Future Technology
Open Access Journal
ISSN 2832-0379

February 2026| Volume 05 | Issue 01 | Pages 195-208

Journal homepage: https://fupubco.com/futech

https://doi.org/10.55670/fpll.futech.5.1.17

Future Publishing LLC

Article

# Deep stacked autoencoder with fractional VCROA for DDoS attack detection using a big data approach in the MapReduce framework

**Rahul Vijay Kotawadekar, Suhasini Vijaykumar*, Priya Chandran**

Department of Computer Applications, Bharati Vidyapeeth's Institute of Management & Information Technology,
Navi Mumbai, Maharashtra, India

ARTICLE INFO

*Article history:*
Received 12 August 2025
Received in revised form
14 October 2025
Accepted 20 November 2025

Keywords:
Internet, Deep Learning, Cybersecurity,
Attack detection, Distributed denial of service

*Corresponding author
Email address:
suhasini.kottur12@gmail.com

DOI: 10.55670/fpll.futech.5.1.17

ABSTRACT

The rising dependence on internet-based services has exposed network infrastructure to increased vulnerability to cyberattacks, especially DDoS attacks. The attacks flood target systems with unwarranted traffic that disrupts legitimate access and undermines service reliability. To overcome this issue, the present paper proposes an optimization-based deep learning model, called Fractional Velocity Contour-based Remora Optimization Algorithm-Deep Stacked Autoencoder (FVCROA_DSA), for high-efficiency DDoS attack detection in a MapReduce environment. The model combines a mean-substitution method for filling data gaps and Support Vector Machine Recursive Feature Elimination (SVM-RFE) in the mapper step to identify the most significant network attributes. This step is followed by the reducer stage, which trains a Deep Stacked AutoEncoder (DSA) to recognize attack patterns, which is then fine-tuned by the proposed FVCROA algorithm. Fractional Calculus leads to increased optimization stability and faster convergence during training. Experimental tests on the BOT-IoT and DDoS Attack datasets show that the FVCROA architecture with DSA achieves higher detection accuracy, with a precision of 93.857, a recall of 94.827, and an F-measure of 94.340, surpassing the current baseline techniques in scalability and reliability.

## 1. Introduction

In recent years, a wide range of internet-based services—such as newsgroups, social networking, e-commerce, banking, and online trading—have become integral to modern digital infrastructure [1]. The growing dependency on interconnected communication systems has accelerated data transfer and management across distributed networks [2]. However, these advancements have simultaneously expanded the surface for cyber threats and malicious activities [3]. Among these threats, Distributed Denial of Service (DDoS) attacks remain one of the most persistent and damaging forms of cyber intrusion [4]. A DDoS attack overwhelms targeted servers or applications with excessive illegitimate requests, thereby exhausting resources and disrupting legitimate access to services [5]. For instance, Amazon Web Services (AWS) experienced a 2.3 Tbps DDoS attack in February 2020 [6], while Google Cloud reported 46 million requests per second directed at one of its clients in June 2022 [7]. Recent analyses further reveal that the frequency and magnitude of DDoS incidents have continued to increase globally [8]. These attacks cause severe financial losses, service unavailability, and reputational damage, making continuous protection of network infrastructure a critical cybersecurity requirement [9]. To ensure secure and reliable operations, sectors such as government, finance, healthcare, and defense increasingly rely on intrusion detection systems (IDSs) for threat monitoring [10]. However, traditional IDS architectures face scalability and responsiveness limitations when processing large, heterogeneous network traffic [11]. With the explosive growth of network data, big data frameworks such as Hadoop, Spark, and MapReduce have emerged as effective platforms for parallel and distributed analysis [12]. Among these, MapReduce has proven particularly efficient due to its scalability and ability to divide high-volume network traffic into smaller, manageable batches for distributed processing [13]. This capability makes it highly suitable for modern DDoS detection systems that require real-time adaptability and computational resilience [14]. Previously existing detection systems operated mainly using signature-based and anomaly-based methods [15]. Signature-based solutions are effective at identifying familiar attack patterns [16] but ineffective against new or emerging attacks. On the other hand, anomaly systems can detect unknown behavior but tend to be very sensitive, with a high number of false positives, and prone to instability under changing conditions [17]. These shortcomings limit their usefulness in large-scale, fast-changing network environments [18].

**Abbreviations**

| | |
|---|---|
| AWS | Amazon Web Services |
| BOT-IoT | Botnet Internet of Things (dataset) |
| CIC | Canadian Institute for Cybersecurity (implied from CICflow meter tool) |
| CSV | Comma-Separated Values |
| DDoS | Distributed Denial of Service |
| DNS | Domain Name System |
| DoS | Denial of Service |
| DSA | Deep Stacked Autoencoder |
| DNN | Deep Neural Network |
| FC | Fractional Calculus |
| FVCROA | Fractional Velocity Contour-based Remora Optimization Algorithm |
| GAN | Generative Adversarial Network |
| HHO | Harris Hawks Optimization |
| IP | Internet Protocol |
| LDAP | Lightweight Directory Access Protocol |
| MSSQL | Microsoft SQL Server |
| NTP | Network Time Protocol |
| PCAB | Packet Capture (pcap) files (noted as PCAB in text) |
| PSO | Particle Swarm Optimization |
| ROA | Remora Optimization Algorithm |
| RFE | Recursive Feature Elimination |
| SFO | Sailfish Optimizer |
| SVM-RFE | Support Vector Machine–Recursive Feature Elimination |
| SVM | Support Vector Machine |
| SNMP | Simple Network Management Protocol |
| SSDP | Simple Service Discovery Protocol |
| SYN | Synchronize (TCP flag) |
| UDP | User Datagram Protocol |
| TFTP | Trivial File Transfer Protocol |
| VCROA | Velocity Contour-based Remora Optimization Algorithm |
| WOA | Whale Optimization Algorithm |
| XAI | eXplainable Artificial Intelligence |

To address these limitations, machine learning (ML) methods that learn discriminative statistical features to distinguish normal from abnormal network traffic have gained increased attention [19]. Popular algorithms for detecting DDoS and classifying network anomalies include Support Vector Machines (SVMs), Decision Trees, and Random Forests [20]. These methods, however, are highly manual and fail to achieve high detection rates in complex, high-dimensional data sets [21]. In recent years, deep learning (DL) methods have become highly promising, as they can automatically extract hierarchical and nonlinear representations from massive data [22]. Convolutional Neural Networks (CNN) [23], Recurrent Neural Networks (RNN) [24], and Autoencoder models have been shown to perform better at detection than more traditional approaches to ML. However, these models commonly face high computational cost, sensitivity to hyperparameters, and limited flexibility in real-time big-data environments [26].

To overcome these issues, optimization algorithms were implemented within DL frameworks to improve training efficiency and detection performance [27]. Whale Optimization Algorithm (WOA) [28], Particle Swarm Optimization (PSO) [29], and Genetic Algorithms (GA) [30] have been optimized to improve parameter optimization and reduce the false-positive rate. Nonetheless, most traditional optimization approaches are slow and tend to get stuck in local optima. Besides, the use of Fractional Calculus (FC) in optimization has not been thoroughly investigated, despite its ability to provide higher accuracy and dynamic stability for more intricate search algorithms [31]. Based on these difficulties, this paper suggests a deep learning framework based on optimization that combines the Fractional Velocity Contour-based Remora Optimization Algorithm (FVCROA) with a Deep Stacked Autoencoder (DSA) in a MapReduce setup:

(1) enhance scalability and real-time detection capability through distributed MapReduce-based data processing;

(2) improve feature selection and dimensionality reduction using SVM–Recursive Feature Elimination (SVM-RFE) in the mapper phase; and

(3) Optimize DSA hyperparameters via FVCROA to improve convergence speed, classification accuracy, and robustness against high-dimensional traffic.

Lastly, the effectiveness of the framework is evaluated using two benchmark datasets, namely BOT-IoT and DDoS Attack, to demonstrate that it achieves better detection accuracy, convergence, and scalability than current ML-, DL-, and optimization-based models. The proposed method is an intelligent, scalable, and efficient solution to the problem of modern DDoS attack detection by combining distributed big-data processing with fractional-calculus-based optimization.

## 2. Literature review

Several researchers have investigated deep learning and optimization algorithms to enhance the accuracy of DDoS attack detection while minimizing computational cost. Cil et al. [32] developed a Deep Neural Network (DNN) to identify and categorize DDoS attacks from network data. The model was very accurate, converged quickly, and required minimal computational time despite the small amount of training data. Nonetheless, it was not capable of functioning in real-time detection cases. A deep neural network model was proposed by Sumathi and Karthikeyan [1] to classify known and unknown DDoS attacks, achieving a high packet delivery ratio with low overhead. Despite its strengths, the approach failed to incorporate machine learning algorithms, such as Support Vector Machines (SVMs), to perform comparative analysis and hybrid performance assessment. Akgun et al. [6] introduced a Convolutional Neural Network (CNN)-based methodology that used information gain to select features. This model was not very suitable for high-speed, real-time detection of DDoS attacks, though it had fewer parameters and incurred lower computational cost. On the same note, Anley et al. [29] proposed a CNN architecture using transfer learning that could be used to classify binary and multi-class. This model minimized overfitting and was robust, but could not withstand adversarial attacks or achieve stable performance across varied data. Novaes et al. [33] developed a Generative Adversarial Network (GAN) to detect DDoS attacks via adversarial training and Internet Protocol (IP) packet analysis. By tracking network traffic, the model minimized response time and maximized detection accuracy. However, it did not work well in actual traffic conditions and did not extrapolate to nonhomogeneous data. A Deep

Contractive Autoencoder (DCAE) was proposed by Aktar and Nur [30] for the detection of DDoS attacks, based on a semi-supervised learning approach. Even though it had high reliability in detecting complex attacks with minimal reconstruction error, it performed poorly in multiclass classification and on large benchmark datasets. Agarwal et al. [27] designed a Feature Selection Whale Optimization Algorithm Deep Neural Network (FS-WOA-DNN) to optimize the accuracy of DDoS attack detection. The model used the Whale Optimization Algorithm (WOA) to identify the best feature subsets, reducing computation time and improving classification accuracy. But it was unable to detect new or previously unseen attack types. Sumathi et al. [34] proposed a hybrid Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) units, optimized using the Harris Hawks Optimization (HHO) and Particle Swarm Optimization (PSO) algorithms. This model significantly reduced overfitting and improved overall detection accuracy, yet lacked global interpretability despite its use of explainable AI (XAI) components. In a similar vein, the RHS-RBM model presented in [27] was successful in detecting DDoS attacks quickly and precisely but failed to account for network traffic on virtual machines; hence, it does not apply to the current cloud computing scenario.

