2024 Second International Conference on Data Science and Information System (ICDSIS)

Securing Cloud Computing Environment via Optimal Deep Learning-based Intrusion Detection Systems

1st Durga Prasada Rao Sanagana
Lead Security Architect
Provident Credit Union,
Redwood City, Bay south of San Francisco
California, USA
durga.dprs@gmail.com

2nd Chaitanya Kanth Tummalachervu
Sr.Site Reliability Engineer/DevOps – Cloud
"RingCental Inc",
Denver, Colorado, USA
tummalachervu@gmail.com

***Abstract*—Cloud computing (CC) has revolutionized the way businesses operate, providing unparalleled scalability and flexibility. However, security concerns loom large with the large quantity of data processed and stored in the cloud. Security is paramount in Cloud Operations to safeguard data, applications, and infrastructure from cyber threats. Intrusion Detection Systems (IDS) have been instrumental in protecting cloud infrastructure by consistently monitoring system and network traffic activities for indications of malicious behavior or unauthorized access. Leveraging advanced anomaly detection techniques and Machine Learning (ML) algorithms, IDS can quickly detect and respond to security risks, helping to reinforce cloud environments against cyber threats. In the digital era, organizations can increase their security posture and ensure the integrity, availability, and confidentiality of their information and services by incorporating strong intrusion detection abilities into CC infrastructures. This study introduces a novel Salp Swarm Algorithm-Based Feature Selection with Deep Learning-Based Intrusion Detection (SSAFS-DLID) method for cloud infrastructure. The proposed method incorporates the SSA for FS, Long Short-Term Memory (LSTM) classification for IDS, and the Adam optimizer for the optimization task. In the context of CC, where security is of great significance, the SSAFS-DLID approach focuses on improving the effectiveness and efficiency of IDSs. The SSA efficiently selects important features from massive datasets, reducing computational complexity and dimensionality while maintaining crucial data. Leveraging LSTM classification, the model can effectively detect anomalies and potential security breaches in cloud infrastructure, offering a strong defence mechanism against different cyberattacks. Furthermore, the application of the Adam optimizer ensures effective convergence and optimization during the process of training. The empirical study highlights the efficacy of the SSAFS-DLID method in accurately detecting intrusions in cloud infrastructures while preserving computational overhead and low false positive rates with 99.71% accuracy. Overall, the proposed model demonstrates promising potential in reinforcing the security environments of CC system.**

Keywords—cloud computing, cloud operations, cloud security, intrusion detection systems, deep learning, adam optimizer

I. INTRODUCTION

Cloud Computing (CC) is utilized in numerous diverse research fields due to its great network capability and computing power [1]. Cost-effectiveness, data security, and flexibility of functioning choices are prepared for this technology to be smart nowadays for remote employees. At present, servers in CC must defend themselves from dangers more cleverly and deliver safety by averting a novel threat [2]. CC is one of the greater technologies linked with consumers dependent upon on-demand facilities. It has significantly transformed the IT organization by providing smart services with spaces, servers, networking, applications, databases, resources, and much more [3]. Depending upon the use of computation effects; the idea of manifold clouds has been modified to satisfy the customer's necessities. Stimulated by the benefits of service excellence and the optimizer of assorted resources, the multi-cloud idea was commonly implemented in the atmosphere of IoT to contract with safety and performance problems [4]. Multi-cloud performs as a middle among associated IoT devices, and the requirement of focusing on safety is the core attention of this work.

The Intrusion Detection System (IDS) is employed to observe network traffic activities to classify malicious events [5]. The IDS can be separated into dual classes as per the recognition device named signature-based recognition, and anomaly-based recognition method. The initial is to use exact forms from the system, for instance, a bytes sequence [6]. It will equate such a sequence with current signature datasets. The next is to use the behavior of network beside recognized baselines. This technique is appropriate for identifying every known and unknown malicious substance. While numerous Machine Learning (ML) and Swarm Intelligence (SI) methods have been developed in the survey, directing intruder assaults in an effective method has been very difficult until now [7]. Therefore, the foremost aim of the function is to define the attacker of intruder method utilizing the deep learning (DL) approach. IDS detects the activities in a system and inspects them for warnings of IDS. IDS can be software hardware, or a combination of both to mechanize the ID procedure [8]. It takes data from the computer below surveillance and updates the system supervisor by logging or sending intrusion actions [9]. On the other hand, the cautions prepared by IDS were infrequently associated with real intrusion since false negatives and positives affect the IDS performance. Whereas, numerous IDS approaches are accessible development is wanted for the recognition efficacy [10]. Furthermore, Feature Selection (FS) models are very vital to removing redundant features that increase false alarms and upsurge the precision of the method.

