sensors

MDPI

Article

Multi-Objective Seagull Optimization Algorithm with Deep Learning-Enabled Vulnerability Detection for Secure Cloud Environments

Mohammed Aljebreen [1], Manal Abdullah Alohali [2,*], Hany Mahgoub [3], Sumayh S. Aljameel [4], Albandari Alsumayt [5] and Ahmed Sayed [6]

[1] Department of Computer Science, Community College, King Saud University, P.O. Box 28095, Riyadh 11437, Saudi Arabia
[2] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[3] Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Abha 61413, Saudi Arabia
[4] SAUDI ARAMCO Cybersecurity Chair, Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
[5] Department of Computer Science, Applied College, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
[6] Research Center, Future University in Egypt, New Cairo 11835, Egypt; a.sayed@fueinstitute.net
* Correspondence: maalohaly@pnu.edu.sa

Abstract: Cloud computing (CC) is an internet-enabled environment that provides computing services such as networking, databases, and servers to clients and organizations in a cost-effective manner. Despite the benefits rendered by CC, its security remains a prominent concern to overcome. An intrusion detection system (IDS) is generally used to detect both normal and anomalous behavior in networks. The design of IDS using a machine learning (ML) technique comprises a series of methods that can learn patterns from data and forecast the outcomes consequently. In this background, the current study designs a novel multi-objective seagull optimization algorithm with a deep learning-enabled vulnerability detection (MOSOA-DLVD) technique to secure the cloud platform. The MOSOA-DLVD technique uses the feature selection (FS) method and hyperparameter tuning strategy to identify the presence of vulnerabilities or attacks in the cloud infrastructure. Primarily, the FS method is implemented using the MOSOA technique. Furthermore, the MOSOA-DLVD technique uses a deep belief network (DBN) method for intrusion detection and its classification. In order to improve the detection outcomes of the DBN algorithm, the sooty tern optimization algorithm (STOA) is applied for the hyperparameter tuning process. The performance of the proposed MOSOA-DLVD system was validated with extensive simulations upon a benchmark IDS dataset. The improved intrusion detection results of the MOSOA-DLVD approach with a maximum accuracy of 99.34% establish the proficiency of the model compared with recent methods.

Keywords: cloud computing; deep learning; intrusion detection system; sooty tern optimization algorithm; seagull optimization algorithm

Citation: Aljebreen, M.; Alohali, M.A.; Mahgoub, H.; Aljameel, S.S.; Alsumayt, A.; Sayed, A. Multi-Objective Seagull Optimization Algorithm with Deep Learning-Enabled Vulnerability Detection for Secure Cloud Environments. *Sensors* **2023**, *23*, 9383. https://doi.org/10.3390/s23239383

Academic Editors: Paolo Trunfio and Francesco Mercaldo

Received: 8 August 2023
Revised: 8 November 2023
Accepted: 14 November 2023
Published: 24 November 2023

1. Introduction

Cloud computing (CC) offers numerous services to users including infrastructure, storage capabilities, and applications [1]. A cloud user can manipulate or access software and hardware over the internet based on their requirements. Though CC provides several advantages to its users, it also has certain limitations and challenges. These challenges include performance management, privacy, security, cost, and load balance [2]. Among the issues encountered by the cloud computing phenomenon, security plays a major role in

user data and applications on the cloud infrastructure. CC security encompasses policies and procedures to protect cloud-based information, applications, and frameworks from unauthorized access and attacks [3]. Also, it protects data and infrastructure against Structured Query Language (SQL) injection, software vulnerability, flooding attacks, cross-site scripting, data alteration, and data leakage. In parallel, cloud providers and subscribers continuously report security problems raised by different types of attacks. Hence, it is necessary to provide security against malicious activities and attacks [4].

Intrusion detection systems (IDSs) [5] in cloud networks play a crucial role in terms of providing security against attacks from both outsiders as well as insiders [6]. Traditional IDSs are used in the detection of attacks in internet environments. However, they cannot adjust their working mechanisms for cloud platforms and so remain non-scalable. Furthermore, researchers found them to be not appropriate for cloud platforms and not deterministic [7]. Therefore, new and reliable anomaly based IDSs have been proposed, developed, and validated. Mostly, the existing methods for anomaly detection from cloud platforms used machine learning (ML) approaches. These methods can enhance their performance by upgrading their data according to the pattern detected from the input datasets [8]. When a novel pattern is detected from the input dataset, the ML technique parameters are upgraded for the detection of the same anomalies in future traffic flow. According to the data extracted from the prior outcomes, the solution of the method is enhanced by altering the implementation approach, if required. The feature selection (FS) process helps to focus only on the most related information. FS is an ML method that reduces the quantity of the data to be analyzed [9]. It can be achieved by detecting the relevant features (such as the attributes) of a dataset, leaving behind the insignificant ones. By reducing the dimensionality of a dataset, i.e., retaining only the relevant features, the ML technique can make the classification prediction process an efficient and effective one [10]. This efficacy is particularly related to the intrusion detection (ID) process that needs real-time performance.

The current study designs a new multi-objective seagull optimization algorithm with a deep learning-enabled vulnerability detection (MOSOA-DLVD) system for a secure cloud platform. In the developed MOSOA-DLVD algorithm, the feature selection process is performed with the help of the MOSOA technique. Furthermore, the MOSOA-DLVD technique uses a deep belief network (DBN) method for intrusion detection and classification. To enhance the detection results of the DBN algorithm, the sooty tern optimization algorithm (STOA) is implemented for the hyperparameter tuning process. The performance of the MOSOA-DLVD system is examined with simulations using a benchmark IDS database. The main contributions of the current study are briefly given below.

- Development of an automated intrusion detection system for the cloud platform, named the MOSOA-DLVD algorithm, which involves MOSA-based FS, DBN-based classification, and STOA-related hyperparameter tuning. To the best of the authors' knowledge, the MOSOA-DLVD system was previously non-existent in the literature.
- The development of the MOSOA approach supports the selection of related features, increases accuracy, and reduces higher dimensionality issues.
- Hyperparameter tuning of the DBN model, using the STOA, enhances the prediction outcomes of the MOSOA-DLVD algorithm for hidden data.

The remaining sections of this paper are explained here. Section 2 offers the related works, and Section 3 provides details about the developed model. Next, Section 4 discusses the outcomes of the analyses, and Section 5 concludes this paper.

2. Related Works

Kavitha et al. [11] examined filter-based ensemble-FS (FEFS) and used the DL method to overcome the problems faced in CC. FEFS is an integration of three feature extraction approaches, namely, embedded, filter, and wrapper methods. In these feature extraction models, the important features were selected to enable the trained model in the DL technique. Lastly, the classifier accomplished the FS. The DL method was an integration of two

techniques including the Tasmanian devil optimization (TDO) and the recurrent neural network (RNN). The authors [12] developed an innovative IDS, which incorporates the fuzzy C-means (FCM) technique with SVM to improve the accuracy of the recognition systems at CC platforms. Maheswari et al. [13] suggested a hybrid soft computing-assisted IDS, i.e., ST-IDS for cloud and web platforms. The authors proposed an IDS system for CC and web infrastructure by utilizing the hybrid teacher learning-enabled-DRNN (TL-DRNN) and cluster-related feature optimizer. In their study, the modified manta ray foraging optimizer (MMFO) was used after feature extraction in the selection of optimum features for accurate detection. The hybrid TL-DRNN was devised to classify the intrusions from the web and cloud platforms. In [14], the authors proposed a dual-channel capsule generative adversarial network (GAN) optimized with RFO algorithm-fostered IDS (IDS-CC-DCCGAN-RFOA) to ensure privacy and secure the CC platform from different types of attacks. According to the best features, the data were categorized into two models, namely, privacy attack and secured data, depending on the DCCGAN outcome. Then, the weight of the DCCGAN model was optimally fine-tuned utilizing the RFO method to accomplish the efficient and best outcomes in terms of intrusion detection.

