SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I4P114

Volume 12 Issue 4, 148-158, April 2025

Original Article

Intelligent Detection of Distributed Denial-of-Service Attacks in Cloud Platforms Using the Osprey Optimized Dense-SEQNET Architecture

Dinesh Kumar Budagam

Sr Cybersecurity Engineer at VISA, Inc., Foster City, CA, USA.

Corresponding Author : dbudagam@gmail.com

Received: 10 February 2025 Revised: 12 March 2025 Accepted: 14 April 2025 Published: 29 April 2025

Abstract *- Cloud Computing platforms are one among the most commonly used internet-based applications. Adapting to cloud computing leads to reduced expenses for maintaining and managing internet applications. However, with its growing popularity, these platforms are prone to several attack types, including Denial Of Service (DOS), Distributed DoS and spoofing. Among these, DDoS is a widely recongnized intrusion attack on cloud platforms causing a downfall of services or denial of services. Hence, this paper focuses on developing an intelligent identification of DDoS attacks using a Deep Learning (DL) based classification model. To improve the raw data quality, data pre-processing techniques such as data normalization, visualization, and feature encoding are deployed, further enhancing the data classification process. To achieve this, a Dense Sequential Network (Dense-SeqNeT) is utilized which analyzes the data to detect anomalies and malicious DDoS attacks with real-time detection. Additionally, for attaining augmented classification, the Osprey Optimization Algorithm (OOA) technique is integrated with Dense-SeqNeT, which enables solving complex problems with rapid convergence speed. The proposed work is validated using PYTHON software, and the attained experimental results demonstrate that the developed model performs well with higher accuracy in DDoS detection and reduced execution time. Therefore, the developed system ensures accurate detection and evaluates its performance.*

Keywords *- Cloud computing, DDoS, Intrusion detection, Osprey optimized, Dense-SeqNeT, Data pre-processing techniques.*

1. Introduction

Cloud computing embodies significantly increased popularity in various fields due to its drastic benefits, including cost-effectiveness, unlimited storage, increased diversity, scalability, backup and recovery [1, 2]. Cloud provides highly scalable and reliable services, which are made available in three distinct modes: private, public or hybrid cloud [3]. Generally, cloud services are categorized into Software-as-a-Service (SaaS), Platform-as-a-Service (Paas) and Infrastructure-as-a-Service (IaaS) [4]. However, the cloud faces significantly increased difficulties in protecting the data from unwanted attacks and cyber threats in public cloud infrastructure [5]; among several attacks on cloud platforms, DDoS attacks largely impact cloud computing. DDoS attack takes place using thousands of vulnerable hosts referred to as zombies [6]. Figures 1 and 2 represent the various cloud security threats and DDoS attacks on cloud platforms.

The major cause of increased DDoS attacks is data extrusion. DDoS attacks create quite intricate security threats to the cloud, making it more vulnerable and unreliable. This type of attack hacks over millions of user information and redirects the data traffic towards the attacker. Most commonly, DDoS affects the cloud servers' processing ability, bandwidth and network. DDoS exploits the cloud resources by producing malicious traffic within the cloud platform [7]. DDoS is recognized as an intrusion attack that restrains the real user from processing the data by producing data traffic and later robbing this information. As a result of this unwanted traffic, the cloud server is forced to leave apt cloud resources to users [8]. Considering the intensity of DDoS attacks, various rectification and detection approaches are innovated [9].

Significantly, DDoS detection faces certain difficulties as these attacks often tend to disguise their identity. DDoS analyzes and classifies packets, which is further characterized by a signature-based approach that utilizes different attack signatures placed within its database for intrusion detection [10]. At the same time, anomaly-based approaches evaluate the traffic characteristics pattern in a specific period for detection. Utilizing these approaches effectively detects attacks [11]. However, a larger amount of data needs to be pre-processed using various data pre-processing techniques before

the classification process to attain improved data quality, which increases detection accuracy while minimizing computational complexity [12].

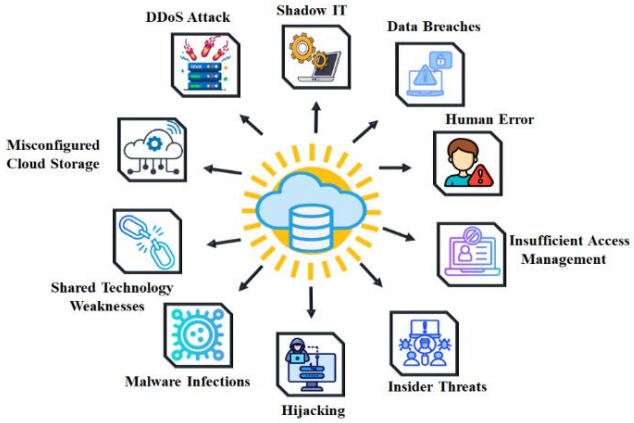


Fig. 1 Cloud security threats

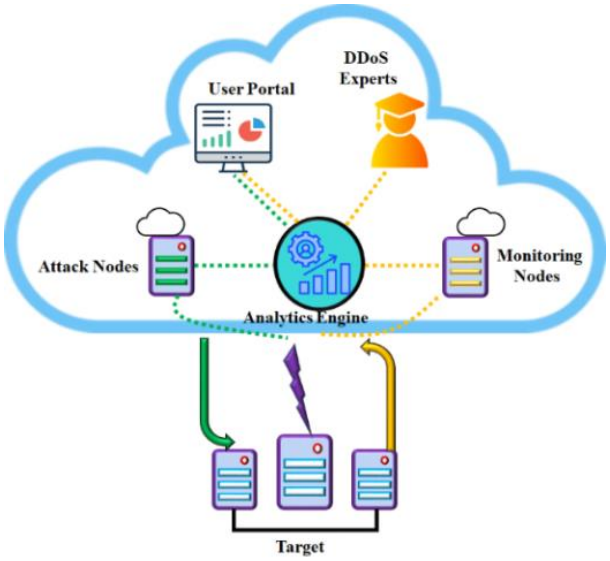


Fig. 2 Cloud-based DDoS attack

Another major aspect of DDoS detection is the selection of appropriate classification models, as they are capable of defining and distinguishing multiple objects. Several classifiers like Naïve Bayes, K-Nearest Neighbour (KNN), Support Vector Machine (SVM). In [13], the authors developed a Random Forest (RF) based classification model for DDoS intrusion detection, which enables the definition of regular network traffic and DDoS attacks. RF performs more efficiently in identifying the DDoS attacks within the network traffic, ensuring improved classification accuracy. Nevertheless, RF effectiveness depends on the quality of the data and the complexity of network traffic results in a reduced ability to deduce concealed patterns. Consecutively, authors in [14] integrated LR and SVM for classifying DDoS attacks among chosen multi-class supervised classification datasets. SVM, together with LR, attained higher classification but still faces many difficulties, such as handling large amounts of datasets and imbalanced class distribution, which leads to reduced model performance. [15] Proposes a DDoS attack detection system using hybrid DL models such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM), which is executed in two different levels using a specially trained algorithm in the first level and second level incorporates input data and output data attained from the first level, thereby, achieving high-performance model. Despite this, the hybrid DL-based approach possesses certain limitations, like high computational cost and increased complexity. Henceforth, to prevail over the shortcomings faced by the above-mentioned classifiers, this paper proposes an innovative Dense-SeqNeT classifier for achieving enhanced detection accuracy, increased scalability and improved robustness in identifying DDoS attacks among diverse and varying network traffic. Optimization algorithms play a major role in increasing the classifier performance and tuning classifier parameters further.

