# Machine Learning-Driven Resource Provisioning in Modern Cloud Environments: A Taxonomic Survey

Stefanus Albert Kosim, Bagus Jati Santoso*, Deka Julian Arrizki, Riki Mi'roj Achmad,
I Nyoman Gede Artadana Mahaputra Wardhiana, Royyana Muslim Ijtihadie
Department of Informatics Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

***Abstract*—Dynamic resource provisioning is a critical challenge in cloud computing, offering the necessary elasticity to guarantee reliable services within a usage-based payment framework. With the evolution of distributed systems, traditional threshold-based provisioning methods are increasingly inadequate for managing highly dynamic workloads. This inadequacy necessitates adaptive, machine learning (ML)-driven approaches capable of forecasting demand and autonomously optimizing scheduling. This survey presents a comprehensive review of recent ML-based resource provisioning strategies in cloud computing. Through a rigorous taxonomic analysis of 35 key studies, with a focus on developments from 2023 to 2025, the research categorizes existing work along two primary dimensions: ML methodology, including classical, deep learning, and advanced reinforcement learning, and optimization objectives, such as cost, Quality of Service (QoS), sustainability, and security-aware paradigms. The findings reveal a paradigm shift from reactive heuristics to proactive, hybrid forecasting-optimization models, Multi-Agent Reinforcement Learning (MARL), and serverless computing orchestration. Quantitative synthesis demonstrates that intelligence-driven interventions offer measurable improvements over traditional methods. For example, Deep Reinforcement Learning (DRL) models have reduced resource consumption by 10% and improved performance by 30%, while hybrid architectures have achieved user cost reductions of up to 44%. The survey concludes by discussing fundamental tradeoffs and identifying critical open challenges and future research directions in the edge-cloud continuum, including predictive container pre-warming and carbon-aware green AI orchestration.***

***Keywords*—Deep learning; cloud computing; machine learning; resource provisioning; taxonomy***

## I. Introduction

Dynamic Resource Provisioning (DRP) is the process of allocating and releasing computational resources (CPU, memory, storage, and bandwidth) based on real-time workload demands in cloud environments [1]. As distributed systems evolve, traditional threshold-based provisioning methods that rely on static, manually defined heuristics struggle to handle highly dynamic workloads. This model has reached a bottleneck due to the extreme variability of modern distributed tasks, necessitating adaptive, machine-learning (ML)-driven capabilities that can forecast demand and optimize scheduling autonomously [2]. To address the traditional method's bottleneck, DRP relies on cloud elasticity, allowing applications to automatically scale resources rather than relying on fixed, preassigned capacity [3]. DRP systems continuously monitor resource usage at fine-grained intervals and employ ML techniques to predict future load or classify it as high, medium, or low. Based on these predictions, systems can add or remove instances or adjust concurrency limits and resource allocation per instance [4]. Although DRP addresses several shortcomings of traditional methods, it continues to face challenges, including highly variable workloads [1], complex decision-making processes [5], limited and heterogeneous edge resources [6], slow scaling response [7], and inherent ML limitations [8]. In this study, we present a comprehensive survey of recent ML-based resource provisioning strategies in cloud computing.

This survey examines the ongoing paradigm shift from reactive heuristics to proactive, hybrid forecasting-optimization models, Multi-Agent Reinforcement Learning (MARL), and orchestration in serverless computing. Although previous surveys have addressed cloud resource provisioning, the rapid advancement of distributed systems in recent years has introduced new complexities that remain insufficiently explored. Earlier reviews primarily focused on foundational heuristics and traditional predictive modeling, but most predate the shift toward dynamic, decentralized orchestration. This work addresses this gap by offering an updated analysis with particular emphasis on state-of-the-art developments from 2023 to 2025. The primary contribution of this survey is its focus on the integration of Deep Reinforcement Learning (DRL) and MARL within emerging environments, such as serverless computing (Function-as-a-Service) and the edge-cloud continuum.

To systematically categorize and evaluate these provisioning strategies, it is necessary to first define the formal challenges constrained by specific system requirements that modern ML models seek to address. Latency is a primary concern, encompassing the need to minimize execution delays and address cold start latencies inherent in serverless environments to ensure rapid response times. Additionally, scalability must be maintained, which involves autonomously adjusting and distributing tasks across decentralized nodes during bursty or highly variable workloads while avoiding over-provisioning. Finally, Service Level Agreement (SLA) guarantees are critical, requiring strict adherence to negotiated Quality of Service (QoS) metrics, such as availability and 99th-percentile response times, where violations can result in financial penalties and degraded user experience.

By categorizing the state-of-the-art literature according to ML methodologies and optimization objectives, specifically cost and QoS, this survey offers stakeholders a structured framework for decision-making when deploying intelligent,

*Corresponding author.

autonomous scaling mechanisms in dynamic cloud environments.

This survey is organized as follows. Section II outlines the survey methodology, capturing the broad landscape of resource management and highlighting significant machine learning interventions. Section III reviews related work in resource provisioning. Section IV introduces foundational concepts, including objectives and existing provisioning methods. Section V presents the core taxonomy of machine learning-based approaches and discusses the findings in detail. Section VI concludes with a summary of the research.

## II. Survey Methodology

Systematic literature mapping is a structured methodology that surveys and organizes research on a broader topic. It builds an overview of what has been studied, how, and where the gaps are [9].

### A. Search Strategy and Date Range

A comprehensive search was conducted to capture the chronological evolution of resource provisioning strategies. Based on the publication dates of the collected literature, the search date range was defined as 2014 to 2025. This timeframe captures the foundational transition from classical predictive algorithms to MARL orchestration.

### B. Database

To guarantee the inclusion of high-quality, peer-reviewed content, literature retrieval was executed across the most prominent academic databases in electronic databases, namely IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and reputable open-access publishers, including MDPI and Wiley.

### C. Search Keywords

The search strategy used Boolean combinations of primary technical terms extracted from the core domain. The specific search keywords included are "Resource Provisioning", "Resource Allocation", "Auto-scaling", "Virtual Machine Migration", "Cloud Computing", "Edge Computing", "Serverless / FaaS", "Container Orchestration", "Machine Learning", "Deep Reinforcement Learning (DRL)", "Multi-Agent Reinforcement Learning (MARL)", and "SLA Guarantees".

### D. Inclusion and Exclusion Criteria

A multi-stage screening process was applied to the initial search results to identify the thirty-five key studies in our final analysis. To be included in this survey, papers were required to specifically address dynamic resource provisioning, allocation, or task scheduling within modern distributed deployment environments, including Cloud, Edge, Serverless, or the edge-cloud continuum. Furthermore, selected studies had to propose, utilize, or evaluate machine learning-driven techniques ranging from supervised learning to advanced reinforcement learning. Conversely, we excluded studies that relied strictly on static, threshold-based heuristics without an adaptive AI component, as well as papers that failed to provide rigorous quantitative evaluations of critical system constraints such as latency, cost, energy consumption, and QoS/SLA violations. Finally, research focusing solely on hardware-level networking was omitted to maintain a clear focus on high-level compute resource orchestration.

## III. Related Works

This section examines and explores prior survey research within the domain of resource provisioning in cloud computing. This review encompasses recent investigative efforts and methodologies that researchers have implemented to comprehend and regulate resource provisioning within cloud computing environments.

Zhang et al. [10] surveyed the algorithmic landscape of resource provisioning. This study evaluated over 150 articles and produced a comprehensive technical table addressing resource targets, optimization precision, and algorithm classifications, spanning both offline and online variants. Furthermore, the objectives of these algorithms are delineated from a business success perspective, emphasizing metrics such as cost, service quality, and utility. Vasoya et al. [11] conducted a survey of diverse resource provisioning techniques that integrate application and client requirements while adhering to Quality-of-Service (QoS) standards defined in service-level agreements (SLAs). This survey categorized provisioning techniques into six distinct types: execution cost, budget constraints, automated techniques, heuristics, prediction, and priority. Singh and Chana [12] provided a methodological analysis concerning the evolution of resource provisioning, various mechanism types, and a comparative assessment of their respective advantages and emergent issues. This research further underscores prior studies, the current state of the field, and future research trajectories for resource management in cloud computing. Focusing on predictive modeling, Kumar and Umamaheswari [13] surveyed models that facilitate the precise allocation of resources to applications at the optimal moment, thereby preventing SLA violations and upholding QoS standards. Their work identifies provisioning challenges and various machine learning-based prediction schemes designed to enhance provisioning effectiveness.

Varshney et al. [14] conducted a survey to examine research in resource allocation and availability, specifically focusing on QoS requirements. This paper details numerous challenges inherent to cloud computing, such as limited end-user control, security, privacy, and service quality concerns. A primary focus is QoS management, specifically the allocation of resources to guarantee service performance, availability, and reliability. Additionally, the authors discuss various QoS models proposed to address issues including latency, resource scheduling, network speed, power consumption, and load balancing. Sumalatha and Anbarasi [15] reviewed various optimization methods to identify how customers can minimize rental costs for virtual machine (VM) resources while maintaining cost-effectiveness. Shakarami et al. [16] presented a systematic and comprehensive survey of resource provisioning within edge/fog computing environments. They classified provisioning into five categories: framework-based, heuristic/metaheuristic-based, model-based, machine learning-based, and game-theoretic mechanism-based. This work offers comparisons through metrics, case studies, and evaluation tools. Li et al. [17] investigated key resource management issues, including

resource prediction, pricing, security, and service monitoring. Their research explores various approaches, such as bare metal supply chain models mediated by brokers and the application of seasonal autoregressive moving average models for probabilistic resource demand allocation. Overall, this body of literature provides an in-depth overview of provisioning approaches and identifies open issues requiring further exploration.

