I. J. Computer Network and Information Security, 2026, 3, 105-121
Published Online on June 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2026.03.06

Adaptive Osprey-bowerbird Optimized Green Cloud Computing with Randomized Attention Coupled Fair Resource Distribution in Scalable Systems

Aishwarya Shekhar
School of Computing Science and Engineering, Galgotias University, Uttar Pradesh, India
E-mail: aishwarya1.2.shekhar@outlook.com
ORCID iD: https://orcid.org/0009-0001-7005-3760

Abdul Aleem*
School of Computing Science and Engineering, Galgotias University, Uttar Pradesh, India
E-mail: abdulaleem.cse765@gmail.com
ORCID iD: https://orcid.org/0000-0001-6676-9072
*Corresponding author

Received: February 26, 2025; Revised: August 25, 2025; Accepted: October 9, 2025; Published: June 8, 2026

Abstract: Cloud computing forms the basis for the emerging technologies in various fields, providing a reliable framework for managing resources to meet the needs of different applications. The rapidly increasing energy requirements inherent to cloud computing pose a real problem concerning sustainability. Energy efficiency, fair resource sharing, and performance consistent across the dynamic and heterogeneous cloud computing system are essential since existing approaches introduce inefficiency, energy consumption, and unfair distribution of loads. This research introduces Adaptive Osprey-Bowerbird Optimized Green Cloud Computing with Randomized Attention Coupled Fair Resource Distribution in Scalable Systems (AO-BO-RNCN-MAN) to address these challenges. The proposed framework integrates the Randomized Neural Coupling Network to learn diverse data representations, with the Multi-instance Attention Network to prioritize tasks, and Adaptive Osprey-Bowerbird Optimization, which is a combination of the Osprey Adaptive Algorithm and the Adaptive Bowerbird Optimization for further fine-tuning of the system. By optimizing the placement of virtual machines and scheduling of tasks, the proposed framework guarantees fairness and high energy utilization with low turnaround time. Performance assessments indicate that the proposed framework outperforms the existing systems with an energy efficiency of 99.82%, task scheduling accuracy of 99.61% and fair resource allocation of 99.74%. AO-BO-RNCN-MAN not only proposes a new way of addressing green computing challenges but also opens the gates to sustainable, adaptive, and scalable cloud infrastructures for resource management in cloud ecosystems and establishes the proposed conceptual framework as a new standard.

Index Terms: Cloud Computing, Green Computing, Resource Allocation, Service-Level Agreements, Virtual Machine.

1. Introduction

As the impact of the energy used by computing systems and their supporting structures continues to grow, green computing has become an essential paradigm [1]. Energy efficiency in cloud computing is one area that has received much attention because of the characteristics of this rapidly growing and highly flexible but resource-intensive industry [2]. Optimized green computing, on the other hand, focuses on strategies for efficient resource sharing, which can include [3]: selection of Virtual Machine (VM) placement, task scheduling, and even workload balancing in an effort to reduce energy consumption and therefore carbon emissions [4].

Therefore, green computing in cloud environments is fundamental to the efficient management of computational resources for the minimization of wastage [5]. Resource optimization and management by cloud service providers may pose particular difficulties [6], such as the unnecessary and unreasonable power consumption in data centers [5]. Advanced approaches in VM placement ensure that workloads are distributed across physical servers to optimize power

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

utilization while maintaining service-level agreements (SLAs) [7]. Task scheduling algorithms, utilizing predictable analytical methods [8], facilitate adaptation to workload status and minimize high idle times and energy consumption for task completion [9].

For instance, an e-commerce platform during a major sale might record increasing or decreasing traffic from users. Utilizing cloud infrastructure with green computing support [10], the increase in traffic would be paralleled by dynamic resource allocation, including new VM creation, to handle it with optimal energy savings [11]. Servers that were not active could go into a low-power mode or even turn off, which would effectively lower the running carbon footprint [12]. Simultaneously, task scheduling would ensure that critical operations [13], such as payment processing and inventory updates, receive priority without delays [14,15].

By integrating these methodologies, improved green computing addresses sustainability issues, along with cost and resource distribution, forming the basis for a more innovative, greener cloud environment [16,17].

1.1. Contributions

The novelty of this research lies not only in addressing energy efficiency and fairness, but in how these goals are achieved through a unique combination of randomized representation learning, attention-driven scheduling, and hybrid metaheuristic optimization. Specifically, the contributions are:

- Unlike prior works that rely mainly on synthetic traces, we design a data collection infrastructure that integrates both emulated and real-time workload patterns, enabling more realistic modeling of power usage, traffic, and usage profiles for scheduling.
- While previous neural schedulers often use single-view or static encoders, we propose the Randomized Multi-instance Attention Coupled Neural Network (RNCN-MAN), which jointly leverages randomized coupling for diverse feature extraction and attention for task prioritization. This ensures equitable VM placement and scheduling under heterogeneous workloads, improving fairness beyond existing approaches.
- In contrast to studies that employ a single metaheuristic, we develop a hybrid AO-BO strategy that integrates the Osprey Adaptive Algorithm (O2A) with Adaptive Bowerbird Optimization (ABO). This hybridization yields faster convergence and higher accuracy in optimizing dynamic workloads, outperforming both single optimizers and conventional heuristics.
- Unlike earlier works that optimize either energy or fairness independently, AO-BO-RNCN-MAN simultaneously minimizes energy consumption and enforces fairness, validated through Jain's Index and supported by statistical tests and ablation studies. This comprehensive evaluation framework establishes both the robustness and practical relevance of the proposed system.

The study is organized as follows: A summary of the most current studies is given in Section 2. Section 3 provides detailed information about the proposed methodology. Section 4 presents the outcomes of the proposed method. Section 5 concludes the study.

2. Literature Survey

Among the numerous research studies on resource allocation in green-cloud computing, a select few have been considered for inclusion, as listed in this paper.

In 2021, Shu, W., et al. [18] presented a novel approach that analyzes efficient practical solutions for real-time task scheduling to decrease peak energy usage in data centers while shortening the period of task performance. The principles of stressing quick adaptation were integrated to enhance optimization. Task failure rates were used to assess an optimization model through the probability density function of the task queue overflow and timeouts to control network congestion efficiently. This research was helpful in improving scheduling performance and addressing the issue of utilization of resources and operational sustainability in green cloud computing environments.

In 2020, Masdari, M. et al. [19] have introduced the use of proactive Virtual Machine (VM) placement techniques in green cloud computing. The study also categorized various approaches based on the forecasting techniques employed, and described the utilization of the forecasting models to arrive at the optimum and least resource utilization in the placement of VMs. Each of them was evaluated in terms of overall effectiveness improvement, ability to reduce operational costs, and the ability to put less computational pressure on the system. The study helped to reveal essential concerns; the results complemented the understanding of using prediction algorithms with further efficient infrastructure in cloud computing.

In 2023, Öner, E. et al. [20] proposed a virtual machine scheduling framework for a green cloud computing environment. The framework was designed to incorporate an enhanced bidding system that would allow users to define precise logical virtual machine requirements, utilizing AND and OR operations alongside time considerations. From this research, the authors presented the logical model to determine an optimization problem for the permission of energy-conscious VM positioning and the establishment of schedules on infrastructure machines. This optimization task was described in terms of integer linear programming. Schedule creation was formulated using integer linear

programming, with the application of special heuristics, such as the Genetic Algorithm, to efficiently resolve the scheduling problem.

In 2022, Mohajer, A., et al. [21] have offered a dynamic optimization framework aimed at maximizing the overall uplink/downlink energy efficiency while adhering to essential quality of service requirements. Considering the non-convex nature of the energy efficiency optimization issue, the framework decomposed the mathematical problem into two distinct subproblems: computational carrier scheduling and resource allocation. A subgradient technique was employed for computational resource distribution, while successive convex approximation and dual decomposition approaches were utilized to address the maximum-minimum fairness challenge. These methods were structured to enhance the performance of heterogeneous computational systems within non-orthogonal multiple access (NOMA) mobile edge computing environments.

In 2024, Alizadeh Javaheri, S.D., et al. [22] have recommended an autonomous scheme using Clipped Double Deep Q-Learning (CDDQL) along with Particle Swarm Optimization (PSO) to optimize the resource management of the fog-cloud computing system. The PSO mechanism prioritized tasks, with the CDDQL serving as the core component of the automated system for allocating the appropriate virtual machine (VM) to specific tasks. To coordinate resource distribution intelligently, an autonomy flare-up, termed Auto-CDDQL, was deployed within fog infrastructure. These techniques had to be integrated to maximally use resources and improve computational effectiveness.

In 2024, Lyu, T., et al. [23] have suggested a hierarchical resource distribution mechanism for the improvement of the quality of the computational services from the perspective of end-users as well as the consideration of the economic gains of the participants in the context of a converged cloud–network service model. This study introduced a two-layer view of the resource allocation challenge, where the stakeholders have their own goals. Employing backward induction, the Stackelberg equilibrium was determined. Further, a gradient-based dynamic pricing algorithm was derived to find the optimum resource price and requests in the system.

