Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062657

ARTICLE

A Comprehensive Study of Resource Provisioning and Optimization in Edge Computing

Sreebha Bhaskaran* and **Supriya Muthuraman**

Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru, 560035, Amrita Vishwa Vidyapeetham, India

*Corresponding Author: Sreebha Bhaskaran. Email: b_sreebha@blr.amrita.edu

Received: 24 December 2024; Accepted: 10 April 2025; Published: 19 May 2025

ABSTRACT: Efficient resource provisioning, allocation, and computation offloading are critical to realizing low-latency, scalable, and energy-efficient applications in cloud, fog, and edge computing. Despite its importance, integrating Software Defined Networks (SDN) for enhancing resource orchestration, task scheduling, and traffic management remains a relatively underexplored area with significant innovation potential. This paper provides a comprehensive review of existing mechanisms, categorizing resource provisioning approaches into static, dynamic, and user-centric models, while examining applications across domains such as IoT, healthcare, and autonomous systems. The survey highlights challenges such as scalability, interoperability, and security in managing dynamic and heterogeneous infrastructures. This exclusive research evaluates how SDN enables adaptive policy-based handling of distributed resources through advanced orchestration processes. Furthermore, proposes future directions, including AI-driven optimization techniques and hybrid orchestration models. By addressing these emerging opportunities, this work serves as a foundational reference for advancing resource management strategies in next-generation cloud, fog, and edge computing ecosystems. This survey concludes that SDN-enabled computing environments find essential guidance in addressing upcoming management opportunities.

KEYWORDS: Cloud computing; edge computing; fog computing; resource provisioning; resource allocation; computation offloading; optimization techniques; software defined network

1 Introduction

Moving from cloud computing to edge computing and fog computing reflects an increasingly overarching requirement for decentralized, near real-time computing capabilities. Cloud computing, which started with firms such as Amazon and Google in the early 2000s [1], revolutionized the way enterprises and people managed their access to computing resources. Centralization of data management in remote data centers enabled scalability, cost efficiency, and on-demand usage of resources. The entry of 5G has significantly leveled up cloud computing and provided a much more powerful, flexible, and responsive service. The requirement for ultra-low latency and high-speed support for real-time applications, such as Augmented Reality (AR), Virtual Reality (VR), and cloud gaming, has emerged as a constraint. As highlighted in [2], the gaming market is rapidly evolving with increasing reliance on cloud-based services for real-time responsiveness. Scalable connectivity brings accessibility to more devices to connect to cloud services, thus further expanding cloud-based capabilities. While cloud computing offers centralized and scalable resources for data processing, rapid data transfer, and reduced latency form feedback loops that drive cloud computing into a new era of low-latency services scalable enough for next-generation applications and devices.

Fog computing [3] evolved as a distributed approach to cloud computing, positioning computational resources near the network edge and end-users. Minimizes latency and network congestion, making it ideal for real-time video analytics, autonomous systems, and industrial automation. Fog computing integrates cloud scalability with edge processing benefits, ensuring low latency and reduced bandwidth usage. Building on fog computing, edge computing further decentralizes resources to edge devices like smartphones and IoT sensors. This method has requirements for latency and bandwidth consumption, thus exploiting real-time data processing and local decision-making. Edge computing [4,5] helps to diminish the reliance on centralized cloud infrastructure to keep pace with the great need for real-time applications, data privacy, and security owing to the multiplication of IoT devices. The integration of cloud, fog, and edge computing into a multi-layered architecture enhances performance, scalability, reliability, and security while reducing network traffic. Towards low latency, high throughput, and flexible resource allocation, reference [6] demonstrates distributing computing resources across layers with improved user experience and system resilience. SDN [7] plays a critical role in resource management in cloud, fog, and edge computing. SDN [8] has decoupled the data and control planes [9] for scalability and security, and also offers centralized management, orchestration, and network virtualization. In edge computing, SDN [10] facilitates real-time device control and automation to allow for efficient integration and operation of an IoT network.

Meanwhile, mobile applications are significantly improved with the introduction of 5G technology which serves as a cornerstone for the advancement of edge and fog computing, offering unprecedented capabilities in terms of high bandwidth, low latency, and massive device connectivity. The platform ensures real-time processing capabilities at distributed locations for time-critical applications. The combination of 5G's high bandwidth power enables smooth transmission of high-definition multimedia content across its ability. This maintains simultaneous connections among millions of devices which promotes the growth of IoT technology. Mobile applications across gaming, healthcare, and smart cities achieve better service reliability and premium user experiences through the combination of reduced latency and enhanced 5G connectivity. Through its innovations, 5G serves as a foundational element that promotes edge and fog computing programs enabling the replacement of centralized cloud computing limitations. This paper builds on these advancements by exploring how resource management strategies can be further leveraged to optimize performance and scalability in dynamic environments.

Edge and fog computing has become essential for handling IoT and data-intensive applications by bringing processing closer to data sources. Performance can be improved by task offloading [11] to cloud, fog, or edge layers, but such an approach also introduces challenges like resource provisioning, system management, and security [12]. In decentralized environments, efficient resource allocation [13,14] needs to be addressed for scalability, energy efficiency, heterogeneity, and mobility, and to be reliable enough for applications in smart cities and AR. Moreover, distributed resources make managing, observing, orchestrating, and enforcing security more complex.

The study of cloud, fog, and edge computing is motivated by these paradigms' ability to transform data processing for IoT and real-time applications. Each paradigm has specific resource provisioning challenges: cloud computing is inherently scalable, whereas fog and edge environments require low latency processing. It is also motivated by the demand for efficient computation offloading and dynamic network management using SDN for resource optimality. This survey covers the paradigms of resource provisioning, computation offloading, and integration with SDN, creating trends, research gaps, and future directions for multi-layered systems.

This paper explores resource provisioning, allocation, and computation offloading as key factors in enhancing performance, scalability, and resource efficiency in edge computing environments. Resource provisioning [15] involves estimating and allocating needed resources to support maximum estimated resource consumption; whereas resource allocation attempts to allocate these resources like Central Processing Unit (CPU), memory, bandwidth, or storage between users, such that the edge servers retain optimal load distribution for Quality of Service (QoS) constraint. They enable the reduction of latency, large-scale deployment, reliability, and QoS through efficient resource management and smart response to dynamic requirements. This paper presents a systematic review of techniques and mechanisms in cloud, fog, and edge computing, analyzing their advantages and identifying research gaps in existing literature. The systematic review, conducted following the PRISMA 2020 guidelines, is illustrated in Fig. 1a, while the search process is carried out according to the research questions shown in Fig. 1b. Additionally, the key contributions of this study are summarized in Fig. 1c.

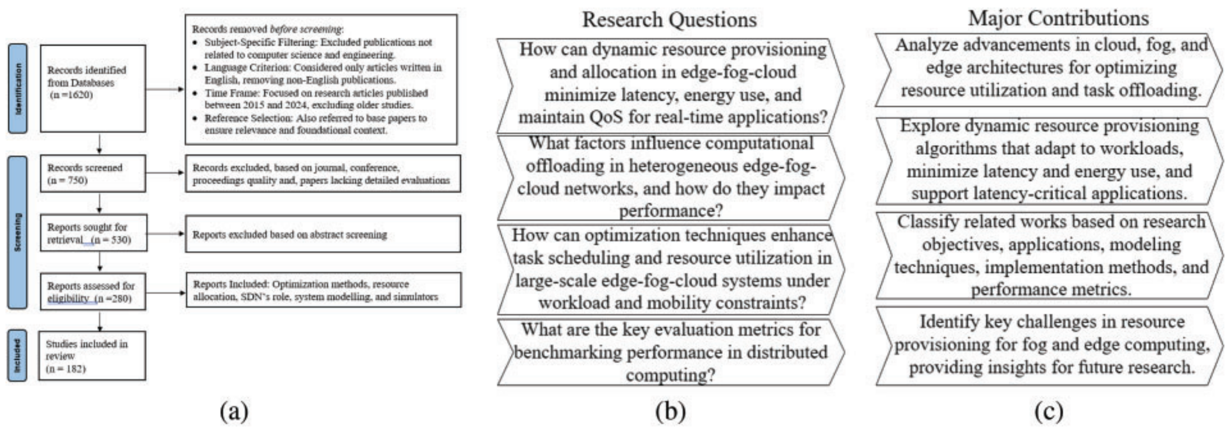


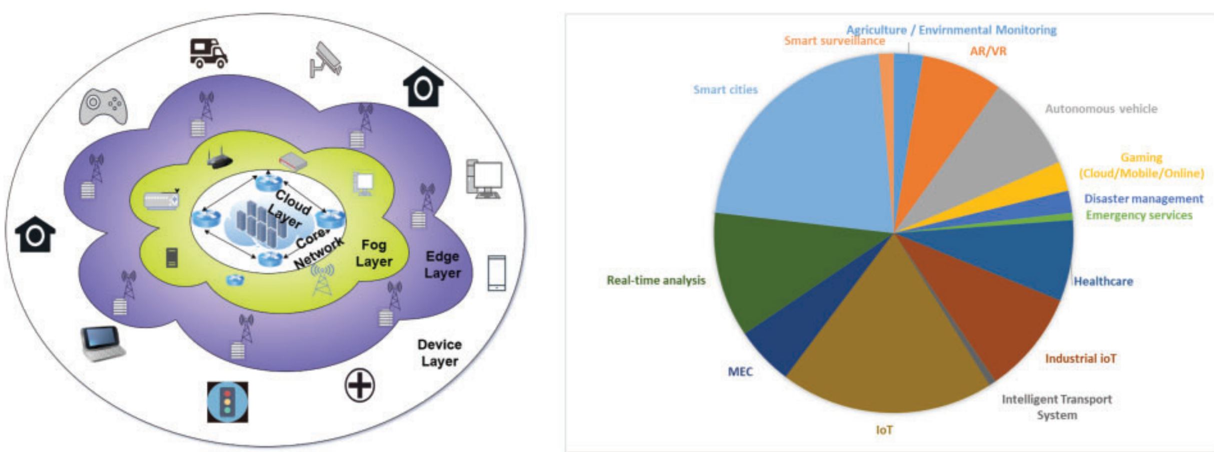
Figure 1: Identification of studies: **(a)** PRISMA_2020 Flow diagram **(b)** Research questions **(c)** Major contributions of this work

This paper is organized to comprehensively review research on resource provisioning in cloud, edge, and fog computing. Section 2 introduces the architecture of cloud, edge, and fog computing; this section serves as a foundation. Resource provisioning is discussed in Section 3 with its different types and the research mechanisms used. Computational offloading strategies are discussed in Section 4. Section 5 details resource allocation techniques, followed by efficient resource provisioning in edge computing based on time synchronization in Section 6, while Section 7 presents the framework of the system model. Section 8 discusses evaluation metrics, followed by the simulators used for testing and validation in Section 9; Section 10 demonstrates the limitations of current methods for resource provisioning and provides useful insights into the challenges posed by resource provisioning. The paper concludes by providing the potential advancements and research directions in this domain.

2 Architecture

In the context of cloud, fog, and edge computing with SDN and smart devices, the integrated architecture defines a layered framework based on data processing, resource allocation, and real-time network control in diverse applications. This architecture, ensures efficient resource utilization, low latency processing, and agile network configuration so that scalable and responsive services can be offered across diverse applications. Fig. 2a presents the structure of this layered architecture:

- Cloud Layer: This layer provides centralized computational resources, storage, and analytics catering to large datasets that require complex analysis. While centralized, the cloud layer may lack the low latency needed for modern applications.
- Fog Layer: This layer, positioned between the cloud and the edge, brings processing closer to users through network gateways or local servers. Pre-processing, filtering, and aggregation of data is done in the fog nodes to minimize latency and lower cloud traffic, which in turn facilitates regionalized real-time applications.
- Edge Layer: This layer processes data at its sources, such as IoT sensors, smart devices, or inside the autonomous vehicle or smart healthcare system, thereby delivering an ultra-low latency experience for the critical application. It is also useful for reducing the bandwidth costs on expensive network segments, by small incremental data transfers.
- SDN: By separating network control from hardware, SDN decouples the management of the network from that of the corresponding hardware, which leads to centralized control, dynamic resource allocation, and optimized data flow across the Cloud, Fog, and Edge layers. This approach provides scalability, flexibility, and load balancing, which are key issues for adaptive resource provisioning in distributed systems.
- Smart Devices Layer: IoT sensors and mobile devices that generate data and facilitate localized processing together constitute the outermost layer. However, most of these devices tend to offload computational tasks to edge, fog, or cloud resources, especially in applications that consume higher resources.



(a) (b)

Figure 2: **(a)** Architecture of cloud, fog and edge computing. **(b)** Applications across cloud, fog, and edge computing

3 Resource Provisioning

Resource provisioning [16] refers to the strategic allocation of essential resources such as CPU, memory, storage, and network bandwidth to fulfill user requirements. This involves estimating resources, deploying them, and adjusting capacity in response to changing workloads, ensuring that the infrastructure is prepared to meet anticipated demands. In the realm of cloud computing, resource provisioning consolidates resources within data centers, facilitating flexible scaling, cost efficiency, and high availability. At the edge, it places resources closer to users, helping to minimize latency, and supports dynamic applications such as IoT and video streaming. Fog computing [17] takes a distributed approach, spreading resources across local and intermediary devices to balance the load between the cloud and edge layers. This model enables processing near the data source, alleviates network congestion, and improves responsiveness. Effective resource provisioning is crucial for maintaining scalable, efficient, and reliable operations across cloud, edge, and fog environments. Fig. 2b showcases various applications, together addressing diverse requirements by optimizing performance. Fig. 3a–c showcases the challenges faced in the computing paradigm, the need for resource provisioning to enhance the performance, decrease latency and the types of resource provisioning available in the literature.