In [11], developed a Sail Fish Dolphin Optimizer-based Deep Recurrent Neural Networks (SFDO-based DRNN) technique. The proposed SFDO is designed by combining Dolphin Echolocation (DE) and Sail Fish Optimizer (SFO) models. Virtual Machine (VM) immigration and cloud organization are proficient in utilizing ChicWhale model. The feature fusion procedure is implemented by using Deep RNN model which was trained utilizing the identified SFDO model. In [12], a novel IDS technique is projected dependent upon the

979-8-3503-8166-5/24/$31.00 ©2024 IEEE

mixture of DL and optimizer model. At first, a feature extractor model reliant on CNN is proposed. Next, a novel FS technique is employed based on an adapted type of Growth Optimizer (GO) named MGO. The method also employed the Whale Optimizer Algorithm (WOA) to upsurge the searching procedure of GO. Zhang et al. [13] project public cloud systems-oriented DNNs for effectual ID. Specifically, a new ID structure is projected, including fuzzy logic-based FS, consecutive SSA optimized GRU and deep belief network (DBN), combined CNN. Complete approaches were offered for the fuzzy logic model, sequential salp optimizer, DBN architecture, and CNN.

Balasubramaniam et al. [14] developed gradient hybrid leader optimizer (GHLBO) model. This optimized technique is highly liable to train the deep SAE (DSAE) which perceives the attack in an effective method. While, deep maxout networks (DMN) implement the combination of features by an overlay co-efficient, and data growth is executed by the oversampling procedure. Also, the projected GHLBO was made by combining the hybrid leader-based optimizer (HLBO) and gradient descent method. Arvind et al. [15] present a new model to enhance the capability of Cloud service suppliers to examine behaviors of users. This study paper utilized Particle Swarm Optimizer (PSO)-based DL method for the classification and optimization process. In detection process, the system changed consumer's actions into an understandable layout and recognized hazardous performances utilizing multi-layer NNs. Salvakkam et al. [16] intend an exclusive model for perceiving CC intrusions. This projected method is intended to attain dual objectives. Initially, it analyses the difficulties of present IDS, and then it offers a precision development method of IDS. The developed Ensemble IDS for CC Using DL (EICDL) is mainly intended to perceive intrusions efficiently.

This study introduces a novel salp swarm algorithm-based feature selection with deep learning-based intrusion detection (SSAFS-DLID) method for cloud infrastructure. In the context of CC, where security is of great significance, the SSAFS-DLID approach focuses on improving the effectiveness and efficiency of IDSs. The SSA efficiently selects important features from massive datasets, reducing computational complexity and dimensionality while maintaining crucial data. Leveraging LSTM classification, the model can effectively detect anomalies and potential security breaches in cloud infrastructure, offering a strong defence mechanism against different cyberattacks. Furthermore, the application of the Adam optimizer ensures effective convergence and optimization during the process of training. The empirical study highlights the efficacy of the SSAFS-DLID method in accurately detecting intrusions in cloud infrastructures while preserving computational overhead and low false positive rates.

II. The Proposed Model

In this work, we have introduced a novel SSAFS-DLID algorithm for cloud infrastructure. The proposed model incorporates the SSA for FS, LSTM classification for intrusion detection, and the Adam optimizer for the optimization task. Fig. 1 shows the working flow of SSAFS-DLID algorithm.

