© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715299

A Hybrid ACO-GA Model for High-Quality Resource Allocation

LESSON, W.A[1], AKAZUE, M.I.[2]
[1,2]Computer Science Department, Delta State University, Abraka, Nigeria.

Abstract- The distribution of resources is a challenging issue due to the dynamic nature of operations, the diversity of resources, and the varying priorities of activities and applications. Allocating resources involves giving virtual machines the characteristics that clients have chosen. Effective use of cloud resources also includes managing workloads and assigning them to virtualization. Allocating resources is one of cloud computing's most important duties. The different resource allocation paradigms and cost-based resource allocation method for the heterogeneous cloud environment are examined in this research. Site visits and document review were the analysis tools employed to collect data. A review of current methods for allocating resources was examined. The models were evaluated based on the algorithms' strengths, weaknesses, and functionality. The results demonstrated that the ACO-GA model hybridization offered a high-quality resource allocation in cloud computing.

Index Terms- Resource Allocation, Ant Colony, Genetic Algorithm, Optimization, Cloud Computing; Hybridization.

I. INTRODUCTION

Numerous studies on cloud-based applications include community-cloud frameworks (Ojigo et al., 2015), smartphones for banking transactions (Ojugo & Yoro, 2018), cloud compute resources (Allenotor et al., 2015), and more. Additional studies have been conducted on genetic algorithms (Ojugo & Okobah, 2017; Oweimieotu et al 2024; Okofu et al, 2024a; Ojie et al 2023a; Okofu et al, 2024b; Ojie et al, 2023b, Ojugo et al 2021) and optimization algorithms (Omede, 2022; Ajenaghughrure et al. 2016; Oyemade & Ojugo, 2021).

The allocation of resources in cloud computing is a challenging issue due to the dynamic nature of operations, the variety of resources, and the varying priority of activities and apps (Senthilkumar et al., 2023; Ojugo & Nwankwo., 2021). Scholars have proposed several approaches to resource allocation in cloud computing to address this problem. Resource allocation in cloud computing refers to the practice of assigning appropriate resources to efficiently fulfill client requests (Manzoor et al., 2020). In order to do this, virtual machines must be assigned the characteristics that the clients have chosen. Users submit their jobs, which can take a variety of times to finish. Effective use of cloud resources also includes managing workloads and assigning them to virtualization.(Pingulkar et al, 2023).

Optimizing Cloud Computing innovation services is essential since so many businesses, big and small, are searching for methods to reduce expenses without compromising productivity. This is due to the numerous advantages that these services provide, which are directly related to what businesses require. Users enjoy simple, reliable, and scalable access to a shared pool of programmable network assets when cloud computing performance methodologies are built on a utility-based commercial model (Jeyaraman et al., 2024).

Because workloads are inherently unpredictable and variable, allocating resources presents a number of difficulties. Every service and application has different consumption patterns and resource requirements. It is difficult to create a resource allocation strategy that works for everyone because of this heterogeneity, which calls for adaptable and responsive techniques. Furthermore, cloud systems' elasticity makes it difficult to scale resources up or down in response to shifting demand levels. Static provisioning is a common component of traditional resource allocation techniques, which can result in underutilization during times of low demand or resource contention and performance degradation during high loads (Al-Asaly et al., 2022). But choosing how and when to distribute resources is a difficult undertaking.

As a result, allocating resources is one of cloud computing's most important duties. The different resource allocation paradigms and cost-based resource allocation method for the heterogeneous cloud environment are examined in this research.

II. REVIEW OF RELATED RESEARCH WORK

Chen et al. (2021) created and applied RAMPVR to solve the resource allocation time issue by speeding up the multi-objective optimization model's solution. An optimization model for multi-objective resource allocation was developed. CloudSim was utilized by the simulation tools. and employed criteria such as the number of PMs used, resource performance matching distance, resource percentage matching distance, resource usage, and time cost of resource allocation to benchmark their work with RR, SEPA2, NSGA-II, and RAMPVR. For the sequence L1, 5-point values are required.

