Journal of Engineering Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Engineering Research

journal homepage: www.journals.elsevier.com/journal-of-engineering-research

Task scheduling and load balancing in SDN-based cloud computing: A review of relevant research

Masoumeh Mahdizadeh, Ahmadreza Montazerolghaem *, Kamal Jamshidi

Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran

ARTICLE INFO

Keywords:
Software-defined cloud computing
Resource allocation
Cloud computing
Load balancing
Task scheduling

ABSTRACT

This article presents a comprehensive exploration of the architecture and various approaches in the domain of cloud computing and software-defined networks. The salient points addressed in this article encompass: *Foundational Concepts*: An overview of the foundational concepts and technologies of cloud computing, including software-defined cloud computing. *Algorithm Evaluation*: An introduction and evaluation of various algorithms aimed at enhancing network performance. These algorithms include Intelligent Rule-Based Metaheuristic Task Scheduling (IRMTS), reinforcement learning algorithms, task scheduling algorithms, and Priority-aware Semi-Greedy (PSG). Each of these algorithms contributes uniquely to optimizing Quality of Service (QoS) and data center efficiency. *Resource Optimization*: An introduction and examination of cloud network resource optimization based on presented results and practical experiments, including a comparison of the performance of different algorithms and approaches. *Future Challenges*: An investigation and presentation of challenges and future scenarios in the realm of cloud computing and software-defined networks. In conclusion, by introducing and analyzing simulators like Mininet and CloudSim, the article guides the reader in choosing the most suitable simulation tool for their project. Through its comprehensive analysis of the architecture, methodologies, and prevalent algorithms in cloud computing and software-defined networking, this article aids the reader in achieving a deeper understanding of the domain. Additionally, by presenting the findings and results of conducted research, it facilitates the discovery of the most effective and practical solutions for optimizing cloud network resources.

Introduction

Cloud computing encompasses a diverse range of resources, including physical servers, networks, storage, and applications, allowing users to access these resources through networking for their service needs [1]. By renting services from cloud providers for specific durations, users can avoid the high costs associated with purchasing hardware and software [2]. Cloud service providers offer three distinct models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), each tailored to meet varying user requirements [3]. The IaaS model allows users to leverage computational resources such as processing capabilities, storage, and networking elements without directly overseeing the underlying cloud infrastructure. However, users retain control over the operating system and hardware resources, providing a degree of flexibility in their operations [4]. Currently, there is a remarkable surge in the adoption of cloud services, necessitating the enhancement of network infrastructures to meet contemporary demands. Traditional networks, which rely on routers and switches for decision-making, are structured vertically [5]. In these configurations, both the control layer, responsible for traffic management decisions, and the data layer, which forwards traffic based on those decisions, are integrated within individual network devices [6]. Such conventional networks struggle to efficiently manage vast data volumes, particularly in scenarios involving virtual machines, migration, and network setup [4]. To address these challenges, there is a pressing need for networks that are efficient, adaptable, swift, and scalable [7]. Software-Defined Networking (SDN) emerges as a novel paradigm designed to overcome the limitations of traditional networks by decoupling control from network devices and establishing centralized oversight [4]. This shift allows for a comprehensive and flexible view of the network, enabling streamlined and unified management. Consequently, SDN significantly enhances network efficiency while reducing both the costs associated with high-end equipment and the human resources required for network administration [8]. This paper aims to

* Corresponding author.
E-mail addresses: m.mehdizadeh@eng.ui.ac.ir (M. Mahdizadeh), a.montazerolghaem@comp.ui.ac.ir (A. Montazerolghaem), jamshidi@eng.ui.ac.ir (K. Jamshidi).

https://doi.org/10.1016/j.jer.2024.11.002
Received 25 June 2024; Received in revised form 15 October 2024; Accepted 3 November 2024
Available online 9 November 2024

explore these motivations further, focusing on the architectural advancements and methodologies in both cloud computing and SDNs, ultimately providing insights into optimizing network performance and resource management.

Fig. (1) illustrates the Software-Defined Cloud Computing (SDCC) architecture to enhance understanding [9]. As shown in the diagram, the controller component is distinct from the switching devices and plays a centralized role in monitoring and managing the network. Within this architecture, transmission devices focus exclusively on routing packets between input and output ports. In this advanced cloud system, both the controller and transmission devices are interconnected via a communication protocol SDN technology transfers network control from the switching devices, which traditionally manage network routing, to forwarding devices [10]. This separation enables the expansion of network infrastructure by deploying only forwarding devices that are managed by a central controller. The SDN system oversees the network through this central controller, which collects a comprehensive overview of the network from the forwarding devices, selects the optimal decision-making strategy based on this information, and implements it on the forwarding devices via the southbound interface [7].

Fig. (2) presents an overview of the SDN architecture. This framework is structured into three tiers: the Data Plane, Control Plane, and Application Plane, arranged sequentially. The foundational layer, the Data Plane, consists of transmission devices that lack intrinsic control or decision-making software, connecting to establish a network [11]. The Control Plane stands out as the pivotal component in SDN architecture. It operates based on two fundamental principles: 1) Network Monitoring, which gathers a network's comprehensive view from the Data Plane and forwards it to the Application Plane. 2) Network Control, which entails transmitting policies set by the Application Plane to the transmission devices. This ensures network awareness and facilitates optimal decision-making. Positioned above the Control Plane, the Application Plane accesses a holistic, real-time network overview via the Control Plane. Leveraging this data, the Application Plane can enforce and adapt policies to manage the network effectively [8]. SDCC is characterized as an approach to cloud service development where software handles the management and oversight of various resources like computing, storage, data centers, and security [12]. Fig. (3) illustrates the integration of SDN technology into cloud computing. It showcases the interplay between the SDN controller and the cloud controller, along with the connections between controllers and network equipment, encompassing transmission devices, conventional network switches, storage resources, and processing units [13].

Cloud computing faces numerous challenges that researchers are actively working to address. One of the most significant challenges is load balancing and task scheduling [14]. Inefficient resource allocation can result in either over-provisioning or under-provisioning, adversely affecting Service Level Agreements (SLAs) and diminishing profits for cloud providers, while simultaneously increasing costs for users [12]. Consequently, it is crucial to allocate requests to suitable resources to enhance Quality of Service (QoS). A primary objective of task scheduling algorithms is to achieve effective load balancing. Load balancing involves distributing workloads across multiple distributed servers, thereby maximizing resource utilization [15,16]. The main aim of task allocation in a balanced load scenario is to optimize the distribution of tasks among available resources and minimize response times [17].

Scope and contribution

Recently, various methods, techniques, and algorithms have been

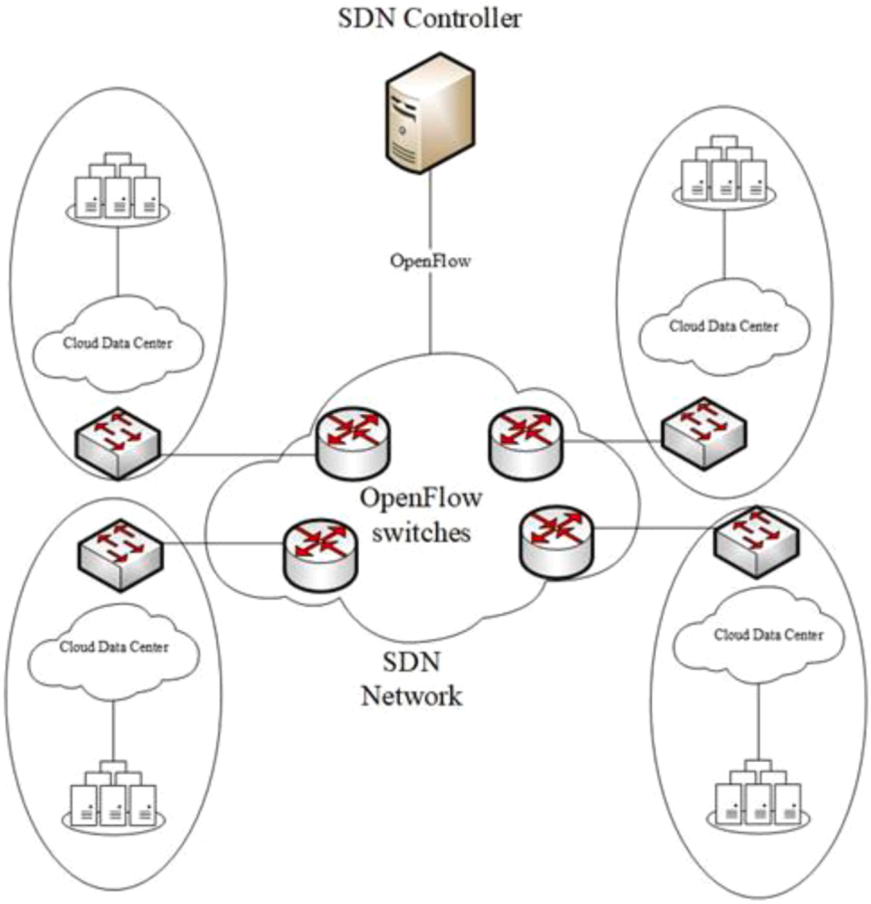

Fig. 1. Cloud computing architecture based on software-defined network.

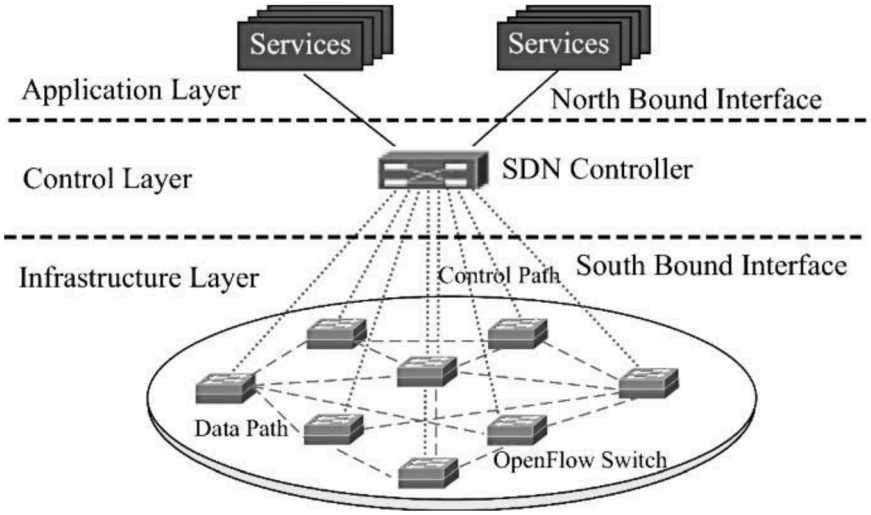

Fig. 2. SDN architecture.