Even though these approaches have produced laudable outcomes in terms of detection precision and computational complexity, they have several issues. The majority of the existing schemes are not scalable to large data settings and are unable to handle large or continuous network flows effectively. Most methods exhibit low generalization performance when applied to unknown or heterogeneous data. The process of feature selection and optimization is common and often leads to suboptimal model performance. Also, not many models provide explainable information about their detection decisions, and most depend on offline training, which does not apply to real-time network security monitoring. Recent literature has also focused on hybrid deep learning networks that combine various neural elements to increase flexibility and resilience. The combination of CNNs and LSTMs, or GRU LSTMs, is very useful for representing global and temporal patterns in network traffic, thereby enhancing the identification of dynamic and burst-based DDoS patterns. Nonetheless, these models are computationally costly, and scaling to distributed big-data systems cannot be done without significant effort, though they are exact. Similarly, Bi-LSTM and attention-based models also improve sequential knowledge of packet flows, but they all require manual hyperparameter tuning and therefore cannot be applied to real-time systems that are dynamic. Optimization algorithms have also entered the realm of enhancing the accuracy and convergence properties of learning-based intrusion detection systems. Genetic Algorithms (GA), the Grey Wolf Optimization (GWO), the Harris Hawks Optimization (HHO), and the Ant Colony Optimization (ACO) are metaheuristic algorithms used to optimize hyperparameters, select the best features, and reduce classification errors. For example, GA-optimized CNN models and PSO-based DNN models have shown faster convergence and fewer alarms than manually adjusted models. Nevertheless, the majority of these algorithms are predetermined by initial parameter values and may become stuck in local optima and lose consistency on large-scale or non-homogeneous data. Furthermore, there has been an increasing body of research on the use of explainable artificial intelligence (XAI) in conjunction with deep learning to enhance transparency in intrusion detection decisions. Even though tools such as SHAP (Shapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) have been employed to understand deep network outputs, integrating them can increase computational cost, making them less useful in real-time. In addition, although several studies suggest distributed frameworks based on Spark or Hadoop for scalable processing, few have integrated these frameworks with adaptive, optimization-based models that support continuous online learning. The combination of these drawbacks underscores the need for a single, scalable system that integrates distributed big-data processing with intelligent optimization and deep feature learning. It must be capable of managing large, heterogeneous network traffic on a scale, automatically selecting the right features, dynamically tuning hyperparameters, working effectively in real-time settings, and being explainable and robust to adversarial conditions.

## 3. Designed the FVCROA_DSA technique for DDoS attack detection

This article presents an FVCROA_DSA model for detecting DDoS attacks using the MapReduce framework. The input data is initially retrieved from the database and then undergoes data preprocessing to impute missing values using the mean substitution strategy. Then, the pre-processed data is fed into the MapReduce framework, which comprises a mapper phase and a reducer phase. The proposed framework was implemented using the Hadoop MapReduce architecture on a distributed 16-node cluster (Intel i9 processors, 16 GB RAM per node, Ubuntu 20.04). The implementation utilized the Python Pydoop interface for efficient parallel execution and scalability. The selection of suitable features using SVM-RFE [35] is performed from the pre-processed data in the mapper phase. Here, the SVM hyperparameters are optimally adjusted by training the SVM with the VCROA method. The VCROA is developed by incorporating the velocity-contour-based concept with ROA [36]. Later, the selected features are merged and fed into the reducer phase to accurately detect DDoS attacks. The DDoS attack is detected using the DSA [37] model, where the DSA's optimal weights are fine-tuned using the FVCROA model. Here, FVCROA is designed by incorporating FC [38] with VCROA. Further, Figure 1 shows the diagrammatic view of FVCROA_DSA to detect DDoS attacks.

### 3.1 Data collection

The input data is primarily derived from the BOT-IoT database [39] and the DDoS attack database [40] for DDoS attack detection. Here, the BOT-IoT database comprises about 72 million records with real-time and simulated scenarios. The database consists of four attack categories, each with data types related to DoS and DDoS attacks. Similarly, the DDoS attack database comprises instances of more than 5 crore recorded from background traffic flows. The database considered for the detection task is represented by,

$$W = [W_1, W_2, W_3, \ldots, W_D, \ldots, W_Z] \qquad (1)$$

wherein, the total data presented in the dataset is signified as$Z$, the input database used for detecting DDoS attacks is given by$W$, and the $D^{th}$ input data taken for the detection task is signified as $W_D$.

The BOT-IoT and DDoS Attack datasets were divided into 70% training, 15% validation, and 15% testing subsets. Data were shuffled using stratified sampling to maintain class balance, and all experiments were performed with a fixed random seed (42) to ensure reproducibility.

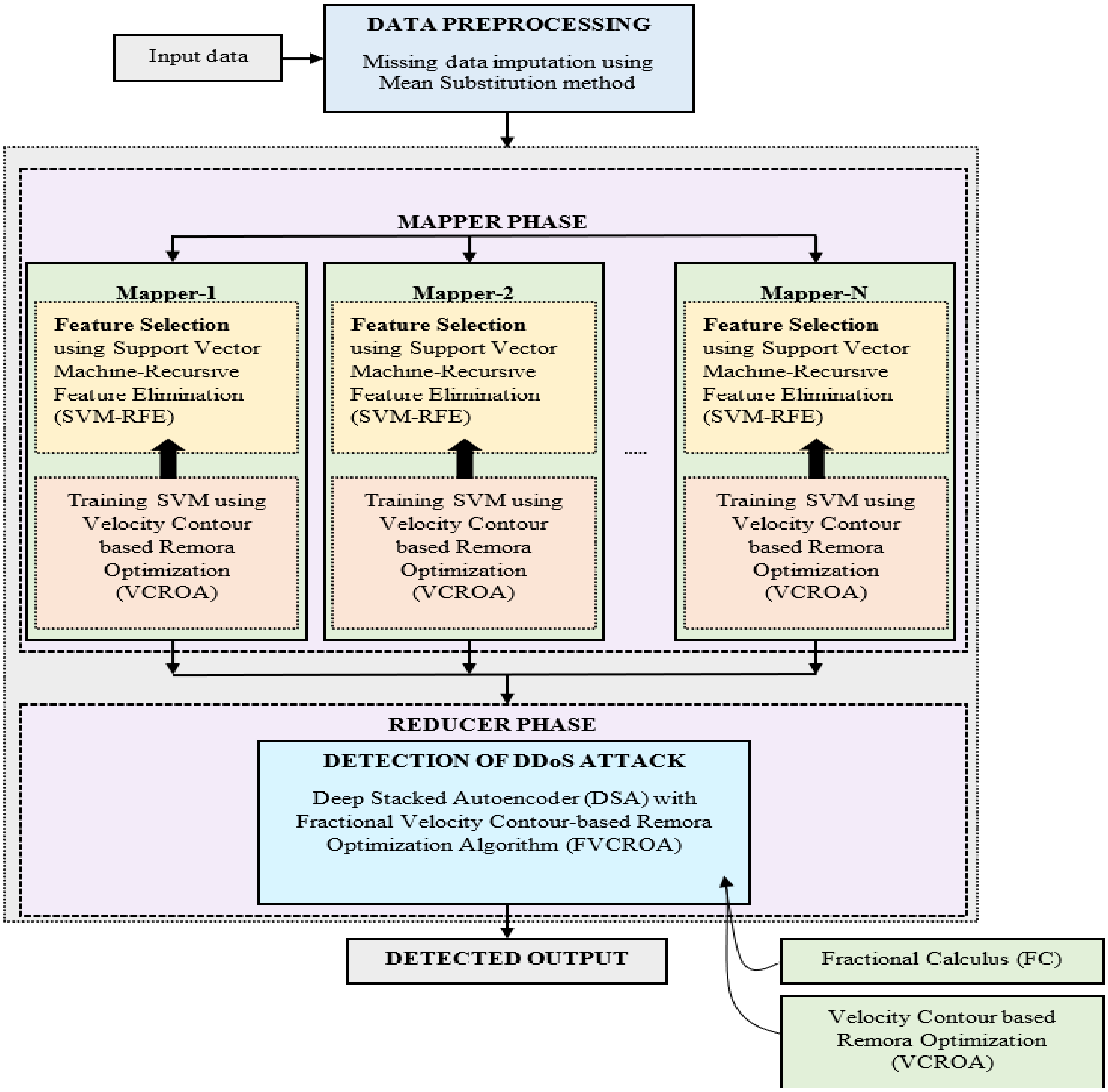

Input data
DATA PREPROCESSING
Missing data imputation using Mean Substitution method
MAPPER PHASE
Mapper-1
Mapper-2
Mapper-N
Feature Selection using Support Vector Machine-Recursive Feature Elimination (SVM-RFE)
Training SVM using Velocity Contour based Remora Optimization (VCROA)
.....
REDUCER PHASE
DETECTION OF DDoS ATTACK
Deep Stacked Autoencoder (DSA) with Fractional Velocity Contour-based Remora Optimization Algorithm (FVCROA)
DETECTED OUTPUT
Fractional Calculus (FC)
Velocity Contour based Remora Optimization (VCROA)

**Figure 1.** Diagrammatic view of the FVCROA_DSA technique to detect DDoS attacks

### 3.2 Data pre-processing

The aggregation, transforming, and cleaning of raw data by removing inaccurate or corrupted records from the input data is termed data pre-processing. The input data$W_D$is fed to data pre-processing by utilizing missing data imputation. Missing data imputation is the process of retaining data by replacing missing values with substitute values based on available information in the database. Here, the missing data is imputed by applying the mean substitution method, where the missing data of a variable is replaced by computing the mean of the input data$W_D$for each variable. Thus, the pre-processed data $D_{Pre}$is obtained by executing the mean substitution method and is further fed into the MapReduce framework for attack detection. Additionally, categorical variables were label-encoded, and all numerical attributes were normalized using min–max scaling to the range [0, 1] before model training.