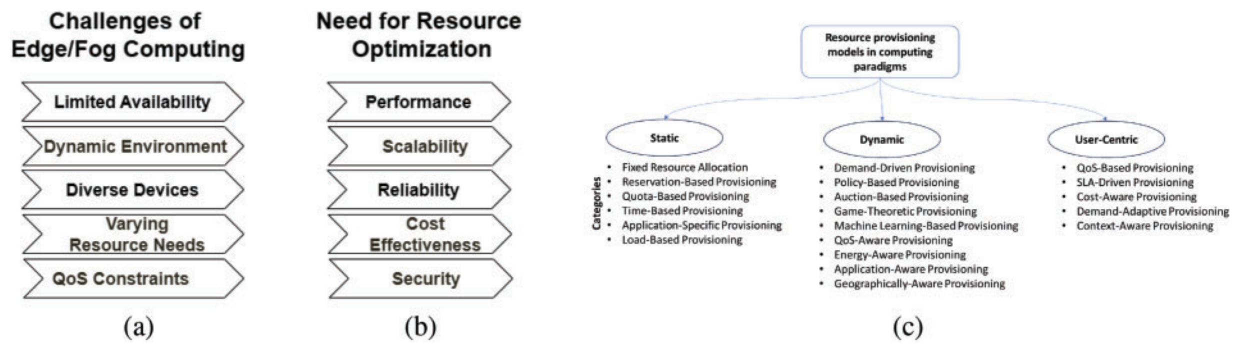


Figure 3: Resource provisioning: **(a)** Challenges of Edge/Fog computing; **(b)** Need for resource provisioning; **(c)** types of resource provisioning

In the cloud computing framework [18], resource provisioning is central to helping the cloud providers scale their resources with high load while reducing costs and maintaining service continuity. It enables edge and fog computing [19], to reduce latency, minimize network congestion, and optimize resource utilization for time-critical applications [20] like IoT, real-time analytics, and AR/VR, ensuring responsive and reliable services. The investigations recognize the multiplicity of actual environments due to different infrastructure resources as well as shifting service demands and unexpected device failures.

Effective resource provisioning in cloud, edge, and fog computing requires tailored strategies for optimizing resource allocation across these architectures such as dynamic, demand-based provisioning in cloud computing [21] using virtualization and auto-scaling tools like Kubernetes and containers. Edge and fog computing have been used to optimize resource utilization by minimizing proximity-based provisioning, distributed frameworks, and techniques like load balancing, context-aware allocation, and scheduling to decrease latency and increase efficiency. Machine learning [22] is used to predict future demands, while SDN dynamically adjusts network configurations to minimize resource inefficiencies, increase responsiveness, and lower costs, while meeting the architectural specific needs.

In [23], key challenges in fog computing, including dynamic workloads, device heterogeneity, latency, and security are presented, as well as emerging trends of machine learning, blockchain, federated learning, and IoT-5G integration. The study in [24] examines dynamic resource provisioning for Cyber-Physical Systems (CPS) within cloud, fog, and edge environments to enable real-time data processing with minimal latency. There have also been works that try to address IoT device provisioning to reduce cloud dependence and latency, for instance, reference [25] explored IoT device provisioning through edge gateways to provide services for the IoT extension agents on the Microsoft Azure IoT and IBM IoT platforms. The study in [15] examines resource allocation in multi-agent cloud robotics, focusing on challenges in latency-sensitive and data-intensive tasks across Industry 4.0, agriculture, healthcare, and disaster management. It reviews existing issues, categorizes techniques such as offloading and scheduling, and highlights research gaps for future exploration. To meet IoT or real-time application edge computing challenges, reference [14] studied resource scheduling optimization, task offloading, and cloud edge coordination, as well as self-adaptiveness, to suggest Artificial Intelligence (AI) driven approaches to solving dynamic environments. More studies must explore the scalability, robustness, and efficiency of these algorithms as they apply to real-world implementation within operational settings.

3.1 Types of Resource Provisioning

In computing paradigms, resource provisioning mechanisms are categorized into user-centric, dynamic, and static models, as illustrated in Fig. 3c. User-centric provisioning provides resources such as virtual machines on a per-user request basis with the risk of having high costs and under-utilization [26]. Dynamic provisioning decides to accommodate resources according to unexpected workloads, thus continuous monitoring is necessary to prevent under or over-provisioning inefficiency [27]. On the other hand, static provisioning reserves predefined resources for urgent tasks, but can result in wastage when not in use [28].

Static provisioning can be used for applications with stable demands such as in low loads environments, e.g., smart buildings, and homes. Users send tasks or IoT data to a provisioning agent, which, using information from the resource information center, can allocate appropriate resources. Categories include: Application-based Provisioning where resources are allocated to respective services based on the SLAs, Fixed Resource Allocation wherein resources are assigned without workload adjustment, Time Based Provisioning that allocates resources to peak demand periods, Quota Provisioning to limit resource usage to prevent utilization, Reservation Provisioning for ensuring availability during peak usage via reservations and Load Based Provisioning wherein resources are statically allocated based on predicted trends as opposed to their real-time fluctuations. Recent research mentioned in Table 1 on static provisioning focuses on application-based or load-based approaches.

Table 1: Papers based on static resource provisioning

Paper	Problem addressed	Layers	Metrics
[29]	Resource constraints of mobile devices and high latency of cloud computing for resource-intensive applications	Cloud	Response time, VM synthesis time, and cloudlet setup time
[28]	To support mobile edge provisioning, particularly regarding competitive resource bidding and user mobility	Cloud, Edge, Mobile users	Latency, VM utilization, allocation time, and provisioning cost
[30]	Resolves to optimize resource allocation and workload balancing in fog-cloud networks with QoS compliance	Cloud, Fog	Latency, execution cost, energy consumption, and execution time
[31]	Framework to manage edge nodes efficiently by incorporating provisioning and auto-scaling mechanisms	Cloud, Edge, Mobile users	Latency
[32]	Presents the use of edge and fog computing in IoT for supporting intelligent, interoperable services	Cloud, Fog, Edge, Things	Energy consumed, integration and interoperability

Dynamic provisioning guarantees real-time resource allocation to satisfy the current workload demand allowing for higher responsiveness than static methods [33]. Provisioning methods include demand-driven methods, which involve predictive models of resources along with reactive prediction of sudden spikes, and policies-based provisioning which allocates the resources based on certain rules. Provisioning is done via market mechanisms (auction-based) and game theoretic models (incentivize efficient distribution). Adaptive allocation is provided by machine learning through predictive analytics, reinforcement learning, and deep learning. Energy-aware provisioning is used to minimize the resources at non-demand times to minimize energy usage, and QoS-aware provisioning satisfies metrics such as latency and throughput for SLA. Application-aware provisioning is based on specific contexts (user behavior or location) to support the IoT and smart cities, while content-aware provisioning responds to requirements such as media storage or AI processing. Provisioning in a geographically aware manner mitigates latency by locating the resources closer to users for critical applications like autonomous vehicles and remote healthcare. Together these strategies enhance resource efficiency, scalability, and performance. To address the problem of edge node placement, a framework, called EdgeON [34], is proposed. A review of AI techniques for resource management optimization in fog computing using task scheduling, resource allocation, load balancing, and energy efficiency while considering scalability, security, and real-world validation was conducted in [35]. The goal is to minimize the deployment and operation costs to maximize the utilization of the network resources. Dynamic resource provisioning allows systems to adapt to the changing workloads by allocating, and deallocating resources dynamically, maximizing cost-effectiveness, and providing better end-user experience as discussed in Table 2.

Table 2: Papers based on dynamic resource provisioning

Paper	Problem addressed	Algorithms	Type	Layers	Metrics
[36]	Optimizing application placement in fog computing system	Integer Linear Programming	Application-based provisioning	Cloud-Fog	Network Relaxation Ratio, Resource Gain, and Processing Time Reduction Ratio
[37]	To minimize fog computing resource provisioning costs for multiple users	Bipartite graph matching along with optimal greedy and approximation algorithm are utilized	Geographically aware Provisioning	Cloud-Fog	Cost minimization (replication and transmission cost)
[38]	Reduces data transfer delay while still allowing efficient task execution for data-intensive applications	Integrates four data placement algorithms and three task scheduling strategies	Context-aware provisioning	Edge	Task turnaround time, queuing time, execution time, and data transfer overhead
[39]	To maximize service provider profits while ensuring tasks meet their deadlines	0-1 knapsack problem and Ant Colony Optimization algorithm	QoS-Aware Provisioning	Cloud-Fog	Profit and proportion of tasks that meet deadlines
[40]	Resource allocation and task offloading in MEC[1] systems for urban rail transit system is challenging	Task Classification Twin Delayed Deep Deterministic Policy Gradient algorithm	Machine Learning-Based, Application-Aware, Geographically Aware	MEC[1]-rail devices	Task completion rate and task processing delay
[41]	Balancing energy efficiency and performance in fog computing for latency-sensitive applications is challenging	Prediction-based dynamic resource allocation algorithm using the ARIMA[2] model	Demand-Driven and Energy-Aware Provisioning	Cloud-Edge-user equipment	Energy consumption and task delay

(Continued)

Table 2 (continued)

Paper	Problem addressed	Algorithms	Type	Layers	Metrics
[42]	Resource constraints can create edge servers with higher latencies than cloud servers, a phenomenon we refer to as edge performance inversion	Queueing theory and optimization techniques	QoS-Aware, Demand-Driven provisioning	Cloud-Edge	Response time and resource costs
[43]	Focuses on application of dynamic resource overbooking and container scheduling to maximize utilization	Actor-critic reinforcement learning approach	Machine Learning, Demand-Driven, QoS, and Geographically Aware provisioning	Edge	Latency, computation cost, and container eviction cost

Note: [1]MEC: Multi-Access Edge Computing, [2]ARIMA: Autoregressive Integrated Moving Average.

Table 3: Papers based on user-centric resource provisioning

Paper	Problem addressed	Algorithms	Type	Layers	Metrics
[44]	Optimizing resource management in MEC environments to support IoT applications in smart cities	Federated Learning-based Deep Deterministic Policy Gradient algorithm and Lyapunov optimization	QoS-Aware, Application-Aware provisioning	Cloud-MEC-IoT devices	Energy consumption, convergence efficiency, and reward value
[45]	Transition from a network function-centric to a user-centric architecture in 6G networks	Machine learning models to dynamically optimize the placement of network functions	Context-aware provisioning	Cloud-Edge-User	Round Trip Time
[46]	Efficient resource allocation in edge-computing-enabled environments for the metaverse	Reviewed 19 algorithms on resource allocation strategies, for offloading, caching, and distributed resource scheduling	Demand-Driven provisioning	Cloud-Fog-Edge	Latency, energy consumption, throughput, and QoE
[22]	Resource management and task scheduling to balance QoS and energy efficiencies	ML models like DRL, Q-learning, and hybrid methods (e.g., PSO with SVR)	SLA-Driven, Context-Aware Provisioning	Cloud-Fog	Latency, cost, energy consumption, scalability, and QoS
[47]	Dynamically allocating resources and optimizing AP clusters based on user needs and network conditions	Proximal and multi-agent proximal policy optimization	Context-Aware Provisioning	Edge (MEC server)-User	Throughput, total delay and execution time
[48]	Resource allocation in edge-assisted Mobile Augmented Reality in 6G networks	Digital Twin based approach, Markov decision process, machine learning techniques	SLA-Driven and Context-Aware provisioning	Edge-Mobile Users	Resource utilization, delay, traffic prediction accuracy

User-centric provisioning adapts resource allocation to user needs such as QoS, latency, and cost, the details of which are presented in Table 3. QoS Based Provisioning guarantees performance metrics such as latency and throughput, crucial for real-time application examples including AR/VR gaming, and video conferencing, by allocating resources closer to end users. Provisions are based on the Service-Level Agreement (SLA), and resources are based on predefined SLA metrics such as uptime and response time. Cost-aware provisioning tries to minimize costs, which is suitable for pay-as-you-go models in cloud computing. Provisioning based on demand-constrained usage with IoT and smart cities has proven advantageous due to the dynamic environment. Mobile gaming applications like reducing latency exhibit the need for Context-Aware Provisioning to adjust resources based on user location.

3.2 Mechanisms for Resource Provisioning

This section categorizes and evaluates resource-provisioning approaches in edge, fog, or cloud systems while focusing on modern techniques that optimize resource availability and utilization. Based on the state-of-the-art research work, resource provisioning mechanisms are categorized as shown in Fig. 4. The papers focusing on resource management mechanisms are summarized in Table 4 based on problems solved, algorithms used, evaluation metrics, and potential paths for future work.

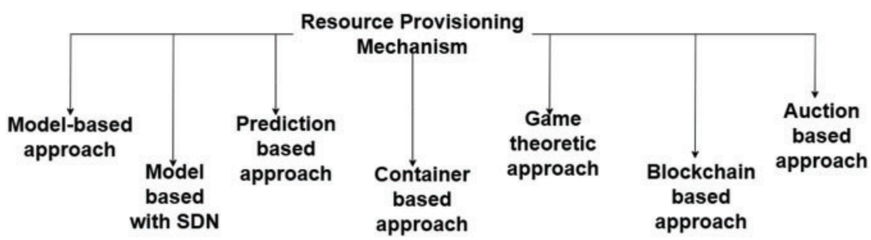


Figure 4: Categorization of resource provisioning mechanisms

3.2.1 Model-Based Approach

The following studies discuss advancements and challenges in resource provisioning in cloud, edge, and fog computing paradigms. Ma et al. [49] minimize cost under QoS in a cloud-assisted Multi-Access Edge Computing (MEC) framework but does not focus on real-time load balancing. The work in [50] minimizes deployment costs and workload allocation, but does not tackle dynamic scaling. Ascigil et al. [51] describe a distributed resource management system for Function as a Service (FaaS) edge cloud network, which aims to reduce latency without any dynamic workloads. Dynamic IoT environments are addressed by the Dynamic Multi Resource Management algorithm introduced in [52]. Zhou et al. [53] present a fixed-cost resource provisioning framework by Lyapunov optimization that is proven to be feasible but emphasizes issues with fixed contracts and cooperation. Resource adaptability is improved in [54], particularly through the use of the delay-aware Lyapunov optimization technique. Latency and resource consumption issues are addressed in a heuristic algorithm developed in [55] for scalability to a wider range of bandwidths and models.

To effectively optimize resource allocation and minimize energy consumption, reference [56] proposes a new distributed MEC architecture that integrates, the cloud, metro fog node, and the vehicle seamlessly. The results show that once the processing demands of a workload remain unchanged, traffic volume is a more important factor in power consumption. reference [57] addresses latency optimization through tabu search for heterogeneous IoT environment. In a UAV network environment, reference [58] suggests an energy-efficient trading mechanism relying on bilateral negotiation and convex optimization respectively. Unlike others [59,60] that focus on mobility-aware Virtual Network Function (VNF) placement in a multi-user MEC environment, the resource and security challenge is formulated as an optimization problem and its solution is formulated to achieve fairness in VNF placement from an economic perspective. Resource sharing complexities impose difficulties for the VNF reliability optimization using Integer Linear Programming (ILP) and dynamic programming in [61].

Chang et al. [62] propose additional work on their dynamic scenarios, and the resources are balanced based on CPU frequency and energy efficiency. Realizing the reliability and ultra-low latency properties in MEC, Toka et al. in [63] use a heuristic algorithm to solve it, but ignore the issues of multi-node failures.