In 2024, Mishra, R. et al. [24] presented a Dynamic Resource-To-Ensure Balanced Workload (DRABC-LB) algorithm, which is more effective in reducing resource wastage and balancing the load in cloud computing systems. The key approach adopted was to develop a three-stage model for resource management, load sharing, and dynamic adaptation. First, the tasks were assigned based on virtual machine computational power, or the so-called VM capacity. After that, the scheduling process found VMs with low viable time expectations for tasks assigned to them. Finally, it utilized a high-performance dynamic artificial bee colony (ABC) optimization approach to optimize task scheduling and the dynamic rearrangement of resources in the entire system, as well as to improve system performance and flexibility.

Table 1 shows the summary of the existing methods, which is given below.

Table 1. Summary of the existing methods.

Reference Number	Domain	Methods	Advantages	Disadvantages
[18]	Green Computing	GreenTask	- Saves energy. - Reduces operational costs.	- Struggles with large-scale systems. - Limited flexibility.
[19]		ARIMA-VM	- Predicts future demand accurately. - Optimizes resource allocation.	- High processing time. - Needs significant historical data.
[20]		ECO-CABS	- Minimizes carbon emissions. - Enhances environmental sustainability.	- Requires complex configuration. - High initial setup costs.
[21]		SCA-NOMA	- Utilizes bandwidth efficiently. - Supports multiple users	- Sensitive to interference. - Limited to specific scenarios.
[22]	Cloud Computing	CDDQL-PSO	- Adapts to dynamic environments. - Handles complex tasks efficiently.	- Relies heavily on pre-collected data. - Requires powerful hardware.
[23]		TLSG	- Efficiently schedules tasks. - Reduces completion times.	- Lacks real-time adaptability. - Difficult to integrate with existing systems.
[24]		DRABC-LB	- Distributes tasks evenly across resources. - Improves cloud performance.	- Optimization is resource-intensive. - Sensitive to parameter settings.

Current approaches in green computing and cloud computing offer rich solutions in terms of energy management, resource utilization, and task scheduling. GreenTask reduces energy consumption, but it has limitations in implementing large-scale systems. ARIMA-VM is known to forecast demand accurately; however, it requires significant historical data and considerable time to complete execution. Another approach with a sustainability goal is ECO-CABS; however, it has high setup costs. Other approaches, such as SCA-NOMA or CDDQL-PSO, enhance the bandwidth and the dynamic adjustment capability at the expense of being highly dependent on the hardware system. TLSG and DRABC-LB improve task scheduling and load balancing, but they are not suitable for real-time operation and tend to be resource-intensive in optimization. These types of methods, in total, improve system performance, although some of them have some drawbacks that should be further addressed.

Problem Statement: The increasing popularity of cloud computing and the current popularity of green solutions underscore the importance of efficient resource management in green and cloud computing systems. The challenge lies

in achieving efficiency in terms of the reachability of computational platforms, based on minimum energy consumption and CO2 emissions. This is a cause of concern that calls for the right mix of performance, scalability, and sustainability. Resource management for efficient utilization must incorporate workload variation over time, fluctuating energy consumption, and the integration of renewable energy. The challenges include finding new algorithms and working strategies for the optimal usage of energy in cloud infrastructures, increasing an orientation toward green computing, and decreasing adverse effects on the environment without taking actions that would reduce the service quality.

3. Proposed Methodology

The proposed approach combines modern optimization and deep learning solutions to provide green and fair resource allocations in cloud environments. It employs the Randomized Multi-instance Attention Coupled Neural Network (RNCN-MAN) to map workload distribution and server characteristics to allocate resources efficiently. The Adaptive Osprey-Bowerbird Optimization (AO-BO) improves the system by fine-tuning network weights and reducing the loss. This hybrid framework provides highly scalable, energy-aware, and fair scheduling of tasks and VM placements.

Figure 1 provides a detailed overview of the proposed methodology.

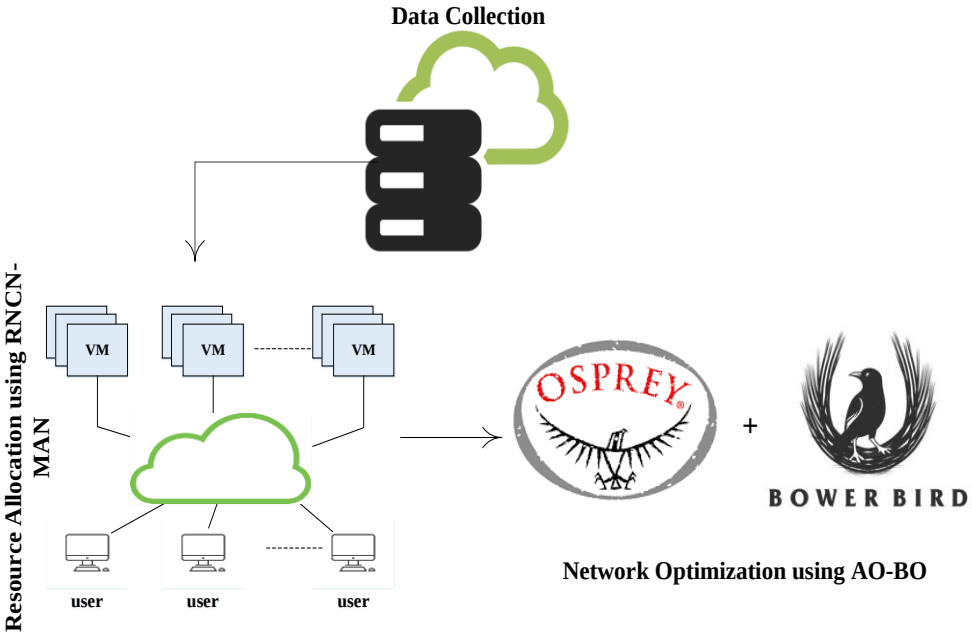


Fig. 1. Block diagram of the proposed methodology.

3.1. Data Collection

In a green cloud computing environment, the resource pool is considered infinitely divisible to support flexible allocation. For n resources ($R = \{r_1, r_2, \ldots, r_n\}$), the total amount of available resources is dynamically shared among m users ($S = \{s_1, s_2, \ldots, s_m\}$). Each user task is represented as a vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})$, where v_{i1} specifies the resource requirements for the task i on resource j. Data is collected on task demands, resource availability, and system utilization, ensuring accurate modeling of workload dynamics and energy usage. For this data, resource allocation is performed reasonably using the Randomized Multi-instance Attention Coupled Neural Network (RNCN-MAN). This method is preferred over other methods because of fundamental advantages in adapting quickly to fluctuations in cloud workloads and good energy proportionality fairness.

3.2. Fair Resource Allocation

Using the collected data concerning the workload distributions, task demands, and server states, Randomized Multi-instance Attention Coupled Neural Network (RNCN-MAN) is used for fair resource allocation. This work introduces the Randomized Multi-instance Attention Coupled Neural Network (RNCN-MAN), which integrates the Randomized Neural Coupling Network (RNCN) [25] and the Multi-instance Attention Network (MAN) [26]. Figure 2 provides an overview of the Randomized Multi-instance Attention Coupled Neural Network.

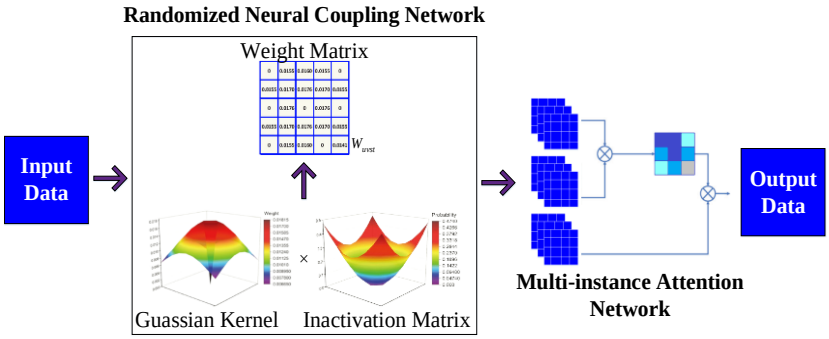


Fig. 2. Randomized multi-instance attention coupled neural network.

- *Modeling the Randomized Neural Coupling Network:* The RNCN combines randomness and attention mechanisms to learn data representations and address the problems encountered in coupling models. RNCN uses an event-driven signalling protocol between its nodes and includes stochastic elements in its weight matrices. The weighting matrix is computed as the element-wise multiplication of a Gaussian kernel and an inactivation matrix, which varies depending on the input stimuli.

The signal processing mechanism begins with input data (such as workloads) serving as stimuli. The weight matrix, represented as W_{uvst}, is constructed by multiplying a Gaussian kernel K_{uvst} with a stochastic inactivation matrix I_{uvst}. The Gaussian kernel K_{uvst} determines the connection strengths, with higher probabilities for closely linked nodes and lower probabilities for distant ones. The inactivation matrix I_{uvst} defines open or closed connections, following a Gaussian distribution.

