International Journal of Innovative Technology and Interdisciplinary Sciences
https://journals.tultech.eu/index.php/ijitis
ISSN: 2613-7305
Volume 9, Issue 1
DOI: https://doi.org/10.15157/IJITIS.2026.9.1.366-401
Received: 27.11.2025; Revised: 22.01.2026; Accepted: 08.02.2026

Review Article

AI-Driven Cloud Administration: A Literature Review and Comparative Synthesis of Forecasting, Resource Allocation, Cost Optimization and Load Balancing Approaches

Lindita Loku Nikçi, Afërdita Ibrahimi, Artan Dermaku*, Basri Ahmedi

Faculty of Computer Science, University "Kadri Zeka", Gjilan, Kosovo
***prof.artan.dermaku@uni-gjilan.net**

Abstract

This review article examined AI-driven approaches for cloud administration through a structured literature review and comparative synthesis of studies published between 2016 and 2025 (N = 57). The review focused on four interdependent administrative functions: predictive workload analysis, dynamic resource allocation and scheduling, cost–energy–QoS optimization, and AI-enhanced load balancing under reliability and security constraints. Publications were retrieved from major scholarly databases and were screened using eligibility criteria requiring direct relevance to cloud operations, explicit use of AI/ML/optimization for operational decision-making, and reported operational metrics or comparative evidence. The synthesis indicated that short-horizon forecasting models generally reduced over-provisioning and supported proactive scaling, but forecasting was often evaluated in isolation, limiting end-to-end evidence for sustained SLO improvement under concept drift and multi-cloud variability. Reinforcement learning and meta-heuristic schedulers commonly improved utilization and makespan relative to rule-based baselines, although many studies relied on simulator settings and reported limited reproducibility and generalization under realistic constraints. Cost- and energy-aware methods frequently reduced OPEX and energy via consolidation, DVFS, and cost-aware placement, yet they exposed trade-offs with QoS stability and used heterogeneous modelling assumptions. AI-based load balancing approaches improved latency and robustness in burst and failure scenarios, while explainability and portable trust/reliability metrics remained underdeveloped. Based on cross-stream evidence, a conceptual reference framework was derived that linked forecasting, scheduling, cost–energy objectives, and load balancing as a unified decision pipeline and highlighted gaps in benchmarks, portability, and transparency.

Keywords: AI-driven Cloud Administration; Workload Forecasting; Reinforcement Learning; Cost–energy Optimization; Multi-Cloud Load Balancing.

INTRODUCTION

Artificial Intelligence (AI) has become a central enabler of modern cloud administration as large-scale infrastructures grow more heterogeneous, service-oriented, and increasingly

https://doi.org/10.15157/IJITIS.2026.9.1.366-401

distributed across multi-cloud and edge–cloud continuums. Cloud platforms today must support latency-sensitive and bursty applications while operating under strict Quality of Service (QoS) and Service Level Objective (SLO) constraints, often with limited budgets and rising operational expenditures (OPEX) [1-6]. In this context, conventional reactive management such as threshold-triggered scaling or static placement heuristics frequently underperforms when workloads shift rapidly, when resources are constrained by quotas and affinity rules, or when pricing and availability vary across providers [7-9]. Consequently [10], an extensive research domain has developed that investigates AI-driven methodologies to enhance decision-making in cloud administration, particularly in forecasting, dynamic resource allocation, cost-aware optimization, and intelligent load balancing [11-15]. Figure 1 depict the fog computing architecture with cloud, fog, and IOT layers.

