International Journal of Basic and Applied Sciences, 14 (6) (2025) 12-20

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/bp2nsc52
Research paper

A Review of Meta Heuristic Algorithms and Its Evaluation for Load Balancing in Cloud Computing

S. Balaji [1, 4] *, S. Silvia Priscila [2], Praveen B. M. [3]

[1] *Institute of Computer Science and Information Science, Srinivas University. Mangaluru-574146, Karnataka, India*
[2] *Department of Computer Science, Bharath Institute of Higher Education and Research, Tamil Nadu 600126, India*
[3] *Institute of Engineering and Technology, Srinivas University, Mangaluru-574146, Karnataka, India*
[4] *Department of Information Technology, Al Zahra College for Women, Ghala, Muscat*
**Corresponding author E-mail: balaji@zcw.edu.om*

Received: July 18, 2025, Accepted: September 7, 2025, Published: October 2, 2025

Abstract

Cloud computing(CC), which utilizes massively virtualized data centers to deliver quick and affordable computing solutions, has developed into an established industrial standard that is growing quickly. To handle such a massive amount of data effectively, cloud computing mostly relies on automation and dynamic resource management. In cloud computing, load balancing (LB) is a vital technique for maximizing resource utilization and making sure that no resource is used up. Without the requirement for physical infrastructure, cloud LB allows online platforms to adjust their resources in response to traffic demands. In a cloud environment, workload and resource allocation entail determining the best way to divide up work among several servers. For increasingly severe uncertainty problems, traditional LB approaches are simple but ineffective; for this reason, meta-heuristic methods are employed. This algorithm is heuristic and is independent of the complexity of the challenges. Meta-heuristics approaches based on Artificial Intelligence (AI) are employed to analyze real-time data and intelligently distribute workload among servers. This ensures efficient operations by preventing bottlenecks and enabling proactive LB decisions. The review offers a thorough analysis of meta-heuristics techniques based on artificial intelligence (AI) for static and dynamic LB in both homogeneous and heterogeneous cloud systems.

Keywords: *Artificial Intelligence; Cloud Computing; Load Balancing; Meta-heuristic Algorithms.*

1. Introduction

With characteristics like high tolerance, scalability, availability, robustness, and so on, CC is a distributed computing paradigm that stores enormous amounts of data in order to offer software, platforms, or infrastructure services to customers on demand. Load balancing is a method of optimizing a resource on cloud Virtual Machines(VMs). It must efficiently distribute the load across the cloud nodes in order to prevent situations where a node is overburdened or underloaded because it processes enormous amounts of data. The cloud applications perform better overall when LB is implemented, as seen by reduced latency, avoided bottlenecks, and higher job completion rates. Still, there are a number of problems with LB in the cloud, such as different user requirements for Quality of Service (QoS), unexpected resource failures, traffic variation across different regions, communication overhead, frequent traffic migration, resource waste, inconsistent service abstractions, and so forth [1].

The Cloud Service Provider (CSP) offers services to clients for rent. Because of the readily available virtual cloud resources, the CSP's role in offering services to the consumer is very intricate. Numerous VMs are run by physical machines in a cloud environment and are made available to clients as computing resources. An analogous actual computer serves as the foundation for the architecture of a VM. Actually, a VM is a guest application that runs like a real computer and has software resources. A crucial system for optimizing server usage by dynamically distributing resources according to user application demands. The burden of non-preemptive jobs is balanced by the dynamic execution of this process. "LB is an intricate Non-Polynomial-hard optimization problem in CC". Because of this, LB has received increased attention from researchers and has improved system performance [2]. Figure 1 presents the benefits of LB.

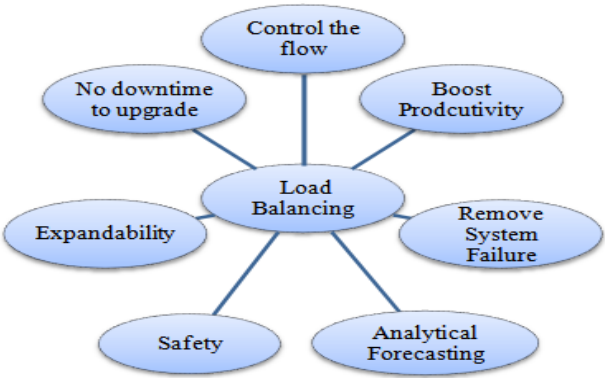

Fig. 1: Advantages of LB.

Figure 2 shows the framework for LB in CC. The load balancer allocates the jobs by employing LB techniques to the VMs in turn, and then allocates them to the physical machines via the VM monitor.

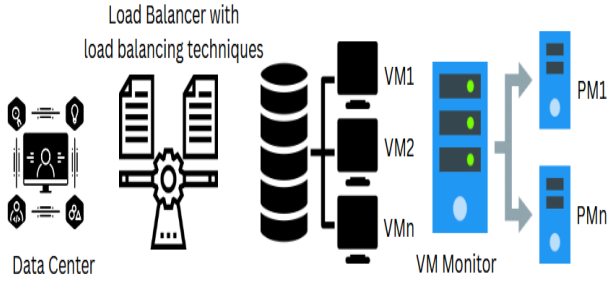

Fig. 2: Framework for LB in Cloud Environment.

The use of AI in resource allocation can significantly cut costs and wastage. It seems like the main area of research is cloud resource management and allocation. In the rapidly developing field of CC, one of the main concerns is the efficient scheduling of resources or the best distribution of requests. More efficient resource management techniques will be needed for future Cloud systems due to the increasing complexity of CC. In certain intricate situations, when assessing the effectiveness of scheduling solutions directly proves challenging, traditional algorithms will not yield a successful plan [3].

One practical way to increase energy efficiency is to suggest dynamic LB. It entails allocating network resources in a way that fairly distributes the weight of requests among devices. Gains in energy efficiency may result from this strategy since resources are only active when a load is assigned to them. Intelligent algorithms that give resource distribution and load optimization top priority are needed to implement dynamic LB in networks. Such intelligent algorithms can be developed using meta-heuristics algorithms, and it is combined with AI to enhance optimization. The growth of meta-heuristic algorithms has taken many years. Originally, these algorithms were created to address optimization issues in surroundings and systems. With time, dynamic LB has also used more meta-heuristic techniques, particularly in distributed and parallel systems [4].

