Computers, Materials & Continua
DOI: 10.32604/cmc.2023.033677
Article

Chaotic Metaheuristics with Multi-Spiking Neural Network Based Cloud Intrusion Detection

Mohammad Yamin[1,*], Saleh Bajaba[2] and Zenah Mahmoud AlKubaisy[1]

[1]Department of Management Information Systems, Faculty of Economics and Administration, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
[2]Department of Business Administration, Faculty of Economics and Administration, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
*Corresponding Author: Mohammad Yamin. Email: myamin@kau.edu.sa
Received: 24 June 2022; Accepted: 29 September 2022

Abstract: Cloud Computing (CC) provides data storage options as well as computing services to its users through the Internet. On the other hand, cloud users are concerned about security and privacy issues due to the increased number of cyberattacks. Data protection has become an important issue since the users' information gets exposed to third parties. Computer networks are exposed to different types of attacks which have extensively grown in addition to the novel intrusion methods and hacking tools. Intrusion Detection Systems (IDSs) can be used in a network to manage suspicious activities. These IDSs monitor the activities of the CC environment and decide whether an activity is legitimate (normal) or malicious (intrusive) based on the established system's confidentiality, availability and integrity of the data sources. In the current study, a Chaotic Metaheuristics with Optimal Multi-Spiking Neural Network-based Intrusion Detection (CMOMSNN-ID) model is proposed to secure the cloud environment. The presented CMOMSNN-ID model involves the Chaotic Artificial Bee Colony Optimization-based Feature Selection (CABC-FS) technique to reduce the curse of dimensionality. In addition, the Multi-Spiking Neural Network (MSNN) classifier is also used based on the simulation of brain functioning. It is applied to resolve pattern classification problems. In order to fine-tune the parameters relevant to the MSNN model, the Whale Optimization Algorithm (WOA) is employed to boost the classification results. To demonstrate the superiority of the proposed CMOMSNN-ID model, a useful set of simulations was performed. The simulation outcomes inferred that the proposed CMOMSNN-ID model accomplished a superior performance over other models with a maximum accuracy of 99.20%.

Keywords: Cloud computing; security; intrusion detection; feature selection; multi-spiking neural network

1 Introduction

Cloud Computing (CC) has become an essential component in today's world, especially after the outbreak of COVID-19 [1]. Most companies started moving towards cloud-based operations to sustain their commercial activities. The CC platform provides many services to its end-users free of cost, and it includes data storage and access from any place across the globe [2]. Both the concepts of CC and the distributed mechanisms are similar in nature. In distributed systems, the data gets dispersed in diverse locations, whereas it can be retrieved anywhere across the globe. Conversely, the entire data is stored in the cloud computing platform and can be accessed by any user from different places [3]. Network traffic analysis in cloud networks is one of the most significant tasks in cloud management. It is important to guarantee service quality, authenticate new services and applications, form precise network methods, and identify the variances in the cloud. The flow of the cloud computing network exhibits the behavioural patterns of the users in terms of service function or usage [4]. Traffic analysis and the detection of important application flows are important tools in using the model services and framing the paradigms for the identification of normal system functions.

The CC network experiences a number of security challenges. Though most security issues are fixed up to a certain extent [5], some security issues still exist. So, it is important to recognize these security issues before the transformation of institutions from conventional systems to cloud-based systems. The CC network needs two types of security models such as the network security model and the data security model, to protect itself from cyber-attacks [6]. Several menaces attack the cloud data centres and harm the cloud by snatching or stealing the data through cyberattacks. Various authors have proved that if an Intrusion Detection System (IDS) is connected with all sorts of cloud gadgets, the gadgets can remain secure with a few to less number of chances for cyberattacks. Cyber-attackers devise multiple types of attacks along with encryption and decryption methods to steal the data from the cloud servers [7]. These methods can abolish the data in a server, and every data may get corrupted. In such scenarios, a safe digital infrastructure can protect the cloud server from cyberattacks. The security aspects of the cloud server must be considered whenever saving huge volumes of the data in it. The cloud server can remain safe only when it contains a set of techniques, implementations and methods. As mentioned earlier, various authors have devised diverse data protection methods, processes and policies to ensure network cloud security [8]. However, IDS is the only optimal solution that can protect the networks. It is a system that controls uncertain actions and policy defilements in a cloud environment. It can identify the malware in the cloud and alert the cloud administrator whenever an invader tries to attack the cloud data centre [9]. The most important benefit of installing the IDSs in a cloud network is that the arriving actions in the network can be monitored, and such actions can be categorized as either invalid or valid. Certain IDSs are capable of providing instant replies to the administrator when malware is detected [10]. Various IDSs are accessible in antivirus products which identify the intrusions in the cloud servers.

Shyla et al. [11] proposed a new IDS by integrating the Leader-related K-means clustering (LKM) method and the Optimum Fuzzy Logic (FL) method. Initially, the input datasets were assembled into clusters with the help of the LKM method. Afterwards, the cluster data was provided to the Fuzzy Logic System (FLS). In this study, both abnormal and normal data were analysed by the FLS method,

whereas it was trained with the help of the Grey Wolf Optimization (GWO) method to maximize the classification results. In literature [12], a Host-related IDS (H-IDS) was proposed to protect the Virtual Machines (VMs) in the cloud atmosphere. To end the security issues, a set of significant features was chosen initially for every class with the help of LR. Then, those values were enhanced with the help of the regularization method. Afterwards, several assaults were categorized by employing a blend of three distinct classifiers such as the Linear Discriminant Analysis (LDA), Neural Network (NN) and the Decision Tree (DT) method with a bagging technique for every class.

Chiba et al. [13] suggested the optimization of a famous soft computing tool, namely, Back Propagation Neural Network (BPNN), that is extensively utilized for IDS operations. The optimization was performed using the Improved Genetic Algorithm (IGA). The Genetic Algorithm (GA) method can be enhanced via the optimization strategies such as Fitness Value Hashing and Parallel Processing. These processes minimize the duration of the performance, its convergence period and the processing power. Further, the momentum term and the learning rate were amongst the most-related variables that affect the classification performance of the BPNN method. Further, the IGA approach was leveraged to identify the optimum or near optimum values for two such variables to ensure a low false alarm rate, a high detection rate and maximum accuracy. Jaber et al. [14] suggested a new IDS combining the Fuzzy c-means Clustering (FCM) method and the Support Vector Machine (SVM) approach to improve the accuracy of the detection mechanism in the cloud computing environment. The presented system was implemented and compared against the existing systems. In the study conducted earlier [15], a next-gen cloud IDS was proposed at the hypervisor layer and was evaluated for the detection of the depraved actions in the CC environment. The cloud IDS employed a hybrid method combining the FCM clustering technique and the Back Propagation ANN technique to enhance the detection accuracy of the cloud IDS. The suggested system outcomes were compared and contrasted against the classic FCM method and the K-means algorithm.

