the values of the features that have qualitative values are converted to quantitative values. To do this, a number is assigned to each distinct value of each feature, and these numbers are replaced by qualitative values in the database. In addition, records with lost data are deleted from the database.

In the next step, the data is normalized. Normalization is for mapping data from the current interval to a specified interval. This is done due to the variety of features, reducing the impact of features with high values as well as bringing the predictions of the classification models closer together. This paper uses the ZScore technique for normalization, as shown in equation 1. This method sets the mean and standard deviation for each feature to be 0 and 1, respectively.

$$x_{i,j}^{ZScore} = \frac{x_{i,j} - \mu_j}{\sigma_j} \qquad (1)$$

Where, $x_{i,j}$ and $x_{i,j}^{ZScore}$ the actual value and the normalized value are the j-th feature in the i-th sample, respectively. μ_j and σ_j the mean and standard deviation are the values of all samples for the j-th feature, respectively.

3.2 Features selection with PSO algorithm

Due to the large volume of features of attack detection databases in cloud computing environments, in this paper, a subset of optimal features are searched and selected using the particle swarm optimization algorithm. The steps of the algorithm are as follows:

- Step 1: Create the initial population of particles at random, so that each particle expresses a subset of features. The length of each particle is equal to the size of the principal features, m. Each element of a particle has a value of 0 or 1, respectively, means to select or not to select a feature. Figure 3 shows the particle structure.

f_1	f_2	…	f_k	…	f_m

Figure 3. Particle structure in the features selection problem

Here, f_j in the sense of the state, the j-th feature is in the particle and is expressed as a binary number (0 no feature selection and 1 feature selection), where $j = 1,2,\dots,m$.

- Step 2: The fitness function is calculated for each particle. Here, the accuracy of the classification model proposed in the next section, the sum of the similarities between the selected features, and the number of selected features are used to calculate the fitness function. Equation 2 defines the fitness function for the k-th particle.

$$Fitness(P_k) = \frac{Acc(p_k)}{|p_k| \times \sum_{f_i,f_j \in p_k} Sim(f_i,f_j)} \qquad (2)$$

Where, $Acc(p_k)$ accuracy of classification according to the selected features in the k-particle, $|p_k|$ the number of features selected by the k-th particle and $Sim(f_i,f_j)$ Shows the similarity between the features f_i and f_j of the k-th particle. Here, according to equation 3, the Pearson correlation coefficient is used to calculate the similarity between the features.

$$sim(f_i,f_j) = \frac{\sum_{k=1}^{n}(f_{i,k} - \bar{f}_i)(f_{j,k} - \bar{f}_j)}{\sum_{i=1}^{n}(f_{i,k} - \bar{f}_i)^2 \sum_{i=1}^{n}(f_{j,k} - \bar{f}_j)^2} \qquad (3)$$

Where, $f_{i,k}$ and $f_{j,k}$ and the k-th sample in order for the features f_i and f_j shows, $\bar{f}_i$ and $\bar{f}_j$, respectively the average of all samples for features is f_i and f_j and n refers to the total number of samples.

- Step 3: The position of the particles is updated using equation 4.

$$p_k = \langle p_k + v_k \rangle \qquad (4)$$

Where, v_k and p_k the current velocity and position of the particle are k-m, respectively. The term $\langle p_k + v_k \rangle$ is classically used for the continuous environment, however the above problem space is discrete. Hence, this term is developed discretely according to equation 5.

$$p_{k,j} = \begin{cases} 0 & \langle p_{k,j} + v_{k,j} \rangle < 0.5 \\ 1 & otherwise \end{cases} \qquad (5)$$

Where, $p_{k,j}$ is the value of the j-th feature (position) of the k-particle. Here, the position of the feature is updated based on the velocity of the particle.

- Step 4: The particle velocities are updated using equation 6.

$$v_k = v_k \times \omega + c_1 \times r_1 \times (pbest_k - p_k) + c_2 \times r_2 \times (gbest - p_k), \quad v_k \in [v_{min}, v_{max}] \qquad (6)$$

Where, c_1 and c_2 fixed learning parameters, r_1 and r_2 random numbers in the range 0 and 1, $pbest_k$ the best position of a k-particle so far and $gbest$ the best global position of all particles. The particle velocity in each feature is the same v_{max} are limited. If the sum of the accelerations cause the velocity in a feature of v_{max} the greater the value of the velocity in that feature v_{max} is located. Similarly the particle velocity in each feature is equal to one value v_{min} are also limited.