Utilizing the optimization algorithm enables enhanced hyper-parameter tuning, thus increasing the classification accuracy and reducing overfitting. Consider [16] the authors used Particle Swarm Optimization (PSO) to enhance the working of the Decision Tree (DT) classifier. PSO helps with the optimum selection of parameters by depicting the particle characteristics, enabling the DT classifier to achieve increased accuracy. Nonetheless, PSO has a high chance of being stuck in local optima, limiting the overall system reliability.

Significantly, in [17], Tunicate Swarm Optimization (TSA) is integrated to adjust the LSTM model's hyper-parameters for achieving improved DDoS detection. The TSA-based LSTM model shows considerable advancement in detection accuracy with enhanced precision and recall. However, this approach consumes a longer training duration with increased computational complexity. Thus, in [18], Harris Hawks Optimization (HHO) is utilized to choose the potential attributes to improve the performance efficacy of the LSTM model. The implementation of HHO enables improved selection of the most relevant attributes by replicating the hunting behaviour of Harris hawks, thus improving the overall efficiency and reliability of the system. Despite this, HHO cannot handle complex datasets.

2. Literature Survey

In [19], a machine learning-based detection technique is presented utilizing the Naive Bayes algorithm and the DDoS attacks that occur in virtual cloud computing environments. It is ideal for dynamic cloud infrastructures due to its main benefits, which include high detection speed, ease of use, low computational cost and adaptability to real-time data. However, this approach has some drawbacks, like handling complicated or unbalanced datasets with less accuracy than more sophisticated algorithms.

The vulnerability of SDN-based cloud environments to DDoS attacks is demonstrated, and a thorough detection framework for accurate and timely identification is provided in [20]. Among the benefits are high accuracy, quicker detection, and successful identification of coordinated attacks through traffic clustering and event correlation. However, drawbacks include the intricacy of combining several algorithms, possible computational overhead, and difficulties sustaining performance in real-time or high-traffic scenarios.

In [21], a sophisticated Gradient Hybrid Leader Based Optimization algorithm is designed to improve DDoS attacks in cloud computing settings. Because of its high accuracy and increased detection efficiency, the proposed approach is especially well-suited for intricate and expansive cloud systems. Improved accuracy, optimized training, and increased detection efficiency are some of the approach's benefits. However, overfitting from data augmentation and higher computational complexity could be drawbacks.

The RDAER model for early and precise detection of DDoS vulnerabilities in SDN-based cloud environments is presented [20]. In order to identify coordinated attacks, the model correlates clustered events and forecasts traffic anomalies at the switch level. High detection accuracy, quick reaction times, and efficient technique integration are advantages. However, there are disadvantages, such as the need for continuous tuning to adapt to evolving threats and computational overhead.

In [22], a Low-Rate DDoS Attack Detection Framework is suggested, which uses a Hybrid Deep Learning Approach with CNN and autoencoders, as well as a mitigation algorithm, to detect and mitigate low-rate DDoS attacks in cloud environments. Accurate detection, efficient mitigation, and maintained service availability are some benefits. However, high implementation complexity, resource consumption, and the requirement for frequent updates are among the drawbacks.

While the existing models exhibit promising results, they are often limited by high computational complexity, sensitivity to imbalanced datasets, overfitting, and suboptimal feature selection methods. Moreover, existing optimization techniques suffer from slow convergence, local optimization issues, and inefficiency in handling complex datasets**.** These limitations affect real-time DDoS attack detection and adaptive classification performance**.** Henceforth, the proposed model uses OOA to attain enhanced parameter tuning with increased convergence speed and reduced complexity to overcome these limitations. Therefore, with both Dense-SeqNeT and OOA, the overall system performs well, with increased efficiency and reliability, and the detection of DDoS attacks within the cloud platform is accurate. The major outline and contributions of the proposed system are listed as follows.

- To develop an intelligent DDoS attack detection system using DL based classifier with enhanced accuracy and reliability.
- Improve the initial data quality using data pre-processing techniques such as data normalization, data visualization, and feature encoding, thus improving the classification process.
- To deploy a Dense-SeqNeT based classifier to analyze the data for precise detection of DDoS attacks with real-time detection of anomalies and malicious activities.
- To integrate OOA to achieve enhanced hyper-parameter tuning of the Dense-SeqNeT classifier, ensuring rapid convergence and increased reliability.

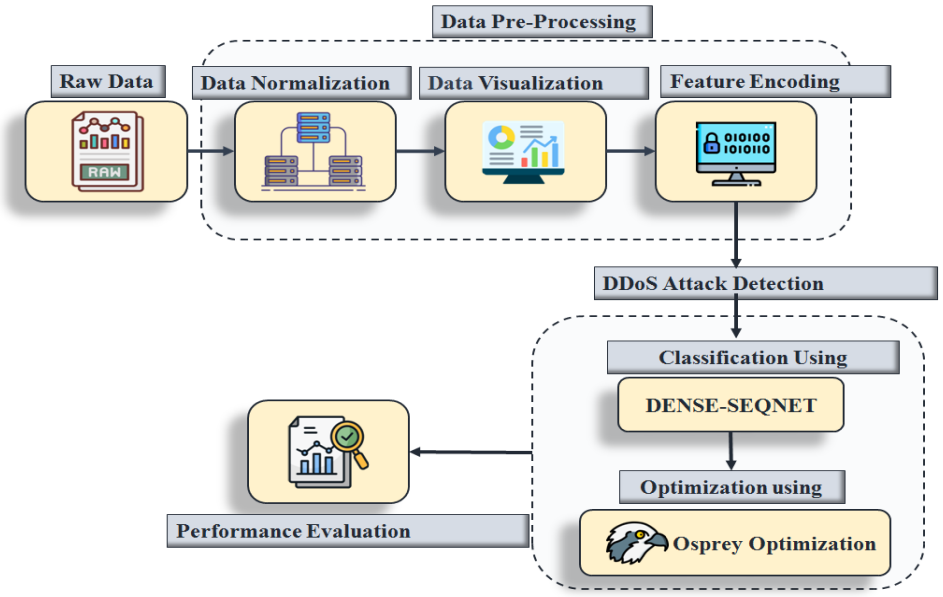


Fig. 3 Proposed system block diagram

3. Proposed Modelling

An intelligent DDoS attack detection system using an optimized DL-based classifier is represented in Figure 3, which is developed to attain increased accuracy and precision detection of DDoS attacks within the cloud platform. The system process begins with collecting raw data, which is then fed as input for further detection. The raw data containing unwanted interference needs to be removed. Thus, the input data undergoes three stages of data pre-processing techniques: data normalization, data visualization, and feature encoding. Initially, data is scaled to ensure all the data are in a similar feature range. Then, data visualization is carried out to provide understandable representations of the data and identify intricate data patterns. Later, these data are converted into numerical format, making it applicable for the classification process.

