Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

Anomaly-Based IDS for Cloud Environments Using Black Widow Feature Selection and Maclaurin–Deep Belief Classifier

Vivek Parganiha*1, Soorya Prakash Shukla2, Lokesh Kumar Sharma3, Monika Verma4

1*Associate Professor at Department of Computer Science & Engineering,
Bhilai Institute of Technology, Durg, Chhattisgarh 491001, India

2Professor at Department of Electrical Engineering, Bhilai Institute of Technology,
Durg, Chhattisgarh 491001, India

3Scientist-E at ICMR-New Delhi, India

4PhD Scholar at Department of Computer Science & Engineering,
Bhilai Institute of Technology, Durg, Chhattisgarh 491001, India

***Email:** vivekparganiha@gmail.com

ARTICLE INFO

Received: 01 Nov 2023

Revised: 17 Dec 2023

Accepted: 26 Dec 2023

ABSTRACT

Cloud computing is a large domain making efficient use of resources. The key concerns in cloud computing are performance and security. The selection of ideal features and high false alarm rates, required nevertheless for the highest levels of accuracy, are also major challenges. To resolve these problems and improve accuracy, an efficient cloud intrusion detection system (IDS) using the black widow optimization (BWO) algorithm and Maclaurin series–deep belief network (Maclaurin-DBN) is introduced in this paper. BWO is availed to select the ideal features from a feature set. BWO is a new optimization methodology; early convergence and achieving an optimized fitness value are its main advantages. Then a Maclaurin-DBN is deployed to classify the selected feasible features. The BWO's multiple iterations create the best total neurons, but the hidden layers face different types of attacks. The IDS based on Maclaurin-DBN obtains the maximum identification rate with a compact structure. The introduced deep learning–based IDS is simulated on the Python platform, and the performance of the introduced model is computed and compared with previous deep learning algorithms and existing works. The UNSW_NB15, NSL_KDD, and CICIDS-2017 are the three datasets utilized for the implementation. The simulation outcomes show that the proposed scheme achieved a maximum classification accuracy and detection rate of 95.38% and 93.34%, 99.41% and 98.83%, and 99.42% and 97.12%, respectively, on the NSL-KDD, UNSW-NB15, and CICIDS-2017 datasets.

Keywords: IDS, cloud computing, Black Widow, Maclaurin-DBN, virtual machine.

1. Introduction

Cloud computing is designed to enable on-demand, suitable, universal network access to a common configurable computing property that can be rapidly released and provisioned with minimal service provider engagement or administrative effort [1] [2]. There are two ends mainly used in cloud computing: a front end and a back end. Communication with the cloud and the service is offered by the front end. The software and hardware resources process the obtainable services and the user's queries. A virtual machine monitor (VMM), or hypervisor, is supported by a host system that processes multiple virtual machines (VMs) [3]. To uncover a VM's vulnerabilities, various techniques are adopted in attacks. Exploiting a VM's vulnerabilities compromises it [4]. A VM's normal operation is affected when it is under an attacker's control because malware is injected into the system. Malware identification software are not widely available to keep a VM in a good state all the time [5]. Thus, the IDPS is a prime requirement to avoid many attacks types and threats and protect the whole VM.

In cloud computing, which provides a wide variety of utility-oriented services from different service providers [6] [7], one of the chief concerns today is security in recent applications of VMMs, as there are many susceptibilities. Multiple vulnerabilities are deliberated, such as a hypervisor, VM, and virtual network at the virtualization layer of cloud computing [8]. Using both external and internal attackers, the exact virtualization assaults can be performed. External attackers execute the attacks frequently, such as a virtual code injection, footprinting, and virtualized botnets over VMs or hypervisors [9]. The internal attackers could be the cloud providers themselves, cloud users, or clients at the provider side. They can carry out attacks such as VM to VM, side channel, communication among VMs, VM escape, and VM observation from the host framework. The cloud system identifies internal attackers as more challenging than external attackers [10]. A firewall is an effective solution for network security tasks [11]. The system has become a demanding constituent in terms of network security [12].

For the cloud computing surroundings, the cloud security alliance suggested an automated security control: that is, an intrusion detection system (IDS) [13]. The IDS comprises numerous components used for the detection and identification of malicious actions. To detect intrusions, a user over host frameworks observes system logs, calls, file-system changes, and other host accomplishments; this is known as host-based IDS [14]. The network traffic is monitored by the network IDS (NIDS), and the content of packets is utilized to discover malicious traffic [15]. Various threats can affect the virtualization in the cloud environment, like a DDoS attack, because of important data stored in the cloud [16]. The IDS detects attacks and threats in a virtualized server or cloud; it is a strong mechanism and plays a vital role in securing networks [17]. Presently, cloud computing is attracting researchers, and it is undergoing continual development; now many software organizations and educational institutions are adopting the cloud due to the benefits of cloud resources and to utilize distributed computing. However, the security issues in the cloud constitute a key challenge [18].

The cloud computing environment is threatened with various types of cyberattacks. Nowadays, hackers are employing tricky techniques to steal and exploit users' credentials without their knowledge [19]; malicious users can inject malicious code into any website and gain unauthorized access to the website users' confidential information stored in the cloud [20]. NIDS is modelled to detect malicious activities like DDoS, worm, and virus attacks. Machine learning (ML) approaches can be used to improve it for low false alarm rates (FARs) and better accuracy [20]. Deep learning (DL) approaches are an advanced form of ML adopted in NIDS. The software-defined framework is a new network model to present NIDS using ML approaches [21]. Detecting many attacks and reducing the FAR is the research focus for IDS [22]. Many critical applications are used to deliver a small degree of service even when under attack – thus, the need for the attack-tolerant systems [23].

In the past few years, a very significant problem has been cloud security, and it is the focus of many development and research efforts, especially because, to deploy a large-scale DDoS, attackers can compromise VMs and explore a cloud system's vulnerabilities. The service availability and data threats are increased because of the several challenges in cloud computation. To enhance cloud security, numerous security services are required for the providers as well as the users. Location independence and rapid information processing are offered by cloud computing. But one of the foremost problems here is the lack of trust between cloud users because of location-independent processing. Thus, for effective deployment of cloud services, cloud security becomes very important. The IDS is the most effective attack detection and prevention (IDPS) system between cloud resources and users. To cover the possible vulnerabilities, an effective IDS is the one that has a maximum detection efficiency. According to the alert mechanism, deployment location, and detection method, a new model can be designed.

The foremost contributions of this research task are described below:

- For building the IDS, ML approaches are widely adopted. In this research work, an anomaly-based network IDS is introduced for the cloud environment. To develop an effective system for detecting an intrusion in cloud computing, this paper presents an efficient feature selection and classification algorithm (BWO-DBN).
- To select optimal features, an optimization algorithm named BWO is adopted as a search approach.
- Finally, a Maclaurin-DBN is introduced to classify the selected feasible features. The proposed Maclaurin-DBN algorithm determines the intrusion present within the cloud network using a DBN-based network. The DBN's biases and weights are trained by the Maclaurin series for gaining the intrusion.
- The possible network attacks of the incoming network traffic flow are detected by the network IDS, which distresses the VMs or multiple hosts in the cloud environment.

The remaining portion of the paper is organized as follows: Section 2 contains a brief survey of IDS; Section 3 contains an explanation of the proposed IDS model; Section 4 contains an assessment of the performance of the proposed methodology; and Section 5 contains the overall conclusion of the research work and recommendations for future enhancement.

2. Related Works

The following are some of the most recent research articles on cloud intrusion detection models:

VMI-based security design was discussed by Mishra *et al.* [24] for intrusion detection in the surroundings of the cloud. A VMI-based flexible and effectual security strategy was proposed in this paper. Known attacks and their variants were detected by fine granular observation of VMs. The network called VM Guard analysed the traces of a series running on the TVM, based on the introspection feature at the VMM-layer. For the residents and the CSPs, the security framework was advantageous, and it had low false-positive rates. But the detection power of the IDS needs to improve, which is a flaw of this research.

A countermeasure selection and system intrusion detection was introduced by Chung *et al.* [25] in virtual network systems. NICE was suggested in this research to investigate and mitigate combined assaults in a cloud-based virtual system. The feasibility of NICE and the reduced risk of cloud systems were demonstrated by the system's performance, but they were also abused and exploited by external and internal attackers. To significantly improve the detection of an attack and mitigate the consequences of the attack, this research introduced open flow network programming APIs. The scalability of this research work was absent, and the detection accuracy of the IDS was not correctly determined, just investigated. These issues were the main problems of this research.

A machine learning approach was offered by Bhat *et al.* [26] for an IDS on VMs of the cloud. For a VM on cloud computing, an anomaly-based IDS using a machine learning approach was proposed. The NB tree's original implementation was utilized to get better performance, and the JAVA platform was utilized for the implementation through NSL-KDD of KDD'99 datasets. ML approaches were utilized for the anomaly-based IDS and produced better outcomes related to accuracy and false positives, yet the computation speed was low compared to other models.

A well-organized prevention system and DDoS TCP flood attack detection were discussed by Sahi *et al.* [27] in cloud surroundings. This paper proposed a novel classifier approach for perceiving and avoiding DDoS TCP flood assaults (CS DDoS) on public clouds. Whether a packet was standard or created from an aggressor was identified and determined by CS_DDOS at the time of detection. Malicious packets were blocked from accessing cloud services, and the IP source was blacklisted as part of the prevention. In public clouds, the DDoS TCP flood attacks (CS_DDoS) were effectively predicted and solved by a new classifier framework. The merit of this work was that stored documents were secured by classifying the packets. It failed to solve the DDoS issue and identify the intruders, which were the demerits of this research.

Sakr *et al.* [28] presented an anomaly-based network IDS, which analysed and monitored the traffic flow of the network aiming at the cloud atmosphere. The classifier of the network connections utilized here was support vector machine (SVM). While the standard-based particle swarm optimization (SPSO) was employed to tune the SVM control parameters, the binary PSO was employed for choosing the most relevant network features. To evaluate and build the proposed system, the benchmark NSL-KDD dataset was utilized as the source of network data. Low FARs and high detection accuracy were achieved by this scheme, as demonstrated by its results.