Zheng et at. (2024) utilized AI-driven predictive analytics to create an effective resource allocation system for cloud computing environments. Workload consolidation, resource oversubscription, and elastic resource pools were the strategies used to optimize utilization efficiency. The work offered a solution to the challenges of contemporary cloud settings. The simulation program utilized was CloudSim. MAE, RMSE, MAPE, Resource Utilization, SLA Violation Rate, PUE, and Cost Efficiency are the performance metrics that are employed. The results of their benchmarking with XGBoost and LSTM indicated that the model's scalability and flexibility needed to be improved in order to handle more complex and dynamic cloud environments.

Mohammad (2024) We out an empirical study on RACC to illuminate the challenges faced by SMEs in their quest for successful RACC. This study provided important insight into how small and medium-sized businesses allocate their cloud computing resources. The program utilized was CloudSim. The findings indicated that in order for RACC to be successful in SMEs, organizational and environmental barriers must be investigated and removed.

Gogula et al. (2024) created customized particle swarm optimization to optimize resource allocation in cloud computing for improved performance. The method employed was ORA-CC. They developed a task processing framework that can make decisions in real-time to select the optimal resource for managing a variety of complex uses on virtual computers (VC). equilibrium in cloud computing between conflicting objectives. CloudSim was employed. MoPSO ORA-CC and MoPSO measures were employed for assessment. The efficacy of MoPSO in attaining optimal resource allocation techniques that result in improved overall performance must be proven.

Syed (2024) Utilizing DevOps, AWS, and Azure Effectiveness Deployment Dataset, machine learning techniques were deployed to optimize cloud resource management. In order to improve adaptability and responsiveness to changing workloads, their work incorporated real-time adaption algorithms. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were used to assess their system. Advanced Techniques, Emerging Technologies, and Extended Data Sources must be used. AWS, Azure Effectiveness Deployment Dataset, and DevOps for cloud resource management. In order to improve adaptability and responsiveness to changing workloads, their work incorporated real-time adaption algorithms. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were used to assess their system. Advanced Techniques, Emerging Technologies, and Extended Data Sources must be used.

Lakshmikanth et al. (2025) Provides a framework to represent workloads and cloud data centers. When CSA-based allocation was used, resource utilization and workload responsiveness were greatly enhanced. Resource Utilization, Response Time, and Cost Effectiveness were used to evaluate their system. The outcome demonstrated the necessity of fine-tuning CSA parameters in a variety of cloud settings and integrating them with machine learning methods for improved predictive capabilities and real-time adaptation to changing workload demands.

Rajawat et al. (2025) utilized machine learning's transformational potential to redefine job scheduling and resource allocation inside cloud computing environments. To maintain equilibrium in the distribution of resources, a load balancer and scheduler strategy was employed. Their system optimized throughput and minimized late environments while ensuring that resources are neither overused nor overly stressed. Round robin and least connection were used to test it. Future resource requirements for IoT applications and workloads must be projected.

Seisa et al. (2025) conducted an experimental evaluation that examined the behavior of the system under different conditions, offering important insights into the performance, scalability, and robustness of multi-agent control systems deployed on edge infrastructure. They used a resource allocation strategy that provided enough resources for the controller's execution throughout. The system was evaluated using the CNMPC and UAVs CPU Multi-agent System and Kubernetes Cluster. In order to improve delay compensation mechanisms, it is necessary to incorporate the predictive network latency and bandwidth fluctuations.

Amarnath (2025) employed DeepRM machine learning algorithms to develop self-regulating systems and enable autonomic cloud management concepts and real-time optimization. Cost reduction was the evaluation metric. They used Total Cost Reduction and Computational Waste Reduction to benchmark their system. Cloud success still needs to change in tandem with company needs and technology advancements.

Arvindhan and Kumar (2023) employed Actor Critic (AC) and AC-CIWAS, which are based on Deep Reinforcement Learning (DRL), to minimize the burden of the job scheduled with QoS assurance by about 20% when compared to current baseline allocation approaches. By efficiently distributing resources and distributing workloads among several servers, this reduced the overall job execution time. Response time, CPU utilization, throughput, and task completion time were their evaluation criteria. GA, DSOS, MSDE, PSO, WOA, DQL, and ACD-RL were used for benchmarking. Setting up an edge cloud computing network architecture that enables multiple individuals to collaborate on computing tasks is still necessary.