- Step 5: The optimal positions of $pbest$ and $gbest$ are calculated according to the particle fitness values.
- Step 6: The previous steps are repeated until the termination condition of the algorithm is established. In this paper, a constant number of iterations is used for the termination condition of the algorithm.

3.3. Classification based on calculating the entropy

In this section, a classification model based on the use of a balanced binary search tree data structure and an association array (dictionary) is presented. In this algorithm, the parameters “input rate per host”, “output rate per host”, “rate of increase in input-output ratio per host” and “entropy” are used to identify attacks. The input rate is the amount of input packet to a host and the output rate is the amount of output packet to each host, which is considered relative to the equation 7.

$$R(p) = \frac{input - rate(p)}{output - rate(p)} \tag{7}$$

Where, p is a specific address link for a host.

The rate of increase of the input to output ratio in each host is considered to be $R_t(p)$, where it indicates the ratio of the input rate to the output rate at time t for host p. To calculate the rate of increase of the ratio$R(p)$, time windows with length w are considered. Here, equation 8 is used to apply the rate of change of $R(p)$ to time.

$$Slope(p) = \frac{R_{t_{end}}(p) - Slope_{t_{start}}(p)}{t_{start} - t_{end}}, \quad w = [t_{start}, t_{end}] \tag{8}$$

Where, considering the Slope of the previous step, the amount and how the $R(p)$ changes to detect attacks. Therefore, in general, two parameters R and $Slope$ are used to detect DDoS attacks.

In this paper, for classification work, a tree data structure is used to count and maintain the host input and output rate and its increment rate. The intended data structure is a quadratic balanced binary search tree of 256, which is a four-level address link for complete coverage. By calculating the values $R(p)$ ،$Slope(p)$ and considering two values R_{min} and R_{max} are identified as two thresholds for DDoS attacks according to equation 9.

$$\begin{cases} If\ (R_{min} \le R \le R_{max})\ and\ (slope \le slope_{max}), & valid\ \ action \\ If\ \ (R < R_{min})\ or\ (R > R_{max})\ and\ (slope > slope_{max}), & invalid\ \ action \end{cases} \tag{9}$$

Due to storing all the address links in the tree and repeating the update operation, the tree must be scanned once after receiving each package. Hence, there is a need for a mechanism to reduce the number of tree navigation operations. For this purpose, the binary search tree compression approach is used, where 256 tree nodes are mapped to a balanced binary search tree. Here a grammar pattern is used to compress the tree, where the path from the tree is converted to a grammar. In this mapping, the root node is considered as the start symbol and each node is considered as a variable in the grammar. Also, each edge is a transfer from the source variable to the destination variable. Additionally, when scrolling through a binary search tree, scrolling into any node that enters, the tree that has been scrolled so far is compressed using the compression procedure and stored in an association array (dictionary). The reason for saving is that in other packages, if part of the navigation path is the same, the tree is no longer navigated, but this path is extracted from the relevant community array at a fixed time and added to the current path.

The proposed method is then developed by calculating the entropy of information. Entropy is an important concept in information theory that measures the randomness (irregularity) of data entering the network. The more random the data, the more entropy it will have. Accordingly, an entropy threshold is set to detect deviations (changes) in packet behavior. By comparing the true value of packet entropy and the threshold value, if the entropy value of the packets is not the same as the threshold value, an attack is made or changes are made to the data disorder. A change or deviation in entropy to detect traffic is an attack on address links. Here, the entropy value according to the time window w is calculated. Assuming that the number of packets exchanged in the time window w is N; The entropy is then calculated according to equation 10.

$$H_w = -\frac{1}{N}\sum_{c} n_i \log_2(\frac{n_i}{N}) \tag{10}$$

Where, H_w entropy value for time window w and n_i number of URL link repeats the i-th is along the window.

