International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I3P127

Volume 74 Issue 3, 388-407, March 2026

Original Article

VCROA-DNFN: A Big Data Approach in MapReduce Framework for DDoS Attack Detection using Optimized Deep Neuro Fuzzy Network

Rahul Vijay Kotawadekar[1*], Suhasini Vijaykumar[2], Priya Chandran[3]

[1,2,3]*Department of Computer Applications, Bharati Vidyapeeth's Institute of Management & Information Technology, Sector 8, C.B.D.Belapur, Navi Mumbai, Maharashtra, India.*

[1]*Corresponding Author : rahulkotawadekar457@gmail.com*

Received: 22 December 2025 Revised: 24 January 2026 Accepted: 06 February 2026 Published: 28 March 2026

Abstract *- DDoS attacks have emerged as a major menace to network security, which is extremely challenging to privacy and service provision. Although different methodologies have been established to detect attacks and prevent their outcome, the methodologies remain ineffective in offering effective identification accuracy because of the high rate of false alarms. This paper will overcome these shortcomings by introducing an optimized version of deep learning methodology called Velocity Contour-based Remora Optimization Algorithm (VCROA)-Deep Neuro Fuzzy network (DNFN), which can be used to identify DDoS attacks in a MapReduce big data system. The VCROA algorithm incorporates the velocity contour mechanism in the Remora Optimization Algorithm (ROA) to choose the best features and optimize the weights of the network to increase the learning ability of the DNFN classifier. The experimental findings show that the proposed VCROA-DNFN to detect DDoS attacks has achieved an optimal accuracy, recall, and F-measure of 91.10, 93.70, and 91.70. The large values of precision, recall, and F-measure indicate that the proposed model is strong and consistent in the detection of DDoS attacks.*

Keywords *- Deep Neuro Fuzzy Network (DNFN), MapReduce, Velocity Contour-based Remora Optimization Algorithm (VC-ROA), Intrusion Detection System (IDS), Distributed Denial of Service (DDoS).*

1. Introduction

Network security has become an issue of concern with the fast development of network-based services and the growing volume of confidential information being transferred across computer networks. Intrusion detection methods [27] are critical towards securing systems against cyber-attacks that cannot be completely dealt with using conventional security systems like firewalls and access control policies. Although there are a number of intrusion prevention methods available, it is still very difficult to safeguard networked systems against advanced attacks. Intrusion Detection Systems (IDSs) are therefore an essential point of protection in the contemporary network security systems [14, 28].

The use of network security is prevalent in many fields, such as in the personal computing environment, in the military, and in the organizational network. The advent of the internet has heightened the privacy issues, and to create effective security mechanisms, a better understanding of the attack mechanisms is required [15]. Generally, network security is the process of regulating access to information resources. Firewalls, though, have an authorization policy that limits the services available to the users of the network, but in most cases, they cannot stop the spread of malicious content like worms and Trojans over the legitimate communication channels. In this regard, antivirus programs and Intrusion Detection Systems (IDSs) are complementary in that they track the behavior of the network and detect possible attacks [16].

Network Intrusion Detection Systems (NIDSs) form one of the most important elements in the network management systems, which allow security threats in institutional networks to be detected. A NIDS is used to monitor both inbound and outbound traffic and send an alert in case of abnormal or malicious traffic. Nevertheless, creating a successful NIDS against the unknown attacks or the emerging attacks is not without a number of challenges. The selection of relevant and discriminative features among high-dimensional network datasets is one of the primary problems, as it directly influences the detection accuracy. The second issue is that there is limited availability of properly labeled traffic measurements of real-world network settings, which limits the training of credible detection models [11, 17]. The complexity of the network traffic can cause the samples of intrusion to be overwhelmed by the high number of legitimate samples,

leading to imbalanced datasets that can reduce the performance of the detection and also increase the rates of false alarms. As the internet-based services continue to grow at a very alarming rate, intrusion detection has taken the form of a precondition to ensure that the services are available and that the economic losses incurred by cyber-attacks are minimal. Subsequently, network intrusion detection has become a significant field of study in the discipline of network security [12, 18].

To address the drawbacks of conventional methods, many different Deep Learning (DL) models have been suggested as intrusion detectors in recent years. In [19], a deep learning model was created to identify and prevent Hello flood attacks that are based on DoS in clinical IoT networks. In [20], a Long Short-Term Memory (LSTM) network was used to detect DDoS attacks at the control layer of Software Defined Networks (SDNs), and this improved the security of cloud and fog computing systems.

In [11], a Deep Convolutional Neural Network (DCNN) was proposed to identify DDoS attacks in optical switching networks. In addition, a hybrid intrusion detection model with Non-symmetric Deep Autoencoder (NDAE) and a Random Forest (RF) classifier was suggested to enhance SDN security [12]. Despite the high performance of deep learning methods in detecting samples, recent reports reveal that these algorithms are susceptible to adversarial samples developed by malicious agents that can result in the misclassification of samples. To overcome this weakness, adversarial training methods have been investigated to incorporate adversarial samples during training. In this respect, Generative Adversarial Networks (GANs) have been deployed to produce adversarial traffic patterns, and thus boost the resilience of NIDSs to adversarial DDoS attacks [3].

2. Motivation / Problem Statement

With the fast development of cloud computing and large-scale networked systems, Distributed Denial of Service (DDoS) attacks have become more sophisticated. DDoS attack detection is designed to detect abnormal traffic flow and block malicious flooding attacks that reduce service availability. Despite the many proposed detection techniques, there are several crucial challenges that have not been solved.

One significant drawback of current DDoS detection techniques is that most of them are not capable of being generalized to previously unseen attacks because most of the models are trained and tested on a single dataset. As a result, detection performance tends to decrease when such models are subjected to changing or cross-domain traffic patterns. Besides, some sophisticated methods have high computation costs because of the complicated feature selection and parameter optimization, limiting their large-scale or real-time usability. Furthermore, the inflexible classification schemes adopted by most of the classical schemes are susceptible to noise and uncertainty in real-life network traffic, which results in higher false alarms and unreliable detection outcomes. To address these issues, the VCROA-DNFN framework is created as an effective and powerful DDoS detection tool. The proposed method combines feature selection based on Velocity Contour-based Remora Optimization Algorithm (VCROA) and an adaptive Deep Neuro-Fuzzy Network (DNFN) classifier. VCROA is more generalized and less costly to compute, using optimal features and tuning model parameters, and the fuzzy inference mechanism in DNFN is more resilient to noisy and uncertain traffic. The proposed model can be used in this integrated design to overcome scalability, generalization, and robustness constraints that exist in the current DDoS detection methods.

3. Contribution

The core contribution of this paper is as follows:

An effective VCROA-DNFN-based framework is created to detect a DDoS attack in a MapReduce environment, which can process large-scale network traffic data with scalability.

A Velocity Contour-based Remora Optimization Algorithm (VCROA), which is an extension of the velocity contour concept to ROA, is used to identify the best features and optimize the hyperparameters of the Deep Neuro-Fuzzy Network (DNFN), which lowers the computational cost and enhances the detection performance.

DNFN classifier includes fuzzy inference to deal with uncertainty and noise in network traffic efficiently, leading to a high degree of robustness and stable DDoS attack detection.

The suggested method is confirmed by the extensive experiments, such as comparative analysis, cross-dataset analysis, and statistical performance test on standard detection measures.

The rest of this paper is structured in the following manner. Section 2 conducts a literature review on DDoS attack detection. Section 3 outlines the suggested VCROA-DNFN approach. The results of the experiment are mentioned and discussed in Section 4. Section 5 is the conclusion of the paper and the future research directions.

4. Literature Survey

This section gives a description of the available methods of detecting DDoS attacks and explains their advantages and limitations.

4.1. Optimization-Based Methods

Agarwal, A. et al. [1] suggested a Whale Optimization Algorithm with a Feature Selection-based Algorithm (FS-WOA-DNN) that is integrated with a Deep Neural Network to detect DDoS attacks. The technique was accurately detected

with an efficient implementation. Nevertheless, it was tested with small datasets, which limited its capacity to be applied to unknown and changing attack patterns.

Ahmed Jamal Ibrahim et al. [31] designed a feature-optimized machine learning architecture to detect and mitigate DDoS attacks, which is capable of balancing the accuracy of detection and the computation costs. However, the lack of multi-dataset validation did not allow the evaluation of its strength in different network settings.

Novaes, M.P. et al. [3] proposed an SDN-based DDoS detection system based on Generative Adversarial Network (GAN) to increase resistance to adversarial manipulation and improve detection results. Although these benefits were present, the strategy was tested in a small-scale simulated setting and was based on one deep learning architecture, which limited its applicability to large, heterogeneous network conditions.

4.2. CNN-Based Methods

Haizhen Wang et al. [5] introduced a Convolutional Long Short-Term Memory Network that has multi-head attention and three-way decision (ConvLSTM-MHA-TWD) to detect DDoS attacks. This model was better at detecting; it used attention and decision mechanisms, but it had a long training period and had not been validated in real-world SDN environments.

Doriguzzi-Corin, R. et al. [6] proposed LUCID, a lightweight deep learning-based model of DDoS detection that showed the ability to detect in real-time and with high resource efficiency. In spite of these benefits, the model demonstrated poor generalization to hidden and real-world attack patterns.

Elsaeidy, A.A. et al. [7] introduced a hybrid deep learning model that combines a Restricted Boltzmann machine and deep CNN (RBM+deep CNN) to detect replay and DDoS attacks in smart cities. Despite the method possessing high detection accuracy and the ability to support complex data distributions, it had not been deployed and tested in real-world large-scale security platforms.

Cil, A.E. et al. [4] created A Deep Neural Network (DNN)-based DDoS attack detector with high accuracy and cross-dataset generalization; its validation, however, was done on offline tests but not on real-time traffic validation.

4.3. Other DDoS Attack Detection Methods

Rana Abu Bakar et al. [2] suggested a hierarchical traffic representation and a Graph Neural Network (GNN) with ensemble learning based DDoS detection model, called FTG-Net-E. The model was found to have strong detection performance and lower error rates of target and non-target traffic. Nevertheless, it was only restricted to single-attack cases and failed to use contextual data like network topology and previous traffic patterns.

Premkumar, M. and Sundararajan, T.V.P. [8] proposed a lightweight Deep Learning based Defense Mechanism (DLDM) to detect DoS attacks at the Data Forwarding Phase. Despite the model recording high levels of detection and enhanced resilience, it was tested in controlled settings and was not tested on real-world data. The Efficient Gated Recurrent Unit GhostNet (EffiGRU-GhostNet) architecture of DDoS attack detection was proposed by Abdulrahman A. Alshdadi et al. [30], with low computational complexity and good detection. However, the model was not very effective in detecting invisible forms of attack, and this lowered its flexibility to changing cyber attacks.

The current DDoS attack detection methods are still confronted with issues of limited generalization, high computation cost, and vulnerability to dynamic and noisy network traffic. Such constraints encourage the creation of a powerful detection system capable of effectively identifying meaningful characteristics, evolving traffic patterns, and scaling to large network settings.

Here, the feature selection based on the VCROA and the fuzzy inference ability of the DNFN allow effective identification of the unseen DDoS attacks, minimize the computational load, and enhance the stability in the presence of real-world traffic.