Gongada et al. (2024) created a thorough framework that addresses the drawbacks of conventional task scheduling techniques by utilizing the combined strengths of FFO and CNN. In cloud computing contexts, the FFO-CNN architecture improved performance and resource utilization. Average Response Time, Resource Utilization, and Throughput were used to test their system. FFO-CNN, CNN-MBO, and GA were used as additional benchmarks for their model. In order to enable the deployment of the FFO-CNN model in bigger and more varied cloud settings, it will be advantageous to look into methods for enhancing the scalability and efficiency of the training process.

Karthika (2025) developed an empowering, energy-efficient resource allocation in mobile networks using the Deep Q-Learning approach to transform resource allocation optimization. It was accomplished by utilizing artificial intelligence, and it opened the door for more resilient, intelligent, and adaptable mobile networks that satisfy the changing needs of a connected society. They made use of a simulation program called Modulation Scheme. Work was benchmarked using metrics for power consumption, energy efficiency, throughput and fairness, and capacity considerations. For improved performance, it is necessary to investigate how multi-agent reinforcement learning and other network factors could be combined.

Baresi et al. (2025) established a dynamic resource allocation for deadline-constrained neural network training using the DECOR-NN model, which was built on top of PyTorch to address the Leverages control theory problem. The findings demonstrated that DÉCOR-NN allows users to adjust the deadline at runtime to account for shifting work priorities and dynamically assigns either GPUs or fractions of CPUs to satisfy user deadlines. CPU-based training (RQ1), GPU-based training (RQ2), and variable deadlines (RQ3) were used to benchmark the work. Distributed Data Parallel for GPUs is required, as is the implementation of support for both early-stopping and fractional GPU allocations, as well as the development of control mechanisms to manage several concurrent training programs.

Logeshwaran et al. (2023) used the IDM paradigm to address the issues of excessive expenses, subpar network performance, and low customer satisfaction. Utilize machine learning techniques and predictive analytics to find the most effective resource allocation. 5G was utilized by the simulation tools. Optimal multitier resource allocation (OMRA), learning-based energy-efficient resource management (LERM), adaptive resource management (ARM) using metrics, and resource management based on reinforcement learning (RMRL) were used to benchmark the work. Intelligent decision models are necessary to assist businesses in making better choices on the distribution of resources among different departments, enabling them to maximize their resources and optimize their operations.

Kashyap et al. (2025) offered notable improvements in accuracy and efficiency by using the VSSW model to address the issue of resource workload prediction in dynamic cloud systems. The results of their work demonstrated the development of cloud data center management techniques and performance optimization in a changing and dynamic environment. They benchmarked their work using MAE and RMSE measures and used CloudSim as a simulation tool. Nevertheless, their model needs to be expanded in order to provide an improved solution that makes use of Explainable Generative AI.

Kadam et al. (2025) utilized models of genetic algorithms and reinforcement learning to provide real-time adaptability and efficient global optimization for scalable and effective cloud resource management. It was accomplished in order to investigate a large search space of potential resource allocation configurations. Work that was benchmarked utilizing metrics for convergence to local optima and scalability issues: system dependability, energy efficiency, and resource use. Scalability and real-time adaptation had to be addressed.

Bao et al. (2025) Several high-performance servers were set up as rendering nodes, and several pathways' concurrent rendering demands are supported. A single pool of rendering resources was created by connecting the nodes via the network, and hardware resources were carefully optimized to guarantee that rendering tasks could be assigned and completed quickly and effectively. The simulation program utilized was CloudSim. GPU, CPU, and UE metrics were used to benchmark the job. To guarantee that every node renders its performance and prevents idle or overburdened resources, tasks must be assigned to CPU, GPU, memory, and other resources.

Huang et al. (2023) employed the Deep Reinforcement Learning (DRL) model to address the CORA algorithm issue. Their approach fared better than other algorithms in terms of processing delay and cost, according to the results. CORA algorithm and non-DRL algorithms with metrics, DQN and DDPG, were used to benchmark the work. The situation where several cars vie for edge servers' processing power will need to be taken into account.

Liu et al. (2025) utilized the BWO model to reduce the overall work completion time and increase the operators' overall profit. The loading balance, total operator revenue, and total task completion time were all attained. 5G simulation tools were utilized. MOGA, MOPSO, and IMOBWO measures were used to benchmark the work. The multi-objective resource allocation problem in a 5G edge computing (EC) network requires more innovative algorithms.

