ONLINE ISSN: 3007-3197 | PRINT ISSN: 3007-3189 http://amresearchreview.com/index.php/Journal/about

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 5 (2025)

Optimized Artificial Neural Network-based Approach for Task Scheduling in Cloud Computing

[1]Asif Ali, [2]Sophia Ajaz, [3]Syed Muhammad Daniyal

Article Details

Keywords: Cloud Computing, Task Scheduling, Optimization, Artificial Neural Network

Asif Ali
Faculty of Engineering Science and Technology, Iqra University Karachi, Pakistan
asifali@iqra.edu.pk

Sophia Ajaz
Faculty of Engineering Science and Technology, Iqra University Karachi, Pakistan
sophia.ajaz@iqra.edu.pk

Syed Muhammad Daniyal
Faculty of Engineering Science and Technology, Iqra University Karachi, Pakistan
syed.daniyal@iqra.edu.pk

ABSTRACT

Cloud computing has transformed how businesses utilize computing resources. The fundamental aim remains to convene distributed resources such that they operate most efficiently and thus improve overall throughput. This shift is proving useful in developing complex computations on a large scale in commercial practice. The best feature of commercial clouds is their elasticity, which recommends dynamic resource allocation adjustments by users accordingly to real-time demand. Alongside the pay-as-you-go cost model, this ensures that organizations optimise usage and costs. However, good allocation of resources is one of the biggest challenges, known as job scheduling. This refers to how to optimally assign end-user requests to cloud resources, based on the premise that each task can run as quickly as possible. A job scheduler's main objective is to select the best resource to satisfy a user's task, taking into consideration statistical data and dynamic parameters of user jobs. For this problem, various techniques have been researched by the investigators, with AI playing a major role. Among these techniques, the use of genetic algorithms and ant colony optimization techniques is employed to allocate resources optimally, improving the efficiency of job scheduling in the cloud. This research is geared towards continually improving job scheduling in cloud computing. The introduction of artificial neural networks represents an exciting avenue for enhancing optimization and addressing the ongoing challenges faced in efficiently allocating cloud resources.

INTRODUCTION

In cloud computing, resources and services are given access over a network and can be conveniently managed with minimal effort, as depicted in Fig. 1 [1]. The term cloud is also used to describe some smart things, which can revolutionize the world and change IT infrastructures to become commodities out there for third-party and client consumption [2]. With a cloud computing environment, computing power or resources can be outsourced to another party for usage over the Internet. Emerging-age technology is taking away the power and data from individual private personal computer systems and portable devices to gigantic data centers. More importantly, end users access and use the services fully independently of, or without knowledge of, the physical location and configuration of the system at the provider's site [3]. Over the last few years, cloud computing has taken one of its hottest positions, crowded with the new face of internet technology, with huge providers like Amazon, Google, and Microsoft setting up powerful data centers spread across different corners of our world, making the efficient usage of these resources a concern among different users.

Every cloud computing application, scalable infrastructure, and platforms are made available to carry out consumers' tasks through pay-as-you-go pricing models for accessing resources. Service level agreements between providers and consumers guarantee improvements in service and availability of resources. In line with energy consumption, the cloud computing paradigm corresponds to utility computing on the lines of IaaS, PaaS, and SaaS, where everything is paid-as-per-use on demand and service-level agreements apply. The infrastructure resources are brought together into a single unit in the data center, including storage, computing, and network resources. This unit comprises a package of virtual machines that can manage the operation independently. Virtualization technology is a key enabler in the efficient exploitation of an elastic cloud infrastructure [4].