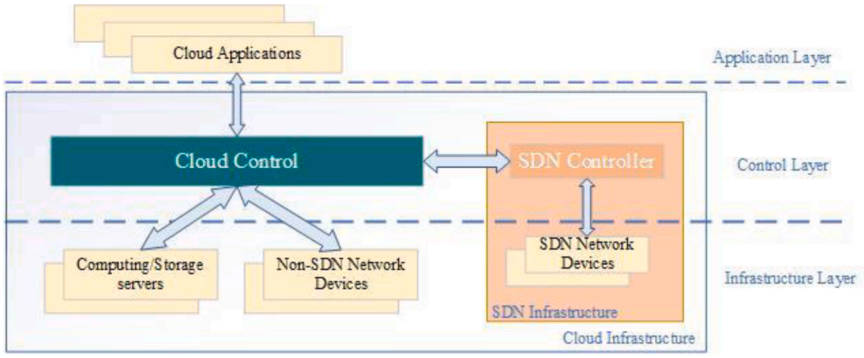

Fig. 3. Software-defined network integration in cloud computing.

implemented with a focus on SDN-based cloud computing in resource management, resource scheduling, resource allocation, energy preservation, load balancing, and QoS. The objective of this paper is to provide a comprehensive review and thorough examination of techniques, frameworks, and models for resource scheduling and load balancing for cloud computing and SDN-based cloud computing. Our contributions can be summarized as follows:

- We have examined review articles that have collected research in the field of service quality in cloud computing and SDN-based cloud computing.
- We have categorized recent trends in resource scheduling mechanization and load balancing, while simultaneously gathering their advantages and disadvantages.
- We have investigated prominent simulators for testing new mechanizations.
- We elucidate potential research endeavors previously outlined, aiding in identifying pathways for current and future utilization.

Organization

The following paragraphs shall be arranged as follows. Reviewing survey articles that have gathered research on service quality in cloud computing and SDN-based cloud computing, and finally, discussing the advantages and disadvantages of each in Section 2. Section 3 examines the latest research on task scheduling and load balancing in cloud environments, concluding with a comparison of the articles. Section 4 focuses on the newest research on task scheduling and load balancing in SDN-based cloud environments. Section 5 evaluates prominent simulators used, providing separate descriptions for each. Open issues and recommendations for future research are discussed in Section 6, concluding the article in Section 7.

Comparison of review articles

Arwa Mohamed et al. [4] conducted a comprehensive review of resource allocation in cloud computing using SDN. The paper explores the enhancement of resource allocation and the dynamic updating of virtual machine traffic demands through SDN, addressing the associated challenges and opportunities. It emphasizes the need for solutions to improve the performance and efficiency of cloud computations. The article provides a detailed comparison of resource allocation in cloud computing with SDN, examines various challenges in the field, highlights the timeliness and novelty of the topics discussed, and contrasts its findings with those of other publications. As a result, this paper generally offers a relative advantage over other articles in this area.

Wenfeng Xia et al. [8] present a comprehensive review of SDN, examining its features, advantages, and disadvantages. The paper contrasts SDN with traditional networking approaches, highlights the benefits and challenges of SDN in the face of evolving communication and information technology trends, and delves into the three-layer architecture of SDN. The findings indicate that SDN can enhance network reliability, security, and scalability. Additionally, the paper mentions aspects such as resource management in mobile radio networks and wireless access.

Abbasi et al. [12] provided a comprehensive review of trends and

developments in Software-Defined Cloud Computing (SDCC). The paper introduces the concepts of cloud computing and SDCC, examining the architectural components of SDCC in detail. It discusses the advancements and challenges associated with SDCC, along with proposed solutions to these challenges. Additionally, the paper compares the practical applications and potential of SDCC across various industries, concluding that the utilization of SDCC can significantly enhance performance, efficiency, security, and flexibility within cloud computing.

Kamlesh Lakhwani et al. [10] conducted an extensive review of data authentication methods in cloud computing. They analyze the key advantages of cloud computing, the security challenges related to data authentication, and strategies to bolster data security in this environment. The review covers a range of topics, including the integration of various tools and techniques for user authentication, the introduction of a factor-based access control model, and the enhancement of trust and reliability in cloud computing settings. This paper distinguishes itself from other studies in the field by thoroughly investigating the benefits and security challenges of data authentication and exploring methods to enhance data security, thereby contributing significantly to improving data protection in cloud environments.

Samah Alnajdi et al. [18] delve into a comprehensive examination of resource allocation techniques in cloud environments. This review encompasses challenges, optimization solutions, various resource allocation models, the advantages and disadvantages of each model and solution, as well as future research directions. The paper conducts a comparative analysis of various models and solutions, delineating their strategies and presenting a comparative table that accentuates the strengths and weaknesses of each option. The results indicate that resource allocation in the cloud still poses challenges and issues that require further research and optimal solutions. The paper's primary strength lies in its thorough examination of various models and solutions, presenting a comparative table, and outlining challenges and future recommendations. Overall, it is recommended as a valuable resource for comprehending the challenges and solutions associated with resource allocation in cloud environments.

Nasrin Akhter et al. [19] investigated the challenges and solutions related to energy resource management in cloud environments. The paper addresses the growing demand for cloud computing, the associated high electricity consumption, energy-aware architectures, and resource allocation techniques, while also highlighting future challenges and potential solutions. It explores a range of topics and offers a thorough comparison of different approaches and solutions. Key findings emphasize the environmental impacts and high operational costs of data centers, the critical need for efficient power management, and the role of renewable energy resources. The paper also discusses future challenges such as workload migration and cooling system optimization. With its broad scope, introduction of innovative solutions, presentation of empirical findings, and practical project examples, this paper serves as a valuable resource for future research and practical implementations in the field.

Abdul Hameed et al. [20] conducted a review focusing on energy-efficient resource allocation in cloud computing environments. The paper examines concepts such as power management policies, various architectures, and the impact of workload on energy savings. The findings indicate that energy-efficient resource allocation in cloud computing environments presents a significant challenge, requiring detailed examination of the effects of different workloads and the provision of solutions for power management and enhanced energy efficiency. The study tackles challenges such as delineating suitable metrics for assessing energy consumption and identifying appropriate methods for achieving energy-efficient resource allocation. Furthermore, the paper sheds light on the advancement of novel approaches for power management and reducing energy consumption in forthcoming endeavors.

Given the complexity of this challenge, the paper emphasizes the importance of establishing proper policies for power management and boosting energy efficiency, underscoring the need for innovative approaches in this field. The importance percentage of SDN in the reviewed articles is shown in Fig. 4.

This section provides a comprehensive comparison of various articles in the field of resource management in cloud environments. Each paper is scrutinized in detail, covering features, advantages, challenges, and proposed solutions. Key findings include recommendations for improving performance and efficiency in cloud environments, the role of emerging technologies such as SDN and SDCC in resource management enhancement, and issues related to energy consumption and data security in these environments. Below is a summary of the articles along with their advantages and disadvantages (Table 1).

Task scheduling and load balancing in cloud computing

Task scheduling in cloud computing

Behera et al. [9] delve into optimizing task scheduling in cloud computing environments using heuristic and metaheuristic algorithms, emphasizing the hybrid GA-GWO algorithm. The Grey Wolf Optimizer (GWO) algorithm is inspired by the behavior of grey wolves and is employed for optimization problem-solving. The GA-GWO algorithm combines the features of the Genetic Algorithm (GA) and GWO to optimize task scheduling in cloud computing environments. Simulation results indicate that this proposed algorithm significantly improves execution time reduction, energy consumption, and overall cost compared to GWO, GA, and PSO algorithms. This research demonstrates that combining genetic and GWO algorithms can enhance task scheduling efficiency in cloud computing environments.

Xiaohan Wang and his colleagues [21] conducted a novel study on task scheduling within collaborative production and computational systems in edge-cloud environments. They introduced the FCRN-assisted random differential evolution approach to enhance scheduling efficiency. This method integrates the differential evolution technique with Feed-forward Convolutional Recurrent Networks (FCRN) to optimize task scheduling processes. The paper evaluates the proposed method, demonstrating its effectiveness in improving efficiency for hybrid task scheduling problems in manufacturing and production systems. When compared to other algorithms and alternative models, the F-RDE method utilizing the FCRN model outperformed its counterparts. The evaluations indicate that this approach is particularly effective in tackling hybrid task scheduling challenges, offering superior accuracy and shorter solution times.

Huayi Yin and colleagues [22] present significant enhancements in optimizing the performance of production lines. They employ heuristic methods like Particle Swarm Optimization (PSO) and Gravitational

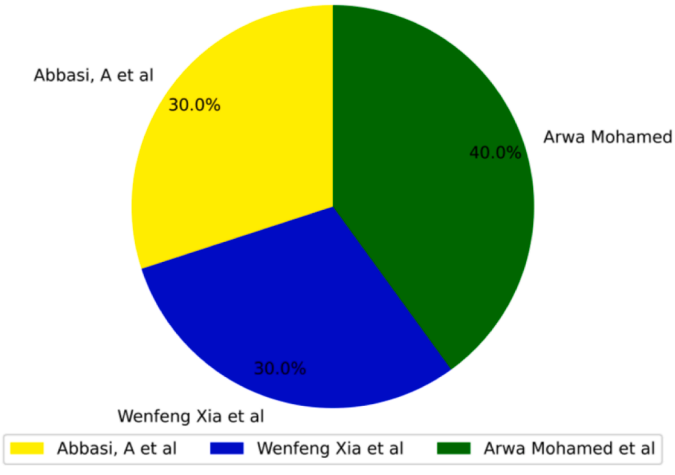

Fig. 4. Distribution of SDN importance in articles.

Table 1
Summary of reviewed articles.

Year	Ref.	Topic	Method	Findings	Advantages
2014	[20]	Energy-Efficient Resource Allocation in Cloud Environments	Review of advantages and disadvantages of energy-efficient resource allocation in cloud environments	Review of advantages and disadvantages of energy-efficient resource allocation in cloud environments	Presents a comprehensive taxonomy and comparison of different methods
2015	[8]	A Comprehensive Survey of SDN	Features, advantages, and disadvantages of SDN - 3-tier architecture	SDN is capable of improving QoS in the network	Comprehensive coverage and overview of the topic of SDN
2016	[18]	Dynamic Resource Allocation in Cloud Environment	Comprehensive review of dynamic resource allocation techniques in cloud environment	Discusses the challenges and issues of resource allocation in cloud environments	Presents various models and solutions for resource allocation
	[19]	Energy Resource Management in Cloud Environments	Review of challenges and solutions for energy resource management in data centers	Discusses the negative environmental impacts and high operational costs of data centers	Introduces innovative solutions and presents experimental results
2018	[10]	Authentication in Cloud Environments	Comprehensive review of data authentication methods in cloud computing	Discusses the security advantages and challenges of authentication in cloud computing	Provides a comprehensive review and comparison of different authentication methods
2019	[12]	A Comprehensive Review of SDCC Trends and Developments	Concepts of cloud computing and SDCC - SDCC architecture elements - SDCC developments and challenges	Using SDCC can help improve QoS in cloud computing	Presents the fundamentals and important principles in SDCC development
2021	[4]	Resource Allocation in Cloud Computing using SDN	A survey of resource allocation in cloud computing using SDN	SDN can improve QoS in the network	Provides a comprehensive comparative analysis of resource allocation using SDN

Search Algorithm (GSA) to develop a combined heuristic approach that reduces the delay of all tasks and diminishes node energy consumption. This optimization method addresses task scheduling issues in cloud-edge computing environments. Using various heuristic techniques, including PSO and GSA, task scheduling problems in intelligent production lines are resolved. Simulation results indicate that the proposed method outperforms other methods like IMBO and IACO in terms of service delay, energy consumption, and task completion rate. This paper introduces a novel approach aimed at enhancing task scheduling in intelligent production lines, resulting in expedited responses and decreased energy consumption.