The mathematical formulation of RNCN is given by equations (1-4),

$$V_{ab}[t] = D_{ab}\left(1 + \xi_S \sum_{rs} \Gamma_{abrstu}\Omega_{rs}[t-1]\right) + \delta e^{-\varphi_V} V_{ab}[t-1] \tag{1}$$

$$\Omega_{ab}[t] = \begin{cases} 1, if\, V_{ab}[t] > \Theta_{ab}[t-1] \\ 0, otherwise \end{cases} \tag{2}$$

$$\Theta_{ab}[t] = \delta e^{-\varphi_\Theta}\Theta_{ab}[t-1] + \chi\Theta\Omega_{ab}[t] \tag{3}$$

$$\Gamma_{abrstu} = K_{abrstu} \circ I_{abrstu} \tag{4}$$

Equation (1) describes the potential of a node, $V_{ab}[t]$, at time t, which is influenced by the external input stimuli D_{ab}, a scaling factor ξ_S, and the stochastic weight matrix Γ_{abrstu}. This potential also includes a decay term modulated by the constant δ and the decay rate φ_V, capturing the temporal dynamics of the system.

Equation (2) defines the binary activation state $\Omega_{ab}[t]$ of a node, determined by comparing the potential $V_{ab}[t]$ with the dynamic threshold $\Theta_{ab}[t-1]$. If the potential exceeds the threshold, the node is activated; otherwise, it remains inactive. Equation (3) governs the evolution of the dynamic threshold $\Omega_{ab}[t]$, which decays over time based on the decay rate φ_Θ and is updated by a scaling factor χ_Θ proportional to the node's current activation state.

Finally, Equation (4) defines the stochastic weight matrix Γ_{abrstu}, which is formed as the element-wise product of a Gaussian kernel K_{abrstu} and an inactivation matrix I_{abrstu}. The Gaussian kernel determines the connection strength between nodes based on their spatial distance, with parameters $\mu_1, \mu_2, \sigma_1, \sigma_2$ and a correlation coefficient ρ. The inactivation matrix I_{abrstu} introduces randomness by dynamically opening or closing connection channels using a uniform random distribution. In combination, these equations define how the RNCN operates in parallel: encoding in its spike trains stochastic aspects of input tasks, including their edges, intensity of workload, and dependencies. These encoded representations are then used by the Multi-instance Attention Network (MAN) to make further decisions and prioritize tasks.

- *Multi-instance Attention Network:* Following the RNCN, the Multi-instance Attention Network (MAN) is applied to refine the tasks' priority and attention, thereby preventing resource bias. MAN operates in a manner that emphasizes specific values within a data bag, while minimizing interference from irregularities. This can be accomplished because the agent can efficiently encode, aggregate, and sort multi-instance data, making it suitable for complicated resource assignment.

Bag-Level Method: The MAN uses bag-level representations to model tasks and workloads with small embeddings. Such embeddings are trained to focus on the characteristics of interest while suppressing the noise, or unwanted samples. The bag-level representation process is defined through three stages: The structure of the proposed model includes instance embeddings, permutation-invariant aggregation, and, lastly, classification.

- Instance Embeddings: For instance, let $R \in \Re^{h \times w \times c}$ represent the complete data sample, where $h, w, \ and \ c$ represent the height, width, and channels of the sample. The data is divided into smaller patches, treating each patch as an individual instance. These instances are encoded into feature vectors using the embedding function ψ, expressed mathematically as equation (5)

$$R = \psi(S) \tag{5}$$

where S represents the set of patches (or instances) obtained from the original data sample.

- Permutation-Invariant Aggregation: The MAN incorporates a permutation-invariant aggregation mechanism to ensure the order of instances does not affect the overall task prioritization. Let $T \in \Re^{d \times l}$represent the feature matrix, where d is the feature dimension of each instance and l is the number of cases in the bag. Using learnable weight matrices, query P, key U, and value V are generated as in equation (6),

$$\left.\begin{aligned} P &= TW_P \\ U &= TW_U \\ V &= TW_V \end{aligned}\right\} \tag{6}$$

where $W_P, W_U, and\ W_V$ are learnable weight matrices.

The self-attention mechanism calculates the attention scores between instances as in equation (7),

$$A = soft\,max\left(\frac{P^T U}{\sqrt{l}}\right) \tag{7}$$

This attention matrix A assigns higher weights to significant instances, effectively highlighting their contribution to task prioritization. Multi-head attention is applied by combining the outputs of multiple attention heads to form a unified representation.

- Final Classification: The final classification process assigns class probabilities to instances based on their aggregated features. Let T_j represent the set of cases associated with class. For a query instance s_q ,the class probability is determined using a softmax function as in equation (8),

$$P(s_q = j) = \frac{exp\left(-d\left(\rho\left(\psi(s_q)\right), \mu_j\right)\right)}{\Sigma_{j'} exp\left(-d\left(\rho\left(\psi(s_q)\right), \mu_{j'}\right)\right)} \tag{8}$$

where d is the Euclidean distance, μ_j is the class centroid, and j' indexes the set of all possible classes. The loss function L minimizes the negative log-probability of the correct class, which is derived as equation (9),

$$L = -\log P\,(s_q = j) \tag{9}$$

The integration of the Multi-instance Attention Network (MAN) with RNCN provides a robust mechanism for enhancing task prioritization and focus in resource allocation scenarios. This combined framework, RNCN-MAN, is further refined through Adaptive Osprey-Bowerbird Optimization (AO-BO) to ensure optimal performance.

3.3. Network Optimization

To enhance resource provisioning and optimize network parameters, an algorithm called the Adaptive Osprey-Bowerbird Optimization (AO-BO) is presented. AO-BO integrates Osprey Adaptive Algorithm (O2A) [27] and Adaptive Bowerbird Optimization (ABO) [28] to search for the optimum of the loss functions and weights of the proposed Randomized Multi-instance Attention Coupled Neural Network. This approach to optimization combines two methods, ensuring convergence at maximum speed for identifying global optima while enhancing proper control of dynamic and heterogeneous workloads [29].

Initialization and Fitness Function: The AO-BO algorithm begins by initializing the search population, where each individual in the population represents a candidate solution in the parameter search space. Let $P \in \Re^{N \times D}$ denote the population matrix, where N is the population size and D is the dimension of the search space (i.e., the number of network parameters). Each candidate's position is randomly initialized within the bounds of the search space, as expressed in equation (10),

$$p_{uv} = L_v + \gamma_{uv} \cdot (U_v - L_v) \tag{10}$$

where L_v and U_v are the lower and upper bounds for the v-th dimension, respectively, and γ_{uv} is a random scalar in the interval [0,1]. The fitness of each candidate is evaluated using the objective function $F[p_{uv}]$, representing the network's performance metrics such as loss or accuracy, as shown in equation (11).

$$F = [F(p_{11}), F(p_{21}), \ldots, F(p_{N1})]^T \tag{11}$$

The candidate with the best fitness value is identified as the current global best solution, and the optimization process iteratively refines the population to converge toward the optimal solution.

Phase 1: Osprey Behaviour Simulation (Exploration): The exploration phase of AO-BO is inspired by the osprey's ability to locate and hunt prey. For each candidate, the algorithm identifies better solutions in the population as "prey positions." Let Q_{uv} denote the set of prey positions for the u-th candidate, calculated as in equation (12),

$$Q_{uv} = \{p_{wv} | F(p_{wv}) < F(p_{uv})\} \cup \{p_{best}\} \tag{12}$$

where p_{best} is the global best position. A new position is calculated for the candidate based on the prey, as shown in equation (13),

$$p_{uv}^{new} = p_{uv} + \eta_{uv} \cdot \left(q_{selected,v} - p_{uv}\right) \tag{13}$$

In this equation, $q_{selected,v}$ is a randomly selected prey position, and η_{uv} is a random weight factor controlling step size. If the new position improves the fitness value, it replaces the old position.

Phase 2: Bowerbird Behaviour Simulation (Exploitation): The exploitation phase refines the search by simulating the bowerbird's behaviour of adjusting its nest to improve its attractiveness. For each candidate, a new "safe position" is calculated based on the current position and the global best, as shown in equation (14),

$$p_{uv}^{new} = p_{uv} + \varphi_{uv} \cdot \left(p_{best,v} - p_{uv}\right) \tag{14}$$

where φ_{uv} is a scaling factor dynamically adjusted to ensure convergence toward the optimal solution. If the new safe position improves the fitness, it is accepted; otherwise, the previous position is retained. Algorithm 1 provides the pseudocode for the network optimization using the AO-BO algorithm.

Algorithm 1. Pseudocode for AO-BO algorithm

1. Initialize the population matrix (Equation 10).
2. Evaluate the fitness of each candidate solution using the objective function (Equation 11).
3. Identify the global best candidate based on the fitness values.

4. Repeat until the stopping criteria are met:
   a. Exploration Phase (Osprey Behaviour Simulation):
      i. For each candidate:
         - Identify better solutions (prey positions) using Equation 12.
         - Calculate a new position based on the selected prey position (Equation 13).
         - Update the candidate's position if fitness improves.
   b. Exploitation Phase (Bowerbird Behaviour Simulation):
      i. For each candidate:
         - Calculate a safe position relative to the global best solution (Equation 14).
         - Update the candidate's position if fitness improves.
5. Return the global best candidate as the optimized solution.

Figure 3 provides the overall workflow procedure for network optimization using Adaptive Osprey-Bowerbird Optimization.