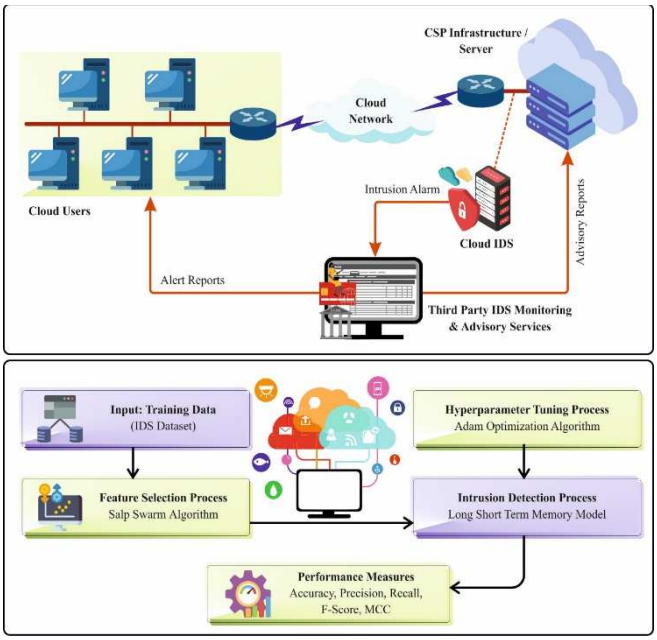

Fig. 1. Working flow of SSAFS-DLID algorithm

A. Feature Selection using SSA

For feature selection process, the SSAFS-DLID method used SSA-FS approach to elect optima features. The consideration is to choose the more suitable and valuable features in the system for predicting the diseases with better proficiency [17]. The SSA is utilized for feature selection (FS), improving the learning method of model by removing undesirable features. The SSA controls the swarming method that can be perceived in salps, a category of aquatic organisms, to arbitrarily choose a population. At the ocean, a salp chain with leader salp is placed in the forward end and alternative salps as a swarm follower. Salp locations are signified in a n-dimensional range wherein 'n' characterizes the overall number of identifiers in a specified issue. The feature optimizer method comprises 3 stages namely the population initialization, the leader's location update, and the follower's location upgrade. Such stages consider the clustering procedure of a SSA. The subsections offer a comprehensive explanation of the working rule of the SSA as given below.

1) Initializing Population: In the $S \times D$ Euclidean method, the population initialization is executed, the variable S becomes the swarm scale and D refers to the dimension with space. Assume the accessible foods as fd and also represented as $\mathrm{fd} = [\mathrm{fd}_1, \mathrm{fd}_2, \ldots, \mathrm{fd}_d]^T$ wherein the location of every salp could be signified as $\mathrm{P}_n = [\mathrm{P}_{n-1}, \mathrm{P}_{n-2}, \ldots., \mathrm{P}_{n-D}]^T$, so that $n = 1,2, \ldots, N$. The lower and upper bound was indicated as U_b, L_b. The lower bound can be characterized as $\mathrm{L}_b = [\mathrm{L}_{b1}, \mathrm{L}_{b2}, \ldots, \mathrm{L}_{bD}]^T$ and upper bound $\mathrm{U}_b = [\mathrm{U}_{b1}, \mathrm{U}_{b2}, \ldots, \mathrm{U}_{bD}]^T$.

The random initialization of the population can be calculated by employing Eq. (1).

$$X_{S\times D} = rand(S \times D) \times (U_b - L_b) + L_b \times ones(S \times D) \quad (1)$$

The followers and leader population in the d-th dimension are denoted as $\chi_{1,d}$ and $\chi_{k,d}$ where, $k = 2,3, \ldots, N$.

2) Updating Leader Position: During a SSA, the leader is accountable for determining the food in the search space. It

should also direct the complete collection for finding food. It becomes crucial to upgrade the location of leader which can be accomplished by applying Eq. (2).

$$x_{1,d} = fd_d + r_1\left((U_{bd} - L_{bd})r_2 + L_{bd}\right) \quad (2)$$

Here, r_1 and r_2 describes randomized numbers at the interval ranges [0, 1]. The individual population diversity, searching capability, and movement of leader should be arbitrarily improved by the parameters stated in Eq. (2). From every meta-heuristic technique, the main factor called as r_1, as determined in Eq. (2). It can be described as the convergence parameter. In the iteration method, this factor equalizes the tradeoff between exploration and exploitation. When r_1 must be higher than 1, the method executes global exploration. Once r_1 is lesser than 1, it considers local exploration for determining a precise estimation value. The value of r_1 must decrease at the range of 2 - 0 for the primary iteration method for performing global search and then enhance the precision of the next iterations. The convergence parameter will be computed by applying Eq. (3).

$$r_1 = 2e^{-\left(\frac{4i}{i_{max}}\right)^2} \quad (3)$$

Now, i characterizes the existing iteration and $i_{\max}$ signifies the overall counts of iterations.

3) Updating Follower Position: The groups assume a chain movement instead of random movement. For determining the motions of followers, specified significant features are required as measured, comprising the speed of movement, acceleration, and initial position of the followers.

According to Newton's law of motion, measure the movement distance, and it will be calculated by applying Eq. (4).

$$Motion\ Distance = \frac{1}{2}\alpha i^2 + s_0 i \quad (4)$$

Here α refers to the acceleration of followers, $i = 1$ the differences that occur among iterations, s_0 defines the followers speed, and it will be 0 with initial iteration, and computed among the primary and last iterations.