R. Nithiavathy and colleagues [23] focus on optimization methods for task scheduling in cloud computing environments. The paper introduces a novel algorithm named AGDESMA,[1] which combines the Slime Mould Algorithm with Adaptive Guided Differential Evolution to enhance task scheduling performance. AGDESMA leverages both exploitative and explorative capabilities to prevent getting stuck in local areas, demonstrating significant improvements in average response time and resource utilization rate. Experimental results indicate that AGDESMA outperforms comparative algorithms such as PPSO-DA,[2] MMHHO,[3] PSO-RDAL,[4] and LBPSGORA.[5] This algorithm delivers significant improvements in response time and resource utilization rate, serving as an innovative approach to optimize task scheduling within cloud computing environments.

Hadi Zavieh and colleagues [24] explored the Artificial Neural Network Dynamic Balancing (ANNDB) method to improve task scheduling and optimize resource allocation in cloud networks. This approach utilizes advanced network architecture and a Multi-Layer Perceptron (MLP) to assign requests to high-capacity and high-quality virtual machines, thereby facilitating energy consumption optimization. Evaluations demonstrated that ANNDB significantly outperforms comparative methods such as WPEG, IRMBBC, and BEMEC in terms of energy consumption and power efficiency. Specifically, ANNDB achieved enhancements of 13.81 %, 8.62 %, and 9.74 % in the energy metric, as well as improvements of 3.93 %, 4.84 %, and 4.19 % in the power metric over these comparative methods. ANNDB not only optimizes cloud computing environments by reducing energy consumption but also enhances task scheduling performance, potentially leading to operational cost savings and environmental benefits. This research highlights ANNDB as an effective and efficient method that can assist organizations in further optimizing their cloud computing infrastructure, thereby contributing to environmental sustainability.

S.M.F D Syed Mustapha et al. [25] explored the utilization of DBSCAN[6] and min-min algorithms to enhance efficiency and resource utilization in cloud environments. The paper delves into data clustering and task scheduling with consideration to error probability. Evaluation results indicate that the proposed algorithm leads to a 25 % improvement in execution time, a 6.5 % increase in the number of completed tasks, and a 3.48 % rise in the number of failed tasks compared to comparative algorithms. These findings indicate that the proposed algorithm has the potential to significantly improve the performance and efficiency of data centers within cloud environments, thereby lowering the probability of errors, which in turn can contribute to enhancing the QoS in the cloud.

Cebrail Barut and his team [26] introduced a method named Intelligent Rule-Based Metaheuristic Task Scheduling (IRMTS), which consists of two primary phases. Initially, suitable solutions are extracted using metaheuristic algorithms based on various scenarios, and the gathered data is utilized to construct a dataset and derive intelligent, interpretable rules. In the second phase, appropriate task scheduling solutions are determined using the established set of rules. The IRMTS approach possesses innovative features, and its performance has been validated across simulation environments with diverse scenarios. This method harnesses rule-based metaheuristic algorithms to provide rapid solutions for immediate resource and user demands. Moreover, it formulates rules that are easily comprehensible for operators and engineers, making it adaptable to various dynamic challenges.

Sondas Oufkir and his team [27] introduced a novel method named HunterPlus. This method enhances a Gated Graph Convolutional Network (GGCN) scheduler by integrating a new Convolutional Neural Network (CNN) scheduler. The approach evaluates the QoS parameters for various hosts and tasks, then decides on the best combination of hosts and tasks to optimize the specified QoS parameter. This innovative approach swiftly adjusts to dynamic environments by continuously updating the neural model during each iteration, successfully circumventing scalability challenges in extensive experimental trials. The

[1] Adaptive Guided Differential Evolution-based Slime Mould Algorithm (AGDESMA)

[2] Phasor PSO and Dragonfly Algorithm (PPSO-DA)

[3] Mantaray Modified multi-objective Harris Hawk Optimization (MMHHO).

[4] PSO-based Resource and Deadline-Aware dynamic Load-balanced (PSO-RDAL)

[5] LB with Particle Swarm Genetic Optimization algorithm to improve Resource Allocation (LBPSGORA)

[6] Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

findings consistently demonstrate the superior performance of the proposed CNN model over both GGCN and BiGGCN[7] schedulers, showcasing improvements in energy consumption per request and job completion rate by a minimum of 17 % and 10.4 %, respectively. However, the study does not specifically address load balancing.

Sadoon Azizi and his team [28] introduced two semi-greedy algorithms: The Priority-aware Semi-Greedy (PSG) and the Priority-aware Semi-Greedy with Multi-Start (PSG-M), designed for task scheduling on Fog Nodes (FNs) with the objective of minimizing energy consumption in fog nodes. The PSG algorithm prioritizes requests based on their priority and allocates them to servers with minimal response times to curtail energy usage in the fog environment while adhering to deadlines. However, if an appropriate resource isn't identified, the algorithm aims to minimize the violation time by assigning the best available resource. The Priority-aware Semi-Greedy with Multi-Start algorithm continually runs the Priority-aware Semi-Greedy algorithm, producing the optimal solution discovered across all iterations. Each iteration stands alone, resulting in a distinct solution. These newly proposed algorithms were benchmarked against First Come First Served (FCFS), Earliest Deadline First (EDF), Detour, and Greedy for Energy (GfE) algorithms. The assessment revealed that the introduced method enhances the percentage of tasks meeting their deadlines by up to 1.35 times and diminishes the total deadline violation time by up to 97.6 % compared to the Detour algorithm. Nevertheless, it doesn't account for load balancing. Additionally, as the number of tasks increases, the EDF algorithm demonstrates superior performance over the proposed method in terms of energy consumption and makespan. The distribution of importance in the scheduling of cloud computing tasks in the reviewed articles is given in Fig. 5. Here's a recap of the articles along with their strengths and weaknesses (Table 2).

Task scheduling for load balancing in cloud computing

Khaleel and his colleagues [29] introduced the Regional Awareness Dynamic Scheduling Algorithm (RASA) tailored for load balancing within cloud computing environments. This algorithm operates through a three-phase strategy: task classification, server classification, and an approach based on coalition games. Additionally, the Sparrow Search Algorithm (SSA) is incorporated to optimize task placement. The algorithm successfully minimized additional costs associated with delay, processing time, load imbalances, energy usage, and idle intervals. The study also delves into the complexities of load balancing in cloud setups, highlighting the significance of workload distribution, task scheduling,

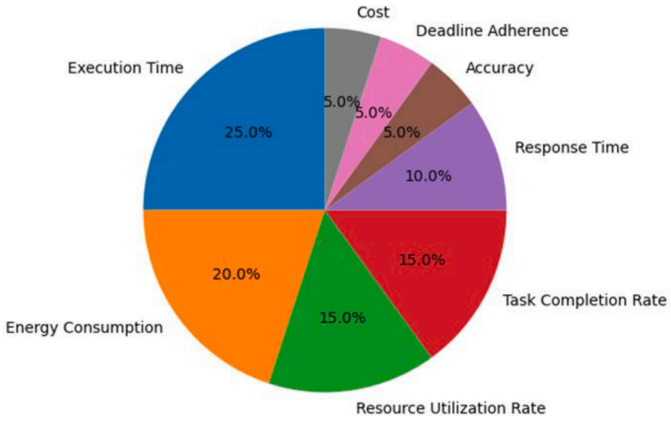

Fig. 5. Distribution of Importance in cloud computing task scheduling.

and the selection of suitable processing servers. By employing game theory techniques and swarm intelligence optimization methods, the algorithm achieves superior load balancing. RASA not only addresses the constraints of previous literature on task-to-VM assignments but also introduces various optimization techniques to boost resource utilization and load balancing. In conclusion, the paper introduces the RASA algorithm, demonstrating its ability to significantly reduce additional costs associated with delay, load imbalances, and energy consumption, while simultaneously improving resource utilization and system performance.

Simaiya et al. [30] introduce a hybrid model named DPSO-GA (Deep learning with PSO-GA) designed for dynamic load balancing within cloud environments. This model integrates deep CNNs and long short-term memory (LSTM) networks with PSO-GA optimization techniques to forecast resource consumption and facilitate load balancing. Through simulations, the proposed model is shown to decrease energy usage in cloud data centers and surpass the performance of current methods. This study presents a novel approach that combines deep learning with optimization strategies for workload prediction and cloud load balancing. Such a method holds promise for enhancing resource efficiency and reducing energy consumption in cloud environments.

Ebadifard et al. [31] improved the PSO algorithm by introducing a novel load balancing technique. In this approach method, every request is first allocated to a virtual machine at random. Subsequently, the virtual machines are categorized as underloaded, overloaded, or balanced. If a machine is identified as overloaded, its requests get shifted to an underloaded counterpart. During this phase, the most compatible machine with the request is chosen, and this procedure persists until no underloaded machines remain. Furthermore, this approach aims to minimize makespan, thereby optimizing resource utilization. While the conventional PSO algorithm assigns tasks to virtual machines randomly, the strength of this new method lies in the additional load balancing step implemented after random task assignment. When juxtaposed with the Round Robin task scheduling, the foundational PSO algorithm, and a stand-alone load balancing technique, the findings revealed that this novel method augmented resource utilization by 22 % and curtailed makespan by 33 % compared to the fundamental PSO algorithm.

Kruekaew et al. [16] introduced a technique called MOABCQ, which combines the Artificial Bee Colony (ABC) algorithm with the Q-learning algorithm, a type of reinforcement learning. The primary goal of this hybrid approach is to enhance the exploitative capabilities of the ABC algorithm. Initially, a request is randomly assigned to a virtual machine, while the search for an optimal machine continues. If a superior machine is found that outperforms the current one, requests are redirected to this machine. In this case, the superior machine is rewarded, while the incumbent machine incurs a penalty, leading to an update of the Q-table. Conversely, if the new machine does not exceed the performance of the current one, it receives a penalty, and the existing machine is rewarded, prompting another Q-table update. This Q-table is also crucial for assigning new requests. To evaluate the effectiveness of this method, it was compared against well-known heuristic task scheduling techniques, including the Max-Min algorithm, First-Come First-Served (FCFS) algorithm, and Largest Job First (LJF) algorithm. Additionally, it was benchmarked against meta-heuristic task scheduling methods such as the Multi-Objective Particle Swarm Optimization (MOPSO) algorithm, Multi-Objective Cuckoo Search (MOCS) algorithm, and the authors' earlier approach known as Heuristic Task Scheduling with ABC and Largest Job First (HABC-LJF) algorithm. The results indicate that MOABCQ surpasses the other methods in terms of makespan, cost reduction, minimization of imbalance degree, and resource utilization.

Ramezani Shahidani and her colleagues [32] introduce a task scheduling algorithm for fog computing named Reinforcement Learning Fog Scheduling (RLFS), which is grounded in reinforcement learning. The algorithm's primary objectives are to equalize the load, diminish the average response time, and curtail energy consumption. Requests are categorized into three groups: real-time, significant, and regular.

[7] Bidirectional Gated Graph Convolution Network (BiGGCN)

Table 2
Task Scheduling in Cloud Computing.