### 3.3 MapReduce framework

The MapReduce framework is generally considered a programming framework that is used to process large datasets to detect DDoS attacks more precisely. The MapReduce framework is a simple, flexible, and reliable method that utilizes the mapper and reducer phases to execute the detection task. Here, the pre-processed data $D_{Pre}$is initially split into key-value pairs and passed to the mapper phase for feature selection; the selected features are then passed to the reducer phase for detecting DDoS attacks. The detection process executed in both phases of the MapReduce framework is delineated below,

**Mapper phase for feature selection:** The mapper phase extracts relevant features from distributed data partitions in parallel, reducing computational load and improving processing efficiency. In general, the transformation of pre-processed data $D_{Pre}$ into a suitable form for processing the data is performed in the mapper phase to accurately detect

DDoS attacks. Here, a subset of relevant features is selected from the pre-processed data $D_{Pr\,e}$by isolating the non-redundant and most consistent data via feature selection to reduce the number of input variables. The mapper phase consists of $G = 8$ mappers in the experimental setup, each performing independent feature selection on distributed data blocks, with selection of relevant features executed separately in each mapper using SVM-RFE.

**Feature selection using SVM-RFE:** SVM-RFE removes one feature at a time during feature selection. In this study, the SVM-RFE process employs a linear kernel Support Vector Machine (SVM) to ensure stable weight-vector computation and consistent feature ranking. The regularization parameter $C = 1.0$and tolerance $= 10^{-4}$ were used as baseline settings, while the kernel parameter $\gamma = 0.01$was dynamically adjusted using the Velocity Contour-based Remora Optimization Algorithm (VCROA) during model training. This automated tuning process helps achieve a balance between model generalization and feature sparsity. The elimination of features proceeds iteratively until the ranking coefficients converge below the stopping threshold. At each step, the feature variables are determined in SVM-RFE, and a ranking coefficient is used during training for the generation of a weight vector $P$ by SVM. Then, the removal of minimum ranking coefficient signature attributes is executed at each iteration, and the resultant signature attributes are organized in descending order. The selection of relevant features in SVM-RFE is demonstrated below:

Initially, the training samples $[A_c, B_c], B_c \in [-1, +1]$ are entered by considering $Z_{select}$as the output feature ordering set. Then, based on the condition$Z_{select} \neq 0$, the initialization of the original feature set$D_{Pr\,e} = 1,2,3,\ldots,H$ is performed, and the procedure is continuously followed until $Z_{select} \neq 0$. Following this, the training set and a candidate feature set are determined, and then, SVM training is performed to identify$P$. After that, the ranking of the criteria score is executed, and is given by:

$$I_g = P_g^2, g = 1,2,3,\ldots,|D_{Pr\,e}| \tag{2}$$

Later, $\underset{g}{argI_g}$is used to determine the score features with the small ranking criteria, and the feature set is updated as $Z_{select} = J \cup Z_{select}$. Here, $J$is the feature with the smallest weight. Finally, the removal of features in the pre-processed data $D_{Pr\,e}$is performed such that $D_{Pr\,e} = \frac{D_{Pr\,e}}{J}$ . Thus, the output of the selected feature $Z_{select}$is determined by utilizing SVM-RFE from the pre-processed data $D_{Pr\,e}$.

For SVM training, the Radial Basis Function (RBF) kernel was used with a regularization parameter C = 1.0, kernel coefficient γ = 0.01, and stopping tolerance $10^{-4}$. These parameters were optimized dynamically using the VCROA algorithm.

**Solution encoding using VCROA:** The solution encoding selects the optimal solution from the available features. The solutions identified using the VCROA method are effectively represented during feature selection in the solution encoding. Here, the solution is randomly initialized to select optimal features based on the features' indices. The solution encoding performed to effectively select features is given in Figure 2.

**Fitness function:** Using the expression provided in equation (2), the fitness of VCROA is calculated based on the ranking criterion score of SVM-RFE during feature selection. The fitness function from a set of variables is selected for the identification of the optimal solution.

**Designed the VCROA model for training SVM:** The hyperparameters of SVM are optimally tuned using the VCROA approach, where VCROA is developed by incorporating a velocity-based concept in ROA. A naturally-inspired metaheuristic algorithmic technique called ROA is based on the parasitic behavior of remora by switching the hosts. The Remora is a marine fish commonly found in tropical waters with a cylindrical, backward, and flat heads that belong to the Echeneidae family. Generally, remora follow the host movement in cold water and evade enemy invasion by swimming over whales, sharks, hulls, and other animals. Remora feed on ectoparasites available over the surface of the host and mainly depend on invertebrates or fish as their food. The host feeding during the eat thoughtfully (exploitation) and free travel (exploration) phases is followed by Remora to escape other enemies. The Remora converges quickly in the search space and is more effective at reducing computational complexity at high execution speeds. The velocity-contour-based concept is incorporated with ROA to handle high-dimensional problems by maintaining good exploratory and underexploited conditions. The VCROA algorithm optimizes the SVM hyperparameters by balancing exploration and exploitation, thereby accelerating convergence while maintaining accuracy. The mathematical modeling of VCROA is expressed as:

**Phase 1:** Population initialization

Let us consider a $X$ search space with$a$number of remora population, which is given by,

$$X_a = (X_{a1}, X_{a2}, X_{a3}, \ldots, X_{aj}) \tag{3}$$

Here, the solution dimension $u^{th}$ remora is given by $j$ , and the total of remoras is indicated as $a$ . Moreover, the remora vector differs from each other in terms of the size of the variant. Thus, the best food target$X_B$ is given by:

$$X_B = (X_1^*, X_2^*, X_3^*, \ldots., X_a^*) \tag{4}$$

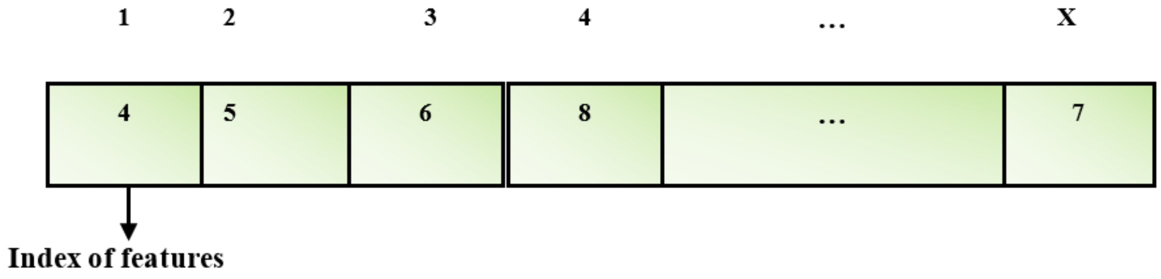


**Figure 2.** Solution encoding executed to select optimal features using VCRO

**Phase 2:** Fitness estimation

The fitness function used to determine the best solution is given in equation (2).

**Phase 3:** Exploration (Free travel) phase

In the exploration phase, Remora follows the experience attack and Sailfish optimizer (SFO) strategy to update its location, as described below.

**SFO procedure:** The remora position is modified when the remora is attached to the swordfish, where the updating of remora location is performed by considering the SFO procedure and is expressed as:

$$X_a^{u+1} = X_B^u - \left(R(0,1) * \left(\frac{X_B^u + X_R^u}{2}\right) - X_R^u\right) \tag{5}$$

Here, the random number is given as$R$, $U$resembles the maximum number of repetitions; the undisturbed place is represented as$X_R$, and the current iteration is signified $u$. Moreover, the exploration of the search space is ensured by adding the random selection of remora. The remora selection criteria for diverse hosts are based on whether the prey is consumed, and the achieved objective function is superior to those of previous functions. Further, the experience attack strategy is followed by the remora to record the current fitness value.

**Experience attack:** The tuyu takes small steps around the host repeatedly to determine the necessary changes to the host, which is similar to accumulating experience. Thus, the experience accumulated by the remora is expressed as,

$$X_A = X_a^u + (X_a^u - X_P) * RV \tag{6}$$

where the tentative step and the earlier generation position of remora are symbolized as $X_A$. Further, the remora performs a "small global movement" while making an active step by using $RV$.

**Phase 4**: Exploitation (Eat thoughtfully) phase

The host feeding as well as Whale Optimization Algorithm (WOA) strategies are followed by the remora during the early thoughtfully phase, and the process performed is demonstrated as follows.