Cloud computing facilitates computation, storage, and bandwidth on a significantly larger scale for a virtual machine. Hence, assigning the task efficiently to the virtual machine is necessary for proper resource management. Scheduling methods include various approaches as static and dynamic methods, aided with additional nature-inspired heuristics for the generation of the best schedule. Time-aware multi-objective performance metrics are used for determining the efficiency of scheduling, which considers that poor resource management leads to almost unbearable operational costs due to the execution time of these cloud resources. Much of the present research has concentrated on nature-inspired and AI-based scheduling methods. Although meta-heuristic approaches might take a longer time than deterministic algorithms, they encounter other challenges such as high execution overhead and the inability to cope with the characteristics of dynamic and scalable cloud scenarios. The present work overcomes the above-mentioned issues by incorporating a neural-based feature classifier. The classification potential of ANNs helps solve scheduling issues in the clouds. This scheduling method utilizes the classification power of intelligent computing paradigms. A neural model is proposed to learn the suitability of virtual machines for incoming requests. The network will be trained using datasets generated with genetic algorithms to avoid restrictions on the number of layers and perceptrons in each layer. The recursive neural network with minimal error rates proves effective in handling the complexity of cloud infrastructure, as shown in Fig. 1.

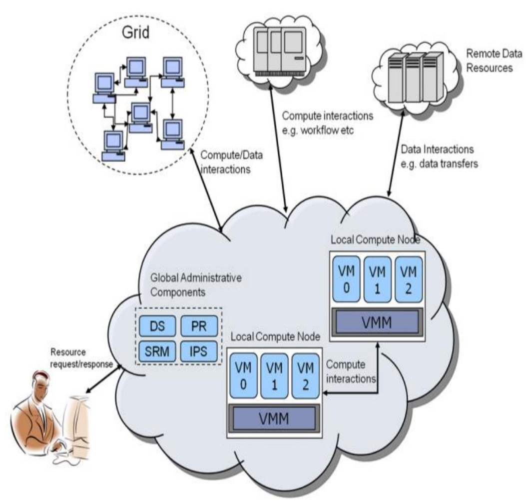

FIG. 1: COMPLEXITY OF CLOUD INFRASTRUCTURE

RELATED WORK

In recent times, there has been a lot of interest in making the scheduling of independent tasks on virtual machines more efficient. The main goals are to save time, cost, and energy, and make the process scalable and available. Different methods like static, dynamic, and meta-heuristic approaches have been tried to find the best solutions. While some machine learning techniques like Artificial Neural Networks (ANN) have been helpful, simpler methods often give suboptimal results, especially in energy-saving situations. A new ANN approach has been proposed to address these challenges and make scheduling decisions more efficient [5]. The combination of managing resources and using machine learning has been crucial in solving scheduling problems. Researchers have explored techniques like ant colony-based optimization and adaptive resource allocation to improve service quality by considering factors like time, fault tolerance, and completing tasks. Using bio-inspired techniques, like neural network models [6], has shown practical applications in making cloud systems work better. As the demand for artificial neural networks increases, researchers are looking into different ways to organize tasks on virtual machines using methods like static plans, dynamic changes, and smart approaches.

The research on scheduling optimization is broad and includes various methods to tackle the difficulties of task scheduling in large cloud environments [7]. Ongoing efforts involve exploring nature-inspired methods, incorporating machine learning, and evolving bio-inspired models to refine scheduling solutions continually. Different studies have demonstrated better service quality and efficient use of resources through smart resource allocation, ant colony optimization, genetic scheduling, and distributed scheduling. The proposed ANN technique is a promising way to make scheduling decisions better by using artificial neural networks, offering a potential path for future improvements in the field.