Year	Ref.	Authors	Proposed Algorithm	Improvement Parameters	Advantages	Disadvantages
2022	[28]	Azizi et al.	PSG and PSG-M	Improved scheduling of IoT tasks, reduced overall energy consumption, increased profitability of fog service providers	Accurately evaluate the performance of algorithms, meeting the deadline requirements for a large number of IoT tasks	Capacity and management limitations of fog resources, high energy consumption, complexity in allocating diverse and dynamic resources to IoT tasks
2024	[9]	Behera et al.	hybrid GA-GWO	Improve execution time, energy consumption and cost	Reduces execution time, energy consumption, and total cost scheduling	Requires complex configuration and tuning
	[21]	Xiaohan Wang et al.	fully convolutional regression network (FCRN)	Using FCRN model to estimate fitness function, using four DE operators to generate offspring, combining several methods to improve performance	Superior performance in solving hybrid task scheduling problems	More calculations are needed to train the FCRN model
	[22]	Huayi Yin et al.	Multi-PSG	Service delay, energy consumption, task completion rate	Improves performance over other methods, reduces energy consumption, increases task completion rate	Requires more computation for executing heuristic algorithms
	[23]	R. Nithiavathy et al.	AGDESMA (Adaptive Guided Differential Evolution-based Slime Mould Algorithm)	Using a combination of Slime Mold Algorithm and Adaptive Guided Differential Evolution, heuristic and discovery capability to avoid getting stuck in local areas	High performance in task scheduling, better results than comparative algorithms, improved response time and resource utilization rate	The need to adjust the parameters of the algorithm, the need for a suitable test environment to evaluate the performance
	[24]	Zavieh et al.	ANNDB	Improves energy and power consumption, optimizes task scheduling, utilizes MLP and advanced network architecture	Increasing the efficiency of energy and power, improving the scheduling of tasks, the possibility of optimizing cloud computing environments	Requires more complex computation for algorithm execution, requires specialized knowledge for correct implementation
	[25]	Syed Mustapha et al.	DBSCAN and min-min	Execution time, number of completed tasks, number of failed tasks	25 % improvement in execution time, 6.5 % increase in the number of completed tasks, 3.48 % increase in the number of failed tasks	-
	[26]	Cebrail Barut et al.	Intelligent Rule-Based Metaheuristic Task Scheduling (IRMTS)	Reduces execution time, applicable in dynamic scenarios, generates interpretable rules	Increased complexity in parameter tuning, may lead to suboptimal solutions	
	[27]	Iftikhar et al.	HunterPlus: AI-based job scheduling for fog-cloud computing environments	Resource scheduling and optimization in fog-cloud computing	Optimizes resource management, improves energy efficiency in fog-cloud environments	Complexity in implementing AI algorithms, increases computational cost

Regular requests are directed to the cloud because they aren't sensitive to delays. In contrast, real-time and significant requests are primarily handled at fog nodes due to their sensitivity to delays. Reinforcement learning governs the scheduling of these priority requests. The reinforcement learning approach in the proposed algorithm operates in two stages. Initially, it aims to handle all real-time and significant requests at the fog nodes. If a chosen fog node cannot meet the request, an alternative node is chosen from the available fog nodes capable of addressing the request using a greedy policy. Subsequently, the action's reward is computed, and the Q-table undergoes an update. Comparative analysis with LBSSA, DRAM, GA, and PSO-SA algorithms reveals that the proposed RLFS method surpasses others in terms of load balancing and average response time. Nonetheless, factors such as Q-table updates, action selection, and the inherent overhead of reinforcement learning contribute to the algorithm's suboptimal performance regarding execution time.

Fatemeh Abbasifard et al. [17] present an efficient method for dynamic request scheduling on virtual machines (VMs) designed to enhance load balancing in cloud data centers. This approach combines an autonomous VM adjustment framework with a predictive component to forecast future system states. The process includes: assessing the available resources in each VM (such as processing power, memory, and bandwidth), evaluating the compatibility between each request's requirements and the available VMs, and selecting the VM that best matches the request's needs based on the required resources.

Through integration of prediction, the method avoids selecting VMs that may become overloaded in the future, thereby reducing the overhead associated with relocating requests from overloaded VMs to others. Moreover, the autonomous architecture enables VMs to adapt automatically to environmental changes, avoiding excessive requests that can lead to overhead and minimizing the need for load balancing algorithms. Simulation results demonstrate that the proposed approach effectively achieves load balancing among VMs, allocates requests to appropriate VMs based on their resource requirements, and reduces both response time and makespan.

Ali Asghari and his colleagues [33] present an innovative approach to cloud resource management that integrates the State-Action-Reward-State-Action (SARSA) learning algorithm with a Genetic Algorithm (GA). Both Q-learning and SARSA are forms of Reinforcement Learning (RL) techniques, with SARSA being preferred due to its superior performance. The fundamental difference between these algorithms lies in their policy update mechanisms: Q-learning updates policies in a greedy manner, while SARSA utilizes its learned policy for updates. The proposed methodology consists of two stages: Stage 1: In this stage, requests are evaluated using the SARSA RL model, resulting in a sorted list of jobs based on increasing execution time, which is then forwarded to the next stage. Stage 2: Here, jobs are assigned to appropriate resources through a combination of GA and RL. Experimental results demonstrate that the proposed algorithm, when compared to the Multi-Objective HEFT (MOHEFT) and Min-Cost Path (MCP) algorithms, achieves a decreased makespan, enhanced resource utilization, and improved load balancing.

Ashish Gupta et al. [34] introduce an innovative method for scheduling independent tasks within the cloud, employing an Ant Colony Optimization (ACO) algorithm termed Load Balancing Ant Colony Optimization (LB-ACO) algorithm. The primary aim of this algorithm is

to optimize load balancing and reduce the makespan. In this strategy, initial task assignments are randomized. Subsequently, the maximum execution time is determined, and further task allocations are conducted using the LB-ACO algorithm based on this computation. The algorithm leverages pheromones and random exploration to identify the optimal VM for each task. A comparative study reveals that the LB-ACO algorithm surpasses existing methods in terms of both load balancing and makespan. However, it's important to note that the algorithm doesn't take task priorities into account.

Wang and colleagues [35] present a novel scheduling algorithm called the Job and Load Balancing Genetic Algorithm (JLGA) tailored for cloud computing environments. This algorithm is based on an enhanced Adaptive Genetic Algorithm (AGA) and integrates the dual-time-scale adaptive algorithm with the Load-Balancing Genetic Algorithm (LBGA). JLGA facilitates job scheduling by prioritizing smaller requests while also incorporating load balancing considerations. The initial population is generated using a greedy algorithm. It is important to note that this study assumes equal job priorities, which may not reflect the complexities of real-world cloud computing scenarios. The advantages of JLGA include enhanced performance compared to existing algorithms, reduced response times, improved load balancing capabilities, and optimized resource utilization.

Lahande et al. [36] explore the potential of Reinforcement Learning (RL) techniques in enhancing cloud resource utilization through load balancing. Through experiments using the SIPHT dataset, they illustrate that resource scheduling algorithms can notably boost load balancing and overall cloud resource efficiency. Their findings indicate that scheduling methods like First Come, First Serve (FCFS), Maximum – Minimum (Max – Min), Minimum Completion Time (MCT), Minimum – Minimum (Min – Min), and Round – Robin (RR) yield substantial enhancements in this domain. The study posits that integrating an RL-driven artificial intelligence framework into load balancing and optimizing cloud resources can markedly elevate the quality of service offered by cloud platforms. The research underscores that the application of RL approaches can enhance both the performance and efficiency of cloud resources, presenting an intelligent remedy for load balancing challenges in cloud environments. The estimated importance of QoS aspects in the reviewed articles is given in Fig. 6.

In this section, we have introduced various methods and algorithms designed to optimize task scheduling and load balancing within cloud computing environments. These methods encompass hybrid algorithmic optimizations and reinforcement learning techniques. The primary objectives of these approaches are to enhance resource efficiency, minimize energy consumption, and ensure equitable load distribution among resources, all while striving to improve overall system performance and efficiency. Below is a summary of the articles, highlighting their strengths and weaknesses (Table 3).

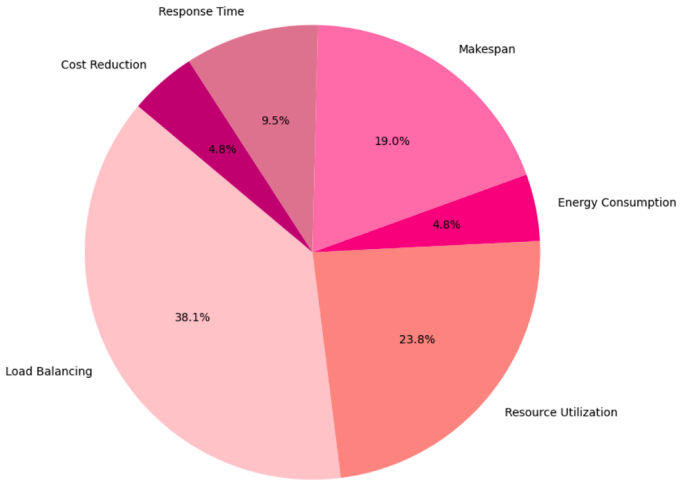

Fig. 6. Estimated Importance of QoS Aspects.

Task scheduling and load balancing in software-defined cloud computing network

Load balancing in software-defined cloud computing network

Çavdar et al. [37] introduce an adaptive load distribution technique designed for data centers operating on SDN. This approach employs a meta-heuristic method known as discrete particle swarm optimization and enhances load distribution across links and switches using a unique combined cost function. The primary goal is to identify routes that minimize connection loads and evenly distribute traffic between the best sources and destinations, thereby reducing strain on both links and switches. The adaptable characteristic of this technique ensures that the most efficient routes are consistently refreshed for peak operational efficiency. Simulated outcomes demonstrate that this approach results in: Decreased flow completion durations, Elimination of packet loss, Diminished energy usage, Lower memory consumption, Enhanced network throughput. Furthermore, this method surpasses current state-of-the-art techniques, demonstrating its potential to significantly enhance load distribution within SDN-enabled networks.

Yaofang et al. [38] explored the Proximal Policy Optimization (PPO) technique for resource allocation in edge computing networks. This approach frames energy consumption optimization and network load balancing as a multi-objective challenge. Findings indicate that PPO minimizes energy use while achieving balanced network loads in edge computing environments. The algorithm notably enhances network efficiency and ensures consistent training stability. Consequently, PPO is identified as a potent strategy for optimizing resources and elevating performance in edge computing networks. The study delves into the Markov decision-making process, representing the issue as an MDP, and underscores the efficacy of PPO in tackling intricate challenges.

Song et al. [39] present the Mixed-Flow Load-Balanced Scheduling (MFLBS) algorithm, specifically designed for software-defined cloud data center networks. This algorithm aims to improve network efficiency and performance by addressing the needs of both cloud and Internet of Things (IoT) networks. MFLBS adapts the distribution of network load according to the distinct characteristics and sizes of data flows, ensuring an equitable balance between smaller and larger flows. By utilizing both preventive and dynamic traffic control strategies, the algorithm optimizes bandwidth resource allocation and routing in response to varying network conditions. Experimental results indicate that MFLBS outperforms other algorithms, including dynamic approaches like DLBS and static ones like First-Come, First-Served (FCFS), leading to reduced delays and enhanced data throughput. This algorithm offers significant improvements in the performance of cloud data center networks and demonstrates considerable practical value.