To consider the entropy parameter, the difference between two entropy values in two consecutive time windows is used. Therefore, if the rate of change in entropy difference in two consecutive time windows is greater than the constant gamma value, the traffic is detected as a DDoS attack. Therefore;

$$\begin{matrix} \Delta H_1 = H_{K+1} - H_K \\ \Delta H_2 = H_{K+2} - H_{K+1} \end{matrix} \quad \begin{cases} If \left(\gamma_{min} < \dfrac{\Delta H_1}{\Delta H_2} < \gamma_{max}\right) & Normal \\ If \left(\dfrac{\Delta H_1}{\Delta H_2} \geq \gamma_{max}\right) \; or \; \left(\dfrac{\Delta H_1}{\Delta H_2} \leq \gamma_{min}\right) & DDoS\ Attack \end{cases} \tag{11}$$

4. Results and discussion

In this section, extensive experiments are presented to evaluate and compare the proposed method. The simulations were performed with MATLAB 2019a software and the tests were performed by a PC with a 7-Core Intel processor, 3.2 GHz frequency and 16 GB of RAM. This section describes the data set and evaluation criteria and then presents the results and comparisons.

4.1 Data set

This paper uses NSL-KDD (Tavallaee *et al.*, 2009) and CICDDoS2019 (Sharafaldin *et al.*, 2019) data set records to simulate the proposed method and perform experiments. NSL-KDD is an extended version of the KDDCUP dataset and includes one normal class and four intrusion classes (Dos, R2L, U2R and Probe). This data set consists of 41 features, 22544 samples in the experimental section and 125973 samples in the training section. In addition, each type of intrusion record includes several types of sub-attacks, details of which are available (Tavallaee *et al.*, 2009). The CICDDoS2019 dataset specifically provides DDOS attacks. This dataset contains 80 features and 23,000 records that provide 12 types of DDoS attacks in the training section and 7 attacks in the test section.

4.2 Evaluation criteria

One of the most important parts of any categorization model is its performance analysis, where this is done by evaluation criteria. Evaluation criteria to analyze the performance of a classification model, they use actual class labels and predictive class labels. The occurrence of different states of these two types of labels is indicated by the terms FN, TN, FP and TP. The occurrence of these states in a model with two classes can be shown as an amorphous matrix according to Table 1.

Table 1. Confusion matrix for a two class classifier model

Actual class	**Attack prediction class**	**Normal prediction class**
Attack record	True Positive (TP)	False Negative (FN)
Normal record	False Positive (FP)	True Negative (TN)

Here, according to Table 2, the number of selected characteristics, accuracy, correctness, recall and f-measure are used as evaluation criteria.

Table 2. Evaluation criteria

Criterion	Description	Relation
Accuracy	Ratio of true positive and false negative samples to the total samples of the test set	$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$
Precision	The ratio of true positive samples to the total positively predicted samples	$Precision = \frac{TP}{TP + FP}$
Recall	Ratio of real positive samples to total existing real samples	$Recall = \frac{TP}{TP + FN}$
F-measure	The harmonic mean considers two criteria of Precision and recall.	$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$

4.3 Results and comparisons

In this section, all results are reported based on the 10-Fold validation technique on the NSL-KDD and CICDDoS2019 datasets, so the comparisons are performed under the same conditions. Using the variable length feature selection technique in the PSO algorithm, in addition to selecting the effective features, also provides the optimal number of these features. Figure 4 shows the accuracy of the proposed method with different number of features. Here is the best accuracy measure for each number of different features. The results show the best classification accuracy of 98.65% with 11 features for the NSL-KDD dataset and 80.52% with 43 features for the CICDDoS2019 dataset.

There are several types of intrusions in the NSL-KDD database, and each type includes several types of attacks. Here the performance of the proposed method for each type of intrusion is reported. Figure 5 shows the results for Dos, R2L, U2R and Probe intrusions, respectively. Here, the results are presented based on the total number of records and the number of correct records detected in the experimental section for each subclass.