Within recent times, job scheduling has become a relevant issue within the scope of cloud computing research, especially in the investigation of the three important algorithms: Min-Min, Max-Min, and genetic algorithms. Furthermore, a new direction has come into play that aims at Min-Min and Max-Min being integrated within a genetic algorithm framework. The Min-Min algorithm starts by having all tasks unassigned, computes minimum completion times for each task, and selects the minimum for scheduling on an appropriate machine [8]. Then, the algorithm calculates execution times for the remaining tasks by adding them to the assigned task's time before removing that task from the list, continuing with that until all tasks are allocated to the available resources. Similarly, in the Max-Min algorithm, tasks are prioritized by maximum execution time on any resource, and similar iterations are made until all tasks are assigned. In cloud task scheduling, LBACO seeks to establish load balancing while achieving the minimum makespan, through simulated testing on the Cloud Sim Toolkit wherein LBACO performs better than FCFS and straightforward ACO [9]. In a different model, genetic algorithms were proposed for cost-based multi-QoS job scheduling in a cloud computing environment, which comprise some well-known crossover operators, such as PMX, OX, CX, and mutation operators, such as swap and insertion mutation, among others. The objectives are to develop an improved schedule for the better performance of the system as a whole [10]. It guarantees that the algorithm has provided the optimal solution in a limited time. Result analysis from experiments reveals that this method not only achieves the Quality of Service (QoS) requirements for customer jobs, but also guarantees maximum profit for cloud providers. Private cloud features for e-learning purposes pose a scheduling challenge for the workloads, which is optimized with a genetic algorithm. This optimization is performed with certain factors enforced by the underlying virtualization technology, including memory overcommitment and IOPS rate distribution.

Experiments have shown that genetic algorithms can be a valid alternate means of encouraging the effective use of both the Planned Scheduling Request and the One-Off Scheduling Request by ensuring more than optimal but uniform utilization of the cloud resources. On top of that, even their co-scheduled workloads have been found optimal concerning the workload profile by means of a genetic algorithm. The author [12] has presented an Improved Differential Evolution Algorithm (IDEA), proposed as a scheduling algorithm for task and resource allocation in cloud computing organizations. Taguchi has been incorporated in the IDEA to generate better offspring and improve the concept of the conventional Differential Evolution Algorithm (DEA). For economical performance, two models: the total minimized costs and time of task scheduling, have incorporated cost model processing and receiving costs with the time model accountable of receiving, processing, and waiting times. The developed algorithm is experimentally evaluated in scenarios of five tasks and five resources as well as in the environment of ten tasks and ten resources. Results reveal that in either situation, IDEA surpassed any alternate scheduling algorithm defined in literature (DEA/NSGA). This backing further benefits decision makers to comprehend the alternatives to select while contradictory objectives are brought into play. In a new direction, Palmerieriet presents a whole distributed scheduling framework that operates on entirely uncoordinated federated cloud environments. This system induces an implicit coordination by setting a small cost on any agent's behavior. Effectiveness in this case has been simulated in an emulation environment where service providers, agents, and the associated protocols operate in a cloud setting [13].

An experimental assessment concludes that the proposed approach exhibits both good scalability and quality and achieves optimum completion time for performance. Particularly beneficial in large-scale cloud environments where many nodes and tasks are to be serviced is the approach's effective partitioning of complex tasks into simple ones. Moreover, the algorithm preserves application functionality despite node failures. For highly available applications, two algorithms are proposed: an optimal one when component loads are known and a suboptimal one when loads are unknown. By maximizing the component-based architecture and application scalability features, both algorithms aid in building highly available applications. I An approach is then presented to find the optimal number of component types per node, recognizing that each node has a load threshold that cannot be crossed and aiming to minimize running costs for the application. To evaluate the performance of the suboptimal algorithm, it will be analyzed in terms of how well it performs in the environment of node loading, how distance it is from optimality, and its success rate, respectively. It is also found in the Priority-Based Job Scheduling Algorithm in Cloud Computing called PJSC, an algorithm that implements the Multiple Criteria Decision Making model under the Analytical Hierarchy Process (AHP) as a way of solution. AHP was chosen because it suited tasks involving multiple attributes for multi-criteria solutions such as scheduling. The algorithm uses a Multiple Criteria Decision Making model based on the Analytical Hierarchy Process. AHP is chosen as it is considered well-suited to the problem of multi-attribute solution tasks such as scheduling. The performance of the proposed algorithm is evaluated in terms of consistency, complexity, and makespan [14].