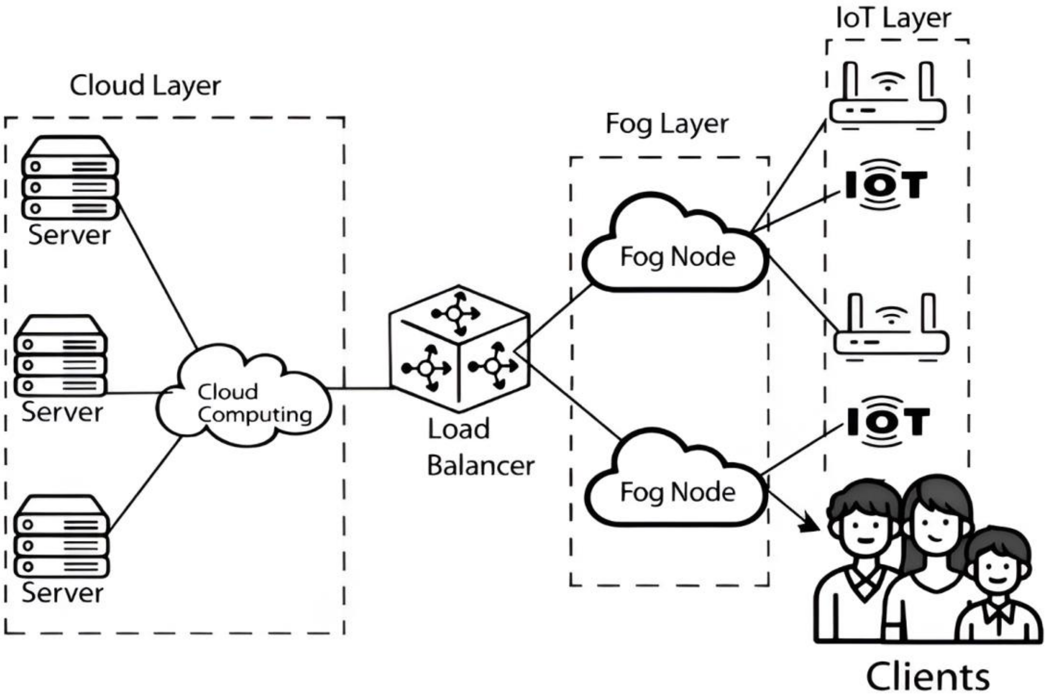

Figure 1. Fog Computing Architecture with Cloud, Fog, and IoT Layers

Among these pillars, predictive workload analysis has been widely studied as a prerequisite for proactive capacity planning [2]. Recent work shows that short-horizon forecasting models can anticipate near-future demand patterns and reduce reactive scaling events that typically lead to instability in tail latency and higher SLO violation risk [4]. However, forecasting is often evaluated as an isolated component rather than as a decision input into orchestration, scheduling, and traffic engineering. Moreover, forecasting accuracy may degrade under concept drift, promotional shocks, or cross-provider variability conditions that characterize real production environments and multi-cloud settings [2]. These limitations motivate a broader comparative view: beyond whether forecasting is accurate, how effectively do different forecasting approaches translate into operational actions that improve QoS, reduce over-provisioning, and preserve stability? A second important area of research is dynamic resource allocation and scheduling. Here, reinforcement learning (RL), evolutionary algorithms, and hybrid meta-heuristic approaches are being employed more and more to improve placement, scaling, and job scheduling. Research indicates enhancements in utilization, energy efficiency and, in some instances, decreases in SLA violations using multi-objective optimization and intelligent search-based controllers [7-13]. However, this body of work still has a number of problems, such as limited reproducibility (missing hyperparameters and incomplete configuration details), high training and optimization costs, weak generalization across different configurations, and narrow evaluation settings (only one provider or simulator). In multi-cloud systems, managing constraints is harder because of quotas, affinity/anti-affinity, region limits, and provider heterogeneity [16-25]. This makes stability and feasibility very important, although studies do not always deal with them in the same manner [1-7]. Cost-aware and energy-aware optimization constitutes a third research pillar, motivated by escalating operational expenditures and environmental goals. Previous research has suggested multi-objective formulations that include financial costs, energy use, and performance penalties [11, 26-29]. These include methods based on DVFS-aware scheduling, consolidation, and cost-aware instance selection [30-35].