After the data is pre-processed, these data are transferred to the DDoS attack detection process, where the data is analyzed with the advanced Dense-SeqNeT classification technique. Further, to improve the classification process, Dense-SeqNeT is optimized using the Osprey Optimization technique, which enables optimum tuning of hyper-parameters and ensures increased convergence speed with enhanced reliability of DDoS attack detection. Therefore, developing a DDoS detection system using the classifier and optimization technique attains higher detection performance with increased accuracy and reliability.

3.1. Modelling of Data Pre-Processing Stages

In the data pre-processing stages, the input data are scaled, visualized and converted to improve the data quality for a better classification process. The data pre-processing stages are discussed in detail below.

3.1.1. Data Normalization

In DL, the feature scaling method is utilized to attain a normalized distribution of attributes within the dataset. Feature Scaling is considered vital as it prevents certain features from overpowering other features due to their scale differences; thus, feature scaling manages to attain similar feature scale ranges, as illustrated in Figure 4. This leads to improved performance of the classifier.

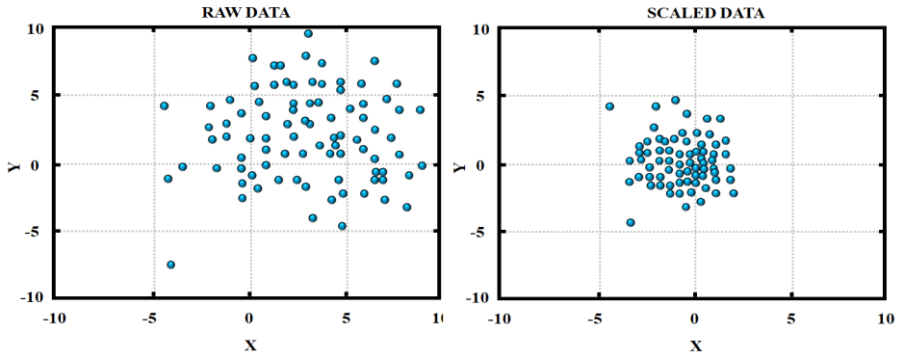


Fig. 4 Data normalization

The Min-Max normalization approach is widely utilized to perform a feature scaling process in which the values are converted so that the mean of their attribute becomes zero and their standard deviations become 1, respectively. This normalized distribution enables easy interpretation of the relationships between the variables. The data Min-Max values are attained using,

$$X_{normalization} = \frac{x - x_{min}}{x_{\max} - x_{min}} \quad (1)$$

Thus, using data normalization effectively functions to put all the feature attributes of the data in similar feature ranges, hence leading to optimal performance.

3.1.2. Data Visualization

Data visualization represents the data in a visual format, making it easier to interpret and understand complex and intricate data patterns for enhanced detection of anomalies.

3.1.3. Feature Encoding

Feature encoding is considered a vital part of the data pre-processing stages as it converts features or non-numerical data into numerical forms, making it suitable for DL-based classifiers. The data that has undergone pre-processing needs to be further classified to analyze DDoS attacks; hence, the Dense-SeqNeT classifier is utilized.

3.2. Modelling of Dense-SeqNeT Classifier

The Dense-SeqNeT classifier incorporates DenseNet and SeqNet architecture as illustrated in Figure 5, with its key advantages to develop an intelligent model for detecting DDoS attacks within the cloud platform.

The proposed classifier operates in sequential stages like the input layer, DenseNet module for feature extraction, SeqNet module for temporal analysis, Fully-Connected layer and lastly, the output layer.

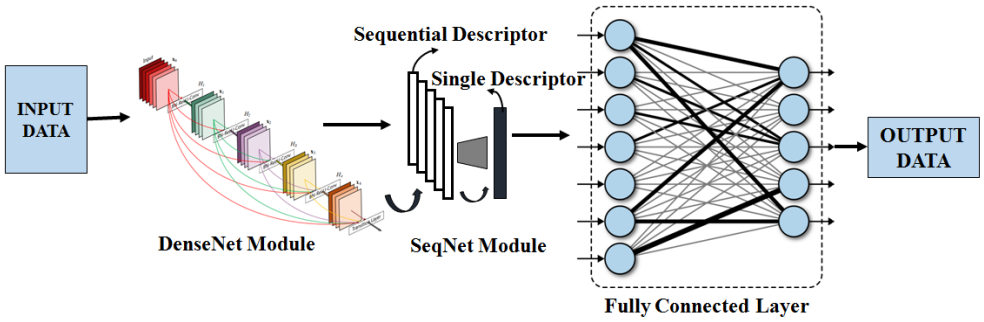


Fig. 5 Architecture of Dense-SeqNeTclassifier

3.2.1. Input Layer

The raw network traffic data is fed as input, containing packet-level data obtained from the pre-processed dataset. This input further undergoes normalization and feature encoding to improve the classification process.

3.2.2. DenseNet Module

The DenseNet module acts as the backbone of the developed architecture as it extracts a higher level of spatial features within the input, thus minimizing data redundancy and assuring efficient data propagation by reusing the features within the network using its densely connected layers. By conserving these spatial features, DenseNet can identify complex relations within the data traffic. DenseNet produces feature maps using the densely connected layers where each layer receives the input from all the previous layers.

$$H_l = g([H_0, H_1, \dots, H_{l-1}]) \quad (2)$$

Where, H_l denotes the output from the $l\,th$ layer, g indicates the normalization composite function, and $H_0, H_1, \dots, H_{l-1}$ represents the sequence of feature maps from the previous layers. To further reduce computational complexity, DenseNet utilizes bottleneck layers with 1×1 convolutional before 3×3 which is expressed as,

$$H_l = g\,(\,Conv\,1 \times 1\,\left(g\big(Conv\,3 \times 3(H_{l-1})\big)\right)) \quad (3)$$

Lastly, Transition layers are used to minimize the size of the feature maps, given by,

$$H_{transition} = g(\,Conv\,1 \times 1\,(H_l)) \quad (4)$$

3.2.3. SeqNet Module

The SeqNet is developed to analyze temporal or sequential data, enabling the understanding of time-dependent network traffic patterns. SeqNet utilizes convolutional layers to capture temporal dependencies, which makes it more efficient. This classifier sequentially processes the data, thus making it easier to detect anomalies in network traffic with increased efficiency. SeqNet classifier functions by encoding the input sequence into a set of hidden conditions utilizing a series of layers, which is represented as,

$$h_t = f\,(W_h h_{t-1} + W_x x_t + b_h \quad (5)$$

Where, h_t indicates the hidden condition at time t , W_h and W_x denotes the matrix weight, x_t represents the input, b_h implies the bias term, and f refers to the activation function. After processing the sequence, the output layer is used for making predictions classifying the sequence or predicting the next value, which is determined as,

$$y = softmax(W_o h_T + b_0) \quad (6)$$

Where, h_T indicates the final hidden condition, W_o implies the output matrix weight and b_0 is the output bias. Later, by training this model, the weights are updated, reducing errors in the predicted and the actual output.