The experimental results show that while the proposed algorithm can be said to have acceptable complexity, its performance concerning makespan can be further improved. Based on a review of 205 journal papers written in the area, an extensive taxonomy of research issues in cloud computing is presented [15]. The papers are classified under four broad categories: technological issues, business issues, domain and application issues, and conceptualizing cloud computing. The study finds that currently, more attention is being paid to technology issues in cloud computing, while its social and organizational issues are being discussed with increasing intensity [16]. Between the two, scheduling algorithms have been identified as one of the pivotal factors to enhance job task workflows and user satisfaction toward service providers in cloud computing. By widening the parameters taken into consideration in scheduling algorithms, it would be able to enhance resource allocation and scheduling in cloud computing environments [17]. Scheduling algorithms can be thought of as taking several factors for input: execution time, deadline, energy efficiency, transmission cost, performance index, and makespan. The job scheduling algorithm proposed in this context is of Berger's distributive justice model. The implementation of this algorithm on the extended CloudSim platform ensures fairness in resource allocation using the two proposed fairness constraints based on Quality of Service (QoS) and a fairness justice function. The experimental findings indicate that while the proposed algorithm exhibits acceptable complexity, there is another proposed algorithm [18] that combines genetic and ant colony algorithms for route scheduling in a cloud database. The genetic algorithm's initial value is used as input for the ant colony algorithm, which transforms it into the pheromone initial value to find an optimal solution. Experimental results demonstrate that this hybrid algorithm improves the efficiency of cloud computing by quickly and effectively identifying suitable application databases.

METHODOLOGY

Using artificial neural networks has gained favor over simple genetic algorithms, which are time- and cost-dependent as well, in performing task scheduling and optimization in dynamic cloud environments [19]. Here, fitness evaluation applies in determining the best schedule. The outcome of an expected value generated at an output layer is required for correctness and accuracy towards the end of the learning or training period. The main focus of this work revolves around task-scheduling issues in a scalable cloud environment; genetic algorithms, ant colony methods, and the Big-Bang-Big-Crunch cost-aware approach are being utilized heavily to optimally allocate virtual machines. For the quality of service of the neural network to be improved, the objective has been to minimize its errors [20]. The task scheduler proposed is based on an artificial neural network, taking as inputs task and virtual machine vectors along with basic parameters for the model, as shown in Fig. 2.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 4 (2025)

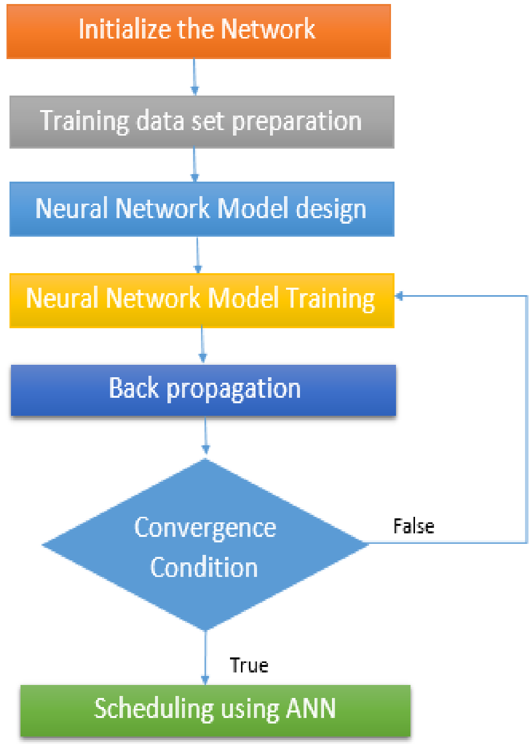

FIG. 2: PROPOSED TASK SCHEDULER BASED ON ANN

INITIALIZATION OF ANN SCHEDULER

Initialize the Artificial Neural Network (ANN) scheduler, setting up the structure and parameters for subsequent training.

PREPARATION OF TRAINING DATASET

Prepare a well-structured dataset that will be used to train the ANN model.

DESIGNING NEURAL MODEL

Designing the architecture of the neural model, specifying how inputs are processed through the network to produce scheduling decisions, is shown in Fig. 3.