Pathan et al. [40] explored the merging of data center networks with SDN to improve network management and adaptability. As the number of network devices in these centers grows, effectively managing the vast data from these devices becomes crucial. This study introduces a fresh approach for admission and routing that takes into account the significance or priorities of network flows, path energy, and path load. By applying the SDN framework to data center networks, they initially devise a mixed integer linear programming (MILP) model that considers flow priority, path energy, and path load concurrently. This MILP model aims to enhance flow count while reducing energy consumption and network load variance, or it can achieve a middle ground among these factors. Subsequently, they present two self-aware heuristic methods: the Priority-based Energy-efficient Maximization Algorithm (PEMA) and the Priority-based Load Balancing Algorithm (PEDL). PEMA strives to maximize flows with minimal energy, whereas PEDL aims to boost flows while minimizing load variance. The proposed routing strategies are then put to test through implementation, and simulation outcomes

Table 3
Task Scheduling for Load Balancing in Cloud Computing.

Year	Ref.	Authors	Proposed Algorithm	Improvement Parameters	Advantages	Disadvantages
2014	[35]	Tingting Wang et al.	JLGA	Improved QoS	Improves execution time and load balancing	Slow convergence compared to AGA
2018	[31]	Ebadifard et al.	PSO-based scheduling algorithm	Improved using load balancing technique	Increases resource utilization and reduces execution time	Requires tuning of algorithm parameters
2019	[34]	Ashish Gupta et al.	LB-ACO	Improves load balancing and reduces computation time	Increases resource efficiency, reduces execution time, improves load balancing	Requires tuning of algorithm parameters
2020	[17]	Fatemeh Ebadifard et al.	Autonomous Load Balancing	Improves communication costs, better load balancing, better workload distribution	Increases system stability, reduces response time, increases resource utilization	Needs further study to fully understand the drawbacks and limitations
	[33]	Asghari et al.	SARSA and Genetic Algorithm	Resource allocation, scheduling and load balancing	Faster, Accurate Resource Allocation in Computing	Requires parameter tuning, high complexity of genetic algorithm
2022	[32]	Shahidani et al.	Fog-based Reinforcement Learning Algorithm	Improves response time and reduces service delay	Cloud Data Center Efficiency and Reduced Service Delays	-
	[16]	Boonhatai Kruekaew et al.	MOABCQ	VM Scheduling Optimization and Cost-Effective Resource Use	Efficient VM Load Balancing and Resource Utilization	Complex Algorithm with High Computational Demand
2023	[36]	Prathamesh Vijay Lahande et al.	Reinforcement Learning Approach	Improving load balancing and productivity of cloud resources, optimizing the load balancing process using resource scheduling algorithms.	Optimizing Cloud Efficiency with Reinforcement Learning	Educational Setup for Reinforcement Learning Implementation

demonstrate their superiority over existing methods in terms of flow success rate, energy conservation, and Load balancing.

Kang et al. [41] developed an SDN-based Intra-Cloud Manager (S-ICM), which consists of two main modules: monitoring and decision-making. The monitoring module collects data on various metrics, including the number of pending requests, delay rates, loss rates, and average response times for each request, either at specified intervals or upon the controller's request. In the decision-making module, requests are queued based on their arrival times, and the system directs each request to the server with the shortest average response time. S-ICM further adjusts the request dispatch rate to servers based on a predefined minimal average response time threshold, decreasing the rate when the number of requests exceeds a certain limit. The study compares S-ICM with the Honey Bee Foraging Algorithm (HFA) and Round Robin (RR) methods. The results demonstrate that S-ICM achieves better average response times than the other two approaches. However, to maintain continuous monitoring of cloud conditions, S-ICM generates additional control messages across the network, resulting in increased network overhead.

Sharma et al. [13] investigated the impact of incorporating a load balancer in a cloud environment. They utilized a combination of OpenStack and OpenDaylight (ODL) to set up a software-defined network-based cloud and integrated a load balancer into the system. Employing the Round Robin scheduling approach, tasks were allocated to two identical web servers, WS-1 and WS-2, within OpenStack. Upon receiving a request, the load balancer directed it to either WS-1 or WS-2 in a rotating fashion. The study compared the processing times of requests with and without the load balancer. Results indicated that implementing a load balancer introduced latency due to the inclusion of an additional processing component and request queuing. However, despite this latency, integrating a load balancer enabled the system to handle multiple requests concurrently without encountering crashes.

Osei Kofi et al. [42] introduced an optimization algorithm tailored for managing network load in SDN within cloud computing environments. This approach aims to enhance network performance by combining the IP hash load distribution algorithm with a weighted scheduler. By employing the hash function, network security is ensured, while dynamic routing adjustments help prevent congestion. Simulations conducted in the study revealed that the proposed algorithm significantly improves data throughput, aids in congestion management, enhances accessibility, and reduces network latency. The study underscores the importance of enhancing network resource efficiency through the integration of weighted scheduling and IP hash load distribution techniques, highlighting their contributions to heightened network reliability, security, and overall performance.

Burke et al. [43] presented a deceptive attack technique in SDN aimed at manipulating load balancing control through false announcements. This approach utilizes a probabilistic model for representation and develops algorithms to create false announcements, allowing attackers to modify attack parameters to achieve specific goals. Such an attack grants the attacker considerable control over traffic flow and the ability to influence the volume of traffic affected via a compromised switch. The study underscores the significant security risks posed by reliance on false announcements, pointing out that they can lead to severe security breaches. When tested against four widely-used load balancing algorithms, the attack demonstrated a marked ability to disrupt load balancing within SDN networks, thereby revealing vulnerabilities in current defense mechanisms. Below is a summary of the articles, highlighting their strengths and weaknesses (Table 4).

Load balancing in SDN-based IoT networks

Ali et al. [44] proposed a load balancing method for multi-domain IoT networks using SDN. This method utilizes a Multi-Criteria Decision Making (MCDM) algorithm called the Analytic Network Process (ANP) for sub-controller selection, and switch migration is modeled using a 0/1 knapsack algorithm. This method helps to improve the load balancing of controllers and enhance the QoS in IoT networks. With increasing traffic in the network, the processing capacity of controllers may not be sufficient to handle the traffic, but this method provides significant improvement by optimal selection of sub-controllers and resource management. By modeling switch migration and using ANP, this method addresses load balancing and QoS improvement in IoT networks.

Montazerolghaem et al. [45] introduced an innovative SDN-centered method designed to balance server and sending device loads while ensuring the QoS demands of diverse IoT services. Initially, they established that simultaneously addressing these two issues is NP-hard. To mitigate time complexity, the problem is bifurcated into: 1) Selection of servers, and 2) Selection of paths. The system's architecture comprises: 1) Infrastructure, where servers, sending devices, and IoT devices connect via specific interfaces, and 2) A software-defined network controller linked to servers through the sFlow protocol. This controller gathers resource consumption statistics (like memory and CPU) and

Table 4
Load Balancing in Software-Defined Cloud Computing Network.

Year	Ref.	Authors	Proposed Algorithm	Improved Parameters	Advantages	Disadvantages
2016	[41]	Kang and Choo	SDN-based load balancing algorithm	Improved efficiency, manageability, scalability, and control in cloud environments	Enhanced security and reliability in cloud environments, improved load balancing and resource utilization	Requires more complex infrastructure and additional implementation costs
2019	[13]	Rinki Sharma et al.	S-ICM (SDN-enhanced inter cloud manager)	Enhances load balancing in SDN-based cloud environments	Elevates service levels, improves resource utilization, and enhances scalability	-
2020	[43]	Quinn Burke et al.	Attack method based on misrepresentation in SDN	Adjustable attack parameters	The possibility of high control over traffic, flexibility in setting attack parameters	Dependence on false declarations that can lead to security failures Damage to load balancing of SDN networks Weaknesses in existing defense systems
2022	[39]	Song et al.	Mixed-Flow Load-Balanced Scheduling (MFLBS)	Optimizing bandwidth allocation, network load management, reducing data transfer delay	Increasing data transmission power, reducing delay, balancing between small and large flows, preventing network disruptions	Requires more complex calculations for proactive settings, requiring more resources to implement
2023	[37]	Tuğrul Çavdar et al.	Discrete Particle Swarm Optimization with a hybrid cost function	Load balancing on switches and links, choosing the path that minimizes the connection load	Reducing the time, it takes for streams to reach their destination, not losing packets, using less energy, using less memory, increasing traffic in the network	-
	[38]	Yaofang Li et al.	Proximal Policy Optimization (PPO)	Reducing energy consumption, improving network load balance	Improved network efficiency, training stability, optimal resource management, achieving optimal load balance	The need to adjust the parameters, the complexity of the multi-objective optimization problem
	[42]	Evans Osei Kofi et al.	HDW (Hash IP load balancing algorithm with Weighted scheduler and Dynamic switching of routing path)	Increased throughput, congestion control, improved access, reduced latency	Increase network security, improve reliability, improve network performance	Requires more complex implementation and management
2024	[40]	Naimul Pathan et al.	PEMA and PEDL Algorithm	Energy consumption, load balance, flow success ratio	Better performance than existing methods, energy saving, better load balancing	Longer runtimes are possible

forecasts future resource needs using the NLMS algorithm. Subsequently, a fuzzy system, informed by the algorithm's output, sets a load receiving window for each server. This window defines each server's capacity to receive, with the server boasting a broader window being prioritized for allocation. The outcomes demonstrate enhancements in IoT QoS metrics, such as throughput and delay, and ensure that IoT servers remain unburdened even during peak traffic.

Here's a recap of the articles along with their strengths and weaknesses (Table 5).

Task scheduling in software-defined networks

Siapoush et al. [47] proposed a method for scheduling large data tasks using the Tabu algorithm in conjunction with SDN. By separating network control logic from transmission devices like routers and switches, this approach improves network performance. The results indicate that the Tabu algorithm significantly reduces the completion time for large data tasks and avoids local optimal solutions. By taking the network state into account and leveraging SDN, this method enables precise task scheduling and enhances network performance. Overall, the use of the Tabu algorithm and SDN in scheduling large data tasks leads to increased network efficiency and shorter task completion times.

Singh et al. [48] introduced an optimization strategy for controller placement within SDN. This strategy employs the Particle Swarm Optimization (PSO) algorithm in conjunction with the Capacitated Controllers Arrangement (CCA) technique, aiming to simultaneously reduce network latency and uphold network reliability, even in the event of failures affecting up to n-1 out of n deployed controllers. Experimental results indicate that this innovative approach to intelligent controller allocation effectively reduces network latency and optimizes switch assignments. The findings suggest that utilizing three optimal controllers can reliably manage the network while minimizing average network delay. Future research should prioritize the application of multi-objective optimization techniques to address controller placement challenges with a focus on enhancing the reliability of the SDN architecture.

Table 5
Load Balancing in SDN-based IoT Networks.