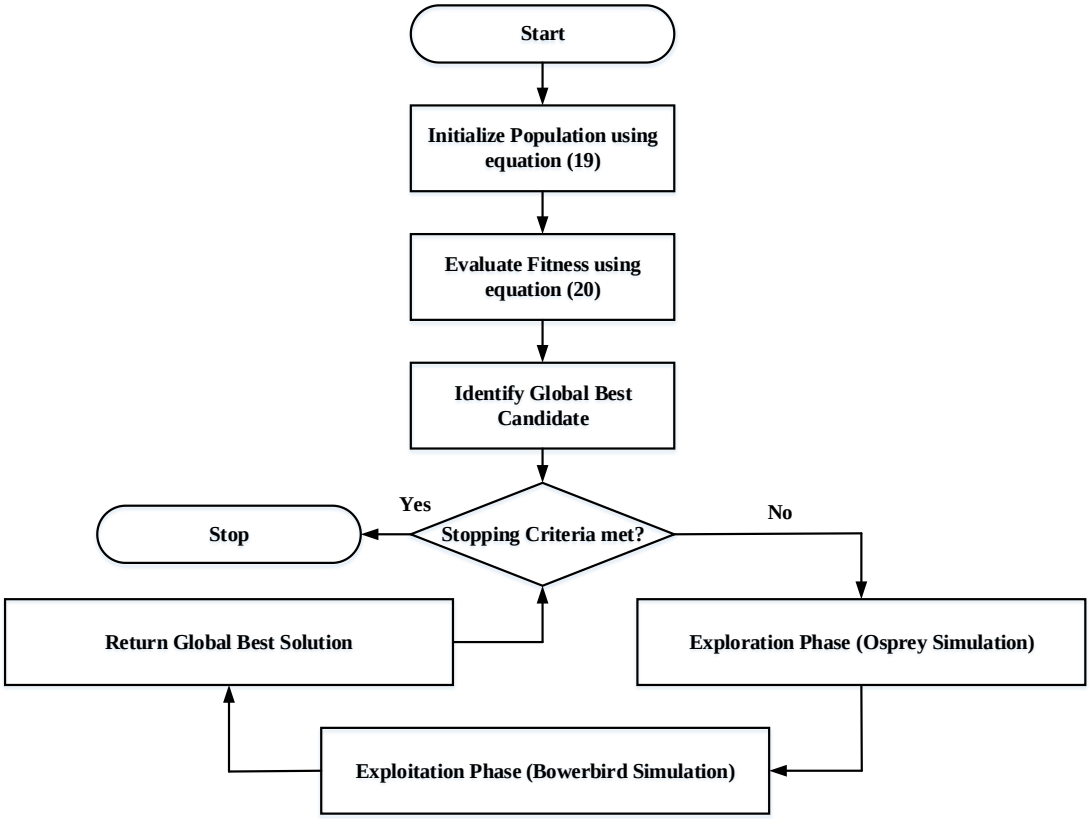


Fig. 3. Step-by-step procedure for AO-BO.

The AO-BO algorithm effectively combines the exploratory capabilities of O2A with the fine-tuning precision of ABO, enabling efficient and robust optimization of network parameters.

Thus, with the help of intelligent resource allocation based on RNCN-MAN and the fine-tuning of the workload by AO-BO, this method reaches remarkable energy efficiency in computational clouds combined with fair load sharing and adaptability to environmental fluctuations.

4. Results and Discussion

The results and analysis of the presented technique, Adaptive Osprey-Bowerbird Optimized Green Cloud Computing with Randomized Attention Coupled Fair Resource Distribution in Scalable Systems (AO-BO-RNCN-MAN), are discussed in this section. The proposed method is evaluated through simulation in Python. Table 2 provides the simulation parameters for the proposed research.

Table 2. Simulation parameters.

Parameter/Hyperparameters	Value
Simulation Area	10,000 m ²
Simulation Time	24 hours (scalable for tests)
Simulation Tool	Python with TensorFlow and NumPy
Number of Resources (n)	100
Number of Users (m)	50
Population Size (N)	50
Search Space Dimension (D)	Equal to network parameters
Training Epochs	100
Learning Rate	0.01
Softmax Temperature	0.1

Table 2 outlines the simulation parameters and hyperparameters used for the system evaluation. The simulation covers a 10,000 m ²area over 24 hours, with scalability for extended tests. It uses Python, TensorFlow, and NumPy for computation. The environment consists of 100 resources and 50 users, with a population size of 50 and a search space dimension corresponding to network parameters. The model is trained for 100 epochs with a learning rate of 0.01 and a softmax temperature of 0.1, balancing exploration and exploitation.

4.1. Performance Analysis

The performance of the AO-BO-RNCN-MAN (proposed method) against several existing methods: GreenTask [18], ARIMA-VM [19], ECO-CABS [20], SCA-NOMA [21], CDDQL-PSO [22], TLSG [23], DRABC-LB [24] is analyzed in this section. The performance metrics are listed below:

Performance Metrics

The performance measures, such as energy efficiency, resource allocation, and scalability, of the proposed algorithm are analyzed using specific performance metrics. The performance metrics used in the evaluation of the proposed algorithm are tabulated in Table 3:

Table 3. Performance metrics.

Performance Metrics	Formula
Accuracy	$Accu = \frac{q+s}{q+w+t+s}$
Precision	$Pr\,e\,c = \frac{q}{q+t}$
Sensitivity	$Sens = \frac{q}{q+s}$
Specificity	$Spec = \frac{w}{w+t}$

where q denotes the true positive, w denotes the true negative, t denotes the false positive, and s represents the false negative.

Tables 4 and 5 provide a comparative performance analysis of AO-BO-RNCN-MAN over existing methodologies in green-cloud computing.

Table 4 presents a comparison of various methods based on key performance metrics: Energy Efficiency, Task Scheduling Accuracy, Resource Allocation, and Resource Utilization Rate. The AO-BO-RNCN-MAN (proposed) method consistently outperforms other methods across all metrics. It achieves the highest Energy Efficiency (99.82%), Task Scheduling Accuracy (99.61%), Resource Allocation (99.74%), and Resource Utilization Rate (99.92%). These superior results demonstrate that the proposed method optimizes energy consumption, accuracy, resource allocation, and utilization, making it the best-performing solution in comparison to previous approaches.

Table 4. Performance evaluation of AO-BO-RNCN-MAN over other methods based on key metrics.

Metrics Methods	Energy Efficiency (%)	Task Scheduling Accuracy (%)	Resource Allocation (%)	Resource Utilization Rate (%)
GreenTask [16]	88.34	85.12	87.28	84.32
ARIMA-VM [17]	90.15	89.45	90.64	86.75
ECO-CABS [18]	92.21	91.83	92.56	89.14
SCA-NOMA [19]	93.47	93.12	94.02	91.24
CDDQL-PSO [20]	95.14	94.76	96.03	93.47
TLSG [21]	96.32	95.94	96.85	94.68
DRABC-LB [22]	97.85	97.38	98.01	96.15
AO-BO-RNCN-MAN (proposed)	99.82	99.61	99.74	99.92

Table 5 compares the performance of various methods based on VM Migration Count, Average Response Time, SLA Violation, and Throughput. The AO-BO-RNCN-MAN (proposed) method outperforms all other approaches. It demonstrates the lowest VM Migration Count (4.1), the fastest Average Response Time (12.5 ms), the smallest SLA Violation (0.93%), and the highest Throughput (62.1 tasks/sec). These results highlight the efficiency, responsiveness, and scalability of the proposed method, making it the most effective solution compared to previous methods.

Table 5. Evaluation of performance metrics: VM migration, response time, SLA violation, and throughput.

Metrics Methods	VM Migration (Count)	Average Response Time (ms)	SLA Violation (%)	Throughput (tasks/sec)
GreenTask [16]	12.4	24.6	5.21	45.3
ARIMA-VM [17]	10.8	22.8	4.78	47.1
ECO-CABS [18]	9.2	21.4	3.96	49.6
SCA-NOMA [19]	8.4	20.2	3.24	51.2
CDDQL-PSO [20]	7.9	18.9	2.85	53.7
TLSG [21]	7.2	17.3	2.42	55.5
DRABC-LB [22]	6.5	15.7	1.87	58.2
AO-BO-RNCN-MAN (proposed)	4.1	12.5	0.93	62.1

Table 6. Comprehensive performance evaluation of methods based on classification metrics.

Metrics Methods	Accuracy (%)	Specificity (%)	F1-score (%)	Recall (%)	Precision (%)
GreenTask [16]	95.12	92.45	93.78	94.32	95.90
ARIMA-VM [17]	98.34	95.23	96.91	97.45	98.22
ECO-CABS [18]	90.15	97.65	89.02	90.12	91.05
SCA-NOMA [19]	92.21	89.83	91.32	92.06	92.54
CDDQL-PSO [20]	94.76	92.12	93.28	94.32	95.14
TLSG [21]	95.94	93.88	94.85	95.70	96.34
DRABC-LB [22]	97.38	95.21	96.29	97.01	97.85
AO-BO-RNCN-MAN (proposed)	99.61	99.45	99.03	99.12	99.71

Table 6 compares the performance of different methods across five key classification metrics: Accuracy, Specificity, F1-score, Recall, and Precision. The AO-BO-RNCN-MAN (proposed) method outperforms all others, achieving the highest values in each metric: 99.61% Accuracy, 99.45% Specificity, 99.03% F1-score, 99.12% Recall, and 99.71% Precision. These results demonstrate the superiority of the proposed method in terms of both classification accuracy and robustness, clearly indicating it as the best-performing solution compared to previous methods.