4.4. Proposed VCROA-DNFN for DDoS Attack Detection in the MapReduce Framework

The VCROA-DNFN model proposed identifies DDoS attacks in the MapReduce system by use of a structured workflow. Firstly, the network traffic logs of the datasets [9] [29] are gathered and fed into the pre-processing phase, where missing values are replaced by mean substitution to have clean data. The processed data is then input into the MapReduce paradigm that contains a number of mappers and reducers in the reducer phase.

At the mappers, the feature selection mechanism is conducted utilizing VCROA to choose the appropriate features. These selected features are aggregated in the reducer phase and fed into the DNFN. Finally, the DDoS attack detection is accomplished by DNFN.

The DNFN is trained and fine-tuned using VCROA to enhance accuracy, robustness, and generalization to unseen attack patterns.

The system outputs traffic classification as benign or malicious. The pictorial illustration of VCROA-DNFN for DDoS attack identification in the MapReduce paradigm is depicted in Figure 1.

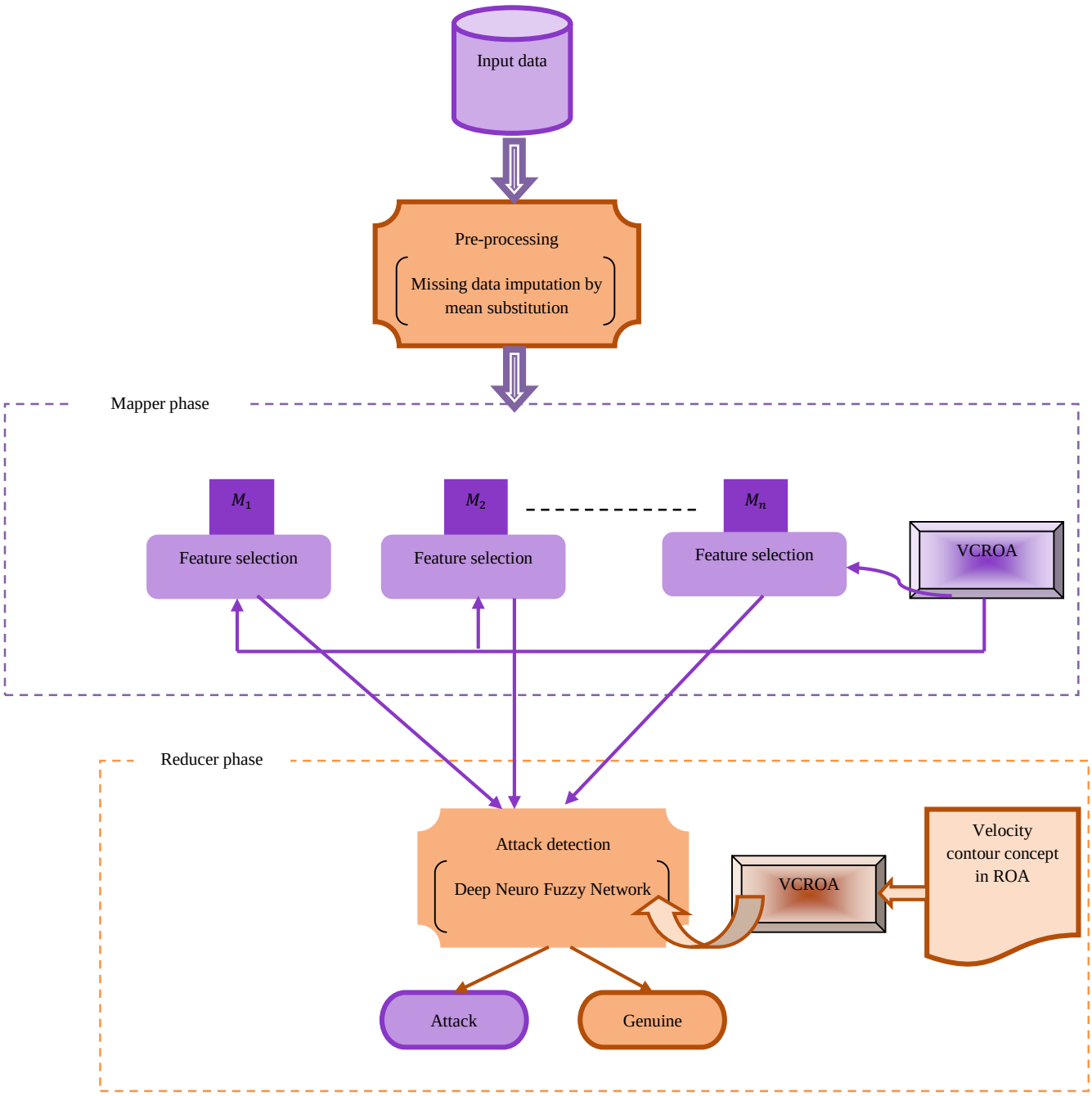


Fig. 1 Pictorial representation of the designed VCROA-DNFN for DDoS attack detection in the MapReduce framework

4.5. Acquisition of Input Data

The initial step is the acquisition of input data specified in [9], and this repository consists of several data samples, which are expressed in the form of:

$$I = \{d_1, d_2, .. d_i, .., d_j\} \quad (1)$$

Here, I indicates the input log file that consists of j a number of data samples. Moreover, i^{th} data in the repository is signified as d_i.

4.6. Data Pre-Processing by Missing Data Imputation

Missing data imputation handles incomplete and missing values in the network traffic dataset, which is common due to packet loss or errors. The chief contribution of imputation is not just to determine what is lost, but to reconstruct the significant characteristics of the database as a whole [22]. Missing At Random (MAR) is the widely utilized scenario in practice, and so far, many methods have been developed to overcome the MAR data gap, such as zero or mean value substitution. The d_i data is allowed to the pre-processing phase, where missing data are filled using mean-based

substitution, assigning the attribute's average in place of missing entries. By replacing missing values using mean substitution, the dataset becomes complete and consistent, which ensures that subsequent feature selection and DNFN training are not biased. The result of the pre-processing step is signified as P_i. This contributes to efficient and accurate DDoS attack detection, as the model can learn from a reliable dataset without being affected by inconsistencies in the input data.

4.7. MapReduce Framework

The MapReduce framework [23] comprises a single master JobTracker and one slave per cluster node. The slaves perform the tasks as instructed by the master and offer task-status data to the master continuously. At the mapper side, a feature selection mechanism is performed at each and every mapper using the proposed VCROA. The results from all the mappers are aggregated at the reducer phase to detect the DDoS attack using the same proposed VCROA.

4.8. Mapper Stage

A map factor is performed over a single record and creates a set of median records, which are in the form of key pairs. There are na number of mappers and their expression is illustrated as,

$$M = \{M_1, M_2, \ldots M_i, \ldots, M_n\} \quad (2)$$

The input $P_i$having the dimension $\kappa \times a$is applied over the mapper side, and the mappers individually perform the feature selection stage.

4.9. Feature Selection using the Proposed VCROA

It is very easy to interpret and provides very short training times. Moreover, it significantly eliminates the problem of dimensionality. In this, feature selection is accomplished by employing VCROA, which is a consolidation of the velocity contour-based concept into ROA.

The output attained from $i^{th}$the mapper after the feature selection process is denoted as $F_i$having the dimension $\kappa \times b$, where $a > b$. The overall output of the feature selection process is expressed as,

$$F = \{F_1, F_2, .. F_i, \ldots, F_n\} \quad (3)$$

Here, F_n represents the features selected by the n^{th} mapper, and F_i represents the feature selected by the i^{th}mapper.

4.10. Remora Position Encoding

The significant use of this encoding is to select appropriate features among the total number of features that exist. Figure 2 presents the structure used for encoding the solution.

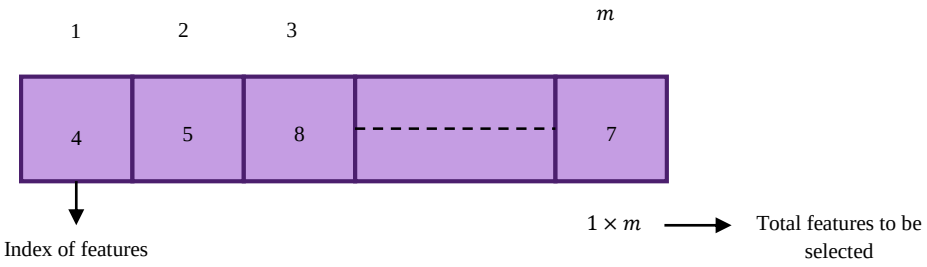


Fig. 2 Solution encoding

4.11. Finding an Objective Factor

The major aim of this is to select the good features for the attack detection process, and it is evaluated with Canberra distance, which is given in the form of an expression as,

$$D(F, E) = \sum_{i=1}^{n} \frac{|F_i - E_i|}{|F_i| + |E_i|} \quad (4)$$

Here, F_i denotes the selected feature by $i^{th}$the mapper, whereas E_i denotes the target feature.

5. Algorithmic Steps

ROA [10] is a nature-inspired metaheuristic algorithm that mimics the parasitic behaviour of remora. Remora is a suckfish, and it is eminent for its high capability of swimming on whales or other marine organisms as hosts. Generally, it has a prolonged body with a flat head, and its suckers are situated on its first dorsal fin. Usually, it is a parasite, and it is fully dependent on other organisms for survival. When it arrives in the sea area, it will detach from the host, swallow food, and become a new host to other organisms, and continue to move to another sea zone. This characteristic is mainly because of the escaping nature of remora from the enemy's attack.

Step 1: Initialization

Consider the remora population in a G search zone with ua number of populations, and it is expressed by,

$$R_u = (R_{u1}, R_{u2}, \ldots\ldots, R_{uk}) \quad (5)$$

where u implies the count of remora and $k$symbolizes the solution dimension G of remora. The remora vector varies with regard to variant sizes. Likewise, R_{best} shows the food target and it is expressed as,

$$R_{best} = (R_1^*, R_2^*, \ldots, R_u^*) \quad (6)$$

Step 2: Calibrate Fitness Factor

The pivotal point of the fitness solution is to provide the good features, which are the finest solution of the whole algorithm, and it is already calculated based on Eq. (4).

Step 3: Exploration Stage

Various kinds of marine organisms are considered as driving constituents to assist remora in determining the optimal position. In this algorithm, two phases are utilized: the exploration and exploitation stages. Both stages include two main mechanisms, which are elaborated separately in the section below.

5.1. Sail Fish Optimizer (SFO) Procedure

If the remora is linked to the swordfish for support, its level is enhanced at the same moment. According to the supreme idea of this algorithm, the place upgrade is improved, and it is mathematically derived as follows,

$$R_u^{\tau+1} = R_{best}^{\tau} - \left(rand(0,1) * \left(\frac{R_{best}^{\tau} + R_{rand}^{\tau}}{2}\right) - R_{rand}^{\tau}\right) \quad (7)$$

where τ signifies the number of present iterations, and the maximum count of repetitions is implied as T. Moreover, R_{rand} represents the undisturbed place. The choosing criterion of remora for a diverse host is highly based on whether it has consumed the food or not, or whether the current objective function achieved is better than that of the earlier function. In general, the latest fitness value is attained through experience attack.