In a study conducted earlier [15], the authors developed the LR-based oppositional tunicate FCM (LR-OTSFCM) method for cloud ID. The important part of this study is the identification of the attacks in the cloud platform. In [16], a novel hybridization approach was suggested for the IDS to enhance the complete security of the cloud-based computing platforms. In addition, the SMO technique was also used in that study to reduce the dimensionality reduction. The datasets were fed into a neural network (NN). The authors [17] recommended the efficient dragonfly-improved invasive weed optimizer-assisted Shepard-CNN (DIIWO-based ShCNN) technique for identifying the attackers and alleviating the attacks in the cloud model. It is highly possible for the model to detect intruders with ShCNN. In [18], an efficient IDS, termed the chronological salp swarm algorithm-based DL model, was designed to identify suspicious intrusions in the cloud platform. The presented method was developed by combining the chronological idea and SSA. The optimum solution to detect the intrusion was exposed by utilizing the fitness function (FF), which considers the minimum error value as the optimal result. In a study conducted earlier [19], a novel design for deep LSTM-based IDS was presented for detecting the network traffic flow designs from the cloud platform and distinguishing them as malicious or normal patterns. The presented IPS avoids the malicious attacks received in the IDS by improving the recognition rate of the malicious attacks and reducing the computational time. The DNN with game theory for cloud security (GT-CSDNN) model was presented in a study conducted earlier [20]. The developed model covered either attacker or defender approaches but used the game theory algorithm. Furthermore, the DNN model utilized the presented game theory approach for classifying the attacks from regular data. In [21], a new ML-based hybrid IDS was presented. In that study, the integrated SVM and GA approach was established with a novel FF to evaluate the accuracy of the system.

Alohali et al. [22] presented the improved metaheuristics with a fuzzy logic-based intrusion detection system for cloud security (IMFL-IDSCS) technique. For their study, an individual IDS sample was deployed, and the IMFL-IDSCS technique used the enhanced chimp optimization algorithm-based feature selection (ECOA-FS) method for the selection of the optimal features, followed by the adaptive neuro-fuzzy inference system (ANFIS) model. In a study conducted earlier [23], the authors suggested a novel IDS by combining leader-based K-means clustering (LKM) and an optimal fuzzy logic system. Initially, the input dataset was grouped into clusters using the LKM technique. Then, the cluster data were fed into the fuzzy logic system (FLS). Both normal and abnormal data were inquired by the FLS, whereas the FLS was trained with the grey wolf optimization algorithm by maximizing the rules. Mahmood et al. [24] proposed an approach for obtaining the optimal number of features so as to build an efficient IDS model. In their study, feature reduction was applied. Generalization ability can be improved in general by generating a small set of

features from the actual input variables using feature extraction. For their study, a hybrid algorithm, named the principal component analysis neural network algorithm (PCANNA), was used to reduce the number of computer resources.

Although several studies have been conducted for intrusion detection in the cloud platform, the prominence of the FS with hyperparameter tuning for differentiating attacked traffic from normal traffic is yet to be completely studied. Though the implementation of the ML-based IDS was developed earlier, the unique dynamics of the cloud platforms, represented by its various and dynamic workloads, demand specified methods. The existing research shortages drive the demand for a comprehensive scheme that can select important and essential features from the massive quantity of accessible data in order to increase the proficiency and performance of the intrusion detection process. On the other hand, fine-tuning the hyperparameters is frequently disregarded, which in turn results in sub-optimum model effectiveness. Additionally, the important aids of ensemble learning, in which many detection frameworks are incorporated to use their collected predictive capability, are not progressively combined into the ID pipeline. To overcome this research gap, it is vital to design a highly robust and effective intrusion detection technique that is customized according to the particular challenges, modeled with cloud platforms. This way, it becomes possible to finally improve their security posture and alleviate the development of threats. So, it is essential to enhance the generalizability, robustness, and accuracy of the intrusion detection methods, mainly in dynamic and developing network infrastructures. However, the attacks endure to develop in such sophistication and complication as well. Both FS and hyperparameter tuning include various search spaces. FS normally contains a discrete search space, whereas various integrations of the features are estimated. Alternatively, hyperparameter tuning often comprises semi-continuous or continuous search spaces for parameter values. The contribution of MOSOA for FS and STOA, in terms of hyperparameter tuning, allows every method to consider its corresponding search space and the multiplication of its efficacy and performance. MOSOA was developed for multi-objective optimizer tasks, which makes it a well-suitable FS. However, the aim is to enhance numerous conflicting criteria, namely, interpretability, accuracy, and dimensionality reduction. On the contrary, STOA can be highly proficient at enhancing hyperparameters because of its unique optimization approaches.

3. The Proposed Model

In the current study, the authors designed the MOSOA-DLVD methodology for accomplishing security in the cloud platform. The aim of the MOSOA-DLVD algorithm is to identify the presence of vulnerabilities or attacks in the cloud platform. The model has three phases of function: the MOSOA-based FS, DBN classification, and STOA-based hyperparameter selection.

3.1. Feature Selection Using MOSOA

The MOSOA technique is used to select the better feature sets. This technique is imitated for the process of FS in which seagulls function as searching agents (features) [25]. SOA is a meta-heuristic optimizer algorithm inspired by the foraging behavior of seagulls. This algorithm provides the benefits of a modest implementation and architecture. The major benefit of the SOA is that its overall construction and composition are relatively simple, while its global search and local search abilities are strong. Here, the migration method is performed to attain the optimum features out of an accessible group of features and to explore the search space. The main function of the FS method includes a reduction in classification errors and the features that are considered as input.

$$\text{Min } F_t = \delta^* \psi + (1-\delta)^* \frac{|f|}{|F|} \tag{1}$$

In this system, the aims are combined into a single objective equation like a preset weight that identifies all the objective importance. In Equation (1), δ corresponds to the

parameter inducing the classifier's output, F denotes the overall number of features in the data, ψ specifies the error rate of the classifiers, and f represents the overall feature extraction counts during the extraction feature. The FF needs to have a low value for the proper FS. Figure 1 exemplifies the workflow of the MOSOA-DLVD method.

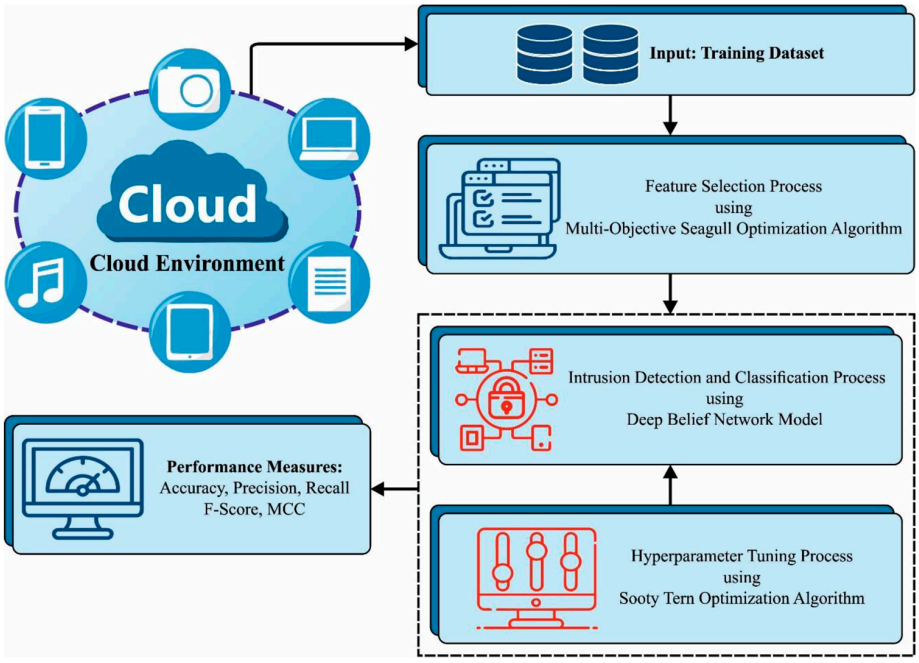

Figure 1. The overall flow of the MOSOA-DLVD algorithm.