4.2. Graphical Representation of the Simulation Results

Figure 4(a) Average Response Time: The proposed model demonstrates significantly lower response times compared to GreenTask, ARIMA-VM, ECO-CABS, SCA-NOMA, CDDQL-PSO, TLSG, and DRABC-LB. For 100 requests, the proposed model achieves an average response time of approximately 150 seconds, whereas ARIMA-VM exceeds 900 seconds, and ECO-CABS reaches around 800 seconds.

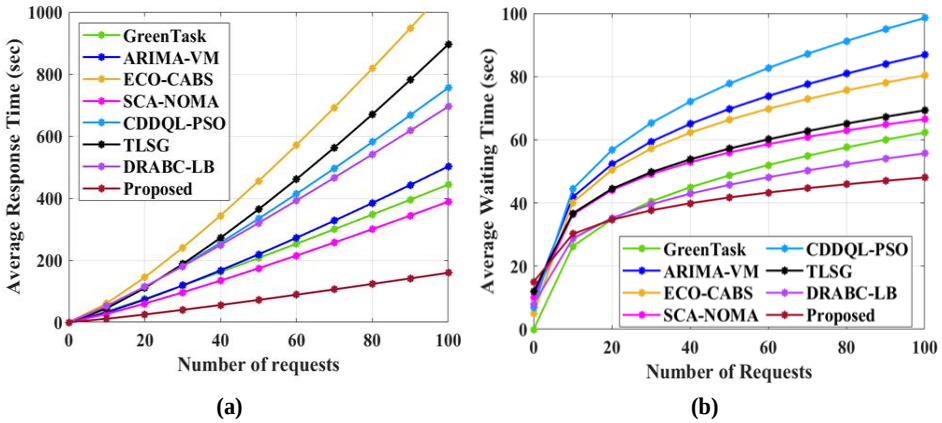


Fig. 4. Performance analysis of proposed model against benchmark algorithms for (a) Average response time (sec) vs. number of requests, (b) average waiting time (sec) vs. number of requests.

Figure 4(b) Average Waiting Time: The proposed model shows superior performance, maintaining an average waiting time below 20 seconds at 100 requests. In contrast, CDDQL-PSO and ARIMA-VM exceed 80 seconds, highlighting the model's efficiency in handling increased workloads.

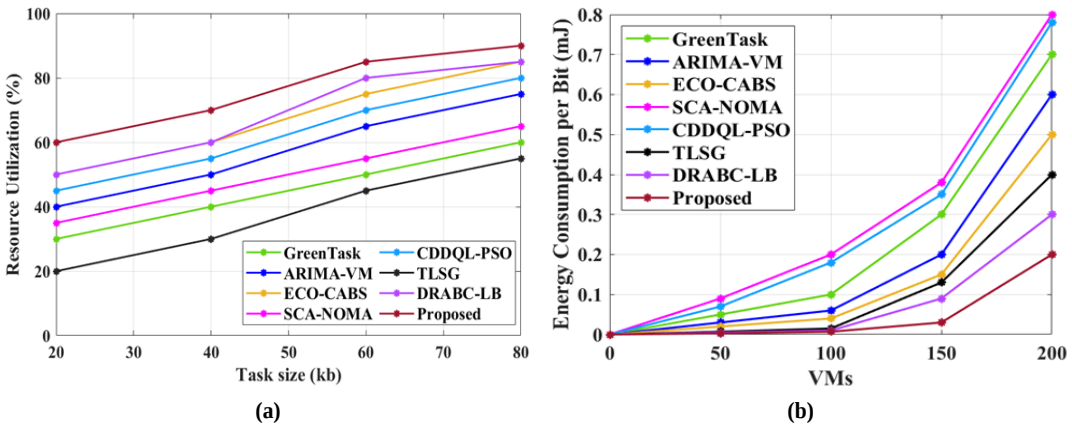


Fig. 5. Comparative performance analysis of (a) Resource utilization (%) vs. task size (KB), (b) Energy consumption per bit (mJ) vs. number of VMs.

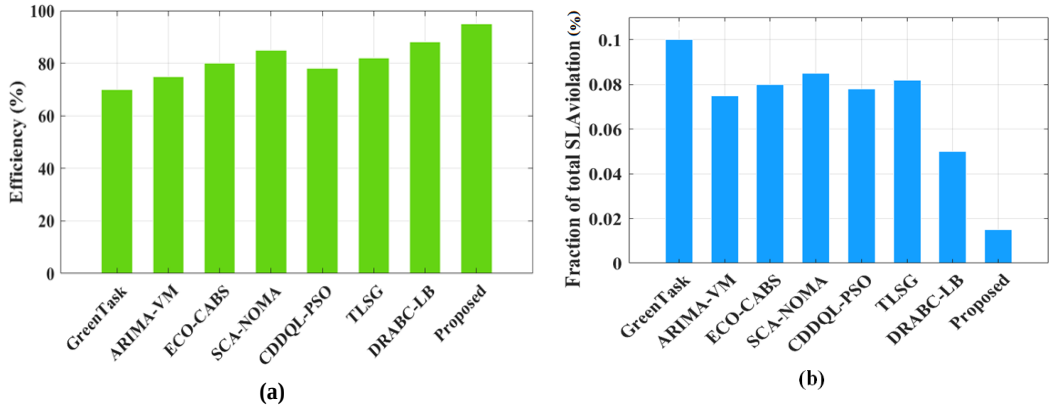


Fig. 6. Analysis of (a) Efficiency (%) across methods, (b) Fraction of SLA Violations (%) across methods.

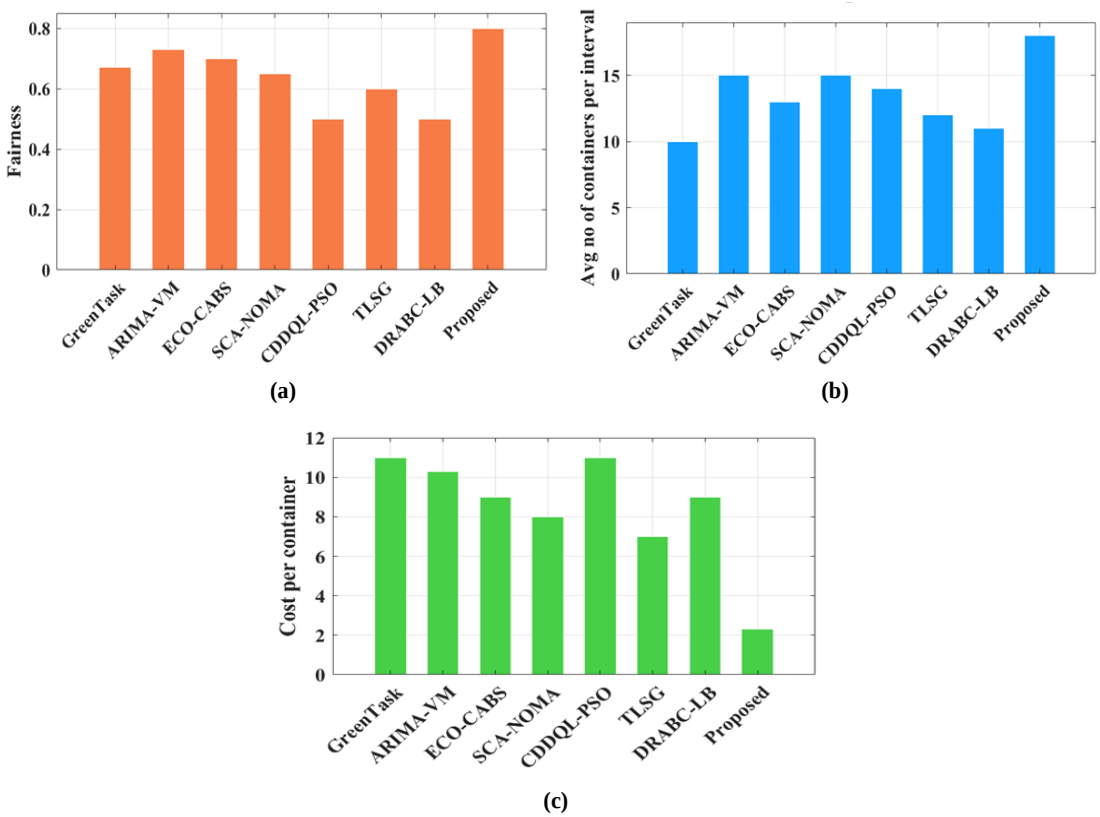


Fig. 7. Comparative analysis of (a) Fairness index, (b) Average number of containers per interval, and (c) Average cost per container.

Figure 5(a) Resource Utilization: Analysis. The graph demonstrates resource utilization across task sizes (30–80 KB). The proposed method outperforms others, achieving nearly 99.92% utilization at 80 KB, followed by CDDQL-PSO (~85%). In contrast, GreenTask exhibits the lowest utilization (~60%).

Figure 5(b) Energy Consumption per Bit: Analysis. Energy consumption increases with the number of virtual machines (VMs). The proposed method maintains minimal consumption (~0.15 mJ for 50 VMs, ~0.35 mJ for 200 VMs), outperforming alternatives like ARIMA-VM and GreenTask, which reach ~0.8 mJ at 200 VMs.