However, the literature often addresses cost and energy in a fragmented manner optimizing cost without regard for dependability and security, or presenting energy proxies without clear assumptions and standardized measurements. In addition, the interaction between cost-aware decisions and downstream QoS outcomes (including stability under workload volatility) is not consistently analysed across studies, limiting actionable guidance for administrators deciding between reserved/on-demand/spot mixes, tiered storage strategies, and cross-region placement [35-37]. Figure 2 illustrate the cloud computing architecture and end-user connectivity

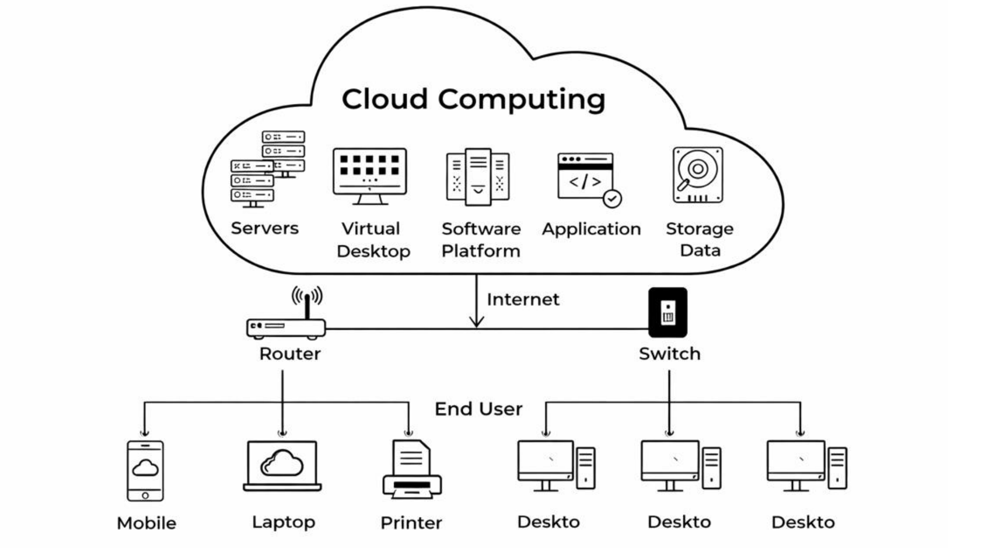

Figure 2. Cloud Computing Architecture and End-User Connectivity

Additionally, AI-enhanced load balancing and traffic engineering extends cloud administration into networked and service-mesh-driven environments where decisions must consider latency, error rates, congestion, and failure behaviour. RL and optimization-based load balancing approaches have been reported to outperform static heuristics under volatile traffic [20], while reliability-aware variants and threat-aware scheduling highlight the importance of robustness and security constraints in operational decision-making [30-33]. Despite progress, many solutions behave as black-box controllers with limited explainability and insufficient analysis of routing stability under failures, bursty demand, or multi-cloud routing constraints. As cloud systems increasingly adopt service meshes and policy-based routing, administrator trust, safety guardrails, and transparent decision logic become essential for real-world adoption [33-38]. Despite the breadth of research across these four domains, the overall literature remains fragmented. Many studies focus on one administration function (e.g., forecasting or scheduling) without systematically comparing methods across the broader cloud administration pipeline or clarifying interdependencies (e.g., how forecasting choices affect placement decisions, or how cost objectives reshape routing behaviour). Moreover, comparative evaluations are frequently shallow, use different datasets/simulators and metrics, and provide limited reproducible details, making cross-study synthesis difficult. Key gaps persist regarding: (i) unified comparative evidence across forecasting, allocation, cost/energy optimization, and load balancing; (ii) clear identification of methodological strengths, limitations, and evaluation contexts in state-of-the-art approaches; and (iii) a coherent conceptual reference that consolidates findings into actionable guidance for practitioners and researchers [9-15].