2. Related Works

Resource management systems can support resource provisioning according to the demands of scientific applications. The most critical problem that needs to be addressed successfully due to the heterogeneity of the resources, interdependencies, and load uncertainties in the cloud environment is the scheduling of cloud resources. The study presented in [5] proposes a hybrid model of dynamic workload provisioning in CC that combines DL with the Particle Swarm Intelligence and Genetic Algorithms (also called DPSO-GA). The paradigm suggested works in two stages. The initial stage involves a hybrid PSO-GA process to solve the prediction challenge by taking advantage of these two methods to optimize the Hyperparameters. In the second stage, the prediction of cloud VM machine workload is performed by LSTM and one-dimensional CNN, in which an LSTM module creates time-related information to forecast the workload of the next VM. The suggested paradigm simultaneously integrates resource use as a multi-resource utilization and provides a solution to the problem of LB and over-provisioning. Large-scale simulations are conducted based on the Google cluster traces benchmarks dataset to confirm the effectiveness of the suggested DPSO-GA method to optimize resource allocation and LB in the cloud.

A revolutionary optimal LB solution in the cloud is provided in [6] using the new Intercrossed Chimp and Bald Eagle Algorithm (IC&BA). The LB model considers the following: (i) energy consumption; (ii) execution and migration cost; (iv) make span; and (v) LB parameters such as response and turnaround time. To allocate the cloud workload, the multiple objective dynamic LB method was proposed in [7] as clustering-based CMODLB. It is a form of supervised (artificial neural network), unsupervised (clustering), and soft computing (interval type 2 fuzzy logic system)-based LB algorithm.

In CC, genetic algorithm, BAT algorithm, ant colony, grey wolf, artificial bee colony, particle swarm, whale, social spider, and dragonfly are optimized in LB [8]. A work in [9] introduces a new technique to LB that dynamically allocates workloads across cloud resources. The approach relies on the self-organizing and adaptive behaviors of natural ecosystems. The methodology employs the ideas of GAs and swarm smarts to enhance load allocation strategies as much as possible.

The [10] offers an optimisation of the energy-aware job-scheduling design in heterogeneous distributed systems, utilizing meta-heuristic methods. Due to its capability to adapt to a broad search space and its utilization of a greedy strategy to prevent premature convergence and local maximum, the Harris Hawk Optimization (HHO) method has been considered in the optimization problem. Experiment in [11] provides an LB technique relying on Rock Hyrax that solves the problems of power consumption and local optima solutions under the parameters of QoS. The algorithm is tested qualitatively and quantitatively, considering both the static and dynamic work modes and VMs. One of the new methods that has been proposed in [12] to improve LB and optimal resource allocation in the cloud is the Max-Min Heuristic

(MMH) and the Enhanced Ant Colony Optimization (ACO). It allocates the overall load equally in the cloud following the MMH approach to LB. LB is achieved by relocating tasks of the overloaded nodes to the underloaded nodes.

In [13], a new dynamic load-balancing method is presented, which calculates the load of each VM based on a DL model that combines convolutional and recurrent neural networks (RNNs). The process is supposed to enhance cloud performance through better scheduling of jobs and distribution of workloads. The suggested paradigm applies a dynamic clustering process on the basis of calculated loads to distinguish VMs into overloaded and underloaded groups. The technique is an enhanced combination of Reinforcement Learning (RL) and an advanced Hybrid Lyrebird Falcon Optimization (HLFO) algorithm to enhance the effectiveness of clustering. HLFO is a combination of the Lyrebird Optimization Algorithm (LOA) and the Falcon Optimization Algorithm (FOA). To solve the optimization issues, a hybrid approach that integrates a meta-heuristic Black Widow Optimization (BWO) algorithm and an energy-aware self-governing job scheduler, i.e., an Artificial Neural Network (ANN), is suggested in [14].

Workloads should be distributed among multiple servers to use the resources efficiently and improve system performance with LB strategies. The LB is an efficient method of allocating resources to the cloud based on Swarm intelligence. Swarm intelligence algorithms can enhance the performance of a system by maximizing resource utilization, minimizing energy usage, and applying decentralized decision-making principles [15]. To effectively balance the load in VMs, a hybrid approach named CSSA, which is a hybrid of the Simulation Annealing (SA) algorithm and Cuckoo Search Optimization (CSO) algorithm, is proposed in [16]. In this method, the multi-objectives of resource utilisation (RU), degree of imbalance (DoI), cost, and reaction time are taken into account to update the SA search space using the CSO technique. In [17], a new method of VM migration, which uses the Capuchin search algorithm (CapSA), is proposed. The idea is to leverage the benefits of migration and scheduling through a hybrid CapSA and inverted ACO algorithm that mixes both. Depending on the nature of the work it receives, it uses a decision-making model to choose the most appropriate algorithm to use in the next work. Relative to EC, execution time (ET), and LB, the proposed strategy is 15 to 20 percent more successful than the other methods.

In [18], a study proposed the Enhanced Marine Predator approach (EMPA), which is a scheduling efficiency enhancement method. First, a makespan-resource-conscious task scheduling model is built. The second is the Marine Predator Algorithm that consists of the golden sine strategy, nonlinear inertia weight coefficient, and WOA operator. Work scheduling produces a product: each person is an output of an algorithm designed to determine the best results of scheduling. The simulation experiment pits EMPA against the following algorithms based on the number of tasks in GoCJ datasets and generated datasets: Whale Optimization Algorithm (WOA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and Sine Cosine Algorithm (SCA). An optimization technique proposed in [19] employs ant colony optimization to assign a number of scheduling goals and determine a fitness function. The algorithm is useful in deciding how to partition the applications between edge and cloud servers to maximize their computing power.

In [20], a two-step process based on intelligent rule-based meta-heuristic task scheduling (IRMTS), which integrates machine learning methods and meta-heuristic job scheduling solutions, is presented. Many cloud task-scheduling problems have demonstrated the effectiveness of the hybrid approach, resulting in a significant reduction in execution time and the ability to apply meta-heuristics to time-critical software. One of the studies in [21] focuses on how VMs can optimally remap resources based on dynamically changing loads to maximize the overall network performance. The main objective of the current study is to critically examine some meta-heuristic algorithms used in CC LB. These are: GA, PSO, ACO, GWO, and Artificial Bee Colony (ABC). The analysis is conducted with the help of CloudAnalyst, which helps to analyze different strategies, key metrics of performance, including Data Centre Processing Time (DCPT) and Total Response Time (TRT). The researchers proposed a combination method that would be a combination of the GWO and GA. The multi-objective task scheduling aims of the hybrid GWO-GA method in CC are to reduce costs, energy consumption, and makepan [22]. In [23], a study proposes the Multi-Objective Whale Optimization-Based Scheduler (WOA-Scheduler), which is an efficient scheduling of tasks in CC.