Datacenter / VM configuration	
Type	CloudSIM VM
Storage /RAM	10GB / 512 MB
Network Bandwidth	10Mbps
No of CPUs	1
VM Ratio	{Small , Medium , Large , X-Large}
No of VMs	{8, 16, 32}
Simulator configuration	
File System	Local
Clustering	Disabled

FIG. 3: ARCHITECTURE OF THE NEURAL MODEL

TRAINING OF NEURAL MODAL

Train the neural model using the prepared dataset to learn the mapping between input vectors and scheduling decisions.

BACKPROPAGATION AND CORRECTING ERRORS

Implement backpropagation to adjust the model's parameters based on the computed errors during training.

TASK SCHEDULING ON THE VIRTUAL MACHINE

The implementation of ANN shows real-time scheduling decisions for tasks over virtual machines. The CloudSim simulator is used here to assess the effectiveness of our methods. CloudSim is an essential tool for simulating a cloud environment. We utilized virtual machines (VMs) on which their own storage can indicate if they are busy. VMs are of different sizes, such as small, medium, large, and extra-large. Our methods were tested under three groups of VMs. The first group consisted of one VM from each size class (4 VMs in total), the second group consisted of four VMs of each size (16 VMs in total), and the third group consisted of eight VMs of each size (32 VMs in total). A set of 12 tasks for testing was run against all three groups of VMs, thus giving us 36 different testing scenarios. Each VM with the configuration of 1 Gbyte storage, 512 Mbyte RAM, 10 Mbit /s of network bandwidth, and one CPU was used for our experiment. Common local file system storage was used, such that each VM has its own storage instead of shared storage, where all data would be stored in one location. The other way CloudSim has a Clustering Engine, which merges tasks being processed together. In our experiments, we turned this feature off.

RESULTS

Algorithms must be in black and white. The caption must be placed before the algorithm and is obligatory. All algorithms must be mentioned in text, like this: Algorithm 1.

ANN is configured with a range of 100 to 500 iterations for thorough training. It's structured with three layers: a single input layer housing 2 neurons, two hidden layers, each with 2 neurons, totalling 6 hidden units, and a single neuron in the output layer for precise predictions. As illustrated in Fig. 4, the learning rate is flexible and adaptable, and may be adjusted between 0.1 and 0.9 to maximize the network's performance while maintaining a balanced approach to learning between speed and accuracy.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 4 (2025)

Tasks Application	ANN	HEFT	MINMIN	MAXMIN	RR	FCFS	MCT
All Task Sizes {Small, Medium, Large}	62.31 ± 4.88	99.55 ± 4.80	119.84 ± 4.39	113.36 ± 5.02	112.95 ± 13.52	119.84 ± 13.84	100.96 ± 9.02

FIG. 4: RESULTS OF OTHER METHODS

The details are given in Fig.5, which shows the overall results of all VMs and Tasks, showing how the execution time varies for different tasks. The tasks submitted range from 50 to one thousand. To compare task scheduling techniques, we use the average execution time as the performance metric. The results indicate that Artificial Neural Network (ANN)-based task scheduling techniques outperform other baseline algorithms in terms of execution time (measured in milliseconds).

Application Size	No Of VMs	ANN	HEFT	MINMIN	MAXMIN	RR	FCFS	MCT
Small	Small	15.9989	**14.1131**	23.211	15.3275	15.4486	16.7487	18.0751
	Medium	**14.0997**	14.2128	16.3827	14.2584	15.5911	15.3292	14.1279
	Large	**6.0264**	9.4706	9.4792	9.441	9.5293	9.4523	9.4436
Medium	Small	13.9852	**7.9588**	25.4003	18.4432	87.3345	87.3345	13.9744
	Medium	**8.0987**	5.707	7.7164	7.0453	29.0844	29.0844	9.6625
	Large	6.5287	**2.8697**	3.4996	2.9647	4.8269	4.9214	3.4289
Large	Small	9.135	8.0487	6.1173	**5.6271**	237.518	237.518	5.9525
	Medium	**3.0255**	4.8862	4.8939	5.9262	89.8922	109.892	42.877
	Large	2.5018	**1.4865**	2.1634	3.5278	11.4675	11.4675	2.4932
X- Large	Small	**103.412**	112.301	162.511	136.627	129.518	120.518	118.525
	Medium	**52.025**	91.886	102.893	106.926	88.892	90.8922	89.877
	Large	**31.501**	94.486	94.163	96.527	94.467	95.467	94.493