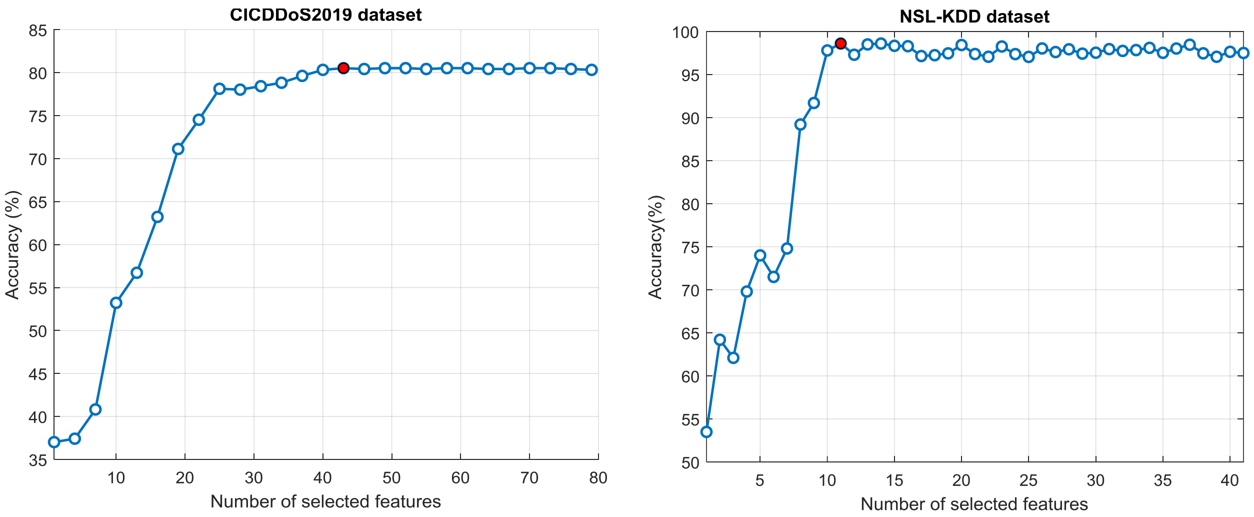

Figure 4. Accuracy of the proposed method with a number of different features

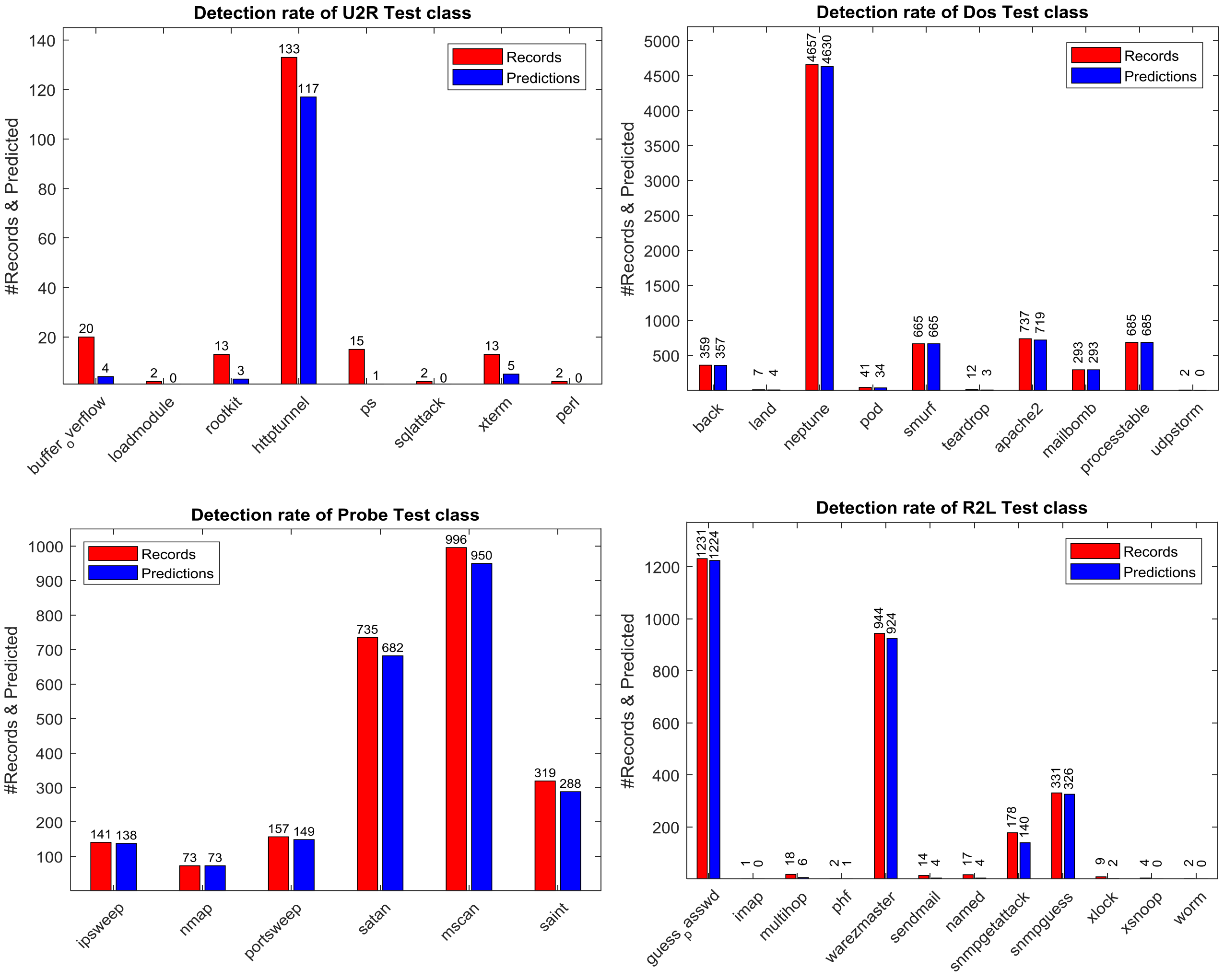

Figure 5. Prediction results for intrusion subclasses from the NSL-KDD dataset

In the following, the results of the proposed method on the NSL-KDD dataset are compared with AGA (Ghalehgolabi *et al.*, 2017) E-SVM (Yang, 2019) and AE-DNN (Bhardwaj *et al.*, 2020) algorithms. This comparison is performed in Table 3 based on the criteria of accuracy, correctness, recall and f-measure. The results show the superiority of the proposed method with 99.84% detection accuracy over AGA and E-SVM algorithms in most criteria. In general, the proposed method is about 0.5% superior to AGA in terms of accuracy. This advantage of the f-measure criterion over the E-SVM is about 7%. In addition, the detection accuracy of the proposed method is 1.4% superior to AE-DNN.

Table 3. Comparison of the proposed method with similar algorithms based on NSL-KDD dataset