**WOA strategy:** The WOA technique is used to determine the location of the remora associated with the whale, and it is supplied by:

$$X_{a+1} = D * e^h * cos(2\pi\delta) + X_a \tag{7}$$

$$\delta = R(0,1) * (h-1) + 1 \tag{8}$$

$$h = -\left(1 + \frac{u}{U_{Max}}\right) \tag{9}$$

$$D = |X_B - X_a| \tag{10}$$

where the position of a remora on a whale is symbolized as$h$, the undistributed measure is symbolized as $\delta$ that sets to the range $[-1,1]$ and deteriorates linearly to $[-2,-1]$, the distance between the prey and the hunter is given as $D$, and the maximum iteration is denoted by $U_{Max}$.

**Host feeding:** The subdivision of the exploitation procedure is host feeding, where host feeding is executed to minimize the solution space to the host location space. The remora follows small steps to move around the host, and is mathematically expressed as,

$$X_a^u = X_a^u + N \tag{11}$$

$$N = T * (X_a^u - \chi * X_B) \tag{12}$$

$$T = 2 * C * R(0,1) - C \tag{13}$$

$$C = 2 * \left(1 - \frac{u}{U_{max}}()\right) \tag{14}$$

where the remora factor is symbolized as $T$that helps to minimize the position of remora, the tiny step of remora motion is represented as $N$ , and the remora factor is given as $\chi$.

**Velocity-contour-based concept:** The remora's velocity towards the host is used to fully update the solution in the search space. Thus, the algorithm for the remora position is derived mathematically by considering the velocity of the remora and is expressed by:

$$X_a^{u+1} = X_a^u + Y * \chi * X_B + T \tag{15}$$

where, $Y$signifies the velocity of remora.

**Phase 5:** Re-estimation of fitness

Equation (2) is used to calculate the fitness function, and if any solutions are found to be better than the obtained solution, the discovered solution is substituted. This yields the optimal solution.

**Phase 6:** Termination

The pseudo-code for VCROA is presented in Table 1, and the best solution is obtained by following the algorithmic steps continuously.

Table 1 illustrates the step-by-step process of VCROA used for SVM hyperparameter optimization. Thus, the selected feature $Z_{select}$is obtained using SVM-RFE, with the hyperparameters of SVM optimally trained using the VCROA algorithm in the mapper phase. The features selected in the mapper phase are further merged and then subjected to the reducer phase for DDoS attack detection.

**Reducer phase for attack detection using FVCROA_DSA:** In the reducer phase, the outputs from all mappers are aggregated and used as input to the Deep Stacked Autoencoder (DSA) classifier for the final DDoS attack detection stage. The resultant outputs obtained by the total number of $G$mappers are processed further to a suitable form for detecting DDoS attacks. Here, the selected features $Z_{select}$ obtained by SVM-RFE in the mapper phase, by merging all the mappers and feeding them into DSA to detect DDoS attacks more accurately. Additionally, by training the ideal weights of DSA, the FVCROA algorithm is utilized to improve detection performance. Thus, the detection process carried out in the reducer phase using the DSA model is explicated below.

**Architecture of DSA:** The most common type of Deep Neural Network (DNN) is the DSA model, which consists of multiple layers of autoencoders interconnected by adjacent-layer neurons. In the proposed model, the DSA architecture consists of an input layer followed by two hidden autoencoder layers and one output layer. The encoder employs ReLU activation with 84, 64, and 32 neurons in successive layers, while the output layer uses Softmax activation for binary classification. To prevent overfitting, dropout rates of 0.3 and 0.2 are applied to the first and second hidden layers, respectively. The network is trained using the Adam optimizer with a learning rate of 0.001, a batch size of 64, and 100 training epochs, while a weight decay of 0.0001 ensures better generalization. In DSA, the autoencoders consist of two parts, an encoder and a decoder.

**Table 1**. Pseudo code of VCROA

```
Pseudo code of VCROA
Input: Initial position$X_a^u$, Distance$D$, Best position$X_B^{u+1}$
Output: Optimal solution$X_a^{u+1}$
Initially set the location of the remora population and memory
Set optimal fitness and determine optimal solution by utilizing the equation (2)
$while u < U_{max} do$
    Compute the value of fitness function for each remora
    Check to see whether any search agent exceeds the search space, then disregard it
    Modify□, $\delta$, $V$
    $for$ each indexed remora of $a do$
        $If$ the Selection factor $Q(a) = 0 then$
            Modify the attached whales' position by utilizing the equation (7)
        $else$
            $if Q(a) = 1 then$
                Modify the attached sailfish position by utilizing the equation (5)
            $end if$
            Execute experience attack by utilizing the equation (6)
            Identify the value of $Q(a)$and check the necessary host replacement
            Perform host feeding mode of remora by utilizing the equation (11)
        $else$
            Modify the remora position by considering the velocity contour-based concept by utilizing the equation (15)
    $end for$
$end while$
```

Here, an input, hidden layer, and mapping function are available in the encoder layer, whereas the expression of the hidden layer output of DSA is given by:

$$Z_g(Z_{select}) = f(\alpha Z_{select} + \rho) \tag{16}$$

where, $Z_g$represents a reconstructed signal, $f(\bullet)$ signifies the activation function of the encoder, and the bias vector is represented as$\rho$, and $\alpha$ resembles an encoding weight matrix. Further, a hidden mapping function and an output layer are presented in the decoder layer of the DSA. Also, in each hidden layer, the dropout technique is executed, and the neural network with a mapping relationship is given by,

$$\hat{Z}_{select}(\lambda) = y(\alpha^*\lambda + \rho^*) \tag{17}$$

where, $\lambda$resembles hidden layer output, $y(.)$resembles the activation function of the decoder, $\rho^*$resembles the bias vector, and the weight matrix of the decoder is represented as$\alpha^*$. In addition, the loss function of DSA is estimated by utilizing Mean Squared Error (MSE), which is expressed as:

$$MSE = \frac{1}{2}\left\|Z_{select} - \hat{Z}_{select}\right\|^2 \tag{18}$$

The model was trained for 100 epochs with an early-stopping patience of 10 epochs to prevent overfitting. The DSA utilizes two operations during training to provide solutions to over-fitting issues that occur due to over-training of samples. The DSA utilizes dropout as the first operation and weight decay coefficients in the second operation to execute the attack task. The dropout function is utilized for neglecting the neurons available in the hidden layer, and the predicted value with the probability $m_i$to generate a sub-network. Later, the confirmation of the discarded matrix $x = [x_1, x_2, x_3, \ldots, x_r]$ using the Bernoulli distribution is performed. Hence, the resultant function is expressed as:

$$\mu(c, m_i) = \begin{cases} m_i, & c = 1 \\ 1 - m_i, & c = 0 \end{cases} \tag{19}$$

Here, $c$ is the possible output. Furthermore, in each hidden layer, a dropout operation is performed, and is expressed by:

$$\lambda_s(Z_{select}) = x_s f\left(\sum_{c=1}^{r} \alpha Z_{select} + \rho\right) \tag{20}$$

where the number of elements is given by $r$and the discarded matrix is represented as$x_s$. Further, the solutions to over-fitting issues are provided by a weight decay coefficient, and the corresponding overall loss function is given by,

$$K(\alpha, \rho) = \varepsilon + \frac{\varpi}{2}\sum_{m=1}^{M-1}\sum_{n=1}^{q}\sum_{o=1}^{q+1}\left(\alpha_{no}^{(m)}\right)^2 \tag{21}$$

Here, $o$and $n$ is the neuron presented in $m^{th}$ and $(m+1)^{th}$ layer, the total number of neurons are represented as $q$, $\varpi$resembles the weight decay coefficient, the total layers is given as$M$, and the connection weight among the neurons is given by $\alpha_{no}^{(m)}$. Also, the categorical cross-entropy function is given by $\varepsilon$ and is given by:

$$\varepsilon = -\frac{1}{\psi}\sum_{c=1}^{\psi}\sum_{d=1}^{\psi_{cc}} w_{no} \bullet log\left(\hat{w}_{no}\right) \tag{22}$$

where, $w$and $\hat{w}$ resembles the actual and expected labels; the number of labels is given by $\psi_{cc}$. Usually, the reconstruction error of DSA is minimized by determining optimal parameters. Thus, the resultant output $d_{output}$ is obtained using the DSA model, and the architecture of DSA is shown in Figure 3.

