Journal of Intelligent Systems and Internet of Things
Vol. 14, No. 02, PP. 62-77, 2025
DOI: https://doi.org/10.54216/JISIoT.140206

Security Inspection for Data Computing Networks Using Deep Learning Techniques

Alaa Q. Raheema[1,*]
[1]Civil Engineering Department, University of Technology-Iraq, Baghdad, 10066, Iraq
40345@uotechnology.edu.iq

Abstract

Deep learning offers practical answers for neural network models when applied to cloud registering security. Via robotization Distinguish dangers, decrease manual checking, and further develop in general security adequacy. Deep learning network models assume a pivotal part in security errands like interruption discovery, malware identification, anomaly recognition, and log examination. requires Deep Learning mix in cloud security cautiously assesses existing frameworks, characterizes goals, chooses dataset with arrangement, model tuning and last changes for consistence. Moreover, applying deep learning methods in cloud security requires thought of variables, for example, computational assets, information assortment, arrangement costs, model turn of events, mix endeavors, and continuous observing and support. This study proposes an artificial neural network (ANN) model portrayal in the cloud to track down cloud security parts and recreate security techniques and researches the essential moves toward coordinate these models in the cloud. Regarding that the adequacy of the ANN scheme relies upon cloud parameters like the nature of the preparation information and the network architecture Also, weight change calculations. The review emploies a dataset from Kaggle.com to approve the recreation and blueprints the means Partake in preparing and assessment of the ANN structure.

Received: March 11, 2024 Revised: June 16, 2024 Accepted: October 07, 2024

Keywords: Cloud Computing; Artificial Neural Networks; Anomally Detection; Network Security; Deep Learning

1. Introduction

Deep learning (DL) procedures have arisen to be successful apparatuses for improving security in different spaces. Such procedures influence the abilities of deep learning networks to gain and recognize features through huge sums of information, empowering further hearty and proficient security arrangements. Whenever employed to cloud computing security, artificial intelligent strategies provides a few advantages having forward ramifications in the expense viability of the items. An extraordinary advantage of the execution of DL in security of the cloud is observed in mechanized danger recognition. DL calculations can naturally break down enormous volumes of information, for example, network traffic logs, framework logs, and client conduct, to distinguish oddities and potential security dangers [1, 2]. This along with different uses because of DL, diminishes the requirement for manual observing and investigation, which prompts quicker recognizable proof of security episodes, opportune reaction, mitigation, furthermore, diminished costs in time and resources. Taking into account this, through the automation of different security undertakings, DL limits the gamble of human mistake, which can prompt security breaks and related costs. Mechanized frameworks fueled by deep learning can perform assignments reliably and precisely, improving by and large security viability. The DL structures learn arrangements with connections in information, considering further exact danger location and classification. Several malware types have been found in literature those affecting compatible cloud networks secure operation. Figure 1 shows different classes of cloud network malware analysis [3-5].

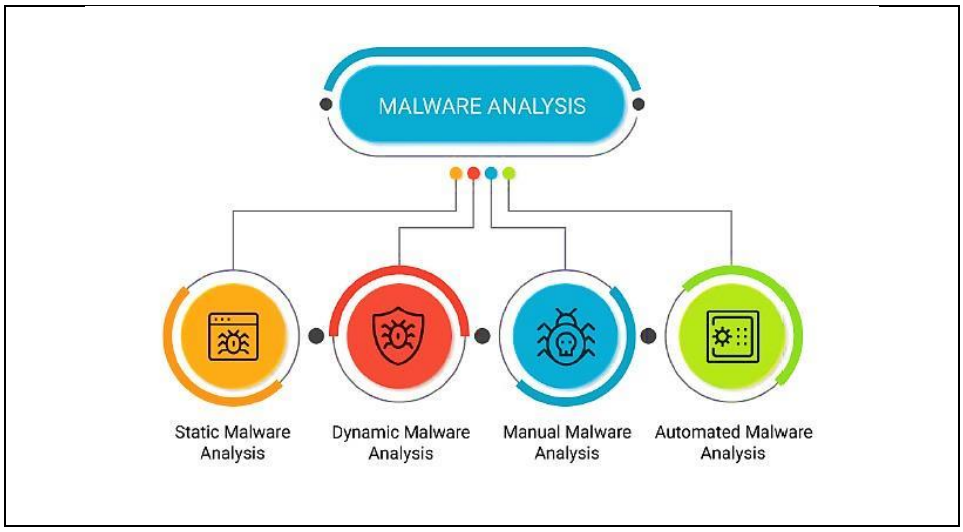


Figure 1. Demonstration of various classes of cloud network malware analysis

Through identifying patterns and abnormalities that could slip by everyone's notice by conventional rule-relied frameworks, DL execution composes it feasible for associations to optimize allocation of resource as well as limit superfluous expenses which is important network security demand. In any case, with the emerging trends and improvements, ongoing identification stays the most significant and testing issue for any framework. To handle this issue, it is important to have the option to offer arrangements that could dissect information continuously and empower brief recognizable proof of potential security threats. Taking into account that DL models might adjust to advancing dangers and gain from new information progressively, such a methodology turns into the best instrument for enhancement. The security framework turns out to be more proficient and smart by limiting reaction times, staying away from misleading up-sides, and diminishing the requirement for successive manual updates and arrangement. In addition, by examining authentic information with arrangements, DL produces important experiences toward expected eventual dangers. Along prescient examination, associations could intensevely designate resources and execute defensive metrics, relieving expected security occurrences and their related expenses [4-7]. Aside from the previously specified advantages, there are other non-specialized advantages to be considered. To be specific, subjective variables, for example, further developed standing, client trust, upper hand, and so on. Such variables ought to likewise be essential for the 10,000 foot view as they add to the general benefit of embracing deep learning strategies. As observed along the expressed advantages, deep learning strategies permit practical answers for cloud computing security by different matters. Coordinating deep learning could optimize resource allocation, improve generally security adequacy, furthermore, alleviate expected monetary misfortunes. Figure 2 displays schematic diagram of cloud network resource allocation security demands using deep learning techniques [4-8].

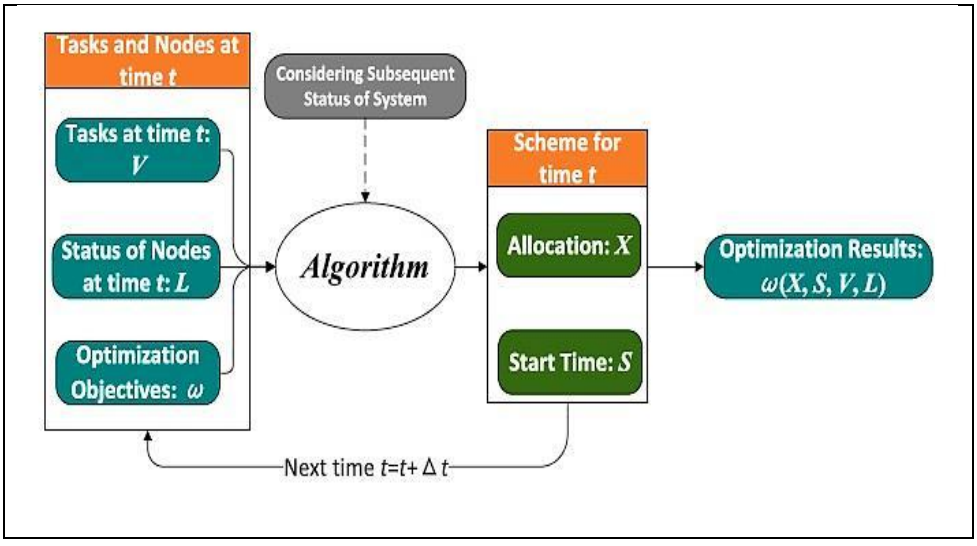


Figure 2. Schematic diagram of cloud network security demands using deep learning techniques.

1.1 Cloud Security Integration with Deep Learning Approaches

Deep learning schemes and techniques are assumed as important part in improving cloud netwirks security. Such structures are used in different cloud security implementations, containing threats recognition, malware identification, oddity discovery, log examination, access control, and so on. As currently referenced, the greatest worth of DL models lies in the capacity to learn complex patterns, distinguish anomalies, and adjust to advancing dangers. Nonetheless, with regards to picking the explicit models for cloud security, it completely relies upon the idea of the issue, accessibility of computational resources, information, sensitivity and the design of the current framework, and different prerequisites case-intended for the association. A portion of the generally utilized deep learning models and algorithms with regards to cloud security incorporate CNN, RNN, LSTM, GAN, DRL and so on [5-10]. Associations can coordinate deep learning methods into their cloud security methodologies in more than one way. As of now examined, associations can have numerous advantages from the integration of deep learning in cloud computing security. No matter what the application in errands like Information Examination, Peculiarity Location, Malware Identification and Classification, Intrusion Location, Avoidance, Client Verification, and Access Control, the effective execution of deep learning procedures requires sufficient techniques and resources [8-12]. Also, Figure 3 presents the fundamental stages for incorporating deep learning into cloud security.