## IV. Foundations and Emerging Cloud Paradigms

Resource management is a fundamental part of cloud computing environments that offer dynamic resource allocation, provide reliable services, and are based on pay-as-you-go pricing [18]. Given that consumers may initiate multiple concurrent service requests, a structured methodology is required to manage the necessary computational assets. Resource management involves three main functions, namely resource provisioning, resource scheduling, and resource monitoring [12]. Specifically, resource provisioning entails efficient allocation constrained by particular requirements, while resource scheduling focuses on performance enhancement through the optimized timing of task distribution. In addition to this, resource monitoring involves real-time oversight of asset performance to detect fluctuating environmental conditions. To systematically categorize and evaluate resource provisioning strategies, it is essential to first establish the formal mathematical formulations, algorithmic architectures, and fundamental tradeoffs that govern machine learning interventions in cloud environments, as summarized in the taxonomy shown in Fig. 1.

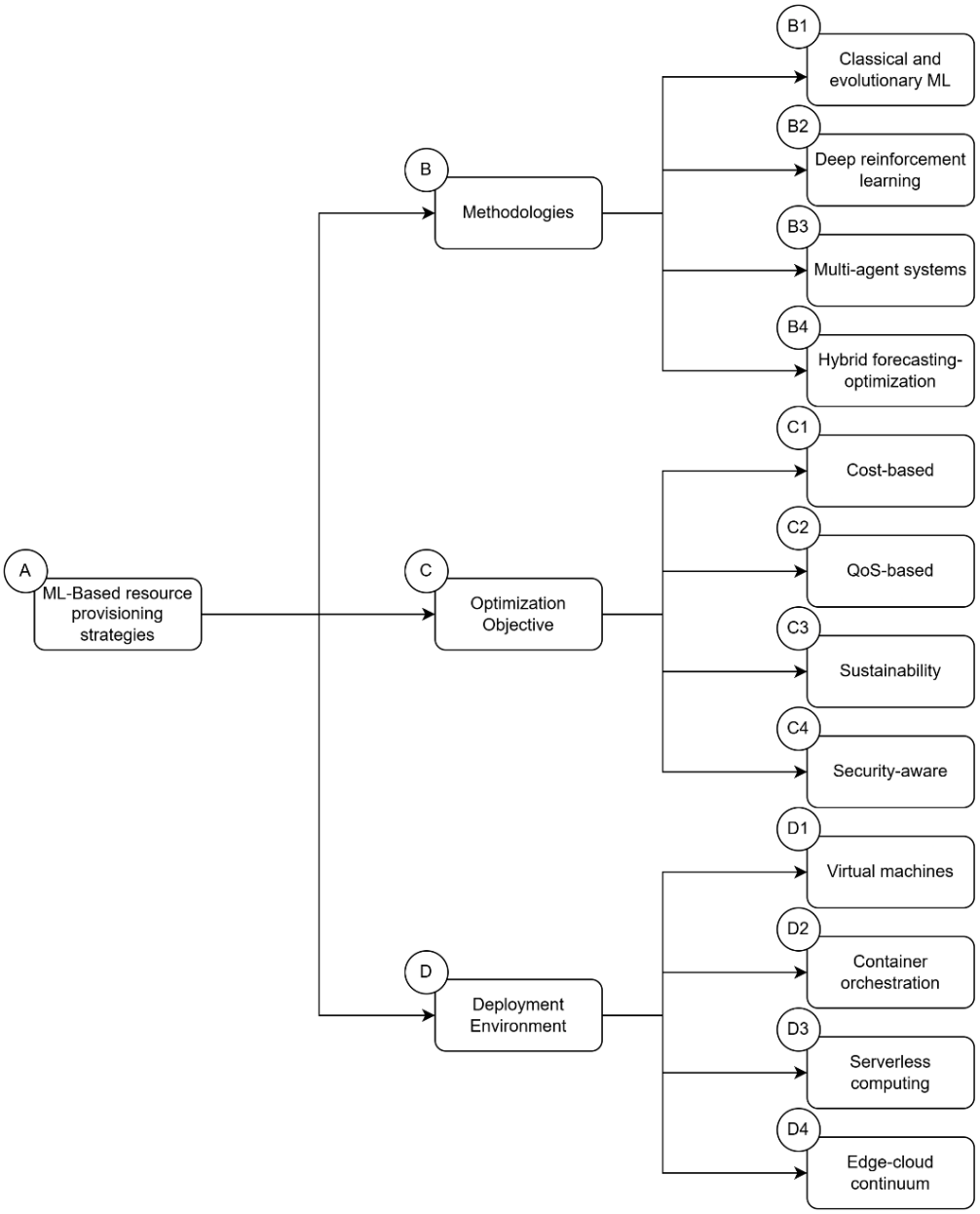


Fig. 1. Machine learning-based resource provision taxonomy.

### A. Resource Provisioning

The concept of resource provisioning originated within Grid computing environments. In modern cloud contexts, this process remains a complex challenge due to the inherent limitations of available physical assets. Effective provisioning is fundamentally dependent on the QoS requirements specified by cloud applications [14]. Determining the appropriate resource volume for specific workloads is an intricate task; identifying the optimal resource-workload pair based on QoS constraints constitutes a significant research problem in the field. Consequently, existing investigations frequently employ minimizing execution time as a primary optimization criterion to formulate solutions to this problem.

### B. Resource Provisioning Objectives

There are several objectives of resource provisioning that serve as key parameters within a cloud computing environment. These parameters guide resource allocation and capacity adjustment, making a deep understanding of them essential for designing effective provisioning strategies.

Response time is the total duration required to respond to a user's service request. Speed is critical for operational efficiency and user satisfaction, especially as workload requirements change rapidly. Provisioning algorithms must be designed to provide necessary resources in a very short time.

Cost optimization: From the perspective of cloud users, minimizing costs is a primary requirement [19]. Strategies focus on cost management to ensure financial efficiency and provide added value at a controlled expenditure level.

Execution time: This refers to the total time needed to complete a task or process. Methods must be designed to achieve execution results with minimal time [20].

Reduced execution time directly lowers operational costs and improves the user experience.

Fault tolerance: This is the system's ability to maintain normal operations despite component failures. For instance, if a server fails, the algorithm should automatically detect the downtime and redistribute tasks to operational nodes [21]. Service continuity and reliability are critical features that ensure system availability and resilience.

Reduction of SLA violations: Algorithms are measured by their ability to minimize violations of the established Service Level Agreement (SLA) [22]. This requires careful resource allocation and task management to ensure every service demand complies with agreed-upon parameters.

Reduced power consumption: Efficient Virtual Machine (VM) placement and migration techniques are necessary to reduce power usage [23]. Intelligent placement strategies aim for energy efficiency, supporting sustainability and environmental goals while maintaining resource availability for workloads.

### C. Resource Provisioning Methods

Cloud resource provisioning strategies can be divided into two main categories: non-machine learning (non-ML) and machine learning (ML)-based methodologies. Non-ML approaches rely on static rules and manually defined heuristics, typically derived from empirical experience [24]. In these approaches, resource allocation is determined by fixed parameters, including instantaneous workload, temporal constraints, and application priority. The main advantage of non-ML methods is their simplicity of implementation and transparency, as the provisioning logic can be easily configured and understood. However, these methods have several limitations. They are not able to adapt dynamically to changes in cloud environments, and their decisions are often static and less optimal in terms of performance and resource efficiency, especially when workloads are highly variable. As a result, non-ML provisioning is most suitable for stable, predictable environments where resources can be managed using consistent business rules and fixed policies.

In contrast, ML-based provisioning strategies have been the focus of recent research. These strategies can be grouped into classical and evolutionary ML, deep reinforcement learning, multi-agent systems, and hybrid forecasting-optimization. Classical and evolutionary machine learning techniques offer distinct but complementary approaches to resource provisioning. Classical machine learning is predicated upon well-established algorithms, including regression, classification, and clustering, to inform provisioning decisions [25]. These methodologies present considerable benefits owing to their elevated interpretability, facilitating a clear comprehension of the model's foundational rationale even when subjected to training on relatively limited datasets. However, they typically require manual feature engineering and can struggle to accurately capture highly nonlinear data patterns. To address more complex and dynamic environments, evolutionary and nature-inspired approaches utilize algorithms based on biological evolution and natural phenomena [26]. Techniques such as genetic algorithms and ant colony optimization mimic processes like natural selection and genetic reproduction to optimize resource distribution. While these evolutionary methods are highly adaptable, they introduce trade-offs, often involving higher computational complexity and limited interpretability compared to classical models.