The acceleration of followers will be computed by applying Eq. (5). Continuously the followers can follow the predecessor salp.

$$\alpha = \frac{(s_{final} - s_0)}{t} \quad (5)$$

Hence, the movement speed of salp will be determined by employing Eq. (6).

$$s_{final} = \left(x^i_{k-1,d} - x^i_{k,d}\right)/t \quad (6)$$

Where $s_0 = 0$, so that the Motion Distance will be considered as Eq. (7).

$$Motion\ Distance = \frac{1}{2}\left(x^i_{k-1,d} - x^i_{k,d}\right) \quad (7)$$

The follower location will be upgraded by employing Eq. (8).

$$x^{i+1}_{k,d} = x^i_{k,d} + Motion\ Distance = \frac{1}{2}\left(x^i_{k,d} + x^i_{k-1,d}\right) \quad (8)$$

Here, $x^{i+1}_{k,d}$ denoted as the followers' place with $(i+1)$th iteration and $x^i_{k,d}$ defines the dth dimensional kth follower at the ith iteration.

The FF utilized in the SSA is intended to have a balance between the amount of attributes chosen and the classifier accuracy accomplished by using those features selected, Eq. (9) shows the FF to estimate the solution.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \quad (9)$$

In Eq. (9), $\gamma_R(D)$ specifies the classifier error rate. $|R|$ refers to the cardinality of selected subset and $|C|$ shows the overall quantity of attributes in the dataset, α and β denote the two parameters respective to the importance of classifier quality and subset length. $\in [1,0]$ and $\beta = 1 - \alpha$.

B. Intrusion Detection using LSTM classifier

At this stage, the LSTM classification can be employed. RNN is a kind of NN with short-term memory different from BPNN [18]. The RNN comprises hidden layer (HL), h. This layer is extended in time step t, and the series data of previous steps disseminated in h_{t-1} to h_t is protected by HL. The resultant of this layer can transferred at each step. Due to the exploding and vanishing gradient problems, RNN could be utilized for long-term assessment. For resolving these problems, LSTM can be presented, as it has a memory layer that saves one of the essential data. Fig. 2 depicts the infrastructure of LSTM. It trained the network with output gate, forget gate, memory cells, and input gate. It is defined as:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (10)$$

whereas x_t and x_t signifies the input and the output layers of final HL, correspondingly; W_f, b_f, and f_t signifies the weighted, biased, and output of forgetting gates, correspondingly; σ demonstrates the sigmoid activation function. The resultant i_t is illustrated in Eq. (11)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (11)$$

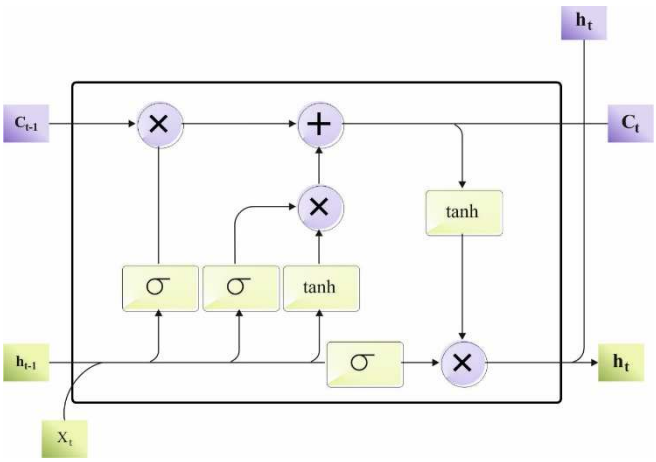

Fig. 2. LSTM architecture

In which, W_i, b_i, and i_t denotes the weighted, biased, and output of input gate, correspondingly.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (12)$$

The parameters W_o, b_o, and 0_t defines the weighted, biased, and output gate, correspondingly. The 3rd value C_t' is measured by Eq. (13)

$$C_t' = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (13)$$

The variables W_c, and b_c are the weighted and biased of unit layer, correspondingly. C_t' and $tanh$ defines the layer of candidate cell and activation function, correspondingly. Additionally, c_t is led in Eq. (14)

$$c_t = f \cdot c + i_t \cdot C_t' \quad (14)$$

c_{t-1}, and c_t demonstrates the unit layers at the preceding and present times. At last, the output h_t is attained by utilizing in Eq. (15)

$$h_t = o_t \cdot tanh(c_t) \quad (15)$$

In which, h_t denotes the outcome of HL at the current time.

C. *Hyperparameter Tuning using Adam optimizer*

Finally, the Adam optimizer ensures effective convergence and optimization during the process of training. Adam is an optimization method that integrates the benefits of two optimization methods namely RMSprop and SGD models, an adaptable learning rate technique [19]. By leveraging both 1st order and 2nd gradient data, Adam dynamically modifies the learning rates for every parameter in the system. This implies that the method automatically alters the learning rate that dependent upon direction and magnitude of the gradients for increasing the effectiveness. The outcome is quicker and highly constant convergence in the model training procedure. Eq. (16) denotes the general formula of Adam:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}}+\varepsilon}\hat{m}_t \quad (16)$$

Here, α describes the learning rate that must be controlled by the sizes of the upgrade steps, θ_t characterizes the model factors in iteration t, $\hat{m}_t$ defines the 1st order moment, evaluated through a weighted movement average of the gradients, $\hat{v}_t$ is the 2nd order moment, assessed by employing a weighted moving average of the squared gradients and ε represents a smaller value for increasing the stability of the method and preventing division by 0.