Criterion	Methods	Normal	Dos	R2L	U2R	Probe	Average
Accuracy	AGA	99.82	99.85	99.64	98.68	99.77	99.81
	E-SVM	-	-	-	-	-	-
	AE-DNN	-	-	-	-	-	98.43
	Proposed method	99.79	99.93	99.50	98.24	99.80	99.84
Precision	AGA	99.87	99.97	99.42	97.58	99.71	-
	E-SVM	99.63	99.76	90.55	76.85	98.32	92.34
	AE-DNN	-	-	-	-	-	99.22
	Proposed method	99.87	99.81	99.39	96.87	99.78	99.80
Recall	AGA	99.77	99.73	99.86	99.83	99.82	-
	E-SVM	99.34	100.0	91.30	79.38	99.12	94.05

	AE-DNN	-	-	-	-	-	97.12
	Proposed method	99.82	100.0	99.96	98.00	99.93	99.87
F-measure	AGA	99.82	99.93	99.64	98.69	99.77	-
	E-SVM	99.48	99.88	90.92	78.10	98.72	93.19
	AE-DNN	-	-	-	-	-	98.57
	Proposed method	99.48	99.90	99.67	97.43	99.85	99.83

The following are similar results to the proposed method on the CICDDoS2019 dataset. Table 4 shows the results of different evaluation criteria for the proposed method and four common machine learning algorithms (ID3 decision tree, random forest, New Biz and logistic regression) (Sharafaldin *et al.*, 2019).