Deep reinforcement learning (DRL) is a sophisticated machine learning paradigm that integrates deep neural networks with reinforcement learning, enabling an agent to learn optimal decision-making strategies through trial-and-error while striving to maximize cumulative long-term rewards. In the context of reinforcement learning, an agent diligently monitors its environment, executes actions, acquires rewards, and systematically refines its policy; the application of deep learning techniques facilitates the scalability of this process to intricate, high-dimensional challenges, such as those encountered in cloud and edge computing systems [27]. The main advantage of DRL is its ability to automatically learn adaptive and near-optimal policies in dynamic and uncertain environments without relying on predefined rules or accurate system models [28]. It can jointly optimize multiple objectives such as latency, energy efficiency, cost, and quality of service, making it suitable for modern distributed systems [29]. However, DRL also has disadvantages, including high training complexity, large data requirements, and potential instability during learning in real-time systems [27]. In the context of resource provisioning, DRL is widely used for

dynamic resource allocation, auto-scaling, task offloading, and virtual machine management in cloud and edge environments, where it improves resource utilization, reduces latency, and enhances energy efficiency compared to traditional rule-based or heuristic approaches [30].

A multi-agent system (MAS) is a distributed system made up of multiple autonomous agents that interact, cooperate, or negotiate with each other to solve complex problems. Agents are independent entities capable of perceiving their environment, making decisions, and acting toward specific goals. MAS is especially suitable for open, dynamic, and distributed environments where centralized control is difficult [31]. In resource provisioning, multi-agent systems are widely used in cloud computing, fog computing, and edge computing to dynamically allocate, scale, and manage resources such as CPU, memory, bandwidth, and virtual machines. MAS enables distributed decision-making for elastic scaling and workload management. For instance, collaborative multi-agent reinforcement learning has been shown to improve cloud resource scaling, SLA compliance, and scheduling efficiency [32]. However, MAS increased communication overhead, coordination complexity, and potential security and monitoring challenges due to decentralization [33].

Hybrid forecasting-optimization is a method that combines demand prediction (forecasting) with decision-making models (optimization) to allocate resources more effectively under uncertainty. Instead of forecasting and optimizing separately, the forecast is directly integrated into the optimization model to improve final decisions. Research shows that this approach improves decision quality and cost efficiency because predictions are aligned with operational objectives rather than just minimizing forecast errors. Its advantages include better cost savings, improved robustness, and more efficient resource use, while disadvantages include higher computational complexity and risk of error propagation if forecasts are inaccurate [34].

### *D. Deployment Environment*

The deployment environment establishes the foundational architecture and operational mechanisms through which computational resources are provisioned, managed, and scaled. As applications evolve to become more distributed and dynamic, resource allocation strategies must adapt to the distinct abstraction layers and operational models of the underlying infrastructure. Within this context, the deployment environment is grouped into four primary paradigms, each presenting unique mechanisms, challenges, and optimization objectives: traditional Virtual Machines (VMs), container orchestration platforms, Serverless/Function-as-a-Service (FaaS) models, and the distributed edge-cloud continuum. Each paradigm dictates specific approaches to scheduling, capacity planning, and runtime adaptation, ultimately influencing resource utilization, latency, and cost efficiency.

Resource provisioning on VMs involves deciding how many VMs to create what sizes they should be, and on which physical servers they should run, so that applications get enough CPU, memory, storage, and network without wasting data-center capacity. Virtualization lets multiple VMs share one physical machine (PM), increasing utilization but making placement and allocation decisions critical for performance, energy, and cost [35], [36]. There are two allocations in VM deployment, namely static allocation and dynamic allocation. In static allocation, VMs are given fixed resources and placements; simple but often underutilized hardware and cannot adapt to workload changes [10], [37]. On the other hand, dynamic allocation adjusts resource or VM placement at runtime using monitoring and migration to track demand [36], [38]. This improves utilization and energy efficiency but adds overhead and complexity [36], [37]. Well-designed VM provisioning and deployment schemes aim to increase resource utilization and reduce idle capacity, lower energy consumption by consolidating VMs and turning off unused servers, reduce response time and SLA violations through better placement and dynamic scaling, and control costs by minimizing the number and size of VMs while maintaining QoS [1], [36], [39], [40].

Container orchestration platforms automate how resources are provisioned and used when deploying containers across clusters of machines. This involves scheduling, scaling, and runtime reallocation of CPU, memory, storage, and network to meet performance, cost, and reliability goals [41], [42], [43]. At deployment, the orchestrator must make decisions regarding the location and resource allocation for containers, including CPU and memory reservations for scheduling and isolation [41], [43]. Modern scheduling policies account for factors such as heterogeneity, priorities, and job structures [44]. Cost-aware strategies can optimize resource allocation, potentially reducing cloud costs by 23-32% through efficient container packing and shutting down underutilized instances [44], with similar approaches applicable in multi-cloud environments for energy and host count reductions [45]. Runtime scaling and adaptation involves continuous provisioning to address dynamic workloads, employing key strategies to enhance performance and cost efficiency. One of the primary approaches is autoscaling, which adjusts application and cluster size both horizontally and vertically to meet demand [43], [44]. Additionally, online resource re-dimensioning is utilized to adapt CPU budgets at runtime for real-time containers, thereby improving predictability and optimizing resource usage [46]. Another important strategy is layer-aware and cost-aware orchestration, which focuses on optimizing joint request scheduling, placement, and server wake-up decisions in edge computing, resulting in approximately a 20% performance improvement [47].

Serverless/FAAS transfers the responsibility of provisioning and deploying compute resources from developers to providers. Developers upload functions and set triggers, while the platform manages CPU, memory, and containers/VMs on demand, scaling automatically and charging based on usage [48], [49]. Applications are divided into stateless functions that create workflows or directed acyclic graphs (DAGs) [49], [50]. Users provide function code, event triggers, and resource hints, while the provider oversees runtime orchestration, monitoring, and scaling. Functions operate in isolated containers or micro-VMs, which can be warm (active between invocations) or cold (created as needed), impacting latency and resource consumption [51]. From the provider's perspective, resource management involves a pipeline that includes workload modeling, resource provisioning, scheduling, scaling, and

deallocation. Key considerations include predicting function behavior for proactive provisioning, determining the number and type of nodes for cost-effective hosting, configuring function resources to optimize utilization, deciding on function placement to enhance performance, and dynamically scaling and deallocating resources based on real-time load while minimizing cold starts and idle resources [49], [52], [53].

Resource provisioning and deployment within the edge-cloud continuum involves strategically allocating compute, storage, and network resources across diverse nodes, from edge devices to centralized data centers, with the primary objective of achieving optimal latency, reliability, security, and cost-effectiveness despite challenges such as mobility and limited resources at the edge. The architectural framework includes servers located at the network edge, such as near 5G base stations, gateways, and fog nodes, as well as centralized cloud data centers, enabling flexible service deployment through microservice or function-based architectures. Key challenges in provisioning encompass determining the optimal node for hosting each microservice or function while adhering to constraints related to CPU, memory, bandwidth, and latency, assigning CPU shares and performance tiers to meet QoS and Service Level Agreements (SLA) while minimizing resource and energy consumption, and dynamically adapting resource allocation in response to changes in load, network conditions, or user locations. The typical objectives involve trade-offs between latency and QoS, cost and energy efficiency, as well as reliability and security, with a strong emphasis on meeting SLAs [54], [55], [56].

## V. Results and Discussion

This section presents the results of the literature survey, structured according to a three-dimensional taxonomy. The taxonomy categorizes existing research by machine learning methodology, optimization objectives, and deployment environment. Thirty-five key studies were mapped to this framework to ensure a comprehensive and systematic review. This structure facilitates the assessment of how various algorithms address the complex challenges inherent in modern cloud infrastructure. The following subsections first outline the current state-of-the-art, followed by an analysis of operational impacts and technical trade-offs identified in the literature.

### A. Taxonomic Categorization of Literature

This subsection presents the results of the literature survey by organizing the findings using a formal taxonomy and providing an analysis of the main results. We first conducted a taxonomic analysis of the existing literature to create a systematic framework for categorizing the methodologies and experimental insights found in the reviewed survey papers. The selected literature is formally organized in, which provides a multidimensional overview of the algorithms, deployment environments, and optimization objectives investigated in this survey. This table serves as the foundational evidence for the thematic categorization presented in the following subsections. The following discussion interprets the taxonomic results and explains their broader implications. We provide an analysis of current trends in resource provisioning, assess the effectiveness of different strategies, and identify gaps in the research that require further study.

*1) Classical and evolutionary approaches:* Classical and evolutionary approaches have traditionally been deployed in virtual machine environments to address fundamental provisioning goals. To meet cost-based objectives, nature-inspired methods such as particle swarm optimization (PSO) and simulated annealing (SA) optimize time and monetary expenses [21]. Other techniques, including the shuffled frog leaping algorithm (SFLA) and ubiquitous binary search (UBS), are applied to improve resource efficiency [57]. Addressing sustainability, spider monkey optimization (smo) minimizes execution costs and power consumption in uncertain environments [58] and is chosen for its effectiveness in uncertain environments. Additional evolutionary approaches include utilizing genetic algorithms with fuzzy c-means to improve QoS-based metrics through workload prediction [59] and combining fuzzy c-means with PSO to optimize execution time and memory [60]. Traditional ML approaches include greedy and heuristic algorithms to reduce memory and storage requirements [61], as well as linear regression and Bayesian learning for cost optimization [62]. Bayesian optimization has also been applied to resource prediction, resulting in higher average virtual machine (VM) utilization [63].[57][58][59][60][61][62][63].