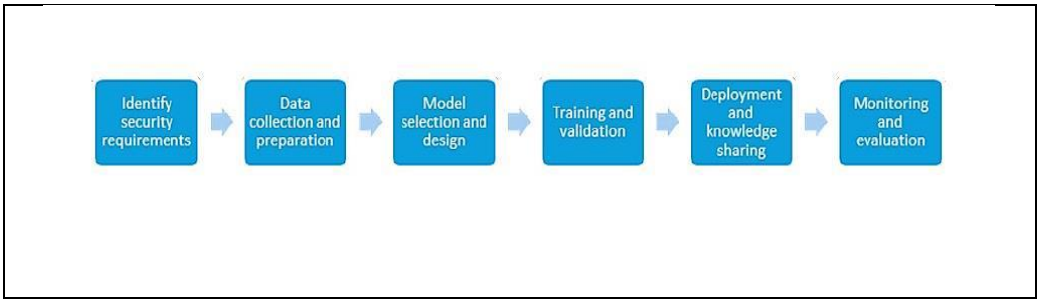


Figure 3. Demonstration block diagram of the fundamental stages for incorporating deep learning into cloud security

Through studying such aspects, frameworks could effectively coordinate deep learning toward their cloud computing security procedures. Notwithstanding, for networks that need to incorporate deep learning toward current arrangements with structures security, the interaction requires a further comprehensive processing. The primary stage of execution needs an evaluation of the qualities and shortcomings of the ongoing framework. This would permit regions to be distinguished all the more effectively and undertakings that might demand or permit DL integration [11-15]. The period of finding the regions and assignments that could be improved utilizing DL is firmly connected with pre-characterized objectives or on the other hand objectives the association has. Having plainly characterized objectives, to decide the particular assignments or difficulties that deep learning can address inside the cloud security framework is vital for the improvement of techniques. The last option has direct sign to the model choice that is appropriate for the recognized errands and purposes. Preparing the model requires a huge measure of information that are pre-handled and painstakingly chose for the given undertaking. Given the significance of dataset choice, information handling, and readiness, this stage requires an intensive examination. The entire achievement of the model relies upon the dataset and information preparation. Subsequent to settling on the information, to adjust to the particular cloud security setting it is expected to consider the tuning of the model including changing hyper-parameters, leading cross-validation, enhancement and so forth. The last integration interaction could require inevitable changes to guarantee similarity and integration with existing parts. Observing and feedback are important for the circle of effective integration, consequently having the right components made arrangements for this stage is fundamental [12-16].

1.2 Deep Learning Optimal Resources Allocation For Cloud Computer Networks

As an example of optimal deep learning (DL) analytical techniques employed to tasks scheduling and resources allocation for cloud computer networks, one might regard the artificial network algorithms (ANNs). These algorithms are distinguished by their ability to adapt to the applied environment and to represent the computer network through their capabilities to provide a number of layers in addition to preparing processing functions and load nodes (neurons). Figure 4 displays a block diagram of the DL ANN algorithm scheme utilized as analytical optimization strategy.

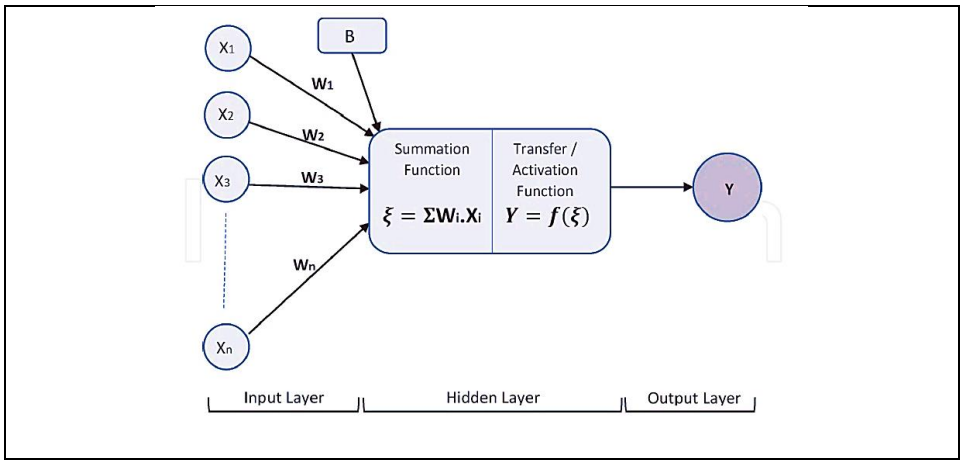


Figure 4. Deep learning ANN structure

Artificial intelligence algorithms are regarded as part of the better optimal analytical techniques employed in scheduling tasks and allocating resources for cloud networks. Its operating mechanisms might be represented along the following mathematical relations. The nets data could be determined as follows:

$$y_{in} = x_1.w_1 + x_2.w_2 + \ldots + x_n.w_n + b \quad (1)$$

The improvement function is one of the basic remains of a neuron. A small starting ability could become through of (restriction function, direct density, and sigmoid function). With such part, one have chosen a sigmoid function, for its nonlinearity that contains of it probable to imprecise either quantity. Also, the resulting value, the neuron might be produced in the hidden layer:

$$y = F(y_{in}) + \sum w_i * x_i + b \quad (2)$$

Thus, Figure 5 illustrates an applied example of utilizing optimal analytical techniques to accomplish task scheduling and resource allocation missions in cloud networks [12-18].

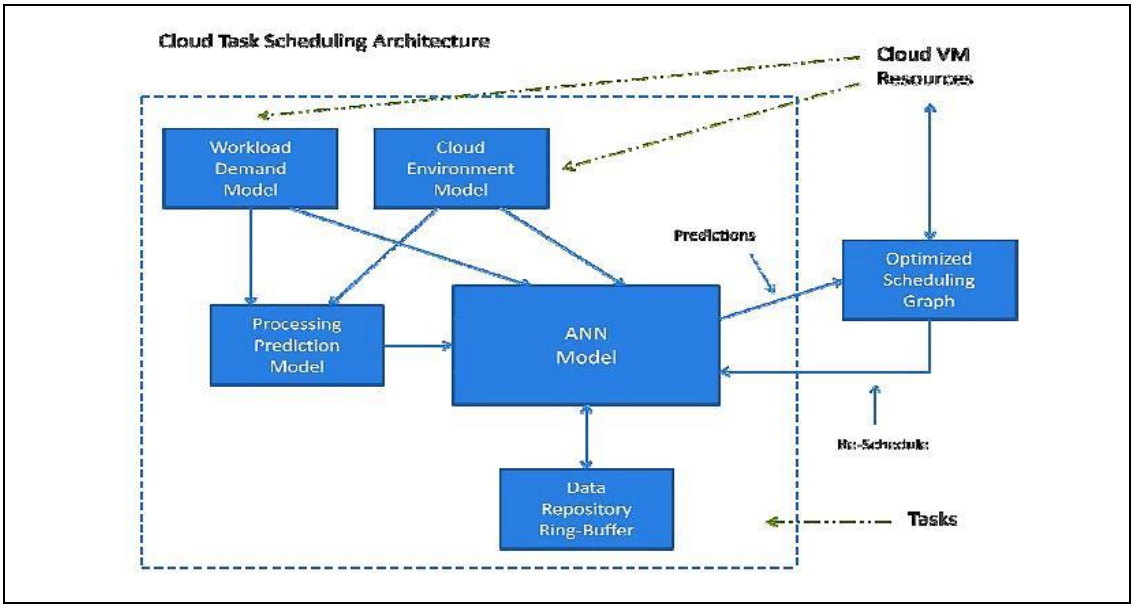


Figure 5. Demonstration of using optimal analytical technique to achieve task scheduling and resource allocation secure operations in cloud networks

As has been shown in previous studies, these ideal mathematical techniques suffer from some important limitations and obstacles. Among the most important of these limitations is the lack of limitation of the data used and not addressing the time series structure of requests, in addition to not addressing the trade-offs between energy efficiency and other metrics, and also "focusing only on CPU usage, ignoring other metrics. Moreover, and to know how effectively an artificial insight algorithm functions, there are another significant evaluation metrics ought to be found to show the strength of the ANN algorithm results execution, In this manner, the evaluation metrics may be communicated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \quad (3)$$

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (4)$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (5)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (6)$$

$$F\ score = \frac{2 \times Precision \times Sensitivity}{Precision + Sensitivity} \times 100\% \quad (7)$$

Where, TP, TN, FP, and FN, are the True positive, True Negative, False positive, and False negative qualities acquired from the ANN preparing results, by alluding to the error histogram. As a matter of fact, the false positive amounts will address the model mistakenly foreseeing positive examples from the model. Moreover, a false negative is one in which the model erroneously predicts the negative class tests from the model. Likewise, true positive qualities result from the model accurately foreseeing the positive examples from the model. At long last, a true negative is one in which the model accurately predicts the negative examples from the model.

2. Literature Review

In this section, the latest specialized articles and scientific research related to the literature on evaluating cybersecurity measures and cases of hacking into virtual cloud network nodes will be reviewed. Where researchers' contributions and suggestions are identified, advantages and limitations are compared, and authors' recommendations are reviewed to improve performance efficiency and reduce the error rate. In 2020, X. Zhang, et., al., [13] proposed a dynamic estimating framework in view of a Stackelberg game model to handle the issue of expanding the pay for cloud computing suppliers of Foundation as a Help (IaaS) and Programming as an Assistance (SaaS). The reenactment aftereffects of this study show that as far as income expansion and resource use, the proposed instrument performs better compared to fixed estimating components and sale based evaluating frameworks. In 2021, J. Chen, et., al., [14], introduced a methodology for allocating resources to address developing issues in cloud computing. A multi-objective enhancement procedure is introduced, with the principal objectives of matching resource execution and rate distances and decreasing the quantity of actual servers utilized. As well as amplifying resource use and diminishing the tackling time, the RAA-PI-NSGAII strategy likewise upgrades the arrangement set quality and distribution uniformity. Nonetheless, the particular difficulties or limits experienced while executing the suggested resource allocation calculation for developing cloud computing demands are excluded from the review. In 2021, N. Swatthong and C. Aswakul, [15] plan to work on the presentation and productivity of cloud computing by scheduling undertakings in compartments utilizing an analytical model. The model purposes whole number direct programming to expand the typical remuneration of errands while thinking about the resource and demand imperatives. This review tried this thought utilizing two cloud situations: an edge-center cluster and a distributed united cloud. The model performed better compared to the standard cooperative scheduling strategy in regards to task fulfillment time and award level. The concentrate additionally features the significance of adaptable and fine-grained task detachment in cloud engineering. In any case, this study doesn't address the time-series construction of the solicitations, which could influence the most extreme sum. In 2023, M. Yadav and A. Mishra, [16], the creators recommended an upgraded ordinal improvement strategy matched with straight relapse to optimize task scheduling in cloud computing to limit the makespan. This technique limits the quest space for scheduling and productively relegates the workload to the best timetable. Furthermore, it estimates future dynamic workload to accomplish an objective least makespan. In any case, there is no correlation with existing ways to deal with approve the viability of the proposed strategy.