Year	Ref.	Authors	Proposed Algorithm	Improved Parameters	Advantages	Disadvantages
2019	[45]	Ahmadreza Montazerolghaem et al.	SDN-based Heuristic Approach	Improved QoS for IoT services, load balancing between IoT servers	Using SDN-based framework to improve QoS and load balancing, using heuristic algorithm to reduce time complexity	Model complexity due to the use of binary variables
2023	[44]	Jehad Ali et al.	Multi-domain SDN Slave Controller Load Balancing (SDN-SC-LB)	Optimal selection of sub-controllers using ANP, switching migration modeling with 1/0 knapsack algorithm	Improving load balancing of controllers, improving quality of service in IoT networks, optimal selection of sub-controllers with empty resources	The need for a simulation environment or emulator to evaluate performance in real conditions
2022	[46]	Ahmadreza Montazerolghaem et al.	Load balanced Software-defined Internet of Multimedia Things	Quality of service and quality of experience, such as throughput, multimedia delay, R factor, and mean opinion score	a tradeoff between efficiency and energy using proactive heuristic algorithms, including the network sizing method and the NFV technology	Using VMs instead of the Docker-Container or OpenStack

Here's a recap of the articles along with their strengths and weaknesses (Table 6).

Task scheduling in software-defined cloud computing network

Sellami et al. [49] proposed a DRL method to develop an intelligent network utilizing SDN, with the primary objectives of minimizing network latency and reducing energy consumption. This method involves training the SDN controller to select the optimal scheduling policy that balances energy efficiency with the minimization of delays when allocating requests to appropriate fog nodes. Evaluations indicate that this approach outperforms both deterministic and random task scheduling strategies in terms of delay and energy efficiency, achieving an average delay of approximately 12.5 ms. However, a notable limitation of this approach is its failure to address load balancing among fog nodes.

Sellami et al. [50] propose an approach that integrates deep reinforcement learning into the SDN controller to select the optimal decision-making policy for task scheduling. This method ranks fog nodes based on their available energy and current workload status during execution time, selecting the node with the lowest energy consumption for task execution, thus reducing processing delays. Subsequently, each time a request is successfully allocated to a node, its reward is increased, guiding future requests towards nodes with higher rewards. The proposed approach is compared with three algorithms: random, deterministic, and A3C. According to the results, this approach better preserves the battery level in fog nodes and achieves an energy efficiency gain of 87 % compared to the other three algorithms.

Al-hammadi et al. [51] Investigated the use of collaborative computing for emergency task scheduling in Software-Defined Mobile Edge Computing (MEC) networks. The paper focuses on replacing regular tasks and emergency tasks on MEC servers. Regular tasks are generated periodically and, if their deadlines are not met, they do not have serious consequences. On the other hand, emergency tasks have higher priority and require prompt and timely execution to prevent serious issues. This paper presents four different scheduling algorithms for managing emergency tasks, including task allocation to nearby servers with sufficient computational resources, network congestion control, selection of suitable collaborative servers, and resource allocation for emergency tasks. Extensive simulation results indicate that this approach performs better than other methods, reducing the overall task execution time and meeting the deadlines for emergency tasks. It is also noteworthy that this paper is authored by a team from various universities in China and Australia and is supported by several research organizations (Table 7).

Scheduling tasks for load balancing in software-defined cloud computing network

Sharma et al. [52] investigated an optimization algorithm called BMU-COA for load balancing in software-defined cloud computing. This algorithm integrates two optimization techniques: Blue Monkey Optimization (BMO) and the Chimp Optimization Algorithm (COA). Simulation results show that BMU-COA significantly enhances load balancing and task allocation optimization in SDN networks, leading to reduced migration costs and improved migration efficiency compared to other algorithms. Moreover, BMU-COA demonstrated considerable advancements in performance and optimization, potentially surpassing other algorithms in addressing optimization challenges and load balancing within networks.

Al-Mansoori et al. [53] proposed a framework that integrates virtual machines and software-defined networking to optimize cloud resources. In this approach, they utilized Complex Event Processing (CEP) mechanisms for data stream processing and analysis. CEP is a powerful technology that provides quick and comprehensive responses to large data streams. The assumption in this paper is that a single CEP is composed of multiple parallel CEPs running in the cloud environment. Each CEP has dynamic load balancing that synchronizes data based on their arrival times. Moreover, a software-defined network controller is employed to optimize cloud network resources. Data processing in this framework is based on a time-series model that determines the type of virtual machine. Data streams are queued based on a first-in-first-out approach. Then, virtual machine allocation is done considering the size of data streams and the processing power of the virtual machine. When the data stream size exceeds the virtual machine's processing capability, the data is buffered and sent to a virtual machine with better processing power. According to the results obtained in this framework, a virtual machine can handle up to 2000 requests in a maximum of 136 seconds compared to traditional cloud computing. Here's a recap of the articles along with their strengths and weaknesses (Table 8).

The Fig. 7 shows that load balancing is the most important QoS factor, followed by throughput, delay, and energy consumption. Packet loss, security, and delay are the least important factors. The legend below the chart explains the different categories.

This section explores various strategies and algorithms for load balancing and task scheduling in software-defined cloud computing networks, with a strong emphasis on Quality of Service (QoS) aspects. We examine methods such as adaptive load distribution, reinforcement learning, meta-heuristic techniques, and multi-criteria decision-making algorithms. Each of these approaches is specifically designed to: Optimize Resource Utilization: Efficient resource allocation ensures that computational resources are used effectively, which is critical for maintaining high QoS levels. Minimize Energy Consumption: By optimizing load distribution, these strategies contribute to lower energy usage, which not only reduces operational costs but also supports sustainability, an increasingly important aspect of QoS. Enhance Network Performance: Improved network performance directly correlates with better QoS, as it leads to reduced latency, increased throughput, and higher reliability in service delivery. Improve QoS in Cloud Computing and IoT Environments: The algorithms discussed are tailored to meet specific QoS requirements, such as response time, availability, and reliability, which are essential for user satisfaction and effective service delivery. Additionally, this section addresses challenges related to controller placement and network security, both of which can significantly impact QoS. For instance, improper controller placement can lead to increased latency and reduced network responsiveness, while security vulnerabilities can compromise service integrity, thereby affecting QoS. Moreover, we explore the effects of deceptive attack techniques on load

Table 6
Task Scheduling in Software-Defined Networks.

Year	Ref.	Authors	Proposed Algorithm	Improved Parameters	Advantages	Disadvantages
2023	[47]	Mina Soltani Siapoush et al.	Tabu Search Approach	Completion time of big data tasks, scheduling accuracy, local optimal optimization	Improve network performance and task completion time, use SDN to improve scheduling accuracy	The need for a network environment that supports SDN is more complex to implement than traditional methods
2024	[48]	Gagan Deep Singh et al.	Particle Swarm Optimization (PSO) and Capacitated Controllers Arrangement (CCA)	Reducing network latency, increasing network reliability up to the failure of n−1 controller out of n installed controllers	Optimizing network latency, increasing network reliability, optimizing switch allocations, intelligent management of controllers	The need for more experiments on different SDN architectures, the need to adapt the proposed method to changing network conditions

Table 7
Task Scheduling in Software-Defined Cloud Computing Network.

Year	Ref.	Authors	Propoed Algorithm	Improved Parameters	Advantages	Disadvantages
2020	[49]	Sellami et al.	Deep Reinforcement Learning (DRL)	Increased energy efficiency, improved response time, optimized task allocation in IoT networks	Reduced energy consumption, improved response time, optimized task allocation, increased network efficiency	Need for large training data, complexity of DRL algorithm training, need for interactive environment for training
2022	[50]	Sellami et al.	Deep reinforcement learning-based task scheduling and transfer	Optimizing energy efficiency, scalability, latency, bandwidth.	1. Using deep reinforcement learning for intelligent scheduling of tasks. 2. Improving energy efficiency in Internet of Things networks. 3.Improving the scalability and reducing the delay in the transfer of tasks.	1. The need for computing resources for deep reinforcement learning. 2. Complexity in implementing and adjusting the algorithm.
2024	[51]	Al-hammadi et al.	Scheduling Algorithms for Emergency Tasks in MEC Networks	1. Allocation of tasks to nearby servers with sufficient computing resources 2. Control of network congestion 3. Selection of suitable partner servers	1. Improving overall task execution time 2. Responding to emergency task deadlines 3. Improving MEC network performance in emergency situations	1. The need for collaborative computing and increasing complexity 2. The possibility of increasing computing costs due to the use of additional resources

Table 8
Scheduling Tasks for Load Balancing in Software-Defined Cloud Computing Network.

Year	Ref.	Authors	Proposed Algorithm	Improved Parameters	Advantages	Disadvantages
2020	[53]	Ahmed Al-mansoori et al.	Cloud-based framework coupling virtual machines and SDN	Cloud resource allocation algorithm	Enhanced resource allocation to streaming data processing applications	Potential lack of utilization of advanced AI methods
2024	[52]	Sonam Sharma et al.	BMU-COA (Blue Monkey Updated Chimp Optimization Algorithm)	Load balancing, optimization in assigning tasks, reducing migration cost, increasing migration efficiency	Significant improvement in load balancing and optimization, reduced migration cost, increased migration efficiency, better performance than other algorithms	Limited information about the simulation and detailed results of the paper

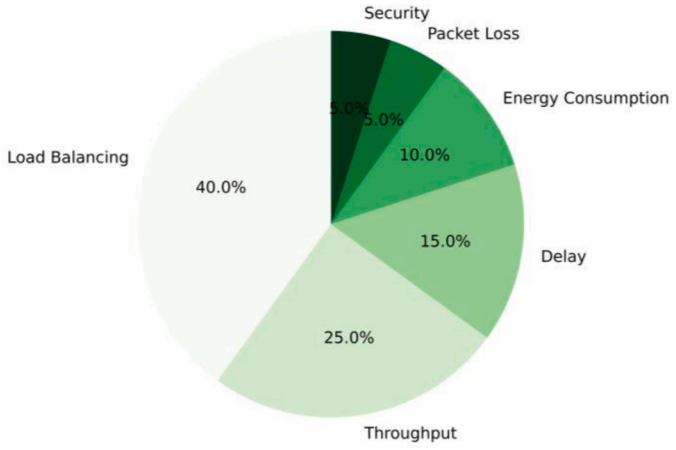

Fig. 7. Estimated Importance of QoS Aspects.

balancing within software-defined networks. Such attacks can disrupt resource allocation and degrade service quality, making it crucial to develop robust strategies that not only optimize performance but also safeguard against security threats. Collectively, these contributions provide valuable insights and solutions for addressing the complexities associated with load balancing and task scheduling in software-defined cloud computing networks, while maintaining a strong focus on enhancing QoS.

Simulation tools

Contrary to real-world deployment, utilizing simulation tools can offer cost-saving benefits and greater management flexibility. Nonetheless, it's essential to note that simulation outcomes are statistically derived based on specific configurations. Therefore, if the settings significantly diverge from the validated environment, the simulation results may not be accurate [54]. Tools like CloudSimSDN and the combined Mininet and POX platform introduced by Teixeira et al. [55], are employed concurrently to assess cloud computing within an SDN framework.

Overview of mininet simulator

Initially conceived by Stanford University professors, Mininet was developed as an educational and research tool focusing on networking technologies. Over time, it has evolved to facilitate the creation of virtual SDN, comprising an open-flow controller [56], an Ethernet network with multiple OpenFlow-enabled switches, and interconnected hosts [57]. Mininet serves as a simulation platform tailored for constructing and emulating SDN networks. It proves beneficial for designing, validating, and assessing network programs and algorithms. Users can configure various networks by integrating components like switches, controllers, servers, and establishing diverse connections. Subsequently, these networks can host and test different programs and algorithms. Leveraging the Python programming language, Mininet facilitates the creation and orchestration of SDN networks, offering comprehensive simulation capabilities. This platform empowers developers and researchers to emulate and evaluate software-driven networks, eliminating the necessity for tangible hardware [58].