Mohammadi et al. (2024) used the NSGA-II paradigm to address the multi-objective evolutionary algorithm challenge. It was accomplished by employing sigma scaling to improve the NSGA-II algorithm's exploration and exploitation capabilities. MATLAB was utilized for the simulation tools, and MOPSO, SPEA2, NSGAIII, MOCS, and MONSGA-II were used for benchmarking. It is necessary to define a system in order to analyze its performance.

Zhao et al. (2025) developed a multi-actor network using the LEO model to accomplish satellite coverage, task offloading, communication, and resource allocation for computing. Their dynamics-aware algorithm regularly outperforms previous learning-based techniques, producing the lowest system delay and the maximum link throughput, according to numerical results. Starlink was a tool for

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715299

simulation. RAND-based TORA, PG-based TORA, PPO-based TORA, and STA-PPO algorithms were used to benchmark the work. Space, aerial, and ground parts must be seamlessly integrated and expanded using various cutting-edge methods.

In conclusion, the Ant Colony Algorithm (ACO) approach has been used by numerous authors for optimization; however, it has several drawbacks, including slow convergence, premature convergence, poor dynamic adaptability, and lack of inherent adaptability to such changes unless hybridized or extended; it can be computationally and memory-intensive to maintain pheromone matrices and track all ant paths; and it is not appropriate for lightweight or real-time scheduling unless optimized. For resource allocation, we proposed the hybrid ACO-GA Model.

III. MATERIALS AND METHODS

3.1 Methodology Adopted

Object-Oriented Analysis and Design was employed in this investigation. The cloud environment is a platform that offers different services to customers and businesses in the changing paradigms. Because of the enormous number of activities that users have, allocating resources to tasks is an issue in cloud computing environments. Here, an effective method for allocating resources is created to deal with these issues.

3.2 Analysis of the developed Model

The following is a discussion of the two meta-heuristic algorithms that make up the Hybrid model (ACOGA): the Genetic Algorithm (GA) and the ACO algorithm.

3.2.1 Genetic Algorithm

Natural selection serves as the model for a genetic algorithm (GA), which is a member of the larger family of evolutionary algorithms. Genetic algorithms are widely utilized to produce great solutions to optimization and search problems using biologically inspired operators such as selection, crossover, and mutation. Applications of GA include improving decision tree performance, solving Sudoku puzzles, hyperparameter optimization, and causal inference. A genetic algorithm expands a population of possible solutions (sometimes referred to as individuals, animals, organisms, or phenotypes) in order to identify better solutions to optimization problems.

Typically, a genetic algorithm needs
i. The solution domain represented genetically
ii. A fitness function to assess the domain of solutions
After identifying the genetic representation and fitness function, a GA begins with a population of solutions and iteratively improves it using the mutation, crossover, inversion, and selection operations.

The fitness function is defined over the genetic representation and evaluates the quality of the represented solution. The problem always determines the fitness function. Interactive genetic algorithms can be used to find the fitness function value of a phenotypic in order to calculate the air resistance of a vehicle whose shape is encoded as the phenotype. The next stage is to create a second-generation population of solutions from those chosen using a combination of genetic operators, such as crossover and mutation.

Developing genetic algorithms is simple, but comprehending their behavior is challenging. It is particularly challenging to explain why, when applied to real-world issues, these algorithms usually yield high-fitness solutions.

3.2.2 Ant Colony Optimization algorithm

In computer science and operations research, the ant colony optimization algorithm (ACO) is a probabilistic technique used to tackle computing problems that may be boiled down to finding effective graph paths. Artificial ants are multi-agent systems that imitate the behavior of real ants. Biological ants frequently communicate through pheromones. Combinations of artificial ants and local search algorithms have emerged as the optimal choice for a range of graph-based optimization problems, including internet and automobile routing (Kniazhyk et al., 2023).

3.2.3 Class diagram of the Developed System

A class diagram is a kind of static structure diagram that illustrates a system's classes, attributes, operations (or methods), and object relationships. Figure 1 displays the hybrid ACO-GA Model's class diagram.