Table 4: Papers based on resource provisioning

Type	Paper	Problem addressed	Algorithms	Metrics	Future scope
Model-based approach	[55]	In MEC, elastic service provisioning, reduces latency and resource usage	Greedy and Heuristic algorithm	Data transmission time, network resource consumption	Context-aware optimization for better provisioning
	[61]	Reliability-aware VNF service provisioning in MEC	Approximation algorithm, ILP, dynamic programming	Network throughput,service blocking probability, computational cost	Real-time optimization for VNF provisioning
Prediction-based model	[64]	Resource provisioning for CPS Systems in cloud-fog-edge computing	Fuzzy C-Means Algorithm for clustering and FPA for optimization	Accuracy, user satisfaction, and resource utilization	Extend the model for heterogeneous CPSS, explore real-time optimization, and mobility issues
	[65]	Inefficient resource provisioning under uncertain fog computing conditions	Double Q-learning combined with State-Action-Reward to reduce overestimation errors	Learning rate, execution time, accuracy, and resource utilization rate	Extend to cloud, cluster computing, and integrate deep learning
Container-based model	[19]	Resource provisioning for latency-sensitive applications in edge computing	Lyapunov optimization for dynamic resource allocation and management. Docker containers for orchestration	Latency, CPU usage, storage, memory consumption, queue size	Scalability, and real-time adaptability in large deployments
	[66]	Dynamic resource provisioning for containerized edge computing	Container scheduling, resource allocation heuristics, machine learning models	Latency, throughput, resource utilization, container startup time, energy efficiency	Real-time adaptability across large-scale edge deployments
Game theoretic	[67]	Joint pricing and resource provisioning in MEC	Game Theory, Optimization Techniques	Revenue, latency, user satisfaction	Heterogeneity, energy efficiency
	[68]	Resource provisioning for vehicle fleets in MEC, balancing QoS	Stochastic traffic analysis with a two-phase algorithm bracketing, binary search	Provisioning cost, QoS, service blocking probability, stability	Further testing on different mobility patterns
Blockchain-based approach	[69]	Resource allocation and pricing for mobile blockchain in cloud-edge environments	Stackelberg game model with iterative greedy-and-search approach	Revenue, resource utilization, task delay, communication cost, user satisfaction.	Multi-server coordination, scalability, and machine learning optimization
	[70]	Examines how BIFL[1] improves data privacy, scalability, and efficiency for IoT healthcare between distributed systems	Surveys BIFL[1] approaches, which combine federated learning	–	Lightweight blockchain frameworks and advanced federated learning tools for decentralized IoT healthcare devices

Note: [1] BIFL: Blockchain Integrated Federated Learning.

As a limitation, reference [71] demonstrates dynamic provisioning for IoT services via fog computing with Raspberry Pi devices. A low latency resource provisioning system for smart cities is presented in [72], though real-world scalability is unexamined. Ma et al. [73] provide heuristic and ILP-based algorithms to optimize resource trade-offs for VNFs in MEC, but the solutions are restricted to dynamic environments and pricing policies.

3.2.2 Model-Based Approach with SDN

The studies highlight the advancements and challenges in dynamic resource provisioning across SDN, edge, and cloud environments. Consequently, according to [74], workflow scheduling in SDN-enabled edge computing is formulated as a multi-objective optimization problem, where the Nondominated Sorting Genetic Algorithm (NSGA)-III performance is achieved along with task assignment strategies such as First Fit and Worst Fit. Qu et al. [75] tackle resilient service provisioning in MEC using a max-min optimization approach and two-stage greedy algorithms, thereby achieving improved utility and resource allocation efficiency. In an SDN-based IoT application, reference [76] presented a Controlled Service Scheduling Scheme (CS3) using predictive power management and deep recurrent machine learning for efficiency in terms of power and latency and its applicability to the real world.

An SDN and NFV orchestrated framework for the Industrial Internet of Things (IIoT) is proposed in [77], which increases scalability, resource utilization, and latency with the caveat of missing out of security and dynamic adaptation to load dynamics. Working at the architectural level, Wang et al. [78] summarize the challenges in integrating cloud, edge, and fog for connected vehicles, focusing on latency-sensitive applications like autonomous driving. In real-time IoT applications, reference [79] proposes hybrid Reinforcement Learning (RL) and Deep RL (DRL) approaches for large-scale dynamic networks and explores such RL and DRL techniques for fog computing to work practically for task scheduling. In a distributed serverless cloud-to-thing model for 5G/6G networks, reference [80] uses SDN and Named Data Networking (NDN) to dynamically deploy WebAssembly modules across devices such as drones and satellites, and NS3 simulations validate low latency and scalability.

3.2.3 Prediction-Based Approach

The following studies showcase recent progress in dynamic resource provisioning over edge, fog, and cloud environments addressing predictive techniques. Deep reinforcement learning-based resource allocation and task offloading for edge nodes are used in [81]. Similar to the work of [82], deep reinforcement learning is used to handle time-varying workloads in fog networks for resource allocation for IoT applications. The work of [24] builds cloud, fog, and edge computing for CPS combined with Dynamic Social Structure of Things (DSSoT), privacy computing, and trust management to handle data problems. Porkodi et al. [64] present Fuzzy C-Means with Flower Pollination Algorithm (FCM-FPA), a resource provisioning and clustering algorithm in highly dynamic cyber-physical-social systems environment.

A reinforcement learning model for mobile edge networks in [83] addresses the real-time dynamic resource provisioning challenge. Shezi et al. [84] present a comprehensive resource allocation strategy for 5G/6G wireless networks and the simulated results show large energy savings. Adaptive demand and topology-aware resource provisioning protocol, a demand-aware resource provisioning protocol with real-time elasticity and allocation optimization, is presented by [85] using deep reinforcement learning. The scalability and performance of traditional approaches are outperformed by the double state temporal difference learning framework for fog computing as presented in [65].

To enhance the efficiency and stability of IoT task provisioning in federated edge computing, Baghban et al. propose [86] a DRL-based Dispatcher Module, running in batch with an Actor-Critic algorithm. Gradient descent and cluster-based provisioning for cooperative systems are proposed by Alsurdeh et al. [87]

for a hybrid workflow scheduling system in edge cloud environments for latency-sensitive tasks. Monte Carlo simulations in Liwang et al. [88] indicate that an overbooking-enabled trading mechanism maximizes task completion rates and energy efficiency at the expense of increased resource risk. Huang et al. [89] propose a profit maximizing edge computing architecture based on cloudlets using Benders decomposition for offline scenarios and a mixed integer programming problem. In MEC, reference [90] employs a risk-based optimization approach incorporating stochastic programming and Sample Average Approximation (SAA) to optimize communication costs as well as server overload.

3.2.4 Container-Based Approach

Hu et al. [66] propose an efficient containerized edge computing (CEC) framework that addresses the need for efficient resource pre-provisioning and the prediction of the latency for container startups to enable efficient resource utilization while minimizing container start times. ElasticFog [91] describes a dynamic resource allocation algorithm that maintains dynamic allocation to resources according to real-time resource usage monitoring and allocation. Santos et al. [92] combine theoretical fog computing resource provisioning concepts with actual applications in IoT and smart city services, pushing for more efficient allocation under dynamic demands. The resource allocation techniques for edge devices in dynamic environments, including reducing latency, increasing performance, and enhancing utilization, are studied in [93]. Zhu et al. [94] present a Cyber-Physical-Human System (CPHS), where cross-layer hybrid resources are combined with orchestration tools and scheduling mechanisms to address complex CPHS scheduling problems.

3.2.5 Game Theoretic Approach

Stackelberg game model for resource provisioning and pricing in delay-aware MEC environments is presented in [67]. Zakarya et al. [95] introduces epcAware, a noncooperative game model for the management of resources in MECs, to achieve energy efficiency and cost reduction with no performance loss. Liu et al. [96] propose a pricing mechanism for allocating limited MEC resources according to smart mobile devices budget to maximize resource utilization, reduce latency, and maximize provider profits. QoS-guaranteed resource provisioning for vehicle fleets is addressed in [68] using a stochastic traffic model and a two-phase algorithm (bracketing and binary search), validated on real-world taxi datasets by achieving minimum costs.

3.2.6 Blockchain-Based Approach

Blockchain enables trusted, decentralized resource provisioning and task management in device-to-device (D2D) edge computing for the gap in efficient task allocation and trust. Sharma et al. [97] combine Proof of Reputation (PoR) with a CDB-based resource market auction algorithm in D2D ECN settings. Fan et al. [69] adopt a Stackelberg resource allocation and pricing strategy for mobile blockchain services based on an iterative greedy-and-search method. Using smart contracts, reference [98] suggest a 5G architecture incorporating network slicing, blockchain as well as MEC to support trust and the correct resource allocation of autonomous systems. Rajagopal et al. [70] investigate blockchain and federated learning (BIFL) for IoT-based healthcare data handling among edge, fog, and cloud systems, and the key issues of addressing scalability, privacy, and communication overhead; and propose lightweight blockchain frameworks and learning algorithms to support future healthcare applications.

3.2.7 Auction-Based Approach

Dynamic resource sharing among the cloudlets is proposed through an incentive-based auction scheme to provide efficient edge computing provisioning in [99], ILP, and a greedy algorithm are used to distribute tasks efficiently and share resources. In MEC, an incentive mechanism [100] enables edge clouds to participate in a profit-maximizing multi-round auction to provide resources under the conditions of fairness and dynamism resource assignments. Reverse auctioning is proposed in [101], where users bid for resources aiming for dynamic allocation in a way that achieves fairness and cost efficiency for applications like e-commerce and data analytics, followed by future extensions of real-time bidding and hybrid cloud integration. To address dynamic cloud provisioning through online auction frameworks, reference [102] developed primal-dual algorithms for VM allocation that optimize social welfare while achieving truthful bidding, and demonstrate the scalability gap in real time of a Software as a Service (SaaS) platform. Auction-based mechanisms for cloud and edge computing have been reviewed by [103] with regards to categories such as game theoretic and machine learning augmented auctions, suggesting the lack of real-time adaptability, proposing blockchain-integrated and federated learning solutions to overcome these shortcomings. The auction mechanisms for the resource allocation in [104] suggest the public blockchain networks, that is, borrowing from the truthful bidding and utility maximization while proposing predictive bidding, and hybrid cloud-fog integration for a better rent. Liu et al. [96] use such auction mechanisms and game theory to control resource allocation in MEC environments to achieve fairness and efficiency, identifying the necessity for scalable and real-world dynamic pricing models. Different resource provisioning mechanisms are used in cloud, fog, and edge computing to tackle specific challenges including latency, energy efficiency, and scalability.

3.2.8 Insights from Reviewed Literature

Despite significant advancements in resource provisioning for edge, fog, and cloud environments, several critical research gaps remain. Current approaches struggle to smoothly integrate the multi-tier architecture which limits their capability to distribute resources between cloud, fog, and edge systems. Both dynamic provisioning models face challenges sustaining real-time mobility along context-specific requirements because they commonly sacrifice performance to achieve energy efficiency. Existing frameworks can be improved to provide reliable solutions and QoS specifications, especially within time-sensitive applications such as healthcare IoT and autonomous vehicles. Several research studies overlook the inevitable uncertainties that emerge from resource requirements and mobile workload patterns typical of edge computing systems. The domain-specific requirements of smart city, AR/VR, and blockchain applications can have a standard benchmark to compare proposed solutions. Dynamic pricing models further highlight the need for improvement. Addressing these gaps is crucial for next-generation computing systems.

4 Computation Offloading

Computational offloading [105] refers to the transferring of resource-intensive tasks from constraint devices like smartphones or IoT sensors to powerful edge, cloud servers. Furthermore, it reduces the local resource demands, optimizes energy usage, and extends the battery life, while enabling efficient use of the centralized computing resources provided to multiple devices. Yet network latency, security, and intelligent decision-making on task offloading and execution location continue to pose challenges. Table 5 highlights the future directions to improve efficiency and integration by using intelligent algorithms and resource management to address scalability and real-time processing challenges.

In this survey, we explore computation offloading about latency, energy efficiency, scalability, and resource optimization over a hybrid cloud, fog, and edge environments. Wang et al. [106] present the edge AI and serverless computing schemes for offloading to guarantee efficient provisioning for migrating hybrid edge cloud systems. A dynamic offloading model presented in [107] optimizes latency-sensitive IoT and autonomous systems, and shows how energy efficiency and scalability are lagging. Kar et al. [108] present an in-depth review of offloading techniques both traditional and from the set of machine learning methods, highlighting the scalability and security gaps in available and practical frameworks. Multi-user task scheduling is integrated using stochastic geometry and queueing theory tradeoff of resource utilization without scalability in [109]. Deadline-aware task placement in hierarchical fog networks proposed in [110] yields better task completion rates, with a focus on working on scalability and energy efficiency in the future.

Table 5: Papers based on computation offloading

Paper	Problem addressed	Algorithms	Metrics	Future scope
[106]	Hybrid edge-cloud computing for efficient resource management and reduced latency	Heuristic approach using Markov model	Latency, energy efficiency	Improve on heterogeneous environments, security protocols Context-aware provisioning, cross-layer coordination
[107]	It studies low latency, delay constrained data offloading for latency-sensitive applications in hybrid cloud fog edge environments	Delay efficient offloading and resource allocation with the synergy of fog and edges	Average task delay, system utilization, task success rate	On energy efficiency, security concerns and scalability
[108]	Explored computation offloading optimization for federated cloud, edge, and fog systems with dynamic decisions, strict latency, resource heterogeneity, and mobility	Optimization Approaches: Sub-gradient methods, queueing models, and mixed-integer programming. Machine Learning Approaches: RL, DRL, and hybrid models incorporating supervised and unsupervised learning	–	Explored computation offloading optimization for federated cloud, edge, and fog systems with dynamic decisions, strict latency, resource heterogeneity, and mobility
[111]	To minimize offloading costs for user devices in dynamic cloud-fog environments	Mixed Integer Nonlinear Programming problem and Simulated Annealing Algorithm	Offloading Cost, Task Completion Time, Resource Utilization Efficiency	The impact of diverse task types and dynamic mobility patterns on resource allocation is unexplored
[109]	Optimizes task offloading in edge-fog-cloud environments for supporting multi-user scenarios	Stochastic geometry and queueing theory-based framework	Latency, energy consumption, system utilization and task success rate	Optimization under extreme traffic and load conditions

Task placement in heterogeneous networks is optimized in [112] for latency and resource use, however, privacy and scalability are not addressed. Yadav et al. [113] consider energy vs. latency tradeoffs in vehicular fog networks, and tackle the scalability problem by introducing adaptive offloading strategies, but contend that privacy remains an open problem. To achieve efficient resource use, reference [114] optimized offloading using game theory in a hierarchical architecture, however, the study shows shortcomings in real-time dynamic allocation and multi-objective optimization. A cost-efficient dynamic cloud fog offloading in [111] proposes the framework and showcases the scope for improving scalability with a focus on multi-objective optimization. The framework for resource-efficient offloading proposed in [115] includes opportunities for privacy-preserving algorithms and large-scale IoT integration.

