Journal of Information Systems Engineering and Management
2025, 10(7s)
e-ISSN:2468-4376
https://jisem-journal.com/

Research Article

Leveraging Machine Learning for Enhanced Cloud Computing Load Balancing: A Comprehensive Review

Swetha.G[1], Md Oqail Ahmad[2]
[1] Research Scholar, Department of CSE,
Vignan's Foundation for science, Technology & Research,
Guntur, Andhra Pradesh, India
Email: swethag4010@gmail.com
[2]Assistant Professor, Department of CSE,
Guntur, Andhra Pradesh, India
*Cooresponding author: oqail.jmu@gmail.com

ARTICLEINFO

Received: 08 Oct 2024

Revised: 10 Dec 2024

Accepted: 24 Dec 2024

ABSTRACT

Abstract: Popularized especially throughout the last decade, cloud computing has significantly changed the approach to the use of computational resources. But load balancing still persists as a very important issue of concern to achieve the right balance in order to optimize the various resources. This review paper aims at discussing the various Machine Learning (ML) techniques that can be used for the improvement of load balancing in cloud computing systems. Therefore, the analysis of the current methods is intended to reveal the advantages of the modern approaches to load balancing and put focus on the shortcomings of the traditional algorithms and the contribution of new machine learning approaches. ML techniques offer dynamic load balancing solutions which can handle the variance of the system load. These techniques mainly involve the use of historical data and applying statistical analysis to arrive at the best decisions thus reducing the time response, server usage, and resource utilization. Further, this paper presents how the state-of-the-art machine learning models can be implemented in the context of cloud environments focusing on the complexity and overhead, data protection, and scalability questions. It continues the review of the most advanced machine learning algorithms in the context of load balancing. Adding several examples of the current case studies and experiment results from the published researches of the recent years, the effectiveness of these techniques has been illustrated in the real cloud context. This paper is concluded with the generalization for the future works where the emphasis is placed for the improvement of load balance of machine learning for more demanding and secured cloud computing service.

Keywords: Cloud Computing, Load Balancing, Dynamic Resource Allocation, Scalability, Resource Optimization, Computational Overhead

INTRODUCTION

Cloud computing technology is becoming an indispensable component of businesses as we move more and more towards using online storage and services. This technology offers services in a variety of formats, including software that runs on web browsers and platforms that are used to create and build cloud-based applications. Cloud Service Providers (CSPs) oversee the backend of the infrastructure, which includes managing servers, data centres, and other equipment. The Infrastructure as a Service (IaaS) model is the main emphasis of this research, despite the fact that there are several additional service delivery models in this technology. It addresses how this technology allocates resources on the server side. The foundation and key component of cloud-based apps is virtualization (R. Begam et al., 2020; O. Cheikhrouhou et al., 2023). If the migration procedure and virtual machine resource allocation are handled inefficiently, this strategy can have a substantial impact on the efficiency of the scalable and on-demand services offered to customers. Cloud performance (Swetha, G and Ahmad, M.O (2024)), has been identified as one of the primary problems in cloud computing. The goal of this study is to improve resource allocation in the Infrastructure as a Service (IaaS) paradigm. This idea is important because it balances the resources supplied to clients with the workload and user requests on servers. Requests, which are embodied in VMs

5 in the cloud environment, are how cloud consumers obtain services. CSPs have to provide services that boost customer happiness and benefit enterprises. As a result, among the three cloud service models, the suggested load balancing method is primarily focused on the Infrastructure as a Service (IaaS) model, with the authors addressing the backend of cloud computing technologies, including server workload. A typical cloud environment consists of two parts: the user side, or frontend, is accessed by an Internet connection (M. A. Shahid et al., 2020). The cloud service models where the information centre stores many physical computers (called servers) are handled by the backend side. The program schedules incoming user requests dynamically and uses virtualization to assign clients to the appropriate resources. In addition, scheduling, effective resource allocation, and load balancing throughout the system are all handled by the virtualization approach. The allocation of the computational burden among several servers is known as load balancing. Load includes server storage capacity, network traffic volume, and CPU load. The workload is transmitted to the servers via the clients' requests (O. Cheikhrouhou et al., 2023; S. S. Sefati et al., 2022).

Distributed system managers use the notion of load balancing to split, distribute, and distribute resources among many servers, networks, and PCs. The capacity to handle expanding hardware architecture and computational demands is offered by load balancing. Ideally, the system's performance does not significantly change as the number of input variables rises. Cloud computing is implemented in several ways. Several topologies can be used in the architecture and configuration of the cloud environment (W.-Z. Zhang et al., 2021). The Private Cloud, Public Cloud, Hybrid Cloud, and Community Cloud are some of these topologies. A cloud provider sets up a private cloud, and the application and infrastructure are fully managed by the application provider. One organization's demands are the only ones served by the cloud (Shafiq et al., 2022). An open setup approach with third-party infrastructure facilities is called a public cloud. Resource sharing across many businesses or customers is made possible by this structure. Anyone can use computer resources through the public cloud by paying a membership fee. The cloud provider owns and maintains all of the underlying software, hardware, and other infrastructure. The most affordable option for application hosting is this model (D. Ding et al., 2020). With hybrid cloud, the trust paradigm is superseded and private and public clouds are combined. Interoperability and mobility of data and applications are necessary for both public and private clouds in order to facilitate communication across the models. When weighed against the private cloud, this solution is less costly. Data and apps are shared between the public cloud and on-premises data centers. Extranets and community clouds are comparable, but community clouds offer on-demand dedicated virtualization. A shared cloud is created by an organization with shared objectives or by a particular community for usage by its members.

Many tenants of the community cloud have similar worries about performance, security, and cloud reach. The services provided are restricted to the community members' needs and computing requirements. The provider of cloud computing services must guarantee the services' accessibility, availability, and speed of access. Growing customer demand for cloud computing necessitates expanding computing infrastructure and improving fault tolerance and resilience. The host must master load balancing, fault-tolerance, virtualization, and cloud security in order to sustain high- quality cloud services (M. Yadav et al., 2020; C. S. M. Babou et al., 2020). By preventing nodes from being overloaded or under loaded, load balancing ensures a balanced allocation of computing resources. As a result, the entire distributed system's speed and throughput are increased. Additional important benefits of load balancing include cloud scalability, which is achieved by starting various services to handle the increasing requests and efficiently dispersing the increased burden to the new virtual server instances. The load balancer's ability to identify idle nodes and recently installed servers is another crucial function in cloud environments. It is the responsibility of the LB component to route new requests to the servers that are found. Having the LB component manage cloud service continuity in the event of catastrophic failures is another crucial aspect of disaster recovery. Requests are redirected to servers that are available by the load balancer. By maintaining service availability in the face of internal malfunctions or heavy workloads, it demonstrates transparency to the user.When it comes to distributing the requests intelligently, the perfect load balancer should behave intelligently (Peng et al., 2019). Efficient resource allocation entails distributing the load equally across all servers at all times. The study of load balancing has moved away from conventional load scheduling algorithms like min-min and round-robin and toward smart load balancing models that utilize machine learning and deep learning as a result of the rise and development of artificial intelligence (R. Begam et al., 2020). By guaranteeing service quality and adherence to set service level agreements (SLAs), a smart load balancer gives cloud providers a competitive edge. Highest response and throughput performance, online traffic management, efficient handling of

ad hoc traffic bursts and surges, and adaptability are a few of its primary significances (S. Negi et al., 2021). The elasticity in the sense of scalable computational queries on client demand is provided by load balancers. Because resource demand is constantly increasing and charging every second, cloud management faces unpredictability. A few of the physical and conceptual issues that make load balancing more difficult are as follows: Since cloud providers have data centres spread across many countries and cloud customers anticipate uninterrupted cloud performance, the physical location of the data centres presents some logistical and reaction time problems for the load balancer. The edge computing problem, which suggests processing cloud requests close to their source, presents another difficulty. Another issue with load balancing is VM migration, which occurs when physically overcrowded computers are forced to move part of their virtual machines to a physically underutilized machine (S. Negi et al., 2021). The difficult element is moving the virtual machine's memory pages from their current location over the network without compromising the services it provided. A significant amount of CPU, memory, and network bandwidth are used during live virtual machine migration, which can easily cause SLA violations. The load balancing technique's algorithmic complexity is called into doubt by the balancing approaches' complexity (Khan and Ahmad, 2020). Fault tolerance and hardware requirements shouldn't be very complicated for an effective load balancer. In order to maintain optimal performance, good strategies must make trade-offs. Because heterogeneous nodes differ in terms of memory, networking components, and processing power, they pose an additional research challenge for load balancers. Different machine architectures (GPUs, CPUs, and multi core CPUs) are present in the nodes. The issue is establishing a consistent system to divide the indifferent resources through job assignment (Peng et al., 2019). When one load balancer component is the only one used by the whole system, this is known as a single point of failure (M. Yadav et al., 2020; M. I. Alghamdi 2020). To construct a failover solution and transfer the burden sharing component to another load balancer inside the same cloud network, a redundant load balancing component is required. Having a main load balancer and a backup load balancer that switches between them on failure is recommended by certain circumstances (C. S. M. Babou et al., 2020; P. K. Bal 2022). Another aspect of load balancing design to take into account is scalability. In order to meet the increasing demand for cloud services, cloud providers install more linked nodes. The load balancing element may be scaled and more hardware added with the help of a scalable load balancer. After task scaling, the load balancer should keep the performance constant. It keeps the ratio of used to unused resources in check (Jena et al., 2022).

This paper's goal is to evaluate the load balancing strategies currently used in cloud computing. It aims to give a thorough grasp of existing approaches and point out any shortcomings. The article will examine several approaches static, dynamic, heuristic, and optimization and emphasize the difficulties in establishing effective load distribution, including managing heterogeneous workloads and preserving flexibility in dynamic cloud systems. This research also suggests improvements to tackle these issues with machine learning. It investigates how load balancing may be optimized using ML methods, such as supervised, unsupervised, and reinforcement learning approaches, by anticipating workload patterns, adjusting to changing circumstances, and enhancing resource use. The study seeks to provide novel solutions that improve overall system performance and reliability through the integration of machine learning, hence leading to more effective and flexible cloud computing services.

The problem statement of effective load balancing in cloud computing remains a critical challenge, particularly as cloud environments become increasingly dynamic and complex. Traditional load balancing techniques often struggle to adapt to the unpredictable nature of workloads, leading to inefficiencies such as bottlenecks, underutilization of resources, and increased latency. While recent advancements have introduced machine learning and deep reinforcement learning methods to enhance load distribution, there is a significant gap in the development of unified, adaptive systems that can efficiently handle diverse and fluctuating workloads in real-time (Jena et al., 2022). Furthermore, existing research predominantly focuses on theoretical models, with limited exploration of practical, large-scale applications and security considerations, leaving a crucial need for robust, scalable, and secure load balancing solutions in cloud computing (Swetha, G. and Ahmad, M.O., 2024)

RESEARCH OBJECTIVES

The objectives of this comprehensive review are as follows:

1. To analyze and summarize the various machine learning approaches, used in load balancing within cloud computing environments. Highlight their effectiveness in improving resource allocation, latency reduction, and cost optimization.

2. To compare traditional and machine learning-based load balancing techniques in terms of their advantages, disadvantages, and applicability. This includes evaluating techniques like Round Robin, Least Connections, Weighted Round Robin, and Dynamic Load Balancing, and assessing how ML-driven methods enhance these traditional approaches.
3. To investigate emerging trends and potential future research areas in the integration of machine learning with cloud computing load balancing. Address current challenges, such as workload prediction accuracy, integration with existing systems, and balancing performance with cost and security. Propose future research directions, including self-learning systems, energy efficiency, and applications in new technologies like edge computing and IoT.

RESEARCH MOTIVATION

The study focuses on a systematic review to generally describe how ML can significantly enhance load balancing in cloud computing. An integral part of modern computing, cloud systems range from large- scale, data-center-sized systems for computation and data-intensive applications, through intermediate levels such as companies' own internal clouds, down to everyday consumer applications running on the public cloud. Effective load balancing is critical to the optimal utilization of resources, minimization of operational costs, and meeting the promising QoS of good response time, efficient energy consumption, and scalability. Despite the huge number of publications investigating traditional and machine learning-based load balancing techniques, there seems to be a vacant area for an exhaustive review that compares such methods in dynamic cloud environments. Most of the surveys published recently are concentrated on particular algorithms or merely briefly discuss mechanisms for proposed solutions. In addition to these features, the adaptability of practical environment, power consumption, and cost-effectiveness are important factors that are left untouched by many reviews-of paramount importance for the changing nature of cloud computing.

The contribution will bridge this gap in literature by critically analyzing and classification of ML-driven load balancing techniques with their pros and cons. In this review, we hope to show potential researchers and users in practice which algorithms will best achieve better performance in cloud systems along with cost-effective lower migrations and greater resource utilization. This will both support good usage of large-scale infrastructures in the clouds and further advancements in the field of adaptive load-balancing techniques in the future. This review thus provides an operational- level understanding of the state-of-the-art techniques and their practical applications to be of greatest help for advancing the field and guiding further research in the frontier fields of computational complexity.