In 2023, K. Y. Tai, F. Y. S. Lin, and C. H. Hsiao, [17] prompted a better calculation for overseeing computing resources and energy in different cloud computing focuses, considering elements, for example, energy use, task scheduling, and execution time. It uses Lagrangian unwinding and numerical programming to lay out superior execution, low-energy cloud computing offices. In any case, the review doesn't address the potential compromises between energy productivity and other execution metrics, for example, task execution time or resource usage. In 2022, M. S. Al-Asaly, et., al., [18] introduced an autonomous and intelligent workload prediction method for cloud resource allocation utilizing a deep learning system, in particular a dispersion convolutional repetitive neural organization (DCRNN) model. The goal is to improve prediction accuracy and diminish the hole among estimated and genuine workloads in cloud computing arrangements. Tried on genuine PlanetLab computer processor utilization information, the model outperformed other deep learning models with a root-mean-square mistake of

2.40% and a normal outright rate blunder of 0.18%. Be that as it may, the review centers explicitly around central processor use as the prediction model's feedback information; it disregards different metrics or elements like memory utilization or organization traffic. In 2019, M. Ghobaei-Arani and A. Souri, [19], the creators proposed a straight programming approach called LP-WSC for web administration piece in geographically circulated cloud conditions, to further develop QoS boundaries. This approach chooses the most effective help for each solicitation and essentially lessens the expense of choosing and designing administrations while expanding the accessibility of administrations and unwavering quality of servers contrasted with different strategies. Be that as it may, the recommended LP-WSC technique isn't contrasted with other state of the art strategies for web administration organization in geographically scattered cloud conditions in this review. In 2021, F. Yao, C. Pu, and Z. Zhang, [20], proposed an undertaking duplication-based scheduling calculation (TDSA) to upgrade the makespan for spending plan restricted workflows, using inactive openings on resources and reallocating the extra spending plan. The TDSA calculation shows prominent improvements in the makespan of workflows (up to 17.4%) and the usage of cloud computing resources (up to 31.6%) contrasted with the four pattern algorithms, as proven by probes randomly created and genuine workflows. Nonetheless, the review doesn't consider the running time of workflow errands, and how much information moved among workflow assignments is exceptionally uncertain. In 2020, P. S. Rawat, et., al., [21], proposed an Enormous detonation Huge Crunch model for resource allocation in cloud arrangements that is financially savvy to give variable work assignments on virtual machines. It focuses on a universally ideal result through a goal capability that considers metrics like expense and time, outperforming conventional static, dynamic, and bio-roused provisioning techniques. Nonetheless, the creators didn't talk about the limits and the information source didn't referenced in the review. In 2021, W. Shi, D. Tang, and P. Zou, [22] suggested a standardized model portrayal of assembling resources utilizing metadata and cosmology demonstrating techniques. They likewise present a technique for cooperative scheduling of assembling resources between undertakings utilizing blended set programming, which improves resource allocation and collaboration proficiency. In future work, the creators recommend that the model and scheduling system be coordinated with the investigation of between big business strategies to additionally work on the effectiveness of between big business resource collaboration. Nonetheless, the review doesn't look at the proposed model and scheduling technique with the current methodologies for leading enormous scope tests. In 2023, K. Huang, et., al., [23] a heuristic multi-objective errand scheduling structure for holder based clouds, leveraging actor-critic reinforcement learning to upgrade task scheduling effectiveness. It tends to the intricacies of resource allocation and undertaking the board in dynamic conditions, showing further developed execution metrics, for example, resource use and execution time through broad recreations. Nonetheless, the examination is restricted by its emphasis on unambiguous cloud conditions, which might influence the generalizability of the discoveries. Furthermore, it might struggle with constant flexibility in profoundly factor workloads.

In 2022, L. Hamid, et., al.,[24] analyzed different algorithms of scheduling errands in cloud computing in view of makespan involving workflows as datasets. Trial results show that the early bird gets the worm calculation beats cooperative effort, min, and max-min in the CyberShake workflow, while max-min outflanks the early bird gets the worm, Cooperative effort, and min in the Montage workflow. Nonetheless, analysts don't examine the makespan estimation when the quantity of server farms or hosts increments. In 2022, S. C. Nayak, et., al., [25], the scientists mean to limit the undertaking rejection proportion and expand the errand acknowledgment proportion in a cloud-based scheduling component for exercises with deadlines. It develops the current inlaying calculation by tending to clashes among comparable rents and permitting the scheduling of new deadline-delicate undertakings during execution. A typical rent acknowledgment proportion of 91.94% and an insignificant normal rent rejection proportion of 8.05% are achieved by the proposed instrument. It thinks about the current time (CT) and whole time (GT) as scheduling boundaries, which are not viewed as in existing works. It permits the appearance of another rent to be planned and doesn't need a leader like AHP to determine clashes among comparative leases. Be that as it may, the exhibition investigation depends on a set number of workloads and boundaries, which may not completely address the assorted necessities of various cloud applications. In 2024, S. Mangalampalli, et., al., [26] recommended a deep reinforcement learning based task-scheduling calculation in cloud computing, planning to optimize critical Nature of Administration (QoS) boundaries like energy utilization, time, and SLA infringement. The calculation centers around tending to boundaries like makespan, SLA infringement, and energy utilization in cloud computing. It works out the execution time of errands in light of a dynamic limit esteem and the heap on actual hosts. Use of hosts is resolved utilizing explicit conditions. Notwithstanding, the review misses the mark on definite conversation on the limits of the proposed calculation, which might have given experiences into its possible disadvantages and regions for development. In 2023, R. Mishra and M. Gupta, [27], introduced a review that looks at heuristic algorithms for scheduling undertakings in cloud computing, like DRALBA, RALBA, DLBA, min, Max-min, and Cooperative effort, in view of execution boundaries like makespan, throughput, and normal resource usage proportion (ARUR). The current DRALBA approach outflanks other methodologies as far as execution boundaries, both on synthetic and reasonable workloads, making it a productive and successful scheduling calculation for cloud computing conditions. In any case, there is no conversation in regards to the adaptability of the proposed algorithms and their exhibition with bigger workloads or in genuine cloud computing

conditions. In 2023, P. Banerjee et., al., [28] presented MTD-DHJS, a clever undertaking scheduling calculation for cloud computing. It leverages the Johnson Sequencing technique to diminish makespan and upgrade resource use. By representing position interdependencies and waiter handling times, the calculation showed striking improvements in makespan decrease and resource usage over current scheduling techniques through careful recreations. Be that as it may, the review didn't unequivocally specify the information utilized in the reenactment arrangement and examination.

In this research, the study aims to improve the evaluation of cybersecurity measures and intrusion cases for virtual cloud network nodes by using the characteristics of deep learning techniques. This article also addresses the problems of data flow protocols to cloud network nodes, improving the efficiency of achieving security requirements and reducing the error rate

3. Methodology

In this section, the important design steps for implementing the proposed security model for systems that achieve security through the use of artificial intelligence (deep learning) techniques will be explained. In this investigation, we relied on an approved website for processing information, namely (kaggle.com), in addition to re-representing some of the information using useful elements of the MATLAB program. Also, the details of the design and preparation of the layers and functions of the deep learning algorithm used in this study to verify security standards in computerized cloud networks are reviewed.

3.1 The Implemented Dataset

In this study, the data set is chosen so that it represents the states of effective cloud network nodes, which indicate normal or abnormal behavior of the information flow through a sequence assigned to each weight value in the network nodes. The dataset was downloaded from kaggle.com and modified and initialized using efficient MATLAB application functions. Table 1 shows a sample of the modified dataset used to identify anomalies to classify the security levels of cloud nodes.