For predicting VM resource requirements to ensure strict QoS adherence, Support Vector Regression (SVR), Neural Networks (NN), and LR have shown success in increasing throughput [64], while models like Random Forest (RF), XGBoost, and LR are applied to predict demand-based usage [65]. In Elastic Cloud services, Support Vector Machine (SVM) models outperform simple forecasting methods to approach optimal allocation [23]. Clustering algorithms like K-means facilitate fault tolerance in sustainable systems [66], and when integrated with Convolutional Neural Networks (CNNs), K-means reduced MySQL deployment costs by 48% on single-tenant architectures [4]. Hybrid classical methods combine NN, LR, and RepTree to improve resource utilization [67], or use Artificial Swarm Intelligence (ASI), SVM, K-Nearest Neighbors (KNN), and Decision Trees to maximize VM resource usage and decrease Service Level Agreement (SLA) violations [25]. Furthermore, standalone neural network architectures classified within this methodology include Diffusion Convolutional Recurrent Neural Networks (DCRNN) for predicting future demand [68], Multi-Layer Perceptrons (MLP) for reducing execution time [69], Learning Automata (LA) for cost-effective cloud infrastructure utilization [70], and Long Short-Term Memory (LSTM) models for optimizing response times against simple forecasting [71].

*2) Hybrid forecasting-optimization models:* Moving beyond static VM provisioning, DRL actively learns policies suited for highly volatile environments like serverless computing and the edge-cloud continuum. In traditional elastic provisioning, short-term memory Q-learning optimizes the number of allocated VMs to decrease CPU usage [72]. Shifting to serverless architectures where cost and QoS are tightly coupled, [74] introduced Freyr+, a system utilizing proximal policy optimization (PPO) to dynamically harvest idle

resources from over-provisioned functions, accelerating invocations and reducing tail latency. targeting sustainability at the edge, [78] designed atom, employing DDPG and RDPG agents to predict cold start latency and optimize energy consumption [72][74][78].

Crucially, DRL allows for the integration of emerging security-aware objectives. [79] proposed an Action-constrained DRL (DQN) framework to ensure resources are securely allocated in multi-cloud edge networks while minimizing system costs. To enhance prediction accuracy before allocation, [80] introduced the DRAW method, integrating Workload-Time Windows directly into the state space of a Deep Q-Network (DQN) to balance QoS and costs. Addressing the distributed nature of the edge-cloud continuum, [30] utilized NESRL-MRM that combining DRL with Neuroevolution of Augmenting Topologies (NES) to achieve balanced multi-dimensional resource allocation and solve fragmentation. Additionally, [85] developed an Actor-critic based DRL (H-A2C) combined with PPO to determine the optimal deployment across hybrid with server and serverless environments, driving down user costs by up to 44%.

*3) Deep reinforcement learning:* As cloud clusters scale, particularly in container orchestration and edge environments, single-agent models face performance bottlenecks. This drives the shift toward multi-agent deep reinforcement learning (MADRL) to maintain strict QoS and sustainability targets across decentralized nodes. To handle dynamic microservice demands, [75] introduced drpc, a distributed RL framework using the TD3 algorithm that scales linearly across container-based clusters to reduce average response times and failed requests. Focusing on energy efficiency, [76] proposed a MADRL framework utilizing a DTS hyper-heuristic where multiple agents collaboratively optimize container placement to minimize resource utilization and avoid sla violations in data centers. In serverless edge computing, [86] proposed CSODQN, a multi-agent approach utilizing importance sampling-based double dueling DQN to manage offloading and maintain a warming pool. This effectively balances the tradeoff between cold start frequency (QoS) and resource utilization (cost and sustainability) [75][76][86].

*4) Multi-agent and distributed orchestration:* Hybrid methodologies combine the predictive power of time-series forecasting (to guarantee QoS) with the decision-making capabilities of RL algorithms (to optimize cost and sustainability), spanning the entire spectrum of deployment environments. In serverless setups, [73] proposed an LSTM-ppo approach, using recurrent learning to intelligently autoscale and maximize throughput with minimal resources. For sustainable container orchestration, [77] designed greenkube, which integrates DRL, graph neural networks (GNN), and LSTM to minimize energy consumption and prevent over-provisioning at the edge and cloud. spanning the broader computing continuum, [81] introduced an intent-driven orchestration model combining DQN, ARIMA forecasting, and Mixed-Integer Linear Programming (MILP) to minimize latency, transformation costs, and power consumption [73][77][81].

A specific focus on dynamic VM migration is seen in [82], which combines DQN and LSTM to automate allocation strategies, saving energy and reducing downtime. Similarly, [83] chained LSTM forecasting with DQN and PSO decision-making to schedule CPU, memory, and bandwidth in Alibaba Cloud ECS clusters, lowering costs by 26.6%. Finally, operating within Kubernetes environments, [84] developed a forecast-based autoscaler that integrates LSTM predictions directly into a model-free DQN scaling logic, adjusting replicas proactively to ensure SLO compliance while reducing resource consumption.

### B. Critical Synthesis and Discussion

Following the thematic categorization of the literature, this subsection shifts from a descriptive overview to a critical synthesis of performance. The taxonomic results are interpreted to identify broader implications and emerging trends in resource provisioning, with particular attention to paradigm shifts in decentralized environments. This discussion assesses the effectiveness of various strategies, establishes a quantitative comparative benchmark, and rigorously evaluates the trade-offs between computational overhead and system elasticity to address gaps identified in current research.

*1) Unified metric framework and objective trade-offs:* To enable a standardized comparison across the reviewed literature, we define a unified metric framework that categorizes provisioning objectives into three principal axes. The first axis, cost, includes both monetary expenditure and resource utilization. The second axis, QOS, encompasses latency, response time, and compliance with service-level agreements. The third axis, energy, addresses power consumption and sustainability considerations. These objectives are inherently conflicting, resulting in fundamental trade-offs that must be addressed by machine learning models. The balance between QoS and cost or energy, for example, often necessitates the maintenance of pre-warmed container pools or idle resources to reduce serverless cold starts. Although this approach improves response times, it also increases financial costs and energy consumption. Conversely, the tension between energy and performance becomes apparent in the use of aggressive power optimization strategies, such as virtual machine consolidation or scaling to zero during periods of low traffic. While these strategies promote sustainability, they also elevate the risk of service-level agreement violations during sudden workload surges. Considering these findings, the trade-off between complexity and efficiency is also significant. Advanced decentralized models, such as multi-agent reinforcement learning, can reduce orchestration bottlenecks in large clusters. However, these models introduce substantial communication overhead and increased training complexity when compared to more traditional heuristic approaches.

*2) Temporal evolution of ML paradigms:* Analysis of publication data from 2014 to 2025 demonstrates a clear paradigm shift in resource provisioning methodologies, which

can be categorized into three distinct chronological phases. Fig. 2 illustrates the distribution of these methodologies throughout the survey period. During the era of classical ML and heuristics (2014-2022), foundational predictive algorithms and heuristic approaches dominated the literature for nearly a decade. From 2014 to 2020, all surveyed papers relied exclusively on classical machine learning or heuristic methods. Even as research volume increased toward the end of this period, classical approaches remained predominant, comprising 8 of 9 published papers in 2021 and 2022.

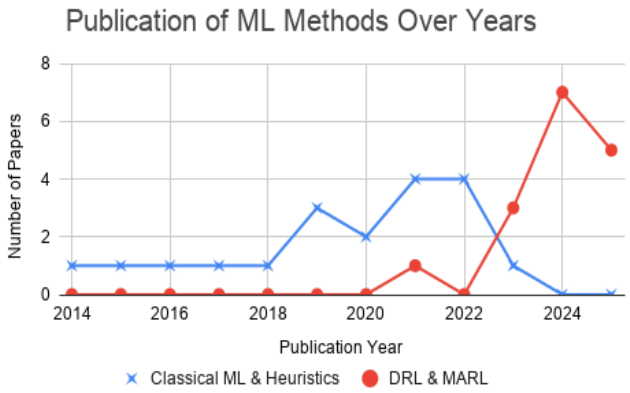


Fig. 2. Temporal evolution of resource provisioning methodologies.

In 2023, the orchestration landscape experienced a crucial inflection point, indicating a transition toward more advanced intelligent systems. For the first time, DRL and MARL surpassed traditional methods, accounting for 3 of the 4 papers published that year. This transition corresponds to the increasing need to manage complex, variable system states, where traditional rule-based and predictive approaches are less effective at balancing QoS and resource costs.

Recent literature from 2024 to 2025 demonstrates the complete dominance of DRL and MARL methodologies. During this period, classical machine learning and heuristic approaches are absent from the surveyed state-of-the-art research. DRL and MARL account for all 12 papers published in these years. This quantitative trend substantiates the assertion that, as microservices and modern deployment environments have expanded, centralized decision-making and classical supervised learning have become significant system bottlenecks. As a result, the research community has rapidly shifted toward DRL and MARL frameworks, whose experience-based learning strategies have proven highly effective for autonomous adaptation in dynamic, nonlinear cloud computing scenarios.

*3) Quantitative synthesis and comparative benchmark:* To quantify the effectiveness of modern ML interventions, we derived a comparative benchmark from the empirical performance claims reported in the 35 studies synthesized in Table I. The synthesis presented in Table II highlights the specific percentage improvements in latency, cost, and energy efficiency achieved by modern ML paradigms compared to traditional baseline models.