5.2. Experience an Attack

To change the host, the tuyu is persistently making a tiny step enclosing the host, similar to that of the agglomeration of experience. The above ideas are mathematically modeled as,

$$R_{att} = R_u^{\tau} + \left(R_u^{\tau} - R_{pre}\right) * randl \quad (8)$$

Here, R_{pre} and R_{att} shows the place of the preceding generation and tentative step, respectively. When remora decides to perform such an active movement, it can be referred to as "small global movement" and thus, $randl$was chosen.

Step 4: Exploitation Stage

The next stage is the eat thoughtfully stage, which contains binary processes, like WOA and host feeding strategy.

5.3. Whale Optimization Algorithm (WOA) Procedure

According to the WOA, the location upgrading expression of remora connected to the whale was refined, and it is expressed using the following expressions,

$$R_{u+1} = A * e^{\aleph} * cos(2\pi\alpha) + R_u \quad (9)$$

$$\alpha = rand(0,1) * (\aleph - 1) + 1 \quad (10)$$

$$\aleph = -\left(1 + \frac{\tau}{T}\right) \quad (11)$$

$$A = |R_{best} - R_u| \quad (12)$$

In a broad solution area, if the remora is on a whale, their place can be referred to as the same. Here, $A$the distance between the hunter and prey $\alpha$is an undisturbed measure that hangs in range $[-1,1]$ and deteriorates linearly to $[-2,-1]$.

6. Host Feeding

Host feeding is a component of the exploitation process. In this step, the dimensional space is reduced and concentrated around the host region. The mathematical steps of moving on or around the host are stated below,

$$R_u^{\tau} = R_u^{\tau} + B \quad (13)$$

$$B = C * (R_u^{\tau} - \beta * R_{best}) \quad (14)$$

$$C = 2 * S * rand(0,1) - S \quad (15)$$

$$S = 2 * \left(1 - \frac{\tau}{max\_iter}\right) \quad (16)$$

Here, B it was employed to express the tiny step of motion which corresponds to the volume area$S$of the host and remora. A remora function β is used to decrease the remora's position.

6.1. Velocity-Contour-based Concept

The update solution completely depends upon the velocity of the remora towards the host. According to the velocity of the remora, the idea for the remora's position is mathematically derived as follows,

$$R_u^{\tau+1} = R_u^{\tau} + V * \beta * R_{best} + C \quad (17)$$

Here, the volume space $S$and remora factor $\beta$are already expressed in Eq. (16) and Eq. (14).

Step 5: Termination

The aforementioned procedures are circulated until they reach a satisfactory solution for an optimal set of features. The pseudo-code implementation of the proposed VCROA is provided in Algorithm 1.

Table 1. Pseudo code of VCROA

SL. No	Pseudo code of VCROA
1	Input:$R_u(\tau)$,A, $R_{best}(\tau)$
2	Output:$R_u(\tau+1)$
3	Provoke the memory location R_{pre}
4	Provokethe finest solution R_{best}
5	While$\tau < Max_iter$do
6	Evaluate the objective factor using Eq. (4)
7	Ensure that any candidate exceeds the search zone and discard it.
8	Renew$\aleph, \alpha$ and S
9	for the individual remora represented by $u$do
10	if$O(u) = 0$then
11	Renew the place of dependent whales utilizing Eq. (9)
12	Else
13	if$O(u) = 1$then
14	Update the place of the attached sail fish using Eq. (7)
15	end if
16	Initialize one-step prediction utilizing Eq. (8)
17	Calibrate the value of $O(u)$
18	If the host is not replaced, Eq. (13) is used as the host feeding mode for remora.
19	Else
20	Update the solution of remora based on the velocity contour concept using Eq. (17)
21	end for
22	end while

7. Reducer Phase

The reduce side is also termed the reducer stage, and it is performed to separate a pool of median records with the same key and create a group of outcome attributes. In the reducer phase, the consolidated results from mappers are utilized by DNFN to detect DDoS attacks accurately.

7.1. DDoS Attack Detection using DNFN

The pivotal target of DDoS is to make the system nonexistent and cause interruption while delivering services to the users with traffic from various sources. The most significant problem that arises while identifying the DDoS attack is the need for quick recognition of malicious users aiming to deteriorate the network bandwidth of the target model. The database created by a server is massive, and processing such a huge dataset consumes more time. Hence, there is a necessity for a quick parallel processing model with the allocation of good storage for quick identification of attacks. Here, the selected feature $F$is subjected to DNFN, where the DDoS attack detection is performed in an efficient way by tuning the hyperparameters of DNFN using the proposed VCROA.

7.2. Architecture of DNFN

DNFN [13] is the hybrid approach of a Deep Neural Network (DNN) and fuzzy logic concept. The procedure begins with the DNN, proceeds with the DNFN, and uses fuzzy logic for objective computation. The classifier's essential parameters involve both the premises and the consequents. The consequent parameters are associated with the defuzzification process [21]. The description of both functions is stated below:

At each layer, the network nodes represent specific input or output factors. Consider that there are two premises p and q one consequent r that is expressed in the equation below,

$$J_{1,v} = \eta X_v(p) or J_{1,v} = \eta Y_{v-2}(q), \forall v = 1,2,,4 \quad (18)$$

In the aforementioned expression, $p$and $q$signifies inputs to a separate v^{th} entity, ηX and ηY_{v-2} represents the antecedent membership parameters. The value of membership is depicted as $J_{i,v}$. The factors are mapped utilizing the bell structure referred to as Gaussian membership factors that are represented with a high value of 1 and a low value of 0.

$$\eta X_v(p) = \frac{1}{1+\left|\frac{p-s_v}{t_v}\right|2o_c} \quad (19)$$

where the membership parameters of the premise are denoted as o_v, s_v and t_v, respectively, that are the learning parameters optimized by VCROA.

The second layer, used to encode the set of rules, is called the rule base layer. To achieve the necessary membership range, each node in this layer doubles the linguistic variables. The product of the membership degrees represents the activation strength of the rule, as mentioned below,

$$J_{2,v} = \mu_v = \eta X_v(p) \eta Y_{v-2}(q), \forall v = 1,2 \quad (20)$$

The third stage, normalization, calibrates the firing strength at every node of v^{th} the rule. μ_v shows the weight. The output of each rule is adjusted based on its firing strength, which is shown as:

$$J_{3,v} = \bar{\mu}_v = \frac{\mu_v}{\mu_1+\mu_2}, \forall v = 1,2 \quad (21)$$

During defuzzification, the consequents of individual rules are tuned to assess their combined effect on the output. This form is given as,

$$J_{4,v} = \bar{\mu}_v Z_v = \bar{\mu}_v(x_v p + y_v q + z_v), \forall v = 1,2 \quad (22)$$

where, x, y and z are the consequent factors set. The summation layer, which is the final layer, evaluates the cumulative result of the previous layer.

$$J_{5,v} = \sum_v \bar{\mu}_v Z_v = \frac{\sum_v \mu_v Z_v}{\sum_v \mu_v} \quad (23)$$

The output gained through the DNFN classifier is signified as H. Figure 3 demonstrates the foundational structure of the DNFN.

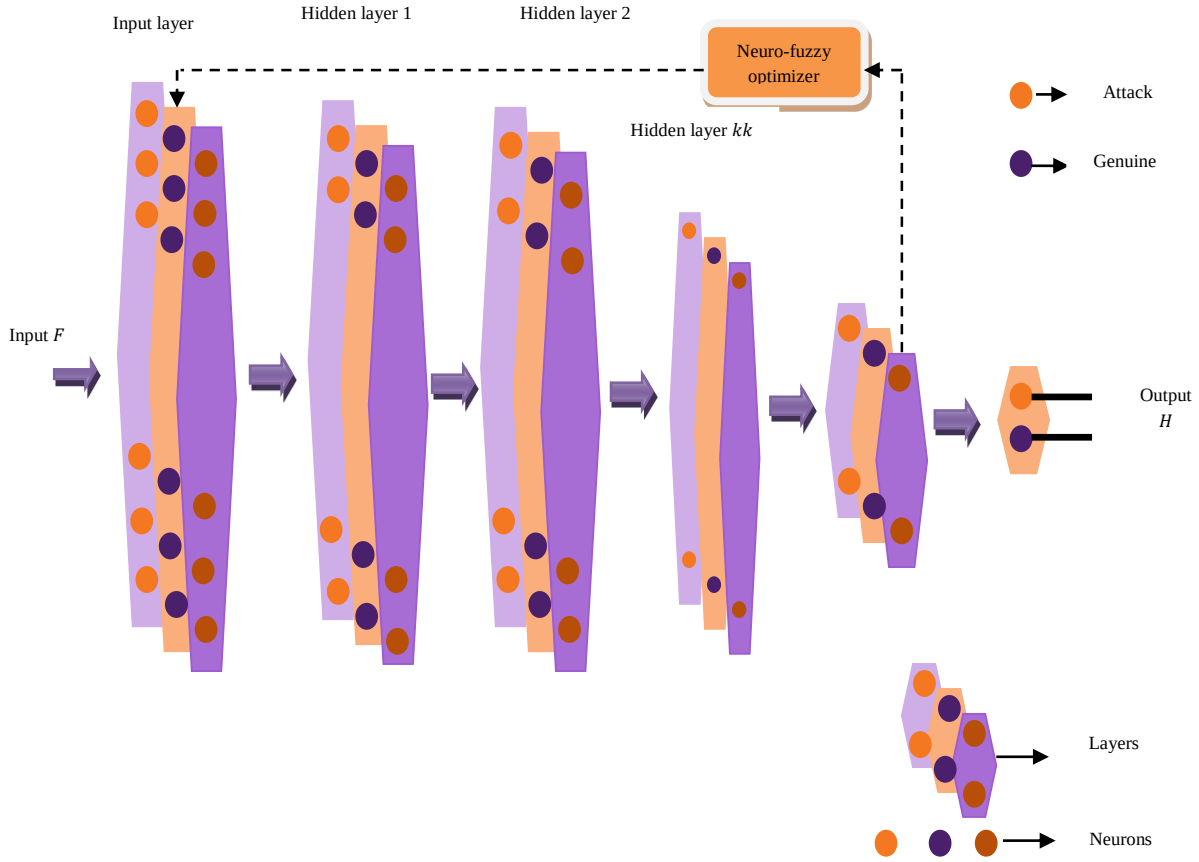


Fig. 3 Organizational layout of DNFN

8. Optimal Tuning of DNFN using the Proposed VCROA

The parameters o, s and $t$of DNFN are adjusted using the designed VCROA r to achieve accurate detection accuracy. However, VCROA is derived by the addition of the velocity contour-based idea into the ROA. The algorithmic procedures of VCROA are elaborated already under segment 3.3.1 a) in a neat manner.

8.1. Remora Position Encoding

Position encoding serves to map the solution vector in the context of a specific optimization task in a G space $G = [1 \times \ell]$. Here, $\ell \in o, s, t$.