The rest of this paper is organized as follows: section 2 background and review organization which contains a thorough list of research questions, data sources, and study criteria. Section 3 is Discussed different types of load balancing Techniques in Section 4 Summary of Review Process, in Section 5 Research Gap in Section 6 Comparative Analyses of the Reviewed Papers in section 7 Applications of Machine Learning-Driven Load Balancing in Cloud Computing. In Section 8 Challenges in Section 9 Metrics and Tools in Section 10 Conclusion and Future Works.

BACKGROUND AND REVIEW ORGANIZATION

The next paragraphs outline the survey framework, which contains a thorough list of research articles, data sources, and study criteria.

Research Questions

How do machine learning algorithms, influence the load distribution and efficiency in cloud computing environments?

What is the comparative effectiveness of traditional load balancing methods versus machine learning- based approaches in improving performance metrics such as makespan, CPU time, and load distribution in cloud computing?

In what ways do cloud service providers perceive the accessibility and usability of machine learning- based load balancing techniques, and how does this perception impact their implementation and effectiveness?

What are the correlations between the uses of machine learning-based load balancing techniques in cloud computing and the overall performance, cost efficiency, and scalability of cloud services?

The questions can be answered by providing accurate and efficient Cloud backend services and making sure that load balancing is done optimally by following the researcher's criteria and the review article. Below are the decisive responses.

1. How do machine learning algorithms, influence the load distribution and efficiency in cloud computing environments?

Intelligent decision making on load distribution comes through ML algorithms, especially those that are self-learning and adaptable. From real-time and historical data, they can predict resource requirements, allowing the cloud to react to dynamic changes in workloads automatically. By discovering usage patterns, the ML algorithms help optimize not only resource consumption but also prevent bottlenecks in the system, making it possible for cloud infrastructures to provide better functionality with faster response times and minimal time wasted on downtime. This is important in managing the dynamic nature of cloud environment, in which classical static or dynamic load balancing methods cannot always be responsive enough.

2. What is the comparative effectiveness of traditional load balancing methods versus machine learning-based approaches in improving performance metrics such as makespan, CPU time, and load distribution in cloud computing?

Traditional load balancing methods, such as round-robin or first-come-first-serve, follow predefined rules and are generally efficient in static environments with predictable loads. However, in more dynamic cloud scenarios with fluctuating demand, ML-based approaches offer significant advantages. They can dynamically adjust load balancing decisions in real-time, minimizing makespan (the total time required to execute all tasks), optimizing CPU usage, and distributing loads more effectively. By learning from the environment and past decisions, ML algorithms offer better scalability, adaptability, and overall system performance compared to their traditional counterparts.

3. In what ways do cloud service providers perceive the accessibility and usability of machine learning-based load balancing techniques, and how does this perception impact their implementation and effectiveness?

Cloud service providers' perceptions of ML-based load balancing significantly influence their adoption and effectiveness. Providers who recognize the potential of ML algorithms to improve system performance, reduce energy consumption, and manage large-scale operations more efficiently are more likely to invest in implementing these techniques. However, if providers view ML solutions as complex or resource-intensive to implement, their uptake may be slower, potentially impacting overall performance gains. Understanding the barriers and facilitators to adopting these technologies can help refine ML-based approaches to become more user-friendly and accessible.

4. What are the correlations between the uses of machine learning-based load balancing techniques in cloud computing and the overall performance, cost efficiency, and scalability of cloud services?

Load-balancing techniques by using ML provide optimization for the performance of cloud systems with resource efficiency along with assured distribution of workloads, thereby creating better performance in the system while cost is saved in the form of reduced energy consumption and operational costs, and better scalability. In line with growth in the complexity of cloud services, there is an appearance of correlation between ML load balancing and system performance—ML approaches ensure the ability to scale quickly with cost-effectiveness, which provides higher-quality services at a better price to users.

Search Engines Selection Criteria

This study focused on machine learning techniques for enhancing cloud computing load balancing andprovidedan in-depth analysis ofvarious approaches.The following key words were used: cloud computing, machine learning, load balancing, dynamic algorithms, metaheuristic algorithms, deeplearning, reinforcement learning, QoS parameters, and security in load balancing. The details of the search engines use dinthis study are presented in Table1.

Table 1. Search Engines Selected for the Evaluation

Journal	Link (Accessed Date)
IEEE Xplore	https://ieeexplore.ieee.org/ (accessed on 15 Sep 2024)
ACM Digital Library	https://dl.acm.org/ (accessed on 12 Aug 2024
ScienceDirect	https://sciencedirect.com (accessed on 18 Aug 2024)
SpringerLink	https://springer.com (accessed on 20 Sep 2024
Elsevier	https://elsevier.com (accessed on 25 Aug 2024
Hindaw	https://hindawi.com (accessed on 10 Sep 2024
Wiley Online Library	https://onlinelibrary.wiley.com (accessed on 22 Sep 2024)

Inclusion and Exclusion Criteria

Initially, 150 records were identified through database searching. After eliminating duplicates, the total number of records was reduced to 130. These 130 records underwent a screening process, resulting in 120 screened records. Out of these, 20 records were excluded based on specific criteria: 10 were not related to load balancing, 5 were not related to cloud computing, and 5 were review papers. This left 100 articles that were evaluated for eligibility. During this evaluation phase, 12 full-text articles were excluded due to insufficient reported data. Consequently, 78 papers were deemed suitable and included in the comprehensive survey. This process ensures that only the most relevant and high- quality studies are included in the review, providing a robust analysis of the current advancements in load balancing techniques in cloud computing using machine learning. The Figure 1 effectively highlights the rigorous selection process and the reasons for exclusion at each stage, ensuring transparency and clarity in the review methodology.

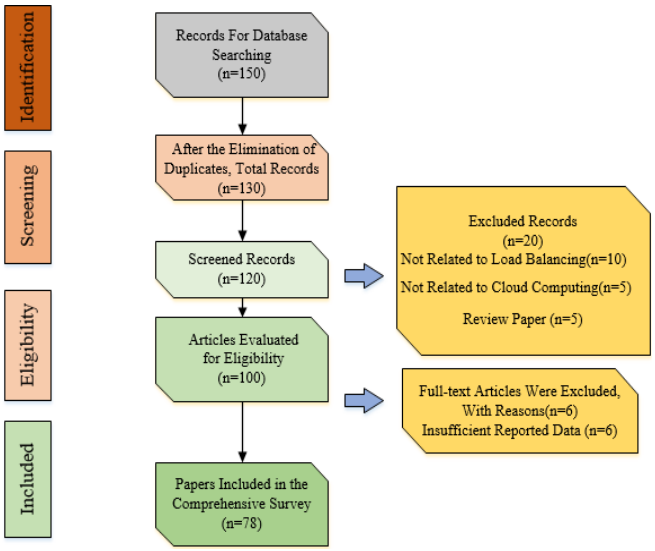

Figure1. Comprehensive Literature Methodology

The inclusion and exclusion criteria for the study on enhancing load balancing techniques in cloud computing using machine learning are summarized in Table 2. The inclusion criteria focus on studies published between 2020 and 2024 that involve cloud computing environments and load balancing techniques, specifically utilizing machine learning algorithms. These studies must include comprehensive evaluations of performance metrics such as make span, CPU time, and load distribution, and be conducted in English to avoid language barriers. Furthermore, the included studies must demonstrate clear performance improvements in load balancing techniques and use

accessible machine learning tools for replication and implementation. Conversely, the exclusion criteria filter out studies unrelated to cloud computing or load balancing, those that do not incorporate machine learning, or those lacking comprehensive evaluation metrics. Studies published before 2020, conducted in languages other than English, or without clear outcome measures related to load balancing performance are also excluded. Additionally, studies employing proprietary or inaccessible techniques and tools are disregarded to maintain the replicability and applicability of the research findings.

Table 2. Inclusion and Exclusion Criteria for Systematic Literature Review

Criteria	Inclusion	Exclusion
ResearchFocus	Studies focusing on machine learning algorithms for load balancing in cloud computing environments.	Studies unrelated to machine learning or load balancing techniques in cloud computing.
Methodologies	Research utilizing machine learning algorithms for load balancing.	Research that does not incorporate machine learning techniques for load balancing.
StudyDesign	Studies with comprehensive evaluations including performance metrics like makespan, CPU time, and load distribution.	Studies lacking comprehensive evaluation metrics or focusing on non-performance-related aspects.

The study aims to investigate the impact of modern load balancing techniques using machine learning on the efficiency and performance of cloud computing environments. The research seeks to address key questions such as the influence of machine learning algorithms on load distribution, the comparative effectiveness of traditional load balancing methods versus machine learning approaches, the perception of usability and accessibility of these techniques by cloud service providers, and the correlations between the use of machine learning-based load balancing and the overall performance, cost efficiency, and scalability of cloud services. Table 2 provides a comprehensive overview of limitations, reasons, and proposed solutions associated with the study. These insights are crucial for understanding the potential challenges in the research design, data collection, and analysis processes, offering a roadmap to enhance the study's robustness and validity. The limitations identified in Table 2 provide a transparent acknowledgment of potential constraints, while the reasons interpret the context and contributing factors. The proposed solutions, meanwhile, offer strategic approaches to mitigate the identified limitations, ensuring the study's reliability and contributing valuable insights to the broader discourse on load balancing in cloud computing using machine learning.

RECENT ADVANCES IN CLOUD LOAD BALANCING: METAHEURISTIC, MACHINE LEARNING, AND HYBRID OPTIMIZATION TECHNIQUESS

Several proposals in the form of metaheuristic algorithms, machine learning techniques, and hybrid optimization strategies have been reported in the key field of cloud load balancing during recent years, particularly when dealing with problems concerning the challenges of resource utilization, scalability, and energy efficiency. These mainly aim at optimizing dynamic and adaptive load balancing techniques in efforts to contend with an increasingly complex environment that poses demands on management. Hybrid optimization techniques, in particular Firefly-Imperialist Competitive Algorithm and GWO-PSO, have shown significant promise to improve convergence and stability of the system but suffer from the same problems of computational complexity.

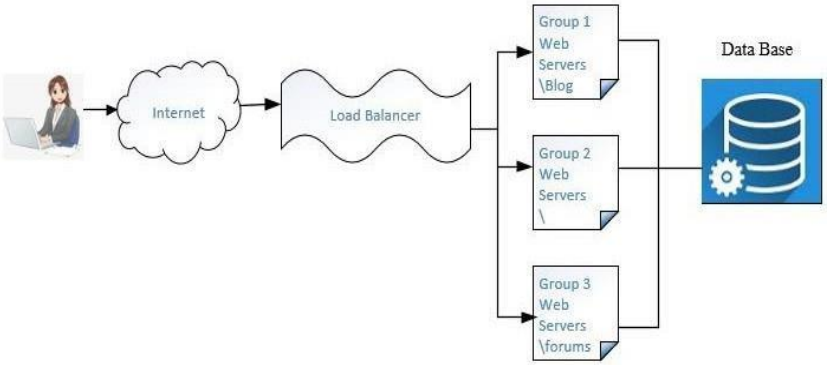

Figure2. Working of Load Balancer in Cloud Computing

Figure 2 shows the working of load balancer in cloud computing is given below. This section covers the different techniques used for load balancing, along with their applications, and current research being conducted to enhance the effectiveness of these techniques and improve performance in a cloud computing environment.

Metaheuristic Algorithms for Load Balancing

Metaheuristic algorithms in the domain of cloud load balancing have been applied in a good number of studies. (Negi et al., 2021, P. Neelima and A. R. M. Reddy, 2020) have proposed CMODLB: A hybrid load balancing approach, referred to as clustering-based multi objective dynamic load balancing technique (CMODLB), is developed to balance the cloud load. It combines elements of unsupervised ML, supervised ANN, and soft computing (interval type 2 fuzzy logic systems). The drawbacks are difficulty in implementation and maintenance, potential scalability issues in large-scale cloud environments and higher requirements for computational resources and expertise. Bal et al. (Z. Xu et al., 2024) proposed RATS-HM, a hybrid approach of machine that effectively improved both the make-span reduction and throughput but had problems associated with complexity and computational overhead. Below are the RATS-HM techniques: The complexity of the hybrid approach, which combines multiple techniques. This complexity lead to increased computational overhead, difficulty in implementation, and potential scalability issues in large-scale cloud computing environments. (A. Ghasemi and A. Toroghi Haghighat, 2020) introduced BPSO which improved the efficiency of load balancing and job scheduling with problems in highly dimensional space and local optima; this can be addressed by the improvement of diversity in optimization techniques. One potential drawback of the ANN-BPSO algorithm is that it does not consider energy consumption, which is a critical parameter in cloud computing environments. This would lead to suboptimal solutions that neglect the energy efficiency aspect, potentially resulting in increased energy costs and environmental impact. (Shafiq et al., 2022) proposed a Dynamic Load Balancing Algorithm (DLBA) that significantly reduces resource usage and shortens execution times. However, it falls short in fully considering all SLA parameters. To address this, the purpose is to enhance SLA integration and develop adaptive SLA management strategies. Similarly, Ebadifard and Babamir (P. Yang et al., 2024) introduced a dynamic scheduling method using CloudSim, which improved workload distribution but faced communication overhead challenges. The goal here is to optimize communication protocols to reduce overhead and implement predictive algorithms for preemptive load balancing.

(Neelima and Reddy, 2020 and S. M. G. Kashikolaei et al., 2020) developed the Adaptive Dragonfly Algorithm (ADA) for load balancing, which improved execution time and cost but struggled with the potential for getting trapped in local optima in complex scenarios. The solution aims to refine ADA parameters and integrate real-time adaptive strategies. (M. S. Al Reshan et al., 2023) introduced a task scheduling scheme using deep learning and reinforcement learning, which enables real-time monitoring and optimal resource utilization, though it suffers from high computational overhead. To mitigate this, more efficient model training and resource allocation strategies are needed.