The scheduler uses the WOA to optimize LB, time, and cost simultaneously. The goal of an efficient task scheduling technique is to achieve maximum output, reduce response time, utilize fewer resources, and save on energy by allocating the right resources to the right tasks. The best VM is scheduled using a hybrid ant genetic algorithm (combining the benefits of genetic algorithms with pheromone values based on ant colony algorithms) [24]. In [25], the Deep Max-out prediction model was presented and used to predict future workload to enable workload balancing. This allowed an assisted Bald Eagle Search (BES) hybrid Tasmanian Devil-aided strategy to be optimised using scheduling. Back-Propagation Algorithm (BPA) using the ANN idea (ANN-BPA) is proposed in [26] to test the entire process of scheduling and problem-detection.

The swarm-based optimization method concept is applied to identify the best features in training the ANN-BP A by using the data in a random simulation. The usefulness of the model is subsequently confirmed by a comparison study involving numerous swarm methods, including Moth Flame Optimization (MFO), ABC, CSA, and PSO. In this case, the MFO + ANN-BPA combination is better than the alternative regarding job distribution, completion, time taken, and energy utilization. An LB and energy-efficient migration model is published in [27] that exploits the latest security capabilities and bioinspired algorithms to optimize VM migrations. The model is the first to leverage the use of GAs to assist in resource scheduling in the context of ACO. Such algorithms were chosen because they are effective in replicating the workings of nature to solve difficult optimization problems.

In [28], a review is conducted on various methods of scheduling workflow based on various factors such as cost, dependability, performance, scalability, and security. The findings indicate that 75 percent of the scheduling algorithms employ different parametric modeling algorithms and AI-based mechanisms, whilst 25 percent of the algorithms employ heuristic-based mechanisms. A layer fit algorithm proposed in a study in [29] will divide work between the fog and cloud equally based on priority levels. Moreover, a meta-heuristic method on the basis of Modified HHO is suggested to allocate the best resource to a job within a layer. The aim is to optimize resource utilization in the fog and cloud layers and reduce makespan time, task execution cost, and power consumption. The simulations are carried out with the iFogSim simulation tools. The proposed layer fit algorithm and the Modified HHO are compared with the traditional HHO, ACO, PSO, and FA. In [30], a hybrid Task Scheduling (TS) approach is proposed, which uses long short-term memory (LSTM) to determine the dependability of task runtime, tuna swarm optimization (TSO) to plan tasks with the greatest expected runtime, and the VIKOR strategy to backfill the leftover work. Table 1 illustrates the various AI-based meta-heuristics methods of LB in the cloud environment that are reported in the literature. In [35], a hybridization of the improved Q-learning algorithm (QMPSO) and the adapted PSO is applied to provide a method of dynamic LB between VMs. Q-learning is a form of reinforcement learning in artificial intelligence, which enables the learner to adapt to its surroundings and transform its state to receive a reward or a punishment based on environmental input [35]. Table 1 compares the main strengths and weaknesses of a few meta-heuristic and machine learning-based LB methods.

Table 1: Meta Heuristics LB methods –Strengths and Weaknesses

S. No	Author(s) & Year	Methods Used for LB	Strengths	Weaknesses
1	Simaiya et al. [5]	DL, PSO, GA	PSO & GA converge moderately fast; good optimization accuracy	DL convergence is slow; energy efficiency is low
2	Geetha et al. [6]	IC, BA	BA has fast convergence; balanced performance	IC convergence is only moderate
3	Negi et al. [7]	CMODLB	Balanced performance across metrics	No significant strength; moderate scalability
4	Elmagzoub et al. [8]	GA, BAT, ACO, GWO, ABC, PSO, Whale, Social Spider, Dragonfly, Raven Roosting	Swarm-based methods improve convergence speed	Scalability and energy efficiency are low to moderate
5	Khan et al. [9]	Swarm Intelligence, GA	Good convergence and moderate scalability	No major energy efficiency improvement
6	Li et al. [10]	HHO, Greedy Algorithm	Greedy achieves fast convergence	HHO slower than Greedy; moderate scalability
7	Singhal et al. [11]	Rock Hyrax	Moderate energy efficiency; simple implementation	Low-moderate scalability; moderate convergence
8	Banupriya et al. [12]	MMH, EACO	Hybrid improves convergence speed	Scalability only moderate
9	Khan et al. [13]	CNN, RNN, HLFO, FOA, LOA	High scalability; FOA & LOA converge moderately	CNN/RNN slow; low energy efficiency
10	Selvakuamr et al. [14]	ANN, BWO	Balanced performance across metrics	Moderate convergence; no major energy improvement
11	Raghav et al. [15]	Swarm Intelligence	Fast convergence (swarm); moderate scalability	Energy efficiency is not optimal
12	Nebagiri et al. [16]	CSSA, CSO	High energy efficiency; moderate-fast convergence	Scalability is moderate
13	Rostami et al. [17]	CapSA, IACO	Balanced performance; moderate convergence	No significant advantage in scalability or energy
14	Gong et al. [18]	EMPA	Moderate convergence; simple implementation	Moderate energy efficiency; scalability is limited
15	Khaleel et al. [19]	ACO	Fast convergence; efficient in small-scale systems	Low-moderate scalability
16	Barut et al. [20]	ML, Metaheuristic Job Scheduling	High scalability; adaptive	Moderate convergence; energy efficiency moderate
17	Syed et al. [21]	GWO, ACO, PSO, GA, ABC	Moderate-fast convergence; hybrid improves performance	Scalability only moderate
18	Behera et al. [22]	GA, GWO	Balanced hybrid; moderate convergence	Moderate energy efficiency and scalability
19	Gupta et al. [23]	WOA	Fast convergence; high energy efficiency	Scalability is moderate; it has been less tested in large-scale systems
20	Thilak et al. [24]	GA, ACO	Balanced performance across metrics	No significant strength; moderate energy
21	Karimunnisa et al. [25]	Deep Max-out Prediction Model, TES	High scalability	Slow convergence; low energy efficiency
22	Kaur et al. [26]	ANN-BPA, PSO, ABC, CSA, MFO	Moderate convergence; hybrid approach	Low energy efficiency; moderate scalability
23	Brahmam et al. [27]	GA, ACO	Balanced hybrid; moderate convergence	Moderate scalability and energy efficiency
24	Khaledian et al. [28]	Workflow Scheduling Methods	Balanced workflow handling	No significant optimization advantages
25	Yakubu et al. [29]	MHHO, HHO, ACO, PSO, FA	Moderate-fast convergence; high energy efficiency	Scalability only moderate
26	Boopathi et al. [30]	TSO, VIKOR, LSTM	TSO/VIKOR moderate convergence	LSTM is slow; low energy efficiency
27	Jena et al. [35]	QMPSO	Fast convergence; high energy efficiency	Scalability moderate-high; hybrid complexity