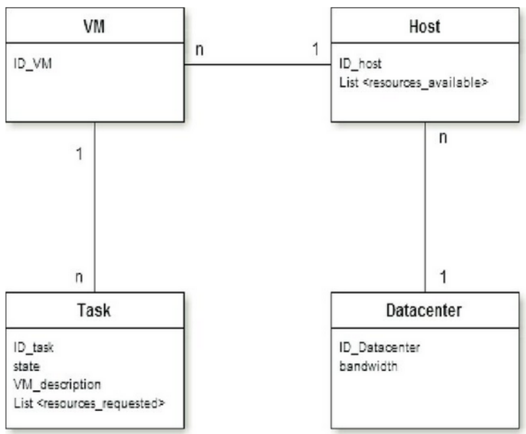

Figure 1: Class diagram of the hybrid ACO-GA Model

3.2.4 Sequence diagram of the developed model

Sequence diagrams are interaction diagrams that use the vertical axis of the graphic to represent time, messages transmitted, and when in order to illustrate the stages involved in performing operations of the hybrid ACO-GA Model (Figure 2). They document how objects interact with one another when working together.

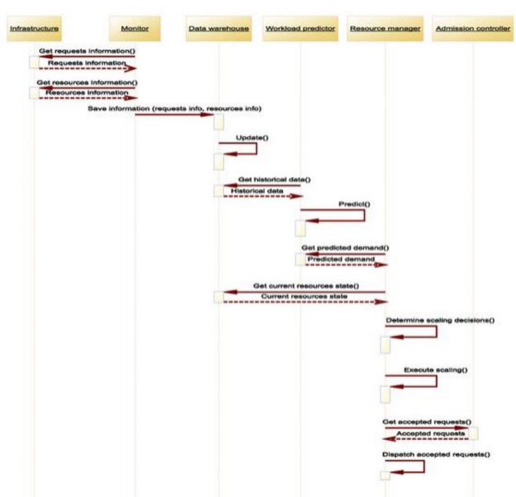

Figure 2: Sequence diagram of the hybrid ACO-GA Model

3.2.5 High level model of the developed System

High-level models serve as basic representations that facilitate analysis, understanding, and decision-making. Graphs with quantitative data, mathematical equations, and visual representations like diagrams and drawings are some of the complementary forms that these models might take (Figure 3).

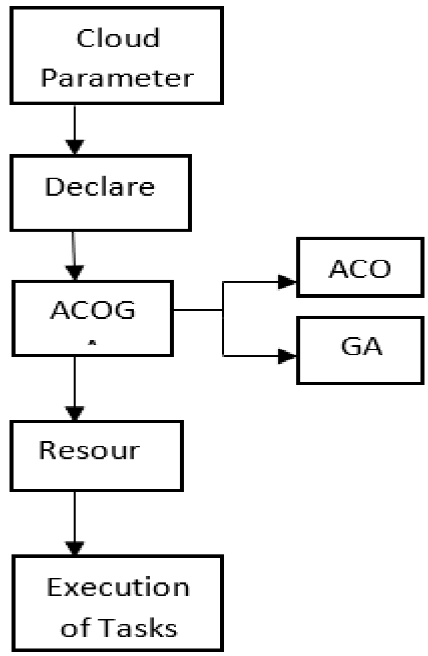

Figure 3: High level model of the hybrid ACO-GA Model

3.3 Implementation of the hybrid ACO-GA Model

3.3.1 System Algorithm

The hybrid algorithm is a novel hybrid optimization method that blends the Genetic Algorithm and the Ant Colonization Optimization algorithm. It is a combination of the ACO and GA algorithms. In actuality, the hybrid idea can withstand optimization issues in practical implementations. The Ant Colonization concept is affected by the Genetic approach in this designed approach to guarantee improved convergence rate and speed that tolerates optimal resource allocation with the fixed objectives.

3.3.2 Mathematical Expression

The exploring power of Ant Colony Optimization (ACO) and the exploitation and diversity preservation of Genetic Algorithm (GA) are combined in the Ant Colony Optimization–Genetic Algorithm (ACO-GA) hybrid. ACO-GA's mathematical formulation combines the two algorithms'

1. ACO Component

ACO mimics the pheromone-based path selection of ants:

Pheromone Update Rule:

$$T_{ij}(t+1)=(1-p)\times T_{ij}(t)+\sum_{k=1}^{m}\Delta T_{ij}^{k}(t) \quad (1)$$