FIG. 5: RESULTS OF OTHER METHODS

Fig. 6 below illustrates the simulation results for the presented ANN task scheduling technique. The focus is on the execution time, considering various VM counts categorized as Small, Medium, and Large. These figures compare how the average execution time changes as the number of user requests (tasks) increases, specifically for three types of VMs.

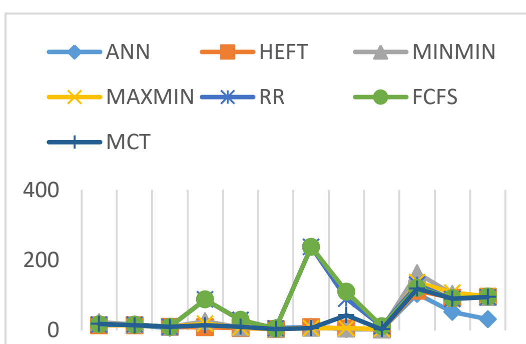

FIG. 6: ANN TASK SCHEDULING TECHNIQUE

RESULTS

This study introduces a neural network scheduling method for optimizing task assignments on virtual machines within a scalable cloud envirobrain's. This approach draws inspiration from the human brain functioning. To evaluate its effectiveness, we assess its performance using both real and fabricated datasets, while also considering resource scaling and task volume expansion. We use various performance metrics such as average start time (in milliseconds), average finish time (in milliseconds), execution time (in milliseconds), fault rate (number of failures), and completed tasks (number of successfully executed tasks). The training, validation, and prediction are conducted using real-time datasets obtained from standard workload files, as well as fabricated datasets, across Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) levels. Our findings in the results and discussions section demonstrate that our proposed Artificial Neural Network (ANN) technique surpasses existing baseline algorithms. We measure performance across a wide range of scenarios, with task counts varying from 1 to 30,000 and iterations ranging from 100 to 500. This paper introduces a scheduling algorithm that uses Artificial Neural Network (ANN) for managing tasks in dynamic cloud computing environments. What makes our approach unique is its ability to handle task scheduling in a cloud environment where tasks regularly appear. Through experiments, we found that our ANN-based model effectively addresses the task scheduling challenge for various workflow patterns with different types of resources in dynamic settings. Our ANN approach performs better than other algorithms in most test situations, especially when considering the overall average time taken for all test scenarios. In the future, we plan to enhance our algorithm to consider multiple factors like cost, security levels, deadlines, and not just the time taken.

REFERENCES

[1] Zhang, H., Shi, J., Deng, B., Jia, G., Han, G., & Shu, L. (2019). MCTE: Minimizes task completion time and execution cost to optimize scheduling performance for the smart grid cloud. *IEEE Access*, *7*, 134793-134803.

[2] Manekar, A. S., & Gera, P. (2021). Optimize Task Scheduling and Resource Allocation Using Nature Inspired Algorithms in Cloud based BDA. *Webology*, *18*(SI01), 127-136.

[3] Chraibi, A., Alla, S. B., & Ezzati, A. (2022). An efficient cloudlet scheduling via bin packing in cloud computing. *International Journal of Electrical and Computer Engineering*, *12*(3), 3226.

[4] Maashi, M., Alabdulkreem, E., Maray, M., Shankar, K., Darem, A. A., Alzahrani, A., & Yaseen, I. (2024). Elevating Survivability in Next-Gen IoT-Fog-Cloud Networks: Scheduling Optimization With the Metaheuristic Mountain Gazelle Algorithm. *IEEE Transactions on Consumer Electronics*.