The Adam optimizer derives an FF to accomplish better classifier results. It describes a positive integer to epitomize the superior performance of the solution candidate. Now, the reduction of the classifier error rate is regarded as the FF.

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{No.of\ misclassified\ samples}{Total\ No.of\ samples} * 100 \quad (17)$$

III. PERFORMANCE VALIDATION

The performance evaluation of the SSAFS-DLID system is tested using the ID dataset, comprising 5 classes and 125973 samples as reported in Table 1.

TABLE I. DETAILS OF DATASET

Class	No. of Instances
DoS	45927
R2l	995
Probe	11656
U2r	52
Normal	67343
Total No. of Instances	125973

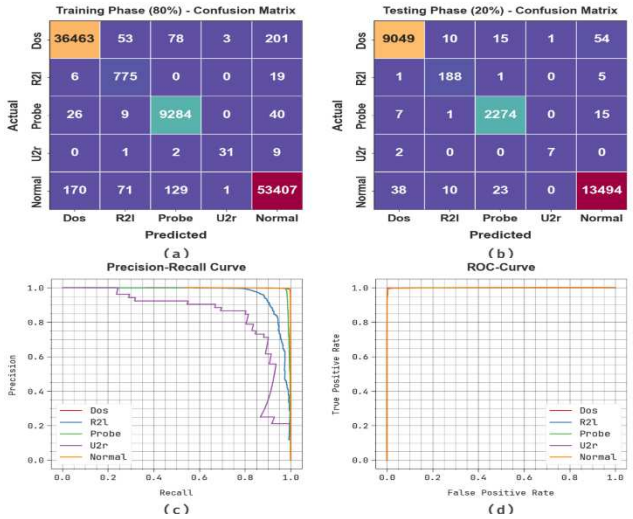

Fig. 3. (a-b) Confusion matrices under 80%:20%TRAS/TESS and (c-d) PR and ROC curves

Fig. 3 examines the classifier results of the SSAFS-DLID system at test dataset. Figs. 3a-3b demonstrates the confusion matrices acquired by the SSAFS-DLID algorithm with 80%:20%TRAS/TESS. This figure pointed out that the SSAFS-DLID algorithm has exactly identified and classified with five class labels. Meanwhile, Fig. 3c represents the PR result of the SSAFS-DLID system. This figure shows that the SSAFS-DLID algorithm gains superior PR effectiveness with every class. In conclusion, Fig. 3d demonstrates the ROC result of the SSAFS-DLID method. This figure emphasized that the SSAFS-DLID technique provides effectual outcomes with increased ROC values at five classes.

TABLE II. ID OUTCOMES OF SSAFS-DLID TECHNIQUE AT 80%:20%TRAS/TESS

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	MCC
TRAS (80%)					
Dos	99.47	99.45	99.09	99.27	98.85
R2l	99.84	85.26	96.88	90.70	90.81
Probe	99.72	97.80	99.20	98.49	98.34
U2r	99.98	88.57	72.09	79.49	79.90
Normal	99.36	99.50	99.31	99.40	98.72
Average	99.68	94.12	93.31	93.47	93.32
TESS (20%)					
Dos	99.49	99.47	99.12	99.30	98.90
R2l	99.89	89.95	96.41	93.07	93.07
Probe	99.75	98.31	99.00	98.66	98.52
U2r	99.99	87.50	77.78	82.35	82.49
Normal	99.42	99.45	99.48	99.47	98.84

Average	99.71	94.94	94.36	94.57	94.36

The ID results of the SSAFS-DLID method are provided in Table 2 and Fig. 4. The experimentation values highlighted that the SSAFS-DLID technique properly recognized the intrusions. With 80%TRAS, the SSAFS-DLID technique offers average $accu_y$ of 99.68%, $prec_n$ of 94.12%, $reca_l$ of 93.31%, F_{score} of 93.47%, and MCC of 93.32%. Additionally, based on 20%TESS, the SSAFS-DLID method gains average $accu_y$ of 99.71%, $prec_n$ of 94.94%, $reca_l$ of 94.36%, F_{score} of 94.57%, and MCC of 94.36%, respectively.