Exploration: The exploration of the search agent includes its movement from one place to another as per the FF. The three most important conditions of the exploration method of MOSOA are given below:

(i). Collision Avoidance: It is also possible for a collision to happen, so a parameter is used to calculate the location of the searching agent while exploring the search range. The equation is given below.

$$\vec{c}_s = A^* \vec{p}_s(x) \tag{2}$$

In Equation (2), $\vec{c}_s$ shows the location of the searching agent not included in a collision, $\vec{p}_s$ represents the existing location of the searching agent, x denotes the present iteration, and parameter A shows the movement of the searching agent from the performance space. The formula for the parameter is given below.

$$A = f_l - \left(x^* \left(\frac{f,}{Itr_{\max}} \right) \right); x = 0, 1, \ldots Itr_{\max} \tag{3}$$

In Equation (3), f controls the frequency of the A parameter.

The movement to the optimum neighbor location: The searching agent that avoids the collision moves to a better neighborhood position, for which the formula is as follows.

$$\vec{m}_s = B^* \left(\vec{p}_{bs}(x) - \vec{p}_s(x) \right) \tag{4}$$

In Equation (4), $\vec{p}_s$ corresponds to the searching agent, $\vec{p}_{bs}$ stands for the place of the better search agent, and $\vec{m}_s$ represents the movement of $\vec{p}_s$ toward $\vec{p}_{bs}$. The random value B is accountable for maintaining the balance between the exploitation and exploration phases. The formula for B is given below.

$$B = 2^* A^{2*} rnd \tag{5}$$

In Equation (5), *rnd* denotes a random value within [0, 1].

(ii). Position Update: Finally, the searching agent updates the location based on the location of a better searching agent in the group. The location updating formula is as follows.

$$\vec{d}_s = \left| \vec{c}_s + \vec{m}_s \right| \tag{6}$$

In Equation (6), $\vec{d}_s$ denotes the distance between the better one in the group and the searching agent.

The MOSOA technique calculates the fitness function of the searching agent, whereas a better solution is upgraded to the archive. Once the archive is established to overflow, the grid technique is used to avoid the crowded solution in the available solutions from the archive. Next, a novel solution is upgraded to archive and later, the boundary of the searching agent is adjusted and evaluated. Finally, the FF estimates the position of the searching agent in the archive, whereas the better searching agent is upgraded with a novel location.

Exploitation: This procedure is imitated during the attacking behavior of the searching agent based on the experience and history of the exploitation. The searching agent spirally moves from the air in a 3D axis and is defined as follows.

$$x' = \alpha^* \cos(l) \tag{7}$$

$$y' = \alpha^* \sin(l) \tag{8}$$

$$z' = \alpha^* l \tag{9}$$

$$l = u^* e^{lv} \tag{10}$$

where α represents the radius of each turn in a spiral movement, l denotes the arbitrary value selected in the range of $[0, 2\pi]$, and u and v are the constants that represent the spiral motion. The last upgraded location of the search agent is shown below.

$$\vec{p}_s(x) = \left(\vec{d}_s * x' * y' * z' \right) + \vec{p}_{bs}(x) \tag{11}$$

In the MOSOA technique, the better Pareto optimum result is compared with that of the current solution. Therefore, this method selects the leader for the group to achieve it. The minimum crowded space from the archive is occupied with the roulette wheel selection process, whereas the better solution in the optimum boundary is taken into account as given below.

$$U_l = \frac{h}{N_l} \tag{12}$$

In Equation (12), N_l shows the amount of Pareto optimum solutions for the segment and h denotes the constant value higher than l.

3.2. Vulnerability Detection Utilizing the DBN Model

The DBN model has been applied in the detection and classification of the vulnerabilities. DBNs can automatically learn the hierarchical representations of the input data. For the purpose of intrusion detection, it is used for learning and extracting the important features from raw network traffic data and reducing the requirement for manual feature engineering. Primarily, this characteristic is valuable for a network intrusion model as it is complex and develops over some time. DBNs have been well-appropriated for anomaly detection activity, which is an important module of the intrusion detection process. It can model the normal behavior of a network and indicate abnormalities from learned regularities as

possible intrusions. It is supported to identify new or earlier hidden attack patterns. It is capable of taking reliance and correlation among the diverse phases of multi-stage attacks. Generally, this is significant as advanced attacks include several stages, and identifying them as a whole could be more efficient than detecting different types of anomalies.

DBN is considered a fusion of unsupervised network models like RBM that act as a hidden layer (HL) of each subnet and a visible layer (VL) of the second layer [26]. The DBN model comprises multiple VLs, HLs, and an LR for classification in the final layer. Initially, the feature vector is mapped, after which, each layer of the RBM is trained using an unsupervised method for maintaining the feature data. Next, a fine adjustment is made. In the RBM technique, the v_i in the VL and HL are characterized as h_i. w_{ij} represents the weights between v_i and h_j, while the latter denotes the guided values. The VL and HL nodes have biases and are denoted by the c and b vectors. The b_i, c_i, and w_{ij} values of the RBM form the parameter θ in the DBN and appear in the model with a probability of the energy function and the HL. Figure 2 represents the framework of the DBN.

$$E(\theta, v, h) = -\sum_{i=1}^{m} v_i c_i - \sum_{j=1}^{n} h_j b_j$$

$$-\sum_{i=1}^{m}\sum_{j=1}^{n} v_i h_i w_{ij} \tag{13}$$

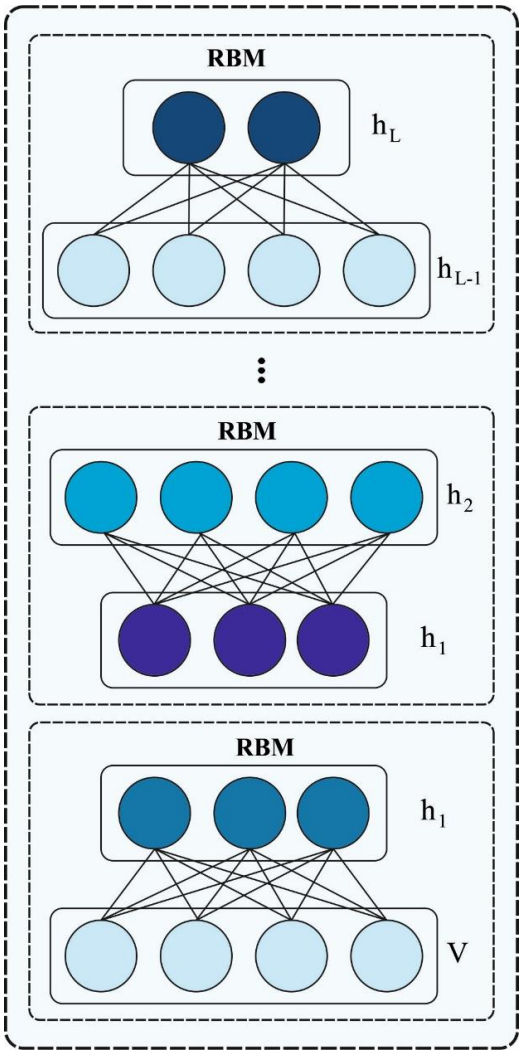

Figure 2. DBN structure.