Table 4. Comparison of the proposed method with machine learning algorithms on the CICDDoS2019 dataset

Algorithms	Accuracy	Precision	Recall	F-measure
ID3 decision tree	76.91	78.41	65.00	71.08
Random forest	76.25	77.32	56.25	65.12
Naive Bayes	40.05	41.87	31.54	35.98
Logistic regression	25.41	25.46	20.98	23.01
Proposed method	80.52	84.05	73.12	78.20

According to the three evaluation criteria (accuracy, recall and f-measure), the highest accuracy is related to random forest algorithms and ID3 decision tree. Also, ID3 has the best performance in terms of calling criteria. Logistic regression reported the worst overall result. According to experiments, it only takes a few minutes to train the ID3 decision tree. After random forest ID3, New Biz and logistic regression are next in terms of time, respectively. In terms of runtime and evaluation criteria, ID3 is the best machine learning algorithm with the lowest runtime and highest accuracy. However, the proposed method offers better performance with 80.52% accuracy compared to the four conventional machine learning algorithms.

5. Conclusion

Prominent features of cloud computing (such as on-demand service, resource collection, wide area network access, data storage, etc.) are used by attackers to launch a distributed service denial attack. In general, DDoS attacks in such an environment use a large amount of malicious traffic to disrupt the victim server resources. With the prominent features of cloud computing, a fast and sophisticated DDoS attack makes it easy for an attacker. Therefore, this paper focuses on identifying and analyzing DDoS attacks in the cloud computing environment. In general, the cloud computing environment is vulnerable to malicious attacks due to the use of the Internet for services. Therefore, this environment requires an intrusion detection system to deal with such attacks. This paper proposes a hybrid system for detecting this type of attack, consisting of entropy and particle swarm optimization to improve detection accuracy in a cloud computing environment.

The proposed method is compared with existing algorithms based on NSL-KDD and CICDDoS2019 datasets. The results show that the proposed method can detect attacks with higher detection accuracy than existing techniques. This paper encourages security researchers to develop effective solutions for preventing, detecting, and securing DDoS attacks in the cloud. Future work discusses how attackers can take advantage of prominent cloud computing features to launch various DDoS attacks.

References

Agrawal, N. and Tapaswi, S. (2017), Published. "A lightweight approach to detect the low/high rate IP spoofed cloud DDoS attacks". 2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2), 2017. IEEE, 118-123.

Agrawal, N., Tapaswi, S. J. I. C. S. and Tutorials (2019), "Defense mechanisms against DDoS attacks in a cloud computing environment: State-of-the-art and research challenges", Vol. 21, No. 4, pp. 3769-3795.

Agrawal, N., Tapaswi, S. J. J. O. N. and Management, S. (2021), "An SDN-Assisted Defense Mechanism for the Shrew DDoS Attack in a Cloud Computing Environment", Vol. 29, No. 2, pp. 1-28.

Bamakan, S. M. H., Wang, H., Yingjie, T. and Shi, Y. J. N. (2016), "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization", Vol. 199, No., pp. 90-102.

Bawa, P. S., Rehman, S. U. and Manickam, S. J. I. J. a. C. S. A. (2017), "Enhanced mechanism to detect and mitigate economic denial of sustainability (EDoS) attack in cloud computing environments", Vol. 8, No. 9, pp. 51-58.

Bhardwaj, A., Mangat, V., Vig, R., Halder, S. and Conti, M. J. C. S. R. (2021), "Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions", Vol. 39, No., pp. 100332.

Bhardwaj, A., Mangat, V. and Vig, R. J. I. A. (2020), "Hyperband Tuned deep neural network with well posed stacked sparse AutoEncoder for detection of DDoS attacks in cloud", Vol. 8, No., pp. 181916-181929.

El-Sofany, H. F. J. I. J. O. I. E. and Systems (2020), "A New Cybersecurity Approach for Protecting Cloud Services against DDoS Attacks", Vol. 13, No. 2, pp. 205-215.

Ghalehgolabi, M., Rezaeipanah, A. J. I. J. O. C. a. T. and Research (2017), "Intrusion detection system using genetic algorithm and data mining techniques based on the reduction", Vol. 6, No. 11, pp. 461-466.