Kumar *et al.* [29] presented an integrated rule-based IDS. The integrated classification-based model was designed by the UNSW-NB15 dataset to identify malicious actions in the network. The real-time dataset of the proposed scheme was generated from the NIT Patna CSE lab (RTNITP18). The proposed IDS's performance was evaluated by this RTNITP18 and UNSW-NB15 (benchmark) datasets. The different types of threats in the network were detected by the proposed IDS design.

Moustafa *et al.* [30] presented a collaborative anomaly detection system (CADF) for big data handling. The methodology of installation and implementation were illustrated by the deployment and technical functions of the system. Its believability was assessed using the UNSW-NB15 dataset while it was being installed in cloud computing environments. In terms of detection rate (DR) and false-positive rate, the performance of the proposed scheme was evaluated.

Sarumi *et al.* [31] proposed a discovering computer networks intrusion. The comparison between two IDSs was showed in this paper. Apriori was one association rule, and the other was a machine learning scheme named SVM. The UNSW-NB15 and NSL-KDD datasets were compared to the two systems' performance.

Krishna *et al.* [32] presented a system for NIDS through fast kNN. For cloud location on CICIDS2017 dataset, an FkNN classifier was presented as an IDS. According to computational time, recall, accuracy, and precision, the FkNN was compared with PDS-kNN and kNN. More than 88% of gain in computational time was achieved by the proposed scheme. The FkNN approach was the best ML scheme for NIDS to reduce economic losses and mitigate attacks of CSPs within a lesser span of time.

Singh and Ranga [33] proposed a multilayer perceptron and genetic algorithm (MLP-GA) based IDS. The connection weights were predicted by the MLP-GA. The CICIDS 2017 dataset was utilized to simulate and test the offered design. The proposed model ability was demonstrated by the implementation results with the highest DR and minimal false alarm warnings.

For a real-time IDS, the approach of SwiftIDS was presented by Jin *et al.* [34]. The SwiftIDS's novel IDS method is capable of both analysing large amounts of traffic information in high-speed systems and retaining reasonable detection performance. Using the two methods, these objectives were accomplished in SwiftIDS. First, LightGBM (light gradient boosting machine) was used to manage the massive amount of traffic information. This served as the ID scheme for the system. This approach also employs its benefit to abridge data preprocessing for categorical features. Second, a parallel ID approach was used to investigate the traffic information inwards during various time periods. Delays could be eliminated, which was provided by the later-arriving information for the end of the ID cycle of the first-arriving data. Via the offline tests, the effectiveness of time and suitable detection performance was proved for the SwiftIDS using three datasets.

Table 1: Literature of existing related works

Author and Reference	**Purpose**	**Methodology**	**Demerits**	**Datasets**
Mishra *et al.* [24]	To detect hidden malware through basic memory introspection	Bag of *n*-grams (BonG) with TF-IDF & random forest (RF)	Detection power of introduced IDS	University of New Mexico (UNM)
Chung *et al.* [25]	To mitigate and detect collaborative attacks in the cloud	NICE: Network intrusion detection and countermeasure	Low detection accuracy	NVD, OSVDB, and CVE
Bhat *et al.* [26]	Intrusion detection of cloud virtual machines	NB tree and RF	Low operation speed and changing ratio of attacks	NSL_KDD
Sahi *et al.* [27]	DDoS TCP flood attack detection and prevention	CS_DDoS (Least squares support vector machine)	Want to extend CS_DDoS to solve the DDoS issue	NSL_KDD
Sakr *et al.* [28]	Analyse and monitor network traffic flow and improve detection accuracy	SVM, BPSO, and SPSO	Optimizing the control parameters, security, and privacy to guard cloud resources and assets	NSL_KDD
Kumar *et al.* [29]	To detect an attack in a network	Integrated rule-based IDS	Low DR and ability of classification	UNSW_NB15
Moustafa *et al.* [30]	To handle big data	CADF	Real cloud computing environment	UNSW_NB15
Sarumi *et al.* [31]	Comparison between two IDSs	Apriori-SVM	Real-life application	UNSW_NB15
Krishna *et al.* [32]	To overcome lazy learning kNN classifier nature	FkNN	Real-time environment	CICIDS-2017

Singh and Ranga [33]	Anomaly detection	MLP and GA	High false positives	CICIDS-2017
Jin *et al.* [34]	Minimizing false alarm rates and enhancing detection rate	LightGBM	Adaptive length of time windows	CICIDS-2017

Rapid data processing and location independence are offered by cloud computing. Trust is considered as one of the key problems among cloud users because of the independent-position processing for using its resources. Thus, for effective deployment of cloud services, cloud security becomes vital. Also, outer-side security provision is not as smart as an in-built security feature. So, to address these issues, the best feature selection and classification schemes are introduced in this research work.

3. Proposed Methodology

Presenting an anomaly-based IDS for cloud environs is the main goal of this research. The network IDS is attached with switches and installed at the cloud network's entry point for cloud servers or VMs networks. Initially, a packet sniffer captures the network traffic (raw data). Then, before the classification procedure, the network data flow's features are obtained and finely preprocessed.

3.1. Problem Statement

The rapid development and popularization of cloud networks have brought many problems to cloud network security. Generally, the foremost challenges in IDPs are high FARs, high-speed system limitations do not deal with cloud necessities, difficulty recognizing core intrusion attacks, and not utilizing appropriate parameters or standards. Besides, a high FAR is the main problem for anomaly-based IDS, and the only limitation is the large FPR of identification. It will be reduced by the proper classification strategy. Moreover, another problem is selecting optimal features to achieve maximum testing accuracy. The deployment of an effective IDS can be summarized as below:

- For the detection of malicious movement, other strategies had a large training time.
- After studying the entire system, the alert creation is placed, and calls are traced for outcomes into a late or slow reaction over interference.

To overcome these issues and enhance the classification accuracy, a BWO optimized Maclaurin-DBN scheme is introduced in this cloud IDS system.

3.2. System Architecture of a Proposed Cloud IDS

Internet security is critical in the modern global context due to the growth of e-learning, e-government, and e-business. Due to scattered data in different storage devices and machines, including PCs, servers, and different mobile devices such as smartphones and wireless sensor networks, data security becomes serious in the cloud computing atmosphere. The proposed system architecture uses five modules: data collection, feature selection, classification, alert module, and decision module. The dataset is collected and stored in the data collection module. Next, the data are sent to the IDS for preprocessing. In preprocessing, feature selection and classification take place. In feature selection, the BWO is used to eliminate unwanted and redundant features and choose the optimal features from the dataset. Then the selected features are given to the classification module, which uses a new ML approach (i.e., Maclaurin-DBN) to classify the normal and attack types. The classified features are given to the decision module, and it takes the decision and alerts the VM via the alert module. Figure 1 demonstrates the system architecture for the introduced model.

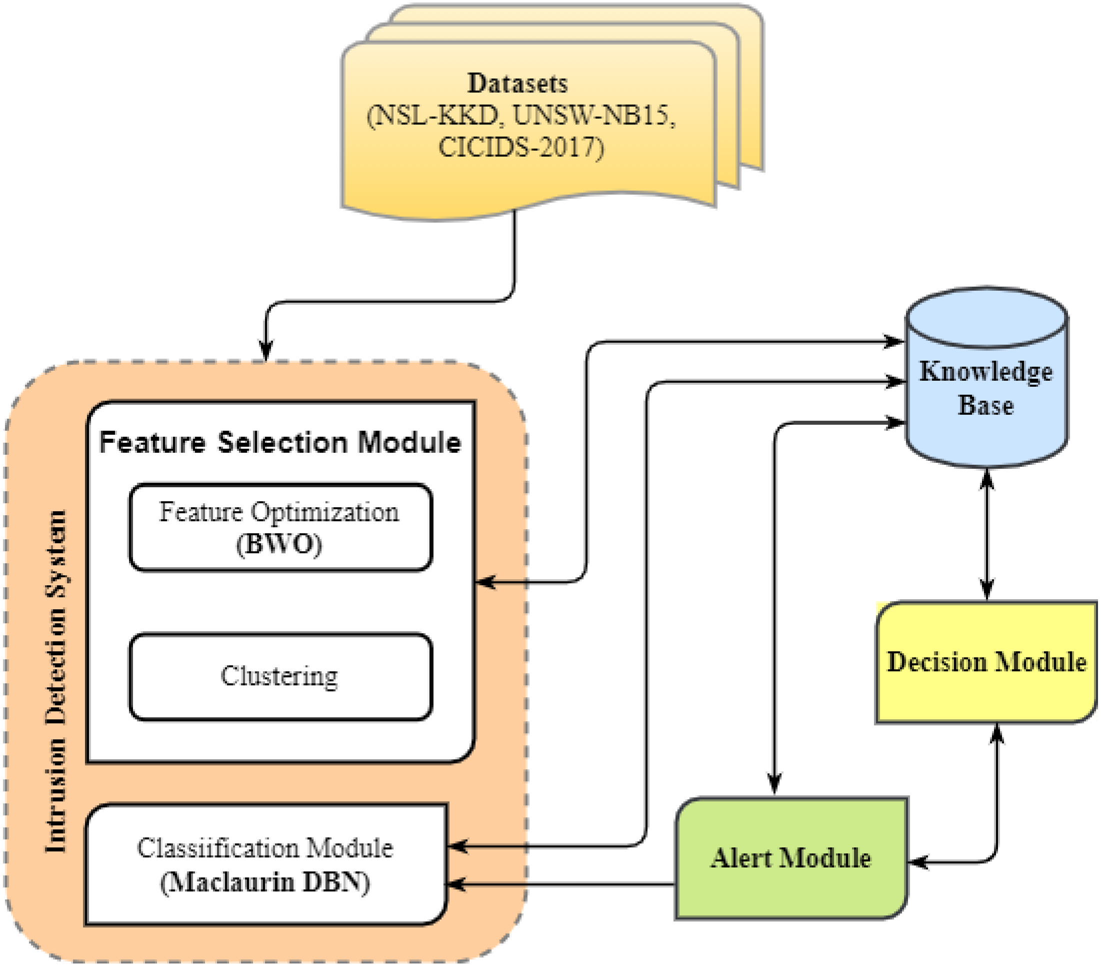

Datasets
(NSL-KKD, UNSW-NB15, CICIDS-2017)
Intrusion Detection System
Feature Selection Module
Feature Optimization (BWO)
Clustering
Classiification Module (Maclaurin DBN)
Knowledge Base
Decision Module
Alert Module

Figure 1: System architecture for proposed model

- ***Dataset***

NSL-KDD, UNSW-NB15, and CICIDS-2017 datasets are utilized here and given as input to the IDS for research.