Machine Learning and Reinforcement Learning Techniques

Machine and reinforcement learning are applied to learn and adapt for efficient load balancing. An ML-based VM replacement strategy was proposed by (Ghasemi and Toroghi Haghighat 2020) to enhance the CPU, memory, and bandwidth load balance; however, high computation demand remains as a challenge. In the next episodes, the learner agent will be able to pick and carry out the optimal action in each environment state in order to achieve further development, thanks to the environment rewarding them and allowing them to update the action value table. Prior to the algorithm's adoption, the inter-HM load balance in terms of CPU, memory, and bandwidth had increased by, on average, 25%, 34%, and 32%, respectively. It was determined that our suggested solution, which used significantly less runtime and turned off more HMs, was more successful in load balancing. The drawback is the approach requires significant computational resources and runtime to learn and update the action value table, which could be a limitation in environments where real-time decision-making is crucial. Optimization of learning algorithms to minimize overhead has been suggested as one of the possible ways to overcome this. (Karthiban and Raj, 2020.) have applied deep reinforcement learning for resource distribution. With this, there would be more elegant allocation schemes; however, scalability became a challenge and can be overcome using distributed computing and model optimizations. Yang et al. (S. Wang et al., 2020) offered a scalable policy for load balancing, based on neural networks that significantly reduces idleness and promotes improvements in balancing loads; such simulations need to be done by using real data to increase the accuracy.

Hybrid optimization techniques are emerging in recent times as efficient methods towards solving modern complex load balancing problems. Kashikolaei et al. (E. Mathanraj and R. N. Reddy, 2024) integrated the Firefly Algorithm with the Imperialist Competitive Algorithm and in the process improved the criteria of make span, load distribution, and system stability, despite its increased computational need. Hybrid algorithms have improved convergence strategies and more efficient optimization of local and global search capabilities. Al Reshan et al. (P. Li et al., 2024) presented a hybrid GWO-PSO approach, which achieved rapid convergence and better performance but has issues regarding computational complexity that could be addressed by parallel processing in achieving real-time processing. Khan (B. Kruekaew and W. Kimpan, 2022) made the presentation of Hybrid Lyrebird Falcon Optimization (HLFO), combining reinforcement learning for multi objective optimization but also presented challenges regarding complexity and implementation in various cloud environments. By merging DL, reinforcement learning, and hybrid optimization approaches, this research tackles the complexity of dynamic load balancing in cloud systems and provides a complete solution to improve cloud performance under shifting workloads and resource situations. Jyoti and Shrimali (M. Sudheer et al., 2023) developed MADRL-DRA with a Dynamic Optimal Load-Aware Service Broker, which enhances dynamic resource allocation and improves throughput but relies heavily on VM cost analysis, necessitating enhanced learning models and diverse cost metrics. (D. Tennakoon et al., 2023) introduced the Improved Firework Algorithm (IFA), which adapts to dynamic network conditions but suffers from resource insufficiency due to clustering inefficiencies, prompting the need for refined clustering methodologies. (E. Mathanraj and R. N. Reddy, 2024) proposed the Principal Component Gradient Round Robin Load Balancing (PCGRLB) method, which increases load balance efficiency but faces complexity when handling varied workloads. The objective is to develop adaptive load balancing mechanisms for diverse workload types. (P. Li et al., 2024) introduced a meta-optimization algorithm based on natural selection principles, showing improved throughput and scalability, but the design remains complex. Streamlining the algorithmic approach for better performance is the proposed solution.

Dynamic and Adaptive Load Balancing Algorithms

The real-time resource allocation methods indicate that dynamic load balancing is found promising. The MOABCQ algorithm found by (B. Kruekaew and W. Kimpan, 2022) with a better factor of make span and cost reductions than the methods available in the literature suffered from uncertainty over their optimization potential, further suggesting the requirement for fine-tuning optimization algorithms based upon the characteristics of the dataset. The MOABCQ_LJF algorithm's potential for optimization remained uncertain, which is a disadvantage. However, it is not possible to maximize the system's performance in every test dataset. (M. Sudheer et al., 2023) suggested a deep Q-learning network model with very impressive reductions in make span, SLA violations, and energy consumption, although the high computation requirements at the starting phase can be compensated by employing incremental learning techniques. (D. Tennakoon et al., 2023) proposed a modified double Q-learning algorithm to reduce unsatisfied cloud consumers with high computational complexity, which is a concern that can potentially be

solved with optimized data structures. (T. Renugadevi et al., 2020) achieved effective workload distribution with their Task Distribution Algorithm (TDA), ensuring energy efficiency in datacenters, yet complexity arises when managing diverse workload demands, warranting the development of adaptive mechanisms.

Some studies were based on energy efficiency along with load balancing. (C. Dutta et al., 2024) came up with a Gallant Ant Colony Optimized Machine Learning Framework, named GACO-MLF, in order to enhance the load balancing of IoT and PCN environments. However, it was tough for large environments due to scalability. (Singhal et al., 2024) proposed Rock Hyrax-based load balancing algorithm; however, dynamic job and VM management require refinement. Velliangiri et al. (Velliangiri et al., 2021) developed a Hybrid Electro Search with Genetic Algorithm (HESGA), which reduces make span and cost but increases latency due to long execution times of the genetic algorithm, thus necessitating optimization of genetic algorithm execution. Brahmam and R (M. G. Brahmam and V. A. R, 2024) combined load balancing and energy-efficient migration models with deep reinforcement learning enhancements, achieving improved performance metrics but facing complexity in integrating multiple advanced techniques, calling for further optimization. (E. Mathanraj and R. N. Reddy, 2024) utilized the Principal Component Gradient Round Robin Load Balancing (PCGRLB) method for higher load balance efficiency, yet complexity arises in managing varied workload sizes, highlighting the need for adaptive load balancing mechanisms. Lastly, (J. Ramkumar et al., 2023) introduced the Gallant Ant Colony Optimized Machine Learning Framework (GACO-MLF) to enhance load balancing in IoT-PCN but encountered scalability challenges in complex environments, prompting the need for optimization and refinement of ant colony algorithms. (P. Neelakantan and N. S. Yadav, 2023) reduced job execution time with their Whale-Based Convolution Neural Framework (WbCNF) but faced higher computation times for complex tasks, suggesting the implementation of parallel processing techniques to expedite computation. (Naiem et al., 2024) enhanced a Gaussian Naïve Bayes classifier for detecting DDoS attacks, achieving improved accuracy, though it is hindered by the zero-frequency issue and the independence assumption. The proposed solution involves using iterative feature selection and SMOTE for handling data imbalances. (F. Ramezani Shahidani et al., 2023) introduced a reinforcement learning-based fog scheduling algorithm that improves load balancing in fog computing, but real-time dynamic workload handling remains a challenge. The focus is on continuous learning and adaptation for evolving workloads. Lastly, (J. Ramkumar et al., 2023) developed the Gallant Ant Colony Optimized Machine Learning Framework (GACO-MLF) for enhanced load balancing in IoT environments, though scalability issues arise in larger, more complex settings. Optimizing ant colony algorithms for larger datasets is the proposed direction for improvement.

These studies reflect that metaheuristic along with machine learning-based approaches can address the issues related to the challenges of load balancing within dynamic cloud computing environments. Due to such efforts, some research has been focused on its refinement so as to obtain better performance scalability and energy efficiency.

Emerging Trends in AI-Driven Load Balancing Techniques

Recent works also include (P. Yang et al., 2023) suggested a scalable policy based on neural networks to solve the online bi-criteria load balancing problem in financial services, obtaining over 130% less server idleness compared to the traditional method. Based on Federated Learning improvement, (Jiang et al., 2024) designed and implemented FedMP, an efficient framework based on adaptive model pruning, which improved training performance and reduced resource consumption. On the same topic, (Yuan et al., 2024) proposed FedNAS: a novel federated neural architecture search algorithm that efficiently automates model generation with up to 200% cost savings on the client side while maintaining model accuracy. Further, (K. Rajakumari et al., 2021) presented a Dynamic Weighted Round-Robin (DWRR) algorithm that delivers performance improvements in cloud computing task scheduling by optimizing resource allocation based on task priorities and lengths. Building on this, (M. Núñez-Merino et al., 2024) conducted a multi-case study in investigating the role of quantum-inspired computing technology in Industry 4.0. Their paper proposes a model that develops this technology in connection with capabilities for improved manufacturing and logistics. In connection with fog computing, (Baburao et al., 2024) proposed EDRAM, employing particle swarm optimization in load balancing to enhance QoE and reduce latency. In the final account, (S. Aslam et al., 2021) outlined in detail deep learning methods applied to forecasting renewable energy sources power generation and load forecasting with an emphasis on the volume of historic data as holding the key for the efficiency of the models.

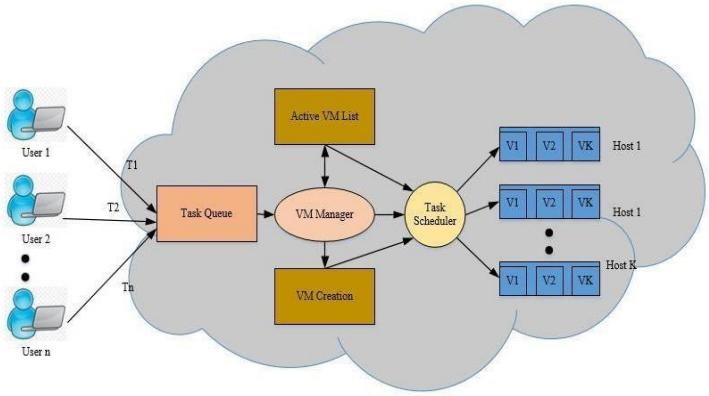

Figure 3. Task Scheduling in Cloud Computing with Active VM Management

The Figure 3 illustrates a cloud computing architecture where tasks from multiple users are queued and managed by a VM Manager, which oversees task allocation to virtual machines (VMs). A task scheduler assigns tasks to the appropriate VMs across multiple hosts. The active VM list and VM creation processes ensure resource availability and efficient task execution.

Hybrid Algorithms for Cloud and Edge Load Balancing

(H. Wang and B. Alidaee, 2023) proposed a hybrid-heuristic framework that combines Scatter Search, Tabu Search, and Genetic Algorithm to tackle large-scale combinatorial optimization problems. Their approach achieves new best solutions for 61 out of 70 instances of the quadratic assignment problem, showcasing the power of automated heuristics in complex problem-solving. (Devaraj et al., 2020) introduced the FIMPSO algorithm, which integrates the Firefly algorithm with Improved Multi-Objective Particle Swarm Optimization for efficient workload distribution in cloud computing. Their research demonstrates significant improvements in response time (13.58 ms) and CPU utilization (98%), highlighting the effectiveness of hybrid optimization techniques. (R. Etengu et al., 2020) discuss the challenges of managing explosive data Growth in IP networks and proposes a hybrid SDN/OSPF framework that leverages deep reinforcement learning for energy-efficient routing. Their approach aims to reduce global network energy consumption while maintaining quality of service, addressing technical and organizational hurdles in network management. (X. Wei, et al. 2020) addresses existing challenges in task scheduling optimization within cloud computing by proposing an improved ant colony optimization algorithm. This model focuses on minimizing waiting time, balancing resource loads, and reducing task completion costs, demonstrating enhanced performance metrics, including faster convergence and higher resource utilization. (K. Ramya and S. Ayothi, 2023) present the HDWOA-LBM, a hybrid load balancing mechanism that combines the dingo and whale optimization algorithms to effectively allocate tasks to virtual machines. Their simulations reveal significant improvements in throughput, reliability, and resource allocation, underscoring the need for dynamic load balancing strategies in cloud environments. (N. Rana et al., 2021) introduce the M-WODE technique, a hybrid multi-objective optimization algorithm for virtual machine scheduling, combining whale optimization and differential evolution. Their experimental results indicate that M-WODE significantly outperforms existing algorithms in terms of makespan and cost trade-offs, proving its effectiveness for VM scheduling. (S. V. Nethaji and M. Chidambaram, 2022) highlight the critical role of load balancing in fog computing for IoT applications and propose the DGW-SBDR algorithm, which integrates differential evolution and deep reinforcement learning. This approach enhances resource management by optimizing job allocation across virtual machines, improving quality of service by reducing energy consumption.