Table 1: The modefied data set description from Kaggle.com, & GitHub.com web sites

Cloud Network Nodes								
N_1	**N_2**	**N_3**	**N_4**	**N_5**	**N_6**	**N_7**	**N_8**	**Network Case**
0	**1**	**1**	**1**	**1**	**1**	**1**	**0**	Normal=1
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	Normal =1
0	**1**	**0**	**0**	**1**	**0**	**1**	**0**	Ubnormal=0
0	**1**	**1**	**1**	**1**	**0**	**1**	**1**	Normal =1
0	1	**1**	**1**	**1**	**1**	**0**	**1**	Normal =1
1	**0**	**0**	**0**	**0**	**0**	**0**	1	Ubnormal =0
1	**0**	**0**	1	**1**	**0**	**1**	**0**	Ubnormal=1
1	1	**0**	**1**	**0**	**1**	**1**	**0**	Normal =1
0	**0**	**0**	**0**	**0**	**0**	**0**	1	Ubnormal =0
0	**1**	**0**	**1**	**0**	**1**	0	1	Ubnormal =0
0	**0**	**1**	**0**	**0**	**1**	**0**	**1**	Ubnormal =0
0	**0**	**1**	**1**	**1**	**1**	**1**	1	Normal=1
1	**0**	**0**	**0**	**1**	**1**	**0**	1	Ubnormal =0
1	**1**	**0**	**0**	**0**	**0**	**1**	1	Normal=1
0	**1**	**0**	**0**	**0**	**0**	**0**	1	Ubnormal =0
0	**0**	**0**	**0**	**1**	**0**	**1**	1	Ubnormal =0
1	**1**	**0**	**1**	**1**	**1**	**1**	**0**	Normal=1
1	**1**	**1**	**1**	**1**	**1**	**1**	**0**	90=1
0	**1**	**1**	**0**	**1**	**0**	**0**	**1**	Normal =1
0	**1**	**0**	**1**	**1**	**0**	**1**	**0**	Normal =1
1	**1**	**1**	**0**	**1**	**1**	**0**	**1**	Normal =1
1	**1**	**0**	**1**	**1**	**1**	**1**	**0**	Normal =1
0	**0**	**1**	**0**	**1**	**0**	**1**	**1**	Ubnormal =0
1	**0**	**0**	1	**1**	**1**	**1**	**0**	Normal =1

Such that;

1) Low Node Case=0, High Node Case=1,
2) If total nodes Sum >= 4 then Network Case=1, Normal data flow (Secure Flow),
3) If total nodes Sum < 4 then Network Case =0, Ubnormal data data (Anomally Flow).

Thus, and along the employing of the dataset shown in Table 1 above, the smart deep learning algorithm will be trained using artificial neural networks (ANNs) to learn its weights and update their values according to the anomalies and security cases shown for each node of the virtual cloud network.

a. Design of ANN Deep Learning Security Systems

In this part, the layers of the smart algorithm will be configured and designed to be compatible with the work and classifications of the data set, so that the data is divided into 70% allocated to the trained data and 15% for each of the test and verification data sets. Figure 6 shows the flow chart of the proposed model methodology plan.

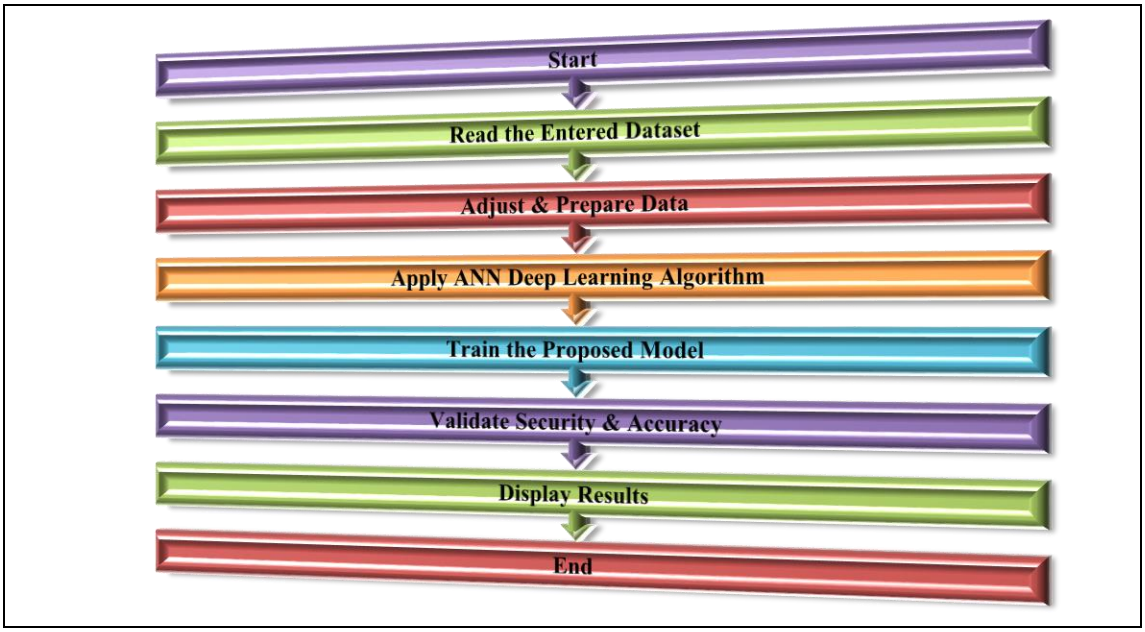


Figure 6. Flow chart of the proposed model methodology plan.

As one could notice from the suggested schemodelme flow chart displayed in Figure 5, the proposed scheme will combine the important information data stream and adjust their classes. There will be an initialization and preparation process for the input data set to organize its flow, examine it, and divide it according to the number of nodes of the virtual cloud network. This is followed by applying the proposed smart algorithm and defining the input layers and internal layers along with the output layers and supporting functions. The data values and design settings for each of the inputs and details of the smart algorithm will also be specified to complete the required software design mechanism. The hidden intelligent algorithm layers are then trained according to the states of the cloud nodes to match them. After that, the extent of eliminating anomalies and the level of efficiency of the proposed model in achieving cloud network security levels are verified. Finally, the training results and performance efficiency measures are presented. Also, the block diagram of the deep learning data security model is displayed in Figure 7.

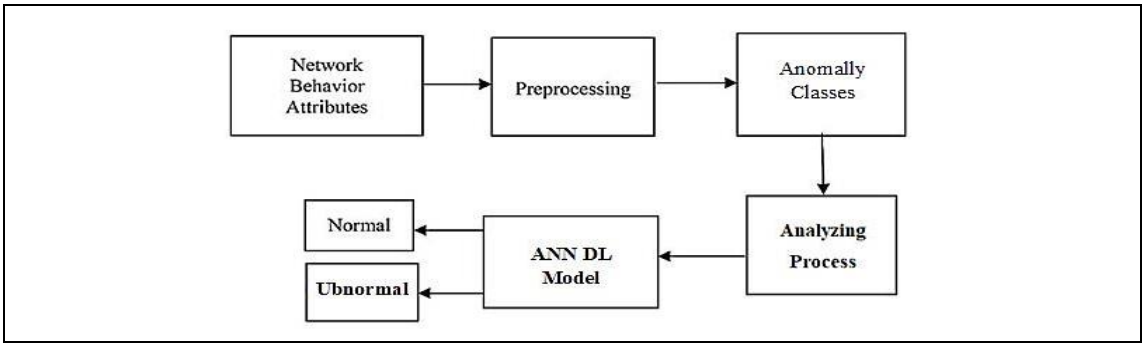


Figure 7. Block diagram of the DL data security model.

The candidate software model describes the operation of an artificial neural network (ANN) algorithm based on a training data set with an input (N = 200 tests) to be characterized to train the algorithm parameters to the point where security rates are appropriate and an appropriate MSE rate is reached. To understand the artificial neural network algorithm tool, the data information represents the main security standards that are sent across the nodes of the computer network. This data is propagated by collecting it into the input layer of the ANN algorithm so that it is mapped and ready for training. This data is then processed in classification techniques by providing it with the basic fingerprint and then changing it by examining more anomalies according to the basic description to enhance the classification precedence. The parameters of the artificial neural network algorithm (neurons) are trained, and their weights are updated by comparing the algorithm outputs with the original data set and finding error streams and anomalies. The resulting amplitude gains are then reduced by passing along the Relu layer, and these quantities are then summed in the pooling layer. After these procedures, the data is processed in the internal weight layers to prepare and update the algorithm layers to find the results and deduce the amount of error and the best match. Figure 8 presents the design simulation diagram of the proposed deep learning ANN algorithm.

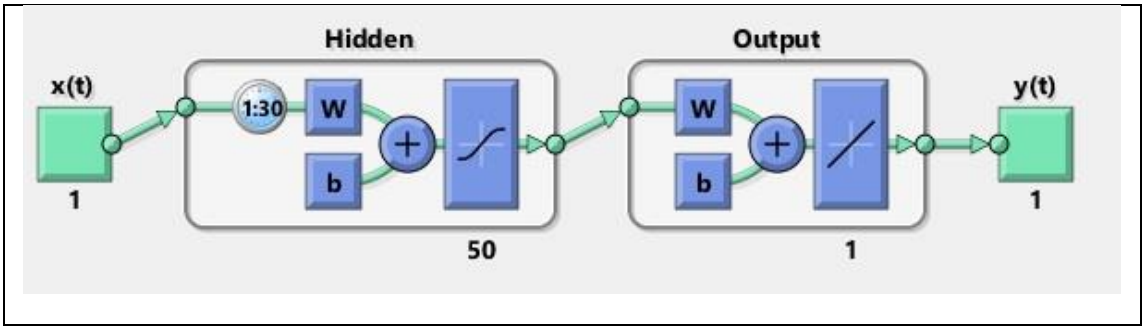


Figure 8. The design simulation diagram of the proposed deep learning ANN algorithm.

As one might recognize from Figure 8, the suggested ANN simulated model will contain three layers input, hidden, and output layers respectively with 50 neurons as weight counts inside the internal hidden layer. Also, their will be bias adjusting parameters employed to compensate any small drift in the results values. Activation and summing units will be added to complete the deep learning design operation. Finally, the settings and control parameters of the proposed ANN algorithm are shown in Table 2.