In cloud, fog, and edge computing, computation offloading optimizes resources, reduces latency, and improves performance by transferring some tasks to different infrastructures. The research gap for computational offloading, where the need for analysis of partial offloading using numerous system levels from edge through fog to cloud. Also, minimal focus was given to peer-to-peer offloading methods, that would let devices share tasks directly in decentralized networks and task migration along with system resilience during failures and mobility.

5 Resource Allocation

Resource allocation is the task of assigning computational resources, such as CPU, memory, bandwidth, and storage of applications based on their requirements to balance workloads, improve utilization, and maintain QoS compliance [116]. Deep reinforcement learning as a technique takes dynamic resource allocation under more complex environments, where decisions must be based on multiple network states. Resource requirements for allocation are also predicted by other machine learning methods such as neural networks and support vector machines. The resource allocation problem is modeled by game theory as a problem of interactions among multiple actors, resulting in equilibrium states by balancing various objectives. Greedy methods, such as heuristic algorithms, make locally optimal decisions that attempt to approximate a global optimum to solve simple, yet effective problems. Resource allocation is a key component in modern distributed computing systems to ensure system reliability, energy efficiency, cost optimization, and scalability, for latency-sensitive applications like IoT, and real-time analytics. Table 6 presents the summary of the papers based on the problem addressed, algorithms used, metrics, and future scope.

Table 6: Papers based on resource allocation

Paper	Problem addressed	Algorithms	Metrics	Future scope
[117]	Efficient resource allocation in fog computing to reduce latency and improve efficiency.	Ant colony Optimization, Particle swarm Optimization, FCFS, Round Robin, Min-Min/Max-Min.	Response time, data center processing time, cost	Enables runtime on-demand resource allocation and load balancing to increase efficiency.
[118]	Inefficient centralized scheduling, high latency, and single points of failure in MEC networks.	Decentralized scheduling, resource allocation, and task migration using potential game theory.	Resource Utilization, Migration Cost, and Performance Ratio	Study additional quality of service metrics while exploring scalability and migration overhead optimizations through deep learning for predictive scheduling.

(Continued)

Table 6 (continued)

Paper	Problem addressed	Algorithms	Metrics	Future scope
[119]	Trust and transparency in resource transactions are improved using a blockchain-aided auction mechanism.	For the IIoT edge computing, game theoretic principles are used to ensure efficient and fair resource allocation.	Latency, Scalability, Fairness, security, and trustworthiness	Expanding the blockchain-based auction framework on IIoT and edge to support scalable resource allocation in large IIoT and edge environments.
[120]	Resources contention, fairness, and scalability issues in dynamic MEC environments are addressed.	Optimize pricing and resource allocation for offloading tasks using adaptive best response bidding strategies.	Resource allocation efficiency, load balancing, auction fairness, and latency	Extend the auction model to accommodate heterogeneous resources and provide large-scale MEC systems with complicated user needs.

To achieve optimal performance characterized by minimum latency, we need efficient resource allocation techniques to be performed within the edge, fog as well as cloud environments. Agarwal et al. [117] propose a modified first fit packing algorithm and efficient resource allocation to achieve better efficiency and lower latency in fog computing. A Lyapunov optimization framework together with RDC and RDC-NeP algorithms to minimize cost and response time in an MEC environment is proposed in [121]. Effectively implementing a decentralized task scheduling and Resource Allocation Protocol [118] enables real-time tasks with heterogeneous resources to help improve scalability and removal of bottlenecks. In edge environments, multi-criteria decision analysis is integrated by Edgify [122] to optimize resource decisions, using GWA-T-12 datasets. CloudSim experiments are used to validate the work of [123] that uses a modified Ant Bee Colony algorithm to optimize offloading and bandwidth usage. In edge computing, a utility-aware resource-sharing mechanism in the form of an auction is proposed in [124] to improve resource utilization and decrease latency. In IIoT environments, a blockchain-aided auction-based resource allocation [119] applies blockchain and auction-based mechanisms to provide fairness, scalability, and security. In [125], a reverse auction framework for mobile cloud-edge computing is introduced, which contains two algorithms related to task allocation and pricing for optimal matching between tasks and servers while minimizing cost. Similarly in [96], the microeconomic theory to resource allocation is applied in MEC, while at the same time keeping user satisfaction as well as system performance balanced. The author presents a dynamic, generalized second price-based repeated auction for real-time resource allocation for IoT and mobile applications [120]. The presented methods satisfy fairness, scalability, and energy efficiency in modern distributed systems with robust solutions.

The efficient transfer of data, together with optimized communication, remains essential for cloud, fog, and edge computing systems. Distributed systems benefit from innovative protocols introduced in [126] and [127] which enhance their communication efficiency and scalability while reducing their energy consumption. The established techniques deliver relevant data for time-sensitive systems while serving restricted resource conditions and low-latency requirements. Resource allocation solves the dynamic requirements of a variety of applications through the use of intelligent strategies and adaptability mechanisms. Future work in Table 6 focuses on developing scalable, context-sensitive allocation techniques to improve resource efficiency and support next-generation technologies in heterogeneous computing. Studies investigated multiple gaps that exist regarding resource allocation strategies within cloud, fog and edge environments. The main obstacles in these systems relate to scalability and real-time adaptability across highly dynamic heterogeneous infrastructure which intensifies with machine learning and decentralized applications. Energy efficiency and multi-objective optimization represent targets for expansion when considering latency-sensitive and IoT-intensive systems. The potential of blockchain, federated learning together with predictive analytics and bio-inspired algorithms exists despite their difficulty to scale and perform tasks under real-world scenarios.

6 Time Synchronization

Time synchronization plays a vital role in efficient resource provisioning and optimization in edge, fog, and cloud computing by ensuring coordinated task execution, efficient resource allocation, and seamless communication across distributed nodes. Computing systems that use various heterogeneous devices require synchronized operation to avoid data inconsistencies, properly distribute workloads, and cut down on response times for time-sensitive operations [128]. System performance decreases when tasks experience misalignment due to delays and resource conflicts as well as offloading inefficiencies. Time-aware resource scheduling controls workflow progression across different edge or fog devices to stop performance problems in autonomous vehicles and industrial systems, as well as smart grids. Synchronization of sleep-wake cycles in energy-limited computing devices optimizes power utilization that requires synchronized time for IoT-based systems to coordinate operations and to reduce unnecessary power consumption [129]. Time synchronization for edge, fog devices creates a consistent timestamp framework that prevents both duplicated data and inconsistencies that are crucial for federated learning and decentralized analytics [130] demanding timely data aggregation. The examination of restricted environments forms the focus of [131] and [132] which introduce compound methods to merge precise timing with better scalability in hierarchical systems.

Time synchronization beyond scheduling and energy efficiency achieves improvements in fault tolerance, latency optimization, and QoS. The accurate measurement of network delay enables synchronized offloading which distributes tasks efficiently to low-latency edge nodes, thereby improving applications like real-time video analytics and AR/VR streaming. Management of resources through blockchain depends on accurate time synchronization to stop duplicate allocation and create transparent transaction logs in decentralized edge networks [133]. Hence, edge/fog computing technology greatly benefits from time synchronization, but this field requires additional detailed investigation to build advanced synchronization systems for dynamic and large-scale distributed structures.

7 System Model

The interaction between edge providers and users in edge computing is a multi-actor problem involving different priorities and constraints per actor, based on a combination of computing, storage, and network resources. Offloading intensive tasks to the edge decreases computing burden and allows storage to host or cache content near users to shorten data transfer distances, decrease latency, and improve throughput. Similar to cloud infrastructure, edge providers run and monetize resources to optimize revenue while balancing acceptable QoS, while other end users value low-cost, low-latency access to computing power. Resource management is a complex challenge due to conflicting objectives coupled with resource availability constraints and varying user load patterns as shown in Table 7.

$$E_{local} = P_{local} * \frac{f_{local}}{C} \tag{1}$$

$$E_{offload} = P_{transmit} * \frac{D}{R} \tag{2}$$

$$T_{local} = \frac{C}{f_{local}} \tag{3}$$

$$T_{offload} = \frac{D}{R} + \frac{C}{f_{local}} \tag{4}$$

$$minE_{total} = \lambda * E_{local} + (1 - \lambda) * E_{offload} \tag{5}$$

$$E_{local,i} = KC_i f^2_{local,i} \tag{6}$$

$$E_{offload,i} = P_{transmit,i} * \frac{D_i}{R_i} \tag{7}$$

$$E_{UAV,j} = E_{hover,j} * E_{compute,j} \tag{8}$$

$$T_{local,i} = \frac{C_i}{f_{local,i}} \tag{9}$$

$$T_{offload,i} = \frac{D_i}{R_i} + \frac{C_i}{f_{UAV,j}} \tag{10}$$

$$min_{\lambda i, f_{local,i}, f_{UAV,j}, q_j(t)} = [\sum_{i=1}^{n}[\lambda_i * E_{offload,i} + (1 - \lambda) * E_{local,i}] + \sum_{j=1}^{m} E_{UAV,j}] \tag{11}$$

Table 7: Actors and concerns in edge computing

Edge provider	Edge users
Improve the utilization of available resources in Edge.	Minimize resource consumption of user devices. • Compute resources • Storage
Minimize energy consumption of Edge Maintain acceptable levels of QoS for edge users. • latency • throughput • service availability	Minimize energy consumption of device Provide better user experience to users by using edge resources when feasible. • Use cached content • Offload computation to Edge
Achieve better pricing Handle different types of users and load patterns with available resource	Optimize edge usage to keep costs low. –

In cloud, fog, and edge computing, resource provisioning, allocation, and computation offloading are essential to achieve the best task execution and overall system performance. In [14], a unified model is discussed for dynamic demands in [134]. QoS requirements based provisioning of cloud resources is discussed in [135] and [136]. Tasks originate from user devices that offload to edge, fog, or cloud layers based on system conditions and the requirements of the task. Resource information centers (RICs) collect data to facilitate resource-to-task provisioning using Resource Provisioning Agents (RPAs). The architecture includes three layers; where Edge Layer-handles latency-sensitive tasks that require their immediate execution, the Fog Layer-balances between latency and complexity of computation, and the Cloud Layer-handles compute-sensitive tasks with less latency. Offloading decisions are dependent on task characteristics (data size, processing density, parallelizable fraction, delay constraints), QoS constraints (latency, energy, cost), and environmental factors (bandwidth, resource availability). The optimization objective is to find the minimum cost, execution time, and violations of resources and service-level agreements while maximizing resource utilization and user satisfaction. This integrated approach would make sure that tasks are done efficiently through the use of cloud, fog, and edge computing layers.

In [137] and [138], the idea of energy-efficient computational offloading in MEC systems is studied, where the energy consumption is minimized while satisfying task deadlines. According to energy cooperation, execution delay, and wireless channel conditions, tasks are executed locally or offloaded to edge servers. The problem is modeled as a constrained optimization problem and solved, using deep learning-based approaches with neural networks predicting the best offloading decisions over conventional iterative methods. The approach ensures task feasibility, reduces energy consumption, and adheres to QoS standards by leveraging mathematical constructs for energy consumption see Eqs. (1) and (2), delay see Eqs. (3) and (4), and optimization objectives see Eq. (5). Each task is represented as data size D, Computation workload C in CPU cycles per bit, Deadline constraint T_{max}, P_{local} is the local device power, f_{local} is the local CPU frequency. $P_{transmit}$ is transmission power, R is wireless transmission rate and $\lambda \in [0, 1]$ denotes the offloading ratio. The comparison of the survey paper based on the research problem, its objectives, and the parameters used are mentioned in Table 8.

Table 8: Comparison of papers based on research problem and its objectives

Paper	Research type	Objective	Parameters
[137]	Computation offloading	To minimize total energy consumption to meet task deadlines, balancing local computation and offloading to MEC servers	Energy consumption, execution delay, and wireless channel conditions
[138]	Computation offloading and resource allocation	Jointly optimizing task offloading, UAV trajectories, and resource allocation in dynamic networks aims to minimize total energy consumption in UAV enabled MEC systems while maintaining mobile user QoS	Energy-efficient computation offloading, resource allocation, UAV mobility, and latency
[139]	Computation offloading and resource allocation	Optimization at edge by minimizing vehicle energy consumption, execution delays (local or offloaded), edge resource and bandwidth utilization	Energy efficiency and latency
[140]	Resource provisioning	Maximize resource utilization of the MEC while satisfying all the QoS requirements.	Latency, throughput, and reliability
[141]	Resource allocation	Optimizing task allocation across edge and cloud resources to minimize the completion cost like latency, energy, and SLA violation	Energy efficiency and QoS
[142]	Resource provisioning	A multi-objective optimization problem to minimize energy consumption, task execution, and transfer delays, maximum system fault tolerance and reliability	Energy consumption, task completion time and system resilience
[143]	Resource provisioning	To maximize user satisfaction and system utility by optimizing resource allocation for the mobile metaverse	Latency, bandwidth, and computational demands
[144]	Resource allocation	Minimize the total system cost while maintaining QoS for vehicular applications	Energy Consumption, delay and task completion time
[145]	Resource allocation	Minimize total system cost and meet URLLC constraints such as task deadlines, success rates, and computation and communication based on energy	Latency, reliability and energy consumption

In the UAV-enabled MEC systems, reference [146] put forward a task offloading, UAV trajectory optimization, and resource allocation problem toward minimizing energy consumption satisfying QoS. The framework can be tailored to user mobility and task variations by using alternating optimization and successive convex approximation for non-convex trajectory optimization. The main metrics are energy efficiency, task completion rate, and latency. The mathematical constructs for energy consumption see Eqs. (6), (7) and (8), delay see Eqs. (9) and (10), and optimization objectives see Eq. (11). Each task is described as D_i as task input data size (bits), c_i as computational workload (CPU cycles), T_i^{max} as deadline for task completion, K is a hardware-dependent constant, $f_{local,i}$ is the local CPU frequency, $P_{transmit,i}$ is the transmission power, $E_{compute,j}$ is energy for computing offloaded tasks, λ_i means Offloading decision variable (fraction of task offloaded) and $q_j(t)$ is UAV trajectory at time t.