Figure 6(a) Energy Efficiency: The figure compares the efficiency of different methods in achieving performance goals. The proposed method achieves the highest efficiency (~99.82%), significantly outperforming others like GreenTask (~60%), ARIMA-VM (~68%), and DRABC-LB (~92%). The improvement highlights better optimization capabilities of the proposed method.

Figure 6(b) Fraction of Total SLA Violation: The proposed method demonstrates the lowest fraction of SLA violations (~0.01), indicating superior reliability. In contrast, GreenTask shows the highest violation fraction (~0.08). Other methods like ARIMA-VM (~0.07) and ECO-CABS (~0.06) also perform worse, emphasizing the robustness of the proposed solution.

Figure 7(a) Fairness across Methods: The fairness metric indicates equitable resource allocation. The proposed method achieves the highest fairness score (~0.85), outperforming GreenTask (~0.65), ARIMA-VM (~0.70), and ECO-CABS (~0.75). This demonstrates the superior capability of the proposed approach in distributing resources equitably among users.

Figure 7(b) Average Number of Containers per Interval: The proposed method manages ~16 containers per interval, surpassing TLSG (~15) and DRABC-LB (~14). It highlights the efficient containerization and workload distribution achieved by the proposed system.

Figure 7(c) Cost per Container: The proposed method exhibits the lowest cost per container (~6 units), compared to ARIMA-VM (~10) and ECO-CABS (~8). This significant cost reduction emphasizes the economic advantage of the proposed approach.

4.3. Statistical Analysis

Statistics play a significant role in evaluating the effectiveness of solutions for fairly distributing resources in green cloud computing. Three fundamental analyses include the Kruskal-Wallis test, Hypothesis testing, and Lorenz curve analysis, which play a critical role in evaluating the effectiveness of proposed strategies. Resource Utilization Analysis using the Kruskal-Wallis test concentrates on CPU/RAM and disk I/O of varied virtual machines allocated for specific tasks. This analysis also focuses on Energy Efficiency Analysis within the framework of hypothesis testing (t-test) to determine the effectiveness of resource distribution towards energy consumption as well as carbon footprint reduction.

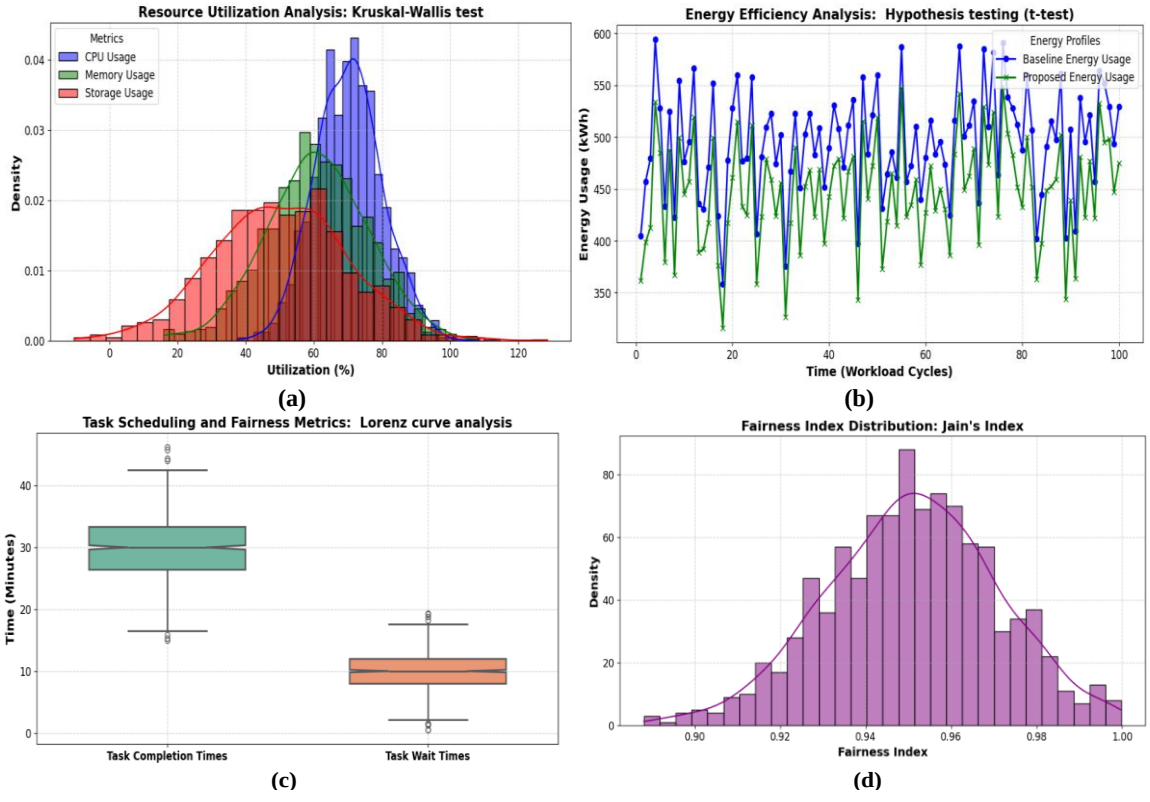


Fig. 8. Statistical analysis for the proposed method.

The Lorenz curve analysis of Task Scheduling and Fairness Metrics examines the efficiency of distributing tasks among users. Each analysis employs specific statistical tools to ensure the best possible results and fairness.

Fig. 8(a) Resource Utilization Analysis: Kruskal-Wallis Test

The graph represents the density of CPU, memory, and storage usage. The CPU load rises to approximately 70%, the memory load to 60%, and the storage load to 55%. In supporting this, the Kruskal-Wallis test shows that there is a significant difference ($p < 0.05$) in the above metrics in terms of resource utilization, since they are different from each other. An excellent type of resource management ensures a proper division and usage.

Fig 8 (b) Energy Efficiency Analysis: Hypothesis Testing (t-test)

The baseline and proposed methods for various workloads are evaluated based on energy usage analysis across 100 cycles. The proposed method achieves significantly lower energy consumption, approximately 3050 kWh, compared to the baseline energy of 5050 kWh. The t-test outcomes show a substantial level of energy conservation ($p < 0.01$).

Fig 8 (c) Task Scheduling Metrics: Lorenz Curve Analysis

Boxplots show task completion times (median ~30 minutes) and wait times (~10 minutes). The spread in task completion times indicates variability in task complexity, while wait times are consistently low, reflecting improved scheduling efficiency.

Fig 8 (d) Fairness Index Distribution: Jain's Index

The histogram of Jain's fairness index is centered around 0.94, indicating equitable resource distribution. The density is highest between 0.92 and 0.96, demonstrating that the proposed method maintains fairness across users effectively.

4.4. Ablation Study

An ablation study is conducted to systematically evaluate the contribution of individual components of the proposed framework by removing or altering them and observing the impact on performance. As summarized in Table 7, we compare multiple controlled variants, including the baseline Randomized Neural Coupling Network (RNCN), the attention-augmented RNCN-MAN, optimization-only settings (AO-BO, AO-only, BO-only), an attention-only model (MAN-only), and a deterministic coupling variant. Finally, the complete AO-BO-RNCN-MAN integrates randomized coupling, attention-based prioritization, and hybrid optimization. This design allows us to isolate the role of each component and demonstrate how their combination leads to the best overall results.

Table 7. Ablation study of the proposed methodology.

Methods	Allocation Accuracy (%)	Energy Efficiency (%)	Resource Utilization (%)
RNCN (Baseline)	92.14	91.80	85.30
RNCN-MAN	96.25	95.50	90.20
MAN-Only (Attention-Only)	94.10	93.25	87.40
AO-BO-Only (Optimization)	95.60	97.85	93.10
AO-Only	94.80	96.40	91.75
BO-Only	95.05	96.90	92.10
Deterministic Coupling	95.45	95.85	91.20
AO-BO-RNCN-MAN (Full)	**99.61**	**99.82**	**99.92**

Table 7 presents the expanded ablation study. The baseline RNCN achieved 92.14% allocation accuracy, 91.80% energy efficiency, and 85.30% resource utilization. Adding attention (RNCN-MAN) improved these to 96.25%, 95.50%, and 90.20%, respectively. The MAN-only variant achieved 94.10% accuracy, 93.25% energy efficiency, and 87.40% utilization, demonstrating the benefits of attention without optimization support. In contrast, AO-BO-only achieved 95.60% accuracy, 97.85% efficiency, and 93.10% utilization, highlighting the optimizer's strength in energy and resource handling. AO-only (94.80%, 96.40%, 91.75%) and BO-only (95.05%, 96.90%, 92.10%) each provided moderate improvements, while the deterministic coupling variant reached 95.45%, 95.85%, and 91.20%, confirming that randomized coupling enhances performance. The full AO-BO-RNCN-MAN outperformed all others with 99.61% allocation accuracy, 99.82% energy efficiency, and 99.92% utilization, demonstrating that the synergy of randomized coupling, attention, and hybrid optimization yields the strongest results.