Ingre, B. and Yadav, A. (2015), Published. "Performance analysis of NSL-KDD dataset using ANN". 2015 international conference on signal processing and communication engineering systems, 2015. IEEE, 92-96.

Kanakarajan, N. K. and Muniasamy, K. (2016), Published. "Improving the accuracy of intrusion detection using gar-forest with feature selection". Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015, 2016. Springer, 539-547.

Kholidy, H. a. J. F. G. C. S. (2021), "Detecting impersonation attacks in cloud computing environments using a centric user profiling approach", Vol. 117, No., pp. 299-320.

Osanaiye, O., Choo, K.-K. R., Dlodlo, M. J. J. O. N. and Applications, C. (2016), "Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework", Vol. 67, No., pp. 147-165.

Praseed, A., Thilagam, P. S. J. I. C. S. and Tutorials (2018), "DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications", Vol. 21, No. 1, pp. 661-685.

Rastegari, S., Hingston, P. and Lam, C.-P. J. a. S. C. (2015), "Evolving statistical rulesets for network intrusion detection", Vol. 33, No., pp. 348-359.

Rezaeipanah, A., Mojarad, M. and Sechin Matoori, S. J. J. O. B. D. S. R. (2021), "Intrusion Detection in Computer Networks Through Combining Particle Swarm Optimization and Decision Tree Algorithms", Vol. 1, No. 1, pp. 14-22.

Saied, A., Overill, R. E. and Radzik, T. J. N. (2016), "Detection of known and unknown DDoS attacks using Artificial Neural Networks", Vol. 172, No., pp. 385-393.

Saisindhutheja, R. and Shyam, G. K. J. a. S. C. (2021), "An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment", Vol. 100, No., pp. 106997.

Shah, S. Q. A., Khan, F. Z. and Ahmad, M. J. C. N. (2021), "The impact and mitigation of ICMP based economic denial of sustainability attack in cloud computing environment using software defined network", Vol. 187, No., pp. 107825.

Shameli-Sendi, A., Pourzandi, M., Fekih-Ahmed, M., Cheriet, M. J. J. O. N. and Applications, C. (2015), "Taxonomy of distributed denial of service mitigation approaches for cloud computing", Vol. 58, No., pp. 165-179.

Sharafaldin, I., Lashkari, A. H., Hakak, S. and Ghorbani, A. A. (2019), Published. "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy". 2019 International Carnahan Conference on Security Technology (ICCST), 2019. IEEE, 1-8.

Sharma, A., Agrawal, C., Singh, A. and Kumar, K. (2020), Real-time DDoS detection based on entropy using Hadoop framework. *Computing in Engineering and Technology.* Springer.

Shidaganti, G. I., Inamdar, A. S., Rai, S. V., Rajeev, A. M. J. I. J. O. C. A. and Computing (2020), "Scef: a model for prevention of ddos attacks from the cloud", Vol. 10, No. 3, pp. 67-80.

Somani, G., Gaur, M. S., Sanghi, D., Conti, M. and Buyya, R. J. C. C. (2017a), "DDoS attacks in cloud computing: Issues, taxonomy, and future directions", Vol. 107, No., pp. 30-48.

Somani, G., Gaur, M. S., Sanghi, D., Conti, M., Rajarajan, M. and Buyya, R. J. I. C. C. (2017b), "Combating DDoS attacks in the cloud: requirements, trends, and future directions", Vol. 4, No. 1, pp. 22-32.

Subashini, S., Kavitha, V. J. J. O. N. and Applications, C. (2011), "A survey on security issues in service delivery models of cloud computing", Vol. 34, No. 1, pp. 1-11.

Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A. A. (2009), Published. "A detailed analysis of the KDD CUP 99 data set". 2009 IEEE symposium on computational intelligence for security and defense applications, 2009. Ieee, 1-6.

Yang, C. J. C. C. (2019), "Anomaly network traffic detection algorithm based on information entropy measurement under the cloud computing environment", Vol. 22, No. 4, pp. 8309-8317.

Zargar, S. T., Joshi, J., Tipper, D. J. I. C. S. and Tutorials (2013), "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks", Vol. 15, No. 4, pp. 2046-2069.