Subsequently, there is no interlayer linked from the DBN model, whereas the probability distribution of the VL and HL is computed as given below.

$$\mathrm{P}(\mathrm{v_i} = 1|\mathrm{h}) = 1/1 + \mathrm{e}^{-\mathrm{b_i}\Sigma\mathrm{h_i}\mathrm{w_{ij}}} \tag{14}$$

$$P(h_i = 1|v) = 1/1 + e^{-c_i\Sigma v_i w_{ij}} \tag{15}$$

The reconstructed data are returned and defined with the $p(vh)$ computation after the weight calculation is completed. The output σ takes place once the data are transferred back to the HL. Now, the logistic function σ can be described as follows.

$$\sigma(x) = \left(1 + e^{-x}\right)^{-1} \tag{16}$$

Similarly, if $v_i = 1$, the conditional probability of v_i can be computed as follows:

$$P(v_i = 1|v) = \sigma\left(a_i + \sum_{i=1} W_{ij}h_j\right) \tag{17}$$

3.3. Hyperparameter Tuning Using the STOA

Eventually, the STOA is utilized for the optimum hyperparameter selection of the DBN approach. The STOA is a new optimization technique derived from the natural foraging behavior of seabirds [27]. The sooty tern is an omnivorous bird that preys on fish, earthworms, and other insects. The technique has high precision and a strong global search ability. The STOA can be a population-based technique separated into local and global search phases. The global search phase mainly comprises collision avoidance, position update, and convergence to the optimum solution.

(1) The mathematical equation is used for collision avoidance is as follows.

$$B = \gamma \times P(k) \tag{18}$$

$$\gamma = \alpha - (k \times (\alpha - \mathrm{Max}_{iieraiion}))k = 0, 1, 2, \cdots \mathrm{Max}_{iteration}, \tag{19}$$

where B refers to the safer location to make sure that no collision occurs between the black terns; γ denotes the collision avoidance aspect; and $P(k)$ shows the existing location of the black tern. k represents the number of iterations; and the α value is 2.

(2) Convergence to the optimum solution is formulated as follows.

$$\begin{cases} M = \beta \times (P_b(k) - P(k)) \\ \beta = 0.5 \times r \end{cases} \tag{20}$$

In Equation (20), $p(k)$ shows the existing optimum tern, M denotes the optimum location of the sooty tern colony; β refers to the arbitrary regulator; and r is an arbitrary integer in the range of $[0,\ 1]$.

(3) To update the position, the following equation is used.

$$D = B + M \tag{21}$$

In Equation (21), D denotes the existing and optimum locations of a sooty tern.

During the local exploration stage, the bird uses its wings to gain height and also changes its angle and speed of attack during the migration process. The hovering behavior at the time of attacking prey is described as follows.

$$\begin{cases} x' = R \times \cos(\theta) \\ y' = R \times \cos(\theta) \\ z' = R \times \theta \\ r = u \times e^{kv} \end{cases} \tag{22}$$

In Equation (11), θ represents the angle of attack in the range of $[0, 2]$, R denotes the spiral radius, and u and v show the spiral constant and are fixed as 1. The equation to update the location of the sooty tern is as follows.

$$P(k) = \left(D \times \left(x' \times y' \times z'\right)\right) \times P_b(k) \tag{23}$$

FF is a key feature of the STOA system. The encoder performance is used to develop the optimum candidate outcome. Presently, accuracy is the main condition deployed to develop the FF.

$$Fitness = \max\left(\frac{TP}{TP + FP}\right) \tag{24}$$

where TP and FP stand for true and false positive values, respectively.

4. Results and Discussion

The MOSOA-DLVD methodology was experimentally validated using the NSL-KDD database [28]. The dataset has a total of 125,973 samples under five classes, as shown in Table 1.

Table 1. Description of the dataset.

Class	No. of Samples
Dos	45,927
R2l	995
Probe	11,656
U2r	52
Normal	67,343
Total no. of Samples	125,973

In Figure 3, the confusion matrices generated using the MOSOA-DLVD system are shown. The outcomes indicate that the MOSOA-DLVD algorithm accurately recognized all five classes.

In Table 2 and Figure 4, the overall detection results of the MOSOA-DLVD method at 80:20 of the TRS/TSS are given. The achieved outcomes show that the MOSOA-DLVD system proficiently recognized all five class labels. At 80% of the TRS, the MOSOA-DLVD algorithm achieved an average $accu_y$ of 99.23%, $prec_n$ of 74.15%, $reca_l$ of 73.44%, F_{score} of 73.78%, and an MCC of 73.14%. Next, with 20% of the TSS, the MOSOA-DLVD system obtained an average $accu_y$ of 99.28%, $prec_n$ of 74.05%, $reca_l$ of 73%, F_{score} of 73.50%, and an MCC of 72.90%.

The overall detection outcomes of the MOSOA-DLVD algorithm at 70:30 of TRS/TSS are portrayed in Table 3 and Figure 5. The outcomes illustrate that the MOSOA-DLVD method efficiently recognized all five classes. For 70% of the TRS, the MOSOA-DLVD methodology attained an average $accu_y$ of 99.34%, $prec_n$ of 74.37%, $reca_l$ of 74.13%, F_{score} of 74.24%, and an MCC of 73.76%. With 30% of the TSS, the MOSOA-DLVD system attained an average $accu_y$ of 99.31%, $prec_n$ of 73.21%, $reca_l$ of 73.28%, F_{score} of 73.24%, and an MCC of 72.73%.

Figure 6 represents the training accuracy TR_accu_y and VL_accu_y values attained with the MOSOA-DLVD algorithm. TL_accu_y is determined by validating the MOSOA-DLVD methodology using the TR database, whereas VL_accu_y is measured as the effectiveness of

the model upon a distinct TS dataset. The results show that the TR_accu_y and VL_accu_y values increase with an increase in the number of epochs. Accordingly, the effectiveness of the MOSOA-DLVD algorithm is enriched with the TS and TR datasets.