Despite the rapid development of AI-driven cloud management research, the current state of the art (SOTA) remains fragmented. Existing studies often treat load forecasting, resource allocation and planning, cost/energy optimization, and load balancing as separate components, evaluated under heterogeneous experimental contexts, with non-standardized metrics and varying levels of transparency and reproducibility. As a consequence, there is a lack of end-to-end comparative evidence to explain how choices in one pillar (eg prediction) directly affect subsequent decision-making (eg scheduling, costs, QoS stability). Existing reviews typically focus on a single administrative function or provide generic taxonomies, without a cross-pillar synthesis linking techniques, evaluation contexts, and real implementation constraints in multi-cloud and edge–cloud environments. This fragmentation makes it difficult to derive practical guidelines and identify research priorities for reliable, portable and transparent cloud management systems.

To address these gaps, this paper presents a literature review with comparative synthesis of AI-driven cloud administration approaches, focused on four analytical streams: predictive workload analysis, dynamic resource allocation, cost-aware optimization, and AI-enhanced load balancing. The review does not claim an implemented system or a new algorithm; instead, it critically consolidates reported techniques, evaluation settings, target objectives, and limitations across the literature (2016–2025; N =

57). To guide the synthesis, the paper is structured around four explicit research questions: The main contributions of this paper lie in several key points such as, (i) a structured literature review (2016–2025; N = 57) is provided that integrates four interrelated functions of AI-driven cloud management: load forecasting, dynamic resource allocation and scheduling, cost–energy–QoS optimization, and intelligent load balancing; (ii) a comparative synthesis of existing evidence is performed using consistent analysis dimensions (technique class, objectives, context of assessment, reported metrics and constraints), in order to distinguish real benefits from results dependent on simulation or assumptions; (iii) a conceptual reference framework is derived that models cloud management as a unified decision-making pipeline (forecasting → planning → cost/energy constraints → load balancing), identifying key points of integration between these functions; and (iv) articulate the most critical research gaps and practical implications related to benchmarks, multi-cloud portability, transparency and reproducibility, and standardization of reliability and security metrics. To move from a description of the literature to a comparative analysis with synthesizing value, this review is structured around research questions that aim not only to identify the techniques used, but also to assess the strength of the evidence, the conditions under which benefits are reported, and the repeated trade-offs between performance, cost, energy, and reliability. Based on existing literature, proactive AI-driven approaches are expected to provide measurable improvements over reactive policies, but only when they are coherently integrated into the management pipeline and evaluated under realistic constraints (e.g. load variability, multi-cloud heterogeneity, and failure scenarios). The following research questions are formulated to test these expectations through a synthesis of the evidence reported in the literature.

- RQ1: What evidence does the literature provide that short-horizon AI-based forecasting reduces SLO violations and resource over-provisioning compared with reactive policies?
- RQ2: How do RL and meta-heuristic schedulers compare with rule-based baselines in latency, makespan, utilization, and stability under volatile workloads?
- RQ3: What trade-offs are reported between cost, energy efficiency, and QoS/reliability in state-of-the-art cost-aware optimization methods?
- RQ4: To what extent do AI-enhanced load balancing approaches improve latency/error performance and robustness under failures and multi-cloud constraints?

Based on the cross-stream comparison, the paper derives a conceptual reference framework that organizes how this administration functions relate at a high level and where integration points and open challenges remain. Figure 3 summarizes the four conceptual blocks and their relationships in an AI-driven cloud administration pipeline, serving as a unifying map for the review. The remainder of the paper is organized as follows. The review methodology and scope are first described, followed by a presentation of the related work structured around four analytical streams and consolidated in an evidence matrix. A comparative synthesis and conceptual framework are then provided. This is followed by a discussion of open research gaps and implications for practice. The paper concludes with recommendations and directions for future research.