$T_{ij}(t)$: Pheromone on edge (i,j) at time t
p: Evaporation rate ($0<\rho<1$)
$\Delta T_{ij}^{k}(t)$: Pheromone deposited by ant k on edge (i,j)

Transition Probability:

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{[T_{ij}(t)]^{\propto}\cdot[n_{ij}]^{\beta}}{\sum_{l\in N_i^k}[T_{ij}(t)]^{\propto}\cdot[n_{ij}]^{\beta}}, & if\ j\in N_i^k\\ 0, & otherwise\end{cases} \quad (2)$$

n_{ij}: Heuristic desirability (e.g., inverse of task execution time or cost)
α, β: Influence weights of pheromone vs. heuristic
N_i^k: feasible nodes for ant kk from node ii

2. GA Component
GA introduces crossover and mutation to improve solution quality and diversity.
Crossover Operation:

Offspring = Crossover (Parent1, Parent2) (3)

One-point or uniform crossover selects genes from two parents to create a new solution.
Mutation Operation:

Offspring′ = Mutate (Offspring) (4)

Mutation introduces small random changes to explore new solution spaces.

3. ACO-GA Hybrid Framework
ACO generates a population of solutions (ants), and GA improves them. The hybrid can be expressed as:
Overall Optimization Objective:

$$\min_{s\in\infty} f(s)\ where\ f(s)=w_1\cdot Cost(S)+w_2\cdot Makespan(S)+w_3\cdot Reliability(S)+w_4\cdot Throughput(S) \quad (5)$$

S: A scheduling solution
w_1, w_2, w_3, w_4: weights for multi-objective optimization

Hybrid Loop:
- Initialize pheromone T_{ij}
- For each ant:
- Construct solution based on P_{ij}
- Evaluate fitness f(S)
- Apply selection, crossover, and mutation to top solutions
- Update pheromones using improved solutions

Repeat until convergence or max iterations
Final Mathematical Expression (ACO-GA Update)
Let S_ACO be solutions from ACO and S_GA be refined by GA:

$$S_{best}=\arg\min_{S\in S_{GA}} f(s) \quad (6)$$

Then pheromone update uses the best GA solution:

$$T_{ij}(t+1)=(1-p)\cdot T_{ij}(t)+\Delta T_{ij}^{best} \quad (7)$$

In summary, a Core i5 laptop running 2.50GHz with 8GB RAM and a 64-bit Windows 11 operating system was utilized for the simulation. The cloud computing environment was simulated in this assessment using the popular CloudSim toolkit and Java programming language. The ACO Algorithm used 10 ants and a maximum iteration of 50, while the GA used a population of 20. The hybrid model convergence was taken into consideration for the stop condition. The simulations were conducted in a variety of environments.

IV. CONCLUSION

Faster convergence is provided by the hybrid paradigm, particularly in big or dynamic cloud environments. This improves response time and resource utilization efficiency. By improving the ACO with GA, computational efficiency is increased and premature convergence—where the algorithm becomes stuck at local optima—is avoided. The hybrid model is affordable and appropriate for cloud environments due to its adaptability.

REFRENCES

[1] D. Allenotor., D.A. Oyemade., A.A. Ojugo., (2015). A financial option model for pricing cloud compute resources based on cloud trace characterization. African J. of Computing & ICT, 8(2-2), pp99 – 108, [web]: www.africanjournalofcomputingict1.files.com /2017/08/vol-8-no-2-issue-2-august-2015merged.pdf

[2] A.A. Ojugo., R.E. Yoro., (2018). An intelligent lightweight market basket associative rule mining for smartphones cloud-based application to ease banking transactions, Advances in Multidisciplinary and Scientific Research, 4(3): pp79-90, [web]: www.isteams.net/aims-journal

[3] E.U. Omede. Optimization of Determinant Diagnostic Symptoms for Febrile Diseases using Genetic Algorithm. Communication in Physical Sciences, 2022, Vol 8 (4), 556-572

[4] A.A. Ojugo., I.P. Okobah., (2017). Hybrid fuzzy-genetic algorithm trained neural network model for diabetes diagnosis, Digital Inno. & Contemporary Res. in Sci., Eng. & Tech., 5(4): pp69-90, [web]: ww.isteams.net/digital-innovations-journal