TABLE I. RESEARCH PAPERS ON MACHINE LEARNING-BASED RESOURCE PROVISIONING

| Papers | Algorithm / Method | Environment | Objective | Result | Taxonomy Labels |
|---|---|---|---|---|---|
| [26] | PSO and SA | Multi-tier Cloud Computing | Improving resource provision with minimum time and cost | Effectively reduces the time required to provide resources, which leads to cost reductions | B1, C1, D1 |
| [61] | Greedy and heuristic | Clouds | Decreased resource usage (core, memory, and storage) | Effectively captures dynamic demand, provides computing resources to match demand, and generates high revenue with low execution time | B1, C1, C2, D1 |
| [64] | SVR, NN and LR | Web VM Resource Provisioning | Improving performance in throughput and response time | The use of SVR provides the best performance in reducing response time and increasing throughput | B1, C2, D1 |
| [69] | MLP | Heterogeneous infrastructures | Meeting user demands in terms of cost and execution time | Increased throughput and reduced execution time | B1, C1, C2, D1 |
| [70] | LA | Cloud infrastructure | Minimizing cloud infrastructure costs | The proposed method has the most cost-effective (minimum) use compared to other methods | B1, C1, D1 |
| [67] | NN, LR, and RepTree | Cloud Computing | Improving resource utilization | Decrease in CPU usage | B1, C1, D1 |
| [23] | SVM | Elastic Cloud services | Improving QoS | The SVM model outperforms simple forecasting methods and approaches optimal allocation | B1, C2, D1 |
| [60] | Fuzzy C-means and PSO | Cloud computing environment | Improving execution time, cost, and memory | Improvements of 20.62, 40.14, and 39.92% in execution time, cost, and memory usage | B1, C1, C2, D1 |
| [63] | Bayesian optimization | Cloud Computing | Optimizing VMs so that resources are used effectively to improve QoS | Increase in average VM utilization ratio | B1, C2, D1 |
| [59] | Genetic and Fuzzy C-Means | Cloud workloads | Improve QoS by predicting workloads so that VM scaling can be performed effectively | Reduction in power consumption, execution time, cost, and latency | B1, C1, C2, C3, D1 |
| [65] | RF, XGBoost, and LR | Cloud Resource Provisioning System | Predict resource usage based on demand | More optimal CPU usage | B1, C1, D1 |

| | | | | | |
|---|---|---|---|---|---|
| [66] | K-mean | Sustainable Cloud Computing Systems | Developing mechanisms for tolerance and fault tolerance | Reduction in downtime and power consumption | B1, C2, C3, D1 |
| [57] | SLFA-UBS | Multi-cloud Computing | Cost and resource efficiency | Decrease in execution time and increase in throughput | B1, C1, D1 |
| [68] | DCRNN | Autonomic cloud computing environment | Predict future demand for resource allocation | Decrease in CPU usage on VMs | B1, C1, D1 |
| [71] | LSTM | Cloud Services | Improving QoS | The LSTM model outperforms SVM and other forecasting methods, and approaches optimal allocation | B1, C2, D1 |
| [58] | SMO | Cloud platforms | Minimizing time, cost, and energy consumption | Reduction in execution costs, power consumption, and task rejection | B1, C1, C2, C3, D1 |
| [62] | LR and Bayesian | Service-based cloud applications | Reducing expenses and improving resource utilization | Better resource utilization and shorter response times | B1, C1, D1 |
| [25] | ASI, SVM, KNN and Decision Tree | Virtual machine allocation and migration | Maximizing resource usage on VMs to decrease power usage and SLA violations | Decrease in power usage and SLA violations | B1, C1, C2, C3, D1 |
| [4] | K-means and CNN | Cloud resource provisioning | Reducing costs while improving application performance | Reduced MySQL deployment costs by 48% on a single-tenant architecture | B1, C1, C2, D1 |
| [72] | Short-Term memory Q-Learning | Elastic provisioning | Optimizing the number of VMs allocated to the workload to decrease the number of provisioned VMs and CPU usage | Decrease in the number of provisioned VMs and decrease in CPU usage | B2, C1, D1 |
| [73] | LSTM-PPO | Serverless Computing / FaaS | Intelligent autoscaling to maximize throughput while using minimal resources (Cost & QoS-based) | Improved throughput by 18% and function execution by 13% compared to baseline | B4, C1, C2, D3 |
| [74] | DRL utilizing PPO | Serverless Computing platforms | Dynamically harvest idle resources from over-provisioned functions to maximize utilization (Cost & QoS-based) | Harvested 38% of idle resources, accelerated 39% of invocations, and reduced 99th-percentile latency by 26% | B2, C1, C2, D3 |
| [75] | Distributed RL (TD3 algorithm) | Container-based cloud microservice clusters | Scalable autoscaling to handle dynamic demands, reducing response time and failed requests (QoS & Scalability-based) | Reduced average response time by 15% and decreased failed requests by 24% compared to baselines | B3, C2, D2 |
| [76] | MADRL and DTS hyper-heuristic | Container-based cloud environments / Data centers | Maximizing energy efficiency, avoiding SLA violations, and minimizing resource utilization (Energy, Cost, & QoS-based) | Achieved significantly lower energy consumption, reduced memory/CPU utilization, and faster response times | B3, C1, C2, C3, D2 |
| [77] | DRL, GNN, and LSTM | Container Orchestration platforms (Edge and Cloud) | Minimizing energy consumption and preventing over-provisioning while maintaining strict QoS (Energy & QoS-based) | Outperformed Kubernetes HPA by reducing average latency, task execution time, and active node usage | B4, C2, C3, D2, D4 |
| [78] | DRL (DDPG and RDPG) | Serverless Edge Computing Environments | Predict cold start latency and reduce occurrence frequency to optimize energy consumption (Energy & QoS-based) | DDPG model outperformed baselines in latency/request prediction with superior stability and convergence | B2, C2, C3, D3, D4 |
| [79] | Action-constrained DRL (DQN) + Security Mechanisms | Serverless multi-cloud edge computing networks | Securely allocate resources to minimize costs, satisfy latency requirements, and ensure privacy (QoS, Cost, & Security-based) | Reduced the system cost by 24.4% and 22.6% compared to standard DQN and DQN+NN methods respectively | B2, C1, C2, C4, D3, D4 |
| [80] | DRL utilizing a DQN (DRAW method) | Cloud-based software services | Adaptively allocate resources to balance QoS and reduce overall resource costs (QoS & Cost-based) | Prediction accuracy reached 90.69%, outperforming classic ML and standard DQN methods by 3% to 13% | B2, C1, C2, D1 |
| [81] | DQN, ARIMA, and MILP | Computing Continuum (Multi-cluster Edge and Cloud) | Intent-driven orchestration to minimize latency, transformation cost, and power consumption (QoS, Energy, & Cost-based) | Achieved only 1.52% QoS violations, reduced transformation costs by 22%, and power consumption by 24.2% | B4, C1, C2, C3, D3, D4 |
| [30] | DRL combined with NES (NESRL-MRM) | Distributed Data Centers in the edge-cloud continuum | Multi-dimensional allocation to achieve balanced utilization across resource types (Balanced utilization) | Significantly improved balanced resource utilization and demonstrated an 85% improvement in time efficiency | B2, C1, D4 |
| [82] | DRL utilizing DQN and LSTM | Cloud computing environments focusing on VM Migration | Optimizing migration to allocate resources, maximize utilization, and minimize energy and downtime (Cost, Energy, & QoS-based) | Automated VM migration strategies significantly outperformed manual settings or static rule-based allocation | B4, C1, C2, C3, D1 |
| [83] | LSTM, DQN, and PSO | Cloud Computing Environments | Dynamically schedule CPU/memory/bandwidth to improve utilization, reduce | Enhanced utilization by 32.5%, reduced average response time by 43.3%, and lowered costs by 26.6% | B4, C1, C2, D2 |

| | | | | | |
|---|---|---|---|---|---|
| | | (AliCloud ECS cluster) | latency, and minimize expenses (Cost, QoS, & Resource-based) | | |
| [84] | Model-free RL utilizing DQN and LSTM | Containerized Cloud Computing / Kubernetes | Proactively predict request rates to adjust replicas, ensuring SLO compliance while minimizing usage (QoS & Resource-based) | Outperformed Kubernetes HPA, reducing resource consumption by up to 10% and improving performance by up to 30% | B4, C1, C2, D2 |
| [85] | Actor-critic based DRL (H-A2C) with PPO | Serverless and Serverful (IaaS/VM-based) Hybrid Environments | Determine optimal deployment environment and execution node to optimize latency and cost (Cost & QoS-based) | Outperformed baselines, achieving user cost reductions of up to 44% and relative response time improvements of up to 11% | B2, C1, C2, D1, D3 |
| [86] | Multi-agent DRL utilizing CSODQN | Serverless Edge Computing / Cloud-edge-device systems | Balancing tradeoff between cold start frequency and resource utilization while maximizing success rates (QoS, Energy, & Cost-based) | Increased task success rate by 70%, reduced monetary costs by 58%, and decreased average delay by 1.51% | B3, C1, C2, C3, D3, D4 |