The existing heterogeneity of resources, interdependencies, and the unpredictability of workloads continue to make cloud resource management and scheduling critical to the effective, reliable, and cost-effective functioning of contemporary cloud systems. Recent studies show that hybrid algorithms integrating deep learning and meta-heuristic optimization methods, including DPSO-GA, can greatly enhance the workload prediction and dynamic resource allocation based on temporal patterns (LSTM) and feature extraction (CNN) to use VMs. On the same note, swarm intelligence-oriented algorithms like PSO and ACO. GWO and WOA provide decentralized load-balancing features with reduced energy use and response times. Multi-objective scheduling can be further optimized by evolutionary and bio-inspired algorithms like GA, HHO, BWO, and HLFO, which reduce the cost of execution, migration cost, and makepan, especially in a heterogeneous distributed setting. Several works also combine reinforcement learning and hybrid meta-heuristic approaches, e.g., Q-learning and PSO or deep max-out models and Tasmanian Devil-assisted BES as a means of dynamically adapting to workload variability and, at the same time, keeping the system running. Although most studies prove the efficiency of the particular algorithms to enhance resource usage, LB, and energy efficiency, such close comparisons have shown that swarm intelligence algorithms prove effective in decentralized and adaptive environments, and evolutionary algorithms are more efficient in global optimization on a complex and multi-objective landscape. Models that fuse deep learning and meta-heuristics seem to be especially promising and can provide predictive power along with optimization, but their computational costs should also be taken into account. In general, the literature highlights the fact that the key to scalable, energy-efficient, and high-performance cloud resource management lies in leveraging AI-based hybrid approaches, especially those involving prediction, clustering, and optimization.

3. LB methodologies

The main aim of the LB approach is to maximize the duration of the operations using the resources available, since the work capacity varies with the run time. In a CC environment, the load is the allocation of various workloads to the VMs. The LB problem can be defined in several ways, as shown below. (1) Assignment of tasks: a handful of workloads are randomly assigned to several Physical Machines (PMs), which are then assigned to a single VM on the PM. The success of the LB algorithm is determined by the ability to delegate the duties to the cloud. (2) VM/Task Migration Management: VM Migration in the CC environment involves moving a VM between one PM to another to maximize the resources in the data centre, which imposes too much strain on the PM. Similarly, task migration: This refers to the transfer of the current state of a task between VMs or the VMs on different hosts. CC LB, therefore, requires task migration or VMs. [31]. Workload balancing (LB) refers to the distribution of workload within a distributed system, such as that found in CC, to ensure that no machine is underused, overused, or not used at all. LB enhances cloud performance by attempting to accelerate many of the bottlenecks, such as execution time, reaction time, system stability, etc. This optimization method is applied when the job scheduling problem is NP hard. Despite the numerous LB techniques suggested by academicians, most of their attention has been on resource scheduling, resource allocation, job scheduling, and resource management [32].

3.1. Static load vs dynamic LB algorithms

Static LB algorithms divide the incoming demand on a server using algorithms that have previous knowledge about the servers that are currently in the distributed network. This is accomplished by giving prior information about the system. A pre-established load schedule used in static LB techniques establishes the maximum amount of load that can be placed on other systems. Static LB is intended for systems where the incoming load fluctuates little. The servers' traffic is split equally in this LB scheme. While real-time contact with the servers is not necessary for static LB, it does require more detailed knowledge about the available system resources. The assigned load cannot be re-transferred to other servers during runtime in static LB [33]. The static load balancers include round robin, weighted round robin, and IP hash. Static LB techniques will focus appropriately when the VMs' load fluctuates very little. Because of the unpredictable variations in loads during run time, the static LB method will not function as intended [35].

Dynamic LB modes are more flexible LB schemes that dynamically determine which system should bear the load and how much weight has to be shed during runtime. They do this by using node performance information. Dynamic LB is intended for systems where the incoming load fluctuates significantly. Traffic is dynamically distributed across the servers in this LB scheme. Dynamic LB may not always require more in-depth knowledge about system resources in advance, but it does involve active real-time communication with the servers. To lessen resource underutilization, the assigned load in dynamic LB may be retransmitted among servers. The dynamic load balancers include least connection, weighted least connection, weighted response time, and resource-based [34]. In situations where loads will fluctuate during runtime and require consideration of load information and maintenance, dynamic LB is preferable to static LB. Dynamic approaches are very important and effective in distributing the load across the diverse resources because of the network's quick expansion and the need for resources during run time [35].

3.2. Artificial intelligence-based meta-heuristic algorithms

Artificial intelligence (AI) is using more nature-inspired meta-heuristic (MH) algorithms and is being used in many different applications to solve complicated optimization problems. Not only has there been a push for the use of MHs to effectively combine MH algorithms with machine learning (ML) techniques, but for some time, it has become more prevalent to use the combination of ML methods with MH to be able to solve hard combinatorial optimization problems in new ways. Research studies show that meta-heuristics using machine learning methods can use this novel algorithm and can search more efficiently, effectively, and robustly with the goal of now producing better solutions, improving convergence, and increasing resilience.

Machine learning (ML) is a branch of AI that seeks to train algorithms to infer patterns from data and learn new tasks. There has been a growing interest in the combination of ML approaches with MHs. As solutions are generated during a search, it is possible to have ML identify knowledge from those solutions that will inform the MH algorithm to make better decisions. The fusion of these two forms of algorithms predicts greater adaptability, speed to convergence, and solution quality [36]. In CC, if tasks are not scheduled properly, one may experience under- or over-utilization of resources, leading to wasted cloud resources or performance degradation of services. To adequately share complex and diverse incoming tasks (cloudlets) to constrained resources in an acceptable timeframe, researchers have increasingly utilized meta-heuristic algorithms for scheduling tasks. MH-based approaches are highly effective for allocating resources within CC to find the optimal or near-optimal solution [37]. Current load-balancing techniques experience some difficulty due to dynamic workloads and heterogeneous resources with varying service-level requirements. Meta-heuristic-oriented solutions have addressed these issues and have provided solutions that are timely, accurate, reliable, efficient, and contribute to high-quality service performance. For decades, researchers have used meta-heuristic strategies and approaches to solve LB problems because of their ability to manage large-scale optimization problems, when reliability and optimized performance are required [38].