8.2. Objective Function

To determine a good solution for the detection accuracy of DDoS attacks, an objective function is employed, and it is shown as the variation between the actual output and the result of the DNFN classifier.

$$\Im = \frac{1}{j}\sum_{i=1}^{j}[TA_{out} - H]^2 \quad (24)$$

Here, $j$implies the overall amount of data and Hindicates the output of DNFN. In addition, the targeted outcome is notated as TA_{out}.

8.3. Results of VCROA-DNFN

The outcomes of the devised VCROA-DNFN are illustrated in this segment. Moreover, a comparative table is also studied in this section.

8.4. Experimental Setup

Experiments with VCROA-DNFN were performed in Python on a Windows 10 PC with 4 GB RAM and an Intel Core i3 processor. The parameters for the experiments are presented in Table 2.

Table 2. Experimental parameters

Parameters	Values
Epoch	100
Batch size	32
Activation Function	Leaky-ReLU
Learning rate	0.001
Activation function for dense	ReLu
β_1	0.9
β_2	0.999
Kernel size	5x5
Number of fuzzy rules	50
Membership-Function (MF)	Triangular MF (a=0, b=1, c=2)
Number of Layers	5
Neurons per layer	[128, 256, 128, 64, 10]
Dropout rate	0.2
Regularization (L2)	0.001
Parameters of VCROA	
Population size	30
Maximum iterations	100
Search space bounds	[-10,10]
Dimensionality of decision variables	30
Constant for “host-feeding” small step (C)	0.1
Poisson-like randomness parameter (λ)	6

8.5. Dataset Description

The BoT-IoT dataset [9] (dataset 1) was developed by the UNSW Canberra Cyber Range Lab and represents a realistic network environment containing both normal and botnet-generated traffic. The University of New South Wales-Network Behavior (UNSW-NB15) dataset [29] (dataset 2) consists of 2,540,044 network traffic records that include normal and multiple attack categories. The original UNSW-NB15 dataset provides predefined training and testing files; however, for experimental consistency, the datasets are re-partitioned during evaluation. In this study, a two-way holdout validation strategy is employed. The dataset is divided into training and testing subsets by varying the proportion of training data from 50% to 90%, while the remaining data (50% to 10%) is used for testing. This variable split strategy enables performance assessment under different training data availability conditions and supports evaluation of the model’s generalization capability. In addition, K-fold cross-validation is employed to assess the robustness and stability of the proposed model, as reported in Section 4.6.2.

8.6. Evaluation Measures

The VCROA-DNFN performance is evaluated using three metrics: precision, recall, and F-measure.

8.6.1. Precision

It is the ratio of true positive DDoS attacks to all attacks identified in the network.

$$P = \frac{K_p}{K_p + L_p} \tag{25}$$

where, K_p and L_p denotes the true and false positives.

8.6.2. Recall

It is the ratio of actual DDoS attacks that were successfully identified, and it is given by,

$$\Re = \frac{K_p}{K_p + L_n} \tag{26}$$

Here,$L_n$refers to the false negative.

8.7. F-Measure

This term indicates test performance and is expressed as the mean of precision and recall.

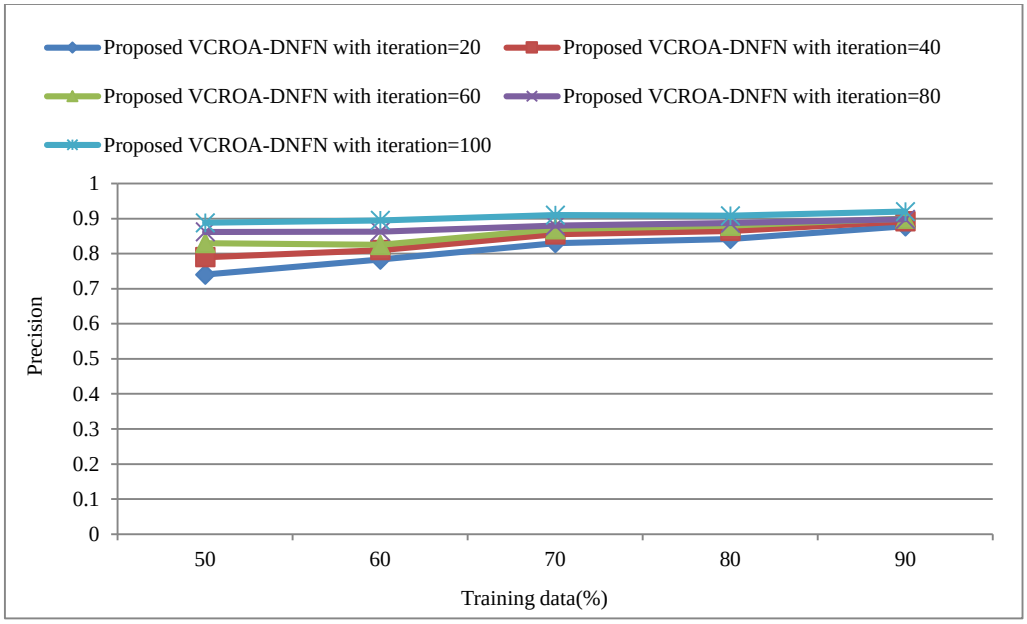
$$F - measure = 2 \cdot \frac{P * \Re}{P + \Re} \tag{27}$$

8.8. Performance Evaluation

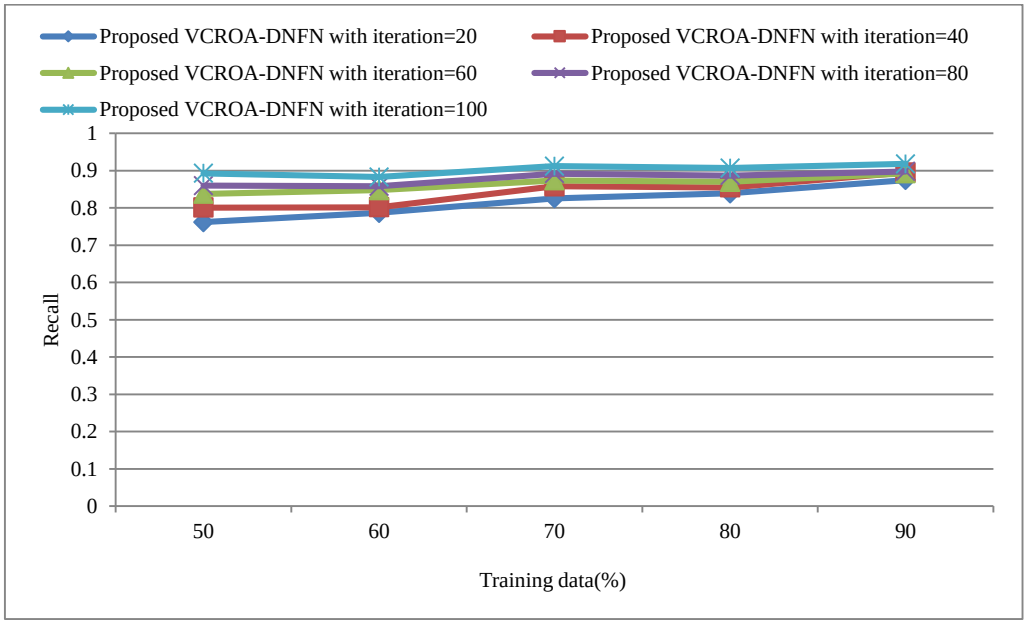
Figure 4 illustrates the performance estimation of VCROA-DNFN when the data ranged between 50% and 90% and the number of iterations ranged between 20 and 100. Figure 4(a) is the precision-based evaluation.

The precision of VCROA-DNFN at the 20th iteration is 0.879, at the 40th iteration is 0.891, at the 60th iteration is 0.901, at the 80th iteration is 0.897, and at the 100th iteration is 0.920 when 90 percent of the data is taken into account. Figure 4(b) shows the VCROA-DNFN performance of recall. At 90 training data, the recall values of 20, 40, 60, 80, and 100 iterations are 0.875, 0.897, 0.892, 0.897, and 0.918, respectively. The F-measure evaluation is presented in Figure 4(c). Based on 90 percent of the data, the F-measure of the developed method at iterations 20, 40, 60, 80, and 100 is 0.869, 0.895, 0.883, 0.890, and 0.918, respectively.

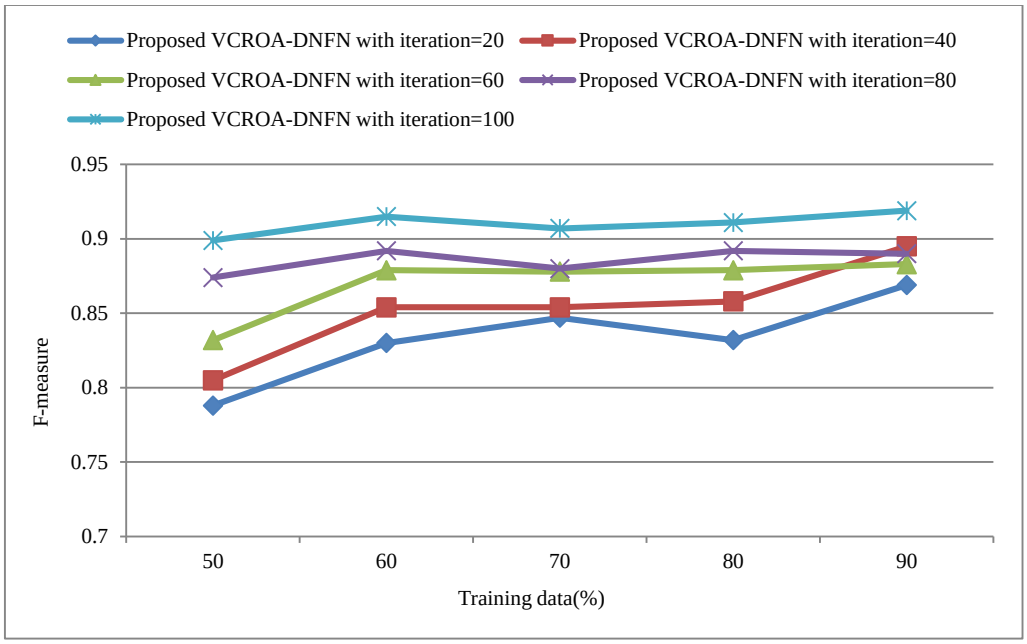
The results of the analysis show that the suggested VCROA-DNFN has high and stable performance on the various iterations, hence demonstrating its reliability and strength in the detection of DDoS attacks.



(a)



(b)



(c)

Fig. 4 Performance analysis with training data: (a) Precision, (b) Recall, and (c) F-measure.