3.2.4. Fully Connected Layer

The output obtained from SeqNet is fed into the fully connected layer to combine the extracted spatial and temporal features, which acts as the final decision-making stage. This layer processes the data to create predictions, and the final output layer classifies the network traffic into two types including ‘normal’ or ‘DDoS’.In addition, to enhance the detection accuracy and reduce the execution time, the classifier is optimized using the OOA, which enables better tuning of the hyperparameters.

3.3. Modelling of OOA

OOA is developed based on the hunting behavior of Osprey, which is also known as the fish, river and sea hawk, a non-nocturnal fish-eating bird. The osprey’s tactics to catch fish and carry the fish to the optimum position for eating is considered an intelligent characteristics of the bird, which inspired the development of OOA.

The modelling of OOA starts with initialization, then updating Osprey’s position based on two phases namely exploration and exploitation phases.

3.3.1. Initialization

OOA functions based on the population strategy, which provides the finest solutions based on the search power of its members via a repetition process. Each osprey is regarded as a member of the OOA population that allocates values for the problem variables based on their position within the search space. Hence, each osprey is considered as the solution for the problem. The initial position of osprey is represented as,

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \quad (7)$$

$$x_{i,j} = lb_j + r_{i,j}\,.\,(ub_j - lb_j), i = 1,2,\ldots\ldots\ldots,N\,, j = 1,2,\ldots\ldots\ldots m, \quad (8)$$

Where X denotes the matrix population for the position of osprey's, x_i indicate the $j\,th$ osprey, $x_{i,j}$ implies its $j\,th$ dimension, N represents the number of osprey's, m refers to the number of problem variables, $r_{i,j}$ denotes a random number between the range of [0,1], lb_j and $r_{i,j}$ indicates the lower bound and upper bound. The objective function is determined using,

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F\,(X_1) \\ \vdots \\ F\,(X_i\,) \\ \vdots \\ F\,(X_N) \end{bmatrix}_{N\times 1} \quad (9)$$

Where F denotes the objective function vector values and F_i indicates the $i\,th$ osprey's attained objective function value. The depicted objective function value represents the efficiency of the member solutions. Henceforth, the optimum solution refers to the best objective function value corresponding to the best member of the population and vice versa. Along with the updated osprey position in the search space, the best member solution must also be updated for each iteration.

Phase 1: Exploration (Selecting the Best Position)

Ospreys are powerful hunter capable of locating fish even underwater because of their great eyesight. Subsequent to finding the better position of the fish, they attack and catch the prey by diving underwater. The fish set for each osprey is depicted as,

$$FP_i = \{X_k \mid k \in \{1,2,\ldots.,N\} \wedge F_k < F_i\} \cup \{X_{best}\} \quad (10)$$

Where, FP_i indicates the location for the set of fishes for $i\,th$ osprey and X_{best} denotes the best member solution. Osprey randomly selects one among these fishes and hunts it; according to this, the position of the osprey is further updated, replacing the previous objective function values.

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j}\,.\,(SF_{i,j} - I_{i,j}\,.\,x_{i,j}), \quad (11)$$

$$x_{i,j}^{P1} = \begin{cases} x_{i,j}^{P1},\ lb_j \le x_{i,j}^{P1} \le ub_j\,; \\ lb_j, x_{i,j}^{P1} < lb_j; \\ ub_j, x_{i,j}^{P1} > ub_j. \end{cases} \quad (12)$$

$$X_i = \begin{cases} X_i^{P1}, F_i^{P1} < F_i; \\ X_i,\ else, \end{cases} \quad (13)$$

Where, X_i^{P1} indicates the updated $i\,th$ osprey position, $x_{i,j}^{P1}$ represents its $j\,th$ dimension, F_i^{P1} implies its objective function value, SF_i the chosen fish and$SF_{i,j}$ denotes its $j\,th$ dimension. Figure 6 showcases the flow chart for the proposed OOA.

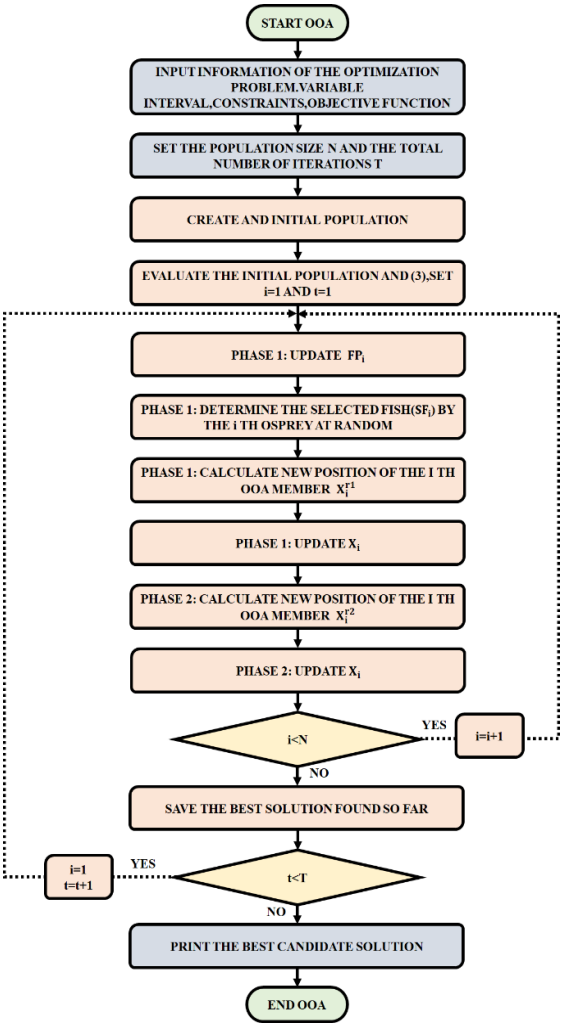


Fig. 6 OOA flow chart

Phase 2: (Taking the fish to the best location)

Following hunting the process, the osprey takes the fish to a better place to eat it, which in turn produces alterations in the osprey's position. This process leads to improved exploitation strength for OOA during local search and convergence towards optimum solutions. Therefore, integrating both advanced classifier and optimization techniques assures accurate and precise detection of DDoS within the cloud platform.