8 Evaluation Metrics

The evaluation metrics are being used to compute the resource provisioning, allocation, and offloading strategies in cloud, fog, and edge computing as the quantitative measures to compare the performance of the system, utilization of resources, and user satisfaction. They enable an analysis of latency [79], throughput, energy efficiency, and computation cost to ensure that the provisioning strategy can respond to changing workloads and is application-specific. Such metrics achieved; resource allocation optimization, delay reduction, energy performance [82], and scaling up without degrading service reliability and availability. Martinez et al. Table 9 provides an overview of the evaluation metrics.

Table 9: Evaluation metrics for resource provisioning

Type	Key metrics	Description	Formula
Performance metrics	Latency	Measures the time delay between request and response in service execution	$t_{response} - t_{request}$
	Throughput	The rate at which tasks are processed successfully within a time frame	$\frac{NumberofTasksCompleted}{TimeInterval}$
	Response time	The total time taken to process a task, including delays in execution and communication	$t_{execution} + t_{transfer} + t_{queueing}$
	Execution time	The time taken to complete a specific task or job on allocated resources	$\frac{Workload(CPUcycles)}{ProcessingSpeed(cycles/sec)}$
Resource utilization metrics	CPU utilization	Indicates how effectively the CPU resources are being used	$\frac{TotalCPUtimeUsed}{TotalAvailableCPUTime} * 100$
	Memory usage	Tracks memory consumption relative to the available capacity	$\frac{UsedMemory}{TotalMemory} * 100$
	Bandwidth utilization	Measures the efficiency of network resource usage	$\frac{DataTransfered}{TotalBandwidthCapacity} * 100$
	Storage utilization	Reflects the consumption of storage resources across layers	$\frac{UsedStorage}{TotalStorage} * 100$
QoS metrics	Service availability	The percentage of time the service is operational and accessible	$\frac{Uptime}{TotalTime} * 100$
	Reliability	The consistency of a system in delivering accurate results over time	$\frac{NumberofSuccessfulTransactions}{TotalTransactions}$
	SLA compliance	Measures adherence to predefined QoS parameters, such as uptime and latency guarantees	$\frac{TimeSLAMet}{TotalOperationalTime} * 100$
Energy efficiency metrics	Energy consumption	The total energy utilized for task execution and data transfer	$P * t$
	Energy-delay product	Evaluates the trade-off between energy consumption and latency	$EnergyConsumption * Latency$
	Power efficiency	The ratio of computational performance to energy consumed	$\frac{ComputationalThroughput}{EnergyConsumption}$

(Continued)

Table 9 (continued)

Type	Key metrics	Description	Formula
Scalability metrics	Scalability	Assesses the ability of the system to handle increased workloads by scaling resources	$\frac{PerformancewithIncreasedLoad}{PerformancewithBaselineLoad} * 100$
	Elasticity	Measures the adaptability of resource allocation to sudden changes in demand	$\frac{ResourcesScaled}{ResourcesRequired} * 100$
Network specific metrics	Network latency	The time taken for data to travel through the network	$t_{propagation} + t_{transmission} + t_{processing}$
	Packet loss rate	The percentage of data packets lost during transmission	$\frac{PacketsLost}{TotalPacketsSent} * 100$
	Data transfer rate	The speed of data movement between computing layers	$\frac{TotalDataTransferred}{TransferTime}$
User-centric metrics	User satisfaction	Evaluates the end-user experience through satisfaction scores or feedback	$\frac{PositiveFeedbacks}{TotalFeedbacks} * 100$
	Task success rate	The percentage of tasks completed successfully as per user requirements	$\frac{TasksSuccessfullyCompleted}{TotalTasksSubmitted} * 100$
Security and privacy metrics	Data security	Evaluates measures to protect data during computation and transmission	No standard formula;
	Privacy preservation	Ensures user data confidentiality in multi-tenant environments	$\frac{ProtectedData}{TotalData} * 100$

Paper [147] shows that processing data at the edge in a distributed fog-cloud system reduces cloud data exchange, easing network congestion and improving energy efficiency, as seen in weapon detection and face recognition applications. To understand the system behavior, optimize the use of resources, and increase user satisfaction these metrics are important. Comprehensive literature surveys on various types of resource provisioning, as presented in Tables 1–3 and Tables 4–6, explore research in resource allocation, provisioning, and computation. These studies identify key metrics essential for analyzing system behavior, improving efficiency, and enabling adaptive strategies, as summarized in Fig. 5a.

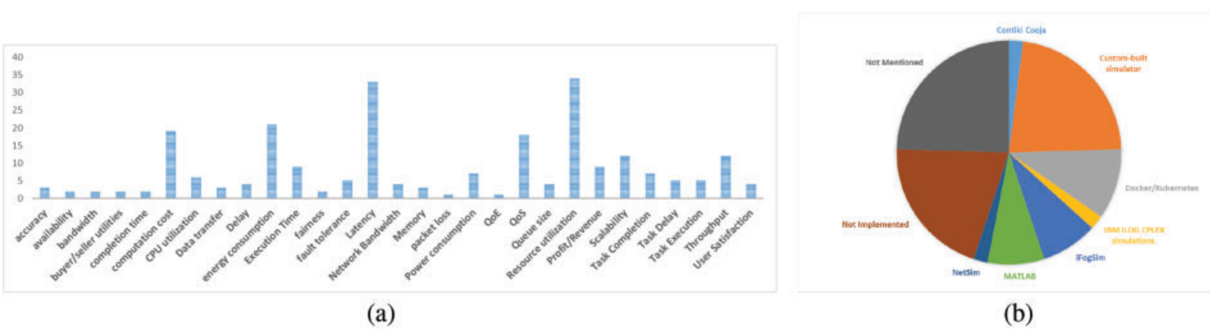


Figure 5: Summarized **(a)** evaluation metrics; **(b)** simulators used in various review

The critical performance metrics that determine cloud and edge computing resource allocations include latency [78] alongside energy efficiency standards. The combination of artificial intelligence techniques together with optimization algorithms and collaborative solution frameworks simultaneously offer better task offloading capabilities coupled with decreases in latency and enhanced energy efficiency [66]. Network simulations driven by the SDN framework analyze programming decisions which results in efficient service delivery for applications with varied traffic and processing requirements. Cloud, fog, and edge computing systems require latency measurements combined with energy efficiency evaluations, throughput monitoring [98], and assessment of packet Loss rates during their operations. Time-sensitive operations including autonomous driving systems healthcare devices and virtual gaming activities need immediate responses to reduce safety risks. Limited power resources present in IoT devices and edge systems make energy efficiency a fundamental requirement. The performance of throughput allows seamless operation in data-intensive applications such as video streaming and cloud gaming while packet loss [80] creates problems in real-time communication and mission-critical operations. The optimization of the metrics ensures the necessary scalability, reliability, and operational efficiency for a wide range of applications.

9 Simulators

Evaluating and optimizing cloud, fog, and edge computing environments is essential and requires the use of simulators as discussed in [148,149]. This results in tools for researchers to evaluate specific strategies optimized in each domain, including energy efficiency, mobility, and SDN integration [150–152]. Table 10 lists the commonly used simulators, each designed for a specific purpose to meet unique research needs for system design and performance evaluation. Each of the major simulators used in these research studies is explained as follows and summarized in Fig. 5b.

Table 10: Simulators used for cloud, fog, edge computing with SDN

Simulator	Focus	Key features	Best for
CloudSim	Cloud computing	Centralized resource management, scalability	Datacenter modeling and scheduling policies
iFogSim	Fog/Edge computing	Latency, energy consumption in IoT	IoT-driven fog deployment and resource allocation
EdgeCloudSim	Edge computing	Low-latency, mobile user simulations	Mobile edge computing and dynamic resource use
PureEdgeSim	Edge computing	Lightweight, scalability for IoT workloads	Large-scale edge network simulations
FogNetSim++	Fog computing	SDN, topology generation, energy efficiency	Fog-based IoT resource orchestration
GreenCloud	Cloud computing	Energy and network traffic analysis	Energy-efficient cloud system studies
CloudSimSDN	Cloud/Edge + SDN	SDN flow control, service chaining	Network-aware simulations in cloud-edge systems

CloudSim is a widely used simulation framework to model cloud infrastructures that provide an environment for data center, VM, and resource management policies [153]. Furthermore, reference [154] discusses the usage of provision resources in centralized clouds, task scheduling, or analyzing energy efficiency. It is used by the researchers to optimize VM scheduling algorithms, like Round Robin (RR), First-Come-First-Serve (FCFS), and Shortest Job First (SJF), to distribute tasks among VMs, and test dynamic resource provisioning techniques and VM placement strategies [155]. CloudSim extends to power-aware studies on the energy-efficient allocation of cloud data center resources [156]. Experiments in progress on latency, scalability, energy consumption, and cost efficiency advance resource management strategies [157].

To evaluate resource management in fog and edge computing, iFogSim is a simulation toolkit that expands CloudSim features for simulating fog nodes, sensors, actuators, and IoT environments [158]. For instance, it is geared up as an explorative platform for researchers, supporting the study of various research areas and the examination of latency-sensitive IoT applications such as industrial IoT [159], healthcare monitoring [160] and smart traffic systems [161]. It also helps in reducing the power utilization in fog-distributed nodes and IoT devices through its energy efficiency tools. iFogSim further evaluates placement and scheduling algorithms for maximizing service utility (in terms of latency and resource utilization) in IoT workloads [162].

An EdgeCloudSim [163] extension of CloudSim is proposed which models edge computing systems for mobility, load generation, and network simulation, applicable to low latency applications. It evaluates real-time IoT workloads [164] for health and traffic management, reflecting on how mobility and variance in networks affect resource allocation. By allowing testing of offloading algorithms against metrics such as latency, energy efficiency, resource utilization, and scalability, the simulator supports performance evaluations of applications as well as testing of applications offloading algorithms [165]. Being modular, it is adaptable to different IoT and edge computing situations.

PureEdgeSim [166], a simulation framework for cloud, edge, and mist computing environments, is developed focusing on dynamic heterogeneity, task offloading, resource allocation, and workload orchestration. It can be used for applications such as vehicular networks and healthcare monitoring from IoT applications that require strict latency and energy constraints [167]. Metrics including latency, energy consumption, resource utilization, and task success rates are evaluated [168,169]. Due to its modular design, its uses include simulating heterogeneous devices and mist-edge-cloud integration, allowing the use of the tool to optimize resource allocation in dynamic environments [170].

iFogSim is further extended with advanced fog computing features such as network topology modeling, dynamic task allocation, and mobility management to create FogNetSim++ [171]. Resource provisioning, schedule of tasks, and service placement aiming to minimize application latency in real-world fog and IoT scenarios [172]. Latency, energy, network overhead, response time, and resource utilization are evaluated to support the performance analysis of complex fog architectures [173].

GreenCloud [174] provides insights about energy use in the computing, communication, and cooling components of data centers. Energy awareness research with GreenCloud [175] is in the energy-aware scheduling, resource allocation, and network optimization to minimize the environmental impact of cloud infrastructures. Energy consumption, carbon emissions, task completion time, PUE, and network performance are key evaluations, that support studies of sustainability tradeoffs in large-scale cloud systems [176,177].

CloudSimSDN [178] uses the extensions of CloudSim to simulate the SDN cloud environment, where network-aware resource management is integrated with SDN controllers. More specifically, research using CloudSimSDN [179,180] has explored the problem of network performance optimization, task scheduling, and resource allocation. Network latency, throughput, energy efficiency, task execution time, and load balancing are all being evaluated. Further studies conduct dynamic traffic routing, energy-aware VM migrations, and QoS improvements in SDN-based clouds [181].

10 Challenges in Resource Provisioning

Given the dynamic and heterogeneous nature of cloud, fog, and edge computing, as discussed in the literature survey, we arrived at the following challenges for resource provisioning:

Scalability: With the exponential growth in connected devices, managing resources efficiently across large-scale, heterogeneous environments with low latency and high throughput is crucial.

Heterogeneity: The need for interoperability standards arises as the computing and network devices become more diverse for handling varying capacities.

Dynamic Workloads: To maintain QoS under changing workloads, real-time adaptation mechanisms are essential.

SDN Controller Placement: Hierarchical models for fault tolerance concerns are used to optimize the placement of SDN controllers minimizing latency, load balancing, and scalability.

Latency and Network Bottlenecks: Traffic flow and the network configurations must be optimized to minimize communication delays and congestion in the network, particularly for the latency-sensitive case.

Time Synchronization: Real-time coordination in cloud, edge, and fog computing depends heavily on precise time synchronization for better scalability in hierarchical systems.

Energy Efficiency: For reduction in cost and sustainability across cloud, fog, and edge infrastructures, energy-aware provisioning strategies are needed.

Mobility: Mobility-aware frameworks need to adapt resource provisioning to handoffs and connectivity changes while retaining QoS.

QoS and SLA Compliance: In the dynamic environment, robust monitoring and adaptive systems are required to satisfy consistent QoS and SLA requirements.

11 Future Work

In conclusion, there exists a significant scope for developing efficient resource provisioning and optimization mechanisms in edge and fog computing systems. In particular, edge computing remains relatively unexplored, presenting numerous opportunities for further research and innovation. The following are a few of the future directions that could be derived from the above surveys.

Resource Optimization: Create and examine AI-based algorithms for predicting real-time workloads along adaptive resource distributions which are optimized for settings involving auto vehicles and healthcare IoT requiring reduced response times.

Secure Resource Sharing: SDN-based fog and edge systems should utilize blockchain technology for resource sharing because it establishes tamper-proof secure transaction records to defend against privacy and data integrity issues.

Advanced Scheduling Mechanisms: The task scheduling framework requires multi-objective optimization algorithms to balance latency, energy consumption, and resource utilization.

Energy-Aware Resource Management: Green computing frameworks that minimize energy consumption and consume renewable energy sources are an essential prerequisite for a sustainable operation.

Mobility-Aware Provisioning: Low latency and high QoS for mobile users without frequent handoffs require adaptive algorithms.

5G and Beyond Integration: To maximize 5G technologies such as network slicing and Ultra-Reliable and Low-Latency Communications (URLLC) to enhance resource provisioning mechanisms, services must be adapted to cloud, fog, and edge systems.

Elastic Resource Allocation: Two important points essential for dynamic scaling systems are workload burst and fluctuating application demands.

Security: Security of IoT has become essential as this technology grows at an accelerated rate. Lightweight solutions are offered through GS3 [182] which performs shuffling and substitution combined with scrambling operations, unlike AES and ChaCha20 standards. Future research can enhance its resilience and scalability, with the integration of AI and post-quantum cryptography, thereby promoting sustainable growth of IoT.