9. Comparative Methods

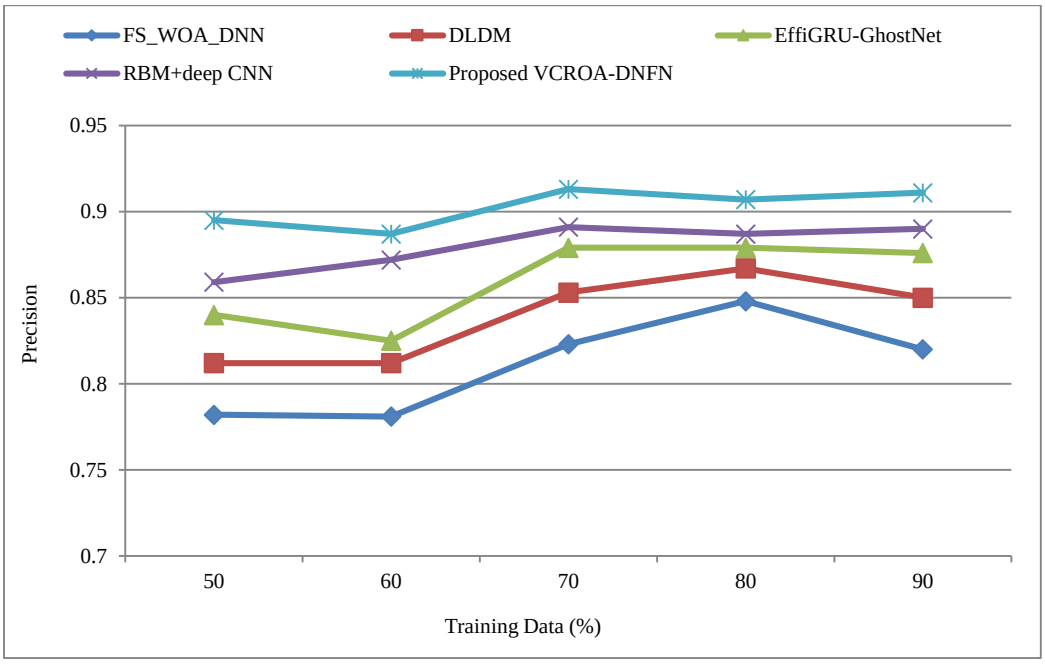
To measure the effectiveness of the proposed strategy, it is compared with current methods such as FS-WOA-DNN [1], DLDM [8], EffiGRU-GhostNet [30], and RBM+deep CNN [7].

9.1. Comparative Estimation

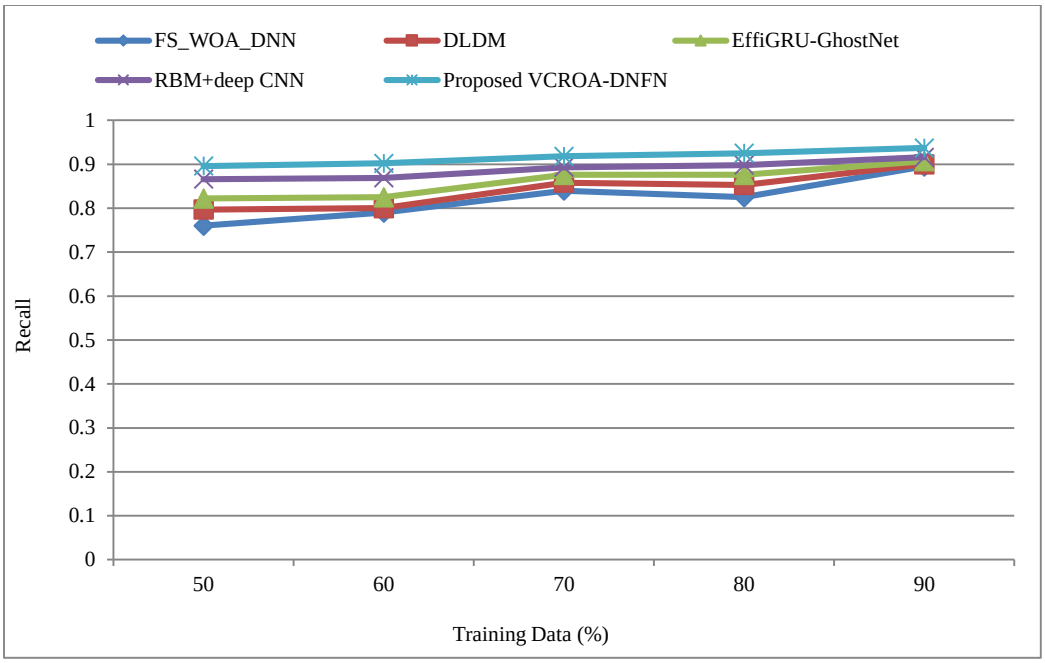
The following section is the comparative evaluation of VCROA-DNFN based on dataset 1, taking into account the differences in the training data proportions and k-fold cross-validation.

9.2. Estimation with Training Data

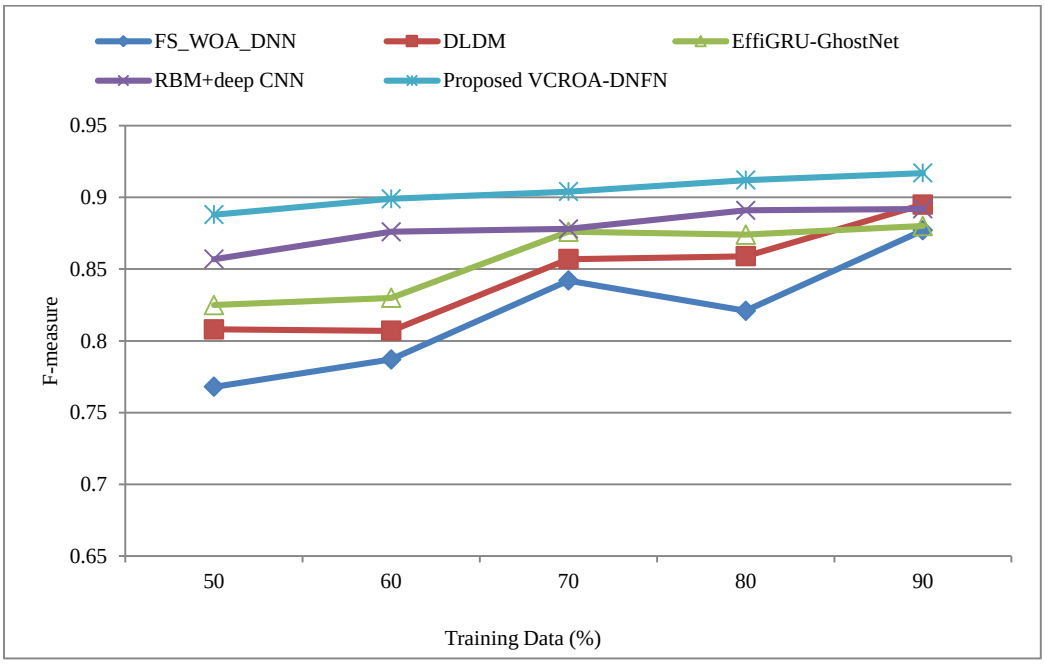
Figure 5 indicates the evaluation of the developed method in terms of evaluation measures of different proportions of training data. Figure 5(a) shows the precision evaluation with 90 percent of the training data, with the proposed VCROA-DNFN having a precision of 0.911.



(a)



(b)



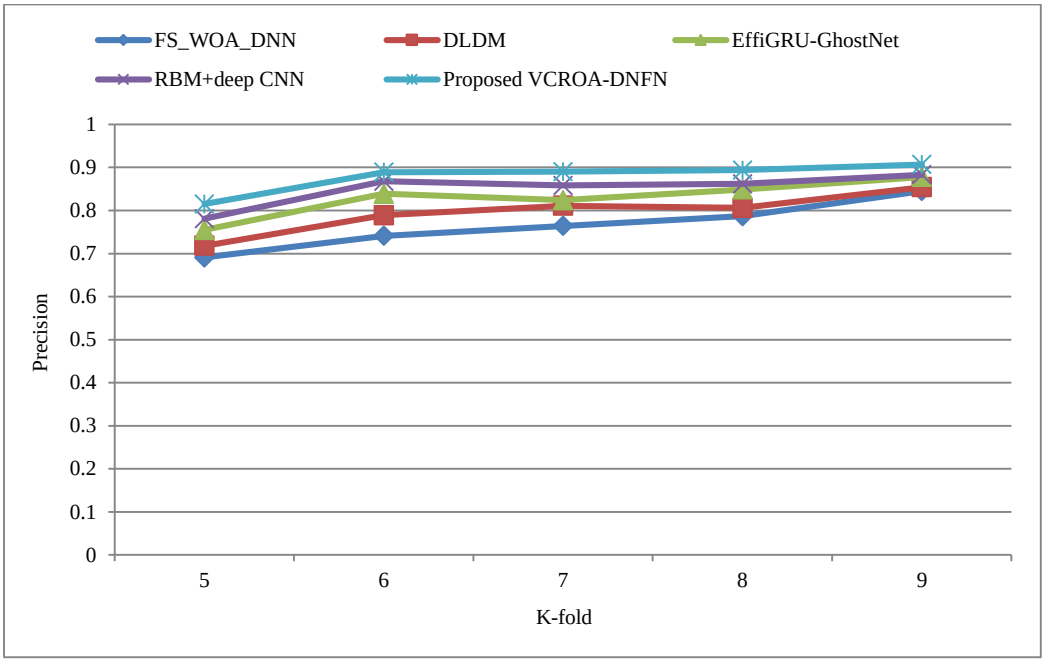
(c)

Fig. 5 Comparative evaluation with training data: (a) Precision, (b) Recall, and (c) F-measure.

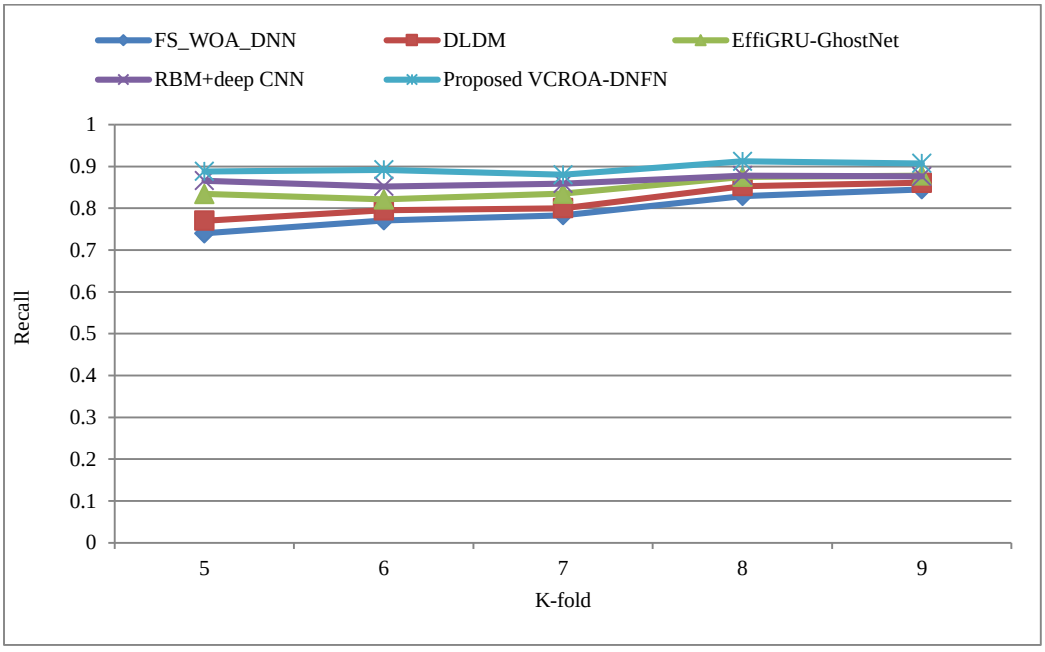
This is a better result compared to the current methods, that is, FS-WOA-DNN by 9.962%, DLDM by 6.663%, EffiGrU-GhostNet by 3.822%, and RBM+deep CNN by 2.313%. Figure 5(b) shows the recall performance of data = 90, where the recall values of FS-WOA-DNN, DLDM, EffiGRU-GhostNet, and RBM+deep CNN are 0.894, 0.900, 0.907, and 0.916, respectively. The relative enhancement of VCROA-DNFN compared to these approaches is 4.538%, 3.917%, 3.2055%, and 2.215%, respectively. The F-measure evaluation is shown in Figure 5(c). The F-measure of VCROA-DNFN at data = 90% is 0.917, and the F-measure of FS-WOA-DNN, DLDM, EffiGRU-GhostNet, and RBM+deep CNN are 0.877, 0.896, 0.881, and 0.892, respectively. Such findings suggest that VCROA-DNFN is always more precise, recalls better, and has a higher F-measure than the current approaches, which proves the effectiveness and reliability of this tool in detecting DDoS attacks.