4. Results and Discussion

The proposed model is implemented using Python Software to analyze and determine the performance efficiency and accuracy of the OOA-optimized Dense-SeqNeT classifier. The implementation is carried out using the CIC-DDoS2019 dataset with 116870 samples split into 80% and 20% for training (93496) and testing (23374). The dataset undergoes a pre-processing phase, including data normalization and encoding techniques to enhance classification efficiency. During the training phase, the Dense-SeqNeT model is trained using extracted features to identify malicious and benign traffic patterns. The optimization of hyperparameters is performed using the OOA to enhance model convergence and accuracy. The network is trained with batch size 64, an adaptive learning rate, and categorical cross-entropy losses function to optimize classification performance. The trained model is evaluated using unseen data to assess its generalization capability for testing. Additionally, the results obtained and comparative analysis are discussed and elaborated on in the section below.

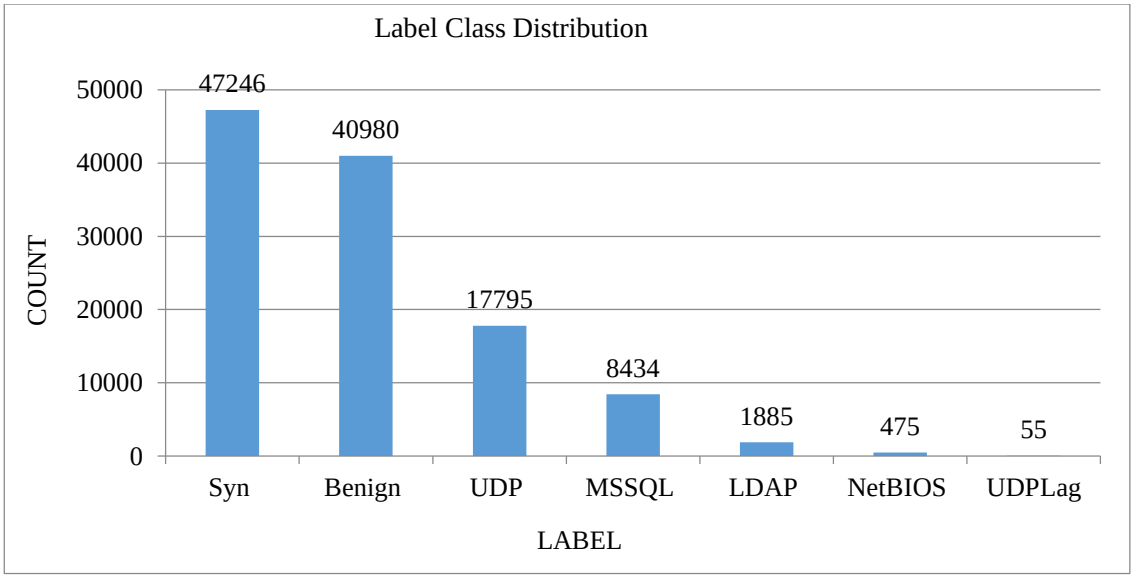


Fig. 7 Class label distribution

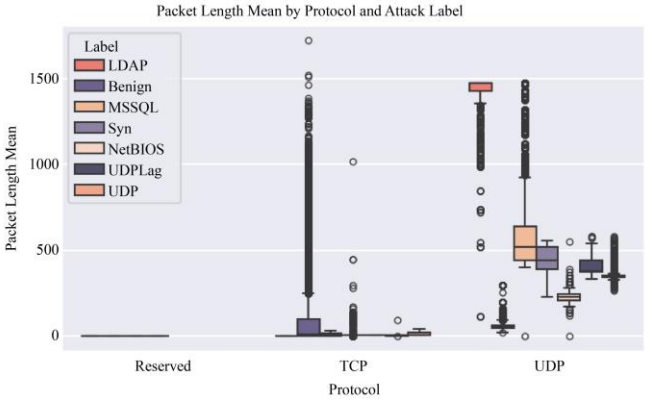


Fig. 8 PLM by protocol and attack label

Figure 7 represents the label class distribution chart displaying the counts of various label classes within the dataset. In this graph, the x-axis shows the label classes such as Syn, Benign, UDP, MSSQL, LDAP, NetBIOS and UDPLag, whereas the y-axis showcases their corresponding counts. Syn contains the highest count of 47,246, followed by Benign with 40,980 counts, UDP with 17,795, and MSSQL with 8,434. In contrast, the other label classes have comparatively less counts of LDAP with 1,885, NetBIOS with 475 and UDPLag with 55, respectively.

Figure 8 indicates the boxplot representing the PLM by Protocol and Attack Label, which analyzes the packet length distributions among various protocols and attack labels. Here, the x-axis indicates the protocol, such as Reserved, TCP and UDP, while the y-axis indicates the PLM value. Each protocol is divided into attack labels like LDAP, Benign, MSSQL, Syn, NetBIOS, UDPLag and UDP. From the graph, it is notable that reserved attained reduced PLM; for TCP protocol, the PLM value is quite high, specifically in Syn and Benign attack labels, with visible outliers exceeding beyond 1,500. Finally, the UDP protocol possesses the highest PLM value, especially in UDP and LDAP attack labels.