Dynamic Load Balancing in Edge Computing

(F. M. Talaat et al., 2020) proposed the Load Balancing and Optimization Strategy (LBOS) using reinforcement learning and genetic algorithms for dynamic resource allocation in fog computing. This approach monitors server loads and evenly distributes requests, achieving an 85.71% load balancing level, enhancing the quality of service in IoT-based healthcare systems. (M. Kaur and R. Aron, 2022) developed a resource-utilization-based workflow execution model for fog computing, incorporating the PSW-Fog Clustering algorithm. This hybrid approach

reduces latency in scientific workflows by balancing load and minimizing energy consumption and costs, outperforming traditional methods according to evaluations with the iFogSim toolkit. (Z. Nezami et al., 2021) introduced EPOS Fog, a decentralized multi-agent system for optimizing load balancing in IoT service placement within fog computing. Their method addresses global workload balance and local quality of service, achieving up to 25% reduction in service execution delay and 90% improvement in load balance. (L. Wang et al., 2020) proposed a UAV-aided mobile edge computing framework that optimizes load among user equipment (UE) on the ground through multiple UAVs with varying trajectories. Their multi-agent deep reinforcement learning algorithm shows significant improvements in geographical fairness and energy efficiency compared to traditional approaches. (S. Wan et al., 2020) developed the COV method for computation offloading in the Internet of Vehicles (IoV) within a 5G edge computing framework, tackling overloaded edge nodes. Their multi- objective optimization problem effectively minimizes offloading delays and costs while balancing loads among edge nodes. (B. Mahato et al., 2021) introduced a biogeography-based optimization (BBO) technique to enhance virtual network functions (VNFs) in network function virtualization (NFV) on cloud platforms. This method improves resource utilization through cooperative bandwidth sharing, resulting in higher bandwidth efficiency and reduced delays and costs. (X. Xu et al., 2022) proposed F-TORA, a task offloading scheme utilizing Takagi-Sugeno fuzzy neural networks and game theory to minimize task processing latency in IoV. The scheme predicts future traffic flow, allowing roadside units to balance loads, and uses a Q-learning algorithm for optimal resource allocation, demonstrating strong performance in experiments.

SUMMARY OF REVIEW PROCESS

Table 3 outlines the limitations identified in previous research on load balancing methods, such as implementation complexity, scalability challenges, and high computational demands. Proposed solutions include optimizing algorithms, simplifying design structures, and incorporating adaptive strategies to enhance performance. Continuous improvement and the integration of advanced methodologies are crucial for addressing these limitations effectively.

Table 3. Limitations, Solutions of Metaheuristic Algorithms for Load Balancing

Reference	Proposed Method	Pros	Limitations	Proposed Solutions	Tool	Metrics
(S.Negietal., 2021)	CMODLB: Clustering-based Multi-Objective Dynamic Load Balancing	Combines unsupervised ML, ANN, and interval type-2 fuzzy logic system; improves load balancing	Difficulty in implementation and maintenance; scalability issues; high computational requirements	Optimizes scalability using distributed systems; enhance simplicity in implementation	MATLAB and Java	Make span, throughput, energy consumption
(P. K. Bal et al., 2022)	RATS-HM: Hybrid Approach	Improves make-span reduction and throughput	Complexity leading to computational overhead; scalability challenges	Simplify hybrid approach; improve computational efficiency for large-scale environments	Cloud Sim	Make span, throughput, response time
(M.I. Alghamdi et al., 2022)	BPSO: Binary Particle Swarm Optimization	Improves efficiency in load balancing and job scheduling	Struggles with high-dimensional spaces; local optima problems; does not consider energy consumption	Improve diversity in optimization; integrate energy consumption considerations	Cloud Analyst	Execution time, load balancing efficiency, energy consumption
(D. A. Shafiq et al., 2021)	DLBA: Dynamic Load Balancing Algorithm	Reduces resource usage; shortens execution time	Does not fully integrate SLA parameters	Enhance SLA integration; develop adaptive SLA management strategies	Cloud Sim	Resource usage, execution time, SLA violation

(F. Ebadifard and S.M.Babamir, 2021)	Dynamic Scheduling with CloudSim	Improved workload distribution	Communication overhead challenges	Optimize communication protocols; implement predictive, preemptive load balancing	Cloud Sim	Workload distribution efficiency, communication overhead
(P. Neelima and A. R.M.Reddy, 2020)	ADA: Adaptive Dragon fly Algorithm	Improves execution time and cost	Risk of getting trapped in local optima in complex scenarios	Refine ADA parameters; integrate real-time adaptive strategies	MATLAB	Execution time, cost efficiency, task completion
(Z. Xu etal.,2024)	Deep Learning & Reinforcement Learning Task Scheduling	Enables real-time monitoring and optimal resource use	High computational overhead	Improve model training efficiency; optimize resource allocation strategies	TensorFlow	Resource optimization, model training time

Table 4 lists the drawbacks of load balancing techniques that have been noted in earlier studies, including high computing needs, implementation complexity, and scalability issues of machine learning and reinforcement learning techniques adaptive methods.

Table 4. Limitations, Solutions of Machine Learning and Reinforcement Learning Techniques

Reference	Proposed Method	Advantages	Limitations	Proposed Solutions	Tool	Metrics
(A. Ghasemia d A. Toroghi Haghighat, 2020)	ML-based VM replacement strategy	Improves CPU, memory, and bandwidth load balance (25%, 34%, 32%)	High computational demands; significant runtime	Optimization of learning algorithms to reduce overhead	Cloud Sim	CPU utilization, memory usage, bandwidth balance
K. Karthiban and J.S.Raj, 2020)	Deep Reinforcement Learning for resource distribution	Elegant allocation schemes	Scalability issues	Distributed computing and model optimization	Tensor Flow	Resource allocation efficiency, scalability
(P.Yang et al., 2024)	Scalable policy using neural networks for load balancing	Reduces idleness; improves load balancing	Needs simulations using real data to improve accuracy	Use real-world data to improve model precision	Python (Neural Networks)	Load balancing, server idleness
(S.M.G. Kashikolaei et al., 2020)	Firefly Algorithm integrated with Imperialist Competitive Algorithm	Improvs make span, load distribution, and system stability	Increased computational needs	Hybrid algorithms with improved convergence and search optimization strategies	MATLAB	Make span, load distribution, system stability

(M. S. Al Reshan et al., 2023)	Hybrid GWO-PSO approach	Rapid convergence; better performance	Computational complexity	Parallel processing to achieve real-time performance	Cloud Sim	Converge cerate, performance efficiency
(A. R. Khan, 2024)	Hybrid Lyre bird Falcon Optimization (HLFO)	Multi-objective optimization using reinforcement learning	Complexity and implementation challenges in diverse environments	Simplify implementation strategies for diverse cloud environments	MATLAB	Optimization efficiency, load balancing
(A.Jyoti and M.Shrimali, 2020)	MADRL- D R A with Dynamic Optimal Load-Aware Service Broker	Enhance dynamic resource allocation; improves throughput	Heavy reliance on VM cost analysis	Enhanced learning models and Diverse cost Metrics for better decision-making	Java, Cloud Sim	Throughput, dynamic resource allocation
(S.Wangetal., 2020)	Improved Firework Algorithm (IFA)	Adapts todynamicnetworkconditions	Resource insufficiency due to clustering inefficiencies	Refined clustering methodologies for better resource management	Cloud Sim	Network adaptability, resource sufficiency
(E.Mathanraj and R.N.Reddy, 2024)	Principal Component Gradient Round Robin Load Balancing (PCGRLB)	Increase load balance efficiency	Complexity in managing varied workload sizes	Adaptive load balancing mechanisms for varied workloads	MATLAB	Load balance efficiency, workload management
(P.Li et al.,2024)	Meta-optimization algorithm based on natural selection principles	Improved through put and scalability	Complex design	Stream line the algorithmic design for better performance	Cloud Sim	Throughput, Scalability

The constraints of load balancing approaches, including high computing needs, scalability issues, and implementation complexity and limitations, solutions of dynamic and adaptive load balancing algorithms are listed in Table 5.

Table 5. Limitations, Solutions of Dynamic and Adaptive Load Balancing Algorithms

Reference	Proposed Method	Advantages	Limitations	Proposed Solutions	Tool	Metrics
(B. Kruekaew and W. Kimpan, 2022)	MOABCQ algorithm for make span and cost reduction	Better make span and cost reduction	Uncertainty over optimization potential	Fine-tuning optimization algorithms based on dataset characteristics	MATLAB	Make span, cost reduction

(M.Sudheer et al., 2023)	Deep Q-learning network model	Reduces make span, SLA violations, and energy consumption	High computation requirements in the initial phase	Incremental learning techniques to reduce initial computational demands	TensorFlow	Make span, SLA violations, energy consumption
(D. Tennakoon et al., 2023)	Modified double Q-learning algorithm	Reduces unsatisfied cloud consumers	High computational complexity	Optimized data structures to address complexity	Cloud Sim	Consumer satisfaction, computational complexity
(T. Renugadevi et al, 2020)	Task Distribution Algorithm (TDA) for energy-efficient workload distribution	Ensures energy efficiency in datacenters	Complexity in managing diverse workload demands	Development of adaptive mechanism for workload management	MATLAB	Energy efficiency, workload management
(C. Dutta et al., 2024)	Gallant Ant Colony Optimized Machine Learning Framework (GACO- MLF)	Enhances load balancing in IoT and PCN environments	Scalability issues in large environments	Optimization for scalability in large environments	Python (Scikit-learn)	Load balancing, scalability
(Singhal et al., 2024)	Rock Hyrax-based load balancing algorithm	Efficient load balancing	Dynamic job and VM management require refinement	Refinement in job and VM management	Cloud Sim	Load balancing efficiency, VM management
(Velliangiri et al., 2021)	Hybrid Electro Search with Genetic Algorithm (HESGA)	Reduces make span and cost	Increased latency due to long execution time soft genetic algorithm	Optimization of genetic algorithm execution times	MATLAB	Make span, cost reduction, latency
(M. G. Brahmam and V.A.R,2024)	Energy-efficient migration models with deep reinforcement learning	Improved performance metrics	Complexity in integrating advanced techniques	Further optimization of integration strategies	TensorFlow	Energy efficiency, performance metrics
(J. Ramkumar et al., 2023)	Gallant Ant Colony Optimized Machine Learning Framework (GACO- MLF)	Enhances load balancing in IoT environments	Scalability challenges in complex environments	Refinement of ant colony algorithms for larger datasets	Cloud Sim	Load balancing, scalability
(P. Neelakantan and N.S. Yadav, 2023)	Whale-Based Convolution Neural Framework (WbCNF)	Reduced job execution time	High computation times for complex tasks	Parallel processing techniques to expedite computation	Tensor Flow, Keras	Job execution time, computational efficiency

(Naiemetal., 2024)	Gaussian Naïve Bayes classifier for DDoS attack detection	Improved accuracy	Zero-frequency issue and independence assumption	Iterative features election and SMOTE for handling data imbalances	Python (Scikit-learn)	Accuracy, detection rate
(F.Ramezani Shahidani et al., 2023)	Reinforcement learning-based fog scheduling algorithm	Improves load balancing in fog computing	Challenges in real-time dynamic workload handling	Continuous learning and adaptation for evolving workloads	Cloud Sim	Load balancing, dynamic workload handling

The Table 6 offers a clear synthesis of the latest advancements in AI-driven load balancing techniques, detailing the proposed methods alongside their advantages and limitations. It serves as a helpful reference for identifying effective strategies and addressing challenges in optimizing load balancing within various computing environments.

Table 6. Limitations, Solutions of AI-Driven Load Balancing Techniques

Reference	Proposed Method	Advantages	Limitations	Proposed Solutions	Tool	Metrics
(P. Yang et al., 2023)	Scalable policy based on neural networks	Achieved over 130% less server idleness compared to traditional methods	Limited to specific applications in financial services	Explore adaptation to other domains	Tensor Flow	Server idleness reduction
(Jiang et al., 2024)	FedMP (Federated Learning framework)	Improved training performance and reduced resource consumption	Dependency on model accuracy and data quality	Implement data validation and pre-processing methods	PySyft	Training performance, resource consumption
(K.Rajakumari et al., 2021)	FedNAS (Federated Neural Architecture Search)	Up to 200% cost savings on the client side while maintaining model accuracy	Complexity in automating model generation	Develop simplified frameworks for automation	TensorFlow	Cost savings, model accuracy
(K. Rajakumari et al., 2021)	Dynamic Weighted Round-Robin (DWRR)	Optimizes resource allocation based on task priorities and lengths	May not handle dynamic workloads effectively	Incorporate real-time monitoring for adaptive scheduling	Cloud Sim	Resource allocation efficiency
(M. Núñez-Merino et al., 2024)	Quantum–inspired computing model	Enhances capabilities for improved manufacturing and logistics	Requires advanced technology and expertise	Focus on training and development in quantum computing	Quantum Development Kit (QDK)	Improvement in manufacturing / Logistics
(Baburao et al., 2024)	EDRAM(Enhanced Dynamic Resource Allocation Model)	Improves Quality of Experience (QoE) and reduces latency	Limited to specific optimization techniques	Explore integration with other optimization algorithms	MATLAB	Quality of Experience (QoE), latency
(S. Aslam et al., 2021	Deep learning methods for forecasting	Enhances efficiency in renewable energy generation and load forecasting	Relies heavily on the volume and quality of historic data	Implement strategies for data enrichment and analysis	TensorFlow	Forecast accuracy, efficiency

Table 7 provides a concise overview of various hybrid algorithms used for load balancing in cloud and edge computing, highlighting their unique advantages, limitations, and proposed solutions. It serves as a valuable resource for understanding the effectiveness and applicability of different hybrid optimization techniques in improving resource allocation and performance in these environments.