- ***Feature Selection Module***

In this module, with the help of BWO, the best features are selected. It consists of two subsystems: feature optimization and clustering. Initially, the BWO selects the optimum features, and a new dataset is formed for further processing. At last, the reduced dataset and optimized features are given to the next module.

- ***Classification Module***

For the classification of the IDS, Maclaurin-DBN is utilized in this module. Additionally, this module is utilized to refer to the knowledge base to create an actual choice on instances based on the output of the selected data.

- ***Alert Module***

It produces the awareness of intrusion that is decided by feature matching. This module contains information about intrusive connections, such as the IP address of the victim VM, the IP address of the attacker, the detection result, the destination port, and the source port.

- ***Decision Manager***

By analysing the results of the classification, a decision is reached as to whether the authority is a normal user or malicious. Overall, constituents' control of the proposed agent is given to this module.

- ***Knowledge Base***

The various types of information, like the necessary rules for picking the features, all the types of features, and creating an efficient choice on the dataset are held by the knowledge base. To create an efficient choice based on feature-designated cases, the classification rules are kept in the knowledge base. The probable, efficient rules are included in the knowledge base, which is greatly employed to detect or identify the attack. For effective decision-making, it offers satisfactory data to the classification.

3.3. Cloud IDS Using BWO and Maclaurin-DBN

Because of the rapid rise of internet technologies and the use of networks in several fields such as e-business, e-learning, social networking, and so on, network infiltration has disproportionately increased, making data security from malevolent hackers increasingly difficult to maintain. The IDS is responsible for monitoring and defending the network from intruders. It can be divided into three types: host based, application based, and network based.

3.3.1 Feature Selection Module

The BWO [35] is utilized to preprocess the data, which mimics the mating behaviour of black widow spiders. The designed BWO method is adopted to tackle several feature selection responsibilities. It reveals a high efficiency in tackling several optimization cases. Feature optimization and clustering are the two subsystems that are used in this module.

Feature Optimization Using BWO: The exclusive mating behaviour of black widow spiders is described in the BWO procedure. This algorithm can be a deliberate grouping of the swarm and evolutionary approaches. The BWO adopts the new special stage named 'cannibalism'. The main benefit of this stage is that classes with unsuitable fitness are eliminated from the loop; thus, the speed of convergence is improved. This algorithm offers better performance in complex issues, provides quick convergence, and eliminates local optima. It also balances the exploitation and exploration stages. For this reason, this approach can be a better algorithm for resolving the different kinds of optimization issues with numerous local optima.

Initialization: The population includes the number of widows with size, which is represented as N, where $1 \times N_{\text{var}}$ represents the array of each widow producing the solution of issues. The array is represented as follows:

$$W = x_1, x_2, \ldots, x_{N_{\text{var}}} \tag{1}$$

where N_{var} represents the optimization problem's dimension, and x_i represents the i^{th} candidate solution.

The widow's fitness is determined by the fitness function evaluation f of every widow of the set $\left(x_1, x_2, \ldots, x_{N_{\text{var}}}\right)$.

$$fitness = f(W) \tag{2}$$

$$fitness = f\left(x_1, x_2, \ldots, x_{N_{\text{var}}}\right) \tag{3}$$

In the proposed methodology, replace f by the fitness function.

$$fitness = \rho_1 \gamma_r(D) + \rho_2 \frac{|C - r|}{|C|} \tag{4}$$

where r represents the length of the designated subset of feature; C represents the entire feature set; $\gamma_r(D)$ refers to the condition attribute set r virtual to decision D's classification accuracy, ρ_1 and ρ_2 are both parameters symmetric to the length of the subset. By randomly initializing a population, the optimization procedure is initiated in a $N_{pop} \times N_{\text{var}}$ matrix size. Then, to activate the procreating stage, parents pairs are randomly selected, here, during, or after the female widow has eaten the male widow.

Procreate: If a widow array used random numbers, then an array named α is created in the procreation step. Then the parameter α is produced as the offspring.

$$\lambda_1 = \alpha \times x_1 + (1 - \alpha) \times x_2 \tag{5}$$

$$\lambda_2 = \alpha \times x_2 + (1 - \alpha) \times x_1 \tag{6}$$

where λ_1 and λ_2 are represented as the offspring.

Cannibalism: This process is divided into sibling cannibalism, sexual cannibalism, and cannibalism that is frequently preserved. Here, the mother spider is eaten by the baby spiders. After performing this process, the new population is generated, and it is kept in pop2.

Mutation: This procedure is accomplished using the arbitrary choice of the Mutepop amount of entities from the inhabitants to be transformed. The two components of the array are swapped arbitrarily by all the selected solutions. Then the new population is created, and it is kept in pop3. At last, by the combination of pop3 and pop2, the new population is determined, and it is arranged to yield the finest widow of threshold values with N_{var} measurement.

Clustering: The ideal features that have been acquired are clustered or categorized according to the normal and attack types. The classification process is carried out using the grouped features.

3.3.2 Classification Module Using Maclaurin-DBN

In this module, the Maclaurin-DBN is used to classify the data, where DBN [36] is a generative design of deep neural networks. These classification results are given to the result for the purpose of alert. This module discusses the knowledge to create a perfect decision on instances. Figure 2 displays the DBN network architecture and the bias and weights used in all the layers. With the suitable bias and weights existing in the hidden and visible layer, the DBN network is trained here. In this work, a linear

Maclaurin series is utilized, so the residual part contains the nonlinear terms. The neural networks are excellently matched to estimate nonlinear functions, so the network is trained by this residual part. To get the fault or residual, the Maclaurin series' forecasted value is detracted from the actual value. Hence, this residual is used to train the DBN.

The proposed Maclaurin-DBN classifier has dissimilar parameters, such as two biases and weights inside the architecture for training. Depending on the feature vector, the intrusion detection process efficiency is influenced by the choice of suitable weight and bias. Therefore, to select the optimal biases and weight value, the feature vector is necessary for the Maclaurin-DBN scheme. Using equivalent input features, the Maclaurin-DBN selects the best value for the bias and the weight for all values in the training phase.

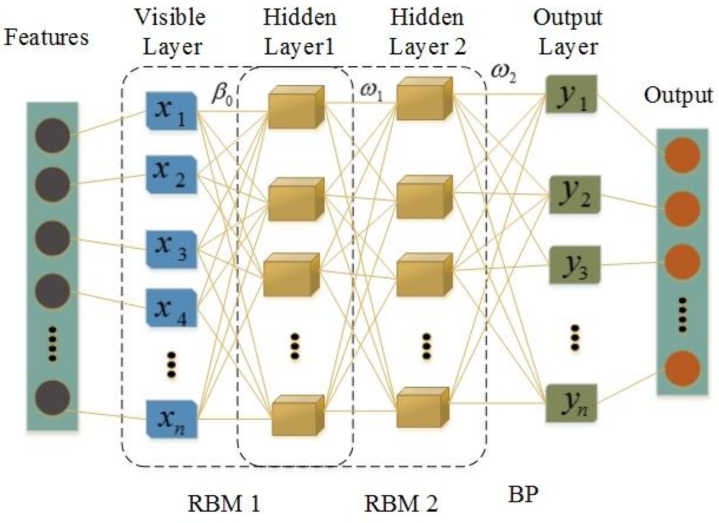


Figure 2: Schematic structure of the DBN model

The Maclaurin-DBN has some parameters such as β and ω, denoting the biases in the visible and hidden layers, with the weights represented as wgt. The obtained features from the BWO are given as the input to the introduced Maclaurin-DBN network as the feature vector. At both layers, the Maclaurin-DBN needs the bias and weight. The common expression for the DBN's function of energy relating to the visible and hidden parameters is expressed as follows:

$$E_{DBN}\left(\vartheta,\eta,\mathrm{I}^{i}\right)=-\sum_{u=1}^{v}\sum_{a=1}^{M}wgt_{ua}\vartheta_{u}\eta_{a}-\sum_{u=1}^{v}\beta_{u}\eta_{u}-\sum_{a=1}^{M}\omega_{a}\eta_{a} \tag{7}$$

where η_a and ϑ_u represent the a^{th} neurons in the hidden layer and the u^{th} neuron presented in the visible layer, individually; β_u represents the visible layer's bias; and ω_a denotes the hidden layer's bias.

3.3.2.1 Training Maclaurin-DBN: For the training purpose, the features extracted by the BWO are given to the Maclaurin-DBN network as the input, and the size of the feature vector is ρ.

Step 1: *Random biases and weights initialization:* For the training of the features, the appropriate biases and the weights are determined by the new Maclaurin-DBN scheme. Initially, the input features and weights are multiplied in the visible layer and then, based on the adjustment given by the bias, are passed on to the next layer. In both layers of restricted Boltzmann machines (RBM), the bias is presented. The Maclaurin-DBN randomly picks the values for each bias β , ω , and weight wgt existing in the hidden and visible layers, respectively, to determine the appropriate biases and weights.

Step 2: *Weights updation:* In this step, with the Maclaurin prediction series [37], the weights convoluted in the network are updated. That determines the weights through eight coefficients for the next iteration. Using a gradient descent scheme, the weights update is expressed, which is the training approach utilized in the DBN by Equation 8:

$$wgt^{\mu+1} = wgt^{\mu} + wgt_{inc}^{\mu} \tag{8}$$

where wgt_{inc}^{μ} represents the incremental weight. The weight value for the following iteration is based on the eight Maclaurin constants according to the Maclaurin series prediction model.