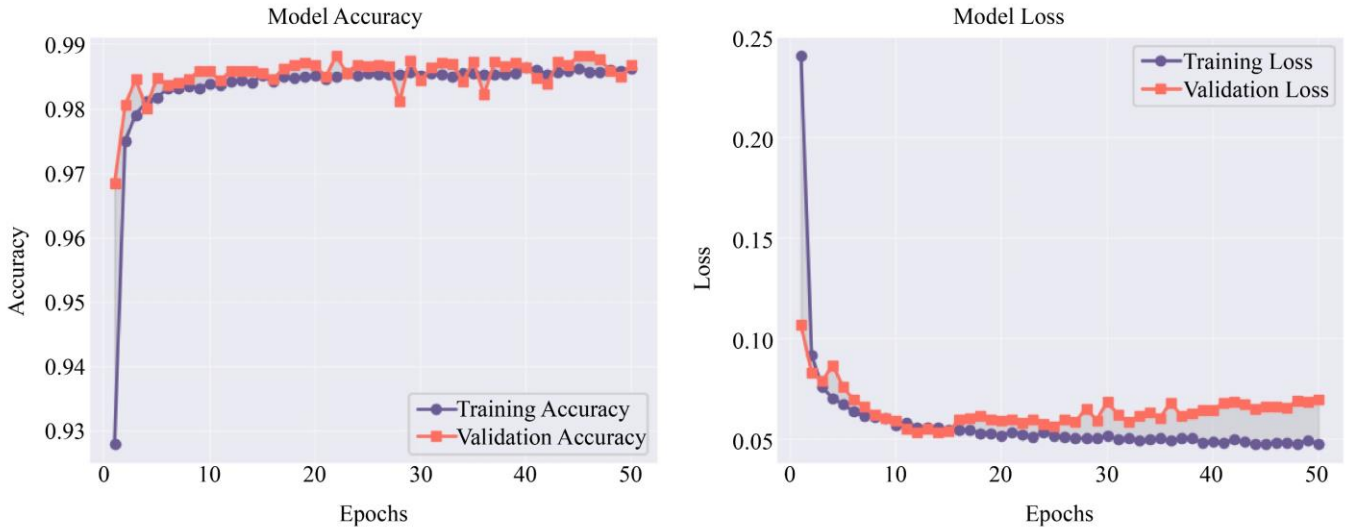


Fig. 9 Model accuracy and loss

Figure 9 showcases the model accuracy and loss for training and testing datasets. Understandably, the proposed classifier attained a higher accuracy of 98.69% for testing and 98.54% for the training dataset, indicating accurate DDoS attack detection and the efficiency of the Dense-SeqNeT classifier. At the same time, the second graph showcases the model loss for training and testing datasets, where the proposed classifier obtained reduced losses for both training and testing datasets of 0.06 and 0.05, which is significantly low, thus depicting the effectiveness of the Dense-SeqNeT classifier.

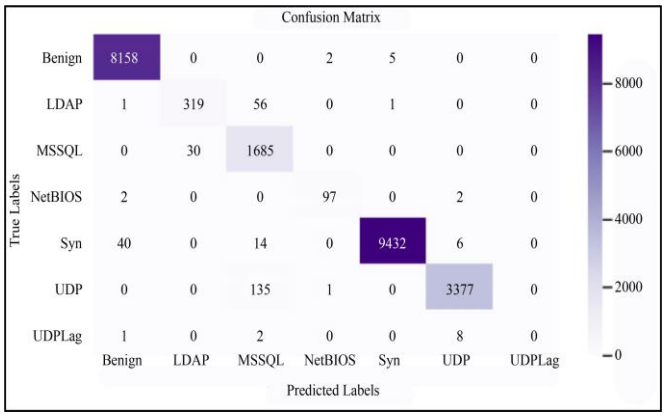


Fig. 10 Confusion matrix

Figure 10 displays the confusion matrix for various attack labels. Here, the x-axis represents the predicted labels, while the y-axis depicts the true labels.

From the matrix, about 8158 datasets are effectively classified as Benign labels, around 1685 are identified as MSSQL labels, 9432 shows Syn, and 3377 labels show UDP attack labels, indicating the effectiveness of the Dense-SeqNeT for classifying datasets based on various attack labels.

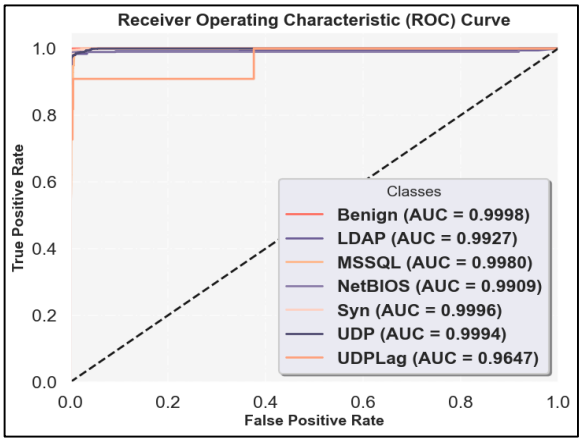


Fig. 11 ROC curve

Figure 11 denotes the ROC curve for evaluating the performance efficacy of the proposed classifier among various attack labels. The x-axis represents the False Positive Rate (FPR), and the y-axis represents the True Positive Rate (TPR). Different classes are displayed along with their respective Area Under Curve (AUC) values. Among these, Benign possesses an AUC of 0.9998, Syn and UDP show 0.9996 and 0.9994 AUC, and LDAP, MSSQL, and NetBIOS showcase 0.9927, 0.9980 and 0.9909 AUC values. Meanwhile, the AUC value attained by UDPLag is the lowest at 0.9647. Table 1 depicts the result obtained by the proposed classifier.

Table 1. Performance matrices of proposed classifier

Accuracy	0.9869
Precision	0.9871
Recall	0.9869
F1-Score	0.9868
Specificity	0.9869
MCC	0.9810