The classifier results of the SSAFS-DLID technique are graphically depicted in Fig. 5 in the form of training accuracy (TRAAC) and validation accuracy (VALAC) curves. The experimental outcome illustrates valuable insight into the behavior of the SSAFS-DLID technique over various epochs, representing its learning task and generalizability. Particularly, the outcome indicates a consistent improvement in the TRAAC and VALAC with maximum epochs. It ensures the adaptive nature of the SSAFS-DLID approach in the pattern detection model on TRA and TES datasets. The growing trend in VALAC showcases its proficiency of the SSAFS-DLID method in adapting to the TRA dataset and excels in providing correct classifier of hidden data, demonstrating strong generalizability.

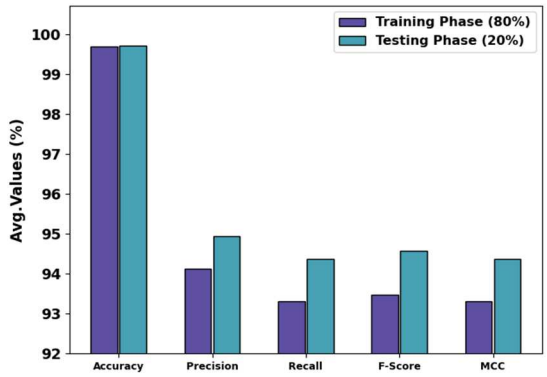

Fig. 4. Average of SSAFS-DLID method on 80%:20%TRAS/TESS

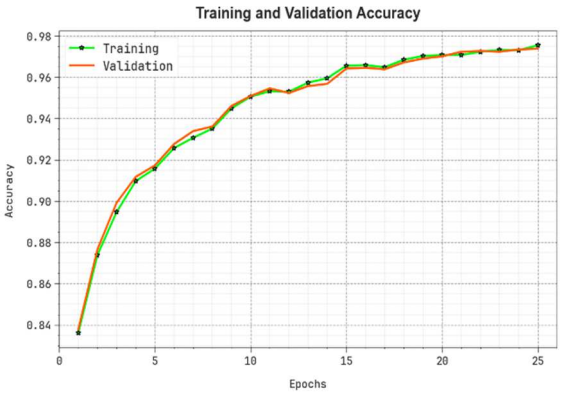

Fig. 5. $Accu_y$ curve of SSAFS-DLID method

Fig. 6 illustrates a comprehensive review of the training loss (TRLA) and validation loss (VALL) results of the SSAFS-DLID technique over different epochs. The progressive decline in TRLA highlights the SSAFS-DLID method reducing the classification error and enhancing the weights on the TRA and TES datasets. The outcomes infers a clear understanding of the SSAFS-DLID approaches association with the TRA dataset, emphasizing its ability to capture patterns within both datasets. Particularly, the SSAFS-DLID method constantly progresses its parameters in diminishing the variances among the prediction and real TRS class labels.

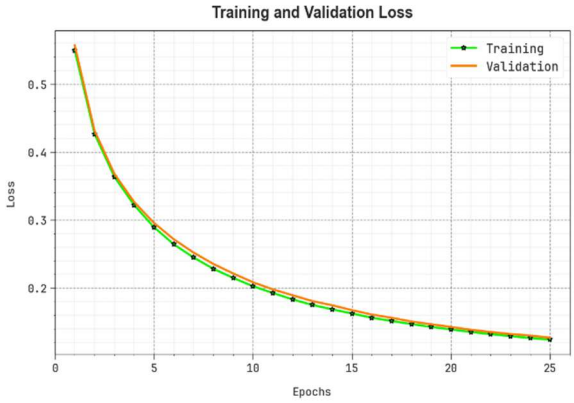

Fig. 6. Loss curve of SSAFS-DLID method

TABLE III. COMPARATIVE OUTCOMES OF SSAFS-DLID SYSTEM WITH OTHER ALGORITHMS

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
SSAFS-DLID	99.71	94.94	94.36	94.57
IMFL-IDSCS	99.31	92.03	78.25	81.80
LKM-OFLS	89.34	84.64	74.68	78.26
K-means-OFLS	91.43	85.74	75.51	78.33
MLP	91.46	86.61	76.76	74.99
PCA-NN	90.08	84.56	76.06	77.54
FCM-OFLS	93.38	82.74	74.43	75.68