Equation 9 shows the Maclaurin series expansion:

$$\begin{gathered} wgt^{\mu+1} = 0.5wgt^{\mu} + 1.359wgt^{\mu-1} - 1.35wgt^{\mu-2} + 0.619wgt^{\mu-3} - 0.2259wgt^{\mu-4} + \\ 0.055wgt^{\mu-5} - 0.0104wgt^{\mu-6} + 0.0013wgt^{\mu-7} - 9.9e^{-5}wgt^{\mu-8} \end{gathered} \tag{9}$$

where wgt^{μ} represents the weight created at the time of the present iteration denoted as μ ; $wgt^{\mu-1}$ denotes the weight iteration at $\mu-1$; and so on. wgt^{μ} is obtained by rearranging Equation 9, represented as follows:

$$wgt^{\mu} = 2\begin{pmatrix} wgt^{\mu+1} - 1.359wgt^{\mu-1} + 1.35wgt^{\mu-2} - 0.679wgt^{\mu-3} + 0.2259wgt^{\mu-4} - \\ 0.055wgt^{\mu-5} + 0.0104wgt^{\mu-6} - 0.0013wgt^{\mu-7} + 9.9e^{-5}wgt^{\mu-8} \end{pmatrix} \tag{10}$$

To obtain the desired weight update, substitute Equation 10 in Equation 8 for the Maclaurin-DBN, and this is represented as follows:

$$wgt^{\mu+1} = 2\begin{pmatrix} wgt^{\mu+1} - 1.359wgt^{\mu-1} + 1.35wgt^{\mu-2} - 0.679wgt^{\mu-3} + 0.2259wgt^{\mu-4} - \\ 0.055wgt^{\mu-5} + 0.0104wgt^{\mu-6} - 0.0013wgt^{\mu-7} + 9.9e^{-5}wgt^{\mu-8} \end{pmatrix} + wgt_{inc}^{\mu} \tag{11}$$

The weight updation based on the Maclaurin-DBN is given as follows:

$$\begin{gathered} wgt^{\mu+1} = 2.71wgt^{\mu-1} - 2.71wgt^{\mu-2} + 1.359wgt^{\mu-3} - 0.451wgt^{\mu-4} + \\ 0.11wgt^{\mu-5} - 0.0208wgt^{\mu-6} + 2.7e^{-3}wgt^{\mu-7} - 19.8e^{-5}wgt^{\mu-8} - wgt_{inc}^{\mu} \end{gathered} \tag{12}$$

Step 3: *Bias β updation:* The DBN's visible layer's bias is represented as β . Based on Equation 13, the visible layer's bias gets updated.

$$\beta^{\mu+1} = \beta^{\mu} + \beta_{inc}^{\mu} \tag{13}$$

where β_{inc}^{t} represents the visible layer's incremental bias. The bias function with respect to the Maclaurin series is represented as follows:

$$\beta^{\mu+1} = 2.71\beta^{\mu-1} - 2.71\beta^{\mu-2} + 1.359\beta^{\mu-3} - 0.451\beta^{\mu-4} + 0.11\beta^{\mu-5} - 0.0208\beta^{\mu-6} + 2.7e^{-3}\beta^{\mu-7} - 19.8e^{-5}\beta^{\mu-8} - \beta^{\mu}_{inc} \tag{14}$$

Step 4: *Bias* ω *updation:* Here, the hidden layer's bias, expressed as ω, is updated by the Maclaurin series. The bias value for the $(\mu+1)$ iteration is indicated by Equation 15, and using the parameter of incremental weight, it is represented as ω^{μ}_{inc}:

$$\omega^{\mu+1} = \omega^{\mu} + \omega^{\mu}_{inc} \tag{15}$$

The hidden layer bias function with respect to the Maclaurin series is represented as follows:

$$\omega^{\mu+1} = 2.71\omega^{\mu-1} - 2.71\omega^{\mu-2} + 1.359\omega^{\mu-3} - 0.451\omega^{\mu-4} + 0.11\omega^{\mu-5} - 0.0208\omega^{\mu-6} + 2.7e^{-3}\omega^{\mu-7} - 19.8e^{-5}\omega^{\mu-8} - \omega^{\mu}_{inc} \tag{16}$$

Step 5: *Stopping criteria:* The Maclaurin-DBN model's iteration is performed for the supreme iteration, represented as η. When the iteration μ touches η, the best biases and weights are achieved, and the best biases and weights are evaluated for feature input.

3.3.2.2 Testing Maclaurin-DBN: The procedure of testing the introduced Maclaurin-DBN is described in this section. The intrusion within the test features presents the testing procedure. By providing the test signal τ, the Maclaurin-DBN testing is done. The proposed Maclaurin-DBN detects the intrusion depending on the suitable bias and weight from the training process. τ is the test signal given to the introduced Maclaurin-DBN as follows:

$$\tau = \left[\tau_x; 1 \le x \le I\right] \tag{17}$$

Then the test signal's features are given to the trained Maclaurin-DBN network, which includes the proper biases and weights. Then it detects the intrusion existing in the test signal and classifies it under the types normal, probe, u2r, and r2l. The offered Maclaurin-DBN algorithm's classified test output for the test input is specified as follows:

$$\chi^{\tau} = Maclaurin\_DBN(\tau) \tag{18}$$

where χ^{τ} indicates the Maclaurin-DBN output and the esteem χ belongs to the set I, which means $\chi \in I$.

The proposed strategy for an efficient cloud IDS is explained by the introduced approaches named BWO for feature selection and Maclaurin-DBN for classification. New approaches are introduced to reduce the issues affected by cloud computing. Because of the newly introduced approaches, the performance of the cloud is increased and the issues are reduced. The following section describes the performance analysis of the proposed strategy, and it will be compared with other ML approaches and existing works later.

4. Results and Discussions

This section introduces the performance of the proposed approach and compares the outcomes of the proposed model with existing works. In this proposed research, UNSW_NB15, NSL_KDD, and CICIDS-2017 datasets are utilized for the implementation. *k*-fold cross-validation is utilized in this research work. The data samples are separated into *k*-folds, and each fold is employed for testing at a particular point. The *k* value is fixed to 10, and this process is repeated 10 times. The findings of the suggested model are produced through the use of the TensorFlow platform and Python 3.7 simulation. The Python programming language is a general-purpose, high-level, and interpreted dynamic programming language that places a strong emphasis on code readability. When compared to other programming languages, such as C++ or JAVA, the syntax in Python requires the least amount of coding. The TensorFlow platform and the Python 3.7 language are used in the IDS to precisely evaluate the detection accuracy and efficiency. Table 2 gives the parameters used in the implementation of the proposed model.

Table 2: Utilized parameters

S. No.	Parameter	Values
1	Population size	100
2	Maximum number of iterations	500
3	Dimensions	5
4	Maximum fitness	1
5	Number of hidden layers	2
6	RBM	2

The results of the simulation are compared to those of other available methodologies, and performance indicators are calculated. The description of the dataset, statistical metrics, and performance analysis of the suggested model are all expanded upon. The introduced model is compared to different existing ML algorithms, including DAE, RNN, DNN, and ANN, which are addressed further below.

- **Deep Neural Network (DNN)**

The DNN [38] is also known as a feed-forward neural network (FFNN), which is a successor to the standard ANN. Output, input, and hidden layers are placed in the DNN. In the network, the preprocessed input data are supplied by the input layer. More than one hidden layer is added to the DNN, as allowed by the DL network. A nonlinear activation function (ReLU) is used by every hidden layer. The elimination of exploding and vanishing gradient issue is the foremost breakthrough of ReLU. From the hidden layer, the output layer processes the inputs to the activation function of the output layers, and it produces the DNN outputs.

- **Recurrent Neural Network (RNN)**

An extension of a traditional FFNN is known as RNN [39]. The RNN activates the same task for every element of a sequence, so it is also known as recurrent. The RNN is trained by the back propagation through time (BPTT) algorithm. The problem of storage space is solved by the RNN because it is the one between ANN unlike other neural networks. The RNN plays a vital part in computer vision, human action recognition, speech recognition, and so on. But, in the design of the potential IDS, it helped to maintain a low FAR and maximum DR, which significantly outdid other methods.

- **Deep Auto-Encoder (DAE)**

The encoder, hidden, and decoder layers comprise the basic framework of the auto-encoder. The decoder layer input is the hidden layer output, and the input of the hidden layer is the encoder layer output. More than one hidden layer is placed in the DAE [40], with each hidden layer by a similar number of neuron inputs as the number of features. By cross-validating the combinations, these parameters are identified, which allows assessment to avoid overfitting risk. The model uses training data with labels and Softmax as the classification layer's activation function.

- **Artificial Neural Network (ANN)**

For supervised classification learning, the ANN [41] is utilized, and it has several high and simple interconnected neurons because it is a computational model. It is the task of individual neurons. An independent processing element is every neuron in a neural network. A summing component shadowed by an activation function is a processing element (neuron). Pattern recognition and categorization or classification is the most effective application of neural networks.

4.1. Dataset Description

The datasets NSL-KDD, UNSW-NB15, and CICIDS-2017 have been used in the suggested technique. The proposed ID is calculated using the NSL-KDD 1999 [42] dataset, which was utilized in this investigation. NSL-KDD 99 is a modified version of the KDD'99 Cup dataset, which was first released in 1999 and has since been updated. This dataset encompasses 148,517 instances, each of which includes 41 attributes and is classified into five categories: DoS, Probe, R2L, and U2R. Also every instance has 41 features and is classified into one of five categories based on those characteristics. The CICIDS-2017 dataset [43] is a collection of 225,745 packages of data with over 80 features and gathered more than five days of network activity. Seven attack categories are there in the CICIDS-2017, including DDoS, DoS, infiltration, web, botnet, heart bleed, and brute force. The UNSW-NB15 dataset [44] contains 2,540,044 records as well as 49 features and nine categories of attack. Reconnaissance, shellcode, worm, genetic, exploit, DoS, backdoor, analysis, and fuzzers are the attacks in the UNSW-NB15 dataset.