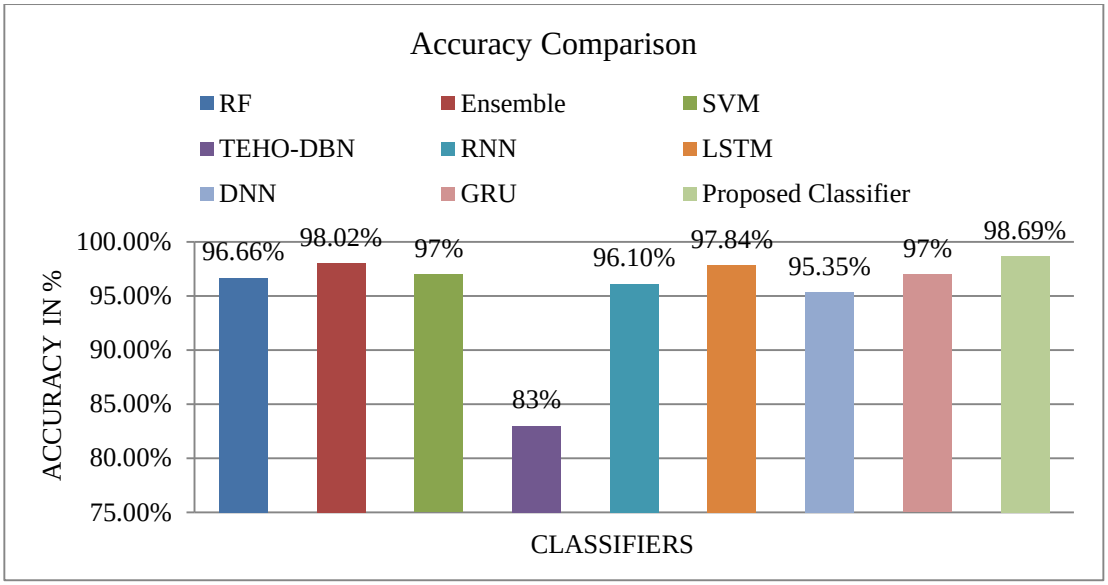
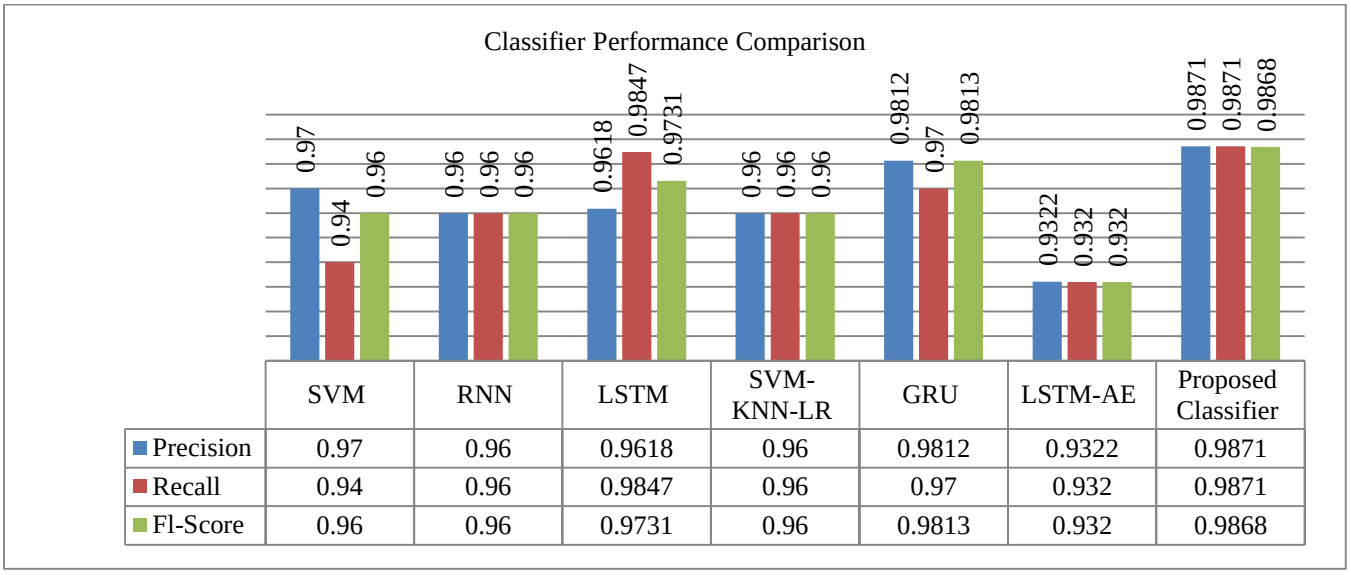


Fig. 12 Accuracy comparison

Figure 12 displays the accuracy comparison between various conventional classifiers with the proposed classifier for analyzing the performance effectiveness of the proposed model. The conventional classifiers such as RF [1], Ensemble [4], SVM [6], TEHO-DBN [7], RNN [12], LSTM [15], DNN [23] and GRU [24] are compared with Dense-SeqNeT classifier in terms of their accuracy percentage. The above graph showcases that the proposed classifier attained a higher accuracy of 98.69% among all the other classifiers, referring to the enhanced system functioning with precise detection of DDoS attacks within the cloud platform.



	SVM	RNN	LSTM	SVM-KNN-LR	GRU	LSTM-AE	Proposed Classifier
Precision	0.97	0.96	0.9618	0.96	0.9812	0.9322	0.9871
Recall	0.94	0.96	0.9847	0.96	0.97	0.932	0.9871
Fl-Score	0.96	0.96	0.9731	0.96	0.9813	0.932	0.9868

Fig. 13 Comparison of performance matrices

Figure 13 showcases the Precision, Recall and F1-Score attained by conventional classifiers with the proposed classifier. The performance matrix obtained by the proposed classifier is slightly higher than that of the other classifiers, with 0.9871 precision, 0.9871 recall and 0.9868 F1-Score, indicating that the proposed classifier performs well, thus leading to efficient system performance in detecting the DDoS attacks precisely.

Figure 14 indicates the specificity values attained by DNN [23], Ensemble [4], and the proposed classifier to determine the effectiveness of Dense-SeqNeT. The proposed classifier attains a higher specificity of 0.9869 when compared to DNN, which attained 0.969, and Ensemble depicts 0.9865, which implies the performance efficacy of the proposed model.

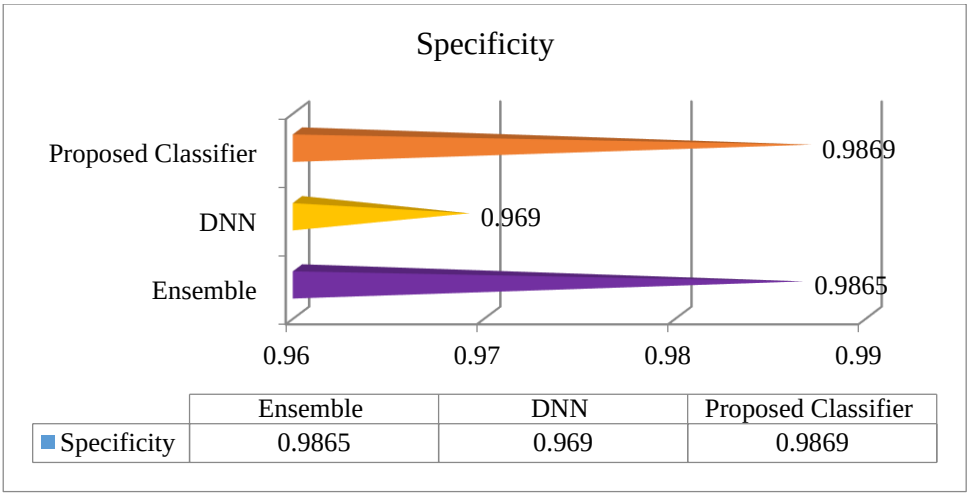


Fig. 14 Specificity comparison

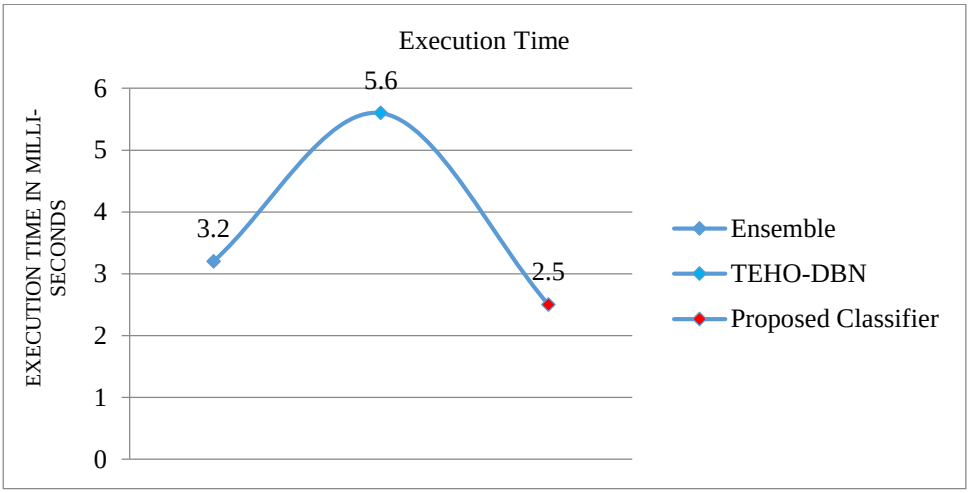


Fig. 15 Execution time comparison

Figure 15 refers to the execution time exhibited by TEHO-DBN [7], Ensemble [4] and Dense-SeqNeT classifier. From the above-displayed graph, it is visible that the proposed model depicted reduced execution time that TEHO-DBN and Ensemble classifiers, thereby achieving reduced execution time with increased real-time detection of DDoS, implying improved system performance.