LITERATURE REVIEW

Recent scholarship converges on the use of AI-driven techniques to improve cloud administration across four core functions: (i) predictive workload analysis, (ii) dynamic resource allocation and scheduling, (iii) cost–energy–QoS optimization, and (iv) AI-enhanced load balancing under reliability and security constraints. Across these streams, studies report measurable improvements in utilization, latency [9], SLA/SLO compliance, energy consumption, and fault tolerance when machine learning (ML), reinforcement learning (RL), and meta-heuristic optimization are embedded into resource-management pipelines [15]. However, the literature is often siloed by component and evaluated under heterogeneous assumptions, which motivates a structured comparative synthesis rather than a single-thread narrative [33].

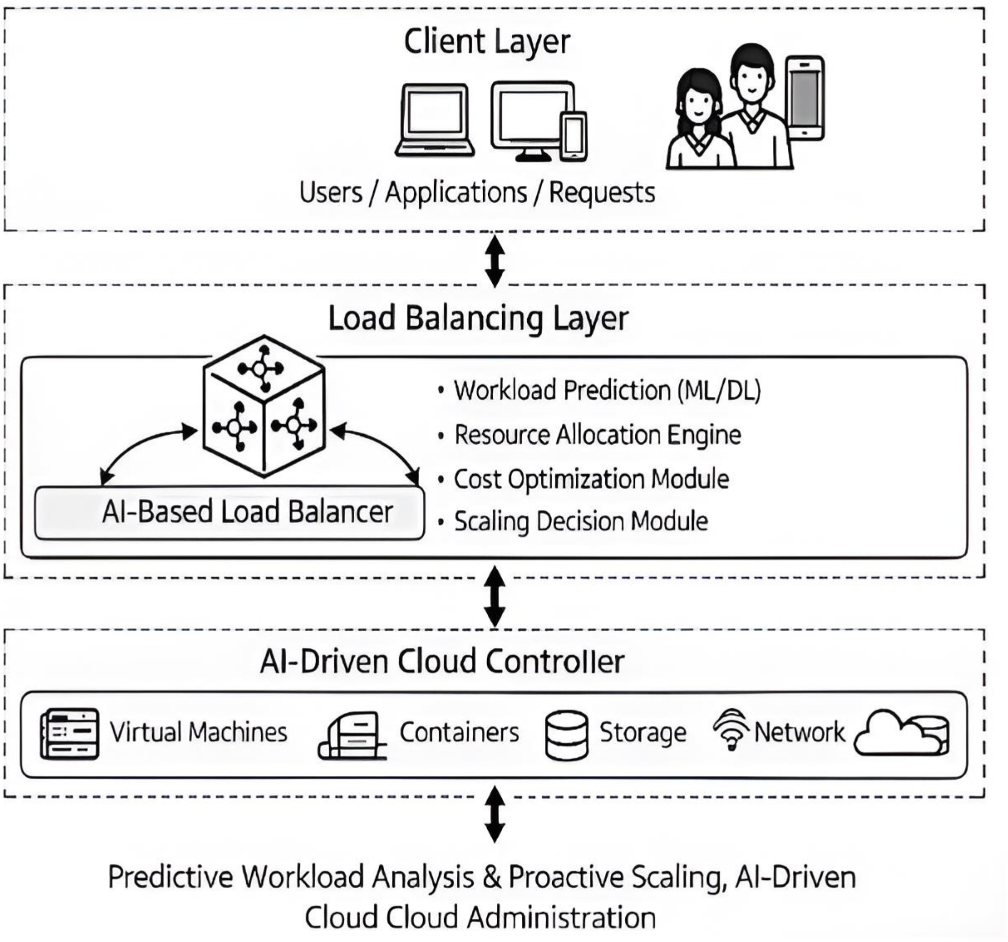
Client Layer
Users / Applications / Requests
Load Balancing Layer
• Workload Prediction (ML/DL)
• Resource Allocation Engine
• Cost Optimization Module
• Scaling Decision Module
AI-Based Load Balancer
AI-Driven Cloud Controller
Virtual Machines
Containers
Storage
Network
Predictive Workload Analysis & Proactive Scaling, AI-Driven Cloud Cloud Administration