9.3. Estimation with K-Fold Cross-Validation

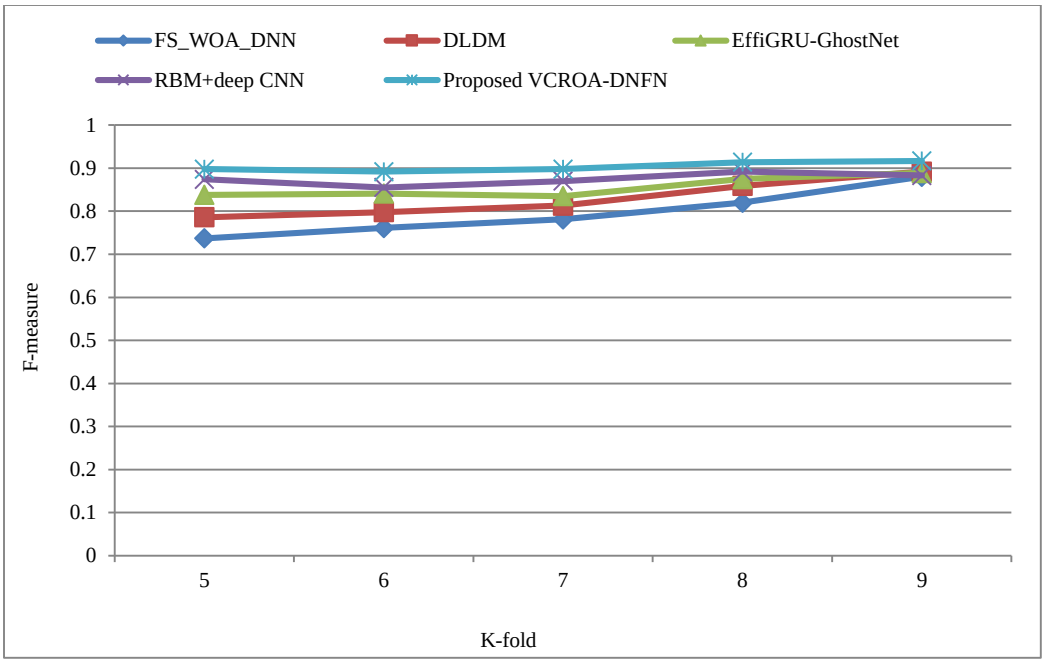
The performance of the proposed VCROA-DNFN model under cross-validation of K-fold is presented in Figure 6, with the value of K ranging between 5 and 9. Figure 6(a) shows the accuracy of VCROA-DNFN. At K=9, the suggested VCROA-DNFN attains a precision of about 0.91, which is significantly higher than FS-WOA-DNN, DLDM, EffiGRU-GhostNet, and RBM+deep CNN. Figure 6(b) demonstrates the performance of recall with various K-fold values. VCROA-DNFN has a recall of about 0.91 at K = 9, whereas FS-WOA-DNN, DLDM, EffiGRU-GhostNet, and RBM+deep CNN have lower recall values, which means that the proposed method has the highest detection ability. The F-measure comparison is given in Figure 6(c). It can be seen that the proposed VCROA-DNFN has the largest F-measure at all K-fold values, with the largest value of about 0.91 at K = 9. These findings validate the claim that VCROA-DNFN has remained consistent and better in terms of cross-validation using K-folds, which indicates its strength and ability to generalize in the detection of DDoS attacks.



(a)



(b)

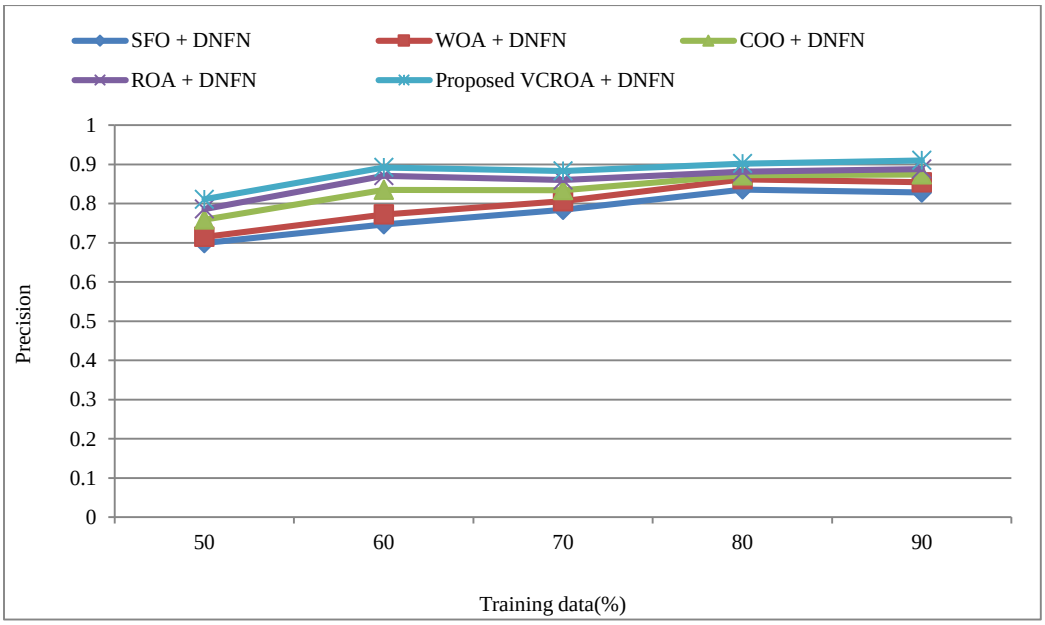


(c)

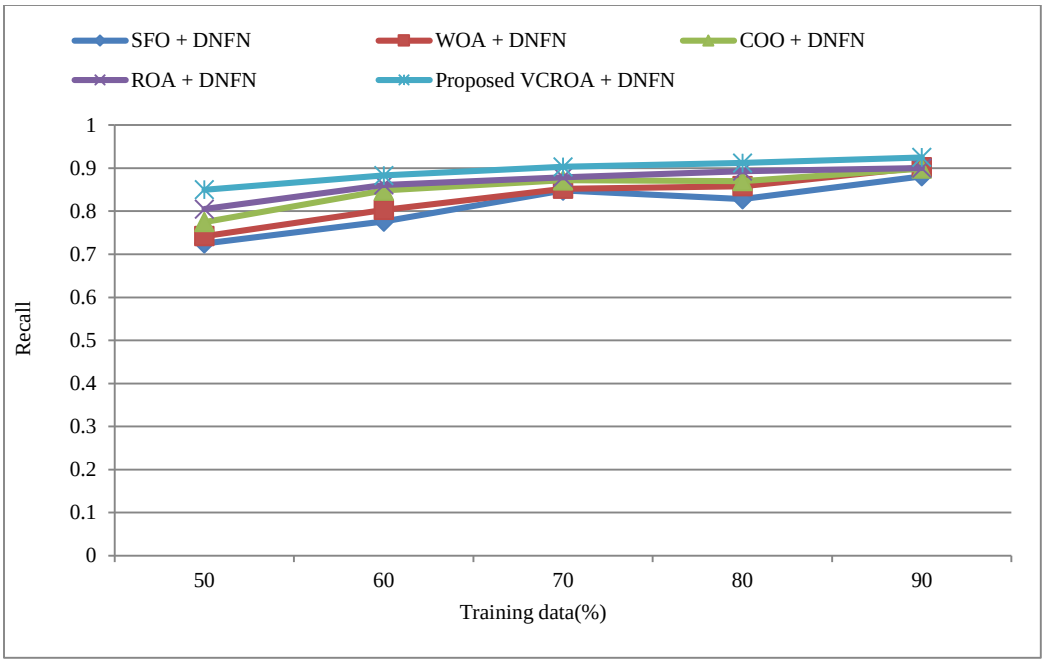
Fig. 6 Comparative analysis with K-fold cross-validation: (a) Precision, (b) Recall, and (c) F-measure.

9.4. Algorithmic Analysis

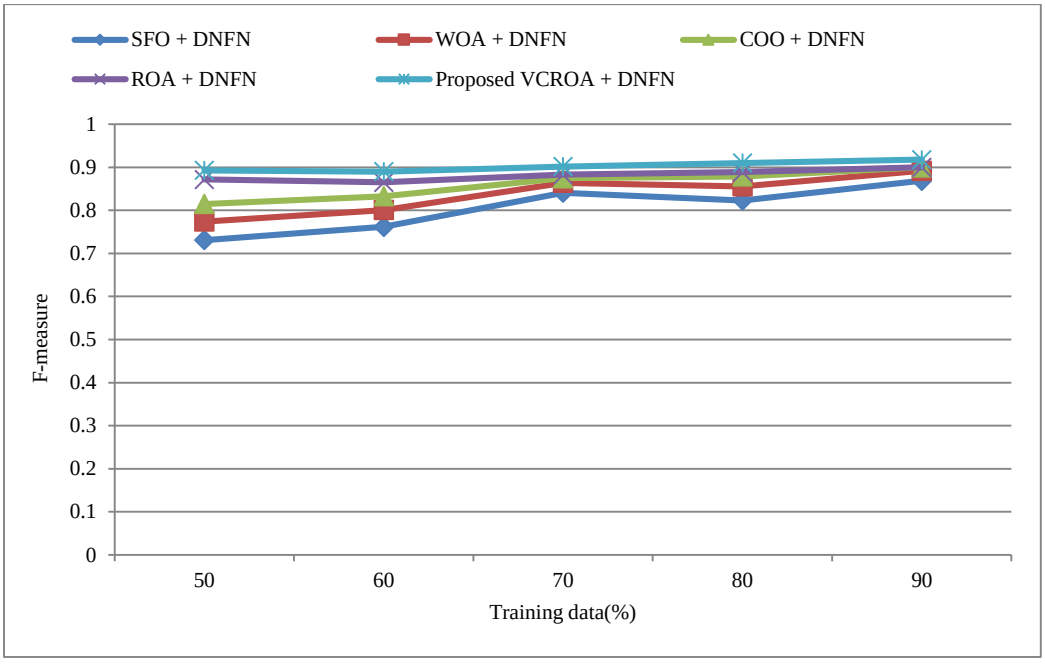
Figure 7 depicts the algorithmic evaluation of VCROA-DNFN with respect to the evaluation measures by changing the training data, and the techniques considered for comparison are Sail Fish Optimizer (SFO) [24] + DNFN, WOA [25] + DNFN, COOT [26] + DNFN, and ROA [10] + DNFN. Figure 7(a) presents the precision assessment of VCROA-DNFN. With the training data increased to 90%, the model attains a precision of 0.910, which shows an increment to that of conventional strategies, named SFO + DNFN, is 8.944%, WOA + DNFN is 6.172%, COOT + DNFN is 3.810%, and ROA + DNFN is 2.155%. Figure 7(b) reports a recall value of 0.926 achieved by VCROA-DNFN. The F-measure performance of VCROA-DNFN is illustrated in Figure 7(c). For the 90% data scenario, the F-measure attained by VCROA-DNFN is 0.920. The evaluation confirms that VCROA-DNFN provides significant improvements over existing optimization techniques. This highlights its strong predictive capability in accurate DDoS attack detection.