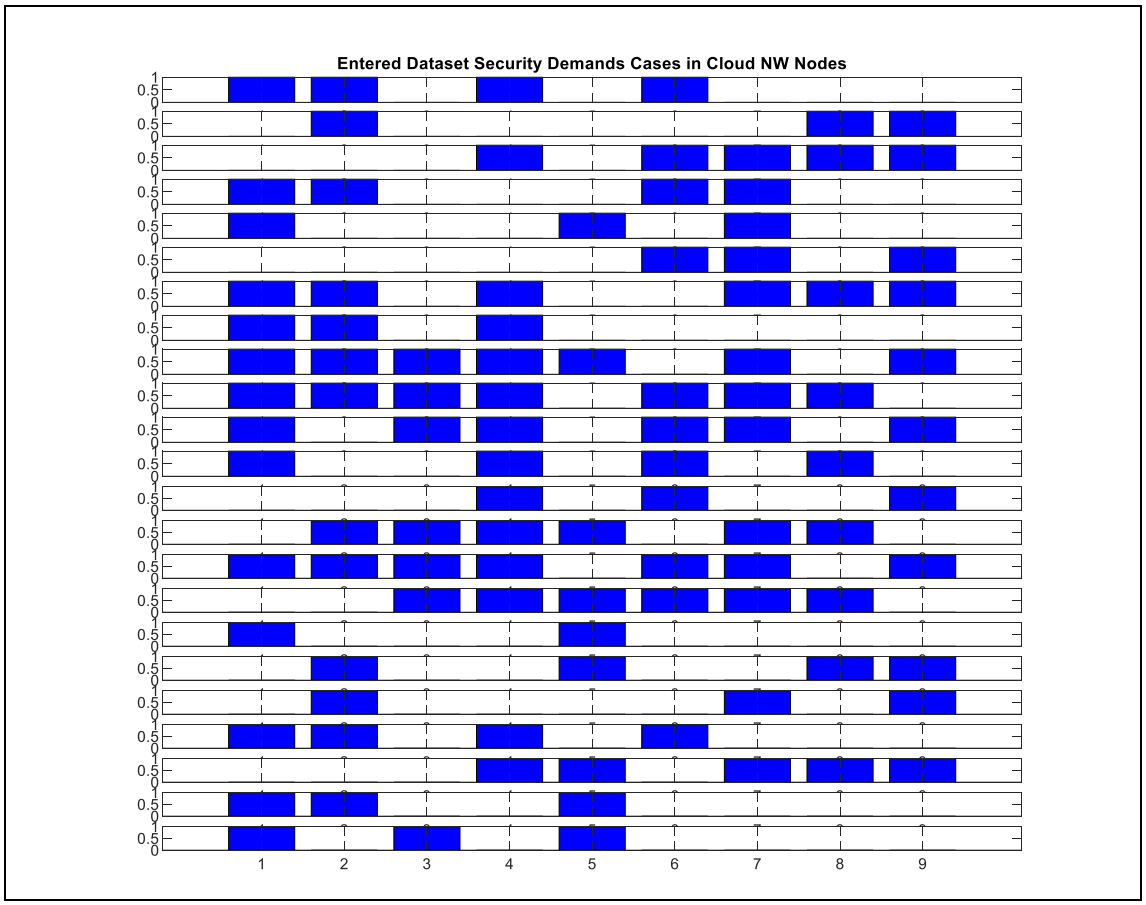
Table 2: The design settings of the proposed ANN model.

Neurons No.	(Training, Testing, Validation) Ratios	Training Function	Max Epoch Counts
10	(0.70, 0,15, 0.15)	Levenberge Marquardt	10-50
30	(0.70, 0,15, 0.15)	Levenberge Marquardt	50-100
50	(0.70, 0,15, 0.15)	Levenberge Marquardt	100-500

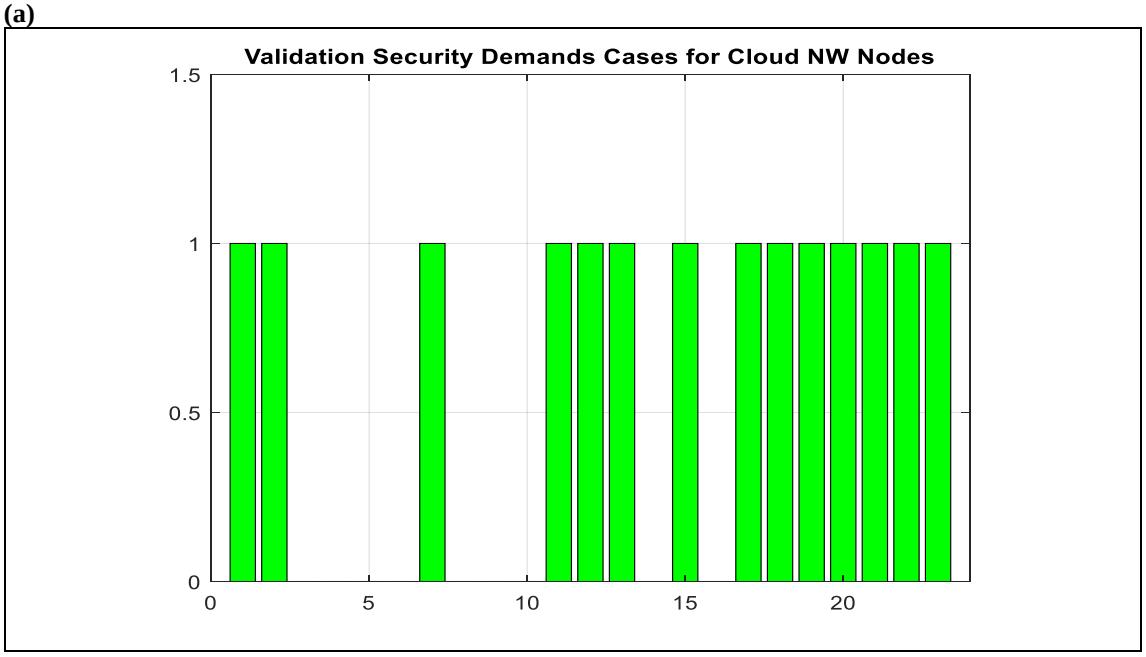
According to the design specifications shown in Table 2, the structure of the proposed DL-ANN algorithm will be used through the MATLAB program, which will be applied to simulate the operation of the security states classification model in cloud networks and calculate its efficiency.

4. Results & Discussions

In this part, the proposed security model was implemented and tested to achieve the security requirements of the virtual cloud network with the help of artificial intelligence (deep learning) algorithms using MatLab2020b m. Text records. The states of the cloud network nodes will also be represented by the data set inputted and modified using MATLAB desktop functions. This model shows the conditions for achieving security conditions for private nodes in the cloud network and the mechanism for achieving this condition as shown in Figure 9.



(a)

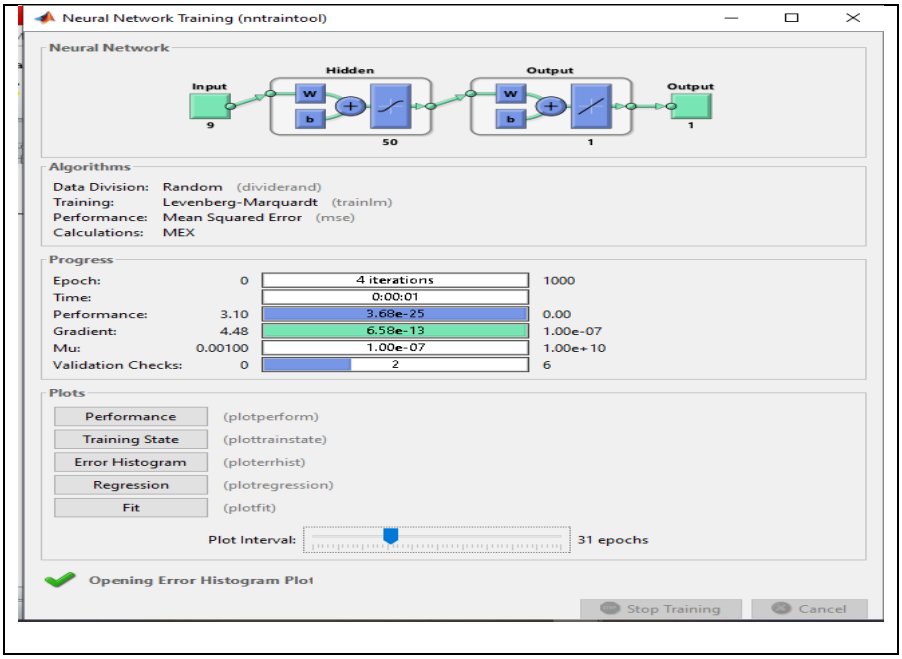


(b)

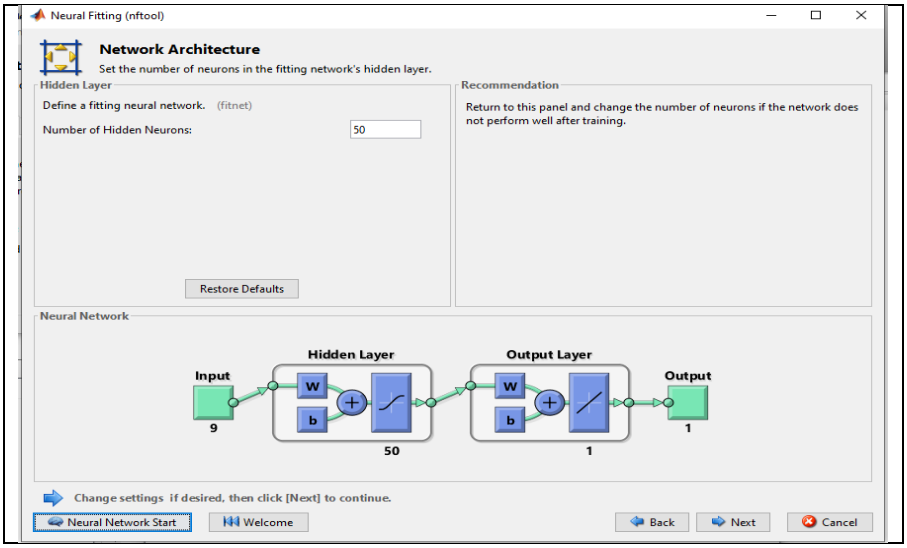
Figure 9. Demonstration of the entered dataset conditions for achieving security demands for private nodes in the cloud computing network, (a) Nodes conditions, (b) Validation cases.