Table 7. Limitations, Solutions of Hybrid Algorithms for Cloud and Edge Load Balancing

Reference	Proposed Method	Advantages	Limitations	Proposed Solutions	Tool	Metrics
(H.Wang and B. Alidaee, 2023)	Hybrid-Heuristic Framework	Achieves new best solutions for 61 out of 70 instances.	May not be scalable for larger problem sizes.	Explore scalability in future research.	Custom Framework	Solution quality, instance count
(Devaraj et al., 2020)	FIMPSO Algorithm	Significant improvements in response time (13.58ms) and CPU utilization (98%).	Limited generalization across different workloads.	Test on diverse cloud environments.	MATLAB	Response time, CPU utilization
(R. Etengu et al., 2020)	Hybrid SDN/OSPF Framework	Reduces global network energy consumption while maintaining quality of service.	Technical and organizational hurdles remain.	Address organizational challenges in implementation.	SDN Controllers	Energy consumption, quality of service
(X.Wei, 2020)	Improved Ant Colony Optimization	Enhanced performance metrics with faster convergence	May struggle with dynamic task environments.	Integrate adaptive mechanisms for dynamic loads.	Custom Simulation Environment	Convergent speed, performance metrics
(K. Ramyaand S. Ayothi, 2023)	HDWOA-LBM	Significant improvements in throughput and reliability.	May require fine-tuning of parameters for optimality.	Implement adaptive parameter tuning methods.	MATLAB	Throughput, reliability
(N. Rana et al., 2021)	M-WODE Technique	Out performs existing algorithms in make span and cost trade-offs.	Complexity in implementation compared to traditional methods.	Simplify implementation while retaining performance.	Custom Framework	Make span, cost trade-offs
(S. V. Nethaji and M. Chidambaram, 2022)	DGW-SBDR Algorithm	Optimizes job allocation, improving quality of service.	Energy reduction may affect performance in high-load scenarios.	Balance energy efficiency with performance needs.	Cloud Sim	Job allocation efficiency, quality of service

Table 8. Limitations, Solutions of Dynamic Load Balancing in Edge Computing

Reference	Proposed Method	Advantages	Limitations	Proposed Solutions	Tool	Metrics
(F.M. Talaat et al., 2020)	Load Balancing and Optimization Strategy (LBOS)	Achieves 85.71% load balancing level, enhancing service quality.	Limited to specific IoT healthcare scenarios.	Explore broader application in different domains.	Custom Simulation Environment	Load balancing level, service quality
(M. Kaur and R. Aron, 2022)	PSW-Fog Clustering Algorithm	Reduces latency and energy consumption in scientific workflows.	Performance may vary with work load types.	Test across various scientific workflows.	Custom Framework	Latency reduction, energy consumption
(Z. Nezami et al., 2021)	EPOS Fog	Achieves 25% reduction in service execution delay and 0% improvement in load balance.	Dependence on decentralized multi-agent coordination may complicate implementation.	Simplify coordination mechanisms for easier deployment.	Multi-Agent Simulation	Service Execution delay, load balance improvement
(L.Wang et al., 2020)	UAV-aided Mobile Edge Computing	Improves geographical fairness and energy efficiency.	Complexity in managing multiple UAV trajectories.	Develop simpler trajectory management techniques.	UAV Simulation Toolkit	Geographical fairness, energy efficiency
(S. Wan et al., 2020)	COV Method for Computation Off loading	Effectively minimizes off loading delays and costs.	May struggle with highly dynamic environments.	Implement adaptive algorithms for dynamic load conditions.	Cloud Sim	Off loading delay, cost reduction
(B.Mahato et al., 2021)	Biogeography-based Optimization (BBO)	Enhances resource utilization and reduces delays and costs.	Complexity in cooperative bandwidth sharing.	Explores simplified models for bandwidth management.	Custom Optimization Framework	Resource utilization, delay and cost reduction
(X. Xu et al., 2022)	F-TORA Task Off loading Scheme	Minimizes task processing latency using predictive algorithms	Dependence on accurate traffic flow predictions.	Incorporate real-time traffic monitoring for better predictions.	Simulation Environment	Task processing latency

Table 8 shows the dynamic load balancing in edge computing faces several limitations, such as varying performance across different workloads and complexities in managing multiple agents or UAV trajectories. These challenges can lead to inefficient resource utilization and increased delays.

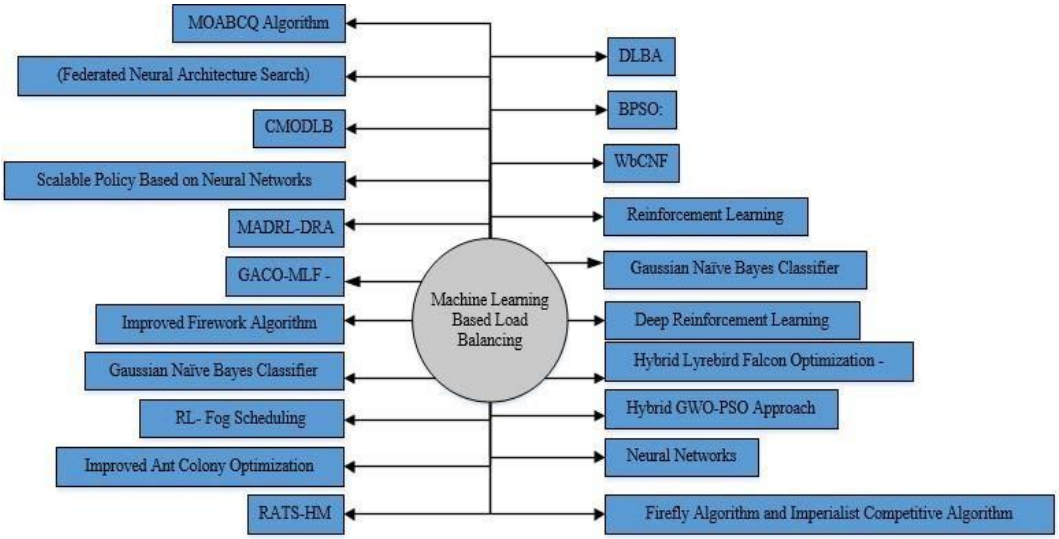

Figure4. Examples of Machine Learning-Based Approaches in Cloud Computing

Figure 4 shows the examples of machine learning-based approaches in cloud computing illustrates arrange of innovative techniques that leverage machine learning to enhance cloud computing efficiency. It highlights the effectiveness of methods like deep Q-learning and federated learning in optimizing resource allocation, load balancing, and energy efficiency.

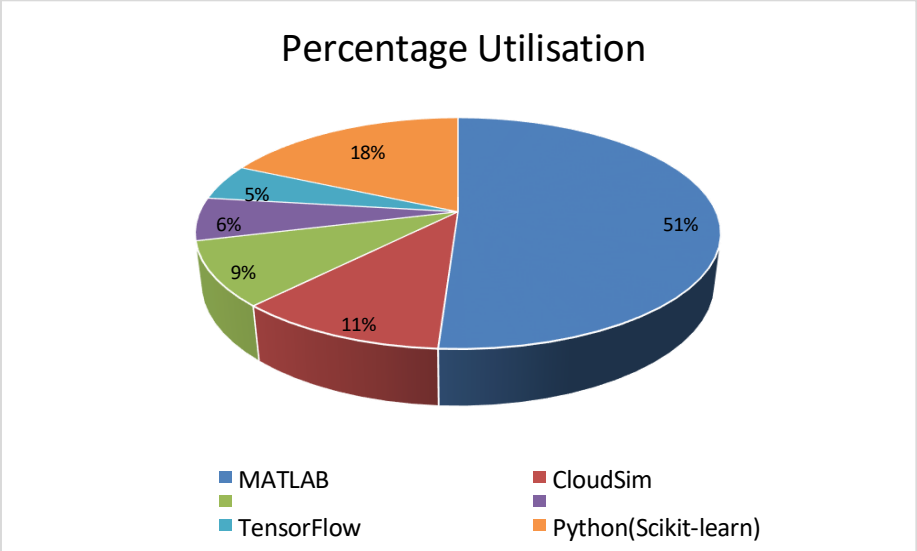

Figure 5. Percentage Utilization of Simulation Tools for Cloud Computing Load Balancing

The Figure 5 shows the percentage utilization of various simulation tools for cloud computing load balancing from 2020 to 2024. It highlights which tools were most commonly used, indicating trends and preferences in simulation tool adoption during this period.

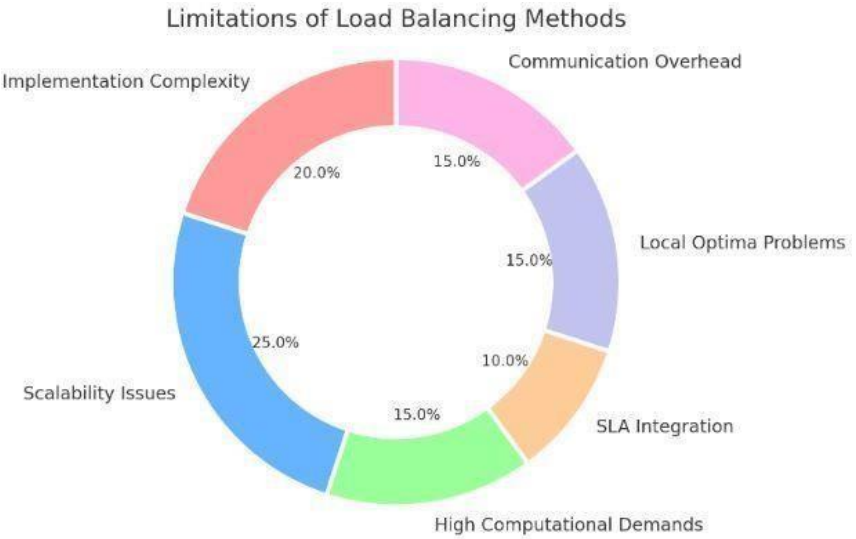

Figure 6. Limitations of Load Balancing Methods

Figure 6 visualizing the limitations of load balancing methods. It highlights the following key areas: Implementation Complexity (20%), Scalability Issues (25%), High Computational Demands (15%), SLA Integration (10%), Local Optima Problems (15%), and Communication Overhead (15%). The visual representation can help provide a quick understanding of the primary challenges faced in load balancing within cloud computing environments.

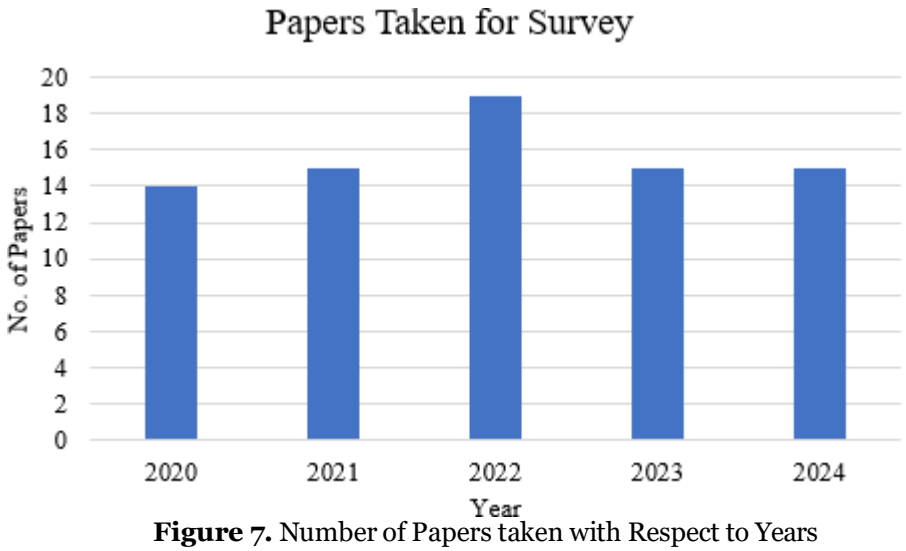

Figure 7. Number of Papers taken with Respect to Years

Figure 7 displays the distribution of research papers taken with reference to the years, 2020 to 2024. There is annual distribution with one paper per year but, there is a significant spike in 2020 and 2024 which has 8 papers each implying that the topic was more focused on in these years or more developments on this topic were registered in those years. It is evident that the trends exist in research output to have an irregular oscillation over the five-year period although a slight inclination towards arise in the beginning and also towards the later periods are observable. This tendency may indicate concerns shift and research priorities within the community, potential new problems, new technologies, or changes in the policies affecting the area/discipline under investigation.

RESEARCH GAP

Indeed, a number of research issues still exist in the field of load balancing techniques in cloud computing even though various improvements have been made concerning these methods. Still, one can observe that few research

works have been dedicated to adaptive load balancing schemes, capable of functioning in the context of dynamic and constantly evolving cloud environments. Modern process models might involve static or semi-dynamic approaches used in organizations that in some cases do not match real life, in particular, they may result in inefficiencies when a huge influx of work arrives, or when workloads are unpredictable (Devaraj et al., 2020) Moreover, load balancing has recently seen improvements in enhancing it by using machine learning algorithms and deep reinforcement learning; however, there is comparatively little research on a single, unified system of load balancing that involves the use of deep reinforcement learning and other advanced AI techniques. The seamy potentially result in more effective and versatile solutions but literature up to now has not given them much attention. Moreover, most scientific studies are dedicated to enhancing particular criteria like make span or CPU time whereas a comprehensive solution that will allow enhancing various characteristics of cloud performance at the same time has not been worked out (Jena et al., 2022).