Figure 3. AI-Based Load Balancing in Cloud Computing

Predictive Workload Analysis and Proactive Scaling

Workload forecasting is widely treated as the starting point for proactive autoscaling and capacity planning. Large-scale empirical studies on workload characteristics emphasize the importance of capturing burstiness, diurnal/weekly seasonality, and non-stationarity when designing forecasting horizons, feature windows, and retraining strategies [2]. Within this context, probabilistic forecasting is frequently highlighted as more operationally useful than point forecasting because it supports risk-aware capacity planning through quantiles and headroom.

CloudAIBus provides a representative example of forecast-driven cloud management, reporting that DeepAR-based forecasting can substantially reduce CPU over-provisioning and improve prediction error metrics compared with baseline models [39-41]. Beyond pure accuracy, reliability-aware prediction and drift-aware updates are increasingly emphasized as necessary to maintain forecasting quality under changing production regimes [49]. In multi-cloud settings, forecasting is further complicated by provider heterogeneity and cross-region variability, motivating approaches that integrate reliability signals and context-aware covariates into forecasting pipelines [15]. Figure 4 shows the Kubernetes-based auto-scaling and load balancing architecture

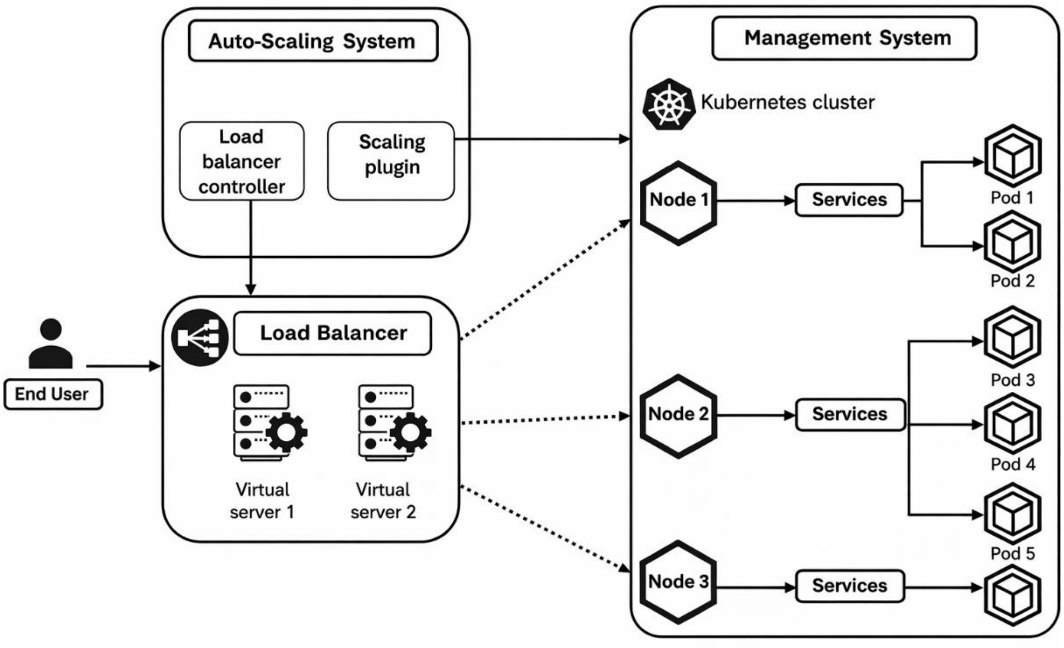

Figure 4. Kubernetes-Based Auto-Scaling and Load Balancing Architecture

A parallel strand focuses on compliance and operational constraints as first-class signals in predictive decision-making. For example, compliance-aware ML in distributed systems underscores how detection and prevention pipelines can influence control decisions, rather than serving only as offline monitoring [21]. Similarly, edge–cloud synergy work demonstrates that splitting prediction tasks across edge (fast anomaly screening) and cloud (sequence-level forecasting) can improve responsiveness and reduce overhead, indicating that "where" prediction runs may be as important as "which model" is used [36]. Collectively, these studies suggest that forecasting is most valuable when tightly coupled to the orchestration layer and augmented with reliability/compliance signals.