Looking at Figure 9(a), it can be seen that the required security conditions for cloud network nodes are met in the form of numerical values of zeros and ones that determine whether the data flow is normal or abnormal. Also, Figure 9(b) shows cases of stability of security requirements along the set of network nodes for all tested cases. In fact, the entered dataset to the proposed ANN algorithm have been divided into 70% trained samples with 15% for each tested and validated samples.

After that, testing the deep learning technique is carried out by creating an ANN algorithm by training the input data set shown in Figure 9(a) to achieve the required security states for the network nodes shown in Figure 9(b). The training results of the implemented smart algorithm are shown in Figure 10.



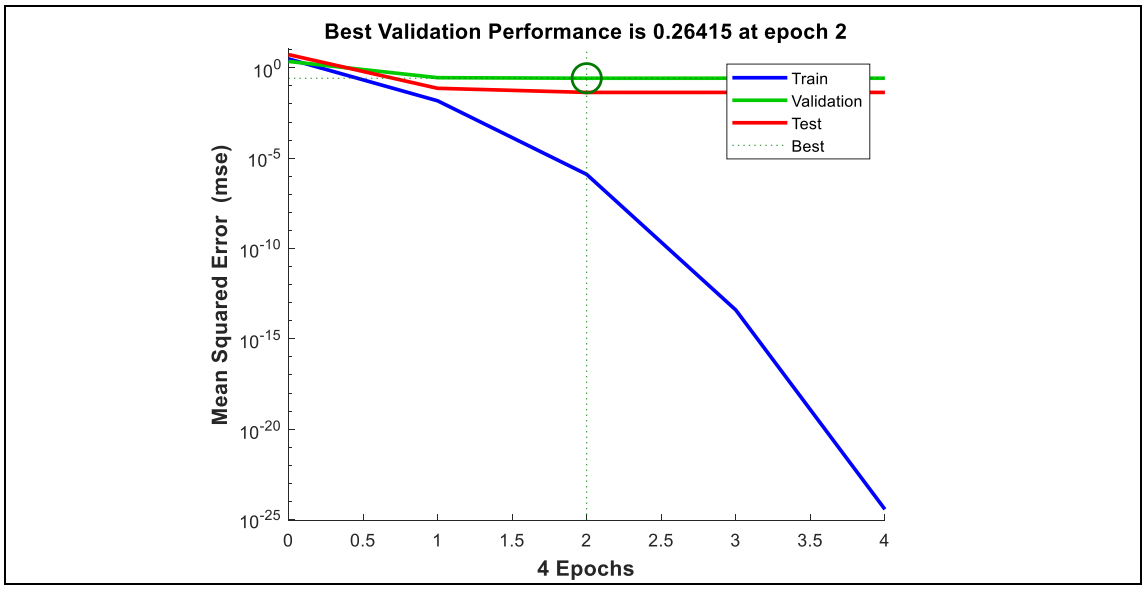
(a)



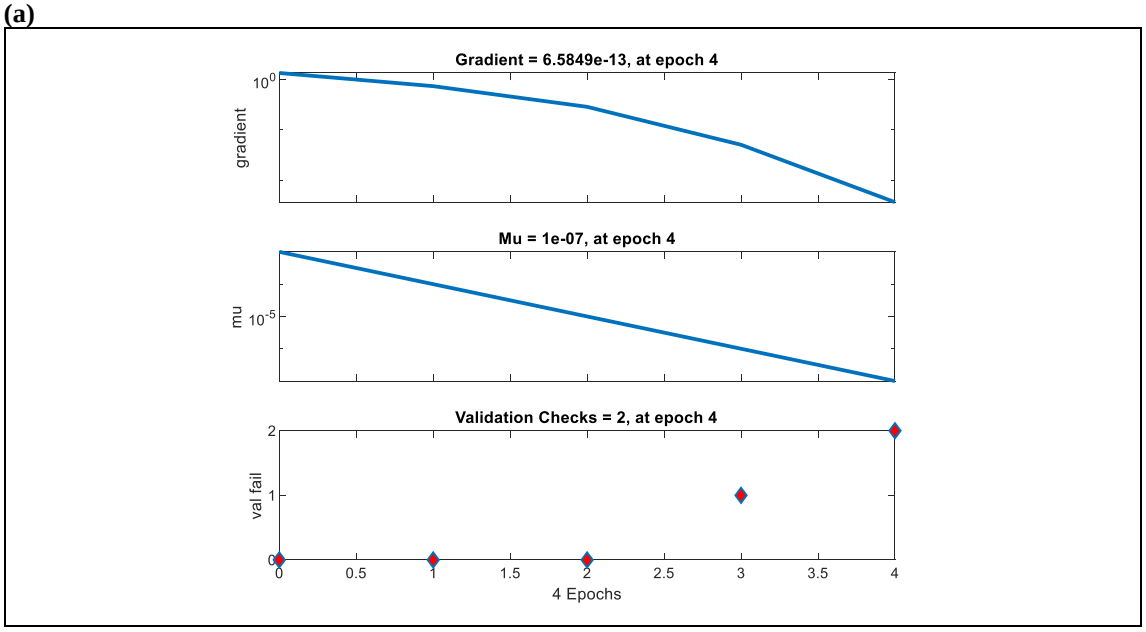
(b)

Figure 10. The training results of the implemented DL-ANN algorithm; (a) Neural Network training details, (b) The obtained DL_ANN architecture model.

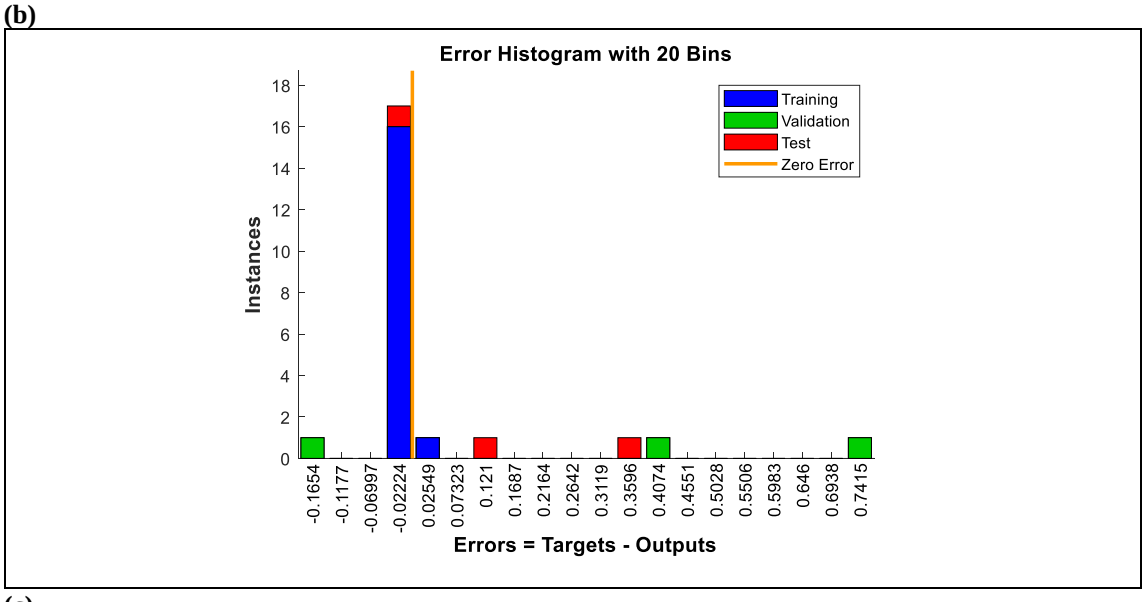
By looking at Figure 9 above, one could observe the training details of the proposed ANN deep learning algorithm to achieve the security conditions of the virtual cloud network in this research. Figure 9(a) shows the training statistics for the smart algorithm by finding measures of the performance error ratio, the amount of the gradient ratio, and the number of steps (epochs). Figure 9(b) also shows the details of the internal structure architecture of the smart algorithm, which appears with a number of layers of 3 and 50 hidden weights (Neuron). Moreover, the resulting performance metrics of the trained DL-ANN algorithm have been displayed as in Figure 11.