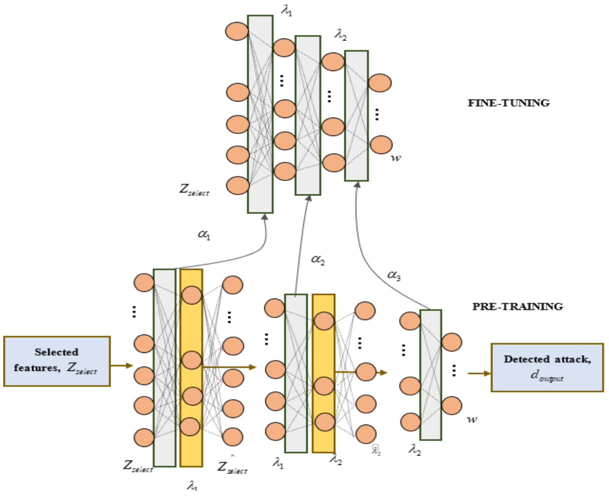


**Figure 3.** Architecture of DSA

**Designed the FVCROA model for the training of DSA:** The hyperparameters of the DSA model are optimally adjusted by training using the FVCROA method, where FVCROA is developed by combining FC with VCROA. FC helps to improve the detection performance of the algorithmic models, which also promotes accurate solutions by utilizing the Laplace transform for derivative and integral equations. Further, the inverse Laplace transform is executed to achieve suitable solutions. The integration of Fractional Calculus (FC) into the VCROA algorithm enhances its global search ability by introducing memory-dependent dynamics to the optimization process. This mechanism helps the algorithm retain useful information from previous iterations, thereby improving convergence stability and reducing the likelihood of getting trapped in local minima during DSA training. Thus, the mathematical modeling of FVCROA is given in equation (15). FC [38] is incorporated with VCROA to attain suitable solutions in the search space. From FC,

$$“z^{l}[X_a^{u+1}] = X_a^u + Y * \chi * X_B + T” \tag{23}$$

$$“X_a^{u+1} - lX_a^u - \frac{1}{2}lX_a^{u-1} - \frac{1}{6}(1-l)X_a^{u-2} - \frac{1}{24}l(1-l)(2-l)X_a^{u-3} = X_a^u + Y * \chi * X_B + T” \tag{24}$$

where, $l$resembles the order of the derivative. Hence, the final modified expression of FVCROA is expressed as,

$$X_a^{u+1} = X_a^u(1+l) + Y * \chi * X_B + T + \frac{1}{2}lX_a^{u-1} + \frac{1}{6}(1-l)X_a^{u-2} + \frac{1}{24}l(1-l)(2-l)X_a^{u-3} \tag{25}$$

where,$X_a^{u+1}$indicates the remora position at$(u+1)^{th}$ iteration, the site of remora at$(u+2)^{th}$ iteration is represented as$X_a^{u+2}$, and $X_a^{u+3}$ represents the Remora site at $(u+3)^{th}$ iteration. Moreover, the best solution is determined using the fitness function given in equation (18). The proposed FVCROA_DSA framework integrates distributed computing, intelligent feature selection, and deep learning optimization. The combined use of MapReduce, SVM-RFE, VCROA, and DSA enables efficient large-scale DDoS detection with improved scalability, faster convergence, and reduced overfitting compared with conventional centralized methods.

## 4. Results and discussion

To quantitatively analyze the supremacy of FVCROA_DSA introduced for detecting DDoS attacks, the results recorded from the experiment and the discussion that follows are demonstrated below.

### 4.1 Experimental set-up

The Python utility is used to carry out the FVCROA_DSA approach, which is used to detect DDoS attacks. All experiments were executed on a workstation equipped with an Intel i9 processor, 64 GB RAM, and an NVIDIA RTX 3080 GPU running Ubuntu 20.04. The model was implemented in Python 3.10 using TensorFlow 2.13 and Scikit-learn. Each experiment was repeated 10 times with different random seeds (42–51) to ensure statistical robustness, and the reported results represent the mean ± standard deviation (SD) of these runs.

### 4.2 Dataset description

The BOT-IoT dataset and DDoS Attack dataset are the databases utilized in this research to evaluate the performance of FVCROA_DSA in identifying DDoS attacks. The databases are explicated as follows.

**BOT-IoT dataset:** The BOT-IoT database utilized the Shark tool for the generation of raw network packets. The data in this database was produced by combining both normal and anomalous traffic. The database source files are available in multiple formats, such as CSV format, generated Argus files, and the original Packet capture (PCAB) files. Moreover, the subcategory and category of attacks are considered for the separation of data files, thus assisting the data labeling process.

**DDoS Attack dataset:** The DDoS Attack database is available in both CSV and PCAB file formats. The database comprises network traffic data collected over two days, and attacks, such as UDP_Lag, WebDDoS, UDP, MSSQL, NetBIOS, NTP, LDAP, SYN, DNS, SSDP, TFTP, and SNMP, are executed to determine the background traffic flows. Further, the database possesses about 4 lakh instances while randomly extracting 84 features of network traffic using the CICflow meter. Before training, both datasets were standardized and preprocessed as described in Section 3. The BOT-IoT dataset contained approximately 72 million records, while the DDoS Attack dataset included 5 million labeled flow instances. From each dataset, 70% was used for training, 15% for validation, and 15% for testing, ensuring class balance between normal and attack traffic.

### 4.3 Performance metrics

The effectiveness of FVCROA_DSA for detecting DDoS attacks is evaluated using evaluation metrics, which are explicated below.

**Precision:** The proportion of positive values detected truly using FVCROA_DSA to the predicted positive values is precision, which is given as:

$$Pr\,e\,cision, p = \frac{\nu_1}{\nu_1+\nu_3} \tag{26}$$

where false negatives and true positives are signified as $\nu_4$, and $\nu_1$, whereas the false positive is given by $\nu_3$.

**Recall:** The proportion of positive values predicted truly using FVCROA_DSA to the total available positive values is recall, which is given by,

$$Re\,c\,all, r = \frac{\nu_1}{\nu_1+\nu_4} \tag{27}$$

**F-measure:** It is the harmonic mean computed between recall and precision and is formulated as,

$$F - measure = 2 * \frac{p*r}{p+r} \tag{28}$$

### 4.4 Comparative techniques

The prevailing DDoS attack detecting techniques, namely FS-WOA-DNN, Neural Network, GAN, RHS-RBM, and VCROA-Deep Neuro-Fuzzy Network (DNFN), are compared with FVCROA_DSA to determine the performance of FVCROA_DSA in detecting DDoS attacks.

### 4.5 Comparative evaluation

The comparative results reported below represent the mean performance values obtained over 10 independent runs, with negligible variance (< 0.5%). This consistency confirms that the FVCROA_DSA model maintains stable detection performance across different random initializations. The superiority of FVCROA_DSA in identifying DDoS attacks is evaluated using both the BOT-IoT and DDoS Attack datasets. The evaluation performed is briefly explained as follows:

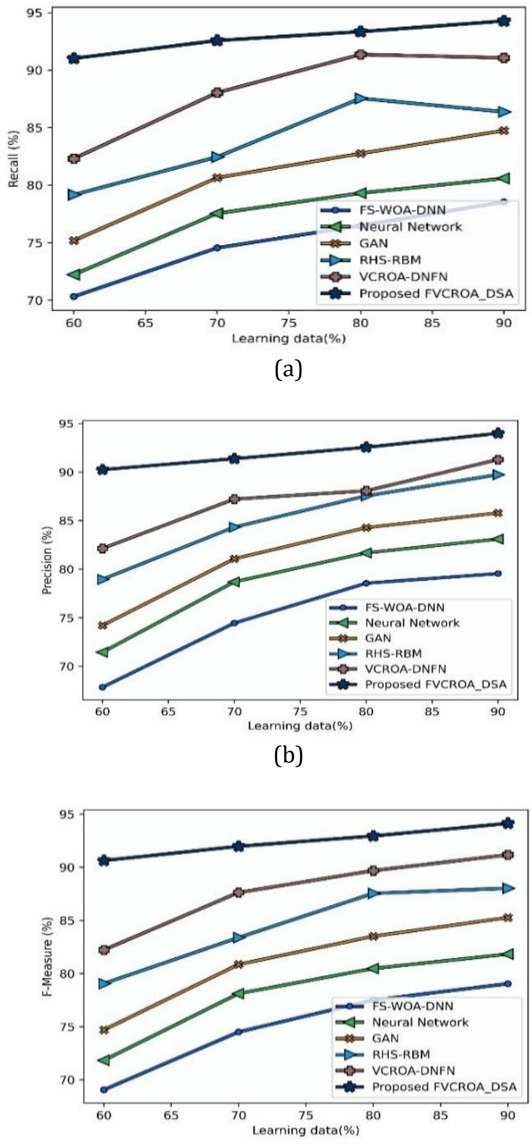
**For the BOT-IoT dataset:** The alterations of K-fold and learning data are performed to identify the supremacy of FVCROA_DSA based on the BOT-IoT dataset, where the validations executed are given below.

**Analysis based on learning data:** Figure 4 shows the experimental results obtained by FVCROA_DSA when using the BOT-IoT database for DDoS attack detection. Figure 4(a) shows the recall-based validation of the planned FVCROA_DSA and other current techniques. In this case, the FVCROA_DSA achieved a recall of 94.275% on 90% of the learning data, which is higher than that of other detection models currently in use. FS-WOA-DNN, RHS-RBM, GAN, Neural Network, and VCROA-DNFN were among the most popular models; they achieved recall values of 86.379%, 84.75%, 80.597%, 78.546%, and 91.072%, respectively. Thus, compared to RHS-RBM, which is used to identify DDoS attacks, FVCROA_DSA achieved a higher recall of 8.38%. Furthermore, the results from the experiment on the DDoS attack detection models, in terms of precision, are shown graphically in Figure 4(b). The precision observed by the DDoS attack detection techniques is 79.548% by FS-WOA-DNN, 83.102% by Neural Network, 85.794% by GAN, 89.734% by RHS-RBM, 91.285% by VCROA-DNFN, and 93.985% by FVCROA_DSA for 90% learning data. Comparatively, a maximum precision of 16.36% is attained by FVCROA_DSA as compared with other existing FS-WOA-DNN detection models. Further, the statistical results obtained by different DDoS attack detection techniques when validated using F-measure are shown in Figure 4(c). The FVCROA_DSA technique developed in this research measured superior results with a maximum F-measure of 94.130% for learning data of 90%. Likewise, the F-measure obtained by existing DDoS attack detection schemes, such as GAN is 85.268%, FS-WOA-DNN is 79.044%, RHS-RBM is 88.024%, Neural Network is 81.830%, and VCROA-DNFN is 91.179%. The outcomes revealed that the FVCROA_DSA model observed a maximum performance of 9.41% to GAN. Overall, these results confirm that the proposed FVCROA_DSA model consistently outperforms all baseline techniques on the BOT-IoT dataset, demonstrating strong generalization capability and stability across learning data variations.