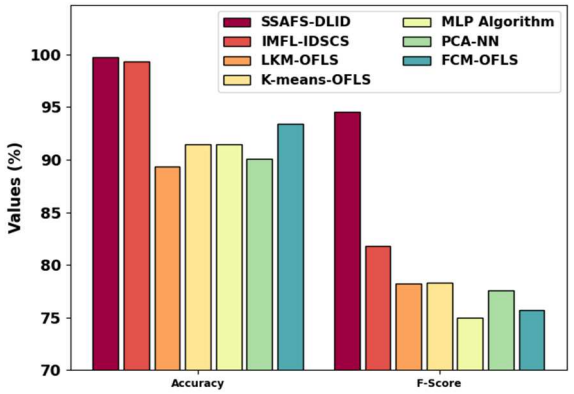

Fig. 7. $Accu_y$ and F_{score} outcomes of SSAFS-DLID technique with recent methods

Table 3 inspects the comparative results of the SSAFS-DLID technique [20]. Fig. 7 provides the comparison study of the SSAFS-DLID technique in terms of $accu_y$ and F_{score}. Based on $accu_y$, the SSAFS-DLID technique offers boosted $accu_y$ of 99.71% while the IMFL-IDSCS, LKM-OFLS, K-means-OFLS, MLP, PCA-NN, and FCM-OFLS techniques obtain degraded $accu_y$ of 99.31%, 89.34%, 91.43%, 91.46%, 90.08%, and 93.38%, correspondingly. Also, with F_{score}, the SSAFS-DLID system achieves higher F_{score} of 94.57% whereas the IMFL-IDSCS, LKM-OFLS, K-means-OFLS, MLP, PCA-NN, and FCM-OFLS algorithms acquires minimized F_{score} of 81.80%, 78.26%, 78.33%, 74.99%, 77.54%, and 75.98%.

Fig. 8 displays the comparison assessment of the SSAFS-DLID method with respect to $prec_n$, and $reca_l$. According to $prec_n$, the SSAFS-DLID algorithm provides increased $prec_n$ of 94.94% but the IMFL-IDSCS, LKM-OFLS, K-means-OFLS, MLP, PCA-NN, and FCM-OFLS systems get diminished $prec_n$ of 92.03%, 84.64%, 85.74%, 86.61%, 84.56%, and 82.74%, respectively. Moreover, based on $reca_l$, the SSAFS-DLID method accomplishes improved $reca_l$ of 94.36% while the IMFL-IDSCS, LKM-OFLS, K-means-OFLS, MLP, PCA-NN, and FCM-OFLS technique offers reduced $reca_l$ of 78.25%, 74.68%, 75.51%, 76.76%, 76.06%, and 74.43%.

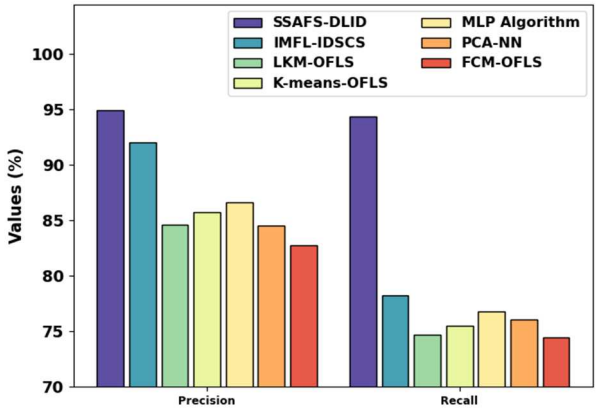

Fig. 8. $Prec_n$ and $Reca_l$ results of SSAFS-DLID model with recent systems

Hence, the SSAFS-DLID technique appears to be an effectual tool for intrusion identification in the cloud environment.

IV. Conclusion

In this article, we have established a novel SSAFS-DLID technique for cloud infrastructure. The SSAFS-DLID model incorporates the SSA for FS, LSTM classification for intrusion detection, and the Adam optimizer for the optimization task. The SSA efficiently selects important features from massive datasets. Leveraging LSTM classification, the model can effectively detect anomalies and potential security breaches in cloud infrastructure, offering a strong defence mechanism against different cyberattacks with 99.71% accuracy. Furthermore, the application of the Adam optimizer ensures effective convergence and optimization during the process of training. The empirical study highlights the efficacy of the SSAFS-DLID method in accurately detecting intrusions in cloud infrastructures while preserving computational overhead and low false positive rates.

References

[1] Krishnaveni, S.; Sivamohan, S.; Sridhar, S.; Prabhakaran, S. Network intrusion detection based on ensemble classification and feature selection method for cloud computing. Concurr. Comput. Pract. Exp. 2022, 34, e6838.

[2] Singh, D.A.A.G.; Priyadharshini, R.; Leavline, E.J. Cuckoo optimisation based intrusion detection system for cloud computing. Int. J. Comput. Netw. Inf. Secur. 2018, 11, 42–49.