The current study proposes a Chaotic Metaheuristics with Optimal Multi-Spiking Neural Network-based Intrusion Detection (CMOMSNN-ID) method to secure the cloud environment. The presented CMOMSNN-ID model involves the Chaotic Artificial Bee Colony Optimization-based Feature Selection (CABC-FS) technique to reduce the curse of dimensionality. In addition, the Multi-Spiking Neural Network (MSNN) classifier is also used based on the simulation of brain functioning. It is applied to resolve pattern classification problems. In order to fine-tune the parameters relevant to the MSNN model, the Whale Optimization Algorithm (WOA) is employed to boost the classification results. In order to demonstrate the superiority of the CMOMSNN-ID model, a useful set of simulations was conducted.

2 Design of CMOMSNN-ID Model

In this study, a new CMOMSNN-ID technique has been proposed to detect and recognize intrusions in a secure cloud environment. The presented CMOMSNN-ID model involves CABC-FS technique to reduce the curse of dimensionality. Following, the WOA-MSNN classifier is used to overcome the pattern classification problems. Fig. 1 depicts the overall process of the proposed CMOMSNN-ID approach.

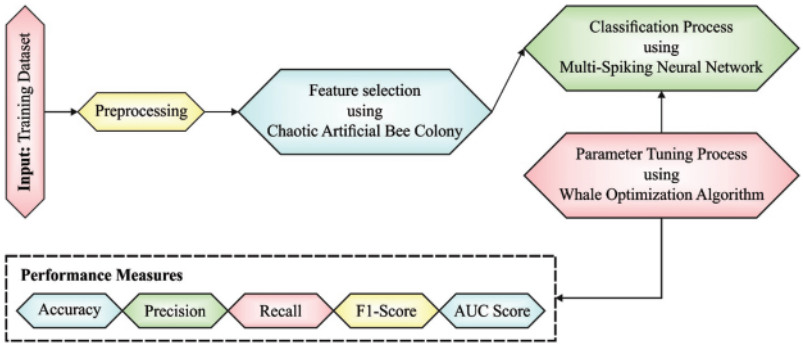

Figure 1: Overall process of the CMOMSNN-ID approach

2.1 Steps Involved in CABC-FS Model

In this study, the presented CMOMSNN-ID model involves CABC-FS technique to reduce the curse of dimensionality. The study introduces an improved ABC approach to fetch the pertinent genes from the COVID-19 transcription dataset. The ABC approach is a nature-inspired optimization technique that was developed on the basis of the foraging behaviour of the swarming honey bees [16]. Karaboga recommended this method since it exhibited excellent developments in the outcomes. The presented method achieved remarkable outcomes for a wide-range of optimization issues. In general, the foraging honey bees are of three types such as scouts, worker bees and onlooker bees. The worker bees exploit the food supplies. Then, a novel candidate solution is produced and represented by each group of honey bees. From dissimilar food sources, nectar is added to the hive. The onlooker bees wait in the hive for data to be shared by the worker bees about the food sources. Based on the data, the onlooker bees explore for food supplies by becoming scout bees. Here, the solution is abandoned by the working bees, if the food source is already exhausted. Next, the scout bees randomly search for novel food sources nearby the hive without utilizing any type of data. Once the scouts find a novel food source, it becomes worker bees. Each scout can become an adventurer and search for food without any specific direction, whereas the scouts are free to explore different kinds of food sources. As a result, the scouts might unknowingly find a rich and a completely-unknown food source too. In this situation, a new solution is proposed by the neighbouring operator to employ bees as well as the onlooker bees. In order to increase the exploitation perspective of the ABC approach, a local seeking process is employed for the solution, which is attained by the neighbouring operator with some probability. Besides the execution of the algorithm, it is also upgraded additionally through the addition of two novel components, which in turn overcomes the shortcomings of the ABC approach.

Algorithm 1: The process of the ABC algorithm

Initialize
Determine the sample and assign the sample to active bee
While (sequence = $MAX_$Sequence) do
Active bee Phase
    for $i = 1$ to SN, do
        Produce novel outcome vi for active bees and scrutinize fitness rate

(Continued)

Algorithm 1: Continued

Present greedy assortment devices amongst v_i and x_i; pick best choice
If result x_i doesn't upgrade, the non-upgraded number $t_i = t_i + 1$; else $t_i = 07$
end for
Observer bee Phase
Examine the selected probability p_i
t = 0, $I = 1$
while ($t < SN$) do
When random $< P_i$
t = $t + 1$
Produce an outcome for observer bees and analyze the fitness rate
Utilize greedy selection technique between v_i and x_i, choose the best one
When x_i result doesn't upgrade, the non-upgraded number $t_i = t_i + 1$; else $t_i = 0$
end if
$i = i + 1$
If $i = SN + 1$, $i = 1$,
end while ($t = SN$)
Detect bee Phase
$t_i >$ limit then
Change x_i with a novel random outcome
end if
Learn current optimum outcome
sequence = sequence + 1
while (sequence = $MAX_$Sequences)

To enhance the performance of the ABC algorithm, the CABC algorithm is derived with the help of a chaotic logistic map. Then, the CABC algorithm derives a fitness function to handle the trade-off amongst the chosen features and achieve classification accuracy by means of the chosen features. The fitness function can be determined as given below.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \tag{1}$$

Here, $\gamma_R(D)$ signifies the classification error rate of the given classifier. $|R|$ denotes the elected subset and $|C|$ implies the total feature count, and finally, α and β denote the constants.

2.2 Intrusion Detection Using MSNN Model

In order to identify and classify the intrusions, the MSNN model is exploited. The MSNN classification technique is utilized as a base classification method [17]. The SNN classification approach utilizes the Least-Squares approach to approximate the synaptic weight alterations that are required to induce the spikes at the chosen output spike time, tmin or cancels the undesired spikes from the learning course. The calculation of the weights should be modified either to induce or cancelling the spikes that are created on a model. This is done to exploit the connection between the input spike time and the output spike time from the neural method determined by Eq. (2), and a model of formulas is established from Eq. (3). The required weight alteration, denoted by Δw, is estimated. Fig. 2 demonstrates the framework of the MSNN technique.