**Analysis based on K-fold:** Figure 5 displays the graphical representation of various outcomes obtained from the experiment by FVCROA_DSA, as well as other existing schemes, during the detection of DDoS attacks based on K-fold by considering the BOT-IoT dataset. Figure 5(a) illustrates the outcomes obtained by different DDoS attack detection models while validating using recall. It is observed that the FVCROA_DSA attained a maximum recall of 94.130%. In contrast, baseline DDoS attack detection techniques showed a recall of 85.279% for RHS-RBM, 78.595% for Neural Network, 76.125% for FS-WOA-DNN, 83.457% for GAN, and 91.006% for VCROA-DNFN for a K-fold value of 8. According to the findings, the FVCROA_DSA model outperformed the current FS-WOA-DNN model by 19.05%. The validation of the designed FVCROA_DSA and other existing DDoS attack detection approaches by employing precision is given in Figure 5(b). The FVCROA_DSA attained a precision of 92.904%, which is superior to the precision recorded by other prevailing models. The prevailing techniques, such as RHS-RBM, GAN, Neural Network, FS-WOA-DNN, and VCROA-DNFN, recorded precision of 88.727%,84.012%, 82.075%, 78.596%, and 90.434% for a K-fold value of 8. Thus, as compared to the current GAN approach, the FVCROA_DSA achieved a better performance of 9.57%. The results obtained from the experiment by the DDoS attack detection schemes employing F-measure are revealed in Figure 5(c). Here, the DDoS attack detection techniques measured F-measure of 77.341% by FS-WOA-DNN, 80.297% by Neural Network, 83.733% by GAN, 86.969% by RHS-RBM, 90.434% by VCROA-DNFN, and 92.904% by FVCROA_DSA for K-fold value 8 9. Here, the superior performance of 2.94% is recorded by FVCROA_DSA than the VCROA-DNFN approach.

**For the DDoS Attack dataset:** The K-fold and learning data were altered to determine the superiority of the FVCROA_DSA technique in detecting DDoS attacks by considering the DDoS Attack dataset.

**Analysis based on learning data:** The graphical representation of different outcomes obtained from the experiment by FVCROA_DSA while detecting DDoS attacks based on learning data by considering the DDoS Attack dataset is depicted in Figure 6.



(a)

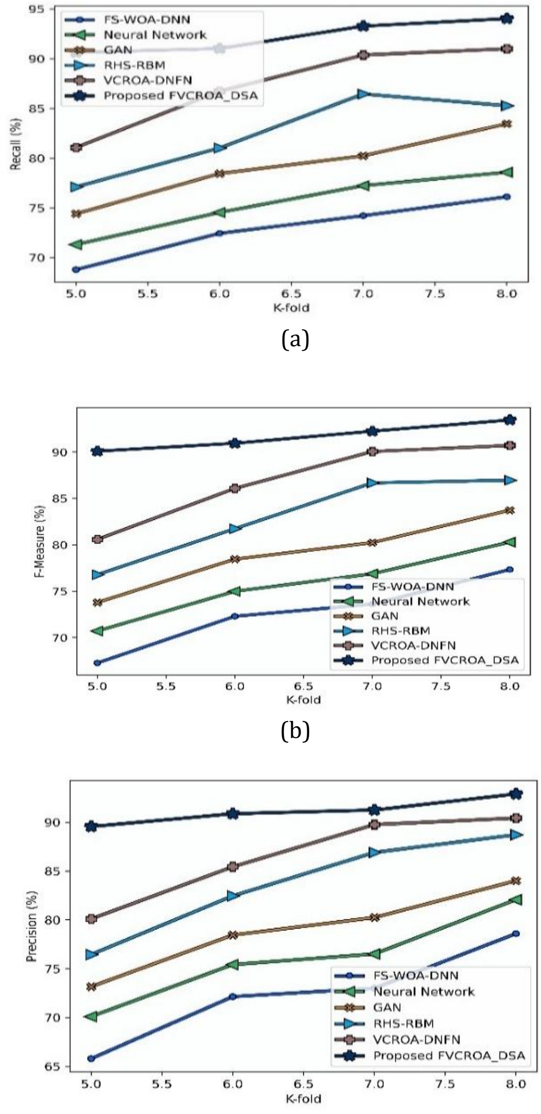
(b)

(c)

**Figure 4.** Comparative evaluation of FVCROA_DSA and baseline methods on the BOT-IoT dataset for varying learning data: (a) Recall, (b) Precision, and (c) F-measure

The results obtained from the experiment by schemes employing recall are graphically given in Figure 6(a). The recall recorded by different detection approaches is 80.276% by FS-WOA-DNN, 84.551% by Neural Network, 86.875% by GAN, 90.346% by RHS-RBM, 92.548% by VCROA-DNFN, and 94.827% by FVCROA_DSA for 90% learning data. Comparatively, the maximum precision of 2.40% is attained by FVCROA_DSA as compared with the VCROA-DNFN approach. Figure 6(b) illustrates the results obtained from the experiment by different DDoS attack detection models while validating using precision. The FVCROA_DSA model measured a maximum precision of 93.857% for 90% learning data. Likewise, the precision measured by other DDoS attack detection schemes, like RHS-RBM, is 88.103%, GAN is

84.275%, Neural Network is 82.367%, FS-WOA-DNN is 77.945%, and VCROA-DNFN is 90.245%.



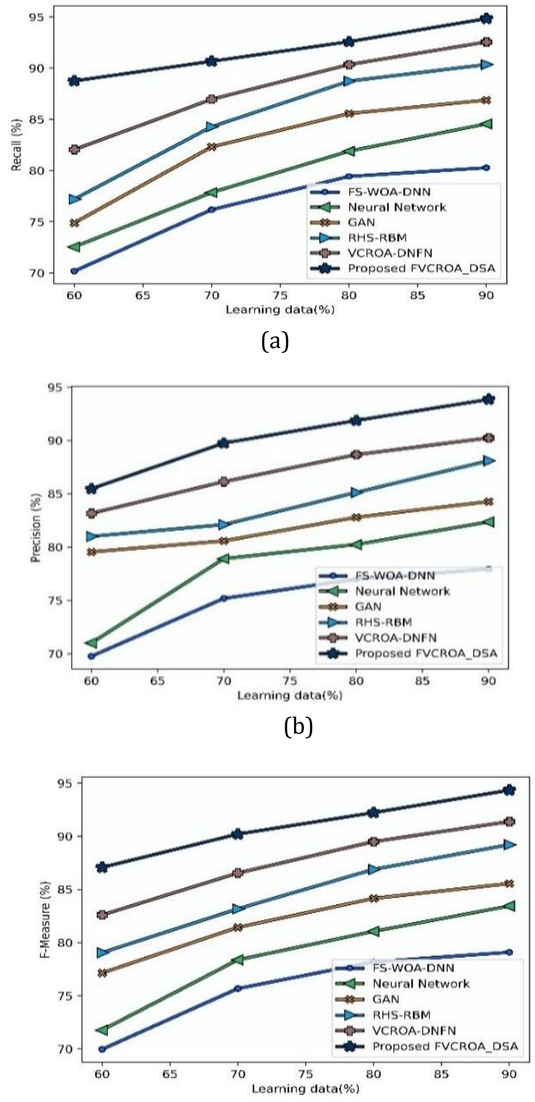
(a)

(b)

(c)

**Figure 5.** Performance of FVCROA_DSA versus existing models on the BOT-IoT dataset under K-fold cross-validation (K=8): (a) Recall, (b) Precision, and (c) F-measure

The results revealed that the FVCROA_DSA model achieved a high detection performance of 16.95%, compared with the existing FS-WOA-DNN model. The validation of the designed FVCROA_DSA and other existing DDoS attack detection schemes employing F-measure is shown in Figure 6(c). The FVCROA_DSA achieved an F-measure of 94.340%, which is higher than the 90% F-measure reported by other existing detection models trained on the learning data. The prevailing techniques, such as RHS-RBM, GAN, Neural Network, FS-WOA-DNN, and VCROA-DNFN, achieved F-measures of 89.210%, 85.555%, 83.445%, 79.093%, and 91.382%, respectively. Thus, the FVCROA_DSA obtained a superior performance of 9.31% as compared with the GAN. These findings indicate that the FVCROA_DSA model achieves consistent improvements over existing schemes when evaluated on the DDoS Attack dataset, confirming its robustness and effectiveness under varying learning data conditions.