5. Conclusion

The developed system for intelligent detection of DDoS attacks within a cloud platform designed using Dense-SeqNeT classifier and OOA effectively performs well with higher accuracy prediction. The utilization of data pre-processing techniques assures enhanced data quality. Also, integrating Dense-Net and SeqNet enables enhanced spatial feature extraction with better acquisition of intricate temporal patterns, thereby ensuring highly efficient and precise detection of anomalies in network traffic. Furthermore, deploying OOA effectively enhanced the detection accuracy with increased convergence and reduced execution time, thus making it highly reliable. Therefore, together with both advanced Dense-SeqNeT classifier and OOA, it assures highly improved DDoS detection with increased accuracy. The developed model is validated through Pyhton Software, and the results determine that usage of the advanced classifier with OOA attains higher accuracy 0.9869, precision 0.9871, recall 0.9869 and F1-Score 0.9868 with increased convergence speed, thereby attaining highly efficient and reliable system for improving the security and data privacy within the cloud platform.

References

[1] Muhammad Aamir, and Syed Mustafa Ali Zaidi, "Clustering Based Semi-Supervised Machine Learning for DDoS Attack Classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 4, pp. 436-446, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Karan B. Virupakshar et al., "Distributed Denial of Service (DDoS) Attacks Detection System for OpenStack-Based Private Cloud," *Procedia Computer Science*, vol. 167, pp. 2297-2307, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Sharmin Aktar, and Abdullah Yasin Nur, "Towards DDoS Attack Detection Using Deep Learning Approach," *Computers & Security*, vol. 129, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Ahmed Abdullah Alqarni, "Majority Vote-Based Ensemble Approach for Distributed Denial of Service Attack Detection in Cloud Computing," *Journal of Cyber Security and Mobility*, pp. 265-278, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Thapanarath Khempetch, and Pongpisit Wuttidittachotti, "DDoS Attack Detection Using Deep Learning," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 382-388, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Mohammad Arafat Ullah et al., "Detecting Distributed Denial of Service Attacks Using Logistic Regression and SVM Methods," *arXiv*, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] S. Velliangiri, P. Karthikeyan, and V. Vinoth Kumar, "Detection of Distributed Denial of Service Attack in Cloud Computing Using the Optimization-Based Deep Networks," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 33, no. 3, pp. 405-424, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Saikat Das et al., "Ensembling Supervised and Unsupervised Machine Learning Algorithms for Detecting Distributed Denial of Service Attacks," *Algorithms*, vol. 17, no. 3, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Amin Sadiq et al., "Detection of Denial of Service Attack in Cloud Based Kubernetes Using eBPF," *Applied Sciences*, vol. 13, no. 8, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ahmed Ahmim et al., "Distributed Denial of Service Attack Detection for the Internet of Things Using Hybrid Deep Learning Model," *IEEE Access*, vol. 11, pp. 119862-119875, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mohamed Amine Ferrag et al., "Deep Learning-Based Intrusion Detection for Distributed Denial of Service Attack in Agriculture 4.0," *Electronics*, vol. 10, no. 11, pp. 1-26, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Furqan Rustam et al., "Denial of Service Attack Classification Using Machine Learning with Multi-Features," *Electronics*, vol. 11, no. 22, pp. 1-20, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] MohammadMoein Shafi et al., "Toward Generating a New Cloud-Based Distributed Denial of Service (DDoS) Dataset and Cloud Intrusion Traffic Characterization," *Information*, vol. 15, no. 4, pp. 1-27, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Ahamed Aljuhani, "Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments," *IEEE Access*, vol. 9, pp. 42236-42264, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Surya Pavan Kumar Gudla et al., "DI-ADS: A Deep Intelligent Distributed Denial of Service Attack Detection Scheme for Fog-Based IoT Applications," *Mathematical Problems in Engineering*, vol. 2022, no. 1, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Aween Abubakr Saeed, and Noor Ghazi Mohammed Jameel, "Intelligent Feature Selection Using Particle Swarm Optimization Algorithm with a Decision Tree for DDoS Attack Detection," *International Journal of Advances in Intelligent Informatics*, vol. 7, no. 1, pp. 37-48, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mohammed Aljebreen et al., "Modified Equilibrium Optimization Algorithm with Deep Learning-Based DDoS Attack Classification in 5G Networks," *IEEE Access*, vol. 11, pp. 108561-108570, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] S. Sumathi, R. Rajesh, and Sangsoon Lim, "Recurrent and Deep Learning Neural Network Models for DDoS Attack Detection," *Journal of Sensors*, vol. 2022, no. 1, pp. 1-21, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Yongqiang Shang, "Prevention and Detection of DDOS Attack in Virtual Cloud Computing Environment Using Naive Bayes Algorithm of Machine Learning," *Measurement: Sensors*, vol. 31, pp. 1-9, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Asha Varma Songa, and Ganesh Reddy Karri, "An Integrated SDN Framework for Early Detection of DDoS Attacks in Cloud Computing," *Journal of Cloud Computing*, vol. 13, pp. 1-22, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] S. Balasubramaniam et al., "Optimization Enabled Deep Learning-Based DDoS Attack Detection in Cloud Computing," *International Journal of Intelligent Systems*, vol. 2023, no. 1, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] M. Jahir Pasha et al., "LRDADF: An AI Enabled Framework for Detecting Low-Rate DDoS Attacks in Cloud Computing Environments," *Measurement: Sensors*, vol. 28, pp. 1-11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Ankit Agarwal, Manju Khari, and Rajiv Singh, "Detection of DDOS Attack Using Deep Learning Model in Cloud Storage Application," *Wireless Personal Communications*, vol. 127, pp. 419-439, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] Mahrukh Ramzan et al., "Distributed Denial of Service Attack Detection in Network Traffic Using Deep Learning Algorithm," *Sensors*, vol. 23, no. 20, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]