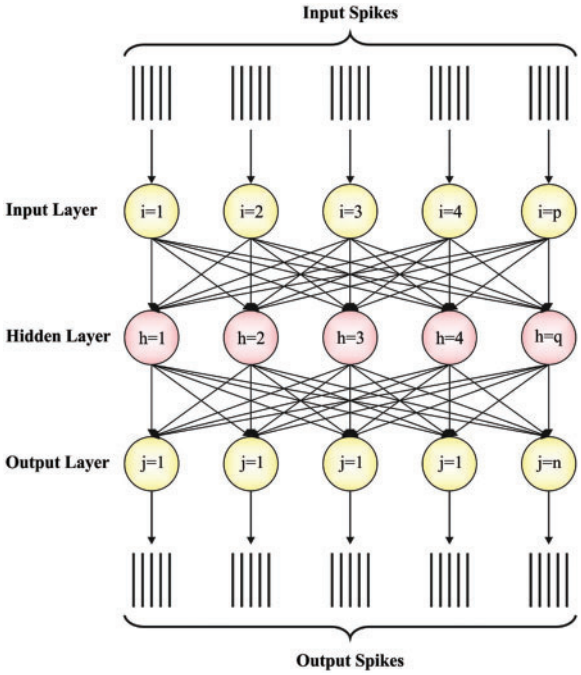

Figure 2: Structure of the MSNN technique

A spike locality model is also employed in the determination of a group of weights to adjust the provided resultant spike time t to minimize the impact of the learning interferences.

$$\frac{dV^j(t)}{dt} = \sum_i w_{ji} \sum_r k\left(t - t_i^r\right) \tag{2}$$

Here, dV^j signifies the membrane potential of the resultant neuron j, w_{ji} represents the weight of the synapses that link the input neuron i with the resultant neuron j, k denotes the membrane potential, and t_i^r implies the fire time of the pre-synaptic neuron, i.

$$\begin{aligned}
&(\Delta z_1 - 1)\,\Delta w_1 + \Delta z_1 \Delta w_2 + \ldots + \Delta z_1 \Delta w, = w_1 - \Delta z_1 \left(\textstyle\sum_{i=1:\mathrm{I}} w_i - 1\right)\\
&\Delta z_2 \Delta w_1 + (\Delta z_2 - 1)\,\Delta w_2 + \ldots + \Delta z_2 \Delta w, = w_2 - \Delta z_2 \left(\textstyle\sum_{i=1:\mathrm{I}} w_i - 1\right)\\
&\qquad\qquad \vdots \qquad\qquad \ddots \qquad\qquad \vdots\\
&\Delta z, \Delta w_1 + \Delta z, \Delta w_2 + \ldots + (\Delta z, -1)\,\Delta w, = w, -\Delta z, \ \left(\textstyle\sum_{i=1:\mathrm{I}} w_i - 1\right)
\end{aligned} \tag{3}$$

Here, $\Delta z_i = \frac{z_d}{z_i}$ (z_d and z_i denote the z-domain transforms of $exp\,(t)$ and $xp\,(t_i)$, correspondingly). Further, w_i represents the primary synaptic weight, and Δw_i implies the changed weight as evaluated earlier. Then, the Eq. (3) is formulated from the matrix procedure as given in Eq. (4).

$$\begin{bmatrix} (\Delta z_1 - 1) & \Delta z_1 & \ldots & \Delta z_1 \\ \Delta z_2 & (\Delta z_2 & -1) & \Delta z_2 \\ \vdots & \vdots & \ddots & \vdots \\ \Delta z_{\mathrm{I}} & \Delta z_{\mathrm{I}} & (\Delta z_{\mathrm{I}} & -1) \end{bmatrix} \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \\ \vdots \\ \Delta w_I \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_I \end{bmatrix} \tag{4}$$

Here, $c_i = w_i - \Delta z_i \left(\sum_{i=1:I} w_i - 1\right)$. The vector of the *unk* nowns (needed weight alteration) $\Delta w = \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \\ \vdots \\ \Delta w_1 \end{bmatrix}$ is then estimated using the Eq. (4) based on the Least-Squares approach. A complete procedure is required for the alteration of the neural method to model the formulas and the computation of Δw. The adjusted weight of the resultant neurons is governed by Eq. (5), whereas, in case of hidden neurons, it is governed by (6). "+" implies the increment in the synaptic weights, whereas ± denotes the reduction in the weights of neurons that contribute to the resultant spike.

$$w_{hi} = w_{hi} \pm \eta \Delta_{hi} \tag{5}$$

$$w_{ji} = w_{ji} \pm \eta \Delta_{ji} \tag{6}$$

Here, η denotes the rate of learning that controls the magnitude of the completely-altered weight at a provided time.

2.3 Parameter Optimization Using WOA

In order to fine tune the parameters involved in the MSNN model, the WOA approach is employed to boost the classification results. WOA is a metaheuristic technique that was developed on the basis of Humpback whales [18]. In the presented method, the optimization algorithm initiates the function by randomly generating the whale population. Then, it tries to find the optimal location of the prey. Then, the position is either added or improved through a bubble-net or an encompassing mechanism. In encompassing methodology, the Humpback whales enhance the existing position on the basis of the optimal position given below.

$$D = |C \odot X^* (t) - X (t)| \tag{7}$$

$$X (t + 1) = X^* (t) - A \odot D \tag{8}$$

In the above-mentioned equations, the distance between the location vector of the prey $X(t)^*$ and the whale $X (t)$ is represented by D, $\odot$ indicates the element-wise multiplication and t denotes the existing iteration count. A and C denote the coefficient vectors as given herewith.

$$A = 2a \odot r - a \tag{9}$$

$$C = 2r \tag{10}$$

Let r be a random vector with a length X; every index of r involves an arbitrary value in the range of 0 and 1 whereas the value of a is linearly reduced from 2 to 0. The Bubble-net mechanism is implemented in two different manners namely, shrinking-encompassing and spiral-updating positions. Initially, the values of a n in Eq. (9) and A are reduced. Next, the model is motivated by the helix-shaped movements of the humpback whales that surround the prey.

$$X (t + 1) = D' \odot e^{bl} \odot \cos\ (2\pi l) + X^* (t) \tag{11}$$

In Eq. (11), $D' = |X^* (t) - X (t)|$ represents the distance between whales and the prey, b indicates a constant value that is applied to specify the logarithmic spiral shape, and l indicates an arbitrary value that lies in the range of $\in [-1.1]$.

Whales swim nearby the prey by following a spiral-shaped path and a shrinking circle simultaneously.

$$X(t+1) = \begin{cases} X^*(t) - A \odot D & if\ p < 0.5 \\ D' \odot e^{bl} \cos(2\pi l) + X^*(t) & if\ p \geq 0.5 \end{cases} \tag{12}$$

In Eq. (12), $p \in [0, 1]$ represents a random number to describe the probability of making spiral-shaped model or shrinking-encompassing model fix the location of the whales. During the discovery stage, the Humpback whales search for their prey in a random manner. The location of the whale is fixed by stating a random searching agent instead of an optimal searching agent.

$$D = |C \odot X_{rand} - X(t)| \tag{13}$$

$$X(t+1) = |X_{rand} - A \odot D| \tag{14}$$

In this expression, X_{rand} indicates the location of the randomly-stated whale amongst the existing population. The initial process demonstrates the overall architecture of the WOA.