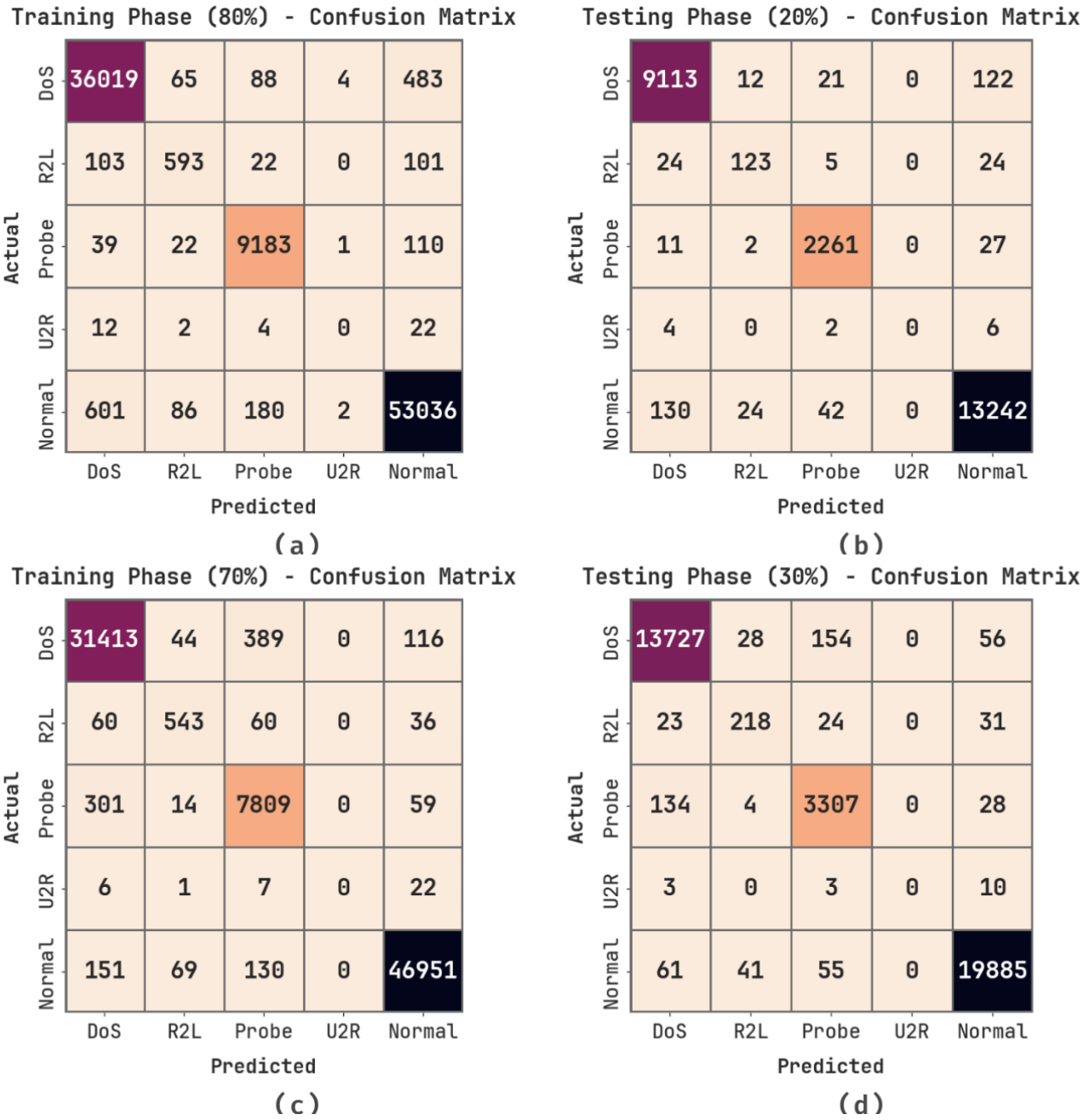

Figure 3. (**a**,**b**) Confusion matrices at 80:20 of TRS/TSS and (**c**,**d**) 70:30 of TRS/TSS.

In Figure 7, the TR_loss and VR_loss curves of the MOSOA-DLVD methodology are illustrated. TR_loss corresponds to the errors between the original and the predicted values in the TR data. VR_loss denotes the measurement of the MOSOA-DLVD system on specific validation data. The obtained outcomes confirm that both TR_loss and VR_loss values are reduced with an increasing number of epochs. This outcome describes the enriched effectiveness of the MOSOA-DLVD approach as well as its ability to achieve accurate classification. The minimal TR_loss and VR_loss values reveal the superior performance of the MOSOA-DLVD algorithm on correlation and capturing patterns.

Table 2. Detection outcomes of the MOSOA-DLVD algorithm on 80% of TRS/20% of TSS.

Labels	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TSR (80%)					
DoS	98.62	97.95	98.25	98.10	97.01
R2L	99.60	77.21	72.41	74.73	74.57
Probe	99.54	96.90	98.16	97.53	97.27
U2R	99.95	00.00	00.00	00.00	-0.02
Normal	98.43	98.67	98.39	98.53	96.84
Average	99.23	74.15	73.44	73.78	73.14
TSS (20%)					
DoS	98.71	98.18	98.33	98.25	97.24
R2L	99.64	76.40	69.89	73.00	72.89
Probe	99.56	97.00	98.26	97.63	97.39
U2R	99.95	00.00	00.00	00.00	00.00
Normal	98.51	98.67	98.54	98.60	97.01
Average	99.28	74.05	73.00	73.50	72.90

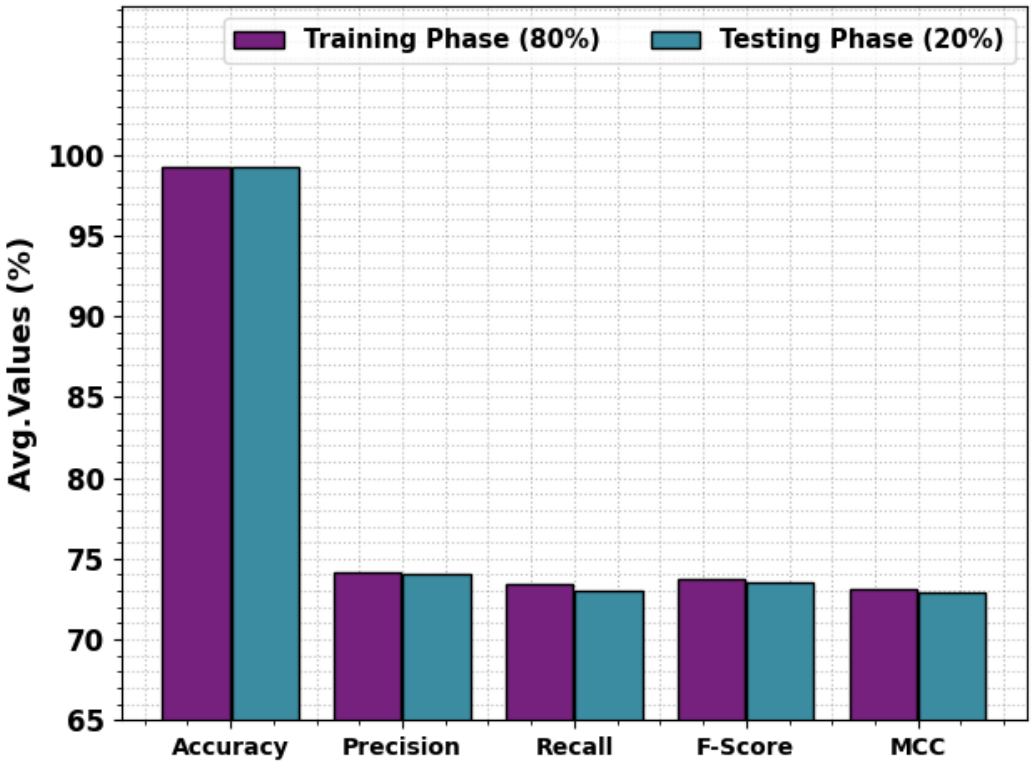

Figure 4. Average analysis outcomes of the MOSOA-DLVD model with 80% of TRS/20% of TSS.

The comprehensive precision–recall examination outcomes produced using the MOSOA-DLVD approach with the test dataset are shown in Figure 8. The MOSOA-DLVD algorithm was found to achieve increased PR values. In addition, it is obvious that the MOSOA-DLVD algorithm attains superior precision–recall values for all five classes.

Table 3. Detection outcomes of the MOSOA-DLVD algorithm on 70% of TRS/30% of TSS.

Labels	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TRS (70%)					
DoS	98.79	98.38	98.28	98.33	97.38
R2L	99.68	80.92	77.68	79.27	79.12
Probe	98.91	93.02	95.43	94.21	93.62
U2R	99.96	00.00	00.00	00.00	00.00
Normal	99.34	99.51	99.26	99.38	98.67
Average	99.34	74.37	74.13	74.24	73.76
TSS (30%)					
DoS	98.79	98.42	98.30	98.36	97.39
R2L	99.60	74.91	73.65	74.28	74.08
Probe	98.94	93.34	95.22	94.27	93.69
U2R	99.96	00.00	00.00	00.00	00.00
Normal	99.25	99.38	99.22	99.30	98.50
Average	99.31	73.21	73.28	73.24	72.73

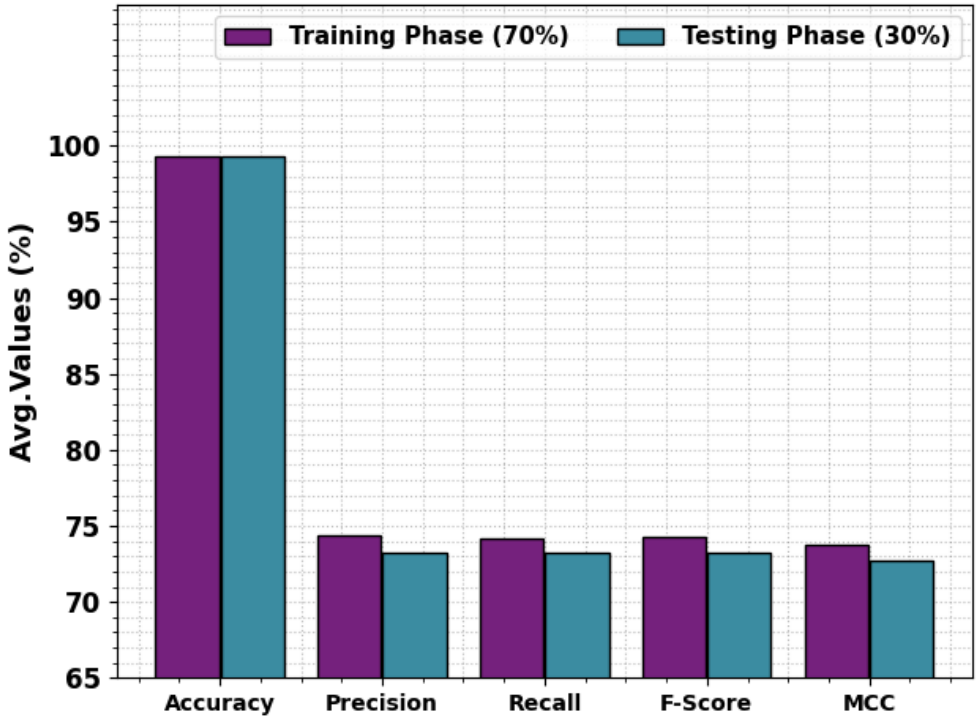

Figure 5. Average analysis outcomes of the MOSOA-DLVD method with 70% of TRS/30% of TSS.