3.3. How machine learning enhances meta-heuristics

Machine learning can improve meta-heuristic algorithms in several ways. Through reinforcement learning, ML can leverage a meta-heuristics search process, allowing it to adjust meta-heuristics parameters dynamically while learning from feedback provided during the search process. This should speed up convergence and/or guide the search toward better solutions. Additionally, through predictive modeling, ML can predict task characteristics or total resources, which will enable meta-heuristics to explore the best solution based on the search space, or potentially abandon areas with no expected promising solutions. In the same light, data-driven heuristics will allow ML to analyze the past with scheduling (for example) or load-balancing data to help the meta-heuristic algorithm go to areas of the search space where it is likely to find optimal solutions. Adaptive hybrid strategies using ML allowed the meta-heuristics to intelligently flip from drawn exploration to drawn exploitation, giving them a status of robustness and efficiency. Finally, through the artificial intelligence (AI) examined in this manuscript, the combination of the decision-making process of ML and the optimization power of meta-heuristic algorithms, the modern AI-based load-balancing algorithms achieve amazing performance in the CC environment by balancing speed, scalability, and resource utilization[

3.4. Hardware/software Co-design for LB

The importance of hardware/software co-design continues to grow for LB in modern cloud systems with heterogeneous resources and dynamic workloads. Cloud systems can schedule workloads better by using software scheduling algorithms combined with the hardware capabilities of a CPU, GPU, FPGA, and smart NICs. In hardware-aware scheduling, workloads can be assigned to nodes that are the best hardware match. Real-time telemetry and hardware counters for feedback can allow workloads to shift for adaptive LB. Co-design allows for software policies and utilized hardware resources to be optimized together rather than separately, enhancing latency, throughput, and scalability. Hardware/software co-design demands more process complexity and reduces portability, but is essential for the increasingly heterogeneous, large-scale clouds. It is not possible to build cloud systems without hardware/software co-design, and it will mostly be encapsulated via AI-driven, predictive solutions on LB and edge-cloud systems with seamless interconnections to effectively deal with highly variable workloads.

4. Performance Metrics for LB Algorithms

A number of measures have been proposed to assess and guide load-balancing algorithms. In order to provide an accurate assessment of the algorithms' supremacy and to support the assertions made by researchers for their particular goals, it is crucial to employ an adequate instrument for evaluating the performance of a wide range of meta-heuristic algorithms [41]. A few of the performance indicators used to assess the efficiency of LB techniques are as follows.

- Root Mean Squared Error (RMSE): It is used to evaluate how well various algorithms anticipate workloads [40].
- Mean Absolute Error (MAE): It is used for workload prediction to evaluate the effectiveness of various methods [40].
- Wilcoxon signed rank: Used to perform statistical analysis[40]
- Friedman with Finner post-hoc multiple comparison tests: Used to perform statistical analysis
- Response time is defined as the overall time taken by the system to accomplish a job. In client-server communication, the latency is the time between when the client makes a request and when the server makes the first response, which can be measured in milliseconds. This measure captures all delays, such as processing, communication, and queuing, and is a primary measure of the speed at which tasks are executed, which is a direct measure of the user experience.

Response time: Time in which the server sends the response -Time in which the client sends the request

- Migration time: Time to move an assignment from a server that is overloaded to one that is underloaded. indicates how long it takes to transfer jobs from hosts that are overloaded to ones that are not. It affects resource usage and system responsiveness. [38], [42].
- Makespan time: The time it takes to distribute resources to a consumer. The highest finish time or the amount of time needed to allocate resources to a customer [38].
- Throughput: The rate at which jobs or processes are finished within the system over time is measured by throughput. It functions as a gauge of the system's effectiveness and processing power.
- Fault tolerance: The application's ability to perform LB when links break. It evaluates the ability of an algorithm to preserve LB in the event of node or connection failures. It guarantees system performance and stability under difficult situations [43].
- Scalability: The application's ability to perform LB as the network grows. It assesses whether an algorithm can evenly distribute the system's burden as the number of nodes rises. Regardless of the number of nodes, a highly extensible algorithm can efficiently manage load distribution.
- Migration time: The time it takes to move an assignment from an overloaded server to an underloaded server
- Degree of imbalance: How evenly VMs are distributed. The difference in workload distribution between VMs or nodes is measured by the degree of imbalance. A well-balanced workload distribution helps ensure optimal system performance [44].
- Average Resource Utilization ratio: The formula for calculating the average utilization of resources ratio is

ARUR= (mean time/makespan)*100.

Where mean time = ∑ Time required by resource (VMj) to complete all the jobs/number of resources. The maximum value of ARUR is 1 (resource utilization is 100%), and the lowest value is 0 (resource is in optimum condition). The range of the average resource utilization ratio is 0 to 1 [39]. These metrics provide important information about a computing system's productivity, effectiveness, and LB operation. Table 2 gives a summary of the major meta-heuristic and machine learning-based LB algorithms in the cloud and distributed computing networks. It considers the performance of each approach according to important key performance measures and its usefulness in different computational problems.

Table 2: Comparison of Meta-Heuristic and Machine Learning-Based LB Methods Based on Key Performance Measures

S. No	Author(s) & Year	Methods Used for LB	Convergence Speed	Scalability	Energy Efficiency
1	Simaiya et al. [5]	DL, PSO, GA	Average	Balanced	Low
2	Geetha et al. [6]	IC, BA	Average	Balanced	Medium
3	Negi et al. [7]	CMODLB	Average	Balanced	Medium
4	Elmagzoub et al. [8]	GA, BAT, ACO, GWO, ABC, PSO, Whale, Social Spider, Dragonfly, Raven Roosting	Average–Fast	Limited–Balanced	Low–Balanced
5	Khan et al. [9]	Swarm Intelligence, GA	Average	Balanced	Medium
6	Li et al. [10]	HHO, Greedy Algorithm	Fast	Balanced	Medium
7	Singhal et al. [11]	Rock Hyrax	Average	Limited–Balanced	Medium
8	Banupriya et al. [12]	MMH, EACO	Average–Fast	Balanced	Medium
9	Khan et al. [13]	CNN, RNN, HLFO, FOA, LOA	Slow	High	Low
10	Selvakuamr et al. [14]	ANN, BWO	Average	Balanced	Medium