4.2. Statistical Measures

The proposed scheme (IDPS) is achieved through proposed algorithms by metrics such as precision, *f*-score, accuracy, FAR, and DR/recall. An IDS should have a minimum false alarm. [14].

- ***Precision:*** The percentage of entire attacks that are accurately discovered. It is measured as follows:

$$\Pr ecision = \frac{tp}{tp + fp} \times 100\,\% \tag{19}$$

- ***F-Score:*** The harmonic mean of precision and recall.

$$F - score = 2\,\frac{pre \times recall}{pre + recall} \tag{20}$$

- ***Classification Accuracy:*** The accurate categorization of entire classifications is known as accuracy. Based on Equation 23, the accuracy is evaluated, which is explained as the summation of diagonal features in the confusion matrix against total components.

$$Classification\ Acc = \frac{tn + tp}{tn + tp + fn + fp} \times 100\,\% \qquad (21)$$

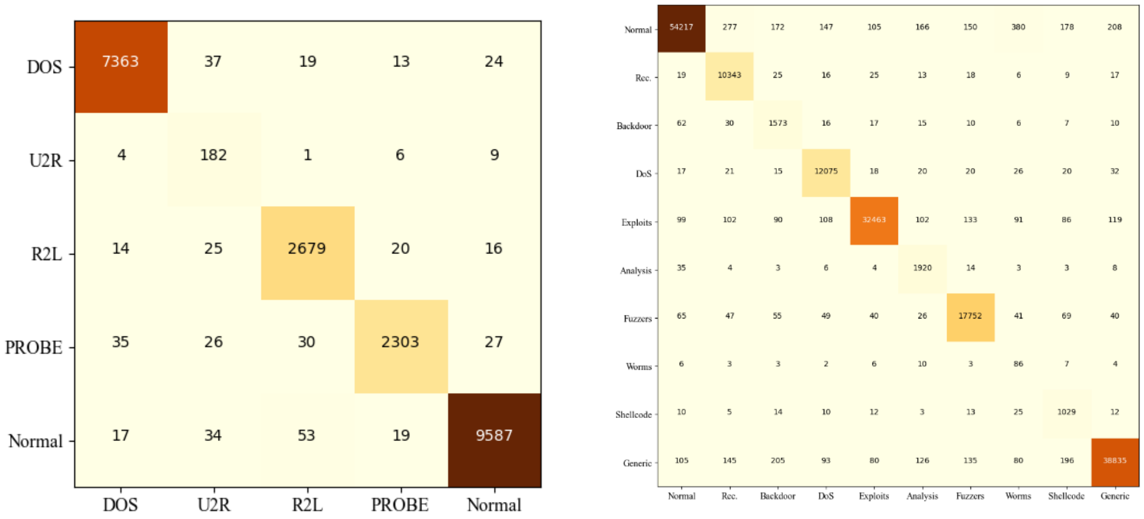
- ***DR and FAR:*** The ratio of the accurate detection rate to the total number of attacks is known as DR, and the ratio of the number of misclassification to the total number of attacks is called FAR.

$$DR = \frac{tp}{tp + fn} \times 100\,\% \qquad (22)$$

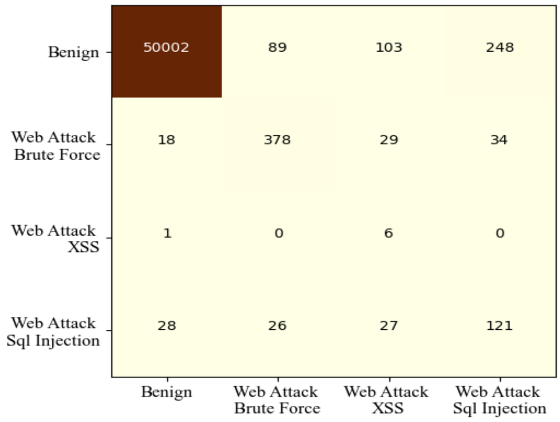
$$FAR = \frac{fp}{fp + tn} \times 100\,\% \qquad (23)$$

4.3. Performance Analysis

Based on the furnished graphical plots, the efficacy of the proposed scheme is compared with the previous works and is shown in this section. In this task, the statistical measures evaluated over the previous ML algorithms like DAE, RNN, DNN, and ANN and existing works such as NB tree and RF [26], CS_DDoS [27], SVM-SPSO [28], integrated rule [29], CADF [30], information gain with SVM [31], FkNN [32], MLP-GA [33], and f C-HMT-BPNN model [34].



(a) (b)



(c)

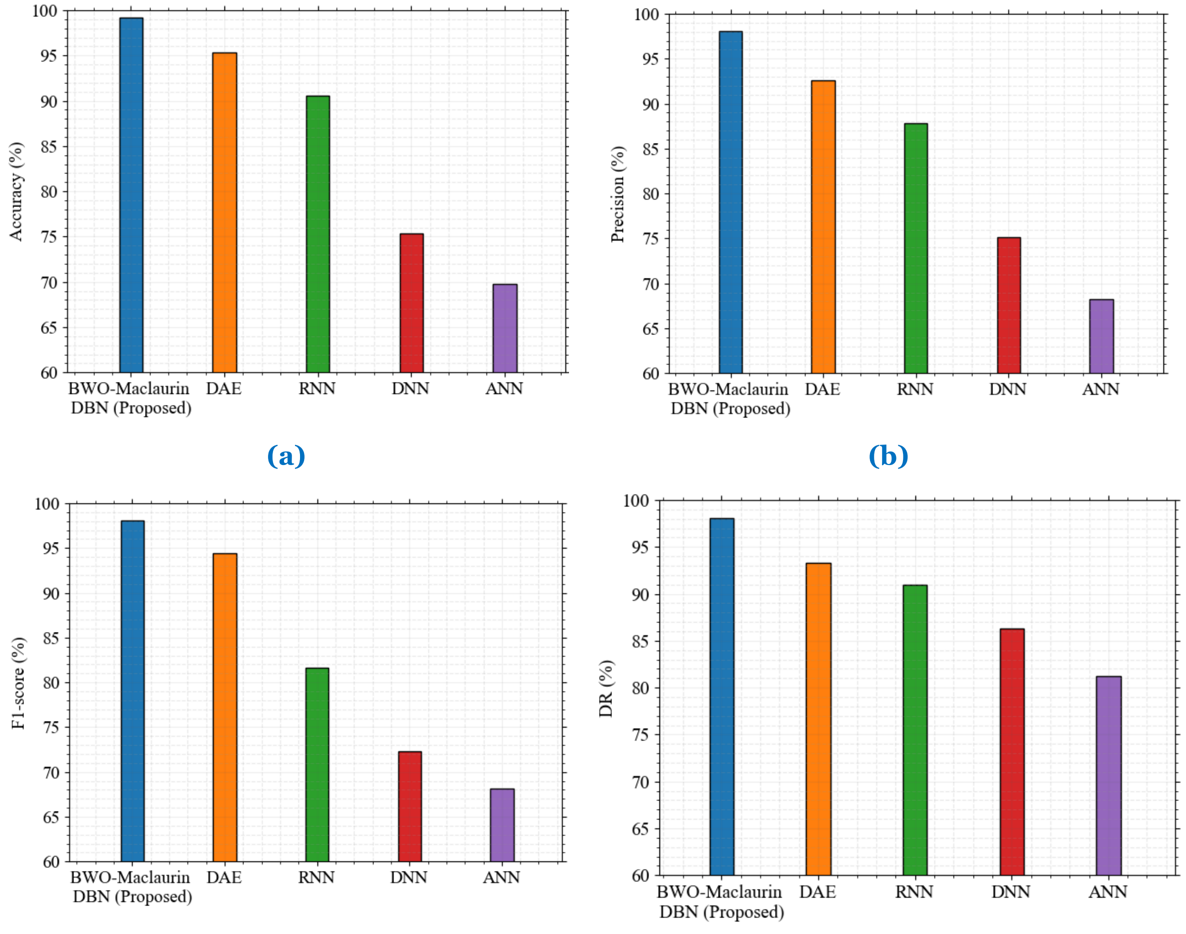
Figure 3: Confusion matrix **(a)** NSL-KDD, **(b)** UNSW-NB15, and **(c)** CICIDS-2017

Figure 3 demonstrates the proposed (BWO-Maclaurin-DBN) strategy's confusion matrix for three varied datasets named (a) NSL-KDD, (b) UNSW-NB15, and (c) CICIDS-2017, which give the confusion matrix between normal and attack. The proposed model achieved 98.9%, accuracy for NSL-KDD, UNSW-NB15, and CICIDS-2017 datasets. The graphical representation of the performance analysis is shown in Figure 3, which represents FAR, precision, DR, accuracy, and F1-score. Table 3 gives the performance values by comparing the proposed scheme with existing ML schemes on the three varied datasets.