(a)



(b)



(c)

Fig. 7 Algorithmic estimation with training data: (a) Precision, (b) Recall, and (c) F-measure.

9.5. Analysis Based on Cross-Dataset

Figure 8 shows the accuracy of VCROA-DNFN in training data between 50% and 90% in a cross-dataset evaluation environment, where dataset 1 is the training set and dataset 2 is the testing set.

The precision of FS-WOA-DNN, DLDM, EffiGRU-GhostNet, and RBM+deep CNN is 0.888, 0.891, 0.896, and 0.904 when 80% of the data are used in training, whereas the VCROA-DNFN proposed has a precision of about 0.886. As the volume of training data increases to 90%, the accuracy of VCROA-DNFN also increases to approximately 0.91, which is higher than all the baseline techniques. These findings show that the proposed VCROA-DNFN model has a better cross-dataset performance and improved generalization ability when tested on heterogeneous network traffic data.

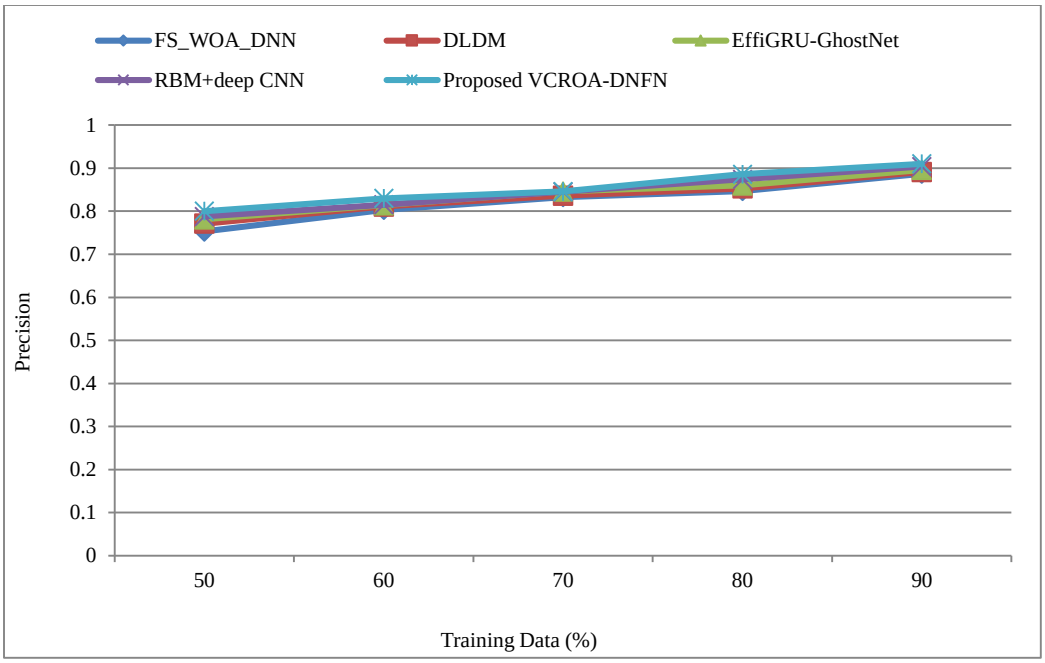


Fig. 8 Cross-dataset evaluation of VCROA-DNFN

9.6. Analysis Based on Feature Selection Methods

Figure 9 shows the analysis of the proposed VCROA using current feature selection algorithms, such as Recursive Feature Elimination (RFE) [32], LASSO Regression [33], and Exhaustive Feature Selection [34].

The paper measures accuracy performance on dataset 1, where the training data are scaled between 50 percent and 90 percent. The precision achieved by RFE at 90% training data is 0.889, LASSO Regression is 0.89, Exhaustive Feature Selection is 0.9, and VCROA is 0.908.

The analysis of the results shows that the suggested VCROA algorithm is more effective than traditional algorithms such as RFE, LASSO Regression, and Exhaustive Feature Selection because it is the most precise at 0.908 when 90 percent of the data is used for training. This shows that VCROA is effective in choosing the best features in detecting DDoS attacks.

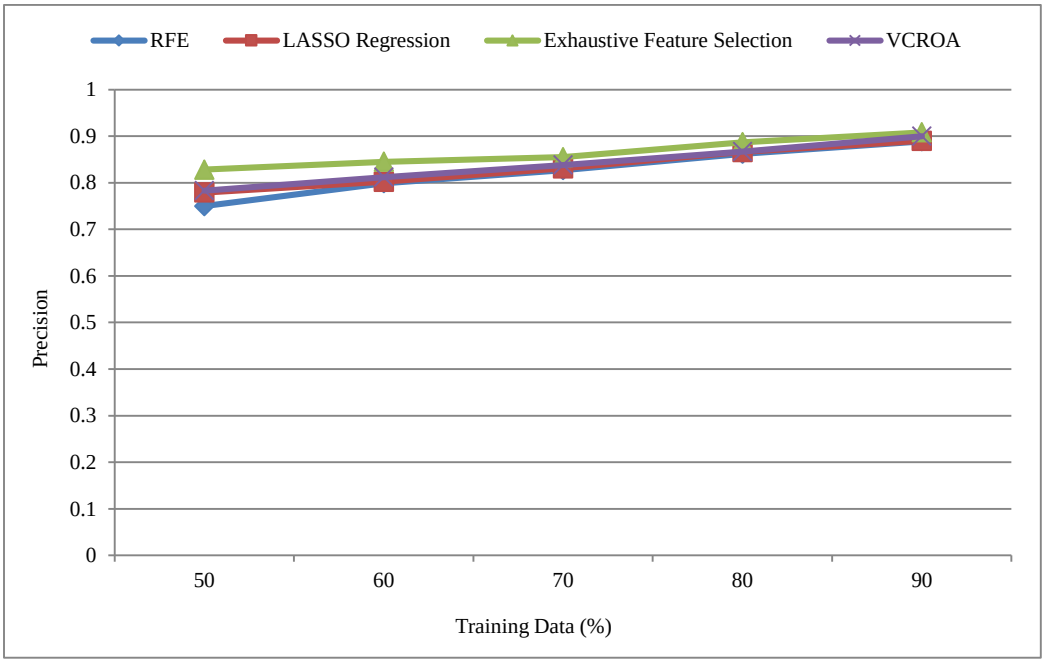


Fig. 9 Assessment of feature selection methods

9.7. Assessment based on Computational Time

Table 3 shows the evaluation of the computational time of the suggested VCROA-DNFN. VCROA-DNFN is compared with existing methods, including FS-WOA-DNN, DLDM, EffiGRU-GhostNet, and RBM+deep CNN, in terms of computational time. The results of the assessment prove that the suggested VCROA-DNFN technique takes the shortest time of 5.32 seconds of computation, thus confirming its efficiency in detecting DDoS attacks. All procedures were conducted and run under the same hardware and software conditions so as to have a fair comparison of computational time.

Table 3. Computational time analysis

Methods	Computational time (sec)
FS-WOA-DNN	9.10
DLDM	8.68
EffiGRU-GhostNet	7.65
RBM+deep CNN	6.99
Proposed VCROA-DNFN	5.32

9.8. Evaluation based on the T-Test

The results of the t-test analysis are given in Table 4. Statistical evaluation is done on the precision, recall, and F-measure values obtained through 10-fold cross-validation to determine the significance of the differences in performance across methods. Statistical analysis is performed on the fold-wise performance values achieved under the same experimental conditions of all the methods compared; a p-value of less than 0.05 is taken as statistically significant improvement.

The findings show that the proposed VCROA-DNFN is always better than FS-WOA-DNN, DLDM, EffiGRU-GhostNet, and RBM+deep CNN in all the metrics considered. In particular, VCROA-DNFN is more precise than FS-WOA-DNN, $t = 3.00$, $p = 0.02$, DLDM, $t = 2.97$, $p = 0.02$, EffiGRU-GhostNet, $t = 2.72$, $p = 0.03$, and RBM+deep CNN, $t = 2.72$, $p = 0.03$. The same statistically significant changes apply to recall and F-measure, which suggests the strength and efficiency of the suggested VCROA-DNFN in detecting DDoS attacks.

Table 4. T-test analysis

Proposed Method	Baseline Methods	T-Statics			P-value		
		Precision	Recall	F-measure	Precision	Recall	F-measure
VCROA-DNFN	FS-WOA-DNN	3.00	2.93	2.97	0.02	0.02	0.02
	DLDM	2.97	2.92	2.96	0.02	0.02	0.02
	EffiGRU-GhostNet	2.72	2.91	2.85	0.03	0.02	0.02
	RBM+deep CNN	2.72	2.79	2.81	0.03	0.02	0.02

9.9. Analysis based on Confidence Interval (CI)

The results of the confidence interval analysis of the proposed VCROA-DNFN are presented in Table 5. The evaluation is done by the measures of precision, recall, and F-measure. The suggested VCROA-DNFN model achieved a precision range of [89.5-91.9], a recall range of [89.1-94.0], and an F-measure range of [88.6-91.8] in this assessment. This means that the VCROA-DNFN model is consistent and reliable in the performance of the model in various independent experimental runs, thus proving the strength of the proposed VCROA-DNFN model in DDoS attack detection.