TABLE II. COMPARATIVE BENCHMARK OF MACHINE LEARNING EFFICACY IN RESOURCE PROVISIONING

| ML Methodology | Primary Performance Impact | Representative Result |
|---|---|---|
| Deep RL (DQN/LSTM) [83] | Cost & QoS Optimization | Reduced response times by 43.3% and costs by 26.6% in AliCloud clusters. |
| Multi-Agent DRL (CSODQN) [86] | Success Rate & Cost | Increased task success rates by 70% and reduced monetary costs by 58%. |
| Intent-Driven Hybrid [81] | QoS & Energy Sustainability | Suppressed SLA/QoS violations to 1.52% while reducing power consumption by 24.2%. |
| Distributed RL (TD3) [75] | System Reliability | Decreased failed requests by 24% and improved average response times by 15%. |
| Hybrid DRL (H-A2C) [85] | User Cost Reduction | Achieved user cost reductions of up to 44% in hybrid serverful/serverless environments. |

TABLE III. COMPARATIVE ANALYSIS OF ML METHODOLOGIES

| ML Technique | Key Advantages | Major Trade-offs & Overhead | Ideal Use Case |
|---|---|---|---|
| Classical & Supervised | - High interpretability<br>- Low inference latency. | - Requires manual feature engineering - Struggles with highly nonlinear patterns. | Stable, predictable VM workloads. |
| Deep RL (DRL) | - Learns near-optimal policies autonomously<br>- No predefined rules required. | - High training complexity and massive data requirements<br>- Risk of real-time instability. | Dynamic VM/Container auto-scaling. |
| Multi-Agent (MARL) | - Superior decentralized decision-making<br>- Scales linearly across large clusters. | - Significant communication overhead<br>- Coordination complexity and security monitoring challenges. | Distributed Edge and Serverless orchestration. |

*4) Technical comparative analysis of ML methodologies:* To satisfy the requirement for a critical evaluation of methodologies, we analyze the trade-offs between the three dominant paradigms. Each method presents a unique profile regarding computational overhead, interpretability, and adaptive capacity, as shown in Table III.

The primary distinction lies in the Training vs. Inference costs: while classical models are inexpensive to deploy, they fail in the volatility of the edge. Conversely, MARL offers high elasticity but introduces coordination delays that can counteract latency gains if not properly optimized.

*5) System-level environmental distinctions:* Resource provisioning logic must fundamentally adapt to the specific abstraction layers and operational constraints of the underlying infrastructure, which are generally classified into three primary deployment tiers. In traditional cloud environments utilizing VMs and containers, the primary focus remains on macro-level resource placement and live migration to optimize hardware utilization and reduce energy consumption through server consolidation.

In contrast, serverless computing FaaS introduces micro-level, event-driven orchestration. The primary technical challenge is managing warm and cold container pools, which requires balancing reducing cold-start latency with implementing a scale-to-zero model to minimize operational costs. Within the edge-cloud continuum, provisioning must address highly distributed, resource-constrained nodes by strategically offloading microservices to meet ultra-low latency requirements at the edge, while also managing high compute costs and limited battery life in decentralized devices.

## VI. CONCLUSION

This paper presented a comprehensive survey of resource provisioning strategies, with a particular focus on the "Intelligence Shift" toward machine learning in cloud, edge, and serverless environments. Through a taxonomic analysis of eleven years of literature, we categorized ML interventions into four primary categories: classical and evolutionary machine learning, hybrid forecasting-optimization, deep reinforcement learning (DRL), and multi-agent systems (MARL). The principal finding of this survey is the increasing prevalence of hybrid architectures that combine time-series forecasting with DRL to address the fundamental trade-off between infrastructural cost and stringent Quality of Service (QoS) requirements. Our quantitative synthesis of the state of the art indicates that intelligence-driven interventions yield measurable improvements over traditional heuristics. For example, model-

free reinforcement learning algorithms reduce resource consumption by up to 10% and improve system performance by up to 30%. Additionally, hybrid Actor-Critic (H-A2C) models achieve user cost reductions of up to 44% and an 11% improvement in relative response times. Intent-driven orchestration models also suppress SLA/QoS violations to 1.52% and reduce data center power consumption by 24.2%.

To address the ongoing evolution of the edge-cloud continuum, we identify several priority areas for future research. Researchers should focus on developing models that reduce latency from serverless "cold starts" by employing intent-driven orchestration and predictive container pre-warming. As decentralized clusters expand, further refinement of MARL is necessary to manage coordination overhead in high-density edge nodes. The integration of federated learning is also essential to ensure secure resource allocation and data privacy in multi-tenant environments. Moreover, future algorithms should explicitly incorporate energy efficiency and carbon-aware metrics to facilitate the transition to green AI orchestration. While this survey offers a rigorous analysis of the current landscape, it is subject to specific scope constraints, including a search limited to high-impact databases such as IEEE Xplore, ACM Digital Library, and ScienceDirect. Our inclusion criteria also prioritized literature from the 2023-2025 period to emphasize the shift toward modern machine learning orchestration. As a result, foundational heuristic models are referenced primarily to contextualize the current paradigm shift and were not subjected to the same level of quantitative benchmarking as contemporary neural-based solutions.

## VII. Declaration on Generative AI

During the preparation of this work, the authors used Grammarly Paraphraser (Academic writing style) to improve the clarity and grammatical precision of the manuscript. After using this service, the author critically reviewed, verified, and edited the generated suggestions to ensure technical accuracy. The authors take full responsibility for the originality, accuracy, and integrity of the final publication.

## References

[1] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih, "Cloud Computing Virtualization of Resources Allocation for Distributed Systems," Journal of Applied Science and Technology Trends, vol. 1, no. 2, pp. 98–105, Jun. 2020, doi: 10.38094/jastt1331.

[2] S. Alharthi, A. Alshamsi, A. Alseiari, and A. Alwarafy, "Auto-Scaling Techniques in Cloud Computing: Issues and Research Directions," Sensors, vol. 24, no. 17, p. 5551, Aug. 2024, doi: 10.3390/s24175551.

[3] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih, "Cloud Computing Virtualization of Resources Allocation for Distributed Systems," Journal of Applied Science and Technology Trends, vol. 1, no. 2, pp. 98–105, Jun. 2020, doi: 10.38094/jastt1331.

[4] S. Chouliaras and S. Sotiriadis, "An adaptive auto-scaling framework for cloud resource provisioning," Future Generation Computer Systems, vol. 148, pp. 173–183, 2023.

[5] I. Cohen, C. F. Chiasserini, P. Giaccone, and G. Scalosub, "Dynamic Service Provisioning in the Edge-Cloud Continuum With Bounded Resources," IEEE/ACM Transactions on Networking, vol. 31, no. 6, pp. 3096–3111, Dec. 2023, doi: 10.1109/TNET.2023.3271674.

[6] A. Abouaomar, S. Cherkaoui, Z. Mlika, and A. Kobbane, "Resource Provisioning in Edge Computing for Latency-Sensitive Applications," IEEE Internet Things J., vol. 8, no. 14, pp. 11088–11099, Jul. 2021, doi: 10.1109/JIOT.2021.3052082.

[7] D. Ardagna, M. Ciavotta, R. Lancellotti, and M. Guerriero, "A Hierarchical Receding Horizon Algorithm for QoS-Driven Control of Multi-IaaS Applications," IEEE Transactions on Cloud Computing, vol. 9, no. 2, pp. 418–434, Apr. 2021, doi: 10.1109/TCC.2018.2875443.

[8] T. Le Duc, R. G. Leiva, P. Casari, and P.-O. Östberg, "Machine Learning Methods for Reliable Resource Provisioning in Edge-Cloud Computing," ACM Comput. Surv., vol. 52, no. 5, pp. 1–39, Sep. 2020, doi: 10.1145/3341145.

[9] K. L. James, N. P. Randall, and N. R. Haddaway, "A methodology for systematic mapping in environmental sciences," Environ. Evid., vol. 5, no. 1, p. 7, Dec. 2016, doi: 10.1186/s13750-016-0059-6.

[10] J. Zhang, H. Huang, and X. Wang, "Resource provision algorithms in cloud computing: A survey," Journal of network and computer applications, vol. 64, pp. 23–42, 2016.

[11] S. Vasoya, L. Gadhavi, J. Bhatia, and M. Bhavsar, "Resource provisioning strategies in cloud: a survey," 2016.

[12] S. Singh and I. Chana, "Cloud resource provisioning: survey, status and future research directions," Knowl. Inf. Syst., vol. 49, no. 3, pp. 1005–1069, 2016.

[13] K. D. Kumar and E. Umamaheswari, "Resource provisioning in cloud computing using prediction models: A survey," International Journal of Pure and Applied Mathematics, vol. 119, no. 9, pp. 333–342, 2018.

[14] S. Varshney, R. Sandhu, and P. K. Gupta, "QoS based resource provisioning in cloud computing environment: a technical survey," in International conference on advances in computing and data sciences, 2019, pp. 711–723.

[15] K. Sumalatha and M. S. Anbarasi, "A review on various optimization techniques of resource provisioning in cloud computing.," International Journal of Electrical & Computer Engineering (2088-8708), vol. 9, no. 1, 2019.

[16] A. Shakarami, H. Shakarami, M. Ghobaei-Arani, E. Nikougoftar, and M. Faraji-Mehmandar, "Resource provisioning in edge/fog computing: A comprehensive and systematic review," Journal of Systems Architecture, vol. 122, p. 102362, 2022.