12 Conclusion

This survey provides a comprehensive analysis of various developments in the field of cloud, fog, and edge computing focusing in terms of architecture, challenges, and the approaches to the solutions. Targeting resource provisioning, allocation, and computation offloading as crucial mechanisms for an efficient system in such environments, by highlighting the shortcomings of current research practices on IoT. Finally, the role of integration with SDN as a key enabler is explored for satisfying the requirements in terms of low latency, high throughput as well as scalability. Static, dynamic, and user-centric resource provisioning and its application in IoT, healthcare, and autonomous systems are discussed. Heterogeneous infrastructure is considered concerning computation offloading strategies for lowering energy consumption, latency, and costs. The role of SDN in resource orchestration, task scheduling, and traffic management is demonstrated for integrated seamless cloud, fog, and edge. The survey also identifies research gaps in scalability, multi-tenancy, and interoperability and proposes advancement in computing systems by hybrid orchestration models, along with real-time coordination and security-aware resource management.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Authors jointly planned the survey study. Sreebha Bhaskaran collected and categorized the relevant literature, conducted a comparative analysis of existing approaches, structured the manuscript, and wrote the initial draft, while Supriya Muthuraman contributed to the critical discussion and refinement of the arguments. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Hayes B. Cloud computing. New York, NY, USA: ACM; 2008.
2. Research GV. Game applications market size, share & trends analysis Report By marketplace (Google Play Store, Apple iOS Store), By Region (Asia Pacific, North America, Europe, MEA), And segment forecasts, 2022–2028; 2022. [cited 2024 Sep 20]. Available from: https://www.grandviewresearch.com/industry-analysis/game-applications-market-report.
3. Varghese B, Wang N, Barbhuiya S, Kilpatrick P, Nikolopoulos DS. Challenges and opportunities in edge computing. In: 2016 IEEE International Conference on Smart Cloud (SmartCloud). New York, NY, USA: IEEE; 2016. p. 20–6.
4. Botta A, de Donato W, Persico V, Pescapé A. Integration of cloud computing and internet of things: a survey. Future Gener Comput Syst. 2016;56(7):684–700. doi:10.1016/j.future.2015.09.021.
5. Kaur K, Garg S, Aujla GS, Kumar N, Rodrigues JJ, Guizani M. Edge computing in the industrial internet of things environment: software-defined-networks-based edge-cloud interplay. IEEE Commun Magazine. 2018;56(2):44–51. doi:10.1109/MCOM.2018.1700622.
6. Bhaskaran S, Supriya M. Study on networking models of SDN using mininet and MiniEdit. In: 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT). Bengaluru, India: IEEE; 2024. p. 1159–64.
7. Avinash S, Srikar AS, Naik VP, Bhaskaran S. SDN-based hybrid load balancing algorithm. In: 2023 3rd International Conference on Intelligent Technologies (CONIT). Hubli, India: IEEE; 2023. p. 1–5.

8. Jazaeri SS, Asghari P, Jabbehdari S, Javadi HHS. Toward caching techniques in edge computing over SDN-IoT architecture: a review of challenges, solutions, and open issues. Multimed Tools Appl. 2024;83(1):1311–77. doi:10.1007/s11042-023-15657-7.
9. Baktir AC, Ozgovde A, Ersoy C. How can edge computing benefit from software-defined networking: a survey, use cases, and future directions. IEEE Commun Surv Tutor. 2017;19(4):2359–91. doi:10.1109/COMST.2017.2717482.
10. Jazaeri SS, Jabbehdari S, Asghari P, Haj Seyyed Javadi H. Edge computing in SDN-IoT networks: a systematic review of issues, challenges and solutions. Clust Comput. 2021;24(4):3187–228. doi:10.1007/s10586-021-03311-6.
11. Almutairi J, Aldossary M. A novel approach for IoT tasks offloading in edge-cloud environments. J Cloud Comput. 2021;10(1):28. doi:10.1186/s13677-021-00243-9.
12. Buyya R, Srirama SN. Fog and edge computing: principles and paradigms. Hoboken, NJ, USA: John Wiley & Sons; 2019.
13. Dolui K, Datta SK. Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing. In: 2017 Global Internet of Things Summit (GIoTS). Geneva, Switzerland: IEEE; 2017. p. 1–6.
14. Luo Q, Hu S, Li C, Li G, Shi W. Resource scheduling in edge computing: a survey. IEEE Commun Surv Tutor. 2021;23(4):2131–65. doi:10.1109/COMST.2021.3106401.
15. Afrin M, Jin J, Rahman A, Rahman A, Wan J, Hossain E. Resource allocation and service provisioning in multi-agent cloud robotics: a comprehensive survey. IEEE Commun Surv Tutor. 2021;23(2):842–70. doi:10.1109/COMST.2021.3061435.
16. Hu Y, Wong J, Iszlai G, Litoiu M. Resource provisioning for cloud computing. In: Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research; 2009; Toronto, ON, Canada. p. 101–11.
17. Skarlat O, Schulte S, Borkowski M, Leitner P. Resource provisioning for IoT services in the fog. In: 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA). Macau, China: IEEE; 2016. p. 32–9.
18. Sumalatha K, Anbarasi M. A review on various optimization techniques of resource provisioning in cloud computing. Int J Elec Comput Eng. 2019;9(1):629. doi:10.11591/ijece.v9i1.pp629-634.
19. Abouaomar A, Cherkaoui S, Mlika Z, Kobbane A. Resource provisioning in edge computing for latency-sensitive applications. IEEE Int Things J. 2021;8(14):11088–99. doi:10.1109/JIOT.2021.3052082.
20. Shakarami A, Shakarami H, Ghobaei-Arani M, Nikougoftar E, Faraji-Mehmandar M. Resource provisioning in edge/fog computing: a comprehensive and systematic review. J Syst Archit. 2022;122(1):102362. doi:10.1016/j.sysarc.2021.102362.
21. Ali NN, Zeebaree SR. Distributed resource management in cloud computing: a review of allocation, scheduling, and provisioning techniques. Indonesian J Comput Sci. 2024;13(2). doi:10.33022/ijcs.v13i2.3823.
22. Fahimullah M, Ahvar S, Agarwal M, Trocan M. Machine learning-based solutions for resource management in fog computing. Multimed Tools Appl. 2024;83(8):23019–45. doi:10.1007/s11042-023-16399-2.
23. Kaur K, Singh A, Sharma A. A systematic review on resource provisioning in fog computing. Trans Emerg Telecomm Technol. 2023;34(4):e4731. doi:10.1002/ett.4731.
24. Xu Z, Zhang Y, Li H, Yang W, Qi Q. Dynamic resource provisioning for cyber-physical systems in cloud-fog-edge computing. J Cloud Comput. 2020;9:1–16.
25. Ali O, Ishak MK, Wuttisittikulkij L, Maung TZB. IoT devices and Edge gateway provisioning, realtime analytics for simulated and virtually emulated devices. In: 2020 International Conference on Electronics, Information, and Communication (ICEIC). Barcelona, Spain: IEEE; 2020. p. 1–5.
26. Li Y, Tang X, Cai W. Dynamic bin packing for on-demand cloud resource allocation. IEEE Trans Parallel Distrib Syst. 2015;27(1):157–70. doi:10.1109/TPDS.2015.2393868.
27. Etemadi M, Ghobaei-Arani M, Shahidinejad A. Resource provisioning for IoT services in the fog computing environment: an autonomic approach. Comput Commun. 2020;161(2):109–31. doi:10.1016/j.comcom.2020.07.028.
28. Tasiopoulos AG, Ascigil O, Psaras I, Pavlou G. Edge-MAP: auction markets for edge resource provisioning. In: 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM). Chania, Greece: IEEE; 2018. p. 14–22.

29. Satyanarayanan M, Bahl P, Caceres R, Davies N. The case for vm-based cloudlets in mobile computing. IEEE Pervasive Computing. 2009;8(4):14–23. doi:10.1109/MPRV.2009.82.
30. Khalid A, ul Ain Q, Qasim A, Aziz Z. QoS based optimal resource allocation and workload balancing for fog enabled IoT. Open Comput Sci. 2021;11(1):262–74. doi:10.1515/comp-2020-0162.
31. Wang N, Varghese B, Matthaiou M, Nikolopoulos DS. ENORM: a framework for edge node resource management. IEEE Trans Serv Comput. 2017;13(6):1086–99.
32. Ferrández-Pastor FJ, Mora H, Jimeno-Morenilla A, Volckaert B. Deployment of IoT edge and fog computing technologies to develop smart building services. Sustainability. 2018;10(11):3832. doi:10.3390/su10113832.
33. Ceselli A, Premoli M, Secci S. Mobile edge cloud network design optimization. IEEE/ACM Trans Netw. 2017;25(3):1818–31. doi:10.1109/TNET.2017.2652850.
34. Santoyo-González A, Cervelló-Pastor C. Network-aware placement optimization for edge computing infrastructure under 5G. IEEE Access. 2020;8:56015–28. doi:10.1109/ACCESS.2020.2982241.
35. Alsadie D. A comprehensive review of AI techniques for resource management in fog computing: trends, challenges and future directions. IEEE Access. 2024;12(3):118007–59. doi:10.1109/ACCESS.2024.3447097.
36. Baranwal G, Vidyarthi DP. FONS: a fog orchestrator node selection model to improve application placement in fog computing. J Supercomput. 2021;77(9):10562–89. doi:10.1007/s11227-021-03702-x.
37. Lu S, Wu J, Duan Y, Wang N, Fang J. Towards cost-efficient resource provisioning with multiple mobile users in fog computing. J Parallel Distrib Comput. 2020;146(5):96–106. doi:10.1016/j.jpdc.2020.08.002.
38. Breitbach M, Schäfer D, Edinger J, Becker C. Context-aware data and task placement in edge computing environments. In: 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom). Kyoto, Japan: IEEE; 2019. p. 1–10.
39. Fan J, Wei X, Wang T, Lan T, Subramaniam S. Deadline-aware task scheduling in a tiered IoT infrastructure. In: GLOBECOM 2017-2017 IEEE Global Communications Conference. Singapore: IEEE; 2017. p. 1–7.
40. Zhang Q, Zhang C, Zhao J, Wang D, Xu W. Dynamic resource allocation for multi-access edge computing in urban rail transit. IEEE Trans Veh Technol. 2024;74(2):3296–310. doi:10.1109/TVT.2024.3479215.
41. Pg Ali Kumar DSNK, Newaz SS, Rahman FH, Lee GM, Karmakar G, Au TW. Green demand aware fog computing: a prediction-based dynamic resource provisioning approach. Electronics. 2022;11(4):608. doi:10.3390/electronics11040608.
42. Wang B, Irwin D, Shenoy P, Towsley D. INVAR: inversion aware resource provisioning and workload scheduling for edge computing. In: IEEE INFOCOM 2024-IEEE Conference on Computer Communications. Vancouver, BC, Canada: IEEE; 2024. p. 1511–20.
43. Tang Z, Mou F, Lou J, Jia W, Wu Y, Zhao W. Joint resource overbooking and container scheduling in edge computing. IEEE Trans Mob Comput. 2024;23(12):10903–17. doi:10.1109/TMC.2024.3386936.
44. Wan X. Dynamic resource management in MEC powered by edge intelligence for smart city internet of things. J Grid Comput. 2024;22(1):29. doi:10.1007/s10723-024-09749-3.
45. Gkatzios N, Koumaras H, Fragkos D, Koumaras V. A proof of concept implementation of an AI-assisted user-centric 6G network. In: 2024 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit). Antwerp, Belgium: IEEE; 2024. p. 907–12.
46. Baidya T, Moh S. Comprehensive survey on resource allocation for edge-computing-enabled metaverse. Comput Sci Rev. 2024;54(16):100680. doi:10.1016/j.cosrev.2024.100680.
47. Qin L, Lu H, Chen Y, Chong B, Wu F. Towards decentralized task offloading and resource allocation in user-centric MEC. IEEE Trans Mob Comput. 2024;23(12):11807–23. doi:10.1109/TMC.2024.3399766.
48. Zhou C, Gao J, Liu Y, Hu S, Cheng N, Shen XS. User-centric service provision for edge-assisted mobile AR: a digital twin-based approach. In: 2024 IEEE/CIC International Conference on Communications in China (ICCC). Hangzhou, China: IEEE; 2024. p. 850–5.
49. Ma X, Wang S, Zhang S, Yang P, Lin C, Shen X. Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing. IEEE Trans Cloud Comput. 2019;9(3):968–80. doi:10.1109/TCC.2019.2903240.