3 Results and Discussion

The proposed CMOMSNN-ID model was experimentally validated under two aspects, namely, binary classification and multi-classification. Table 1 provides an overview of the binary classification dataset, which has a total of 2,500 samples. It includes 500 samples under normal class and 2,000 samples under abnormal class. Fig. 3 shows the confusion matrices generated by the proposed CMOMSNN-ID model on the binary class classification dataset.

With 80% of the Training Set (TRS), the proposed CMOMSNN-ID model recognized 537 samples as normal and 1,607 samples as abnormal. In addition, on 20% of the Testing Set (TSS), the presented CMOMSNN-ID technique categorized 113 samples under normal class and 383 samples under abnormal class. Along with that, on 70% of TRS, the CMOMSNN-ID method recognized 347 samples as normal and 1,388 samples as abnormal.

Table 2 and Fig. 4 demonstrate the overall classification outcomes attained by the CMOMSNN-ID model on binary class classification. With 80% TRS, the proposed CMOMSNN-ID model achieved an average $accu_y$ of 99%, $prec_n$ of 98.58%, $reca_l$ of 98.19%, F_{score} of 98.39% and an AUC_{score} of 98.19%. Moreover, with 20% TSS, the presented CMOMSNN-ID model accomplished an average $accu_y$ of 99.20%, $prec_n$ of 98.87%, $reca_l$ of 98.87%, F_{score} of 98.87% and an AUC_{score} of 98.87%.

Table 1: Binary classification dataset details

Class	No. of samples
Normal	500
Abnormal	2000
Total number of samples	2500

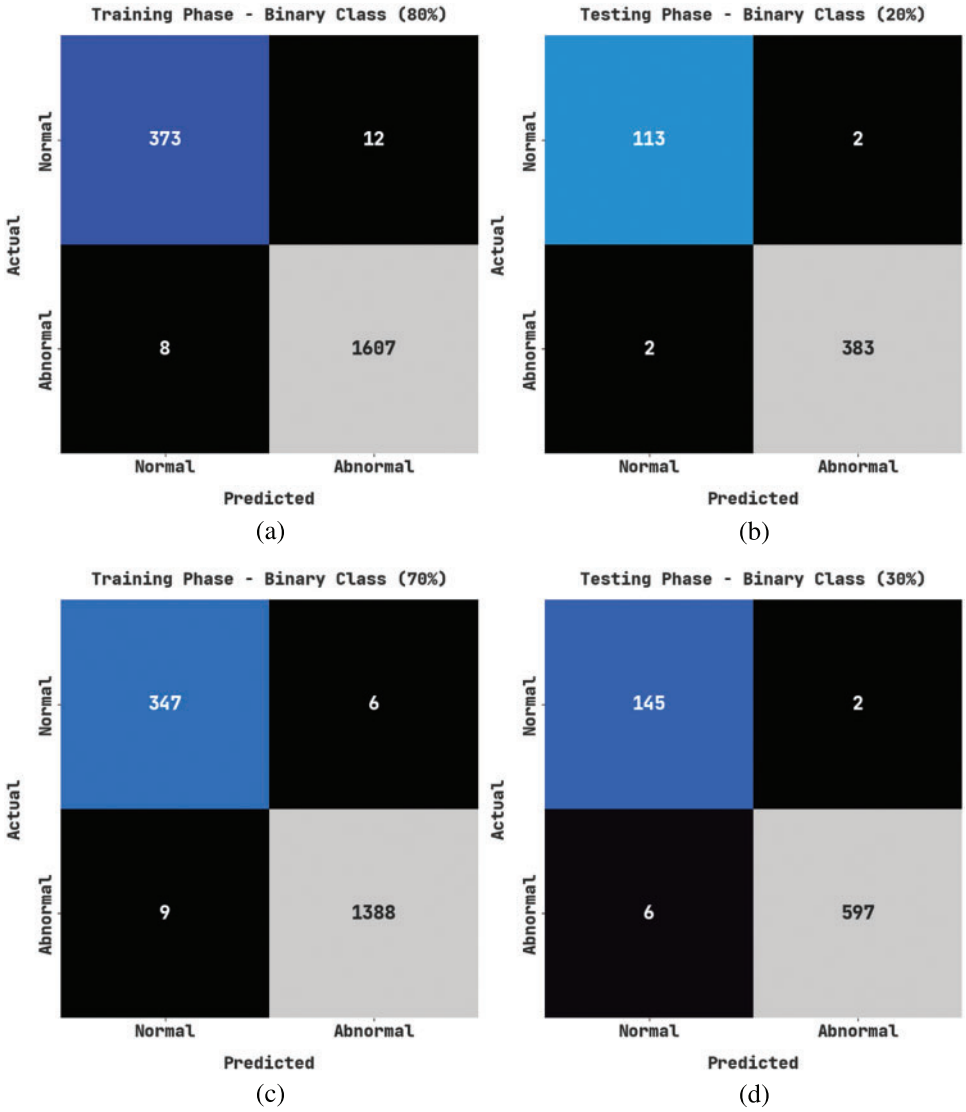

Figure 3: Confusion matrices of the CMOMSNN-ID approach under binary class classification (a) 80% of TRS, (b) 20% of TSS, (c) 70% of TRS, and (d) 30% of TSS

Table 2: Analytical results of the CMOMSNN-ID approach under different measures using binary class classification dataset

Labels	Accuracy	Precision	Recall	F-score	AUC score
Training phase (80%)					
Normal	99.00	97.90	96.88	97.39	98.19
Abnormal	99.00	99.26	99.50	99.38	98.19
Average	99.00	98.58	98.19	98.39	98.19

(Continued)

Table 2: Continued

Labels	Accuracy	Precision	Recall	F-score	AUC score
Testing phase (20%)					
Normal	99.20	98.26	98.26	98.26	98.87
Abnormal	99.20	99.48	99.48	99.48	98.87
Average	99.20	98.87	98.87	98.87	98.87
Training phase (70%)					
Normal	99.14	97.47	98.30	97.88	98.83
Abnormal	99.14	99.57	99.36	99.46	98.83
Average	99.14	98.52	98.83	98.67	98.83
Testing phase (30%)					
Normal	98.93	96.03	98.64	97.32	98.82
Abnormal	98.93	99.67	99.00	99.33	98.82
Average	98.93	97.85	98.82	98.32	98.82