11	Raghav et al. [15]	Swarm Intelligence	Average–Fast	Balanced	Medium
12	Nebagiri et al. [16]	CSSA, CSO	Average–Fast	Balanced	High
13	Rostami et al. [17]	CapSA, IACO	Average	Balanced	Medium
14	Gong et al. [18]	EMPA	Average	Balanced	Medium
15	Khaleel et al. [19]	ACO	Fast	Limited–Balanced	Medium
16	Barut et al. [20]	ML, Metaheuristic Job Scheduling	Average	High	Medium
17	Syed et al. [21]	GWO, ACO, PSO, GA, ABC	Average–Fast	Balanced	Medium
18	Behera et al. [22]	GA, GWO	Average	Balanced	Medium
19	Gupta et al. [23]	WOA	Fast	Balanced	High
20	Thilak et al. [24]	GA, ACO	Average	Balanced	Medium
21	Karimunnisa et al. [25]	Deep Max-out Prediction Model, TES	Slow	High	Low
22	Kaur et al. [26]	ANN-BPA, PSO, ABC, CSA, MFO	Average	Balanced	Low
23	Brahmam et al. [27]	GA, ACO	Average	Balanced	Medium
24	Khaledian et al. [28]	Workflow Scheduling Methods	Average	Balanced	Medium
25	Yakubu et al. [29]	MHHO, HHO, ACO, PSO, FA	Average–Fast	Balanced	High
26	Boopathi et al. [30]	TSO, VIKOR, LSTM	Slow	Balanced	Low
27	Jena et al. [35]	QMPSO	Fast	Moderate–High	High

Table 2 compares different LB methods based on the speed of convergence, scalability, and energy efficiency. The convergence of most methods is mediocre and evenly scalable, meaning not extraordinary performance. Swarm intelligence- and metaheuristic-based algorithms (e.g., GA, ACO, PSO) are of intermediate quality, whereas hybrid algorithms such as HHO with greedy algorithms and QMPSO are faster converging and more scalable. Computational cost is of interest due to the slow convergence of some deep learning-based models (CNN, RNN), and energy efficiency. Interestingly, CSSA, CSO, WOA, and MHHO all have high energy efficiency, and so they are potentially applicable in energy-aware load balancing in large-scale systems.

5. Discussions

CC has become a leading paradigm in the provision of scalable, cost-effective, and on-demand computing services. The increase in virtualized data centers and the rapidity of their expansion have necessitated effective resource management methods, LB being one of the most essential methods to achieve optimal resource usage and avoid system bottlenecks. Simple and straightforward approaches to LB are generally not sufficient to support dynamic, complex, and uncertain loads [46]. This has encouraged the use of AI-based meta-heuristic methodologies, which smartly process real-time data to maximize the workload distribution among servers in homogeneous and non-homogeneous cloud infrastructures. As noted in the review, even though meta-heuristic algorithms, including PSO, GA, WOA, and hybrid deep learning/meta-heuristic models, have enhanced the performance of cloud LB significantly, various open challenges exist:

- Minimizing uncertainty in the multi-objective optimization: It remains not fully considered how to efficiently trade-off execution time, energy consumption, cost, and SLA compliance across dynamic workloads with uncertainty.
- Scalability to large-scale and distributed environments: Most current systems have been tested with small-scale cloud configurations; more complex heterogeneous cloud-edge ecosystems are a reality in the real world.
- Explainability and interpretability: AI-based meta-heuristic algorithms tend to be black-box and are not easily justifiable in terms of scheduling decisions and trustworthiness.
- Interference with new paradigms Edge-cloud hybrid architectures, IoT-enabled systems, and fog computing environments need to be integrated with adaptive LB strategies that Standard algorithms cannot deal with.
- Energy efficiency and sustainability: The ability to optimize energy consumption without sacrificing performance in large-scale clouds is a major issue.

Future Research Directions include the following

- Creation of explainable and interpretable AI-based meta-heuristic scheduling algorithms in a trustworthy and transparent manner.
- Adaptive LB design of edge-cloud hybrid systems based on latency, resource limits, and QoS.
- Observations of quantum computing for large-scale optimization to provide more efficient and faster multi-objective scheduling.
- Hybrid workload prediction models that use deep learning.

6. Conclusion

In this study, the authors provide a thorough analysis of the CC LB idea, concentrating on AI-based meta-heuristic data optimization algorithms that optimize resource allocation to boost system performance and make LB more efficient. The performance measures that are used to gauge the performance of such algorithms are also explained in the paper, but special attention is given to how intelligent mechanisms of allocating resources may be refined to accommodate bigger or smaller requests. In most cases, the use of simple traditional scheduling methods does not work when dealing with complex and dynamic cloud environments. On the other hand, operators are used to carry out exploration, evaluation, and solution adaptation by the meta-heuristic algorithm, and this is the reason why they are appropriate in complex multi-objective optimization problems. In databases like IEEE Xplore, ScienceDirect, ACM Digital Library, Springer, and Elsevier, systematic filters were used to assess the relevance of the material used to build the review literature sources. In spite of the fact that AI-based meta-heuristic LB techniques have already been shown to be highly beneficial, several problems remain. The literature has focused more on centralized clouds, but much less on edge-cloud hybrid systems, which are a key paradigm under latency-sensitive and resource-constrained conditions. Moreover, meta-heuristic algorithms can become more and more complex, so the question of explainability is an urgent issue, particularly in situations where it is crucial to know why this or that decision was undertaken to trust and assume responsibility. The next step in the research should therefore be to come up with a clear, readable, and dynamic meta-heuristic algorithm that can run workloads successfully on hybrid cloud-edge systems. These gaps will play a vital role in the creation of scalable, energy-saving, and smart cloud systems capable of supporting the growing needs of existing applications.

References

[1] Bhargavi, K, Sathish Babu, B, and Pitt, Jeremy. "Performance Modeling of Load Balancing Techniques in Cloud: Some of the Recent Competitive Swarm Artificial Intelligence-based" Journal of Intelligent Systems, vol. 30, no. 1, 2021, pp. 40-58. https://doi.org/10.1515/jisys-2019-0084.

[2] Ghafir, Shabina, M. Afshar Alam, Farheen Siddiqui, and Sameena Naaz. "Load balancing in cloud computing via intelligent PSO-based feedback controller." Sustainable Computing: Informatics and Systems 41 (2024): 100948. https://doi.org/10.1016/j.suscom.2023.100948.

[3] Zhou, Guangyao, Wenhong Tian, Rajkumar Buyya, Ruini Xue, and Liang Song. "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions." Artificial Intelligence Review 57, no. 5 (2024): 124. https://doi.org/10.1007/s10462-024-10756-9.