[17] X. Li, L. Pan, and S. Liu, "A survey of resource provisioning problem in cloud brokers," Journal of Network and Computer Applications, vol. 203, p. 103384, 2022.

[18] S. Mustafa, B. Nazir, A. Hayat, S. A. Madani, and others, "Resource management in cloud computing: Taxonomy, prospects, and challenges," Computers & Electrical Engineering, vol. 47, pp. 186–203, 2015.

[19] R. S. S. Dittakavi, "An extensive exploration of techniques for resource and cost management in contemporary cloud computing environments," Applied Research in Artificial Intelligence and Cloud Computing, vol. 4, no. 1, pp. 45–61, 2021.

[20] B. H. Bhavani and H. S. Guruprasad, "Resource provisioning techniques in cloud computing environment: a survey," International Journal of Research in Computer and Communication Technology, vol. 3, no. 3, pp. 395–401, 2014.

[21] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," IEEE Trans. Industr. Inform., vol. 16, no. 9, pp. 6172–6181, 2019.

[22] K. S. S. Kumar and N. Jaisankar, "An automated resource management framework for minimizing SLA violations and negotiation in collaborative cloud," International Journal of Cognitive Computing in Engineering, vol. 1, pp. 27–35, 2020.

[23] R. Moreno-Vozmediano, R. S. Montero, E. Huedo, and I. M. Llorente, "Efficient resource provisioning for elastic cloud services based on machine learning techniques," Journal of Cloud Computing, vol. 8, no. 1, pp. 1–18, 2019.

[24] Y. Ge, Z. Ding, M. Tang, and Y.-C. Tian, "Resource provisioning for mapreduce computation in cloud container environment," in 2019 IEEE 18th international symposium on network computing and applications (NCA), 2019, pp. 1–4.

[25] S. Talwani et al., "Machine-learning-based approach for virtual machine allocation and migration," Electronics (Basel)., vol. 11, no. 19, p. 3249, 2022.

[26] M. H. Eawna, S. H. Mohammed, and E.-S. M. El-Horbaty, "Hybrid algorithm for resource provisioning of multi-tier cloud computing," Procedia Comput. Sci., vol. 65, pp. 682–690, 2015.

[27] G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: a review and future directions," Artif. Intell. Rev., vol. 57, no. 5, p. 124, Apr. 2024, doi: 10.1007/s10462-024-10756-9.

[28] Z. Chen, J. Hu, G. Min, C. Luo, and T. El-Ghazawi, "Adaptive and Efficient Resource Allocation in Cloud Datacenters Using Actor-Critic Deep Reinforcement Learning," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 8, pp. 1911–1923, Aug. 2022, doi: 10.1109/TPDS.2021.3132422.

[29] D. Li, S. Xu, and P. Li, "Deep Reinforcement Learning-Empowered Resource Allocation for Mobile Edge Computing in Cellular V2X Networks," Sensors, vol. 21, no. 2, p. 372, Jan. 2021, doi: 10.3390/s21020372.

[30] W. Wei, H. Gu, K. Wang, J. Li, X. Zhang, and N. Wang, "Multi-Dimensional Resource Allocation in Distributed Data Centers Using Deep Reinforcement Learning," IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1817–1829, Jun. 2023, doi: 10.1109/TNSM.2022.3213575.

[31] A. Thomas, T. Borangiu, and D. Trentesaux, "Holonic and multi-agent technologies for service and computing oriented manufacturing," J. Intell. Manuf., vol. 28, no. 7, pp. 1501–1502, Oct. 2017, doi: 10.1007/s10845-015-1188-4.

[32] B. Fang and D. Gao, "Collaborative Multi-Agent Reinforcement Learning Approach for Elastic Cloud Resource Scaling," in 2025 7th International Conference on Artificial Intelligence Technologies and Applications (ICAITA), IEEE, Jun. 2025, pp. 415–419. doi: 10.1109/ICAITA67588.2025.11137847.

[33] N. Šatkauskas and A. Venčkauskas, "Multi-Agent Dynamic Fog Service Placement Approach," Future Internet, vol. 16, no. 7, p. 248, Jul. 2024, doi: 10.3390/fi16070248.

[34] P. L. Donti, B. Amos, and J. Z. Kolter, "Task-based End-to-end Model Learning in Stochastic Optimization."

[35] J. Liu, Y. Zhang, Y. Zhou, D. Zhang, and H. Liu, "Aggressive Resource Provisioning for Ensuring QoS in Virtualized Environments," IEEE Transactions on Cloud Computing, vol. 3, no. 2, pp. 119–131, Apr. 2015, doi: 10.1109/TCC.2014.2353045.

[36] Z. Xiao, W. Song, and Q. Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp. 1107–1117, Jun. 2013, doi: 10.1109/TPDS.2012.283.

[37] Prof. P. R. P. Prof. P. R. Patil, M. R. Manasi Raut, A. B. Amruta Bakade, and A. S. Anushka Shingade, "Analysis of Virtual Machine Allocation Strategies and Development of a Novel Policy," Journal of Software Engineering and Simulation, vol. 11, no. 5, pp. 01–14, May 2025, doi: 10.35629/3795-11050114.

[38] S. Liu, C. Li, Z. Liu, and Q. Zhang, "Virtual Machine Dynamic Deployment Scheme Based on Double-Cursor Mechanism," IEEE Access, vol. 8, pp. 214481–214493, 2020, doi: 10.1109/ACCESS.2020.3040912.

[39] A. Laghrissi and T. Taleb, "A Survey on the Placement of Virtual Resources and Virtual Network Functions," IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1409–1434, 2019, doi: 10.1109/COMST.2018.2884835.

[40] W. Wei, K. Wang, K. Wang, H. Gu, and H. Shen, "Multi-resource balance optimization for virtual machine placement in cloud data centers," Computers & Electrical Engineering, vol. 88, p. 106866, Dec. 2020, doi: 10.1016/j.compeleceng.2020.106866.

[41] N. Zhou, H. Zhou, and D. Hoppe, "Containerization for High Performance Computing Systems: Survey and Prospects," IEEE Transactions on Software Engineering, vol. 49, no. 4, pp. 2722–2740, Apr. 2023, doi: 10.1109/TSE.2022.3229221.

[42] M. A. Rodriguez and R. Buyya, "Container‐based cluster orchestration systems: A taxonomy and future directions," Softw. Pract. Exp., vol. 49, no. 5, pp. 698‐719, May 2019, doi: 10.1002/spe.2660.

[43] Z. Zhong, M. Xu, M. A. Rodriguez, C. Xu, and R. Buyya, "Machine Learning-based Orchestration of Containers: A Taxonomy and Future Directions," ACM Comput. Surv., vol. 54, no. 10s, pp. 1–35, Jan. 2022, doi: 10.1145/3510415.

[44] Z. Zhong and R. Buyya, "A Cost-Efficient Container Orchestration Strategy in Kubernetes-Based Cloud Computing Infrastructures with Heterogeneous Resources," ACM Trans. Internet Technol., vol. 20, no. 2, pp. 1–24, May 2020, doi: 10.1145/3378447.

[45] M. A. Altahat, T. Daradkeh, and A. Agarwal, "Optimized Encryption-Integrated Strategy for Containers Scheduling and Secure Migration in Multi-Cloud Data Centers," IEEE Access, vol. 12, pp. 51330–51345, 2024, doi: 10.1109/ACCESS.2024.3386169.

[46] V. Struhár, S. S. Craciunas, M. Ashjaei, M. Behnam, and A. V. Papadopoulos, "Hierarchical Resource Orchestration Framework for Real-time Containers," ACM Transactions on Embedded Computing Systems, vol. 23, no. 1, pp. 1–24, Jan. 2024, doi: 10.1145/3592856.

[47] Z. Li et al., "Online Layer-Aware Joint Request Scheduling, Container Placement, and Resource Provision in Edge Computing," IEEE Trans. Serv. Comput., vol. 18, no. 1, pp. 328–341, Jan. 2025, doi: 10.1109/TSC.2024.3504237.

[48] Grace Joseph, Sunandha Rajagopal, Dr. Amrita Priya K, and Sreelekshmi R, "Dynamic Resource Scheduling Approaches in Server Less Computing," International Research Journal on Advanced Engineering and Management (IRJAEM), vol. 3, no. 05, pp. 1749–1758, May 2025, doi: 10.47392/IRJAEM.2025.0277.

[49] A. Mampage, S. Karunasekera, and R. Buyya, "A Holistic View on Resource Management in Serverless Computing Environments: Taxonomy and Future Directions," ACM Comput. Surv., vol. 54, no. 11s, pp. 1–36, Jan. 2022, doi: 10.1145/3510412.

[50] A. Mampage, S. Karunasekera, and R. Buyya, "Deadline-aware Dynamic Resource Management in Serverless Computing Environments," in 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), IEEE, May 2021, pp. 483–492. doi: 10.1109/CCGrid51090.2021.00058.

[51] P. Benedetti, M. Femminella, G. Reali, and K. Steenhaut, "Experimental Analysis of the Application of Serverless Computing to IoT Platforms," Sensors, vol. 21, no. 3, p. 928, Jan. 2021, doi: 10.3390/s21030928.