Dynamic Resource Allocation and Scheduling

Dynamic allocation covering placement, scaling actions, and scheduling forms the most heavily populated stream, spanning RL, evolutionary computation, and hybrid meta-heuristics. A systematic review of resource allocation and scheduling methods highlights that many approaches optimize multi-objective targets (e.g., makespan, energy, cost, SLA violations) but differ widely in evaluation rigor, reproducibility, and constraint modeling [9]. Recent ML-based allocation directions also include supervised learning formulations for placement decisions, although they often rely on simplified assumptions that may not generalize under multi-cloud constraints [24]. Figure 5 depicts the cloud–edge computing architecture

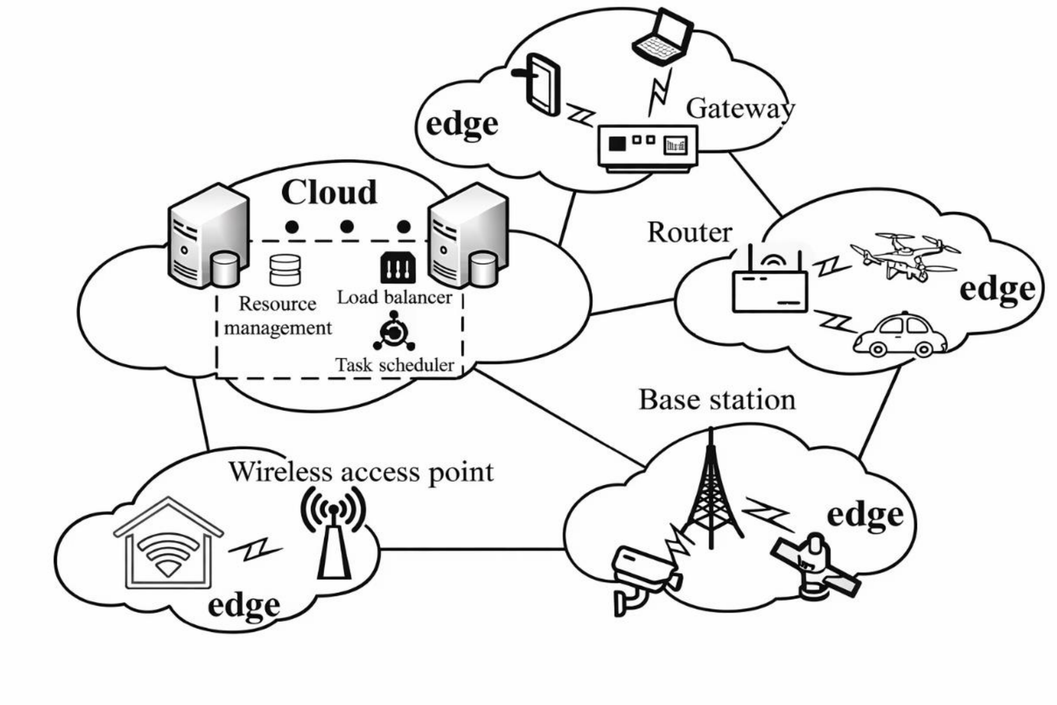

Figure 5. Cloud–Edge Computing Architecture

Multi-objective optimization in multi-cloud workflow scheduling has received particular attention. Hybrid multi-objective algorithms are reported to improve convergence behavior and solution diversity (Pareto quality) under competing objectives such as cost, energy, and throughput [7]. Swarm-intelligence variants and hybrid designs are frequently proposed to balance exploration and exploitation, especially under volatile workloads [29]. In parallel, scheduling methods under adversarial conditions (e.g., DDoS scenarios) show the need for resilience-aware policies, where hybrid meta-heuristics can sustain performance better than single-method baselines in simulation settings [12].