Best Validation Performance is 0.26415 at epoch 2
Mean Squared Error (mse)
4 Epochs
Train
Validation
Test
Best

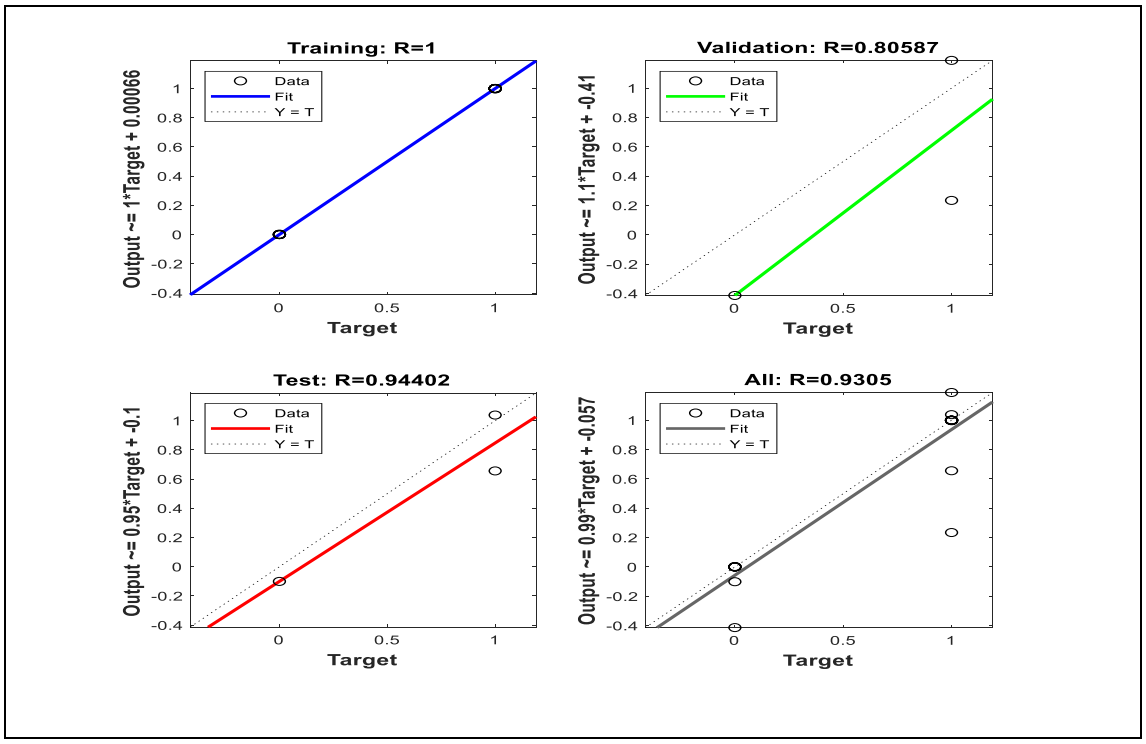
(a)


Gradient = 6.5849e-13, at epoch 4
gradient
Mu = 1e-07, at epoch 4
mu
Validation Checks = 2, at epoch 4
val fail
4 Epochs

(b)


Error Histogram with 20 Bins
Training
Validation
Test
Zero Error
Instances
Errors = Targets - Outputs

(c)



(d)

Figure 12. The performance metrics of the trained ANN DL model, (a) MSE performance, (b) Error histogram, (c) Training states, and (d) Regression (ROC) metrics.

Regarding the results achieved in Figure 10(a), it could be observed that the achieved DL-ANN model produces recorded MSE readings of 0.26415 as performance error over 2 epochs for all tested, trained and validated samples. Likewise, and concerning the results presented in Figure 10(b), a regression value of 6.585*10-13 could be observed at 4 epochs with the approval check measurement similarly evaluated and recording a value of Mu=1*10-7 and validation check of 2 at 4 epoch life cycle tests, which also ensures excellent training. at the proposed algorithm. Moreover, and as introduced in Figure 10.(c), low positive values could be observed for the error histogram records with only 20 bins and maximum error cases of -0.02224 in 18 test cycles. At last, the achieved results displayed in Figure 10(d) provide reasonable regression estimates (also called Receiver Observation Curve "ROC") for training, validation, and testing samples. All regressions were calculated and showed a slope (R = 0.9305) with perfect alignment of the target center with the outcome information. The results of regression estimates (ROC) calculations indicate the extent of correspondence between the values of the models obtained after the training process of the input data set with the values of the target or validation models. The reading of the ROC value = 0.9503 indicates a high match and the success of training the proposed deep learning algorithm on the data set to achieve security requirements. Finally, Table 3 shows a summary of the final results of the training metrics achieved using the proposed deep learning algorithm to achieve cloud network security conditions.

Table 3: Summary of the final results of the training metrics achieved using the proposed deep learning algorithm to achieve cloud network security conditions.

Metrics	Train	Validate	Test	Total	Iterations (Epochs)
MSE	0.26415	0.26415	0.26415	0.26415	2
Gradiant, **Mu,** **Validation Check**	6.585*10-13 1*10-7 2	6.585*10-13 1*10-7 2	6.585*10-13 1*10-7 2	6.585*10-13 1*10-7 2	4 4 4
Error Histogram	-0.02224 20	-0.02224 20	-0.02224 20	-0.02224 20	 18
Regression (ROC)	1	0.80508	0.94402	0.9503	4

Finally, performance evaluation metrics can be found for the efficiency of the work of the proposed ANN algorithm, which are shown in Table 4.

Table 4: The accomplished ANN performance metric results.

Metric Values		Accuracy	Specificity	Sensitivity	Precision	F score
TP	0.99745	98.7605%	99.739%	97.81894%	99.745%	98.7725%
TN	0.97776					
FP	0.00255					
FN	0.02224					

Regarding the results obtained in Table 3, it could be concluded that the proposed DL ANN algorithm used in conducting security verification operations for cloud computer networks has provided a performance efficiency with an accuracy of 98.76% with a detection accuracy of 99.745, which are very excellent results.

5. Conclusion

In this study, a proposal was presented for a model of DL algorithms used in achieving network security, such as intrusion detection systems and cyber protection systems for cloud computing networks. The requirements for achieving cybersecurity were designed and implemented using artificial intelligence techniques while applying the ANN deep learning algorithm methodology. Also, cybersecurity measures and intrusions of virtual cloud network nodes to further detect and block strange and intrusive flows were evaluated in this study with high efficiency using the proposed technique. The training and analysis results of the proposed DL-ANN algorithm model were achieved, showing an error performance of 0.26415, with gradiant of 6.585*10-13 , and an error correlation of 1*10-7 for only 4 epoches. Likewise, the regression metric, or ROC, scores between target and outcome and expected gains for all evaluation tests were 95.03%. Finally, and along the results of examining and testing the proposed model for improving network security requirements, it has been concluded that the proposed DL ANN algorithm used in conducting security verification operations for cloud computer networks has provided a performance efficiency of 98.76% accuracy with a detection accuracy of 99.745, with a relatively low error rate of 0.255%, which are very excellent results.

Funding: “This research received no external funding”

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] Agarwal, N.; Hussain, S. Z. (2018). A Closer Look at Intrusion Detection System for Web Applications. Security and Communication Networks, 2018, pp 1-27.

[2] Manthiramoorthy, C., Khan, K. M. S., & A, N. A. (2023). Comparing Several Encrypted Cloud Storage Platforms. International Journal of Mathematics, Statistics, and Computer Science, 2, 44–62. https://doi.org/10.59543/ijmscs.v2i.7971.

[3] Somani, G.; Gaur, M. S.; Sanghi, D.; Conti, M.; Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications 107, pp. 30-48..

[4] Firas Mahdi Muhsin Al-Salbi, “Investigation of QoS Multicast Routing Based on Intelligent Multiple Constrained”, www.ccsenet.org/cis Computer and Information Science Vol. 4, No. 4; July 2011, 64 ISSN 1913-8989 E-ISSN 1913-8997

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

[6] Seyed Mohammad Mousavi and Marc St-Hilaire, (2019). Early detection of DDoS attacks against SDN controllers. In Computing, Networking and Communications (ICNC), 2019 International Conference on, pages 77–81. IEEE.

[7] Jun Li, (2018). Drawbridge: software-defined ddos-resistant traffic engineering. In ACM SIGCOMM Computer Communication Review, volume 44, pages 591–592. ACM.

[8] Li, S. H., Kao, Y. C., Zhang, Z. C., Chuang, Y. P., & Yen, D. C. (2015). A network behavior-based botnet detection mechanism using PSO and K-means, ACM Transactions on Management Information Systems (TMIS), 6(1), 3.