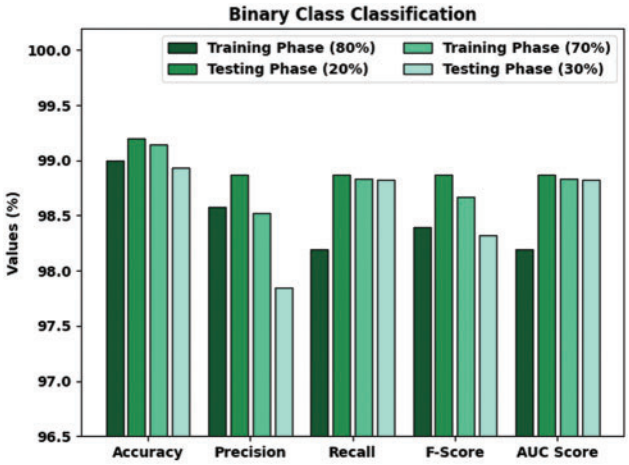

Figure 4: Analytical results of the CMOMSNN-ID approach using binary class classification dataset

Eventually, with 70% TRS, the proposed CMOMSNN-ID model granted an average $accu_y$ of 99.14%, $prec_n$ of 98.52%, $reca_l$ of 98.83%, F_{score} of 98.67% and an AUC_{score} of 98.83%. At last, with 30% TSS, the proposed CMOMSNN-ID method attained an average $accu_y$ of 98.93%, $prec_n$ of 97.85%, $reca_l$ of 98.82%, F_{score} of 98.32% and an AUC_{score} of 98.82%.

Both Training Accuracy (TA) and Validation Accuracy (VA) values, acquired by the proposed CMOMSNN-ID method on binary class classification dataset, are demonstrated in Fig. 5. The experimental outcomes denote that the proposed CMOMSNN-ID algorithm gained the maximal TA and VA values while the VA values were higher than the TA values.

Both Training Loss (TL) and Validation Loss (VL) values, achieved by the proposed CMOMSNN-ID approach on binary class classification dataset, are portrayed in Fig. 6. The experimental outcomes imply that the proposed CMOMSNN-ID algorithm established the minimal TL and VL values whereas the VL values were lower than the TL values.

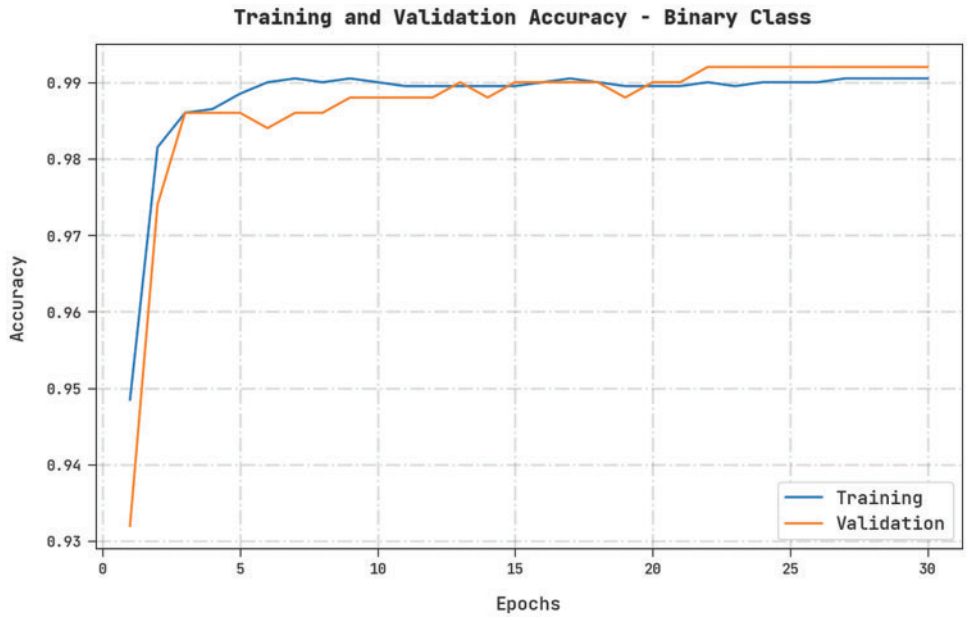

Figure 5: TA and VA analyses results of the CMOMSNN-ID approach under binary class classification dataset

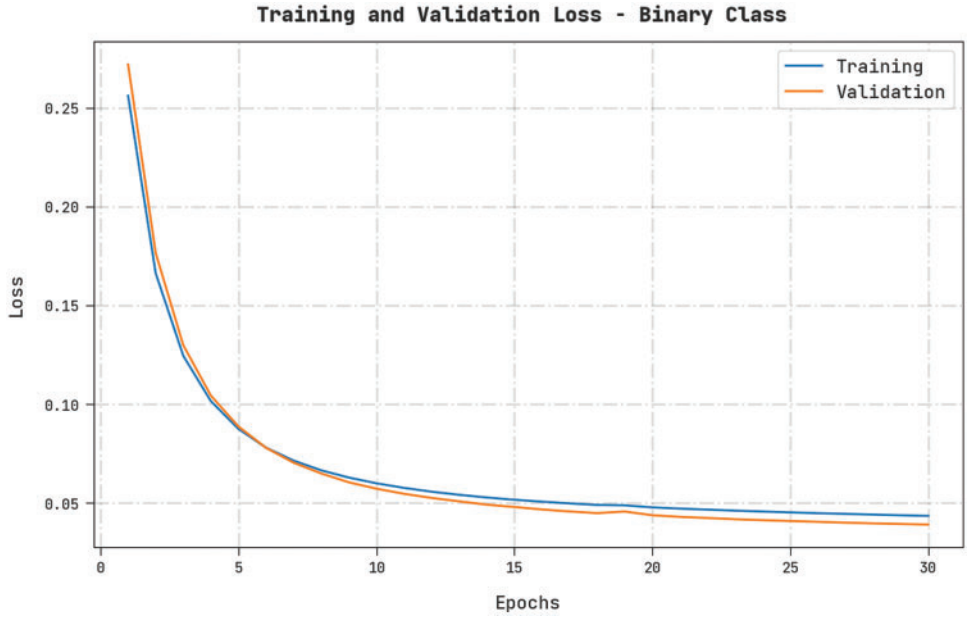

Figure 6: TL and VL analyses results of the CMOMSNN-ID approach under binary class classification dataset

Table 3 shows the details about Multiclass classification dataset which has a collection of 2,500 samples. The dataset includes 500 samples under normal class, 500 samples under Denial of Service (DoS) class, 500 samples under Probe class, 500 samples under Root to Local (R2L) class and 500 samples under User to Root (U2R) class.

Table 3: Multiclass classification dataset details

Class	No. of samples
Normal	500
DoS	500
Probe	500
R2L	500
U2R	500
Total number of samples	2500

Fig. 7 portrays the confusion matrices generated by the CMOMSNN-ID method upon Multiclass classification dataset. On 80% TRS, the proposed CMOMSNN-ID model categorized 383 samples under normal class, 390 samples under DoS class, 375 samples under Probe class, 400 samples underR2L class and 354 samples underU2R class. Moreover, on 20% TSS, the presented CMOMSNN-ID model classified 103 samples under normal class, 88 samples under DoS class, 109 samples under Probe class, 86 samples under R2L class and 94 samples under U2R class respectively. Along with that, on 70% TRS, the proposed CMOMSNN-ID method categorized 383 samples under normal class, 326 samples under DoS class, 342 samples under Probe class, 327 samples under R2L class and 348 samples under U2R class.