The second significant gap in the current literature is the lack of examination of practical solutions and the prospects for the large-scale application of suggested load-balancing strategies. It is shown that majority of the works presents theoretical and simulated designs that have little relevance to realistic experimentation. This disconnection between theory-based academia and real-application oriented execution presents a problem in implementing these techniques in large scale heterogeneous cloud frameworks. Furthermore, only a limited number of studies focus on evaluating the security principles concerning load balancing based on the machine learning algorithm (T. Renuga devi et al., 2020). When such techniques become enhanced in the market, these same techniques become vulnerable to hackers and cyber criminals, thus calling for the establishment of sound and secure algorithms. Finally, the availability of presented initiatives is still an issue. Most questionnaires and analysis methods are developed as patented instruments and mathematical models, which are difficult to reproduce and share with other researchers. These areas could be filled with efficient load balancing solutions through collaborative and open-source projects and interdisciplinary research.

COMPARATIVE ANALYSES OF THE REVIEWED PAPERS

The proposed methods for load balancing in cloud computing include various innovative approaches. CMODLB (Clustering-based Multi-Objective Dynamic Load Balancing) (S. Negi et al., 2021) focuses on clustering and dynamic load balancing to achieve faster task completion and better resource utilization. MOABCQ (Multi-Objective Artificial Bee Colony with Reinforcement Learning) (B. Kruekaew and W. Kimpan, 2022) combines artificial bee colony optimization with reinforcement learning, targeting enhanced make span and cost reduction. RATS-HM (Resource Allocation and Task Scheduling in Hybrid Machines) (P. K. Bal et al., 2022) integrates hybrid approaches to improve make span and throughput. The ML-based VM replacement method leverages machine learning techniques to optimize virtual machine replacement, enhancing multiple load balancing metrics. Additionally, Deep Reinforcement Learning Approaches (C. Dutta et al., 2024), (Z. Xu et al., 2024) employ deep reinforcement learning for task scheduling and load balancing, with an emphasis on efficient resource utilization and real-time optimization. These diverse methodologies collectively address various challenges in cloud computing load balancing, offering improvements across key performance metrics. The summary of parameters of Quality are presented below in Table 9.

- RT (Response Time): Time taken for a system to respond to a request.
- TP (Task Processing): Efficiency and speed in handling and completing tasks.
- TS (Task Scheduling): Effectiveness in organizing and prioritizing tasks.
- MS (Make span): Total time required to complete a set of tasks or operations.
- EC (Energy Consumption): Amount of energy used by the system during operations.
- SC (Scalability): System's ability to handle increased load or expand without performance loss.
- SO (Scalability Optimization): Techniques used to enhance the system’s scalability.
- IM (Implementation Complexity): Difficulty and resources needed to implement the system or algorithm.
- CO (Computational Overhead): Extra computational resources required beyond the basic operations.
- RL (Resource Load): Distribution and efficiency of resource usage across the system.
- ST (System Throughput): Rate at which the system processes and completes tasks.

Table 9. Load balancing Parameters Summary of Recent Literatures

Algorithms	RT	TP	TS	MS	EC	SC	SO	IM	CO	RL	ST
CMODLB (S. Negi et al., 2021)	✓	X	✓	X	X	✓	✓	✓	✓	✓	X
MOABCQ (B. Kruekaew and W.Kimpan, 2022)	✓	X	✓	X	✓	✓	X	✓	✓	✓	X
RATS-HM (S.S. Mangalampalli et al.,2024)	✓	X	✓	X	✓	X	✓	✓	✓	✓	X
ML-basedVM Replacement (E.Guresetal., 2022)	✓	X	✓	X	✓	✓	✓	X	✓	✓	X
LB Algorithm (M. Alam et al.,2017)	✓	X	✓	X	✓	✓	✓	✓	✓	X	X
Dynamic Scheduling (Ebadifard et al., 2024)	✓	X	X	X	✓	✓	✓	✓	✓	✓	X
BPSO(M.I. Alghamdi et al.,2022)	✓	X	✓	X	✓	✓	✓	✓	X	✓	X
DeepRL(Q.Liu et al.,2022)	✓	X	✓	X	✓	X	✓	✓	✓	X	X
ADA(Mishra et al., 2020)	✓	X	✓	X	✓	✓	X	✓	✓	✓	X
Hybrid FA & ICA (M.S.AlReshan eta l.,2023)	✓	X	X	X	✓	✓	X	✓	✓	✓	X
Combined GWO-PSO(B.Alankar et al.,2020)	✓	X	✓	X	✓	✓	X	✓	✓	X	X
HLFO with RL(J. G.Muchori and P. M.Mwangi, 2022)	✓	X	✓	X	✓	✓	X	✓	✓	✓	X
MADRL-DRA (Monalisa Kushwaha, 2024)	✓	X	✓	X	✓	✓	✓	X	✓	X	X
Improved Firework Algorithm(D.Ding et al.,2020)	✓	X	✓	X	✓	✓	✓	X	✓	✓	X
Modified DoubleQ-learning (Shafiq et al., 2022)	✓	X	✓	X	✓	✓	✓	✓	X	✓	X
TDA(Mishraetal.,2020)	✓	X	✓	X	✓	✓	✓	✓	✓	✓	X
Dijkstra'sDAG Scheduling (Shafiq et al., 2022)	✓	X	✓	X	✓	✓	X	✓	✓	✓	X
HESGA(D.Ding et al., 2020)	✓	X	✓	X	✓	✓	✓	X	✓	✓	X
Deep Q-Learning (K.Sekaran and P. V.Krishna, 2022)	✓	X	✓	X	✓	✓	✓	X	✓	X	X
PCGRLB (R. Begam et al., 2020)	✓	X	✓	X		✓	X	✓	✓	X	X

Meta-Optimization Algorithm (W.-Z. Zhang et al.,2021)	✓	X	✓	X	✓	✓	X	✓	✓	X	X
Deep Learning Modified RL	✓	X	✓	X	✓	✓	X	✓	✓	X	X
Rock Hyrax-Based Load Balancing (Z.Xu et al. , 2024)	✓	X	✓	X	✓	✓	X	✓	✓	X	X
Enhanced Gaussian Naïve Bayes(M. A. Shahid et al., 2020)	✓	X	✓	X	✓	✓	X	✓	✓	✓	X
RL Fog Scheduling (Razaq et al.,2024)	✓	X	✓	X	✓	✓	✓	X	✓	X	X
GACO-MLF(J. Ramkumar et al.,2024)	✓	X	✓	X	✓	✓	X	✓	✓	X	X
WbCNF (P. Neelakantan and N. S.Yadav, 2023)	✓	X	✓	X	✓	✓	X	✓	✓	X	X

Likely illustrates key quality parameters for load balancers, such as throughput, latency, and resource utilization. Throughput measures the volume of processed requests, latency tracks response time, and resource utilization assesses efficiency in using available resources. Optimizing these parameters ensures effective and efficient load balancing in cloud computing systems.

APPLICATIONS OF MACHINE LEARNING-DRIVEN LOAD BALANCING IN CLOUD COMPUTING

In the context of cloud computing over the last decade. Many authors suggest different machine learning based load balancing algorithms to improve resource allocation. These techniques fall within the sphere of issues like dynamic workload allocation and analogous real time resource management and also bringing about minimization of latencies and response time. Deep Learning, Reinforcement Learning, Clustering Etc., have become incorporated. in the load balancing frameworks to predict the patterns of load, have adaptive methods of resource allocation, and optimize the load distribution. These are some of the advancements that have boosted the effectiveness and efficiency of cloud services. Based on the analysis of this paper, the following table presents the main findings of machine learning-driven load balancing in cloud computing applicability. Table 10 shows the ML applications in cloud computing is given below.

Table 10. Applications of Load Balancing in Cloud Computing

Application	Description
Dynamic Resource Allocation	Machine learning algorithms can predict work load patterns and dynamically allocate resources, ensuring optimal performance and efficient resource utilization. This helps in maintaining balanced workloads across servers and minimizing response times (G.Jia et al., 2023).
Real-time LoadManagement	Reinforcement learning models can optimize task distribution in real-time, adapting to changes in workload patterns and user demands. This enhances the scalability and robustness of cloud services, allowing providers to handle peak loads effectively (S. Bharany etal.,et al.,2022).
Latency Reduction	Unsupervised learning techniques, such as clustering, can group similar tasks together, optimizing resource usage and reducing computational overhead. This leads to reduced latency and improved user experience (Razaq et al., 2024).
Cost Optimization	Machine learning-driven load balancing can lead to significant cost savings by improving resource utilization and reducing the need for over-provisioning. By predicting future resource demands, cloud providers can allocate resources more efficiently, reducing operational costs (H.Eljak et al., 2024).

Enhanced Security	Integrating machine learning algorithms with security mechanisms can help identify and mitigate potential threats, such as DDoS attacks. Anomaly detection algorithms can recognize suspicious patterns of behavior, allowing for dynamic adjustment of load balancing strategies to prevent security breaches (Z.Xu et al.,2024).

CHALLENGES

In the area of cloud computing, various load balancing approaches based on machine learning have been presented to counter the fluctuating workloads. But there are some issues to face in right application of these techniques. The first major difficulty is the inability to predict the workload due to the variability of users' requests and their randomness (J. Ramkumar et al., 2023). The use of machine learning in the cloud necessitates that a large history of data be fed into the algorithm and the variation in cloud setting restricts data consistency. However, real-time updating of the resource demands and allocation through machine learning of algorithms requires a considerable amount of computation that has to be performed in real-time, which a could slow down cloud services. The combination of machine learning models with other cloud architectures is also other issues that are unique to this model, commonality and scalability being some of them. One important consideration is to ensure that these models can function in existence with the current systems without any hitches or changes to the extant state.

Also, the process of serving large numbers of active users imposes a significant issue of achieving higher performance on one hand when on the other, the costs for doing so are high (C. Dutta et al., 2024). The load balancing that incorporates the use of machine learning seeks to improve the efficiency or utilization of resources and decrease operational costs in the organization yet the deployment and support for advanced machine learning algorithms is expensive (F. Ramezani Shahidani et al., 2023). Strengthening the operational nature is that the workload patterns also have to be periodically monitored and updated, and the need for incorporating these models to reflect change increases its complexity (P. Li et al., 2024) . Also, machine learning approaches in optimization of resource allocation must also meet certain criteria of security and privacy of the user's data. It still remains a challenge to implement security solutions, like machine learning models, together with KPIs to determine malicious traffic, like DDoS attacks, without considerably affecting system performance. The steady and economic building of powerful and durable machine learning models, which meet with these opposite requirements: high performances, low costs, and security, remains an open problem within the field of cloud computing load balancing.

METRICS AND TOOLS

Implementation and Analytical Tools

To implement and evaluate machine learning algorithms for load balancing, the following tools can be utilized.

Machine Learning Tools

- TensorFlow: An open-source platform for developing and deploying models, particularly useful for neural networks and deep learning.
- PyTorch: A flexible machine learning library for research and production, known for its dynamic computation capabilities.
- Scikit-learn: A Python library offering simple and efficient tools for data mining and analysis, including clustering and classification algorithms.

Load Balancing Tools

H A Proxy: Open-source software that provides high availability, load balancing, and proxy services, ensuring load distribution and high availability.
Nginx: A web server that also functions as a reverse proxy, load balancer, and HTTP cache, widely used for load balancing and web serving.
AWS Elastic Load Balancing (ELB): A managed service by Amazon Web Services that distributes incoming application traffic across multiple targets in the cloud.

Analytical Tools

- Prometheus: An open-source monitoring and alerting toolkit, often used for tracking the performance of load balancing systems.
- Grafana: An open-source analytics and monitoring platform that integrates with Prometheus for visualizing metrics and performance data.
- JMeter: An open-source tool for performance testing that can simulate load and measure the response of load balancing systems.

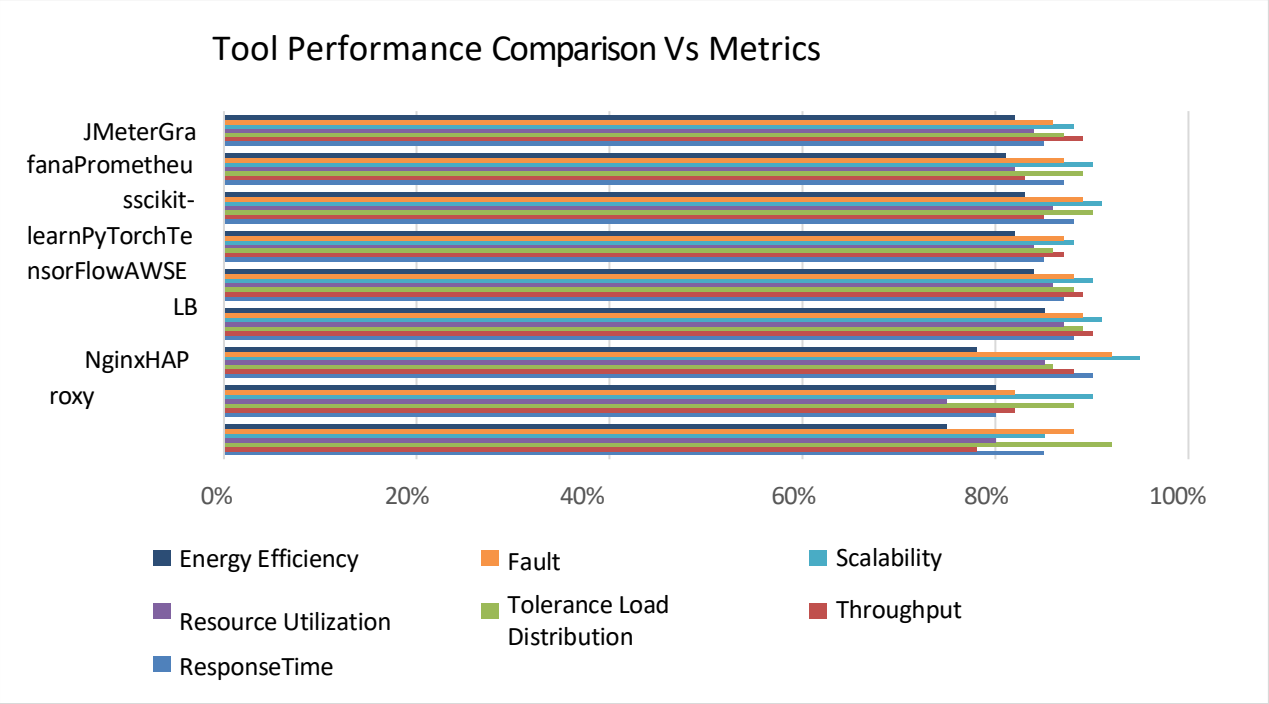

Figure 8. Tool performance comparison vs metrics

Figure 9 visually compares the effectiveness of various tools across key cloud performance metrics such as response time, throughput, and scalability etc. By analyzing tools, it highlights the strengths and weaknesses of each in optimizing cloud environments. This comparison aids in identifying the most efficient solutions for enhancing cloud computing performance.