50. Kherraf N, Alameddine HA, Sharafeddine S, Assi CM, Ghrayeb A. Optimized provisioning of edge computing resources with heterogeneous workload in IoT networks. IEEE Trans Netw Serv Manag. 2019;16(2):459–74. doi:10.1109/TNSM.2019.2894955.
51. Ascigil O, Tasiopoulos AG, Phan TK, Sourlas V, Psaras I, Pavlou G. Resource provisioning and allocation in function-as-a-service edge-clouds. IEEE Trans Serv Comput. 2021;15(4):2410–24. doi:10.1109/TSC.2021.3052139.
52. Chuang IH, Sun RC, Tsai HJ, Horng MF, Kuo YH. A dynamic multi-resource management for edge computing. In: 2019 European Conference on Networks and Communications (EuCNC). Valencia, Spain: IEEE; 2019. p. 379–83.
53. Zhou Z, Yu S, Chen W, Chen X. CE-IoT: cost-effective cloud-edge resource provisioning for heterogeneous IoT applications. IEEE Internet Things J. 2020;7(9):8600–14. doi:10.1109/JIOT.2020.2994308.
54. Cao X, Tang G, Guo D, Li Y, Zhang W. Edge federation: towards an integrated service provisioning model. IEEE/ACM Trans Netw. 2020;28(3):1116–29. doi:10.1109/TNET.2020.2979361.
55. Duan X, Lu H, Li S. Elastic service provisioning for mobile edge computing. In: 2021 IEEE 7th International Conference on Big Data Intelligence and Computing (DataCom). Huizhou, China: IEEE; 2021. p. 57–60.
56. Alahmadi AA, Musa MO, El-Gorashi TE, Elmirghani JM. Energy efficient resource allocation in vehicular cloud based architecture. In: 2019 21st International Conference on Transparent Optical Networks (ICTON). Angers, France: IEEE; 2019. p. 1–6.
57. Nakamura Y, Mizumoto T, Suwa H, Arakawa Y, Yamaguchi H, Yasumoto K. *In-situ* resource provisioning with adaptive scale-out for regional iot services. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). Seattle, WA, USA: IEEE; 2018. p. 203–13.
58. Liwang M, Gao Z, Wang X. Let's trade in the future! A futures-enabled fast resource trading mechanism in edge computing-assisted UAV networks. IEEE J Sel Areas Commun. 2021;39(11):3252–70. doi:10.1109/JSAC.2021.3088657.
59. Ma Y, Liang W, Li J, Jia X, Guo S. Mobility-aware and delay-sensitive service provisioning in mobile edge-cloud networks. IEEE Trans Mob Comput. 2020;21(1):196–210. doi:10.1109/TMC.2020.3006507.
60. Guo C, He W, Li GY. Optimal fairness-aware resource supply and demand management for mobile edge computing. IEEE Wireless Commun Lett. 2020;10(3):678–82. doi:10.1109/LWC.2020.3046023.
61. Huang M, Liang W, Shen X, Ma Y, Kan H. Reliability-aware virtualized network function services provisioning in mobile edge computing. IEEE Trans Mob Comput. 2019;19(11):2699–713. doi:10.1109/TMC.2019.2927214.
62. Chang P, Miao G. Resource provision for energy-efficient mobile edge computing systems. In: 2018 IEEE Global Communications Conference (GLOBECOM). Abu Dhabi, United Arab Emirates: IEEE; 2018. p. 1–6.
63. Toka L, Haja D, Kőrösi A, Sonkoly B. Resource provisioning for highly reliable and ultra-responsive edge applications. In: 2019 IEEE 8th International Conference on Cloud Networking (CloudNet). Coimbra, Portugal: IEEE; 2019. p. 1–6.
64. Porkodi V, Singh AR, Sait ARW, Shankar K, Yang E, Seo C, et al. Resource provisioning for cyber-physical–social system in cloud-fog-edge computing using optimal flower pollination algorithm. IEEE Access. 2020;8:105311–9. doi:10.1109/ACCESS.2020.2999734.
65. Murthy BK, Shiva SG. Double-state-temporal difference learning for resource provisioning in uncertain fog computing environment. In: 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). Vancouver, BC, Canada: IEEE; 2021. p. 0435–40.
66. Hu S, Shi W, Li G. CEC: a containerized edge computing framework for dynamic resource provisioning. IEEE Trans Mob Comput. 2022;22(7):3840–54. doi:10.1109/TMC.2022.3147800.
67. Zhao T, Zhou S, Guo X, Zhao Y, Niu Z. Pricing policy and computational resource provisioning for delay-aware mobile edge computing. In: 2016 IEEE/CIC International Conference on Communications in China (ICCC). Chengdu, China: IEEE; 2016. p. 1–6.
68. Tang G, Guo D, Wu K, Liu F, Qin Y. QoS guaranteed edge cloud resource provisioning for vehicle fleets. IEEE Trans Veh Technol. 2020;69(6):5889–900. doi:10.1109/TVT.2020.2987839.
69. Fan Y, Wang L, Wu W, Du D. Cloud/edge computing resource allocation and pricing for mobile blockchain: an iterative greedy and search approach. IEEE Trans Comput Soc Syst. 2021;8(2):451–63. doi:10.1109/TCSS.2021.3049152.

70. Rajagopal SM, Supriya M, Buyya R. Blockchain integrated federated learning in edge/fog/cloud systems for IoT-based healthcare applications: a survey. Federated Learning. 2024; 237–69. doi:10.1201/9781003497196.
71. Rakshith G, Rahul M, Sanjay G, Natesha B, Reddy GRM. Resource provisioning framework for IoT applications in fog computing environment. In: 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). Indore, India: IEEE; 2018. p. 1–6.
72. Santos J, Wauters T, Volckaert B, de Turck F. Towards dynamic fog resource provisioning for smart city applications. In: 2018 14th International Conference on Network and Service Management (CNSM). Rome, Italy: IEEE; 2018. p. 290–4.
73. Ma Y, Liang W, Huang M, Xu W, Guo S. Virtual network function service provisioning in MEC via trading off the usages between computing and communication resources. IEEE Trans Cloud Comput. 2022;10(4):2949–63. doi:10.1109/TCC.2020.3043313.
74. Xu X, Cao H, Geng Q, Liu X, Dai F, Wang C. Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment. Concurr Comput. 2022;34(14):e5674. doi:10.1002/cpe.5674.
75. Qu Y, Lu D, Dai H, Tan H, Tang S, Wu F, et al. Resilient service provisioning for edge computing. IEEE Internet Things J. 2021;10(3):2255–71. doi:10.1109/JIOT.2021.3078620.
76. Anjum M, Min H, Ahmed Z. User-centric internet of things and controlled service scheduling scheme for a software-defined network. Appl Sci. 2024;14(11):4951. doi:10.3390/app14114951.
77. Okwuibe J, Haavisto J, Kovacevic I, Harjula E, Ahmad I, Islam J, et al. SDN-enabled resource orchestration for industrial IoT in collaborative edge-cloud networks. IEEE Access. 2021;9:115839–54. doi:10.1109/ACCESS.2021.3105944.
78. Wang H, Liu T, Kim B, Lin CW, Shiraishi S, Xie J, et al. Architectural design alternatives based on cloud/edge/fog computing for connected vehicles. IEEE Commun Surv Tutor. 2020;22(4):2349–77. doi:10.1109/COMST.2020.3020854.
79. Abdulazeez DH, Askar SK. Offloading mechanisms based on reinforcement learning and deep learning algorithms in the fog computing environment. IEEE Access. 2023;11(4):12555–86. doi:10.1109/ACCESS.2023.3241881.
80. Silva A, Mendes P, Rosário D, Cerqueira E, da Costa JPJ, de Freitas EP. A distributed computing model based on delegation of serverless microservices in a cloud-to-thing environment. J Internet Serv Appl. 2024;15(1):233–43.
81. Shahidinejad A, Ghobaei-Arani M. Joint computation offloading and resource provisioning for edge-cloud computing environment: a machine learning-based approach. Softw: Pract Exper. 2020;50(12):2212–30. doi:10.1002/spe.2888.
82. Etemadi M, Ghobaei-Arani M, Shahidinejad A. A learning-based resource provisioning approach in the fog computing environment. J Exp Theor Artif Intell. 2021;33(6):1033–56. doi:10.1080/0952813X.2020.1818294.
83. Binh HTT, Le NP, Minh NB, Hai TT, Minh NQ, Son DB. A reinforcement learning algorithm for resource provisioning in mobile edge computing network. In: 2020 International Joint Conference on Neural Networks (IJCNN). Glasgow, UK: IEEE; 2020. p. 1–7.
84. Shezi N, Meschack M, Bakhe N. A resource provisioning framework overview for a joint optical/wireless transport network (JOWTN). In: 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET). Cape Town, South Africa: IEEE; 2023. p. 1–6.
85. Huynh VSH, Radenkovic M, Wang N. Distributed spatial-temporal demand and topology aware resource provisioning for edge cloud services. In: 2021 Sixth International Conference on Fog and Mobile Edge Computing (FMEC). Gandia, Spain: IEEE; 2021. p. 1–8.
86. Baghban H, Rezapour A, Hsu CH, Nuannimnoi S, Huang CY. Edge-AI: IoT request service provisioning in federated edge computing using actor-critic reinforcement learning. IEEE Trans Eng Manag. 2022;71:12519–28. doi:10.1109/TEM.2022.3166769.
87. Alsurdeh R, Calheiros RN, Matawie KM, Javadi B. Hybrid workflow provisioning and scheduling on cooperative edge cloud computing. In: 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid). Melbourne, Australia: IEEE; 2021. p. 445–54.
88. Liwang M, Wang X. Overbooking-empowered computing resource provisioning in cloud-aided mobile edge networks. IEEE/ACM Trans Netw. 2022;30(5):2289–303. doi:10.1109/TNET.2022.3167396.

89. Huang X, Zhang B, Li C. Platform profit maximization on service provisioning in mobile edge computing. IEEE Trans Veh Technol. 2021;70(12):13364–76. doi:10.1109/TVT.2021.3124483.
90. Badri H, Bahreini T, Grosu D, Yang K. Risk-based optimization of resource provisioning in mobile edge computing. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). Seattle, WA, USA: IEEE; 2018. p. 328–30.
91. Nguyen ND, Phan LA, Park DH, Kim S, Kim T. ElasticFog: elastic resource provisioning in container-based fog computing. IEEE Access. 2020;8:183879–90. doi:10.1109/ACCESS.2020.3029583.
92. Santos J, Wauters T, Volckaert B, de Turck F. Resource provisioning in fog computing: from theory to practice. Sensors. 2019;19(10):2238. doi:10.3390/s19102238.
93. Jang Y, Na GJ, Choi I, Chun IG, Nam D, Suh YK. A resource provisioning manager for edge devices. In: 2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). Madrid, Spain: IEEE; 2023. p. 392–3.
94. Zhu J, Wang X, Wo T, Hu C. Hybrid resource orchestration and scheduling for cyber-physical-human systems. In: 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). Yanuca Island, Cuvu, Fiji: IEEE; 2020. p. 1126–33.
95. Zakarya M, Gillam L, Ali H, Rahman IU, Salah K, Khan R, et al. epcAware: a game-based, energy, performance and cost-efficient resource management technique for multi-access edge computing. IEEE Trans Serv Comput. 2020;15(3):1634–48. doi:10.1109/TSC.2020.3005347.
96. Liu J, Guo S, Liu K, Feng L. Resource provision and allocation based on microeconomic theory in mobile edge computing. IEEE Trans Serv Comput. 2020;15(3):1512–25. doi:10.1109/TSC.2020.3000050.
97. Sharma P. Block chain driven marketing resources in mobile edge computing and system. In: 2023 International Conference on Artificial Intelligence and Smart Communication (AISC). Greater Noida, India: IEEE; 2023. p. 1190–4.
98. Queralta JP, Qingqing L, Zou Z, Westerlund T. Enhancing autonomy with blockchain and multi-access edge computing in distributed robotic systems. In: 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC). Paris, France: IEEE; 2020. p. 180–7.
99. Dai S, Hai L, Li Y, Zhang Z. An incentive auction-based cooperative resource provisioning scheme for edge computing over passive optical networks. In: 2019 18th International Conference on Optical Communications and Networks (ICOCN). Huangshan, China: IEEE; 2019. p. 1–3.
100. Wang Q, Guo S, Liu J, Pan C, Yang L. Profit maximization incentive mechanism for resource providers in mobile edge computing. IEEE Trans Serv Comput. 2022;15(1):138–49. doi:10.1109/TSC.2019.2924002.
101. Baranwal G, Kumar D, Raza Z, Vidyarthi DP. Auction based resource provisioning in cloud computing. Berlin: Springer; 2018.
102. Shi W, Zhang L, Wu C, Li Z, Lau FC. An online auction framework for dynamic resource provisioning in cloud computing. IEEE/ACM Trans Netw. 2015;24(4):2060–73. doi:10.1109/TNET.2015.2444657.
103. Sharghivand N, Derakhshan F, Siasi N. A comprehensive survey on auction mechanism design for cloud/edge resource management and pricing. IEEE Access. 2021;9:126502–29. doi:10.1109/ACCESS.2021.3110914.
104. Jiao Y, Wang P, Niyato D, Suankaewmanee K. Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks. IEEE Trans Parallel Distrib Syst. 2019;30(9):1975–89. doi:10.1109/TPDS.2019.2900238.
105. Kumar K, Liu J, Lu YH, Bhargava B. A survey of computation offloading for mobile systems. Mob Netw Appl. 2013;18(1):129–40. doi:10.1007/s11036-012-0368-0.
106. Wang J. Evolving the edge and the cloud: a hybrid computing paradigm. In: 2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C). China: IEEE; 2021. p. 718–21.
107. Fantacci R, Picano B. Performance analysis of a delay constrained data offloading scheme in an integrated cloud-fog-edge computing system. IEEE Trans Veh Technol. 2020;69(10):12004–14. doi:10.1109/TVT.2020.3008926.
108. Kar B, Yahya W, Lin YD, Ali A. Offloading using traditional optimization and machine learning in federated cloud-edge–fog systems: a survey. IEEE Commun Surv Tutor. 2023;25(2):1199–226. doi:10.1109/COMST.2023.3239579.
109. Okegbile SD, Maharaj BT, Alfa AS. A multi-user tasks offloading scheme for integrated edge-fog-cloud computing environments. IEEE Trans Veh Technol. 2022;71(7):7487–502. doi:10.1109/TVT.2022.3167892.