Table 3: Performance evaluation values compared with existing schemes

Dataset	***Performance Metrics***	**Precision (%)**	**F1-Score (%)**	**Accuracy (%)**	**FAR**	**DR (%)**
NSL-KDD	**BWO-Maclaurin-DBN (Proposed)**	98.1	98.09	99.23	0.175	98.12
	DAE	92.61	94.39	95.38	0.20	93.34
	RNN	87.82	81.84	90.52	0.21	90.99
	DNN	75.17	72.28	75.32	0.36	86.28
	ANN	68.27	68.27	69.73	0.445	81.19
UNSW-NB15	**BWO-Maclaurin-DBN (Proposed)**	98.82	98.20	99.41	0.17	98.83
	DAE	94.9	94.2	94.11	0.19	93.25
	RNN	88.6	92.4	89.2	0.21	92.95
	DNN	67	76	75.9	0.35	91.71
	ANN	65	74	73.5	0.41	85.91
CICIDS-2017	**BWO-Maclaurin-DBN (Proposed)**	97.11	97.3	99.42	0.165	97.12
	DAE	93.71	91.72	93.41	0.19	92.27
	RNN	89.67	80.34	91.72	0.22	90.74
	DNN	74.28	74.845	78.54	0.38	88.67
	ANN	66.317	71.27	72.23	0.45	83.41

Figure 4 displays the performance comparison of the proposed scheme in terms of precision, F1-score, accuracy, FAR, and DR on the NSL-KDD dataset. The proposed approach (Maclaurin-DBN) has achieved 98.1%, 98.09%, 99.23%, 0.175, and 98.12% in precision, F1-score, accuracy, FAR, and DR values, respectively. The optimal features are obtained by the BWO approach, which produced good convergence and coverage outcomes. F1-score is the term combining precision and recall; here, precision and recall obtained better outcomes. Hence, the F1-score value is also better than in other approaches. The best feature selection and accurate prediction give the maximum accuracy in the proposed model. The convergence speed and coverage value of the proposed BWO approach are maximum. Hence, the proposed approach obtains the minimum FAR and minimum computation time. The DBN's classification accuracy is greatest; hence, the accuracy of the proposed scheme is also maximum. Besides, the DR is also improved by the DBN, and accurate prediction gives maximum DR performance. Compared to previous strategies, the DR of the proposed approach has a maximum DR value.

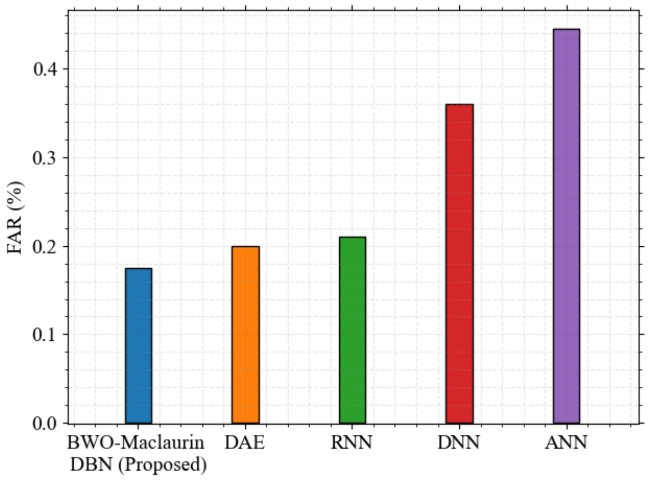


(a)

(b)

(c)

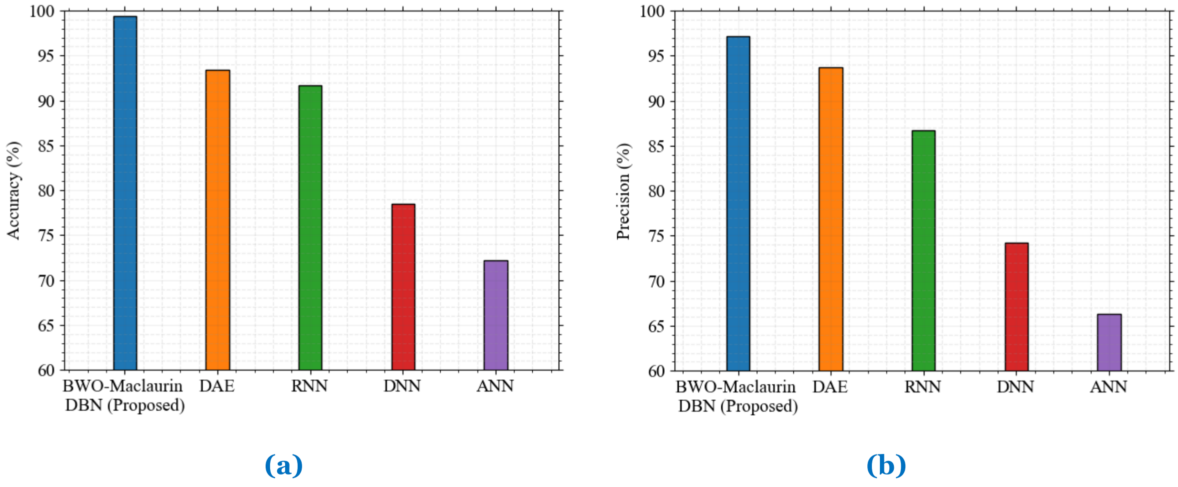
(d)



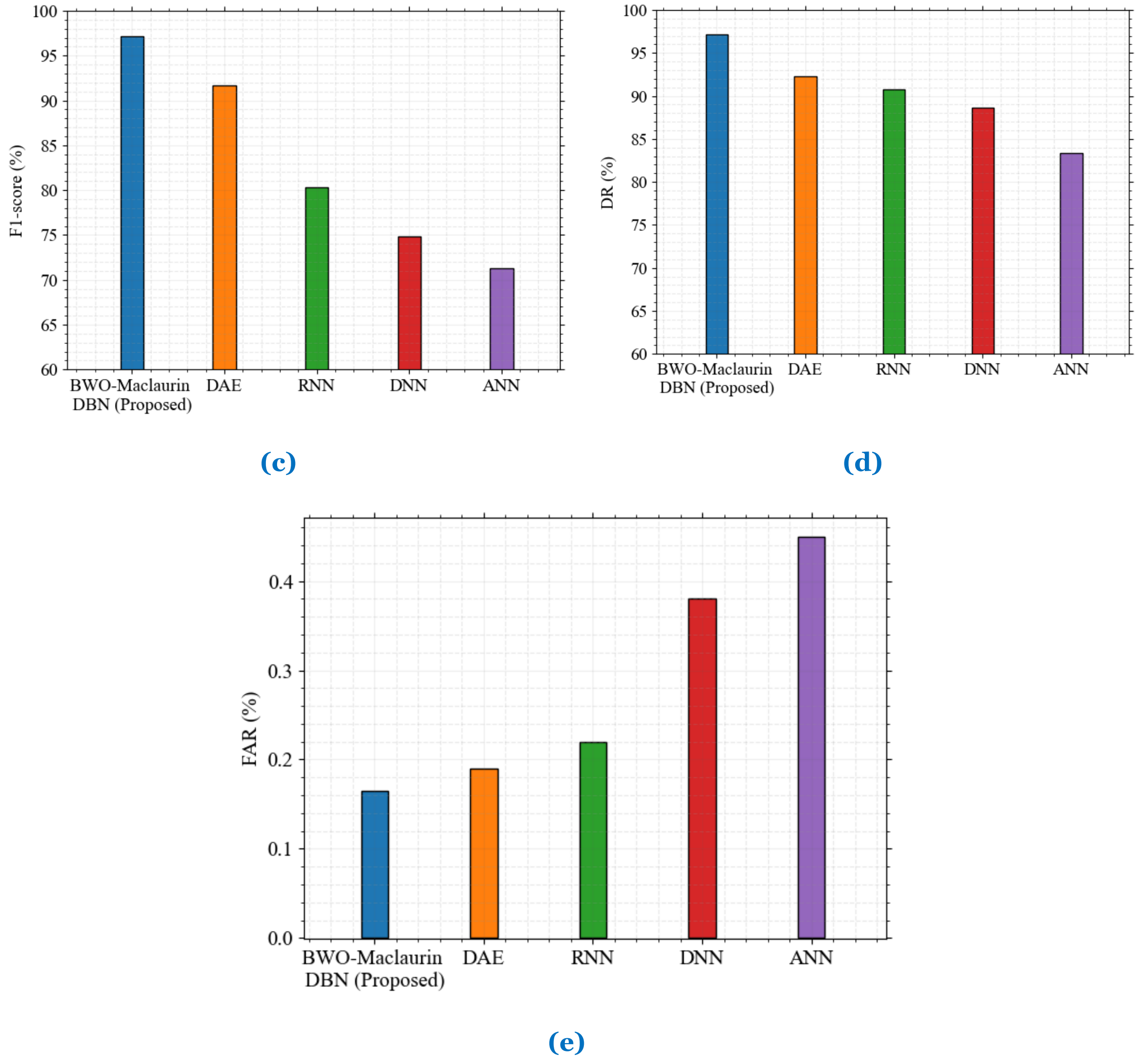
(e)

Figure 4: Performance comparison: (a) Accuracy, (b) Precision, (c) F1-score, (d) DR, and (e) FAR on the NSL-KDD dataset

Figure 5 displays the performance comparison of the proposed scheme in terms of precision, F1-score, accuracy, FAR, and DR on the UNSW-NB15 dataset. The proposed approach (Maclaurin-DBN) achieved 98.82%, 98.20%, 99.41%, 0.17, 98.83% in precision, F1-score, accuracy, FAR, and DR values, respectively. The optimal features are obtained by the BWO approach, which produced good convergence and coverage outcomes. F1-score is the term combining precision and recall; here, precision and recall obtained better outcomes. Hence, the F1-score value is also better than in other approaches. The best feature selection and accurate prediction give the maximum accuracy in the proposed model. The convergence speed and coverage value of the proposed BWO approach are maximum. Hence, the proposed approach obtains the minimum FAR and minimum computation time. The classification accuracy of the DBN is greatest; hence, the accuracy of the proposed scheme is also maximum. Besides, the DR is also improved by the DBN, and accurate prediction gives maximum DR performance. Compared to other existing strategies, the DR of the proposed approach has a maximum DR value.