Being an open-source platform, Mininet allows users to customize and adapt it to suit their requirements for research and development within the SDN domain. Within Mininet, the SDN controller operates as an application within a virtual machine. This application interfaces with network switches via the OpenFlow protocol, managing and directing network traffic flows [59].

Definition of some SDN controllers

Mininet facilitates the integration of diverse SDN controllers, each offering its distinct set of features and capabilities. Some of the known controllers are briefly:

- POX: A Python-based open-source SDN controller. POX stands out for its adaptability and versatility, making it suitable for various research and developmental endeavors [60].

- Ryu: Another Python-based open-source SDN controller. Ryu is tailored to be lightweight and user-friendly, positioning it as an ideal option for those new to the field [61].
- Floodlight: A mature SDN controller crafted by Big Switch Networks. Floodlight is known for its resilience and scalability, making it apt for managing expansive network infrastructures [62].

Installing a controller in Mininet

1. Select a controller: select for an SDN controller that aligns with your specific needs and criteria.
2. Install the controller: connect to the installation guidelines outlined in the controller's documentation for its virtual deployment.
3. Configure the controller: 3. Controller Configuration: Use the controller's CLI to fine-tune its configurations and parameters.
4. Link switches to the controller: Employ the CLI to establish connections between network switches and the designated controller [63].
5. Manage network traffic: Utilize the controller's CLI or its graphical user interface to oversee and regulate network traffic patterns. By adhering to these guidelines, you can effectively deploy an SDN controller within Mininet, initiating the process of simulating and assessing SDN infrastructures [58].

Overview of CloudSimSDN simulator

CloudSimSDN is a simulation platform specifically designed for Software-Defined Networking (SDN) in cloud environments. It enables the creation and evaluation of virtual network performance over simulated physical infrastructures. Building on the capabilities of CloudSim, CloudSimSDN can emulate the computational components commonly found in data centers, allowing for the simulation and modeling of SDN. The platform incorporates various modules to mimic network traffic patterns and the actions of SDN controllers. By providing centralized control over network switches for precise traffic management, CloudSimSDN offers users tools to improve QoS, energy efficiency, and multi-tenant management within cloud data centers [64].

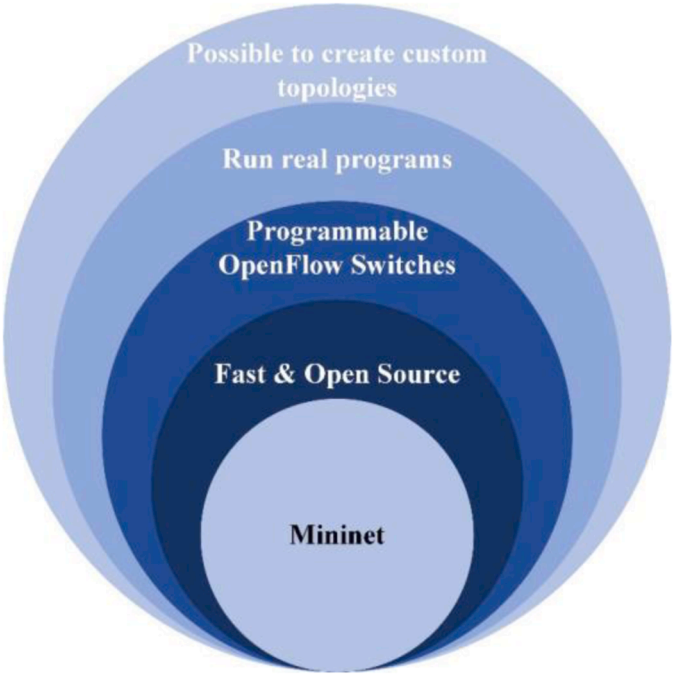

Fig. 8. Mininet features.

Synopsis of the experimental platforms

Both CloudSimSDN and Mininet serve as valuable instruments for simulating and assessing SDN-centric networks, albeit with distinct features catering to varied applications. Here's an overview of their functionalities and uses:

CloudSimSDN

- Acts as a simulation framework tailored for SDN within cloud environments, constructed atop CloudSim.
- Enables the representation and evaluation of virtual network performance over simulated physical infrastructures.
- Utilizes diverse components to replicate network traffic patterns and SDN controller actions.
- Supports the assessment of resource management strategies pertinent to cloud data centers.

Mininet

- Functions as a simulation utility specifically designed for crafting OpenFlow network layouts.
- Operates on the Linux OS, allowing the instantiation of numerous nodes featuring diverse network configurations.
- Emphasizes network resources and facilitates the assessment of traffic management policies under SDN frameworks.
- Has the capability to execute external SDN controllers and Linux-based applications on virtual nodes.

However, CloudSimSDN primarily focuses on centralized network and computational resource management within cloud data centers. On the other hand, Mininet specializes in simulating network architectures, configuring OpenFlow, and assessing SDN controller performance. Therefore, the choice between these tools depends on your specific goals and requirements. If your objective is to simulate and evaluate virtual network performance within cloud environments, CloudSimSDN is the more suitable option. Conversely, if your focus is on designing OpenFlow network structures and testing SDN controller functionalities, Mininet would be the preferred tool.

In this section, simulation tools like Mininet and CloudSimSDN were introduced for evaluating and simulating SDN-based and cloud networks. Mininet serves as a simulation platform enabling the creation and simulation of virtual SDN networks, while CloudSimSDN is used for simulating the performance of virtual networks in cloud environments. These two tools offer various capabilities for traffic management, quality of service improvement, and computational resource management in SDN ad cloud networks.

Future challenges

Future challenges in this domain might encompass:

1. Scalability: With the continuous expansion of cloud computing, it's imperative to ensure algorithms and systems can scale effectively to manage growing workloads and data sizes.
2. Security: Safeguarding data privacy and security within cloud infrastructures remains a pressing concern. Developing resilient security protocols to safeguard confidential data is paramount.
3. Energy Efficiency: Enhancing the energy efficiency of cloud systems to minimize environmental footprint and operational expenses stands as a significant focus for future studies.
4. Dynamic Workloads: Tailoring algorithms to adeptly manage fluctuating workloads and diverse resource requirements in real-time presents a challenge requiring attention for peak efficiency.

5. Interoperability: Boosting interoperability among varied cloud platforms and services is essential to streamline data transfer and resource allocation.
6. Cost Optimization: Identifying strategies to streamline costs linked with cloud services, while upholding superior performance and service quality, is a continuous challenge organization will grapple with.

Addressing these challenges through research is pivotal for the progressive evolution and successful integration of cloud computing technologies across diverse sectors.

Conclusion

Based on the literature reviewed, intelligent algorithms and reinforcement learning techniques are pivotal in enhancing load balancing and optimizing cloud resource utilization. Notably, the Intelligent Rule-Based Metaheuristic Task Scheduling algorithm stands out for its effectiveness in improving service quality within cloud environments. Additionally, hybrid approaches such as MOABCQ, which merges the strengths of the Artificial Bee Colony algorithm with Q-learning, have demonstrated significant improvements in cloud resource performance and efficiency. Consequently, the application of intelligent algorithms and reinforcement learning can lead to optimal outcomes in refining load balancing and boosting cloud resource efficiency. These methods are particularly valuable in enhancing the effectiveness and productivity of cloud resources in complex and demanding scenarios. Moreover, these algorithms not only improve load balancing and resource utilization but also contribute to reducing response times, elevating service quality, and enhancing the overall efficiency and performance of cloud infrastructure. Looking ahead, several challenges must be addressed to further advance the field. These include: Scalability: As cloud environments grow in complexity, ensuring that algorithms can scale effectively to manage increased data volumes and user demands is crucial. Security: With the rise of cyber threats, developing robust security measures within intelligent algorithms is essential to protect sensitive data and maintain system integrity. Interoperability: Future research should focus on ensuring that diverse cloud platforms and services can work seamlessly together, enhancing user experience and resource sharing. Real-time Adaptability: The need for algorithms that can adapt in real-time to changing conditions and workloads will be vital for optimizing performance and resource allocation. By addressing these future challenges, researchers and practitioners can continue to enhance the capabilities of cloud computing and software-defined networks, paving the way for more efficient and resilient systems.