CONCLUSION AND FUTURE WORKS

In the present-day context of cloud environments that are rapidly growing, the efficient means of load balancing is critical to the utilization of available resources and quality of services that are achieved throughout the clouds. In this research, an attempt has been made to evaluate the use of ML in improving load management approaches in cloud environments. Thus, with the help of ML, various approaches to load balancing can be enhanced by using predictive analysis, the automation of algorithms, and optimization of decision making. Machine learning gives a strong backing in the statistical analysis of past data and in making concise estimations and predictions of the work load and the need for resources at any given time. This capability makes a way for better load balancing techniques that are real time based and are sensitive to conditions that can cause latency and low efficiency of the system. The reviewed study clarified several ML techniques such as supervised, unsupervised, and reinforcement learning with the distinct opportunity for load balancing. The study proves that by integrating various load balancing methodologies with the help of ML, performance indicators like response time, throughput, and resource usage can be boosted considerably. These techniques also appear to have the possibility of managing operational costs in a manner that do not over-provide the resources in the organization. In addition, load balancing with the help of ML algorithms can improve the cloud infrastructure to be more adaptive to the different system loads and fail or experience the load increases more quickly than necessary.

The future application of machine learning in load balancing in cloud computing is a large unexplored frontier with much potentiality. Many important fields reveal expending potential for research and development endeavors. It

is another area for further investigation and discovery applicable for creating more complex models that take more factors into the consideration and use advances deep learning for more precise estimations of resource utilization and workload. Taking into account the adaptively of ML the future load balancing systems can be developed as self-learning and self-adaptive. Precisely, the reinforcement learning seems to offer the best chance to design systems, which are capable of adapting their decision-making strategy to achieve maximal performance in a stochastic environment. Thus, with sustainability gaining priority more and more nowadays, integrating energy efficiency in load balancing is a necessity. Predictive analytics can also be made to model ways by which a system could be designed to achieve an optimal level of performance while at the same time operating at an optimal level of power consumption, and this would be made on the basis of power consumption patterns and possible futures. Also, extending the presented techniques in the field of security-aware ML-based load balancing as well as the application of similar ideas to new areas like edge computing, IoT, and 5G networks is a promising direction. Further work can also be made with the exploration of generous and federated studying techniques to ensure the load balance across several data center and various edge nodes which allow sharing of model and insights without necessitating to share the data. Thus, along with considering the above-discussed future research directions, the field will be able to expand and only enhance the demands of the modern cloud environment to create more robust, scalable, and efficient cloud computing structures

REFRENCES

[1] Swetha, G. Ahmad, M.O. ―A Comprehensive Review of Load Balancing Techniques for Cloud Performance‖ , 15th International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2024, 2024, 2, pp. 5417–5424

[2] R. Begam, W. Wang, and D. Zhu, ―TIMER-Cloud: Time-Sensitive VM Provisioning in Resource-Constrained Clouds, ‖ IEEE Trans. Cloud Computing, vol. 8, no. 1, pp. 297–311, Jan. 2020, doi: 10.1109/TCC.2017.2777992.

[3] M. A. Shahid, N. Islam, M. M. Alam, M. M. Su'ud, and S. Musa, ―A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach, ‖ IEEE Access, vol. 8, pp. 130500–130526, 2020, doi: 10.1109/ACCESS.2020.3009184.

[4] O. Cheikhrouhou, K. Mershad, F. Jamil, R. Mahmud, A. Koubaa, and S. R. Moosavi, ―A Lightweight Blockchain and Fog-enabled Secure Remote Patient Monitoring System, ‖ Internet of Things, vol. 22, p. 100691, Jul. 2023, doi: 10.1016/j.iot.2023.100691.

[5] P. Varshney and Y. Simmhan, ―Characterizing Application Scheduling on Edge, Fog and Cloud Computing Resources, ‖ Softw Pract Exp, vol. 50, no. 5, pp. 558–595, May 2020, doi: 10.1002/spe.2699.

[6] S. S. Sefati, M. Mousavinasab, and R. Farkhady, ―Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation, ‖ The Journal of Supercomputing, vol. 78, Jan. 2022, doi: 10.1007/s11227-021-03810-8.

[7] W.-Z. Zhang et al., ―Secure and Optimized Load Balancing for Multitier IoT and Edge-Cloud Computing Systems, ‖ IEEE Internet of Things Journal, vol. 8, no. 10, pp. 8119–8132, May 2021, doi: 10.1109/JIOT.2020.3042433.

[8] Shafiq et al., ―Load balancing techniques in cloud computing environment: A review, ‖ Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 7, pp. 3910–3933, Jul. 2022, doi: 10.1016/j.jksuci.2021.02.007.

[9] D. Ding, X. Fan, Y. Zhao, K. Kang, Q. Yin, and J. Zeng, ―Q-learning based dynamic task scheduling for energy-efficient cloud computing, ‖ Future Generation Computer Systems, vol. 108, pp. 361–371, Jul. 2020, doi: 10.1016/j.future.2020.02.018.

[10] Peng et al., ―Joint optimization for time consumption and energy consumption of multi-application and load balancing of cloudlets in mobile edge computing, ‖ 2019.

[11] M. O Ahmad and R. Z. Khan, ―An efficient load balancing scheduling strategy for cloud computing based on hybrid approach‖ , International Journal of Cloud Computing, Vol. 9, Issue 4, 2020. Inder Science Publication.

[12] M. Yadav and S. Gupta, ―Hybrid meta-heuristic VM load balancing optimization approach, ‖ Journal of Information and Optimization Sciences, vol. 41, no. 2, pp. 577–586, Feb. 2020, doi: 10.1080/02522667.2020.1733190.

[13] C. S. M. Babou et al., ―Hierarchical Load Balancing and Clustering Technique for Home Edge Computing, ‖ IEEE Access, vol. 8, pp. 127593–127607, 2020, doi: 10.1109/ACCESS.2020.3007944.

[14] Jena et al., ―Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment, ‖ Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 6, pp. 2332–2342, Jun. 2022, doi: 10.1016/j.jksuci.2020.01.012.

[15] S. Negi, M. M. S. Rauthan, K. S. Vaisla, and N. Panwar, ―CMODLB: an efficient load balancing approach in cloud computing environment, ‖ J Supercomputing, vol. 77, no. 8, pp. 8787–8839, Aug. 2021, doi: 10.1007/s11227-020-03601-7.

[16] 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, vol. 22, no. 3, Art. no. 3, Jan. 2022, doi: 10.3390/s22031242.

[17] M. I. Alghamdi, ―Optimization of Load Balancing and Task Scheduling in Cloud Computing Environments Using Artificial Neural Networks-Based Binary Particle Swarm Optimization (BPSO),‖ Sustainability, vol. 14, no. 19, Art. no. 19, Jan. 2022, doi: 10.3390/su141911982.

[18] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, ―A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications, ‖ IEEE Access, vol. 9, pp. 41731–41744, 2021, doi: 10.1109/ACCESS.2021.3065308.

[19] F. Ebadifard and S. M. Babamir, ―Autonomic task scheduling algorithm for dynamic workloads through a load balancing technique for the cloud-computing environment, ‖ Cluster Comput, vol. 24, no. 2, pp. 1075–1101, Jun. 2021, doi: 10.1007/s10586-020-03177-0.

[20] P. Neelima and A. R. M. Reddy, ―An efficient load balancing system using adaptive dragonfly algorithm in cloud computing, ‖ Cluster Comput, vol. 23, no. 4, pp. 2891–2899, Dec. 2020, doi: 10.1007/s10586-020-03054-w.

[21] Z. Xu, Y. Gong, Y. Zhou, and W. Qian, ―Enhancing Kubernetes Automated Scheduling with Deep Learning and Reinforcement Techniques for Large-Scale Cloud Computing Optimization, ‖ 2024.

[22] A. Ghasemi and A. Toroghi Haghighat, ―A multi-objective load balancing algorithm for virtual machine placement in cloud data centers based on machine learning, ‖ Computing, vol. 102, no. 9, pp. 2049– 2072, Sep. 2020, doi: 10.1007/s00607-020-00813-w.

[23] K. Karthiban and J. S. Raj, ―An efficient green computing fair resource allocation in cloud computing using modified deep reinforcement learning algorithm, ‖ Soft Comput, vol. 24, no. 19, pp. 14933– 14942, Oct. 2020, doi: 10.1007/s00500-020-04846-3.

[24] P. Yang, L. Zhang, H. Liu, and G. Li, ―Reducing Idleness in Financial Cloud Services via Multi- objective Evolutionary Reinforcement Learning based Load Balancer, ‖ Sci. China Inf. Sci., vol. 67, no. 2, p. 120102, Feb. 2024, doi: 10.1007/s11432-023-3895-3.

[25] S. M. G. Kashikolaei, A. A. R. Hosseinabadi, B. Saemi, M. B. Shareh, A. K. Sangaiah, and G.-B. Bian, ―An enhancement of task scheduling in cloud computing based on imperialist competitive algorithm and firefly algorithm, ‖ J Supercomput, vol. 76, no. 8, pp. 6302–6329, Aug. 2020, doi: 10.1007/s11227- 019-02816-7.

[26] M. S. Al Reshan et al., ―A Fast Converging and Globally Optimized Approach for Load Balancing in Cloud Computing, ‖ IEEE Access, vol. 11, pp. 11390–11404, 2023, doi: 10.1109/ACCESS.2023.3241279.

[27] A. R. Khan, ―Dynamic Load Balancing in Cloud Computing: Optimized RL-Based Clustering with Multi-Objective Optimized Task Scheduling, ‖ Processes, vol. 12, no. 3, Art. no. 3, Mar. 2024, doi: 10.3390/pr12030519.

[28] A. Jyoti and M. Shrimali, ―Dynamic provisioning of resources based on load balancing and service broker policy in cloud computing, ‖ Cluster Computing, vol. 23, no. 1, pp. 377–395, Mar. 2020, doi: 10.1007/s10586- 019-02928-y.

[29] S. Wang, T. Zhao, and S. Pang, ―Task Scheduling Algorithm Based on Improved Firework Algorithm in Fog Computing, ‖ IEEE Access, vol. 8, pp. 32385–32394, 2020, doi: 10.1109/ACCESS.2020.2973758.

[30] E. Mathanraj and R. N. Reddy, ―Enhanced principal component gradient round-robin load balancing in cloud computing, ‖ The Scientific Temper, vol. 15, no. 01, pp. 1806–1815, Mar. 2024, doi: 10.58414/SCIENTIFICTEMPER.2024.15.1.32.

[31] P. Li, H. Wang, G. Tian, and Z. Fan, ―Towards Sustainable Cloud Computing: Load Balancing with Nature-Inspired Meta-Heuristic Algorithms, ‖ Electronics, vol. 13, no. 13, p. 2578, Jun. 2024, doi: 10.3390/electronics13132578.

[32] B. Kruekaew and W. Kimpan, ―Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm with Reinforcement Learning, ‖ IEEE Access, vol. 10, pp. 17803–17818, 2022, doi: 10.1109/ACCESS.2022.3149955.

[33] M. Sudheer, K. Reddy, M. KUMAR, O. Khalaf, C. Romero, and G. Sahib, ―DRLBTSA: Deep reinforcement learning based task-scheduling algorithm in cloud computing, ‖ Multimedia Tools and Applications, vol. 83,
pp. 1–29, Jun. 2023, doi: 10.1007/s11042-023-16008-2.

[34] D. Tennakoon, M. Chowdhury, and T. H. Luan, ―Cloud-based load balancing using double Q-learning for improved Quality of Service, ‖ Wireless Networks, vol. 29, no. 3, pp. 1043–1050, Apr. 2023, doi: 10.1007/s11276-018-1888-8.

[35] T. Renugadevi, K. Geetha, K. Muthukumar, and Z. W. Geem, ―Energy-Efficient Resource Provisioning Using Adaptive Harmony Search Algorithm for Compute-Intensive Workloads with Load Balancing in Data centers, ‖ Applied Sciences, vol. 10, no. 7, p. 2323, Mar. 2020, doi: 10.3390/app10072323.