(a) **(b)**



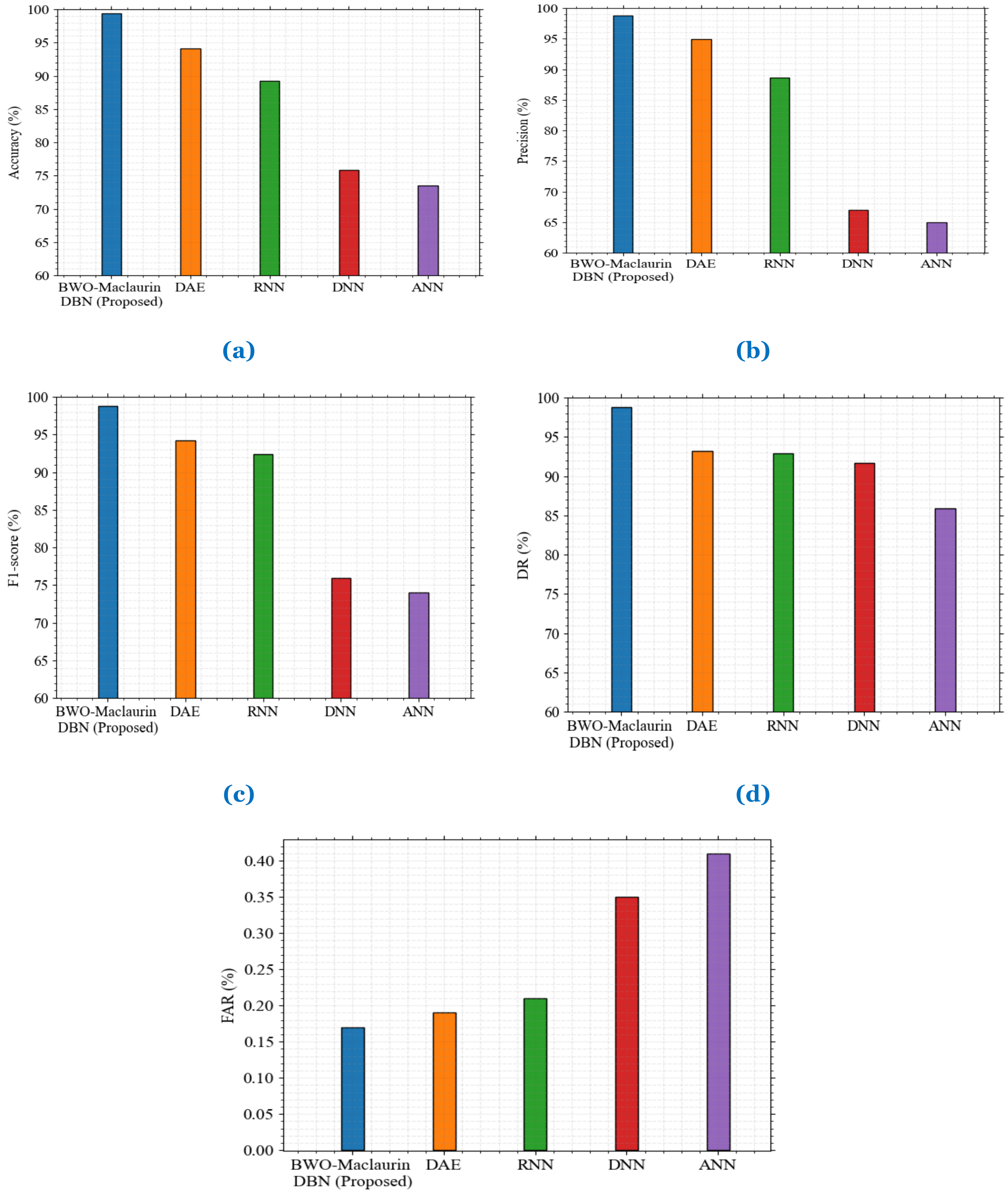
(c)

(d)

(e)

Figure 5: Performance comparison: (a) Accuracy, (b) Precision, (c) F1-score, (d) DR, and (e) FAR on the UNSW-NB15 dataset

Figure 6 displays the performance comparison of the proposed scheme in terms of precision, F1-score, accuracy, FAR, and DR on the CICIDS-2017 dataset. The proposed approach (Maclaurin-DBN) achieved 97.11%, 97.3%, 99.42%, 0.165, and 97.12% in precision, F1-score, accuracy, FAR, and DR values, respectively. The optimal features are obtained by the BWO approach, which produced good convergence and coverage outcomes. F1-score is the term combining precision and recall; here, precision and recall obtained better outcomes. Hence, the F1-score value is also better than in other approaches. The best feature selection and accurate prediction give the maximum accuracy in the proposed model. The convergence speed and coverage value of the proposed BWO approach are maximum. Hence, the proposed approach obtains the minimum FAR and minimum computation time. The DBN's classification accuracy is greatest; hence, the accuracy of the proposed scheme is also maximum. Besides, the DR is also improved by the DBN, and accurate prediction gives the maximum DR performance. Compared to other strategies, the DR of the introduced approach has a maximum DR value.



(a)

(b)

(c)

(d)

(e)

Figure 6: Performance comparison: (a) Accuracy, (b) Precision, (c) F1-score, (d) DR, and (e) FAR on CICIDS-2017 dataset

Table 4 gives the performance comparison values of the proposed approach with existing works such as NB tree and RF [23], CS_DDoS [24], and SVM-SPSO [25] on the NSL-KDD dataset. The precision, F1-score, accuracy, FAR, and DR values are compared with existing works. The proposed scheme achieved very low (0.175) FAR compared to existing works. Hence, the proposed research work achieved high classification accuracy. The proposed scheme detected the intrusion effectively by the Maclaurin-DBN and effective feature selection BWO. So the proposed scheme achieved maximum classification accuracy and a precise DR. The proposed scheme achieved 99.23% classification accuracy and 98.12% DR on the NSL-KDD dataset. The existing works such as NB tree and RF [23], CS_DDoS [24], and SVM-SPSO [25] had 99.07%, 97%, and 99.1% classification accuracy, respectively, and 99.1%, 97%, and 99.08% DR, respectively. This illustrated that the proposed scheme outclassed the other models in performance.

Table 4: Performance comparison with existing works on the NSL-KDD dataset

Performance Metrics	**Precision (%)**	**F1-Score (%)**	**Accuracy (%)**	**FAR**	**DR (%)**
BWO-Maclaurin-DBN (Proposed)	98.1	98.09	99.23	0.175	98.12
NB tree and RF [26]	99	99	99.07	1.2	99.1
CS_DDoS [27]	97	97	97	0.35	97
SVM-SPSO [28]	99.05	99.07	99.1	0.8	99.08

Table 5 compares the suggested technique to current studies such as integrated rule [29], CADF [30], and information gain with SVM [31] on the UNSW-NB15 dataset. It is compared to existing works in terms of precision, F1-score, accuracy, FAR, and DR values. This study endeavour obtained maximum classification accuracy because it achieved a minimum FAR value of 0.17, which was the lowest possible value. A combination of the Maclaurin-DBN and an effective feature selection BWO helped to detect the intrusion efficiently in the proposed methodology. As a result, the proposed approach attained the highest possible classification accuracy as well as the highest possible precise DR. On the UNSW-NB15 dataset, the proposed system achieved 99.41% classification accuracy and 98.83% DR. It was shown that previous works such as integrated rule [29], CADF [30], and information gain with SVM [31] had a classification accuracy of 84.83%, 96.7%, and 90.41%, respectively, and a DR of 57.01%, 95.6%, and 77.91%, respectively. This demonstrated that the proposed system outperformed the other models in terms of performance.

Table 5: Performance comparison with existing works on the UNSW-NB15 dataset

Performance Metrics	**Precision (%)**	**F1-Score (%)**	**Accuracy (%)**	**FAR**	**DR (%)**
BWO-Maclaurin-DBN (Proposed)	98.82	98.20	99.41	0.17	98.83
Integrated rule [29]	90.32	68.13	84.83	2.01	57.01
CADF [30]	-	-	96.7	3.5	95.6
Information gain with SVM [31]	90.75	83.84	90.41	-	77.91

Table 6 gives the performance comparison values of the proposed approach with existing works such as FkNN [32], MLP-GA [33], and SwiftIDS [34] on the CICIDS-2017 dataset. The precision, F1-score, accuracy, FAR, and DR values are compared with existing works. The proposed research work achieved maximum classification accuracy because it achieved a minimum of 0.165 in FAR value. The proposed scheme detected the intrusion effectively by the Maclaurin-DBN and an effective feature selection BWO. So the proposed scheme achieved the maximum classification accuracy and a precise DR. The proposed scheme achieved 99.42% classification accuracy and 97.12% DR on the CICIDS-2017 dataset. The existing works such as FkNN [32], MLP-GA [33], and SwiftIDS [34] had 99.79%, 90%, and 99.02% classification accuracy, respectively, and 99.89%, 89.2%, and 99.93% DR, respectively. This illustrated that the proposed scheme outclassed the other models in terms of performance.

Table 6: Performance comparison with existing works on the CICIDS-2017 dataset

Performance Metrics	**Precision (%)**	**F1-Score (%)**	**Accuracy (%)**	**FAR**	**DR (%)**
BWO-Maclaurin-DBN (Proposed)	97.11	97.3	99.42	0.165	97.12
FkNN [32]	99.89	-	99.79	-	99.89
MLP-GA [33]	-	89.3	90	-	89.2
SwiftIDS [34]	98.88	99.40	99.02	0.048	99.93

Table 7 gives the duration of testing and training with the features extracted. The proposed scheme is compared with the SVM-SPSO [28], and compared to previous works, the proposed system took a lesser amount of time to process the features.

Table 7: IDS training and testing time with features

Measures	**SVM-SPSO [28]**	**Proposed**
Number of features	23	11
Training time (s)	5.3	2.6
Testing time (s)	7.8	3.4

Based on traffic behaviour, malicious activity is identified in the network; this is known as a network IDS. In this proposed research, the NSL-KDD 99 dataset is used to find malicious activity in cloud environs. Here, the collected data are sent to the feature selection section for selecting feasible features. The optimal features are selected based on the BWO procedure, which is the optimization first utilized for feature selection. It has high coverage and convergence because of the target selection, and it improves diversity. Then the obtained features are transmitted to the classification to predict the attacks effectively. Here, the Maclaurin-DBN accurately predicted the malicious and regular records. The Maclaurin-DBN has produced the most accurate outcome, and it produces the results within a short duration. The introduced strategy produces maximum outcomes in terms of accuracy, DR, precision, FAR, and *f*-score compared with many other DL approaches. Every outcome is obtained by implementing the proposed flow in the Python environment. After the intrusion detection, the prevention of intrusion is also activated in this research. The affected VM is detected by the IDS, and then the information from the affected VM is changed to the unaffected VM. Then the unaffected VM is migrated to another free space server to prevent the intrusion. At last, the affected VM from the server is shut down. Hence, the affected server is prevented from operating on a cloud.