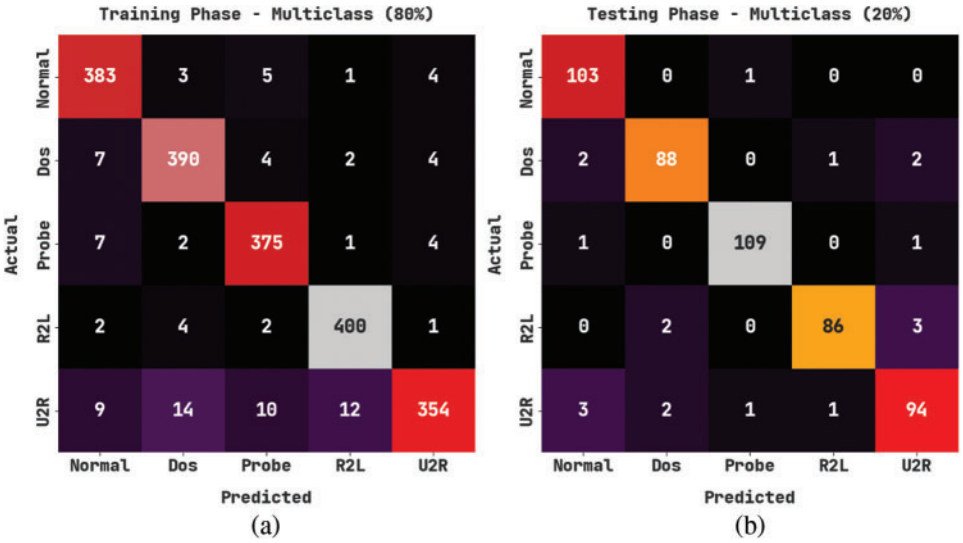

Figure 7: (Continued)

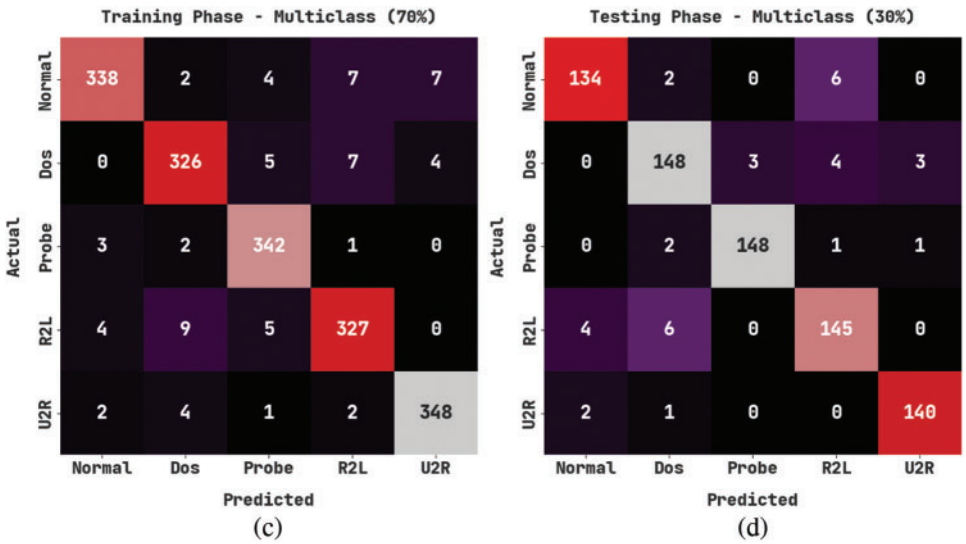

Figure 7: Confusion matrices of the CMOMSNN-ID approach under Multiclass classification dataset (a) 80% of TRS, (b) 20% of TSS, (c) 70% of TRS, and (d) 30% of TSS

Table 4 and Fig. 8 show the overall classification outcomes achieved by the CMOMSNN-ID model on Multiclass classification dataset. With 80% TRS, the proposed CMOMSNN-ID method offered an average $accu_y$ of 98.04%, $prec_n$ of 95.12%, $reca_l$ of 95.09%, F_{score} of 95.07% and an AUC_{score} of 96.93%. Moreover, with 20% TSS, the proposed CMOMSNN-ID model provided an average $accu_y$ of 98.40%, $prec_n$ of 96.01%, $reca_l$ of 95.89%, F_{score} of 95.93% and an AUC_{score} of 97.44%. Eventually, with 70% TRS, the proposed CMOMSNN-ID model accomplished an average $accu_y$ of 98.42%, $prec_n$ of 96.05%, $reca_l$ of 96.05%, F_{score} of 96.04% and an AUC_{score} of 97.53%. At last, with 30% TSS, the proposed CMOMSNN-ID method presented an average $accu_y$ of 98.13%, $prec_n$ of 95.40%, $reca_l$ of 95.37%, F_{score} of 95.38% and an AUC_{score} of 97.10%.

Table 4: Analytical results of the CMOMSNN-ID approach using Multiclass classification dataset under different measures

Labels	Accuracy	Precision	Recall	F-score	AUC score
Training phase (80%)					
Normal	98.10	93.87	96.72	95.27	97.58
DoS	98.00	94.43	95.82	95.12	97.19
Probe	98.25	94.70	96.40	95.54	97.55
R2L	98.75	96.15	97.80	96.97	98.40
U2R	97.10	96.46	88.72	92.43	93.95
Average	98.04	95.12	95.09	95.07	96.93
Testing phase (20%)					
Normal	98.60	94.50	99.04	96.71	98.76
DoS	98.20	95.65	94.62	95.14	96.82
Probe	99.20	98.20	98.20	98.20	98.84

(Continued)

Table 4: Continued

Labels	Accuracy	Precision	Recall	F-score	AUC score
R2L	98.60	97.73	94.51	96.09	97.01
U2R	97.40	94.00	93.07	93.53	95.78
Average	98.40	96.01	95.89	95.93	97.44
Training phase (70%)					
Normal	98.34	97.41	94.41	95.89	96.88
DoS	98.11	95.04	95.32	95.18	97.06
Probe	98.80	95.80	98.28	97.02	98.60
R2L	98.00	95.06	94.78	94.92	96.79
U2R	98.86	96.94	97.48	97.21	98.34
Average	98.42	96.05	96.05	96.04	97.53
Testing phase (30%)					
Normal	98.13	95.71	94.37	95.04	96.69
DoS	97.20	93.08	93.67	93.38	95.91
Probe	99.07	98.01	97.37	97.69	98.43
R2L	97.20	92.95	93.55	93.25	95.85
U2R	99.07	97.22	97.90	97.56	98.62
Average	98.13	95.40	95.37	95.38	97.10