[9] Prasad, K. M., Reddy, A. R. M., & Rao, K. V. (2018). DoS and DDoS attacks: defense, detection and traceback mechanisms-a survey. Global Journal of Computer Science and Technology.
[10] Aggarwal, A., & Gupta, A. (2015). Survey on data mining and IP traceback technique in DDoS attack. International Journal of Engineering and Computer Science, 4(06).
[11] Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications, 107, 30-48.
[12] Kostas Giotis, et. al., (2017). Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments. Computer Networks, 62:122–136, 2017.
[13] Z. Zhu, J. Peng, K. Liu, and X. Zhang, "A game-based resource pricing and allocation mechanism for profit maximization in cloud computing," Soft comput, vol. 24, no. 6, pp. 4191–4203, Mar. 2020, doi: 10.1007/s00500-019-04183-0.
[14] M. Manavi, Y. Zhang, and G. Chen, "Resource Allocation in Cloud Computing Using Genetic Algorithm and Neural Network," Aug. 2023, [Online]. Available: http://arxiv.org/abs/2308.11782
[15] N. Swatthong and C. Aswakul, "Optimal cloud orchestration model of containerized task scheduling strategy using integer linear programming: Case studies of iotcloudserve@tein project," Energies (Basel), vol. 14, no. 15, Aug. 2021, doi: 10.3390/en14154536.
[16] M. Yadav and A. Mishra, "An enhanced ordinal optimization with lower scheduling overhead based novel approach for task scheduling in cloud computing environment," Journal of Cloud Computing, vol. 12, no. 1, Dec. 2023, doi: 10.1186/s13677-023-00392-z.
[17] K. Y. Tai, F. Y. S. Lin, and C. H. Hsiao, "An Integrated Optimization-Based Algorithm for Energy Efficiency and Resource Allocation in Heterogeneous Cloud Computing Centers," IEEE Access, vol. 11, pp. 53418–53428, 2023, doi: 10.1109/ACCESS.2023.3280930.
[18] M. S. Al-Asaly, M. A. Bencherif, A. Alsanad, and M. M. Hassan, "A deep learning-based resource usage prediction model for resource provisioning in an autonomic cloud computing environment," Neural Comput Appl, vol. 34, no. 13, pp. 10211–10228, Jul. 2022, doi: 10.1007/s00521-021-06665-5.
[19] M. Ghobaei-Arani and A. Souri, "LP-WSC: a linear programming approach for web service composition in geographically distributed cloud environments," Journal of Supercomputing, vol. 75, no. 5, pp. 2603–2628, May 2019, doi: 10.1007/s11227-018-2656-3.
[20] F. Yao, C. Pu, and Z. Zhang, "Task duplication-based scheduling algorithm for budget-constrained workflows in cloud computing," IEEE Access, vol. 9, pp. 37262–37272, 2021, doi: 10.1109/ACCESS.2021.3063456.
[21] P. S. Rawat, P. Dimri, S. Kanrar, and G. P. Saroha, "Optimize Task Allocation in Cloud Environment Based on Big-Bang Big-Crunch," Wirel Pers Commun, vol. 115, no. 2, pp. 1711–1754, Nov. 2020, doi: 10.1007/s11277-020-07651-1.
[22] W. Shi, D. Tang, and P. Zou, "Research on cloud enterprise resource integration and scheduling technology based on mixed set programming," Tehnicki Vjesnik, vol. 28, no. 6, pp. 2027–2035, Nov. 2021, doi: 10.17559/TV-20210718091658.
[23] L. Zhu, F. Wu, Y. Hu, K. Huang, and X. Tian, "A heuristic multi-objective task scheduling framework for container-based clouds via actor-critic reinforcement learning," Neural Comput Appl, vol. 35, no. 13, pp. 9687–9710, May 2023, doi: 10.1007/s00521-023-08208-6.
[24] L. Hamid, A. Jadoon, and H. Asghar, "Comparative analysis of task level heuristic scheduling algorithms in cloud computing," Journal of Supercomputing, vol. 78, no. 11, pp. 12931–12949, Jul. 2022, doi: 10.1007/s11227-022-04382-x.
[25] S. C. Nayak, S. Parida, C. Tripathy, and P. K. Pattnaik, "An enhanced deadline constraint based task scheduling mechanism for cloud environment," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 2, pp. 282–294, Feb. 2022, doi: 10.1016/j.jksuci.2018.10.009.
[26] S. Mangalampalli, G. R. Karri, M. Kumar, O. I. Khalaf, C. A. T. Romero, and G. M. A. Sahib, "DRLBTSA: Deep reinforcement learning based task-scheduling algorithm in cloud computing," Multimed Tools Appl, vol. 83, no. 3, pp. 8359–8387, Jan. 2024, doi: 10.1007/s11042-023-16008-2.
[27] R. Mishra and M. Gupta, "Cloud Scheduling Heuristic Approaches for Load Balancing in Cloud Computing," in 2023 6th International Conference on Information Systems and Computer Networks, ISCON 2023, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ISCON57294.2023.10112056.
[28] P. Banerjee et al., "MTD-DHJS: Makespan-Optimized Task Scheduling Algorithm for Cloud Computing with Dynamic Computational Time Prediction," IEEE Access, vol. 11, pp. 105578–105618, 2023, doi: 10.1109/ACCESS.2023.3318553.
[29] T. Dokeroglu, T. Kucukyilmaz, and E. G. Talbi, "Hyper-heuristics: A survey and taxonomy," Comput Ind Eng, vol. 187, Jan. 2024, doi: 10.1016/j.cie.2023.109815.
[30] Q.-Z. Xiao, J. Zhong, L. Feng, L. Luo, and J. Lv, "A Cooperative Coevolution Hyper-Heuristic Framework for Workflow Scheduling Problem."

[31] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke, “Recent advances in selection hyper-heuristics,” Sep. 01, 2020, Elsevier B.V. doi: 10.1016/j.ejor.2019.07.073.
[32] T. M. Shami, D. Grace, A. Burr, and P. D. Mitchell, “Single candidate optimizer: a novel optimization algorithm,” Evol Intell, Apr. 2022, doi: 10.1007/s12065-022-00762-7.
[33] G. R. Raidl, J. Puchinger, C. Blum, G. R. Raidl, J. Puchinger, and C. Blum, “Metaheuristic Hybrids.”
[34] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, “A survey on new generation metaheuristic algorithms,” Comput Ind Eng, vol. 137, Nov. 2019, doi: 10.1016/j.cie.2019.106040.
[35] K. G. Krishnasamy, K. Periasamy, P. M. Veerappan, G. Thangavel, R. Lamba, and S. Muthusamy, “A Pair-Task Heuristic for Scheduling Tasks in Heterogeneous Multi-Cloud Environment,” 2022, doi: 10.21203/rs.3.rs-1903846/v1.
[36] J. Chen, T. Du, and G. Xiao, “A multi-objective optimization for resource allocation of emergent demands in cloud computing,” Journal of Cloud Computing, vol. 10, no. 1, Dec. 2021, doi: 10.1186/s13677-021-00237-7.
[37] M. Ibrahim, S. Nabi, A. Baz, N. Naveed, and H. Alhakami, “Towards a task and resource aware task scheduling in Cloud Computing: An experimental comparative evaluation,” International Journal of Networked and Distributed Computing, vol. 8, no. 3, pp. 131–138, Jun. 2020, doi: 10.2991/ijndc.k.200515.003.
[38] H. Ben Alla, S. Ben Alla, A. Ezzati, and A. Touhafi, “A novel multiclass priority algorithm for task scheduling in cloud computing,” Journal of Supercomputing, vol. 77, no. 10, pp. 11514–11555, Oct. 2021, doi: 10.1007/s11227-021-03741-4.
[39] H. Hamzeh, S. Meacham, K. Khan, K. Phalp, and A. Stefanidis, “MRFS: A Multi-resource Fair Scheduling Algorithm in Heterogeneous Cloud Computing,” in Proceedings - 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC 2020, Institute of Electrical and Electronics Engineers Inc., Jul. 2020, pp. 1653–1660. doi: 10.1109/COMPSAC48688.2020.00-18.
[40] N. A. Almojel and A. E. S. Ahmed, “Tasks and Resources Allocation Approach with Priority Constraints in Cloud Computing,” International Journal of Grid and High Performance Computing (IJGHPC), vol. 14, no. 1, pp. 1–17, 2022.
[41] A. Brandwajn and T. Begin, “First-come-first-served queues with multiple servers and customer classes,” Performance Evaluation, vol. 130, pp. 51–63, Apr. 2019, doi: 10.1016/j.peva.2018.11.001.
[42] Saudi Computer Society., Institute of Electrical and Electronics Engineers. Saudi Arabia Section, Institute of Electrical and Electronics Engineers. Region 8, and Institute of Electrical and Electronics Engineers, 2nd International Conference on Computer Applications & Information Security (ICCAIS’ 2019) : 01-03 May, 2019 Riyadh, Kingdom of Saudi Arabia.
[43] D. Alsadie, “Virtual Machine Placement Methods using Metaheuristic Algorithms in a Cloud Environment-A Comprehensive Review,” IJCSNS International Journal of Computer Science and Network Security, vol. 22, no. 4, doi: 10.22937/IJCSNS.2022.22.4.19.
[44] H. Hamzeh, S. Meacham, K. Khan, K. Phalp, and A. Stefanidis, “MRFS: A Multi-resource Fair Scheduling Algorithm in Heterogeneous Cloud Computing,” in Proceedings - 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC 2020, Institute of Electrical and Electronics Engineers Inc., Jul. 2020, pp. 1653–1660. doi: 10.1109/COMPSAC48688.2020.00-18.
[45] N. A. Almojel and A. E. S. Ahmed, “Tasks and Resources Allocation Approach with Priority Constraints in Cloud Computing,” International Journal of Grid and High Performance Computing (IJGHPC), vol. 14, no. 1, pp. 1–17, 2022.
[46] A. Brandwajn and T. Begin, “First-come-first-served queues with multiple servers and customer classes,” Performance Evaluation, vol. 130, pp. 51–63, Apr. 2019, doi: 10.1016/j.peva.2018.11.001.
[47] Saudi Computer Society., Institute of Electrical and Electronics Engineers. Saudi Arabia Section, Institute of Electrical and Electronics Engineers. Region 8, and Institute of Electrical and Electronics Engineers, 2nd International Conference on Computer Applications & Information Security (ICCAIS’ 2019) : 01-03 May, 2019 Riyadh, Kingdom of Saudi Arabia.
[48] D. Alsadie, “Virtual Machine Placement Methods using Metaheuristic Algorithms in a Cloud Environment-A Comprehensive Review,” IJCSNS International Journal of Computer Science and Network Security, vol. 22, no. 4, doi: 10.22937/IJCSNS.2022.22.4.19.