[3] Hatef, M.A.; Shaker, V.; Jabbarpour, M.R.; Jung, J.; Zarrabi, H. HIDCC: A hybrid intrusion detection approach in cloud computing. Concurr. Comput. Pract. Exp. 2018, 30, e4171.

[4] Kannadhasan, S.; Nagarajan, R.; Thenappan, S. Intrusion detection techniques based secured data sharing system for cloud computing using msvm. In Proceedings of the 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 23–25 March 2022; pp. 50–56.

[5] Meryem, A.; Ouahidi, B.E. Hybrid intrusion detection system using machine learning. Netw. Secur. 2020, 2020, 8–19.

[6] Achbarou, O.; El Kiram, M.A.; Bourkoukou, O.; Elbouanani, S. A new distributed intrusion detection system based on multi-agent system for cloud environment. Int. J. Commun. Netw. Inf. Secur. 2018, 10, 526.

[7] Ma, X.; Fu, X.; Luo, B.; Du, X.; Guizani, M. A design of firewall based on feedback of intrusion detection system in cloud environment. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.

[8] Guezzaz, A.; Asimi, A.; Asimi, Y.; Azrour, M.; Benkirane, S. A.; Asimi, A.; Asimi, Y.; Azrour, M.; Benkirane, S. A distributed intrusion detection approach based on machine leaning techniques for a cloud security. In Intelligent Systems in Big Data, Semantic Web and Machine Learning; Gherabi, N., Kacprzyk, J., Eds.; Springer: Cham, Switzerland, 2021; Volume 1344, pp. 85–94.

[9] Chang, V.; Golightly, L.; Modesti, P.; Xu, Q.A.; Doan, L.M.T.; Hall, K.; Boddu, S.; Kobusi ´nska, A. A survey on intrusion detection systems for fog and cloud computing. Future Internet 2022, 14, 89.

[10] Luo, G.; Chen, Z.; Mohammed, B.O. A systematic literature review of intrusion detection systems in the cloud-based IoT environments. Concurr. Comput. Pract. Exp. 2022, 34, e6822.

[11] Srinivas, B.V., Mandal, I. and Keshavarao, S., 2024. Virtual machine migration-based Intrusion Detection System in cloud environment using deep recurrent neural network. Cybernetics and Systems, 55(2), pp.450-470.

[12] Fatani, A., Dahou, A., Abd Elaziz, M., Al-Qaness, M.A., Lu, S., Alfadhli, S.A. and Alresheedi, S.S., 2023. Enhancing intrusion detection systems for IoT and cloud environments using a growth optimizer algorithm and conventional neural networks. Sensors, 23(9), p.4430.

[13] Zhang, J., Peter, J.D., Shankar, A. and Viriyasitavat, W., 2024. Public cloud networks oriented deep neural networks for effective intrusion detection in online music education. Computers and Electrical Engineering, 115, p.109095.

[14] Balasubramaniam, S., Vijesh Joe, C., Sivakumar, T.A., Prasanth, A., Satheesh Kumar, K., Kavitha, V. and Dhanaraj, R.K., 2023. Optimization enabled deep learning-based DDoS attack detection in cloud computing. International Journal of Intelligent Systems, 2023.

[15] Arvind, S., Balasubramani, P., Hemanand, D., Ashokkumar, C., Ravuri, P., Sharath, M.N. and Muppavaram, K., 2024. Utilizing deep learning and optimization methods to enhance the security of large datasets in cloud computing environments. In MATEC Web of Conferences (Vol. 392, p. 01143). EDP Sciences.

[16] Salvakkam, D.B., Saravanan, V., Jain, P.K. and Pamula, R., 2023. Enhanced Quantum-Secure Ensemble Intrusion Detection Techniques for Cloud Based on Deep Learning. Cognitive Computation, 15(5), pp.1593-1612.

[17] Revathi, T.K., Balasubramaniam, S., Sureshkumar, V. and Dhanasekaran, S., 2024. An Improved Long Short-Term Memory Algorithm for Cardiovascular Disease Prediction. Diagnostics, 14(3), p.239.

[18] Yao, Q., Song, X. and Xie, W., 2024. State of health estimation of lithium-ion battery based on CNN–WNN–WLSTM. Complex & Intelligent Systems, pp.1-18.

[19] Fernandez-Grandon, C., Soto, I. and Zabala-Blanco, D., Extreme Learning Machine for iris-based diabetes detection.

[20] Alohali, M.A., Elsadig, M., Al-Wesabi, F.N., Al Duhayyim, M., Mustafa Hilal, A. and Motwakel, A., 2023. Enhanced chimp optimization-based feature selection with fuzzy logic-based intrusion detection system in cloud environment. Applied Sciences, 13(4), p.2580.