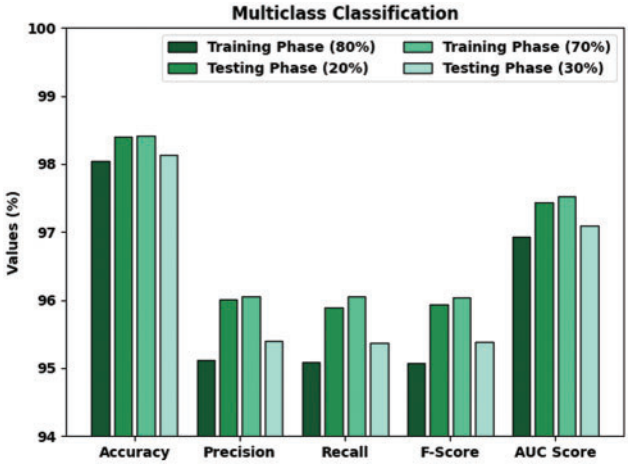

Figure 8: Analytical results of the CMOMSNN-ID approach using Multiclass classification dataset

Both TA and VA values, obtained by the proposed CMOMSNN-ID method on Multiclass classification dataset, are shown in Fig. 9. The experimental outcomes imply that the proposed CMOMSNN-ID technique achieved the maximal TA and VA values whereas the TA values were higher than the TA values.

Both TL and VL values, achieved by the proposed CMOMSNN-ID approach on Multiclass classification dataset, are exhibited in Fig. 10. The experimental outcomes infer that the proposed

CMOMSNN-ID algorithm accomplished the least TL and VL values whereas the VL values were lower than the TL values.

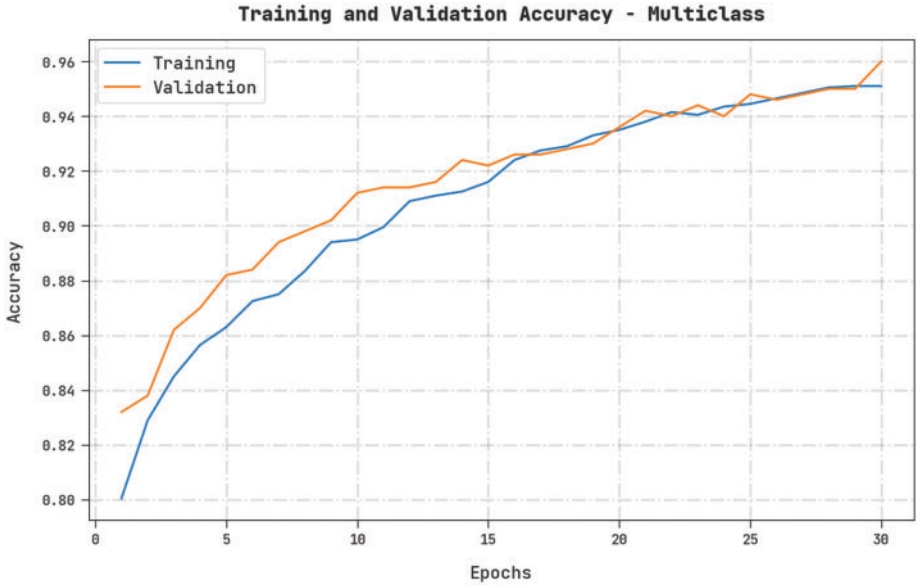

Figure 9: TA and VA analyses results of the CMOMSNN-ID approach under Multiclass classification dataset

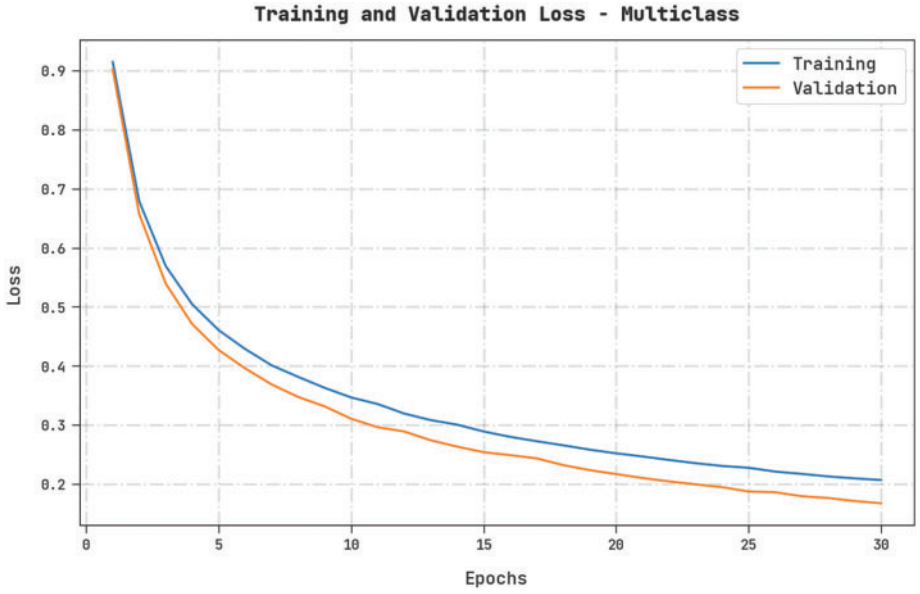

Figure 10: TL and VL analyses results of the CMOMSNN-ID approach under Multiclass classification dataset

To validate the enhanced performance of the proposed CMOMSNN-ID model, a comparative assessment was conducted and the results are shown in Table 5 and Fig. 11. With respect to $accu_y$, the proposed CMOMSNN-ID model achieved the highest $accu_y$ of 99.20%, whereas the k-Nearest

Neighbour (K-NN), Random Forest (RF), Decision Tree (DT), Naïve Bayes (NB), Deep Neural Network with Message Queuing Telemetry Transport (DNN-MQTT), Greedy randomized adaptive search procedure with Annealed Randomness—Forest (GAR-Forest) and the Convolutional Neural Network with Bidirectional Long Short Term Memory (CNN-BiLSTM) models obtained the least $accu_y$ values such as 94.50%, 95.30%, 94.88%, 95.41%, 94.12%, 85.41% and 83.89% respectively. Also, with respect to $prec_n$, the proposed CMOMSNN-ID model attained the highest $prec_n$ of 98.87%, whereas the K-NN, RF, DT, NB, DNN-MQTT, GAR-Forest and the CNN-BiLSTM models gained the least $prec_n$ values such as 97.17%, 98.07%, 98.24%, 97.17%, 94.21%, 88.10% and 86.10% correspondingly. Followed by, with respect to $F1_{score}$, the presented CMOMSNN-ID model achieved the highest $F1_{score}$ of 98.87%, whereas the K-NN, RF, DT, NB, DNN-MQTT, GAR-Forest and the CNN-BiLSTM models attained the least $F1_{score}$ values such as 96.92%, 94.20%, 97.39%, 94.70%, 94.24%, 84.54% and 84.87% correspondingly. Thus, the proposed CMOMSNN-ID model established an effectual performance compared to the existing models.