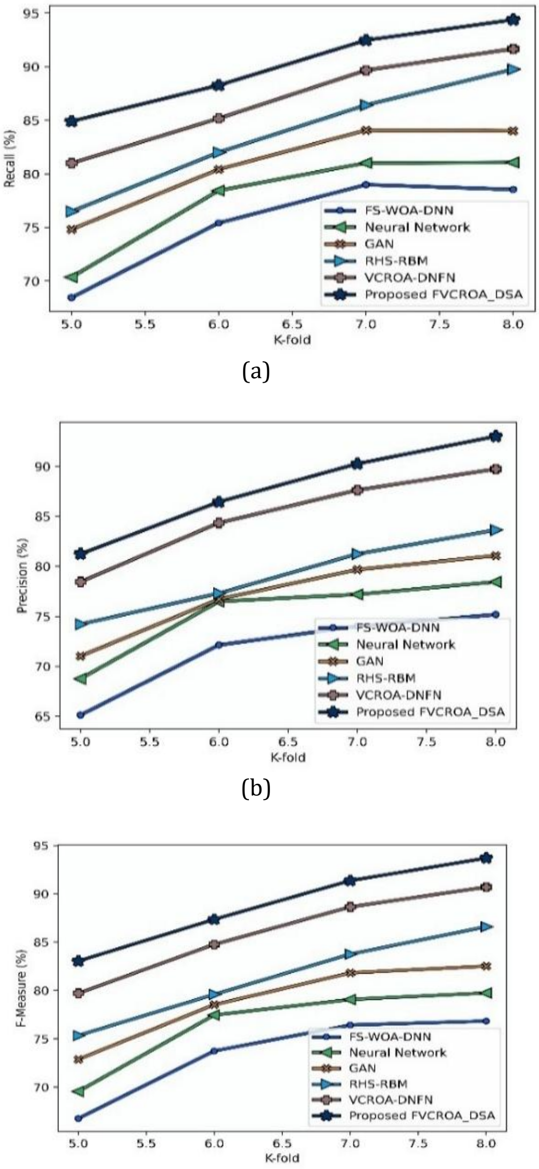
(a)

(b)

(c)

**Figure 6.** Detection accuracy comparison on the DDoS Attack dataset using learning data variation: (a) Recall, (b) Precision, and (c) F-measure

The graphical representation of results recorded by FVCROA_DSA and the prevailing schemes used for comparison while detecting DDoS attacks based on the DDoS Attack dataset, using K-fold, is shown in Figure 7. The analysis of the FVCROA_DSA model, as well as other existing DDoS attack detection schemes using recall, is given in Figure 7(a). Here, the FVCROA_DSA attained a recall of 94.387%, which is the maximum as compared to the recall recorded by other existing detection models, such as RHS-RBM, GAN, Neural Network, FS-WOA-DNN, and VCROA-DNFN at 89.765%, 84.012%, 81.072%, 78.549%, and 91.673% for a K-fold value of 8. Thus, the FVCROA_DSA obtained a superior performance of 10.99% in terms of recall as compared with the existing GAN technique. The results obtained from the experiment by the DDoS attack detection schemes by utilizing precision are exhibited in Figure 7(b). The precision recorded by the DDoS attack detection techniques is 75.186% by FS-WOA-DNN, 78.454% by Neural Network, 81.072% by GAN, 83.645% by RHS-RBM, 89.745% by VCROA-DNFN, and 93.0136% by FVCROA_DSA for a K-fold 8.



(a)

(b)

(c)

**Figure 7.** Comparative performance of FVCROA_DSA and existing models on the DDoS Attack dataset under 8-fold cross-validation: (a) Recall, (b) Precision, and (c) F-measure

Here, the high performance of 3.52% is recorded by FVCROA_DSA as compared with the VCROA-DNFN approach. Also, Figure 7(c) shows the results recorded by various detection models while validating using F-measure. The FVCROA_DSA measured a maximum F-measure of 93.696% for a K-fold of 8. Likewise, the F-measure obtained by other DDoS attack detection schemes, FS-WOA-DNN is 76.830%, RHS-RBM is 86.597%, GAN is 82.516%, Neural Network is 79.742%, and VCROA-DNFN is 90.699%. According to the findings, the FVCROA_DSA model outperformed the current FS-WOA-DNN by 18%.

### 4.6 Comparative discussion

In Table 2, the results obtained by the FVCROA_DSA model and other prevailing DDoS attack detection techniques by considering the BOT-IoT database and DDoS Attack database by varying K-fold and learning data are portrayed. The observations show that the FVCROA_DSA attained high experimental results with a recall of 94.827%, precision of 93.857%, and F-measure of 94.340% for learning data of 90%. Likewise, the recall recorded by prevailing schemes, like GAN is 86.875%, Neural Network is 84.551%, FS-WOA-DNN is 80.276%, RHS-RBM is 90.346%, and VCROA-DNFN is 92.548%. Also, the existing approaches, such as RHS-RBM, GAN, Neural Network, FS-WOA-DNN, and VCROA-DNFN obtained precision of 88.103%,84.275%, 82.367%, 77.945%, and 90.245% and F-measure of 89.210%,85.555%, 83.445%, 79.093%, and 91.382%. The consistent superiority of the proposed FVCROA_DSA across both datasets can be attributed to its hybrid optimization and deep feature-learning design. The SVM-RFE component efficiently eliminates redundant features, while the FVCROA optimizer fine-tunes the DSA parameters to achieve a balanced precision–recall trade-off and rapid convergence. Incorporating Fractional Calculus further stabilizes the optimization process, yielding smoother loss surfaces and improved generalization compared with traditional metaheuristic models. It is observed that the FVCROA DSA technique suggested is better than existing schemes used to make comparisons with the technique in the detection of DDoS attacks. In terms of detecting DDoS attacks based on the information packets found in the BOT-IoT and DDoS attack databases, the FVCROA_DSA is more dependable and highly generalizable. Also, the DSA model reduced the size of the data by removing the most significant information contained in the data packets. As well, at lower values of the computational complexity of problems, the FVCROA model that was developed to optimize the hyperparameters of DSA quickly converges to the optimal value of the results.

**Table 2**. Comparative discussion

| Variations | Metrics | FS-WOA-DNN | Neural Network | GAN | RHS-RBM | VCROA-DNFN | Proposed FVCROA_DSA |
|---|---|---|---|---|---|---|---|
| *For BOT-IoT database* | | | | | | | |
| Learning data of 90% | *Recall* | 78.546 | 80.597 | 84.75 | 86.379 | 91.072 | 94.275 |
| | *Precision* | 79.548 | 83.102 | 85.794 | 89.734 | 91.285 | 93.985 |
| | *F-measure* | 79.044 | 81.830 | 85.268 | 88.024 | 91.179 | 94.130 |
| K-fold value of 8 | *Recall* | 76.125 | 78.595 | 83.457 | 85.279 | 91.006 | 94.037 |
| | *Precision* | 78.596 | 82.075 | 84.012 | 88.727 | 90.434 | 92.904 |
| | *F-measure* | 77.341 | 80.297 | 83.733 | 86.969 | 90.719 | 93.467 |
| *For DDoS Attack dataset* | | | | | | | |
| Learning data of 90% | *Recall* | 80.276 | 84.551 | 86.875 | 90.346 | 92.548 | *94.827* |
| | *Precision* | 77.945 | 82.367 | 84.275 | 88.103 | 90.245 | *93.857* |
| | *F-measure* | 79.093 | 83.445 | 85.555 | 89.210 | 91.382 | *94.340* |
| K-fold value of 8 | *Recall* | 78.549 | 81.072 | 84.012 | 89.765 | 91.673 | 94.387 |
| | *Precision* | 75.186 | 78.454 | 81.072 | 83.645 | 89.745 | 93.016 |
| | *F-measure* | 76.830 | 79.742 | 82.516 | 86.597 | 90.699 | 93.696 |

The conclusions of the FVCROA DSA on identifying DDoS attacks based on packet data were encouraging. Overall, the experimental results confirm that the suggested FVCROA DSA model is state-of-the-art in terms of accuracy and robustness for detecting DDoS attacks. The architecture of its MapReduce system enables easy deployment across large networks, and the hybrid optimization mechanism improves learning stability and detection accuracy. The reliability and efficiency of the model with respect to real-world intrusion detection can be demonstrated by the consistency of its results across both benchmark datasets.

## 5. Conclusion

Various security mechanisms have been used to prevent and detect cyberattacks across the internet. Among them, Distributed Denial of Service (DDoS) is one of the most common and devastating types of attack, as it enables attackers to flood targeted networks with unnecessary traffic. To overcome this issue, a new optimization-based deep learning model, FVCROA_DSA, was developed to effectively detect DDoS attacks in a MapReduce system using the mapper and reducer. The input data were first processed using the mean substitution technique. Then, features were selected using SVM-RFE in the mapper stage, with the SVM parameter optimized using the VCROA algorithm. The identified features were then run through the DSA model during the reducer stage, with hyperparameters optimized using the proposed FVCROA technique to achieve the best possible detection performance. Experimental analyses showed that the FVCROA framework with the DSA achieved higher detection accuracy, with a recall of 94.827, a precision of 93.857, and an F-measure of 94.340, compared to existing machine learning and deep learning models. The framework can be expanded to real-time and streaming networks in the future to increase the scalability and flexibility of the detection process. Moreover, multiclass classification strategies that incorporate large, labelled datasets will enable the system to detect and classify various forms of cyberattacks beyond DDoS, thereby enhancing its resilience and generalization in next-generation intrusion detection systems.

**Acknowledgements**

The authors would like to express their sincere gratitude to the Department of MCA, Bharati Vidyapeeth's Institute of Management & Information Technology, Navi Mumbai, for providing the necessary facilities to carry out this research. Special thanks are extended to my guide, Dr. Suhasini Vijayakumar, Principal, Bharati Vidyapeeth's Institute of Management & Information Technology, Navi Mumbai, and Dr. Priya Chandran, Associate Professor, Bharati Vidyapeeth's Institute of Management & Information Technology, Navi Mumbai, for their valuable guidance and contributions throughout this study.

**Ethical issue**

The authors are aware of and comply with best practices in publication ethics, specifically regarding authorship (avoidance of guest authorship), dual submission, manipulation of figures, competing interests, and compliance with research ethics policies. The authors adhere to publication requirements that the submitted work is original and has not been published elsewhere.

**Data availability statement**

The manuscript contains all the data. However, more data will be available upon request from the authors.

**Conflict of interest**

The authors declare no potential conflict of interest.