110. Hazra A, Adhikari M, Amgoth T, Srirama SN. Joint computation offloading and scheduling optimization of IoT applications in fog networks. IEEE Trans Netw Sci Eng. 2020;7(4):3266–78. doi:10.1109/TNSE.2020.3021792.
111. Bai W, Wang Y. Jointly optimize partial computation offloading and resource allocation in cloud-fog cooperative networks. Electronics. 2023;12(15):3224. doi:10.3390/electronics12153224.
112. Kimovski D, Mathá R, Hammer J, Mehran N, Hellwagner H, Prodan R. Cloud, fog, or edge: where to compute? IEEE Int Comput. 2021;25(4):30–6. doi:10.1109/MIC.2021.3050613.
113. Yadav R, Zhang W, Kaiwartya O, Song H, Yu S. Energy-latency tradeoff for dynamic computation offloading in vehicular fog computing. IEEE Trans Veh Technol. 2020;69(12):14198–211. doi:10.1109/TVT.2020.3040596.
114. Ding Y, Li K, Liu C, Li K. A potential game theoretic approach to computation offloading strategy optimization in end-edge-cloud computing. IEEE Trans Parallel Distrib Syst. 2021;33(6):1503–19. doi:10.1109/TPDS.2021.3112604.
115. Firouzi F, Farahani B, Panahi E, Barzegari M. Task offloading for edge-fog-cloud interplay in the healthcare Internet of Things (IoT). In: 2021 IEEE International Conference on Omni-layer Intelligent Systems (COINS). Barcelona, Spain: IEEE; 2021. p. 1–8.
116. Djigal H, Xu J, Liu L, Zhang Y. Machine and deep learning for resource allocation in multi-access edge computing: a survey. IEEE Commun Surv Tutor. 2022;24(4):2449–94. doi:10.1109/COMST.2022.3199544.
117. Agarwal S, Yadav S, Yadav AK. An efficient architecture and algorithm for resource provisioning in fog computing. Int J Inf Eng Electron Bus. 2016;8(1):48. doi:10.5815/ijieeb.2016.01.06.
118. Peng Q, Wu C, Xia Y, Ma Y, Wang X, Jiang N. DoSRA: a decentralized approach to online edge task scheduling and resource allocation. IEEE Internet Things J. 2021;9(6):4677–92. doi:10.1109/JIOT.2021.3107431.
119. Baranwal G, Kumar D, Vidyarthi DP. BARA: a blockchain-aided auction-based resource allocation in edge computing enabled industrial internet of things. Future Gener Comput Syst. 2022;135:333–47. doi:10.1016/j.future.2022.05.007.
120. Habiba U, Maghsudi S, Hossain E. A repeated auction model for load-aware dynamic resource allocation in multi-access edge computing. IEEE Trans Mob Comput. 2023;23(7):7801–17. doi:10.1109/TMC.2023.3338602.
121. Xiang Z, Zheng Y, Zheng Z, Deng S, Guo M, Dustdar S. Cost-effective traffic scheduling and resource allocation for edge service provisioning. IEEE/ACM Trans Netw. 2023;31(6):2934–49. doi:10.1109/TNET.2023.3265002.
122. Al-Masri E, Olmsted J. Enhancing resource provisioning across edge-based environments. In: 2020 IEEE International Conference on Big Data (Big Data). Atlanta, GA, USA: IEEE; 2020. p. 3459–63.
123. Anuradha C, Ponnavaikko M. Optimization of resource allocation in mobile edge computing using bio-inspired algorithm. In: 2021 International Conference on System, Computation, Automation and Networking (ICSCAN). Puducherry, India: IEEE; 2021. p. 1–4.
124. Xu J, Palanisamy B, Ludwig H, Wang Q. Zenith: utility-aware resource allocation for edge computing. In: 2017 IEEE International Conference on Edge Computing (EDGE). Honolulu, HI, USA: IEEE; 2017. p. 47–54.
125. Zhou H, Wu T, Chen X, He S, Guo D, Wu J. Reverse auction-based computation offloading and resource allocation in mobile cloud-edge computing. IEEE Trans Mob Comp. 2022;22(10):6144–59. doi:10.1109/TMC.2022.3189050.
126. Leon-Garcia F, Palomares JM, Olivares J. D2R-TED: data-domain reduction model for threshold-based event detection in sensor networks. Sensors. 2018;18(11):3806. doi:10.3390/s18113806.
127. Leon-Garcia F, Rodríguez-Lozano FJ, Olivares J, Palomares JM. Data communication optimization for the evaluation of multivariate conditions in distributed scenarios. IEEE Access. 2019;7:123473–89. doi:10.1109/ACCESS.2019.2936918.
128. Zhou K, Wang X, Cui S. Time synchronization in industrial internet of things: a survey. In: 2024 IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). Chongqing, China: IEEE; 2024. Vol. 7, p. 506–10. doi:10.1109/ITNEC60942.2024.10733131.
129. Deng Z, Yan Z, Huang H, Shen H. Energy-aware task scheduling on heterogeneous computing systems with time constraint. IEEE Access. 2020;8:23936–50. doi:10.1109/ACCESS.2020.2970166.
130. Wang Z, Shu Y, Liao H, Zhou Z, Lv L, Zhang L, et al. Edge-end cooperative network resource allocation with time synchronization awareness for federated learning based distributed energy regulation. IEEE Trans Smart Grid. 2025;2671–82.

131. Phan LA, Kim T, Kim T, Lee J, Ham JH. Performance analysis of time synchronization protocols in wireless sensor networks. Sensors. 2019;19(13):3020. doi:10.3390/s19133020.
132. Xue W, Luo Q, Wu H. Pattern-based event detection in sensor networks. Distrib Parallel Dat. 2012;30(1):27–62. doi:10.1007/s10619-011-7087-6.
133. Balakrishnan K, Dhanalakshmi R, Sinha BB, Gopalakrishnan R. Clock synchronization in industrial Internet of Things and potential works in precision time protocol: review, challenges and future directions. Int J Cogn Comput Eng. 2023;4(2):205–19. doi:10.1016/j.ijcce.2023.06.001.
134. Lin L, Liao X, Jin H, Li P. Computation offloading toward edge computing. Proc IEEE. 2019;107(8):1584–607. doi:10.1109/JPROC.2019.2922285.
135. Singh S, Chana I. Cloud resource provisioning: survey, status and future research directions. Knowl Inform Syst. 2016;49(3):1005–69. doi:10.1007/s10115-016-0922-3.
136. Chana I, Singh S. Quality of service and service level agreements for cloud environments: issues and challenges. In: Cloud comput: challenges, limitations and R&D solutions. Cham: Springer; 2014. p.51–72. doi:10.1007/978-3-319-10530-7_3.
137. Ali Z, Jiao L, Baker T, Abbas G, Abbas ZH, Khaf S. A deep learning approach for energy efficient computational offloading in mobile edge computing. IEEE Access. 2019;7:149623–33. doi:10.1109/ACCESS.2019.2947053.
138. Wang C, Dong C, Qin J, Yang X, Wen W. Energy-efficient offloading policy for resource allocation in distributed mobile edge computing. In: 2018 IEEE Symposium on Computers and Communications (ISCC). Natal, Brazil: IEEE; 2018. p. 00366–72.
139. Liu Y, Yu H, Xie S, Zhang Y. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. IEEE Trans Veh Technol. 2019;68(11):11158–68. doi:10.1109/TVT.2019.2935450.
140. Ssemakula JB, Gorricho JL, Kibalya G, Serrat-Fernandez J. Optimized provisioning technique of future services with different QoS requirements in multi-access edge computing. Comput Commun. 2024;224(3):42–59. doi:10.1016/j.comcom.2024.05.023.
141. Tang L, Xu M, Xu C, Ye K. Collaborative resource management and workloads scheduling in cloud-assisted mobile edge computing across timescales. arXiv:2405.20560. 2024.
142. Khaleel MI. Enhancing the resilience of error-prone computing environments using a hybrid multi-objective optimization algorithm for edge-centric cloud computing systems. Neural Comput Appl. 2024;36(18):10733–60. doi:10.1007/s00521-024-09636-8.
143. Ren X, Du H, Qiu C, Luo T, Liu Z, Wang X, et al. Dual-level resource provisioning and heterogeneous auction for mobile metaverse. IEEE Trans Mob Comput. 2024;23(11):10329–43. doi:10.1109/TMC.2024.3377211.
144. Jeremiah SR, Yang LT, Park JH. Digital twin-assisted resource allocation framework based on edge collaboration for vehicular edge computing. Future Gener Comput Syst. 2024;150(3):243–54. doi:10.1016/j.future.2023.09.001.
145. Wu Q, Wang W, Fan P, Fan Q, Wang J, Letaief KB. Urllc-awared resource allocation for heterogeneous vehicular edge computing. IEEE Trans Veh Technol. 2024;73(8):11789–805. doi:10.1109/TVT.2024.3370196.
146. Yang Z, Pan C, Wang K, Shikh-Bahaei M. Energy efficient resource allocation in UAV-enabled mobile edge computing networks. IEEE Trans Wirel Commun. 2019;18(9):4576–89. doi:10.1109/TWC.2019.2927313.
147. Martinez H, Rodriguez-Lozano FJ, León-García F, Palomares JM, Olivares J. Distributed Fog computing system for weapon detection and face recognition. J Netw Comput Appl. 2024;232(2):104026. doi:10.1016/j.jnca.2024.104026.
148. Svorobej S, Takako Endo P, Bendechache M, Filelis-Papadopoulos C, Giannoutakis KM, Gravvanis GA, et al. Simulating fog and edge computing scenarios: an overview and research challenges. Future Internet. 2019;11(3):55. doi:10.3390/fi11030055.
149. Margariti SV, Dimakopoulos VV, Tsoumanis G. Modeling and simulation tools for fog computing—a comprehensive survey from a cost perspective. Future Internet. 2020;12(5):89. doi:10.3390/fi12050089.
150. Zaballa EO, Franco D, Aguado M, Berger MS. Next-generation SDN and fog computing: a new paradigm for SDN-based edge computing. In: 2nd Workshop on Fog Computing and the IoT. Sydney, Australia: Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing; 2020. p. 9–1.
151. Alomari A, Subramaniam SK, Samian N, Latip R, Zukarnain Z. Resource management in SDN-based cloud and SDN-based fog computing: taxonomy study. Symmetry. 2021;13(5):734. doi:10.3390/sym13050734.

152. Shi C, Ren Z, He X. Research on load balancing for software defined cloud-fog network in real-time mobile face recognition. In: Communications and Networking: 11th EAI International Conference, ChinaCom 2016; 2016 Sep 24–26; Chongqing, China: Springer; 2018. p. 121–31.
153. Calheiros RN, Ranjan R, de Rose CA, Buyya R. Cloudsim: a novel framework for modeling and simulation of cloud computing infrastructures and services. arXiv:0903.2525. 2009.
154. Asir Antony Gnana Singh D, Priyadharshini R, Jebamalar Leavline E. Analysis of cloud environment using cloudsim. In: Artificial Intelligence and Evolutionary Computations in Engineering Systems: Proceedings of ICAIECES 2017. Singapore: Springer; 2018. p. 325–33.
155. Chowdhury MR, Mahmud MR, Rahman RM. Implementation and performance analysis of various VM placement strategies in CloudSim. J Cloud Comput. 2015;4:1–21.
156. Alshammari D, Singer J, Storer T. Performance evaluation of cloud computing simulation tools. In: 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). Chengdu, China: IEEE; 2018. p. 522–6.
157. Calheiros RN, Ranjan R, Beloglazov A, de Rose CA, Buyya R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw: Pract Exper. 2011;41(1):23–50. doi:10.1002/spe.995.
158. Gupta H, Vahid Dastjerdi A, Ghosh SK, Buyya R. iFogSim: a toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Softw: Pract Exper. 2017;47(9):1275–96. doi:10.1002/spe.2509.
159. Awaisi KS, Abbas A, Khan SU, Mahmud R, Buyya R. Simulating fog computing applications using iFogSim toolkit. Mobile Edge Comput. 2021;565–90. doi:10.1007/978-3-030-69893-5.
160. Rajagopal SM, Supriya M, Buyya R. FedSDM: federated learning based smart decision making module for ECG data in IoT integrated Edge–Fog–Cloud computing environments. Internet Things. 2023;22(3):100784. doi:10.1016/j.iot.2023.100784.
161. Khan EUY, Soomro TR. Overview of iFogSim: a tool for simulating fog networks of the future. In: 2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS). Karachi, Pakistan: IEEE; 2022. p. 1–4.
162. Rajagopal SM, Supriya M, Buyya R. Resource provisioning using meta-heuristic methods for IoT microservices with mobility management. IEEE Access. 2023;11:60915–38. doi:10.1109/ACCESS.2023.3281348.
163. Sonmez C, Ozgovde A, Ersoy C. Edgecloudsim: an environment for performance evaluation of edge computing systems. Trans Emerg Telecomm Technol. 2018;29(11):e3493. doi:10.1002/ett.3493.
164. Ashouri M, Lorig F, Davidsson P, Spalazzese R. Edge computing simulators for iot system design: an analysis of qualities and metrics. Future Internet. 2019;11(11):235. doi:10.3390/fi11110235.
165. Sonmez C, Ozgovde A, Ersoy C. Performance evaluation of single-tier and two-tier cloudlet assisted applications. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops). Paris, France: IEEE; 2017. p. 302–7.
166. Mechalikh C, Taktak H, Moussa F. PureEdgeSim: a simulation toolkit for performance evaluation of cloud, fog, and pure edge computing environments. In: 2019 International Conference on High Performance Computing & Simulation (HPCS). Dublin, Ireland: IEEE; 2019. p. 700–7.
167. Deshmukh S, Grover L. Implementation & analysis of pureedgesim. Int J Eng Appl Sci Technol. 2020;5(7):129–33. doi:10.33564/IJEAST.2020.v05i07.020.
168. Kokkonen H, Pirttikangas S, Lovén L. EISim: a platform for simulating intelligent edge orchestration solutions. arXiv:2311.01224. 2023.
169. Dudáš A, Demian D. Predictive analysis visualization component in simulated data streams. Discov Comput. 2024;27(1):12. doi:10.1007/s10791-024-09447-4.
170. Mechalikh C, Taktak H, Moussa F. PureEdgeSim: a simulation framework for performance evaluation of cloud, edge and mist computing environments. Comput Sci Inf Syst. 2021;18(1):43–66. doi:10.2298/CSIS200301042M.
171. Qayyum T, Malik AW, Khattak MAK, Khalid O, Khan SU. FogNetSim++: a toolkit for modeling and simulation of distributed fog environment. IEEE Access. 2018;6:63570–83. doi:10.1109/ACCESS.2018.2877696.

172. Qayyum T, Malik AW, Khan MA, Khan SU. Modeling and simulation of distributed fog environment using FogNetSim++. Fog Comput: Theory Pract. 2020;293–307. doi:10.1002/9781119551713.
173. Singh SP, Kumar R, Sharma A, Reddy SR, Vashisht P. Simulation and emulation tools for fog computing. Recent Adv Comput Sci Commun (Formerly: Recent Patents Comput Sci). 2022;15(3):315–22. doi:10.2174/2666255813999201002152003.
174. Kliazovich D, Bouvry P, Khan SU. GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. J Supercomput. 2012;62(3):1263–83. doi:10.1007/s11227-010-0504-1.
175. Agbo CC, David N, Asiegbu C. Evaluation of power saving models in energy-aware data centre using the GreenCloud simulator. Simulation. 2020;9:11.
176. Wadhwa M, Goel A, Choudhury T, Mishra VP. Green cloud computing-A greener approach to IT. In: 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). Dubai, United Arab Emirates: IEEE; 2019. p. 760–4.
177. Wrona Z, Ganzha M, Paprzycki M, Krzyżanowski S, Bădică A, Fidanova S. Scalability of extended green cloud simulator. In: 2024 International Conference on Innovations in Intelligent Systems and Applications (INISTA). Craiova, Romania: IEEE; 2024. p. 1–6.
178. Son J, Dastjerdi AV, Calheiros RN, Ji X, Yoon Y, Buyya R. Cloudsimsdn: modeling and simulation of software-defined cloud data centers. In: 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. Shenzhen, China: IEEE; 2015. p. 475–84.
179. Nyanteh AO, Li M, Abbod MF, Al-Raweshidy H. Cloudsimhypervisor: modeling and simulating network slicing in software-defined cloud networks. IEEE Access. 2021;9:72484–98. doi:10.1109/ACCESS.2021.3079501.
180. Andreoli R, Zhao J, Cucinotta T, Buyya R. CloudSim 7G: an integrated toolkit for modeling and simulation of future generation cloud computing environments. arXiv:2408.13386. 2024.
181. Son J, He TZ, Buyya R. CloudSimSDN-NFV: modeling and simulation of network function virtualization and service function chaining in edge computing environments. Softw: Pract Exp. 2019;49(12):1748–64. doi:10.1002/spe.2755.
182. Alcaraz-Velasco F, Palomares JM, Olivares J. GS3: a lightweight method of generating data blocks with shuffling, scrambling, and substituting data for constrained IoT devices. IEEE Internet Things J. 2024;11(15):25782–801. doi:10.1109/JIOT.2024.3395543.