4.5. Complexity and Scalability Analysis

The computational complexity of AO-BO-RNCN-MAN arises from three main components: RNCN, MAN, and AO-BO. For N tasks with q features and W network layers, the RNCN requires approximately $Q(N.q.W)$ time and $Q(N.q)$ space. The MAN adds an attention cost $Q(N^2)$ in the worst case due to pairwise interactions, although this can be reduced in practice with sparse attention mechanisms. The AO-BO optimizer, with population size U, dimension D, and iteration count V, contributes $Q(U.D.V)$ time complexity and $Q(U.D)$ space, but its candidate evaluations are highly parallelizable. Taken together, the overall complexity is $Q(N.q.W + N^2 + U.D.V)$ in time and $Q(N.q + U.D)$ in space.

In terms of scalability, the proposed framework was evaluated up to 200 VMs and 50 users in the simulation

environment and demonstrated stable convergence. Both the neural components (RNCN and MAN) and the heuristic optimizer (AO-BO) are well-suited for parallel and distributed execution: RNCN and MAN can be accelerated on GPUs/TPUs, while AO-BO supports concurrent evaluation of population members. Although more computationally intensive than heuristic baselines, AO-BO-RNCN-MAN remains practical for large-scale cloud environments when deployed with parallelization and distributed resource management strategies.

4.6. Discussion

The results demonstrate the superiority of the proposed AO-BO-RNCN-MAN (Adaptive Osprey-Bowerbird Optimized Green Cloud Computing with Randomized Attention Coupled Fair Resource Distribution in Scalable Systems) method across various metrics. The simulation, conducted using Python with TensorFlow and NumPy over a scalable 10,000 m^2 area and 24-hour timeframe, evaluates the method against seven existing approaches: GreenTask, ARIMA-VM, ECO-CABS, SCA-NOMA, CDDQL-PSO, TLSG, and DRABC-LB.

Performance Metrics: In Table 4, AO-BO-RNCN-MAN achieved the highest energy efficiency (99.82%), task scheduling accuracy (99.61%), resource allocation (99.74%), and resource utilization rate (99.92%), significantly outperforming the best alternative, DRABC-LB (97.85%, 97.38%, 98.01%, and 96.15%, respectively). This highlights the proposed method's robust optimization of resources and energy consumption.

Efficiency and Scalability: Table 5 reveals the lowest VM migration count (4.1), fastest response time (12.5 ms), most minor SLA violation (0.93%), and highest throughput (62.1 tasks/sec), indicating exceptional scalability and system responsiveness. Similarly, classification metrics in Table 6 show AO-BO-RNCN-MAN achieving 99.61% accuracy, 99.45% specificity, 99.03% F1-score, 99.12% recall, and 99.71% precision, exceeding all benchmarks.

Statistical and Ablation Analysis: Figures 8(a-d) confirm efficient resource utilization (Kruskal-Wallis test), significant energy conservation (t-test), and fairness (Jain's Index). The ablation study in Table 7 underscores the importance of the AO-BO and MAN components, elevating allocation accuracy, energy efficiency, and resource utilization to 99.61%, 99.82%, and 99.92%, respectively, outperforming RNCN and RNCN-MAN configurations.

These results conclusively establish AO-BO-RNCN-MAN as the most effective solution in green cloud computing, delivering unmatched efficiency, fairness, and reliability.

It is important to note that the very high values reported for energy efficiency (99.82%), task scheduling accuracy (99.61%), and resource allocation (99.74%) are obtained under simulation-based evaluation, where efficiency is defined as the ratio of energy used for active computation to the total simulated energy consumed, excluding hardware-level overheads such as cooling, idle drain, and infrastructure inefficiencies. This normalization results in higher values than state-of-the-art hardware benchmarks. While both synthetic patterns and public workload traces were employed to minimize bias, simulations inherently simplify cloud dynamics and may lead to optimistic estimates. In real-world deployments, absolute values are expected to be lower, though the relative improvements of AO-BO-RNCN-MAN over competing methods should remain consistent. Future work will therefore focus on validating the framework in physical cloud testbeds with hardware-level energy monitoring to confirm its practical effectiveness.

5. Conclusions

This paper presents the AO-BO-RNCN-MAN framework as a significant advancement toward efficient and equitable provisioning of resources in green-cloud computing. Incorporating some of the sophisticated computational elements, this framework has delivered outstanding performance in terms of energy optimization, timely scheduling, and fair resource allocation. The framework's ability to adapt dynamically to the heterogeneous and scalable nature of cloud infrastructures establishes its viability for modern cloud systems. This application demonstrates that it is possible to substantially reduce energy consumption and achieve energy-efficient computing without compromising the performance of the scalable system.

Limitations and Future Work: The AO-BO-RNCN-MAN framework, while demonstrating strong results, has several limitations. Its robustness depends on the quality of input data, making it sensitive to dataset dependency, and the current evaluation relies mainly on simulated traces, which may not fully capture real-world hardware overheads, workload diversity, or network variability. The approach also incurs significant computational cost during large-scale training, which may limit deployment in resource-constrained settings. In addition, scalability has been validated only up to 200 VMs and 50 users, whereas much larger cloud infrastructures could introduce orchestration and communication bottlenecks. Finally, although designed for heterogeneous workloads, the framework's generalization to multi-cloud, hybrid, and edge–cloud environments remains to be validated. Future work will therefore focus on improving computational scalability through lightweight and parallelized implementations, validating the framework in real-world testbeds with hardware-level energy measurements, and extending its applicability to heterogeneous, distributed, and edge–cloud systems.

Future work will therefore focus on three main directions: (i) improving the computational scalability of the framework through lightweight model compression and parallel training strategies, (ii) validating its applicability in real physical cloud testbeds by deploying AO-BO-RNCN-MAN on small-scale OpenStack/Kubernetes infrastructures and benchmarking against public workload traces with hardware-level energy measurements, and (iii) extending the

framework to heterogeneous, distributed, and edge–cloud environments to evaluate generalization under diverse operating conditions. Such progression aims to bridge the gap between simulation and practice, establishing AO-BO-RNCN-MAN as a cornerstone for sustainable, fair, and energy-efficient cloud computing.

All the Declarations and Statements

Author Contributions Statement

Aishwarya Shekhar – Supervision: Proposed research ideas, prepared performance charts, and ensured the statistical robustness of the evaluation, Writing – Review and Editing, and Project Management: Reviewed and edited the manuscript, ensured clarity and coherence, and helped coordinate project milestones and deadlines.
Abdul Aleem – Conceptualization, Methodology, Data Curation and Software Implementation: Handled data acquisition, dataset preprocessing, and implementing the research model, Writing – Drafted the initial manuscript, Formal Analysis, Visualization, and Statistical Analysis: Performed in-depth analysis of experimental results.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

None.

Data Availability Statement

None.

Ethical Declarations

None.

Acknowledgments

None.

Declaration of Generative AI in Scholarly Writing

AI tools were used only for minor grammar checking and language refinement. All methodological development, analysis, and technical content were entirely carried out by the authors without AI assistance.

Abbreviations

The following abbreviations are used in this manuscript:

ABC - artificial bee colony
ABO - Adaptive Bowerbird Optimization
AO-BO - Adaptive Osprey-Bowerbird Optimization
AO-BO-RNCN-MAN - Adaptive Osprey-Bowerbird Optimized Green Cloud Computing with Randomized Attention Coupled Fair Resource Distribution in Scalable Systems
CDDQL - Clipped Double Deep Q-Learning
DRABC-LB - Dynamic Resource-To-Ensure Balanced Workload
MAN - Multi-instance Attention Network
NOMA - non-orthogonal multiple access
O2A - Osprey Adaptive Algorithm
RNCN-MAN - Randomized Multi-instance Attention Coupled Neural Network
VM - virtual machine

Appendix

None.

References

[1] R. Mandal, M. K. Mondal, S. Banerjee, G. Srivastava, W. Alnumay, U. Ghosh, et al., "MECpVmS: An SLA aware energy-efficient virtual machine selection policy for green cloud computing," Cluster Comput., vol. 26, no. 1, pp. 651–665, 2023. DOI: 10.1007/s10586-022-03684-2

[2] S. K. Gupta, C. P. Ranjith, R. Natarajan, and M. S. K. Mohideen, 2024. An Energy Efficient Resource Allocation Framework for Cloud System Based on Reinforcement Learning. In Advancements in Science and Technology for Healthcare, Agriculture, and Environmental Sustainability (pp. 325-332). CRC Press. DOI: 10.1201/9781032708348-50

[3] S. Bharany, S. Badotra, S. Sharma, S. Rani, M. Alazab, R. H. Jhaveri, et al., "Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy," Sustain. Energy Technol. Assess., vol. 53, p. 102613, 2022. DOI: 10.1016/j.seta.2022.102613

[4] P. K. Bal, S. K. Mohapatra, T. K. Das, K. Srinivasan, and Y. C. Hu, "A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques," Sensors (Basel), vol. 22, no. 3, p. 1242, Feb. 6 2022. DOI: 10.3390/s22031242

[5] R. F. Mansour, H. Alhumyani, S. A. Khalek, R. A. Saeed, and D. Gupta, "Design of cultural emperor penguin optimizer for energy-efficient resource scheduling in green cloud computing environment," Cluster Comput., vol. 26, no. 1, pp. 575–586, 2023. DOI: 10.1007/s10586-022-03608-0

[6] S. Jayaprakash, M. D. Nagarajan, R. P. D. Prado, S. Subramanian, and P. B. Divakarachari, "A systematic review of energy management strategies for resource allocation in the cloud: Clustering, optimization and machine learning," Energies, vol. 14, no. 17, p. 5322, 2021. DOI: 10.3390/en14175322

[7] A. Mohammadzadeh, M. Masdari, F. S. Gharehchopogh, and A. Jafarian, "Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing," Evol. Intell., vol. 14, no. 4, pp. 1997–2025, 2021. DOI: 10.1007/s12065-020-00479-5

[8] S. Singhal, N. Gupta, P. Berwal, Q. N. Naveed, A. Lasisi, and A. W. Wodajo, "Energy Efficient Resource Allocation in Cloud Environment Using Metaheuristic Algorithm," IEEE Access, vol. 11,126135–126146, 2023. DOI: 10.1109/ACCESS.2023.3330434

[9] S.K. Podder, M. Karuppiah, B. Thomas, and D. Samanta, "Research initiative on sustainable education system: Model of balancing green computing and ict in quality education," In 2022 Interdisciplinary Research in Technology and Management (IRTM) (pp. 1-5). IEEE. 2022.