[4] Forghani, Mohammadreza, Mohammadreza Soltanaghaei, and Farsad Zamani Boroujeni. "Dynamic optimization scheme for load balancing and energy efficiency in software-defined networks utilizing the krill herd meta-heuristic algorithm." Computers and Electrical Engineering 114 (2024): 109057. https://doi.org/10.1016/j.compeleceng.2023.109057.

[5] Simaiya, Sarita, Umesh Kumar Lilhore, Yogesh Kumar Sharma, KBV Brahma Rao, V. V. R. Maheswara Rao, Anupam Baliyan, Anchit Bijalwan, and Roobaea Alroobaea. "A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques." Scientific Reports 14, no. 1 (2024): 1337. https://doi.org/10.1038/s41598-024-51466-0.

[6] Geetha, Perumal, S. J. Vivekanandan, R. Yogitha, and M. S. Jeyalakshmi. "Optimal load balancing in cloud: Introduction to hybrid optimization algorithm." Expert Systems with Applications 237 (2024): 121450. https://doi.org/10.1016/j.eswa.2023.121450.

[7] Negi, Sarita, Man Mohan Singh Rauthan, Kunwar Singh Vaisla, and Neelam Panwar. "CMODLB: an efficient load balancing approach in cloud computing environment." The Journal of Supercomputing 77, no. 8 (2021): 8787-8839. https://doi.org/10.1007/s11227-020-03601-7.

[8] Elmagzoub, M. A., Darakhshan Syed, Asadullah Shaikh, Noman Islam, Abdullah Alghamdi, and Syed Rizwan. "A survey of swarm intelligence based load balancing techniques in cloud computing environment." Electronics 10, no. 21 (2021): 2718. https://doi.org/10.3390/electronics10212718.

[9] Khan, Mohammad Imran, and Kapil Sharma. "An Efficient Nature-Inspired Optimization Method for Cloud Load Balancing for Enhanced Resource Utilization." International Journal of Intelligent Systems and Applications in Engineering 12, no. 7s (2024): 560-571.

[10] Li, Cen, and Liping Chen. "Optimization for energy-aware design of task scheduling in heterogeneous distributed systems: a meta-heuristic based approach." Computing (2024): 1-25. https://doi.org/10.1007/s00607-024-01282-1.

[11] Singhal, Saurabh, Ashish Sharma, Pawan Kumar Verma, Mohit Kumar, Sahil Verma, Maninder Kaur, Joel JPC Rodrigues, Ruba Abu Khurma, and Maribel García-Arenas. "Energy Efficient Load Balancing Algorithm for Cloud Computing Using Rock Hyrax Optimization." IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3380159.

[12] Banupriya, M. R., and D. Francis Xavier Christopher. "Efficient Load Balancing and Optimal Resource Allocation Using Max-Min Heuristic Approach and Enhanced Ant Colony Optimization Algorithm over Cloud Computing." International Journal of Intelligent Systems and Applications in Engineering 12, no. 1s (2024): 258-270.

[13] Khan, Ahmad Raza. "Dynamic Load Balancing in Cloud Computing: Optimized RL-Based Clustering with Multi-Objective Optimized Task Scheduling." Processes 12, no. 3 (2024): 519. https://doi.org/10.3390/pr12030519.

[14] Selvakumar, Sadhana, and Pandiarajan Subramanian. "Intelligent and metaheuristic task scheduling for cloud using black widow optimization algorithm." Serbian Journal of Electrical Engineering 21, no. 1 (2024): 53-71. https://doi.org/10.2298/SJEE2401053S.

[15] Raghav, Yogita Yashveer, and Vaibahv Vyas. "Load Balancing Using Swarm Intelligence in Cloud Environment for Sustainable Development." In Convergence Strategies for Green Computing and Sustainable Development, pp. 165-181. IGI Global, 2024. https://doi.org/10.4018/979-8-3693-0338-2.ch010.

[16] Nebagiri, Manjula Hulagappa, and Latha Pillappa Hnumanthappa. "Multi-Objective of Load Balancing in Cloud Computing using Cuckoo Search Optimization based Simulation Annealing." International Journal of Intelligent Systems and Applications in Engineering 12, no. 9s (2024): 466-474.

[17] Rostami, Safdar, Ali Broumandnia, and Ahmad Khademzadeh. "An energy-efficient task scheduling method for heterogeneous cloud computing systems using capuchin search and inverted ant colony optimization algorithm." The Journal of Supercomputing 80, no. 6 (2024): 7812-7848. https://doi.org/10.1007/s11227-023-05725-y.

[18] Gong, Rong, DeLun Li, LiLa Hong, and NingXin Xie. "Task scheduling in cloud computing environment based on enhanced marine predator algorithm." Cluster Computing 27, no. 1 (2024): 1109-1123. https://doi.org/10.1007/s10586-023-04054-2.

[19] Khaleel, Mustafa Ibrahim, Mejdl Safran, Sultan Alfarhood, and Deepak Gupta. "Combinatorial metaheuristic methods to optimize the scheduling of scientific workflows in green DVFS-enabled edge-cloud computing." Alexandria Engineering Journal 86 (2024): 458-470. https://doi.org/10.1016/j.aej.2023.11.074.

[20] Barut, Cebrail, Gungor Yildirim, and Yetkin Tatar. "An intelligent and interpretable rule-based metaheuristic approach to task scheduling in cloud systems." Knowledge-Based Systems 284 (2024): 111241. https://doi.org/10.1016/j.knosys.2023.111241.

[21] Syed, Darakhshan, Ghulam Muhammad Shaikh, Hani Alshahrani, Mohammed Hamdi, Mohammad Alsulami, Asadullah Shaikh, and Syed Rizwan. "A Comparative Analysis of Metaheuristic Techniques for High Availability Systems (September 2023)." IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3487426.

[22] Behera, Ipsita, and Srichandan Sobhanayak. "Task scheduling optimization in heterogeneous cloud computing environments: A hybrid GA-GWO approach." Journal of Parallel and Distributed Computing 183 (2024): 104766. https://doi.org/10.1016/j.jpdc.2023.104766.

[23] Gupta, Swati, and Ravi Shankar Singh. "User-defined weight based multi objective task scheduling in cloud using whale optimisation algorithm." Simulation Modelling Practice and Theory (2024): 102915. https://doi.org/10.1016/j.simpat.2024.102915.

[24] Thilak, K. Deepa, K. Lalitha Devi, C. Shanmuganathan, and K. Kalaiselvi. "Meta-heuristic Algorithms to Optimize Two-Stage Task Scheduling in the Cloud." SN Computer Science 5, no. 1 (2024): 1-16. https://doi.org/10.1007/s42979-023-02449-x.