In Figure 9, the ROC outcomes of the MOSOA-DLVD methodology are exhibited. The outcomes show that the MOSOA-DLVD system produced enhanced ROC values. Furthermore, it is apparent that the MOSOA-DLVD algorithm extends greater ROC values with all five classes. The ROC curves produced using the MOSOA-DLVD system exhibit its capability to differentiate the classes. This figure indicates the valued perceptions of the trade-off between the FPR and TPR rates over individual categorization thresholds as well as the changing number of epochs. This figure displays the predicted $accu_y$ efficiency of the MOSOA-DLVD model for the categorization of diverse classes.

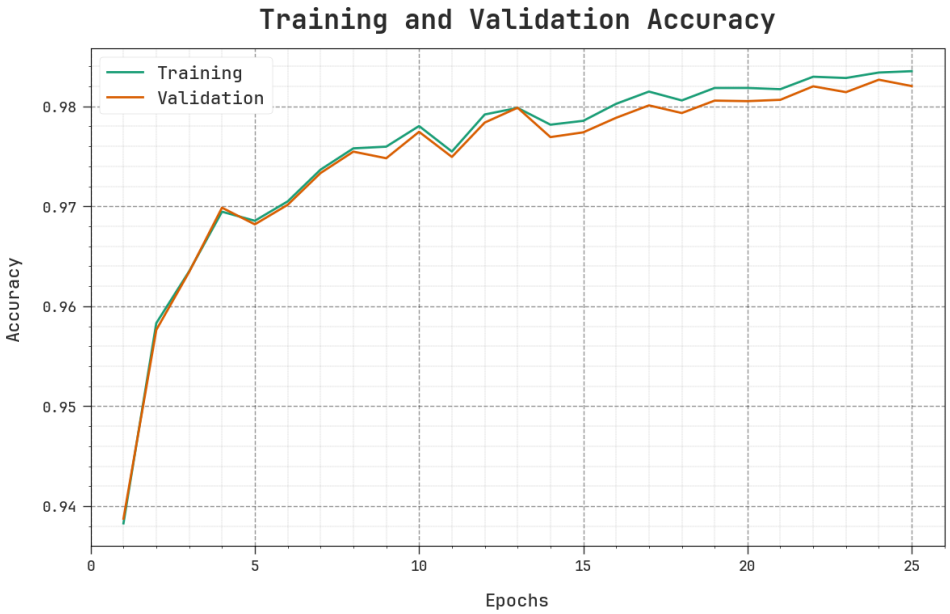

Figure 6. $Accu_y$ curve of the MOSOA-DLVD algorithm.

Figure 7. Loss curve of the MOSOA-DLVD system.

A comparison analysis was conducted between the MOSOA-DLVD methodology and other existing systems such as the leader-based K-means clustering (LKM) with the OFLS [22], K-means with OFLS [23], MLP [23], and PCA with NN [24] methods, and the results are portrayed in Table 4 and Figure 10 [22–24]. The achieved outcomes show that the LKM-OFLS and PCA-NN models obtained poorer results than the rest of the models. Along with that, the K-means-OFLS and MLP techniques accomplished a closer performance. But the MOSOA-DLVD technique reported the maximum performance with $accu_y$, $prec_n$, $reca_l$, and F_{score} values being 99.34%, 74.37%, 74.13%, and 74.24%, respectively. This phenomenal performance establishes the enhanced outcomes of the MOSOA-DLVD methodology.

Table 4. Comparative analysis of the outcomes of the MOSOA-DLVD algorithm and other algorithms [26–28].

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
MOSOA-DLVD	99.34	74.37	74.13	74.24
LKM-OFLS [22]	89.34	64.64	54.68	58.26
K-Means-OFLS [23]	91.43	65.74	55.51	58.33
MLP algorithm [23]	91.46	66.61	56.76	54.99
PCA-NN [24]	90.08	64.56	56.06	57.54

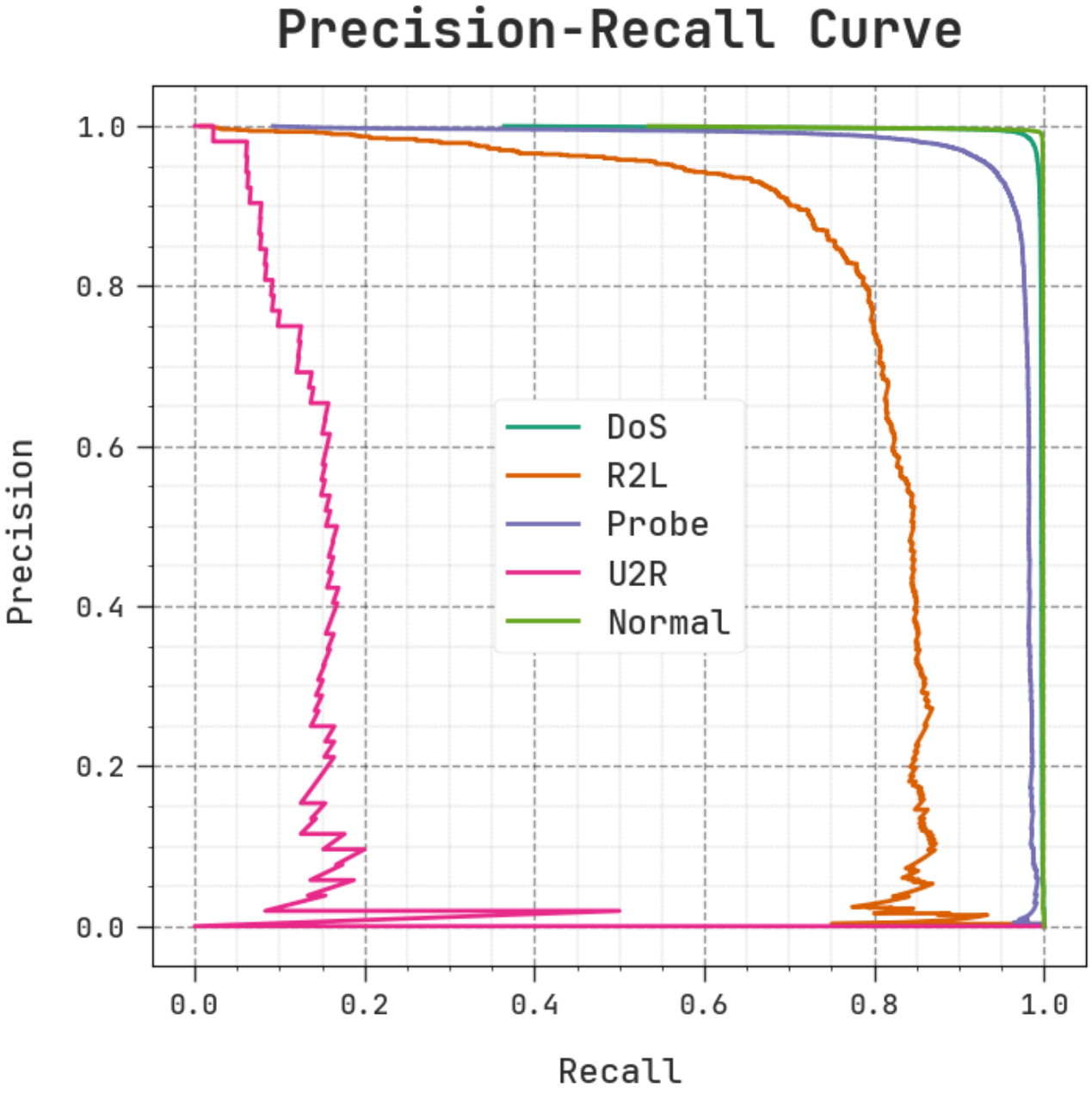

Figure 8. PR analysis of the MOSOA-DLVD model.