[10] Y. Gong, J. Huang, B. Liu, J. Xu, B. Wu, and Y. Zhang, 2024. Dynamic resource allocation for virtual machine migration optimization using machine learning. arXiv preprint arXiv:.

[11] S. Tuli, S. S. Gill, M. Xu, P. Garraghan, R. Bahsoon, S. Dustdar, et al., "HUNTER: AI based holistic resource management for sustainable cloud computing," J. Syst. Softw., vol. 184, p. 111124, 2022. DOI: 10.1016/j.jss.2021.111124

[12] A. Belgacem, "Dynamic resource allocation in cloud computing: Analysis and taxonomies," Computing, vol. 104, no. 3, pp. 681–710, 2022. DOI: 10.1007/s00607-021-01045-2

[13] S. Dahmani, 2024. Computational Intelligence for Green Cloud Computing and Digital Waste Management. In Computational Intelligence for Green Cloud Computing and Digital Waste Management (pp. 248-266). IGI Global.

[14] R. Buyya, S. Ilager, and P. Arroba, "Energy-efficiency and sustainability in new generation cloud computing: A vision and directions for integrated management of data centre resources and workloads," Softw. Pract. Exper., vol. 54, no. 1, pp. 24–38, 2024. DOI: 10.1002/spe.3248

[15] W. Hua, P. Liu, and L. Huang, "Energy-efficient resource allocation for heterogeneous edge-cloud computing," IEEE Internet Things J., 2023.

[16] Wang, Z., Goudarzi, M. and Buyya, R., 2025. TF-DDRL: A transformer-enhanced distributed DRL technique for scheduling IoT applications in edge and cloud computing environments. IEEE Transactions on Services Computing.

[17] Barua, B. and Kaiser, M.S., 2024. AI-Driven Resource Allocation Framework for Microservices in Hybrid Cloud Platforms. arXiv preprint arXiv:2412.02610.

[18] W. Shu, K. Cai, and N. N. Xiong, "Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing," Future Gener. Comput. Syst., vol. 124, pp. 12–20, 2021. DOI: 10.1016/j.future.2021.05.012

[19] M. Masdari and M. Zangakani, "Green cloud computing using proactive virtual machine placement: Challenges and issues," J. Grid Comput., vol. 18, no. 4, pp. 727–759, 2020. DOI: 10.1007/s10723-019-09489-9

[20] E. Öner and A. H. Özer, "An energy-aware combinatorial auction-based virtual machine scheduling model and heuristics for green cloud computing," Sustain. Comput., vol. 39, p. 100889, 2023. DOI: 10.1016/j.suscom.2023.100889

[21] A. Mohajer, M. S. Daliri, A. Mirzaei, A. Ziaeddini, M. Nabipour, and M. Bavaghar, "Heterogeneous computational resource allocation for NOMA: Toward green mobile edge-computing systems," IEEE Trans. Serv. Comput., vol. 16, no. 2, pp. 1225–1238, 2022. DOI: 10.1109/TSC.2022.3186099

[22] S. D. Alizadeh Javaheri, R. Ghaemi, and H. Monshizadeh Naeen, "An autonomous architecture based on reinforcement deep neural network for resource allocation in cloud computing," Computing, vol. 106, no. 2, pp. 371–403, 2024. DOI: 10.1007/s00607-023-01220-7

[23] T. Lyu, H. Xu, F. Liu, M. Li, L. Li, and Z. Han, "Two Layer Stackelberg Game-Based Resource Allocation in Cloud-Network Convergence Service Computing," IEEE Trans. Cogn. Commun. Netw., vol. 10, no. 6, pp. 2412–2426, 2024. DOI: 10.1109/TCCN.2024.3392809

[24] R. Mishra and M. Gupta, "DRABC-LB: A Novel Resource-Aware Load Balancing Algorithm Based on Dynamic Artificial Bee Colony for Dynamic Resource Allocation in Cloud," SN Comput. Sci., vol. 5, no. 2, p. 233, 2024. DOI: 10.1007/s42979-023-02570-x

[25] H. Liu, M. Xiang, M. Liu, P. Li, X. Zuo, X. Jiang, et al., "Random-coupled Neural Network," Electronics (Basel), vol. 13, no. 21, p. 4297, 2024. DOI: 10.3390/electronics13214297

[26] Z. Qin, H. Wang, C. B. Mawuli, W. Han, R. Zhang, Q. Yang, et al., "Multi-instance attention network for few-shot learning," Inf. Sci., vol. 611, pp. 464–475, 2022. DOI: 10.1016/j.ins.2022.07.013
[27] A. A. Ismaeel, E. H. Houssein, D. S. Khafaga, E. Abdullah Aldakheel, A. S. AbdElrazek, and M. Said, "Performance of osprey optimization algorithm for solving economic load dispatch problem," Mathematics, vol. 11, no. 19, p. 4107, 2023. DOI: 10.3390/math11194107
[28] S. Pouramini, "Multi-objective modified satin Bowerbird optimization algorithm used for simulation-based energy consumption optimization of yearly energy demand of lighting and cooling in a test case room," Energy, vol. 292, p. 130508, 2024. DOI: 10.1016/j.energy.2024.130508
[29] H. Zavieh, A. Javadpour, and A. K. Sangaiah, "Efficient task scheduling in cloud networks using ANN for green computing," Int. J. Commun. Syst., vol. 37, no. 5, p. e5689, 2024. DOI: 10.1002/dac.5689

Authors' Profiles



Aishwarya Shekhar working as an Assistant Professor in the Department of Computer Science and Engineering at Sandip University, Sijoul, Madhubani. He is currently pursuing Ph.D. in Computer Science and Engineering from Galgotias University, with an expected completion in 2025. His academic journey has been marked by a consistent pursuit of excellence, as evidenced by securing an A+ grade in his Ph.D. coursework exams. Mr. Shekhar also completed a Post Graduate Diploma in Management (PGDM) in IT from SVKM's Narsee Monjee Institute of Management Studies in 2022. He holds a Master of Technology degree in Computer Science and Engineering from Manipal University Jaipur, obtained in 2015, and a Bachelor of Technology in Computer Science and Engineering from UPTU, where he graduated with first division honors in 2013. Professionally, Mr. Shekhar began his career at AWC Software Pvt. Ltd, where he served as a Software Developer. He then transitioned to Velocis Systems Pvt. Ltd as a Senior Software Developer. Currently, he is the Academic Coordinator for the B.Tech and M.Tech courses at Sandip University, Madhubani. He has published more than 20 research papers in national and international journals, including five Scopus-indexed papers and three IEEE Conference Papers, which are also indexed in Scopus. Additionally, he has Co-authored a book. Mr. Shekhar has six years of experience in the IT industry and three years in academia. He is a member of IEEE, ACM, and IAENG. His research interests include IoT, Cloud Computing, AI, and Machine Learning.



Abdul Aleem is a Professor in the School of Computing Science and Engineering (SCSE), at Galgotias University. He has done his B. Tech.(CSE) from U.P. Technical University, Lucknow. He has done his M. Tech. (Software Engg.) from MNNIT Allahabad. He was bestowed Gold Medal for being the topper of his batch in the master's program. He has obtained Ph.D. (CSE) from MNNIT Allahabad in the area of Data Mining, Machine Learning, and Educational Systems. He has been thrice awarded as Best/Exemplary Faculty, five times conferred the Best Paper award, and two times awarded for coding Excellence. He possesses 13+ years of blended experience including 6+ years in top software industries and 7+ years in premium educational institutions as a value-added academician, software developer, lead engineer, researcher, administrator, and intellectual professional. His research interests include digital image processing, optimization techniques, machine learning and artificial intelligence. He has published more than 30 research papers in reputed conferences and journals of distinguished indexing.

How to cite this paper:

Aishwarya Shekhar, Abdul Aleem, "Adaptive Osprey-bowerbird Optimized Green Cloud Computing with Randomized Attention Coupled Fair Resource Distribution in Scalable Systems", International Journal of Computer Network and Information Security(IJCNIS), Vol.18, No.3, pp. 105-121, 2026. DOI:10.5815/ijcnis.2026.03.06