Table 5. Confidence interval assessment

Methods/Metrics	Precision	Recall	F-measure
FS-WOA-DNN	[77.1-83.2]	[75.4-89.8]	[75.8-88.4]
DLDM	[80.5-86.1]	[79.0-90.5]	[79.4-90.0]
EffiGRU-GhostNet	[83.8-88.2]	[81.5-91.0]	[81.5-89.4]
RBM+deep CNN	[84.7-89.8]	[85.4-92.1]	[84.7-90.1]
Proposed VCROA-DNFN	[89.5-91.9]	[89.1-94.0]	[88.6-91.8]

9.10. Comparative Discussion

Table 6 demonstrates the discussion of VCROA-DNFN with the traditional approaches. It is mentioned that the VCROA-DNFN has led to better performance with a precision of 91.10, a recall of 93.70, and an F-measure of 91.70. Based on the precision results, the performance enhancement achieved by the proposed VCROA-DNFN model is 9.99 percent better than the FS-WOA-DNN, 6.59 percent better than the DLDM, 3.84 percent better than the EffiGRU-GhostNet, and 2.31 percent better than the RBM+deep CNN. Similarly, for the recall, the performance improvement attained by the VCROA-DNFN is 4.49%, 3.95%, 3.2% and 2.24% over the FS-WOA-DNN, DLDM, EffiGRU-GhostNet, and RBM+deep CNN models. The improvement in

performance regarding F-measure is 4.36% over the FS-WOA-DNN, 2.29% over the DLDM, 3.93% over the EffiGRU-GhostNet, and 2.73% over the RBM+deep CNN. The proposed VCROA-DNFN combines the strengths of Velocity Contour-based Remora Optimization to enhance feature selection and parameter tuning of DNFN. This improvement allows the model to quickly converge, reduce computation, and accurately detect DDoS attacks. As a result, the VCROA-DNFN model achieves higher precision, recall, and F-measure compared to traditional methods.

Table 6. Comparative discussion

	Metrics/Methods	FS-WOA-DNN	DLDM	EffiGRU-GhostNet	RBM+deep CNN	VCROA-DNFN
		Dataset 1				
Training data=90%	Precision	82.00%	85.10%	87.60%	89.00%	91.10%
	Recall	89.40%	90.00%	90.70%	91.60%	93.70%
	F-measure	87.70%	89.60%	88.10%	89.20%	91.70%
K-fold value=9	Precision	84.50%	85.40%	87.70%	88.30%	90.80%
	Recall	84.50%	86.10%	87.70%	87.60%	90.70%
	F-measure	88.00%	89.20%	88.90%	88.30%	91.60%
		Cross-dataset				
Training data=90%	Precision	88.77%	89.09%	89.61%	90.36%	91.04%

9.11. Practical Implications

The real-world benefits and applications of the proposed VCROA-DNFN can be provided as follows:

The CROA-DNFN model is able to identify DDoS attacks in realistic network settings with high accuracy, thus assisting administrators in avoiding service disruption and system integrity. Large-scale network traffic can be handled efficiently by the model, making it applicable for enterprise-level deployment. The suggested solution can be applied to other cybersecurity solutions like intrusion detection, malware detection, and anomaly detection in the IoT networks.

The proposed VCROA-DNFN is realistic and representative by the following measures:

The VCROA-DNFN is trained and tested based on benchmark network traffic datasets containing both normal and malicious DDoS activity. The system is deployed with the help of the MapReduce framework, which enables the processing of significant amounts of network traffic in a distributed fashion. The features of the model are optimized with the help of VCROA, which helps to capture the various traffic properties, thus making it manage the variations in packet flows, traffic volumes, and attack types.

10. Conclusion

Cyber-attacks of all types have become very serious threats to both wired and wireless networks in recent years, and Distributed Denial of Service (DDoS) attacks stand out as some of the most disruptive because they can disrupt essential network services. Even though this is an essential network defense, early identification and separation of suspicious traffic is vital. To overcome this issue, this paper suggested a VCROA-DNFN-based DDoS attack detection system deployed in a MapReduce system. The suggested method combines the mean-based missing data imputation, VCROA-based feature selection at the mapper stage, and Deep Neuro-Fuzzy Network (DNFN)-based classification at the reducer stage. The presence of the concept of velocity contour in the Remora Optimization Algorithm makes it possible to select features efficiently and tune parameters of DNFN adaptively. The experimental assessments of benchmark data sets indicate that the suggested VCROA-DNFN model performs better than the current ones, with the precision, recall, and F-measure rates being 91.10, 93.70, and 91.70, respectively. The findings show that the model is efficient in dealing with noisy traffic, generalizing to unseen attack patterns, and processing large-scale network data in an efficient manner. Due to these features, the suggested framework can be deployed to cybersecurity applications, smart cities infrastructure, and other large-scale networked systems that need efficient DDoS detection. Although these are good results, the present study is restricted to offline testing with benchmark datasets. The future work will consider expanding the VCROA-DNFN framework to real-time, high-speed network settings and consider further improvements to add to scalability and robustness.

Code Availability

The source code of the proposed VCROA-DNFN is available at:
"https://github.com/Rahulkwadekar/VCROA-DNFN.git".

Acknowledgement

I would like to express my 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.

References

[1] Ankit Agarwal, Manju Khari, and Rajiv Singh, "Detection of DDOS Attack using Deep Learning Model in Cloud Storage Application," *Wireless Personal Communications*, vol. 127, no. 1, pp. 419-439, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Rana Abu Bakar et al., "FTG-Net-E: A Hierarchical Ensemble Graph Neural Network for DDoS Attack Detection," *Computer Networks*, vol. 250, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Matheus P. Novaes et al., "Adversarial Deep Learning Approach Detection and Defense Against DDoS Attacks in SDN Environments," *Future Generation Computer Systems*, vol. 125, pp. 156-167, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Abdullah Emir Cil, Kazim Yildiz, and Ali Buldu, "Detection of DDoS Attacks with Feed Forward based Deep Neural Network Model," *Expert Systems with Applications*, vol. 169, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Haizhen Wang, Xiaojing Yang, and Na Jia, "DDoS Attack Detection Method based on Improved Convolutional Long Short-Term Memory and Three-Way Decision in SDN," *PLOS One*, vol. 20, no. 5, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] R. Doriguzzi-Corin et al., "LUCID: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876-889, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Asmaa A. Elsaeidy, Abbas Jamalipour, and Kumudu S. Munasinghe, "A Hybrid Deep Learning Approach for Replay and DDoS Attack Detection in a Smart City," *IEEE Access*, vol. 9, pp. 154864-154875, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] M. Premkumar, and T.V.P. Sundararajan, "DLDM: Deep Learning-based Defense Mechanism for Denial of Service Attacks in Wireless Sensor Networks," *Microprocessors and Microsystems*, vol. 79, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Nour Moustafa, "*BoT-IoT dataset*," IEEE Daataport, 2026. [CrossRef] [Publisher Link]

[10] Heming Jia, Xiaoxu Peng, and Chunbo Lang, "Remora Optimization Algorithm," *Expert Systems with Applications*, vol. 185, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Md. Zahid Hasan, K.M. Zubair Hasan, and Abdus Sattar, "Burst Header Packet Flood Detection in Optical Burst Switching Network using Deep Learning Model," *Procedia Computer Science*, vol. 143, pp. 970-977, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Prabhakar Krishnan, Subhasri Duttagupta, and Krishnashree Achuthan, "VARMAN: Multi-Plane Security Framework for Software Defined Networks," *Computer Communications*, vol. 148, pp. 215-239, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Sakeena Javaid et al., "Towards Buildings Energy Management: Using Seasonal Schedules under Time of Use Pricing Tariff via Deep Neuro-Fuzzy Optimizer," *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Tangier, Morocco, pp. 1594-1599, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mrutyunjaya Panda, and Manas Ranjan Patra, "Network Intrusion Detection using Naive Bayes," *International Journal of Computer Science and Network Security*, vol. 7, no. 12, pp. 258-263, 2007. [Google Scholar] [Publisher Link]

[15] P. Vijaya Mehna Sangtani et al., "Implementation Challenges involved in Big Data Analytics," *International Journal of Engineering Sciences and Research Technology*, vol. 5, no. 10, pp. 834-840, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mohan V. Pawar, and J. Anuradha, "Network Security and Types of Attacks in Network," *Procedia Computer Science*, vol. 48, pp. 503-506, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ahmad Javaid et al., "A Deep Learning Approach for Network Intrusion Detection System," *Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS)*, vol. 3, no. 9, pp. 21-26, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[18] Kaiyuan Jiang et al., "Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network," *IEEE Access*, vol. 8, pp. 32464-32476, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] T. Aditya Sai Srinivas, and S.S. Manivannan, "Prevention of Hello Flood Attack in IoT using Combination of Deep Learning with Improved Rider Optimization Algorithm," *Computer Communications*, vol. 163, pp. 162-175, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Rojalina Priyadarshini, and Rabindra Kumar Barik, "A Deep Learning based Intelligent Framework to Mitigate DDOS Attack in Fog Environment," *Journal of King Saud University Computer and Information Sciences*, vol. 34, no. 3, pp. 825-831, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Omolbanin Yazdanbakhsh, and Scott Dick, "A Deep Neuro-Fuzzy Network for Image Classification," *arXiv preprint*, pp. 1-10, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Junnan Yao et al., "Pre-Processing of Incomplete Spectrum Sensing Data in Spectrum Sensing Data Falsification Attacks Detection: A Missing Data Imputation Approach," *IET Communications*, vol. 10, no. 11, pp. 1340-1347, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[23] Vishal Maheshwari, Ashutosh Bhatia, and Kuldeep Kumar, "Faster Detection and Prediction of DDoS Attacks using MapReduce and Time Series Analysis," *2018 International Conference on Information Networking (ICOIN)*, Chiang Mai, Thailand, pp. 556-561, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] S. Shadravan, Hamid Reza Naji, and V.K. Bardsiri, "The Sailfish Optimizer: A Novel Nature-Inspired Metaheuristic Algorithm for Solving Constrained Engineering Optimization Problems," *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 20-34, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[25] Seyedali Mirjalili, and Andrew Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[26] Iraj Naruei, and Farshid Keynia, "A New Optimization Method based on COOT Bird Natural Life Model," *Expert Systems with Applications*, vol. 183, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Boddupally Janaiah, "Attack Detection in IoT using DBN based Optimization Algorithm," *Journal of Networking and Communication Systems*, vol. 5, no. 1, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] A.V. Krishna Prasad, "Deep Learning based Optimization for Detection of Attacks in IoT," *Journal of Networking and Communication Systems*, vol. 4, no. 2, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[29] The UNSW-NB15 Dataset, 2025. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[30] Abdulrahman A. Alshdadi et al., "Big Data-Driven Deep Learning Ensembler for DDoS Attack Detection," *Future Internet*, vol. 16, no. 12, pp. 1-26, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[31] Ahmed Jamal Ibrahim, Sándor R. Répás, and Nurullah Bektaş, "Feature-Optimized Machine Learning Approaches for Enhanced DDoS Attack Detection and Mitigation," *Computers*, vol. 14, no.11, pp. 1-33, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[32] Christopher A. Ramezan, "Transferability of Recursive Feature Elimination (RFE)-Derived Feature Sets for Support Vector Machine Land Cover Classification," *Remote Sensing*, vol. 14, no. 24, pp. 1-25, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[33] Yimeng Guo et al., "Development and Validation of a Survival Prediction Model for Patients with Advanced Non-Small Cell Lung Cancer based on LASSO Regression," *Frontiers in Immunology*, vol. 15, pp. 1-17, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[34] Chintalpudi S.L. Prasanna, Md Zia Ur Rahman, and Masreshaw D. Bayleyegn, "Brain Epileptic Seizure Detection using Joint CNN and Exhaustive Feature Selection with RNN-BLSTM Classifier," *IEEE Access*, vol. 1, pp. 97990 - 98004, 2023. [CrossRef] [Google Scholar] [Publisher Link]