[5] A.A. Ojugo., O. Nwankwo., (2021). Spectral-cluster solution for credit-card fraud detection using genetic algorithm trained modular deep learning neural network, Journal of Information and Visualization, 2(1): pp15-24, doi: 10.35877/454RI.jinav274

[6] Ojie D., Akazue M., Imianvan A. (2023a). A Framework for Feature Selection using Data Value Metric and Genetic Algorithm, International Journal of Computer Applications, Volume 184, Issue 43, p14-21, doi:10.5120/ijca2023922533

[7] S. N. Okofu, K. E. Anazia, M. I. Akazue, M. D. Okpor, A. E. Oweimieto, C. E. Asuai, G. A. Nwokolo, A. A. Ojugo, E. O. Ojei (2024a). Pilot Study on Consumer Preference, Intentions and Trust on Purchasing-Pattern for Online Virtual Shops, International Journal of Advanced Computer Science and Applications, Vol. 15, No. 7, 804 -811

[8] S. N. Okofu, M. I. Akazue, A. E. Oweimieotu, R. E. Ako, A. A. Ojugo, C. Asuai (2024b). Improving customer trust through fraud prevention e-commerce model, Journal of Computing, Science & Technology (JCST), vol1, issue 1, 76-86

[9] Oweimieotu, A. E., Akazue, M. I., Edje, A. E. & Asuai C. (2024): Designing a Hybrid Genetic Algorithm Trained Feedforward Neural Network for Mental Health Disorder Detection. Journal of Digital Innovations & Contemporary Research in Science, Engineering & Technology. Vol. 12, No. 1. Pp 49-62. dx.doi.org/10.22624/AIMS/DIGITAL/V11N4P4

[10] D. Oyemade., A.A. Ojugo., (2021). Optimized input genetic algorithm model for financial market, Int. J. Innovative Sci., Engr. & Tech., 8(2): 408-419, ISSN: 2348-7968, [web]: www.ijiset.com/articlesv8/articlesv8s2.html

[11] Ojie D.V, Akazue M, Omede E.U, Oboh E.O, Imianvan A. (2023b) Survival Prediction of Cervical Cancer Patients using Genetic Algorithm-Based Data Value Metric and Recurrent Neural Network, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307 (Online), Volume-13 Issue-2, May 2023

[12] A.A. Ojugo., C.O. Obruche., A.O. Eboka., (2021). Quest for convergence solution via hybrid Genetic Algorithm Trained Neural Network model for metamorphic malware detection, ARRUS J. Engr. & Tech., 2(1), pp12-23, doi: 10.35877/jetech613

[13] Ajenaghughrure I. B, Akazue M. I. and Onyekweli C. O. (2016) Ant colony optimization algorithm based vehicle theft prediction-prevention and recovery system model (aco-vtp2rsm) International Journal of Computer Science & Engineering Technology, Vol. 7 No.06, pp, 251-260

[14] Gongada, T. N., Girish, P., Desale, B., and Ghodake, S. P. (2024). Optimizing Resource Allocation in Cloud Environments using Fruit Fly Optimization and Convolutional Neural Networks. 15(5), 1173–1182.

[15] Arvindhan, M., and Kumar, D. R. (2023). Adaptive Resource Allocation in Cloud Data Centers using Actor-Critical Deep Reinforcement Learning for Optimized Load Balancing. March, 310–318.

[16] Amarnath, V. K. (2025). Engaging Techniques for Effective Resource Allocation in Cloud Computing : Improving Performance , Availability , and Cost Management. 16(1), 1–11.

[17] Seisa, A. S., Velhal, S., Kotpalliwar, S., and Satpute, S. (2025). Through Dynamic Computational Resource Allocation Optimization of Edge-Offloading for Centralized Controllers Through Dynamic Computational Resource Allocation.

[18] Rajawat, A. S., Goyal, S. B., Kumar, M., and Malik, V. (2025.). 64 Adaptive resource allocation and optimization in cloud environments : Leveraging machine learning for efficient computing. https://doi.org/10.1201/9781003471059-64

[19] Lakshmikanth, R., Dinesh, K., Sohan, S., Nithish, A., and Raju, S. U. K. (2025). An efficient resource allocation using Cuckoo search algorithm. 607–612. https://doi.org/10.1201/9781003559085-104

[20] Syed, S. (2024). Optimizing Cloud Resource Allocation with Machine Learning : A Comprehensive Approach to Efficiency and Performance. 1–21.