5. Conclusion

The IDPs on the cloud proposed in this research work use BWO and Maclaurin-DBN machine learning. Here, the optimal feasible features are extracted by the meta-heuristic BWO, and the extracted features are given to the proposed Maclaurin-DBN classifier as the training data input. The bias and weights in the DBN are trained by the Maclaurin series to categorize the intrusion in the cloud network. The implementation of the introduced Maclaurin-DBN is processed using NSL-KDD, UNSW-NB15, and CICIDS-2017 databases on the Python Spyder 3.7 platform. A low FAR and high detection accuracy are achieved in this research, which was the main aim of this research work. By varying the training percentages, the simulation was done, and the design responses against the FAR, DR, and precision, respectively, were analysed. Besides, the proposed Maclaurin-DBN classifier is compared with several ML algorithms and existing works. This shows that the proposed scheme achieved maximum outcomes compared to others. The proposed strategy, BWO-Maclaurin-DBN, achieved the maximum classification accuracy and a DR of 95.38% and 93.34%, 99.41% and 98.83%, 99.42% and 97.12% on the NSL-KDD, UNSW-NB15, and CICIDS-2017 datasets, respectively. Thus, in the future, research work will be directed to a real-time process.

References

[1] Nagar U, Nanda P, He X and Tan ZT. A framework for data security in cloud using collaborative intrusion detection scheme. In Proceedings of the 10th International Conference on Security of Information and Networks 2017; 188-193.

[2] Nagarajan, N. and Thirunavukarasu, R., 2020. Service-oriented Broker for Effective Provisioning of Cloud Services–a Survey. International Journal of Computing and Digital Systems, 9(5), pp.863-879.

[3] Modi C and Patel D. A feasible approach to intrusion detection in virtual network layer of Cloud computing. Sādhanā 2018; 43(7): 114.

[4] Wahab OA, Bentahar J, Otrok H and Mourad A. Optimal load distribution for the detection of VM-based DDoS attacks in the cloud. IEEE Transactions on Services Computing 2017; 1: 1-1.

[5] Kumara A and Jaidhar CD. Hypervisor and virtual machine dependent Intrusion Detection and Prevention System for virtualized cloud environment. In Telematics and Future Generation Networks (TAFGEN), 2015 1st International Conference on 2015; 28-33.

[6] Rajganesh, N. and Ramkumar, T., 2016. A review on broker based cloud service model. Journal of computing and information technology, 24(3), pp.283-292.

[7] Nagarajan, R. and Thirunavukarasu, R., 2018, June. A review on intelligent cloud broker for effective service provisioning in cloud. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 519-524). IEEE.

[8] Jin H, Xiang G, Zou D, Wu S, Zhao F, Li M and Zheng W. A VMM-based intrusion prevention system in cloud computing environment. The Journal of Supercomputing 2013; 66(3): 1133-1151.

[9] Kholidy HA, Erradi A and Abdelwahed S. Attack Prediction Models for Cloud Intrusion Detection Systems. In Artificial Intelligence, Modelling and Simulation (AIMS), 2014 2nd International Conference on 2014; 270-275.

[10] Deshpande P, Sharma SC, Peddoju SK and Junaid S. HIDS: A host based intrusion detection system for cloud computing environment. International Journal of System Assurance Engineering and Management 2018; 9 (3): 567-576.

[11] Arjunan K and Modi CN. An enhanced intrusion detection framework for securing network layer of cloud computing. In Asia Security and Privacy (ISEASP), 2017 ISEA 2017; 1-10.

[12] Wankhade KRD. Virtualization Intrusion Detection System in Cloud Environment. International Journal of Scientific & Engineering Research 2016; 7(2): 321-328.

[13] Gülmez HG, Tuncel E, and Angin P. A Big Data Analytical Approach to Cloud Intrusion Detection. In International Conference on Cloud Computing, Springer, Cham 2018; 377-388.

[14] Nikolai J and Wang Y. Hypervisor-based cloud intrusion detection system. In Computing, Networking and Communications (ICNC) 2014; 989-993.

[15] Cherkaoui R, Braeken MZ and Touhafi A. Performance Analysis of Intrusion Detection Systems in Cloud-Based Systems. In International Symposium on Ubiquitous Networking, Springer, Cham 2017; 206-213.

[16] Ibrahim AS, Hamlyn-Harris J and Grundy J. Emerging security challenges of cloud virtual infrastructure. Ar Xiv preprint ar Xiv 2016; 1612.09059.

[17] Balamurugan V and Saravanan R. Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation. Cluster Computing 2017: 1-13.

[18] Krishnaveni S, Prabakaran S and Sivamohan S. Automated Vulnerability Detection and Prediction by Security Testing for Cloud SAAS. Indian Journal of Science and Technology 2016; 9(S1).

[19] Dhote PM, Nitnaware UL, Shilpa A, Pimple BJ and Student BE. Cloud Based Attack Detection. International Journal of Engineering Science 2018; 17051.

[20] Tang TA, Mhamdi L, McLernon D, Zaidi SAR and Ghogho M. Deep learning approach for network intrusion detection in software defined networking. In 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM) 2016; 258-263.

[21] Sultana N, Chilamkurti N, Peng W and Alhadad R. Survey on SDN based network intrusion detection system using machine learning approaches. Peer-to-Peer Networking and Applications 2019; 12(2): 493-501.

[22] Ouffoué G, Ortiz AM, Cavalli AR, Mallouli W, Domingo-Ferrer J, David S and Zaidi F. Intrusion detection and attack tolerance for cloud environments: the CLARUS approach. In Distributed Computing Systems Workshops (ICDCSW), 2016 IEEE 36th International Conference on IEEE. 2016; 61-66.

[23] Bharati M and Tamane S. Intrusion detection systems (IDS) & future challenges in cloud based environment. In Intelligent Systems and Information Management (ICISIM), 2017 1st International Conference on IEEE 2017; 240-250.

[24] Mishra P, Varadharajan V, Pilli E and Tupakula U. VM Guard: A VMI-based Security Architecture for Intrusion Detection in Cloud Environment. IEEE Transactions on Cloud Computing 2018.

[25] Chung CJ, Khatkar P, Xing T, Lee J and Huang D. NICE: Network intrusion detection and countermeasure selection in virtual network systems. IEEE transactions on dependable and secure computing 2013; 10(4): 198-211.

[26] Bhat AH, Patra S and Jena D. Machine learning approach for intrusion detection on cloud virtual machines. International Journal of Application or Innovation in Engineering & Management (IJAIEM) 2013; 2(6): 56-66.

[27] Sahi A, Lai D, Li Y and Diykh M. An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. IEEE Access 2017; 5: 6036-6048.

[28] Sakr, M.M., Tawfeeq, M.A and El-Sisi, A.B. Network intrusion detection system based PSO-SVM for cloud computing. International Journal of Computer Network and Information Security, 2019; 10(3): 22.

[29] Kumar, V., Sinha, D., Das, A.K., Pandey, S.C. and Goswami, R.T., 2020. An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. Cluster Computing, 23(2), pp.1397-1418.

[30] Moustafa, N., Creech, G., Sitnikova, E. and Keshk, M., 2017, November. Collaborative anomaly detection framework for handling big data of cloud computing. In 2017 Military Communications and Information Systems Conference (MilCIS) (pp. 1-6). IEEE.

[31] Sarumi, O.A., Adetunmbi, A.O. and Adetoye, F.A., 2020. Discovering computer networks intrusion using data analytics and machine intelligence. Scientific African, 9, p.e00500.

[32] Krishna, K.V., Swathi, K. and Rao, B.B., A Novel Framework for NIDS through Fast kNN Classifier on CICIDS2017 Dataset.

[33] Singh, P. and Ranga, V., 2020. Multilayer Perceptron and Genetic Algorithm-Based Intrusion Detection Framework for Cloud Environment. In Mobile Radio Communications and 5G Networks (pp. 475-485). Springer, Singapore.

[34] Jin, D., Lu, Y., Qin, J., Cheng, Z. and Mao, Z., 2020. SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism. Computers & Security, 97, p.101984.

[35] Hayyolalam, V. and Kazem, A.A.P., 2020. Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 87, p.103249.

[36] Yang Y, Zheng K, Wu C, Niu X and Yang Y. Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. Applied Sciences 2019; 9(2): 238.

[37] Mangai SA, Sankar BR and Alagarsamy K. Taylor series prediction of time series data with error propagated by artificial neural network. International Journal of Computer Applications, 2014; 89(1).

[38] Devan P and Khare N. An efficient XGBoost–DNN-based classification model for network intrusion detection system. Neural Computing and Applications, 2020; 1-16.

[39] Neha N, Priyanga S, Seshan S, Senthilnathan R and Sriram VS. SCO-RNN: A Behavioral-Based Intrusion Detection Approach for Cyber Physical Attacks in SCADA Systems. In Inventive Communication and Computational Technologies Springer, Singapore. 2020; 911-919.

[40] Wang W, Zhao M and Wang J. Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. Journal of Ambient Intelligence and Humanized Computing, 2019; 10(8): 3035-3043.

[41] Manzoor I and Kumar N. A feature reduced intrusion detection system using ANN classifier. Expert Systems with Applications, 2017; 88: 249-257.

[42] Wang W, Du X and Wang N. Building a cloud IDS using an efficient feature selection method and SVM. IEEE Access, 2018; 7: 1345-1354.

[43] Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A., 2018, January. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In ICISSP (pp. 108-116).

[44] Janarthanan, T. and Zargari, S., 2017, June. Feature selection in UNSW-NB15 and KDDCUP'99 datasets. In 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE) (pp. 1881-1886). IEEE.