CRediT authorship contribution statement

Masoumeh Mahdizadeh: Writing – original draft, Software, Resources, Methodology. **Ahmadreza Montazerolghaem:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Kamal Jamshidi:** Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Ahmed Hazim Alhilali, Ahmadreza Montazerolghaem, Artificial intelligence based load balancing in SDN: a comprehensive survey, Internet Things 22 (2023) 100814.
[2] S. Imanpour, A. Montazerolghaem and S. Afshari, Load Balancing of Servers in Software-defined Internet of Multimedia Things using the Long Short-Term Memory Prediction Algorithm, 2024 in: Proceedings of the Tenth International Conference on Web Research (ICWR), Tehran, Iran, Islamic Republic of, 2024, 291-296, doi: 10.1109/ICWR61162.2024.10533321.
[3] S. Imanpour, M. Kazemiesfeh and A. Montazerolghaem, Multi-level threshold SDN controller dynamic load balancing, 2024 in: Proceedings of the Eighth International Conference on Smart Cities, Internet of Things and Applications (SCIoT), Mashhad, Iran, Islamic Republic of, 2024, 88-93, doi: 10.1109/SCIoT 62588.2024.10570100.
[4] A. Mohamed, M. Hamdan, S. Khan, A. Abdelaziz, S.F. Babiker, M. Imran, M. N. Marsono, Software-defined networks for resource allocation in cloud computing: a survey, Comput. Netw. 195 (2021) 108151.
[5] A. Montazerolghaem, Efficient Resource Allocation for Multimedia Streaming in Software-Defined Internet of Vehicles, IEEE Trans. Intell. Transp. Syst. 24 (12) (2023) 14718–14731, https://doi.org/10.1109/TITS.2023.3303404.
[6] Azodolmolky, S., Wieder, P., & Yahyapour, R. (2013, June). SDN-based cloud computing networking, in: Proceedings of the Fifteenth international conference on transparent optical networks (ICTON), IEEE, 1-4.
[7] Mikkilineni, R., & Sarathy, V. (2009, June). Cloud computing and the lessons from the past, in: Proceedings of the Eighteenth IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, IEEE, 57-62.
[8] W. Xia, Y. Wen, C.H. Foh, D. Niyato, H. Xie, A survey on software-defined networking, IEEE Commun. Surv. Tutor. 17 (1) (2014) 27–51.
[9] I. Behera, S. Sobhanayak, Task scheduling optimization in heterogeneous cloud computing environments: a hybrid GA-GWO approach, J. Parallel Distrib. Comput. 183 (2024) 104766.
[10] Lakhwani, K., Kaur, R., Kumar, P., & Thakur, M. (2018, August). An extensive survey on data authentication schemes in cloud computing. In 2018 4th International Conference on Computing Sciences (ICCS) (pp. 59-66). IEEE.
[11] Yunli Cheng, A. Vijayaraj, Kiran Sree Pokkuluri, Taybeh Salehnia, Ahmadreza Montazerolghaem, Roqia Rateb, Vehicular fog resource allocation approach for vanets based on deep adaptive reinforcement learning combined with heuristic information, IEEE Access (2024).
[12] A.A. Abbasi, A. Abbasi, S. Shamshirband, A.T. Chronopoulos, V. Persico, A. Pescapè, Software-defined cloud computing: a systematic review on latest trends and developments, IEEE Access 7 (2019) 93294–93314.
[13] B. Kang, H. Choo, An SDN-enhanced load-balancing technique in the cloud system, J. Supercomput. 74 (2018) 5706–5729.
[14] F. Ebadifard, S.M. Babamir, A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment, Concurr. Comput. Pract. Exp. 30 (12) (2018) e4368.
[15] N. Sarrafzade, R. Entezari-Maleki, L. Sousa, A genetic-based approach for service placement in fog computing, J. Supercomput. 78 (8) (2022) 10854–10875.
[16] B. Kruekaew, 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 10 (2022) 17803–17818.
[17] H. Kim, N. Feamster, Improving network management with software defined networking, IEEE Commun. Mag. 51 (2) (2013) 114–119.
[18] S. Alnajdi, M. Dogan, E. Al-Qahtani, A survey on resource allocation in cloud computing. International Journal on Cloud Computing: Services and Architecture (IJCCSA) 6 (2016) 5.
[19] N. Akhter, M. Othman, Energy aware resource allocation of cloud data center: review and open issues, Clust. Comput. 19 (2016) 1163–1182.
[20] A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P.P. Jayaraman, J. Kolodziej, P. Balaji, A. Zomaya, A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems, Computing 98 (2016) 751–774.
[21] X. Wang, L. Zhang, Y. Laili, Y. Liu, F. Li, Z. Chen, C. Zhao, Large-scale hybrid task scheduling in cloud-edge collaborative manufacturing systems with FCRN-assisted random differential evolution. The, Int. J. Adv. Manuf. Technol. 130 (1) (2024) 203–221.
[22] H. Yin, X. Huang, E. Cao, A cloud-edge-based multi-objective task scheduling approach for smart manufacturing lines, J. Grid Comput. 22 (1) (2024) 9.
[23] R. Nithiavathy, S. Janakiraman, M. Deva Priya, Adaptive guided differential evolution-based slime mould algorithm-based efficient multi-objective task scheduling for cloud computing environments, Trans. Emerg. Telecommun. Technol. 35 (1) (2024) e4902.
[24] H. Zavieh, A. Javadpour, A.K. Sangaiah, Efficient task scheduling in cloud networks using ANN for green computing, Int. J. Commun. Syst. (2024).
[25] S.D.S. Mustapha, P. Gupta, Fault aware task scheduling in cloud using min-min and DBSCAN, Internet Things Cyber-Phys. Syst. 4 (2024) 68–76.
[26] C. Barut, G. Yildirim, Y. Tatar, An intelligent and interpretable rule-based metaheuristic approach to task scheduling in cloud systems, Knowl. -Based Syst. 284 (2024) 111241.
[27] S. Iftikhar, M.M.M. Ahmad, S. Tuli, D. Chowdhury, M. Xu, S.S. Gill, S. Uhlig, HunterPlus: AI based energy-efficient task scheduling for cloud–fog computing environments, Internet Things 21 (2023) 100667.
[28] S. Azizi, M. Shojafar, J. Abawajy, R. Buyya, Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: a semi-greedy approach, J. Netw. Comput. Appl. 201 (2022) 103333.
[29] M.I. Khaleel, Region-aware dynamic job scheduling and resource efficiency for load balancing based on adaptive chaotic sparrow search optimization and coalitional game in cloud computing environments, J. Netw. Comput. Appl. 221 (2024) 103788.
[30] S. Simaiya, U.K. Lilhore, Y.K. Sharma, K.B. Rao, V.V.R. Maheswara Rao, A. Baliyan, R. Alroobaea, A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques, Sci. Rep. 14 (1) (2024) 1337.

[31] F. Ebadifard, S.M. Babamir, A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment, Concurr. Comput. Pract. Exp. 30 (12) (2018) e4368.
[32] F. Ramezani Shahidani, A. Ghasemi, A. Toroghi Haghighat, A. Keshavarzi, Task scheduling in edge-fog-cloud architecture: a multi-objective load balancing approach using reinforcement learning algorithm, Computing (2023) 1–23.
[33] A. Asghari, M.K. Sohrabi, F. Yaghmaee, Task scheduling, resource provisioning, and load balancing on scientific workflows using parallel SARSA reinforcement learning agents and genetic algorithm. The, J. Supercomput. 77 (2021) 2800–2828.
[34] Gupta, A., & Garg, R. (2017, September). Load balancing based task scheduling with ACO in cloud computing. In 2017 International conference on computer and applications (ICCA) (pp. 174-179). IEEE.
[35] Wang, T., Liu, Z., Chen, Y., Xu, Y., & Dai, X. (2014, August). Load balancing task scheduling based on genetic algorithm in cloud computing, in: Proceedings of the IEEE Twelfth International Conference on Dependable, Autonomic and Secure Computing , IEEE, 146-152.
[36] P.V. Lahande, P.R. Kaveri, J.R. Saini, K. Kotecha, S. Alfarhood, Reinforcement Learning approach for optimizing cloud resource utilization with load balancing, IEEE Access (2023).
[37] T. Çavdar, Ş. Aymaz, New approach to dynamic load balancing in software-defined network-based data centers, ETRI J. (2023).
[38] Y. Li, B. Wu, Software-defined heterogeneous edge computing network resource scheduling based on reinforcement learning, Appl. Sci. 13 (1) (2022) 426.
[39] B. Song, Y. Chang, X. Zhang, A. Al-Dhelaan, M. Al-Dhelaan, Mixed-flow load-balanced scheduling for software-defined networks in intelligent video surveillance Cloud Data Center, Appl. Sci. 12 (13) (2022) 6475.
[40] M.N. Pathan, M. Muntaha, S. Sharmin, S. Saha, M.A. Uddin, F.N. Nur, S. Aryal, Priority based energy and load aware routing algorithms for SDN enabled data center network, Comput. Netw. 240 (2024) 110166.
[41] B. Kang, H. Choo, An SDN-enhanced load-balancing technique in the cloud system, J. Supercomput. 74 (2018) 5706–5729.
[42] E. Osei Kofi, E. Ahene, Enhanced network load balancing technique for efficient performance in software defined network, Plos One 18 (4) (2023) e0284176.
[43] Q. Burke, P. McDaniel, T. La Porta, M. Yu, T. He, Misreporting attacks against load balancers in software-defined networking, Mob. Netw. Appl. (2024) 1–16.
[44] J. Ali, R.H. Jhaveri, M. Alswailim, B.H. Roh, ESCALB: An effective slave controller allocation-based load balancing scheme for multi-domain SDN-enabled-IoT networks, J. King Saud. Univ. -Comput. Inf. Sci. 35 (6) (2023) 101566.
[45] A. Montazerolghaem, M.H. Yaghmaee, Load-balanced and QoS-aware software-defined Internet of Things. IEEE Internet of Things, Journal 7 (4) (2020) 3323–3337.
[46] A. Montazerolghaem, Software-defined internet of multimedia things: energy-efficient and load-balanced resource management, IEEE Internet Things J. 9 (3) (2022) 2432–2442, https://doi.org/10.1109/JIOT.2021.3095237.
[47] M.S. Siapoush, S. Jamali, A. Badirzadeh, Software-defined networking enabled big data tasks scheduling: a tabu search approach, J. Commun. Netw. 25 (1) (2023) 111–120.
[48] G.D. Singh, V. Tripathi, A. Dumka, R.S. Rathore, M. Bajaj, J. Escorcia-Gutierrez, L. Prokop, A novel framework for capacitated SDN controller placement: balancing latency and reliability with PSO algorithm, Alex. Eng. J. 87 (2024) 77–92.
[49] Sellami, B., Hakiri, A., Yahia, S.B., & Berthou, P. (2020, November). Deep reinforcement learning for energy-efficient task scheduling in SDN-based IoT network, in: Proceedings of the IEEE Ninteenth International Symposium on Network Computing and Applications (NCA), IEEE, 1-4.
[50] B. Sellami, A. Hakiri, S.B. Yahia, P. Berthou, Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network, Comput. Netw. 210 (2022) 108957.
[51] I. Al-hammadi, M. Li, S.M. Islam, E. Al-Mosharea, Collaborative computation offloading for scheduling emergency tasks in SDN-based mobile edge computing networks, Comput. Netw. 238 (2024) 110101.
[52] S. Sharma, D. Seth, Blue monkey updated chimp optimization algorithm for enhanced load balancing model, Expert Syst. Appl. 242 (2024) 122578.
[53] Al-Mansoori, A., Abawajy, J., & Chowdhury, M. (2020, May). BDSP in the cloud: scheduling and load balancing utlizing SDN and CEP, in: Proceedings of the Twentieth IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID) (pp. 827-835). IEEE.
[54] A. Mohamed, M. Hamdan, S. Khan, A. Abdelaziz, S.F. Babiker, M. Imran, M. N. Marsono, Software-defined networks for resource allocation in cloud computing: a survey, Comput. Netw. 195 (2021) 108151.
[55] Teixeira, J., Antichi, G., Adami, D., Del Chiaro, A., Giordano, S., & Santos, A. (2013, October). Datacenter in a box: test your SDN cloud-datacenter controller at home. In 2013 Second European Workshop on Software Defined Networks (pp. 99-104). IEEE.
[56] Bifulco, R., Canonico, R., Brunner, M., Hasselmeyer, P., & Mir, F. (2012, October). A practical experience in designing an openflow controller. In 2012 European Workshop on Software Defined Networking (pp. 61-66). IEEE.
[57] K.K. Sharma, M. Sood, Mininet as a container-based emulator for software defined networks, Int. J. Adv. Res. Comput. Sci. Softw. Eng. 4 (12) (2014).
[58] N. Gupta, M.S. Maashi, S. Tanwar, S. Badotra, M. Aljebreen, S. Bharany, A comparative study of software defined networking controllers using mininet, Electronics 11 (17) (2022) 2715.
[59] Lantz, B., Heller, B., & McKeown, N. (2010, October). A network in a laptop: rapid prototyping for software-defined networks, in: Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks, 1-6.
[60] POX controller. POX website: ⟨www.pox.readthedocs.io/en/latest/⟩. (Accessed26 April 2024) 2024.
[61] Ryu controller. Ryu website: ⟨www.ryu.readthedocs.io/en/latest/. (Accessed26 April 2024) 2024.
[62] Floodlight controller. Floodlight website: ⟨www.floodlight.readthedocs.io/en/latest/⟩. (Accessed26 April 2024) 2024.
[63] Network simulator. Mininet website: ⟨www.mininet.org⟩. (Accessed26 April 2024) 2024.
[64] Son, J., Dastjerdi, A.V., Calheiros, R.N., Ji, X., Yoon, Y., & Buyya, R. (2015, May). Cloudsimsdn: modeling and simulation of software-defined cloud data centers, in: Proceedings of the Fifteenth IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (pp. 475-484). IEEE.