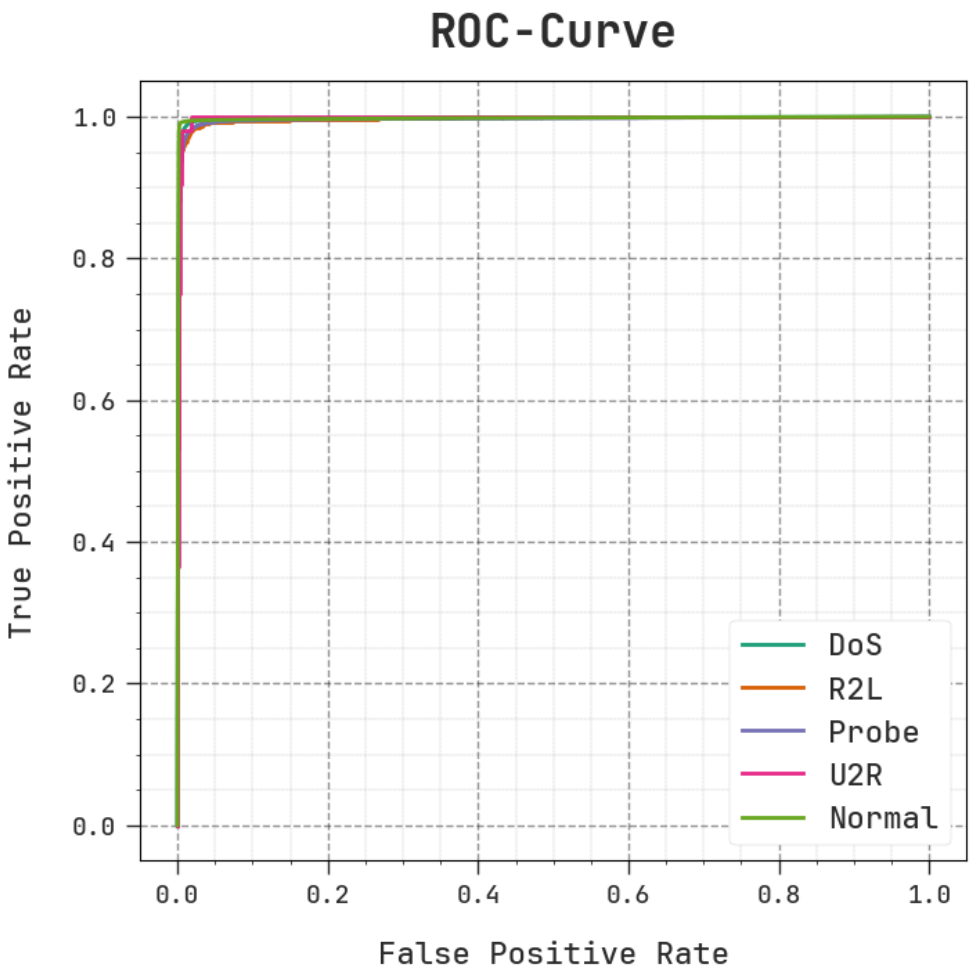

Figure 9. ROC curve of the MOSOA-DLVD algorithm.

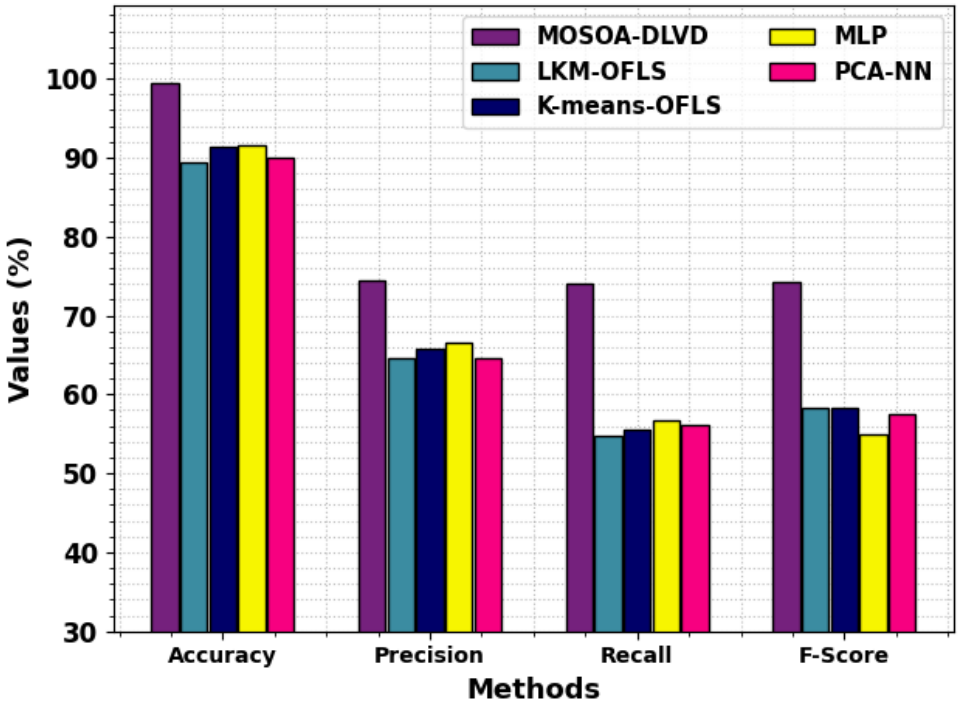

Figure 10. Comparative analysis of the outcomes of the MOSOA-DLVD algorithm and other systems.

In summary, the MOSOA-DLVD method exhibited superior performance with a maximum accu_y of 99.34%. The high effectiveness of the MOSOA-DLVD system is due to the incorporation of the MOSOA-assisted FS algorithm and STOA-based hyperparameter tuning. The MOSOA algorithm selects the relevant and beneficial features at accessible feature sets. With the elimination of unrelated features, the proposed model can be considered a crucial finding in terms of aspects contributing to the classification method. This model can improve the accuracy of classification. Alternatively, the STOA optimizer prefers the optimal values for the hyperparameters of the specified DBN system. If the hyperparameters cannot be learned during the training period, then they should be set before the training. It has an important effect on the model's performance as well, and the selection of the optimum values could result in higher accuracy. By integrating the MOSOA-based FS algorithm and STOA-based hyperparameter tuning, the MOSOA-DLVD system achieved the best solution by emphasizing major related features as well as selecting the optimal sets for the method. These attained outcomes confirm the better performance of the MOSOA-DLVD methodology over other systems.

5. Conclusions

In the current study, the MOSOA-DLVD technique was presented to accomplish security in the cloud platform. The primary aim of the MOSOA-DLVD methodology is to identify the presence of vulnerabilities or attacks in the cloud platform. In the developed MOSOA-DLVD method, three phases of processes are executed such as the DBN classification, STOA-based hyperparameter selection, and the MOSOA-based FS. To enhance the detection results of the DBN algorithm, the STOA was used for hyperparameter tuning. The performance of the MOSOA-DLVD method was examined using the benchmark NSL-KDD dataset. A wide range of simulations was conducted, and the outcomes established the improved intrusion detection outcomes of the MOSOA-DLVD system over existing methodologies with a higher accuracy of 99.34%. In the future, the MOSOA-DLVD method can be extended to the big data environment. Furthermore, the class imbalance data handling issue needs to be resolved in order to achieve improved classification results. Future works can explore further techniques for intrusion detection that can operate on encrypted or privacy-preserving data. This might ensure the confidentiality of sensitive information, while detecting intrusions in an effective manner and remains important, especially in multi-tenant cloud environments.