[36] C. Dutta, R. M. Rani, A. Jain, I. Poonguzhali, D. Salunke, and R. Patel, ―Deep Learning Modified Reinforcement Learning with Virtual Machine Consolidation for Energy-Efficient Resource Allocation in Cloud Computing, ‖ International Journal of Cooperative Information Systems, Mar. 2024, doi: 10.1142/S0218843024500059.

[37] Singhal et al., ―Energy Efficient Load Balancing Algorithm for Cloud Computing Using Rock Hyrax Optimization. ‖ Accessed: Jul. 06, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10477415?denied=

[38] Velliangiri et al., ―Hybrid electro search with genetic algorithm for task scheduling in cloud computing, ‖ Ain Shams Engineering Journal, vol. 12, no. 1, pp. 631–639, Mar. 2021, doi: 10.1016/j.asej.2020.07.003.

[39] M. G. Brahmam and V. A. R, ―VMMISD: An Efficient Load Balancing Model for Virtual Machine Migrations via Fused Metaheuristics with Iterative Security Measures and Deep Learning Optimizations, ‖ IEEE Access, vol. 12, pp. 39351–39374, 2024, doi: 10.1109/ACCESS.2024.3373465.

[40] J. Ramkumar, R. Vadivel, B. Narasimhan, S. Boopalan, and B. Surendren, ―Gallant Ant Colony Optimized Machine Learning Framework (GACO-MLF) for Quality-of-Service Enhancement in Internet of Things-Based Public Cloud Networking, ‖ in Data Science and Communication, Springer, Singapore, 2023, pp. 425–438. doi: 10.1007/978-981-99-5435-3_30.

[41] P. Neelakantan and N. S. Yadav, ―Proficient job scheduling in cloud computation using an optimized machine learning strategy, ‖ Int. j. inf. technology, vol. 15, no. 5, pp. 2409–2421, Jun. 2023, doi: 10.1007/s41870-023-01278-8.

[42] Naiem et al., ―Enhancing the Efficiency of Gaussian Naïve Bayes Machine Learning Classifier in the Detection of DDOS in Cloud Computing. ‖ Accessed: Jul. 06, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10302279?denied=

[43] F. Ramezani Shahidani, A. Ghasemi, A. Toroghi Haghighat, and A. Keshavarzi, ―Task scheduling in edge-fog-cloud architecture: a multi-objective load balancing approach using reinforcement learning algorithm, ‖ Computing, vol. 105, no. 6, pp. 1337–1359, Jun. 2023, doi: 10.1007/s00607-022-01147-5.

[44] P. Yang, L. Zhang, H. Liu, and G. Li, ―Reducing Idleness in Financial Cloud Services via Multi- objective Evolutionary Reinforcement Learning based Load Balancer, ‖ Nov. 23, 2023, arXiv: arXiv:2305.03463.
Accessed: Oct. 15, 2024. [Online]. Available: http://arxiv.org/abs/2305.03463

[45] Jiang et al., ―FedMP: Federated Learning through Adaptive Model Pruning in Heterogeneous Edge Computing. ‖ Accessed: Oct. 15, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9835327

[46] Yuan et al., ―Resource-Aware Federated Neural Architecture Search over Heterogeneous Mobile Devices. ‖ Accessed: Oct. 15, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9973344

[47] K. Rajakumari, M. Kumar, G. Verma, S. Balu, D. Sharma, and S. Sengan, ―Fuzzy Based Ant Colony Optimization Scheduling in Cloud Computing, ‖ Computer Systems Science and Engineering, vol. 40, pp. 581–592, Sep. 2021.

[48] M. Núñez-Merino, J. M. Maqueira-Marín, J. Moyano-Fuentes, and C. A. Castaño-Moraga, ―Quantum-inspired computing technology in operations and logistics management, ‖ International Journal of Physical Distribution & Logistics Management, vol. 54, no. 3, pp. 247–274, Jan. 2024, doi: 10.1108/IJPDLM-02- 2023-0065.

[49] Baburao et al., −Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method | Applied Nano science. ‖ Accessed: Oct. 15, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s13204-021-01970-w

[50] S. Aslam, H. Herodotou, S. M. Mohsin, N. Javaid, N. Ashraf, and S. Aslam, −A Survey on Deep Learning Methods for Power Load and Renewable Energy Forecasting in Smart Microgrids, ‖ Renewable and Sustainable Energy Reviews, Mar. 2021, doi: 10.1016/j.rser.2021.110992.

[51] H. Wang and B. Alidaee, −A New Hybrid-heuristic for Large-scale Combinatorial Optimization: A Case of Quadratic Assignment Problem, ‖ Computers & Industrial Engineering, vol. 179, p. 109220, Apr. 2023, doi: 10.1016/j.cie.2023.109220.

[52] Devaraj et al., −Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments, ‖ Journal of Parallel and Distributed Computing, vol. 142, pp. 36–45, Aug. 2020, doi: 10.1016/j.jpdc.2020.03.022.

[53] R. Etengu, S. C. Tan, L. C. Kwang, F. M. Abbou, and T. C. Chuah, −AI-Assisted Framework for Green-Routing and Load Balancing in Hybrid Software-Defined Networking: Proposal, Challenges and Future Perspective, ‖ IEEE Access, vol. 8, pp. 166384–166441, 2020, doi: 10.1109/ACCESS.2020.3022291.

[54] X. Wei, −Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing, ‖ J Ambient Intell Human Comput, pp. 1–12, Oct. 2020, doi: 10.1007/s12652-020- 02614-7.

[55] K. Ramya and S. Ayothi, −Hybrid dingo and whale optimization algorithm-based optimal load balancing for cloud computing environment, ‖ Transactions on Emerging Telecommunications Technologies, vol. 34, Mar. 2023, doi: 10.1002/ett.4760.

[56] N. Rana, M. Latiff, S. Abdulhamid, and S. Misra, −A hybrid whale optimization algorithm with differential evolution optimization for multi-objective virtual machine scheduling in cloud computing, ‖ Engineering Optimization, vol. 54, pp. 1–18, Sep. 2021, doi: 10.1080/0305215X.2021.1969560.

[57] S. V. Nethaji and M. Chidambaram, −Differential Grey Wolf Load-Balanced Stochastic Bellman Deep Reinforced Resource Allocation in Fog Environment, ‖ 2022, doi: 10.1155/2022/3183701.

[58] F. M. Talaat, M. S. Saraya, A. I. Saleh, H. A. Ali, and S. H. Ali, −A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment, ‖ J Ambient Intell Human Comput, vol. 11, no. 11, pp. 4951–4966, Nov. 2020, doi: 10.1007/s12652-020-01768-8.

[59] M. Kaur and R. Aron, −An Energy-Efficient Load Balancing Approach for Scientific Workflows in Fog Computing, ‖ Wireless Personal Communications, vol. 125, Aug. 2022, doi: 10.1007/s11277-022- 09724-9.

[60] Z. Nezami, K. Zamanifar, K. Djemame, and E. Pournaras, −Decentralized Edge-to-Cloud Load Balancing: Service Placement for the Internet of Things, ‖ IEEE Access, vol. 9, pp. 64983–65000, 2021, doi: 10.1109/ACCESS.2021.3074962.

[61] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, −Multi-Agent Deep Reinforcement Learning Based Trajectory Planning for Multi-UAV Assisted Mobile Edge Computing, ‖ Sep. 23, 2020, arXiv: arXiv:2009.11277. Accessed: Oct. 15, 2024. [Online]. Available: http://arxiv.org/abs/2009.11277

[62] S. Wan, X. Li, Y. Xue, W. Lin, and X. Xu, −Efficient computation offloading for Internet of Vehicles in edge computing-assisted 5G networks, ‖ The Journal of Supercomputing, vol. 76, Apr. 2020, doi: 10.1007/s11227-019-03011-4.

[63] B. Mahato, D. Guha Roy, and D. De, −Distributed bandwidth selection approach for cooperative peer to peer multi-cloud platform, ‖ Peer-to-Peer Netw. Appl., vol. 14, no. 1, pp. 177–201, Jan. 2021, doi: 10.1007/s12083-020-00917-2.

[64] X. Xu et al., −Game Theory for Distributed IoV Task Offloading with Fuzzy Neural Network in Edge Computing, ‖ IEEE Transactions on Fuzzy Systems, pp. 1–1, Mar. 2022, doi: 10.1109/TFUZZ.2022.3158000.

[65] S. S. Mangalampalli et al., −Prioritized Task Offloading Mechanism in Cloud-Fog Computing Using Improved Asynchronous Advantage Actor Critic Algorithm, ‖ IEEE Access, vol. 12, pp. 136628– 136656, 2024, doi: 10.1109/ACCESS.2024.3462720.

[66] E. Gures, I. Shayea, M. Ergen, M. H. Azmi, and A. A. El-Saleh, −Machine Learning-Based Load Balancing Algorithms in Future Heterogeneous Networks: A Survey, ‖ IEEE Access, vol. 10, pp. 37689–37717, 2022, doi: 10.1109/ACCESS.2022.3161511.

[67] M. Alam, Department of Computer Science, Al- Barkaat College of Graduate Studies, Aligarh – 202002, Uttar Pradesh, India, Z. Ahmad Khan, and College of Life Science, Nanjing Agricultural University, Nanjng-

Jiangsu, China, —Issues and Challenges of Load Balancing Algorithm in Cloud Computing Environment, ‖ Indian Journal of Science and Technology, vol. 10, no. 25, pp. 1–12, Jun. 2017, doi: 10.17485/ijst/2017/v10i25/105688.

[68] Ebadifard et al., —A Dynamic Task Scheduling Algorithm Improved by Load Balancing in Cloud Computing. ‖ Accessed: Oct. 03, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9122287

[69] Q. Liu, T. Xia, L. Cheng, M. Van Eijk, T. Ozcelebi, and Y. Mao, —Deep Reinforcement Learning for Load-Balancing Aware Network Control in IoT Edge Systems, ‖ IEEE Trans. Parallel Distrib. Syst., vol. 33, no. 6, pp. 1491–1502, Jun. 2022, doi: 10.1109/TPDS.2021.3116863.

[70] Mishra et al., —Load balancing in cloud computing: A big picture, ‖ Journal of King Saud University - Computer and Information Sciences, vol. 32, no. 2, pp. 149–158, Feb. 2020, doi: 10.1016/j.jksuci.2018.01.003.

[71] B. Alankar, G. Sharma, H. Kaur, R. Valverde, and V. Chang, —Experimental Setup for Investigating the Efficient Load Balancing Algorithms on Virtual Cloud, ‖ Sensors, vol. 20, no. 24, Art. no. 24, Jan. 2020, doi: 10.3390/s20247342.

[72] J. G. Muchori and P. M. Mwangi, —Machine Learning Load Balancing Techniques in Cloud Computing: A Review, ‖ IJCATR, vol. 11, no. 06, pp. 179–186, Jun. 2022, doi: 10.7753/IJCATR1106.1002.

[73] Monalisa Kushwaha, —Advanced Weighted Round Robin Procedure for Load Balancing in Cloud Computing Environment. ‖ Accessed: Jul. 08, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9377049

[74] K. Sekaran and P. V. Krishna, —Content-based load balancing of tasks using task clustering for cost optimization in cloud computing environment, ‖ International Journal of Advanced Intelligence Paradigms, Feb. 2022, Accessed: Jul. 08, 2024. [Online]. Available: https://www.inderscienceonline.com/doi/10.1504/IJAIP.2022.121026

[75] Razaq et al., —Fragmented Task Scheduling for Load-Balanced Fog Computing Based on Q-Learning - Razaq - 2022 - Wireless Communications and Mobile Computing - Wiley Online Library. ‖ Accessed: Oct. 03, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/4218696

[76] J. Ramkumar, R. Vadivel, B. Narasimhan, S. Boopalan, and B. Surendren, —Gallant Ant Colony Optimized Machine Learning Framework (GACO-MLF) for Quality-of-Service Enhancement in Internet of Things-Based Public Cloud Networking, ‖ in Data Science and Communication, Springer, Singapore, 2024, pp. 425–438. doi: 10.1007/978-981-99-5435-3_30.

[77] P. Neelakantan and N. S. Yadav, —Proficient job scheduling in cloud computation using an optimized machine learning strategy, ‖ Int. j. inf. tecnol., vol. 15, no. 5, pp. 2409–2421, Jun. 2023, doi: 10.1007/s41870-023-01278-8.

[78] G. Jia, Y. Zhang, S. Shen, B. Liu, X. Hu, and C. Wu, —Load Balancing of Two-Sided Assembly Line Based on Deep Reinforcement Learning, ‖ Applied Sciences, vol. 13, no. 13, Art. no. 13, Jan. 2023, doi: 10.3390/app13137439.

[79] S. Bharany et al., —A Systematic Survey on Energy-Efficient Techniques in Sustainable Cloud Computing, ‖ Sustainability, vol. 14, no. 10, Art. no. 10, Jan. 2022, doi: 10.3390/su14106256.

[80] H. Eljak et al., —E-Learning-Based Cloud Computing Environment: A Systematic Review, Challenges, and Opportunities, ‖ IEEE Access, vol. 12, pp. 7329–7355, 2024, doi: 10.1109/ACCESS.2023.3339250.