## References

[1] S. Sumathi and N. Karthikeyan, “Detection of distributed denial of service using deep learning neural network,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 6, pp. 5943–5953, 2021. https://doi.org/10.1007/s12652-020-02144-2

[2] A. Salem and M. A. Mohammed, “DoSTDM: A denial of service detection model using firewall data traffic pattern matching,” Doctoral dissertation, Curtin University, 2013

[3] A. Sharma and A. Bhasin, “Critical investigation of denial of service and distributed denial of service models and tools,” in Proceedings of the 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 546–550, IEEE, Oct. 2018. doi: 10.1109/ICACCCN.2018.8748468.

[4] BBC News, "https://www.bbc.com/news/technology-53093611" accessed on August 2024.

[5] S. Sumathi and R. Rajesh, “Comparative study on TCP SYN flood DDoS attack detection: a machine learning algorithm based approach,” WSEAS Transactions on Systems and Control, vol. 16, pp. 584–591, 2021. Doi: 10.37394/23203.2021.16.54

[6] D. Akgun, S. Hizal and U. Cavusoglu, “A new DDoS attacks intrusion detection model based on deep learning for cybersecurity,” Computers & Security, vol. 118, p. 102748, 2022. https://doi.org/10.1016/j.cose.2022.102748

[7] M. S. Elsayed, N. A. Le-Khac, S. Dev and A. D. Jurcut, “Network anomaly detection using LSTM based autoencoder,” in Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, pp. 37–45, Nov. 2020.

[8] A. A. Diro and N. Chilamkurti, “Distributed attack detection scheme using deep learning approach for Internet of Things,” Future Generation Computer Systems, vol. 82, pp. 761–768, 2018. https://doi.org/10.1016/j.future.2017.08.043

[9] D. Gümüşbaş, T. Yıldırım, A. Genovese and F. Scotti, “A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems,” IEEE Systems Journal, vol. 15, no. 2, pp. 1717–1731, 2020. doi: 10.1109/JSYST.2020.2992966.

[10] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May and A. Lakhina, “Impact of packet sampling on anomaly detection metrics,” in Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, pp. 159–164, Oct. 2006.

[11] B. Zhou, J. Li, Y. Ji and M. Guizani, “Online internet traffic monitoring and DDoS attack detection using Big Data frameworks,” in Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), pp. 1507–

1512, IEEE, June 2018. doi: 10.1109/IWCMC.2018.8450335.

[12] D. Han, K. Bi, H. Liu and J. Jia, “A DDoS attack detection system based on spark framework,” Computer Science and Information Systems, vol. 14, no. 3, pp. 769–788, 2017. https://doi.org/10.2298/CSIS161217028H

[13] P. Medeira, J. Grover and M. Khorjiya, “A survey on detecting application layer DDoS using big data technologies,” Journal of Emerging Technologies and Innovative Research (JETIR), 2019.

[14] N. A. Azeez, T. J. Ayemobola, S. Misra, R. Maskeliūnas and R. Damaševičius, “Network intrusion detection with a hashing based apriori algorithm using Hadoop MapReduce,” Computers, vol. 8, no. 4, p. 86, 2019. https://doi.org/10.3390/computers8040086

[15] M. J. Awan, U. Farooq, H. M. A. Babar, A. Yasin, H. Nobanee, M. Hussain et al., “Real-time DDoS attack detection system using big data approach,” Sustainability, vol. 13, no. 19, p. 10743, 2021. https://doi.org/10.3390/su131910743

[16] H. A. Afolabi and A. A. Aburas, “RTL-DL: a hybrid deep learning framework for DDoS attack detection in a big data environment,” International Journal of Computer Networks and Communications (IJCNC), vol. 14, no. 6, pp. 51–66, 2022. DOI:10.5121/ijcnc.2022.14604

[17] B. Zhou, J. Li, J. Wu, S. Guo, Y. Gu and Z. Li, “Machine-learning-based online distributed denial-of-service attack detection using spark streaming,” in Proceedings of the 2018 IEEE International Conference on Communications (ICC), pp. 1–6, IEEE, May 2018. doi: 10.1109/ICC.2018.8422327.

[18] D. Saraladevi and V. K. P. Scholar, “Big Data Analytics Framework for Peer-to-Peer Botnet Detection Using Random Forest and Deep Learning,” International Journal of Computer Science and Information Security (IJCSIS), vol. 15, no. 11, 2017. https://doi.org/10.1016/j.ins.2014.03.066

[19] K. Singh, S. C. Guntuku, A. Thakur and C. Hota, “Big data analytics framework for peer-to-peer botnet detection,” Network, vol. 3, 2014. https://doi.org/10.1016/j.ins.2014.03.066

[20] A. Patcha and J. M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” Computer Networks, vol. 51, no. 12, pp. 3448–3470, 2007. https://doi.org/10.1016/j.comnet.2007.02.001

[21] S. Alzahrani and L. Hong, “A survey of cloud computing detection techniques against DDoS attacks,” Journal of Information Security, vol. 9, no. 1, p. 45, 2017. DOI: 10.4236/jis.2018.91005

[22] J. Koo, G. Kang and Y. G. Kim, “Security and privacy in big data life cycle: a survey and open challenges,” Sustainability, vol. 12, no. 24, p. 10571, 2020. https://doi.org/10.3390/su122410571

[23] J. Esmaily, R. Moradinezhad and J. Ghasemi, “Intrusion detection system based on multi-layer perceptron neural networks and decision tree,” in Proceedings of the 2015 7th Conference on Information and Knowledge Technology (IKT), pp. 1–5, IEEE, May 2015. doi: 10.1109/IKT.2015.7288736.

[24] C. Yin, Y. Zhu, J. Fei and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” IEEE Access, vol. 5, pp. 21954–21961, 2017. doi: 10.1109/ACCESS.2017.2762418.

[25] S. Haider, A. Akhunzada, G. Ahmed and M. Raza, “Deep learning based ensemble convolutional neural network solution for distributed denial of service detection in SDNs,” in Proceedings of the 2019 UK/China Emerging Technologies (UCET), pp. 1–4, IEEE, Aug. 2019. doi: 10.1109/UCET.2019.8881856.

[26] M. Roopak, G. Y. Tian and J. Chambers, “Deep learning models for cyber security in IoT networks,” in Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 452–457, IEEE, Jan. 2019. doi: 10.1109/CCWC.2019.8666588.

[27] A. Agarwal, M. Khari and R. Singh, “Detection of DDoS attack using deep learning model in cloud storage application,” Wireless Personal Communications, vol. 127, no. 1, pp. 419–439, 2022. https://doi.org/10.1007/s11277-021-08271-z

[28] Y. Ding and Y. Zhai, “Intrusion detection system for NSL-KDD dataset using convolutional neural networks,” in Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, pp. 81–85, Dec. 2018.

[29] M. B. Anley, A. Genovese, D. Agostinello and V. Piuri, “Robust DDoS attack detection with adaptive transfer learning,” Computers & Security, vol. 144, p. 103962, 2024. https://doi.org/10.1016/j.cose.2024.103962

[30] S. Aktar and A. Y. Nur, “Towards DDoS attack detection using deep learning approach,” Computers & Security, vol. 129, p. 103251, 2023. https://doi.org/10.1016/j.cose.2023.103251

[31] M. Mittal, K. Kumar and S. Behal, “Deep learning approaches for detecting DDoS attacks: A systematic review,” Soft Computing, vol. 27, no. 18, pp. 13039–13075, 2023. https://doi.org/10.1007/s00500-021-06608-1

[32] A. E. Cil, K. Yildiz and A. Buldu, “Detection of DDoS attacks with feed forward based deep neural network model,” Expert Systems with Applications, vol. 169, p. 114520, 2021. https://doi.org/10.1016/j.eswa.2020.114520

[33] M. P. Novaes, L. F. Carvalho, J. Lloret and M. L. Proença Jr, “Adversarial deep learning approach detection and defense against DDoS attacks in SDN environments,” Future Generation Computer Systems, vol. 125, pp. 156–167, 2021. https://doi.org/10.1016/j.future.2021.06.047

[34] S. Sumathi, R. Rajesh and S. Lim, “Recurrent and deep learning neural network models for DDoS attack detection,” Journal of Sensors, vol. 2022, no. 1, p. 8530312, 2022. https://doi.org/10.1155/2022/8530312

[35] Y. Zhang, Q. Deng, W. Liang and X. Zou, “An efficient feature selection strategy based on multiple support vector machine technology with gene expression data,” BioMed Research International, vol. 2018, no. 1, p. 7538204, 2018. https://doi.org/10.1155/2018/7538204

[36] H. Jia, X. Peng and C. Lang, “Remora optimization algorithm,” Expert Systems with Applications, vol. 185, p. 115665, 2021. https://doi.org/10.1016/j.eswa.2021.115665

[37] Y. Yu, J. Li, J. Li, Y. Xia, Z. Ding and B. Samali, “Automated damage diagnosis of concrete jack arch beam using optimized deep stacked autoencoders and multi-sensor fusion,” Developments in the Built Environment, vol. 14, p. 100128, 2023. https://doi.org/10.1016/j.dibe.2023.100128

[38] P. R. Bhaladhare and D. C. Jinwala, “A clustering approach for the l-diversity model in privacy preserving data mining using fractional calculus-bacterial foraging optimization algorithm,” Advances in Computer Engineering, vol. 2014, no. 1, p. 396529, 2014. https://doi.org/10.1155/2014/396529

[39] The BoT- IoT dataset is taken from https://ieee-dataport.org/documents/bot-iot-dataset accessed on July 2024.

[40] The DDoS attack dataset is taken from https://ieee-dataport.org/documents/ddos-attack-dataset#files accessed on July 2024