[5] Daniyal, S. M., Abbasi, M. M., Hussain, D., Amjad, U., Abro, A. B., & Naeem, M. (2024). A hybrid approach for simultaneous effective automobile navigation with DE and PSO. VAWKUM Transactions on Computer Sciences, 12(2), 01-15.

[6] Chraibi, A., Ben Alla, S., Touhafi, A., & Ezzati, A. (2023). A novel dynamic multi-objective task scheduling optimization based on Dueling DQN and PER. *The Journal of Supercomputing*, *79*(18), 21368-21423.

[7] Chraibi, A., Ben Alla, S., Touhafi, A., & Ezzati, A. (2023). A novel dynamic multi-objective task scheduling optimization based on Dueling DQN and PER. *The Journal of Supercomputing*, *79*(18), 21368-21423.

[8] Yousuf, W. B., Talha, U., Abro, A. A., Ahmad, S., Daniyal, S. M., Ahmad, N., & Ateya, A. A. (2024). Novel Prognostic Methods for System Degradation Using LSTM. IEEE Access.

[9] Rawat, P. S., Dimri, P., Gupta, P., & Saroha, G. P. (2021). Resource provisioning in scalable cloud using bio-inspired artificial neural network model. *Applied Soft Computing*, *99*, 106876.

[10] Zavieh, H., Javadpour, A., & Sangaiah, A. K. (2024). Efficient task scheduling in cloud networks using ANN for green computing. *International Journal of Communication Systems*, *37*(5), e5689.

[11] Nayak, P. K., Singh, R. S., Kushwaha, S., Bevara, P. K., Kumar, V., & Medara, R. (2024). Efficient task scheduling on the cloud using artificial neural network and particle swarm optimization. *Concurrency and Computation: Practice and Experience*, *36*(6), e7954.

[12] Gurusamy, S., & Selvaraj, R. (2024). Resource allocation with efficient task scheduling in cloud computing using hierarchical auto-associative polynomial convolutional neural network. *Expert Systems with Applications*, *249*, 123554.

[13] Shakil, M. H., Abbasi, M. D., Majeed, M. K., Yousuf, W. B., Daniyal, S. M., & Amjad, U. (2025). Exploring Innovations and Security Enhancements in Android Operating System. Spectrum of engineering sciences, 3(2), 472-495.

[14] Mangalampalli, S., Karri, G. R., Kumar, M., Khalaf, O. I., Romero, C. A. T., & Sahib, G. A. (2024). DRLBTSA: Deep reinforcement learning based task-scheduling algorithm in cloud computing. *Multimedia Tools and Applications*, *83*(3), 8359-8387.

[15] Zavieh, H., Javadpour, A., & Sangaiah, A. K. (2024). Efficient task scheduling in cloud networks using ANN for green computing. *International Journal of Communication Systems*, *37*(5), e5689.

[16] Chen, Z., Wei, P., & Li, Y. (2023). Combining neural network-based method with heuristic policy for optimal task scheduling in hierarchical edge cloud. *Digital Communications and Networks*, *9*(3), 688-697.

[17] ul Hassan, M., Al-Awady, A. A., Ali, A., Iqbal, M. M., Akram, M., Khan, J., & AbuOdeh, A. A. (2023). An efficient dynamic decision-based task optimization and scheduling approach for microservice-based cost management in mobile cloud computing applications. *Pervasive and Mobile Computing*, *92*, 101785.

[18] Badri, S., Alghazzawi, D. M., Hasan, S. H., Alfayez, F., Hasan, S. H., Rahman, M., & Bhatia, S. (2023). An efficient and secure model using adaptive optimal deep learning for task scheduling in cloud computing. *Electronics*, *12*(6), 1441.

[19] Gupta, P., Anand, A., Agarwal, P., & McArdle, G. (2024). Neural network inspired efficient scalable task scheduling for cloud infrastructure. *Internet of Things and Cyber-Physical Systems*, *4*, 268-279.

[20] Xing, Y. (2024). Work Scheduling in Cloud Network Based on Deep Q-LSTM Models for Efficient Resource Utilization. *Journal of Grid Computing*, *22*(1), 36.