[25] Karimunnisa, Syed, and Yellamma Pachipala. "Deep Learning Approach for Workload Prediction and Balancing in Cloud Computing." International Journal of Advanced Computer Science & Applications 15, no. 4 (2024). https://doi.org/10.14569/IJACSA.2024.0150477.

[26] Kaur, Surinder, Jaspreet Singh, and Vishal Bharti. "A Comparative Study of Optimization Based Task Scheduling in Cloud Computing Environments Using Machine Learning." In 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), pp. 731-740. IEEE, 2024. https://doi.org/10.1109/ICICV62344.2024.00122.

[27] Brahmam, Madala Guru, and R. Vijay Anand. "VMMISD: An efficient load balancing model for Virtual Machine Migrations via fused Metaheuristics with Iterative Security Measures and Deep Learning Optimizations." IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3373465.

[28] Khaledian, Navid, Marcus Voelp, Sadoon Azizi, and Mirsaeid Hosseini Shirvani. "AI-based & heuristic workflow scheduling in cloud and fog computing: a systematic review." Cluster Computing (2024): 1-34. https://doi.org/10.1007/s10586-024-04442-2.

[29] Yakubu, Ismail Zahraddeen, and M. Murali. "An efficient meta-heuristic resource allocation with load balancing in IoT-Fog-cloud computing environment." Journal of Ambient Intelligence and Humanized Computing 14, no. 3 (2023): 2981-2992. https://doi.org/10.1007/s12652-023-04544-6.

[30] Boopathi, Ramya, and Erode Subramaniam Samundeeswari. "An Optimized VM Migration to Improve the Hybrid Scheduling in Cloud Computing." International Journal of Intelligent Engineering & Systems 17, no. 1 (2024). https://doi.org/10.22266/ijies2024.0229.42.

[31] Mishra, Sambit Kumar, Bibhudatta Sahoo, and Priti Paramita Parida. "Load balancing in cloud computing: a big picture." Journal of King Saud University-Computer and Information Sciences 32, no. 2 (2020): 149-158. https://doi.org/10.1016/j.jksuci.2018.01.003.

[32] Afzal, Shahbaz, and Ganesh Kavitha. "Load balancing in cloud computing–A hierarchical taxonomical classification." Journal of Cloud Computing 8, no. 1 (2019): 22. https://doi.org/10.1186/s13677-019-0146-7.

[33] Krishna Sowjanya, K., Mouleeswaran, S.K. (2023). Load Balancing Algorithms in Cloud Computing. In: Kumar, A., Ghinea, G., Merugu, S., Hashimoto, T. (eds) Proceedings of the International Conference on Cognitive and Intelligent Computing. Cognitive Science and Technology. Springer, Singapore. https://doi.org/10.1007/978-981-19-2358-6_45.

[34] Reddy, Raghavender, Amit Lathigara, and Rajanikanth Aluvalu. "Dynamic load balancing strategies for cloud computing." In AIP Conference Proceedings, vol. 2963, no. 1. AIP Publishing, 2023. https://doi.org/10.1063/5.0182748.
[35] Jena, Uttam Kumar, P. K. Das, and Manas Ranjan Kabat. "Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment." Journal of King Saud University-Computer and Information Sciences 34, no. 6 (2022): 2332-2342. https://doi.org/10.1016/j.jksuci.2020.01.012.
[36] Karimi Mamaghan, Maryam & Mohammadi, Mehrdad & Meyer, Patrick & Karimi Mamaghan, Amir Mohammad & Talbi, El-Ghazali. (2021). Machine Learning at the service of Meta-heuristics for solving Combinatorial Optimization Problems: A state-of-the-art. European Journal of Operational Research. 296. https://doi.org/10.1016/j.ejor.2021.04.032.
[37] Houssein, Essam H., Ahmed G. Gad, Yaser M. Wazery, and Ponnuthurai Nagaratnam Suganthan. "Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends." Swarm and Evolutionary Computation 62 (2021): 100841. https://doi.org/10.1016/j.swevo.2021.100841.
[38] Zhou, Jincheng, Umesh Kumar Lilhore, Tao Hai, Sarita Simaiya, Dayang Norhayati Abang Jawawi, Deemamohammed Alsekait, Sachin Ahuja, Cresantus Biamba, and Mounir Hamdi. "Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing." Journal of cloud computing 12, no. 1 (2023): 85. https://doi.org/10.1186/s13677-023-00453-3.
[39] Kumar M, Sharma SC (2017) Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing. Proced Comp Sci 115(C):322–329. https://doi.org/10.1016/j.procs.2017.09.141.
[40] Kumar J, Singh AK. Performance evaluation of metaheuristics algorithms for workload prediction in cloud environment. Applied Soft Computing. 2021 Dec 1;113:107895. https://doi.org/10.1016/j.asoc.2021.107895.
[41] Halim, A.H., Ismail, I. & Das, S. Performance assessment of the metaheuristic optimization algorithms: an exhaustive review. Artif Intell Rev 54, 2323–2409 (2021). https://doi.org/10.1007/s10462-020-09906-6.
[42] Zhang W, Chen L, Luo J, Liu J. A two-stage container management in the cloud for optimizing the load balancing and migration cost. Future Generation Computer Systems. 2022 Oct 1;135:303-14. https://doi.org/10.1016/j.future.2022.05.002.
[43] Tawfeeg TM, Yousif A, Hassan A, Alqhtani SM, Hamza R, Bashir MB, Ali A. Cloud dynamic load balancing and reactive fault tolerance techniques: a systematic literature review (SLR). IEEE Access. 2022 Jul 5;10:71853-73. https://doi.org/10.1109/ACCESS.2022.3188645.
[44] Kong, Lingfu, Jean Pepe Buanga Mapetu, and Zhen Chen. "Heuristic load balancing based zero imbalance mechanism in cloud computing." Journal of Grid Computing 18, no. 1 (2020): 123-148. https://doi.org/10.1007/s10723-019-09486-y.
[45] Hayyolalam V, Pourghebleh B, Pourhaji Kazem AA. Trust management of services (TMoS): investigating the current mechanisms. Transactions on Emerging Telecommunications Technologies. 2020 Oct;31(10):e4063. https://doi.org/10.1002/ett.4063.
[46] Houssein, Essam Halim, Mohamed Abd Elaziz, Diego Oliva, and Laith Abualigah, eds. Integrating meta-heuristics and machine learning for real-world optimization problems. Vol. 1038. Springer Nature, 2022. https://doi.org/10.1007/978-3-030-99079-4.