Author Contributions: Conceptualization, M.A. and M.A.A.; methodology, M.A.A., H.M., A.S. and S.S.A.; software, A.A.; validation, M.A.A., A.S. and A.A.; investigation, M.A.; data curation, S.S.A.; writing—original draft, M.A., M.A.A., H.M., A.S. and S.S.A.; writing—review and editing, M.A.A., A.S., A.A. and M.A.; visualization, A.A.; project administration, M.A.A.; funding acquisition, M.A.A. and H.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through a large group Research Project under grant number (RGP2/95/44), Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R330), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia, and Research Supporting Project number (RSP2023R459), King Saud University, Riyadh, Saudi Arabia. We would like to thank the SAUDI ARAMCO Cybersecurity Chair for funding this project. This study is partially funded by the Future University in Egypt (FUE).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare without conflict of interest. This manuscript was written with the contributions of all authors. All authors have approved the final version of this manuscript.

References

1. Vinolia, A.; Kanya, N.; Rajavarman, V.N. Machine Learning and Deep Learning based Intrusion Detection in Cloud Environment: A Review. In Proceedings of the 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 23–25 January 2023; pp. 952–960.
2. Tariq, M.; Suaib, M. A Review on Intrusion Detection in Cloud Computing. *Int. J. Eng. Manag. Res.* **2023**, *13*, 207–215.
3. Chang, V.; Golightly, L.; Modesti, P.; Xu, Q.A.; Doan, L.M.T.; Hall, K.; Boddu, S.; Kobusińska, A. A survey on intrusion detection systems for fog and cloud computing. *Future Internet* **2022**, *14*, 89. [CrossRef]
4. Patel, S.K. Improving intrusion detection in cloud-based healthcare using neural network. *Biomed. Signal Process. Control.* **2023**, *83*, 104680. [CrossRef]
5. Liu, Z.; Xu, B.; Cheng, B.; Hu, X.; Darbandi, M. Intrusion detection systems in cloud computing: A comprehensive and deep literature review. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6646. [CrossRef]
6. Thangasamy, A.; Sundan, B.; Govindaraj, L. Dynamic phad/ahad analysis for network intrusion detection and prevention system for cloud environment. In Proceedings of the 2021 4th International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 16–17 December 2021; pp. 273–279.
7. Lin, H.; Xue, Q.; Feng, J.; Bai, D. Internet of Things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine. *Digit. Commun. Netw.* **2023**, *9*, 111–124. [CrossRef]
8. Balamurugan, E.; Mehbodniya, A.; Kariri, E.; Yadav, K.; Kumar, A.; Haq, M.A. Network optimization using defender system in cloud computing security-based intrusion detection system with game theory deep neural network (IDSGT-DNN). *Pattern Recognit. Lett.* **2022**, *156*, 142–151. [CrossRef]
9. Mohamed, D.; Ismael, O. Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing. *J. Cloud Comput.* **2023**, *12*, 41. [CrossRef]
10. Snehi, J.; Snehi, M.; Bhandari, A.; Baggan, V.; Ahuja, R. Introspecting Intrusion Detection Systems in Dealing with Security Concerns in Cloud Environment. In Proceedings of the 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 10–11 December 2021; pp. 345–349.
11. Kavitha, C.; Gadekallu, T.R.; Nimala, K.; Kavin, B.P.; Lai, W.C. Filter-Based Ensemble Feature Selection and Deep Learning Model for Intrusion Detection in Cloud Computing. *Electronics* **2023**, *12*, 556. [CrossRef]
12. Jaber, A.N.; Rehman, S.U. FCM–SVM-based intrusion detection system for the cloud computing environment. *Clust. Comput.* **2020**, *23*, 3221–3231. [CrossRef]
13. Maheswari, K.G.; Siva, C.; Nalinipriya, G. Optimal cluster-based feature selection for intrusion detection systems in web and cloud computing environments using hybrid teacher learning optimization enables deep recurrent neural networks. *Comput. Commun.* **2023**, *202*, 145–153. [CrossRef]
14. Devi, K.; Muthusenthil, B. Intrusion detection framework for securing privacy attacks in a cloud computing environment using DCCGAN-RFOA. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4561. [CrossRef]
15. Kanimozhi, P.; Aruldoss Albert Victoire, T. Oppositional tunicate fuzzy C-means algorithm and logistic regression for intrusion detection on cloud. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6624. [CrossRef]
16. Samriya, J.K.; Kumar, N. A novel intrusion detection system using hybrid clustering-optimization approach in cloud computing. *Mater. Today Proc.* **2020**, *2*, 23–54. [CrossRef]
17. Sathiyadhas, S.S.; Soosai Antony, M.C.V. A network intrusion detection system in a cloud computing environment using dragonfly improved invasive weed optimization integrated Shepard convolutional neural network. *Int. J. Adapt. Control. Signal Process.* **2022**, *36*, 1060–1076. [CrossRef]
18. Karuppusamy, L.; Ravi, J.; Dabbu, M.; Lakshmanan, S. Chronological salp swarm algorithm-based deep belief network for intrusion detection in the cloud using fuzzy entropy. *Int. J. Numer. Model. Electron. Netw. Devices Fields* **2022**, *35*, e2948. [CrossRef]
19. Mani, S.; Sundan, B.; Thangasamy, A.; Govindaraj, L. A new intrusion detection and prevention system using a hybrid deep neural network in the cloud environment. In *Computer Networks, Big Data and IoT: Proceedings of ICCBI 2021*; Springer Nature: Singapore, 2022; pp. 981–994.
20. Varun, P.; Ashokkumar, K. Intrusion detection system in cloud security using deep convolutional network. *Appl. Math. Inf. Sci.* **2022**, *16*, 581–588.
21. Aldallal, A.; Alisa, F. Effective intrusion detection system to secure data in the cloud using machine learning. *Symmetry* **2021**, *13*, 2306. [CrossRef]
22. Alohali, M.A.; Elsadig, M.; Al-Wesabi, F.N.; Al Duhayyim, M.; Mustafa Hilal, A.; Motwakel, A. Enhanced Chimp Optimization-Based Feature Selection with Fuzzy Logic-Based Intrusion Detection System in Cloud Environment. *Appl. Sci.* **2023**, *13*, 2580. [CrossRef]
23. Shyla, S.I.; Sujatha, S.S. Cloud security: LKM and optimal fuzzy system for intrusion detection in cloud environment. *J. Intell. Syst.* **2020**, *29*, 1626–1642. [CrossRef]
24. Mahmood, Z.; Agrawal, C.; Hasan, S.S.; Zenab, S. Intrusion detection in a cloud computing environment using neural network. *Int. J. Res. Comput. Eng. Electron.* **2012**, *1*, 1–4.
25. Jagannathan, P.; Gurumoorthy, S.; Stateczny, A.; Divakarachar, P.B.; Sengupta, J. Collision-aware routing using multi-objective seagull optimization algorithm for WSN-based IoT. *Sensors* **2021**, *21*, 8496. [CrossRef] [PubMed]

26. Al-Khazraji, H.; Nasser, A.R.; Hasan, A.M.; Al Mhdawi, A.K.; Al-Raweshidy, H.; Humaidi, A.J. Aircraft engines remain useful for life prediction based on a hybrid model of autoencoder and deep belief network. *IEEE Access* **2022**, *10*, 82156–82163. [CrossRef]
27. Shen, S.; Du, Y.; Xu, Z.; Qin, X.; Chen, J. Temperature Prediction Based on STOA-SVR Rolling Adaptive Optimization Model. *Sustainability* **2023**, *15*, 11068. [CrossRef]
28. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 13 July 2023).