[52] H. Yu, H. Wang, J. Li, X. Yuan, and S.-J. Park, "Accelerating Serverless Computing by Harvesting Idle Resources," in Proceedings of the ACM Web Conference 2022, New York, NY, USA: ACM, Apr. 2022, pp. 1741–1751. doi: 10.1145/3485447.3511979.

[53] B. Zhu, Y. Zhu, C. Chen, and L. Kong, "Trident: A Provider-Oriented Resource Management Framework for Serverless Computing Platforms," IEEE Trans. Serv. Comput., vol. 18, no. 5, pp. 3334–3347, Sep. 2025, doi: 10.1109/TSC.2025.3603867.

[54] S. Rac and M. Brorsson, "Cost-aware Service Placement and Scheduling in the Edge-Cloud Continuum," ACM Transactions on Architecture and Code Optimization, vol. 21, no. 2, pp. 1–24, Jun. 2024, doi: 10.1145/3640823.

[55] I. Capeletti et al., "Towards Optimizing the Edge-to-Cloud Continuum Resource Allocation," in Proceedings of the 13th International Conference on Cloud Computing and Services Science, SCITEPRESS - Science and Technology Publications, 2023, pp. 90–99. doi: 10.5220/0011995700003488.

[56] A. Bertoncini, A. Ceselli, and C. Quadri, "Latency-Aware Placement of Microservices in the Cloud-to-Edge Continuum via Resource Scaling," in 2025 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, Jun. 2025, pp. 420–425. doi: 10.1109/SMARTCOMP65954.2025.00103.

[57] M. I. Hussain et al., "Hybrid SFLA-UBS algorithm for optimal resource provisioning with cost management in multi-cloud computing," International Journal of Advanced Computer Science and Applications, vol. 12, no. 4, 2021.

[58] M. Kumar, A. Kishor, J. Abawajy, P. Agarwal, A. Singh, and A. Y. Zomaya, "ARPS: An autonomic resource provisioning and scheduling framework for cloud platforms," IEEE Transactions on Sustainable Computing, vol. 7, no. 2, pp. 386–399, 2021.

[59] M. Ghobaei-Arani and A. Shahidinejad, "An efficient resource provisioning approach for analyzing cloud workloads: a metaheuristic-based clustering approach," J. Supercomput., vol. 77, no. 1, pp. 711–750, 2021.

[60] A. Suresh and R. Varatharajan, "Competent resource provisioning and distribution techniques for cloud computing environment," Cluster Comput., vol. 22, no. Suppl 5, pp. 11039–11046, 2019.

[61] M. M. Nejad, L. Mashayekhy, and D. Grosu, "Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds," IEEE transactions on parallel and distributed systems, vol. 26, no. 2, pp. 594–603, 2014.

[62] R. Panwar and M. Supriya, "Dynamic resource provisioning for service-based cloud applications: A Bayesian learning approach," J. Parallel Distrib. Comput., vol. 168, pp. 90–107, 2022.

[63] C. Luo et al., "Intelligent virtual machine provisioning in cloud computing," in Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 1495–1502.

[64] S. A. Ajila and A. A. Bankole, "Using machine learning algorithms for cloud client prediction models in a web VM resource provisioning environment," Transactions on Machine Learning and Artificial Intelligence, vol. 4, no. 1, p. 28, 2016.

[65] F. Mirzad* and M. R. Ghalib, "Forecasting Cloud Resource Provisioning System using Supervised Machine Learning," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 6, pp. 3591–3596, Mar. 2020, doi: 10.35940/ijrte.F8886.038620.

[66] Y. Sharma, J. Taheri, W. Si, D. Sun, and B. Javadi, "Dynamic resource provisioning for sustainable cloud computing systems in the presence of correlated failures," IEEE Transactions on Sustainable Computing, vol. 6, no. 4, pp. 641–654, 2020.

[67] B. Sniezynski, P. Nawrocki, M. Wilk, M. Jarzab, and K. Zielinski, "VM reservation plan adaptation using machine learning in cloud computing," J. Grid Comput., vol. 17, no. 4, pp. 797–812, 2019.

[68] M. S. Al-Asaly, M. A. Bencherif, A. Alsanad, and M. M. Hassan, "A deep learning-based resource usage prediction model for resource provisioning in an autonomic cloud computing environment," Neural Comput. Appl., vol. 34, no. 13, pp. 10211–10228, 2022.

[69] J. Choi and Y. Kim, "Adaptive resource provisioning method using application-aware machine learning based on job history in heterogeneous infrastructures," Cluster Comput., vol. 20, no. 4, pp. 3537–3549, 2017.

[70] S. Ghasemi, M. R. Meybodi, M. D. T. Fooladi, and A. M. Rahmani, "A cost-aware mechanism for optimized resource provisioning in cloud computing," Cluster Comput., vol. 21, no. 2, pp. 1381–1394, 2018.

[71] M. P. Yadav, Rohit, and D. K. Yadav, "Resource provisioning through machine learning in cloud services," Arab. J. Sci. Eng., vol. 47, no. 2, pp. 1483–1505, 2022.

[72] C. Ayimba, P. Casari, and V. Mancuso, "SQLR: Short-term memory Q-learning for elastic provisioning," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1850–1869, 2021.

[73] S. Agarwal, M. A. Rodriguez, and R. Buyya, "A Deep Recurrent-Reinforcement Learning Method for Intelligent AutoScaling of Serverless Functions," IEEE Trans. Serv. Comput., vol. 17, no. 5, pp. 1899–1910, 2024, doi: 10.1109/TSC.2024.3387661.

[74] H. Yu, H. Wang, J. Li, X. Yuan, and S.-J. Park, "Freyr +: Harvesting Idle Resources in Serverless Computing via Deep Reinforcement Learning," IEEE Transactions on Parallel and Distributed Systems, vol. 35, no. 11, pp. 2254–2269, Nov. 2024, doi: 10.1109/TPDS.2024.3462294.

[75] H. Bai, M. Xu, K. Ye, R. Buyya, and C. Xu, "DRPC: Distributed Reinforcement Learning Approach for Scalable Resource Provisioning in Container-Based Clusters," IEEE Trans. Serv. Comput., vol. 17, no. 6, pp. 3473–3484, 2024, doi: 10.1109/TSC.2024.3433388.

[76] S. Nagarajan, P. S. Rani, M. S. Vinmathi, V. Subba Reddy, A. L. M. Saleth, and D. Abdus Subhahan, "Multi agent deep reinforcement learning for resource allocation in container-based clouds environments," Expert Syst., vol. 42, no. 1, Jan. 2025, doi: 10.1111/exsy.13362.

[77] T. Theodoropoulos, A. Makris, I. Korontanis, and K. Tserpes, "GreenKube: Towards Greener Container Orchestration using Artificial Intelligence." [Online]. Available: https://kubernetes.io/

[78] M. Golec et al., "ATOM: AI-Powered Sustainable Resource Management for Serverless Edge Computing Environments," IEEE Transactions on Sustainable Computing, vol. 9, no. 6, pp. 817–829, 2024, doi: 10.1109/TSUSC.2023.3348157.

[79] H. Zhang, J. Wang, H. Zhang, and C. Bu, "Security computing resource allocation based on deep reinforcement learning in serverless multi-cloud edge computing," Future Generation Computer Systems, vol. 151, pp. 152–161, Feb. 2024, doi: 10.1016/j.future.2023.09.016.

[80] X. Chen, L. Yang, Z. Chen, G. Min, X. Zheng, and C. Rong, "Resource Allocation With Workload-Time Windows for Cloud-Based Software Services: A Deep Reinforcement Learning Approach," IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1871–1885, Apr. 2023, doi: 10.1109/TCC.2022.3169157.

[81] N. Filinis et al., "Intent-driven orchestration of serverless applications in the computing continuum," Future Generation Computer Systems, vol. 154, pp. 72–86, May 2024, doi: 10.1016/j.future.2023.12.032.

[82] Y. Gong, J. Huang, B. Liu, J. Xu, B. Wu, and Y. Zhang, "Dynamic Resource Allocation for Virtual Machine Migration Optimization using Machine Learning," Applied and Computational Engineering, vol. 57, no. 1, pp. 1–8, Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.13619

[83] Y. Wang and X. Yang, "Intelligent Resource Allocation Optimization for Cloud Computing via Machine Learning," Advances in Computer, Signals and Systems, vol. 9, no. 1, Mar. 2025, doi: 10.23977/acss.2025.090109.

[84] A. Lipari, G. P. Mattia, and R. Beraldi, "Dynamic and Forecast-Based Containers Autoscaling for Kubernetes with Reinforcement Learning," in Proceedings - 2025 IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2025, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 1081–1088. doi: 10.1109/IPDPSW66978.2025.00169.

[85] A. Mampage, S. Karunasekera, and R. Buyya, "Deep Reinforcement Learning for Scheduling Applications in Serverless and Serverful Hybrid Computing Environments," IEEE Trans. Serv. Comput., vol. 18, no. 2, pp. 718–728, 2025, doi: 10.1109/TSC.2024.3520864.

[86] P. Wu, H. Chen, T. Wu, K. Gu, and Y. Xia, "Cold-Start-Aware Offloading and Resource Allocation by Importance Sampling-Based Double Dueling DQN in Serverless Edge Computing," IEEE Internet Things J., vol. 12, no. 19, pp. 40190–40205, 2025, doi: 10.1109/JIOT.2025.3588727.