[21] Gogula, S., Sridhar, P., Arvind, S., and Sethy, A. (2024). Optimized resource allocation in cloud computing for enhanced performance with modified particle swarm optimization. 01140, 1–8.

[22] Mohammad, A. (2024). Key Challenges of Cloud Computing Resource Allocation in Small and Medium Enterprises. 372–388.

[23] Zheng, H., Xu, K., Zhang, M., Tan, H., and Li, H. (2024). Efficient resource allocation in cloud computing environments using AI-driven predictive analytics. 0, 17–23. https://doi.org/10.54254/2755-2721/82/2024GLG0055

[24] Manzoor, M. F., Abid, A., Farooq, M. S., Nawaz, N. A. and Farooq. U. (2020). Resource Allocation Techniques in Cloud Computing: A Review and Future Directions. Elektronika Ir Elektrotechnika, 26(6): 40-51. http://dx.doi.org/10.5755/j01.eie.26.6.25865

[25] Pingulkar, S., Tiwary, A. and Tyagi, S. (2023). Resource Allocation Techniques in Cloud Computing: A Comprehensive Review. International Journal of Engineering Research & Technology, 12(7): 350-357.

[26] Al-Asaly, M. S., Bencherif, M. A., Alsanad, A., and M. M. Hassan, (2022) “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, doi: 10.1007/s00521-021 06665-5.

[27] Jeyaraman, J., Bayani, S. V., and Arasu, J. N. (2024). Optimizing Resource Allocation in Cloud Computing Using Machine Learning Jawaharbabu Jeyaraman , Samir Vinayak Bayani and Jesu Narkarunai Optimizing Resource Allocation in Cloud Computing Using Machine Learning.

[28] Bao, J., Qiao, Y., Chen, N., New, S., and Surveying, D. (2025). A Cloud Rendering Approach Based on Multi-Node Resource Allocation. 13506(2023), 1–5. https://doi.org/10.1117/12.3057643

[29] Baresi, L., Garlini, M., Quattrocchi, G., Milano, P., Elettronica, D., and Bioingegneria, I. (2025). Dynamic Resource Allocation for Deadline-Constrained Neural Network Training.

[30] Chen, J., Du, T., and Xiao, G. (2021). A multi-objective optimization for resource allocation of emergent demands in cloud computing.

[31] Huang, J., Wan, J., Lv, B., Ye, Q., and Member, S. (2023). Joint Computation Offloading and Resource Allocation for Edge-Cloud Collaboration in Internet of Vehicles via Deep Reinforcement Learning. IEEE Systems Journal, 17(2), 2500–2511. https://doi.org/10.1109/JSYST.2023.3249217

[32] Kadam, S. R., Devaseelan, S., and Jadhav, A. N. (2025). Reinforcement Learning and Genetic Algorithm- Based Approach for Load Balancing and Resource Optimization in Cloud Data Centers. 16(1), 1–8.

[33] Karthika, S. (2025). Empowering Energy-E cient Resource Allocation in Mobile Networks with Deep Q-Learning Intelligence. 0–15.

[34] Kashyap, S., Singh, A., and Singh, S. (2025). Machine learning-centric prediction and decision-based resource management in cloud computing environments. Cluster Computing, 8. https://doi.org/10.1007/s10586-024-04787-8

[35] Kniazhyk, T., and Muliarevych, O. (2023). Cloud Computing with Resource Allocation Based on Ant Colony Optimization. 8(2).

[36] Logeshwaran, J., Kiruthiga, T., and Lloret, J. (2023). A Novel Architecture Of Intelligent Decision Model For Efficient Resource Allocation In 5g Broadband Communication Networks. 2986–2994. Https://Doi.Org/10.21917/Ijsc.2023.0420

[37] Zhao, D., Ding, R., and Song, B. (2025). Satellite-assisted 6G wide-area edge intelligence: dynamics-aware task offloading and resource allocation for remote IoT services. 68(February), 1–21.

[38] Jing Liu, Yuting Huang, Chunhua Deng, Longxin Zhang, Cen Chen and Keqin Li (2025). Efficient Resource Allocation Algorithm Maximizing Operator Profit Edge in 5G Computing network. https://doi.org/10.1007/s10723-024-09790-2