Table 5: Comparative analysis results of the CMOMSNN-ID approach and other existing methodologies

Methods	Accuracy	Precision	Recall	F1-Score
CMOMSNN-ID	99.20	98.87	98.87	98.87
K-NN	94.50	97.17	97.35	96.92
RF	95.30	98.07	94.98	94.20
DT	94.88	98.24	94.87	97.39
NB	95.41	97.17	96.31	94.70
DNN-MQTT Model	94.12	94.21	97.78	94.24
GAR-Forest	85.41	88.10	84.77	84.54
CNN-BiLSTM	83.89	86.10	84.44	84.87

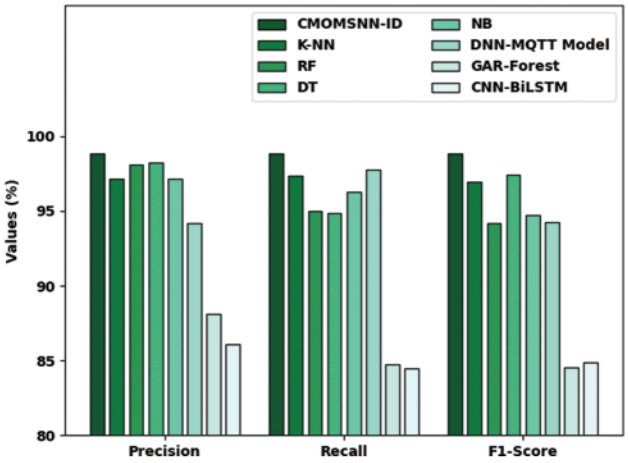

Figure 11: Comparative analysis results of the CMOMSNN-ID approach and other existing methodologies

4 Conclusion

In this study, a new CMOMSNN-ID technique has been developed for detection and recognition of the intrusions in a secure cloud environment. The presented CMOMSNN-ID model involves CABC-FS technique to reduce the curse of dimensionality. Followed by, the MSNN classifier is used based on the simulation of brain functioning. It can be applied to resolve the pattern classification problems. In order to fine-tune the parameters relevant to MSNN model, the WOA approach is employed to boost the classification results. In order to demonstrate the superiority of the proposed CMOMSNN-ID model, a useful set of simulations was conducted. The simulation outcomes confirmed that the proposed CMOMSNN-ID model accomplished superior performance over other models. As a part of future scope, the classification performance can be boosted further using the outlier detection and the clustering approaches.

Funding Statement: This research work was funded by Institutional Fund Projects under Grant No. (IFPHI-099-120-2020). Therefore, authors gratefully acknowledge technical and financial support from the Ministry of education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] O. Achbarou, M. A. El Kiram, O. Bourkoukou and S. Elbouanani, "A new distributed intrusion detection system based on multi-agent system for cloud environment," *International Journal of Communication Networks and Information Security*, vol. 10, no. 3, pp. 526, 2018.

[2] V. Chang, L. Golightly, P. Modesti, Q. A. Xu, L. M. T. Doan *et al.* "A survey on intrusion detection systems for fog and cloud computing," *Future Internet*, vol. 14, no. 3, pp. 89, 2022.

[3] D. G. Singh, R. Priyadharshini and E. J. Leavline, "Cuckoo optimisation based intrusion detection system for cloud computing," *International Journal of Computer Network and Information Security*, vol. 10, no. 11, pp. 42–49, 2018.

[4] V. Balamurugan and R. Saravanan, "Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation," *Cluster Computing*, vol. 22, no. S6, pp. 13027–13039, 2019.

[5] P. Wanda, "A survey of intrusion detection system," *International Journal of Informatics and Computation*, vol. 1, no. 1, pp. 1, 2020.

[6] P. Deshpande, S. C. Sharma, S. K. Peddoju and S. Junaid, "HIDS: A host based intrusion detection system for cloud computing environment," *International Journal of System Assurance Engineering and Management*, vol. 9, no. 3, pp. 567–576, 2018.

[7] Z. Liu, B. Xu, B. Cheng, X. Hu and M. Darbandi, "Intrusion detection systems in the cloud computing: A comprehensive and deep literature review," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, pp. e6646, 2022.

[8] S. Krishnaveni, S. Sivamohan, S. S. Sridhar and S. Prabakaran, "Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing," *Cluster Computing*, vol. 24, no. 3, pp. 1761–1779, 2021.

[9] S. Devi and D. A. K. Sharma, "Understanding of intrusion detection system for cloud computing with networking system," *International Journal of Computer Science and Mobile Computing*, vol. 9, no. 3, pp. 19–25, 2020.

[10] M. M. Sakr, M. A. Tawfeeq and A. B. El-Sisi, "Network intrusion detection system based pso-svm for cloud computing," *International Journal of Computer Network and Information Security*, vol. 11, no. 3, pp. 22–29, 2019.

[11] S. I. Shyla and S. S. Sujatha, "Cloud security: Lkm and optimal fuzzy system for intrusion detection in cloud environment," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 1626–1642, 2019.
[12] E. Besharati, M. Naderan and E. Namjoo, "LR-HIDS: Logistic regression host-based intrusion detection system for cloud environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 9, pp. 3669–3692, 2019.
[13] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri and M. Rida, "New anomaly network intrusion detection system in cloud environment based on optimized back propagation neural network using improved genetic algorithm," *International Journal of Communication Networks and Information Security*, vol. 11, no. 1, pp. 61–84, 2019.
[14] A. N. Jaber and S. U. Rehman, "FCM–SVM based intrusion detection system for cloud computing environment," *Cluster Computing*, vol. 23, no. 4, pp. 3221–3231, 2020.
[15] P. Sharma, J. Sengupta and P. K. Suri, "WLI-FCM and artificial neural network based cloud intrusion detection system," *International Journal of Advanced Networking and Applications*, vol. 10, no. 1, pp. 3698–3703, 2018.
[16] I. J. Jacob and P. E. Darney, "Artificial bee colony optimization algorithm for enhancing routing in wireless networks," *Journal of Artificial Intelligence and Capsule Networks*, vol. 3, no. 1, pp. 62–71, 2021.
[17] S. G. Dastidar and H. Adeli, "A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection," *Neural Networks*, vol. 22, no. 10, pp. 1419–1431, 2009.
[18] I. Aljarah, H. Faris and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Computing*, vol. 22, no. 1, pp. 1–15, 2016.