Reliability-aware allocation is increasingly treated not merely as a constraint, but as an objective integrated into the fitness function. Works emphasizing resilience in error-prone environments and edge-centric cloud settings present hybrid multi-objective approaches where reliability and energy are jointly optimized [31]. Trust-aware resource allocation also appears in multi-cloud contexts, proposing allocation decisions that incorporate trust and integrity signals into the placement objective [1]. These studies jointly motivate comparative questions about what algorithm families perform best under constraints (quotas, affinity rules, heterogeneous pricing) and what trade-offs are consistently reported across evaluation environments.

Cost- and Energy-Aware Optimization

Cost optimization has evolved from simple "min-cost" placement to multi-objective formulations that jointly consider monetary spend, QoS penalties, and energy consumption, see Figure 6. Power modeling is often treated as a prerequisite for energy-aware control, enabling scheduling algorithms to incorporate DVFS and consolidation decisions in more principled ways [11]. Several studies propose cost-aware scheduling under deadlines and heterogeneous VM performance [42-50], emphasizing that provisioning time and performance variability can materially influence cost–QoS outcomes [51-57].

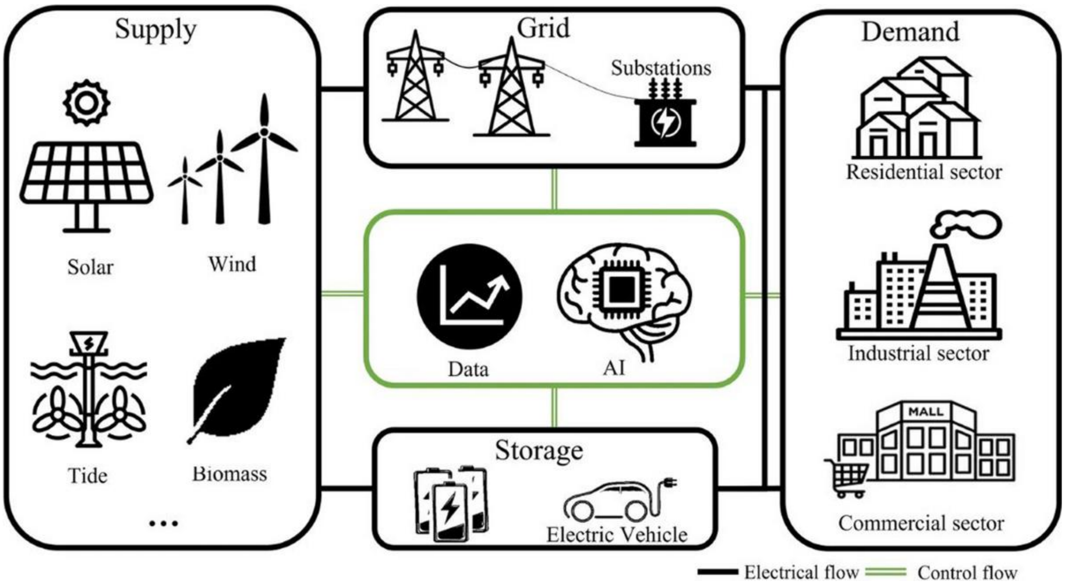

Figure 1. Smart Energy Management System with AI and Data Analytics (Author design)

Within cloud and IoE/IoT workloads, energy-efficient task management approaches highlight the role of joint objective design and the need to explicitly trade off makespan, energy, and reliability [46]. Work on reducing cost and energy through integrated prediction and scheduling similarly reinforces the view that forecasting and optimization should not be isolated modules if the goal is end-to-end OPEX reduction [35]. For business intelligence and multi-cloud governance, proposed frameworks emphasize interoperability and monitoring as enabling layers for cost control, suggesting that cost optimization is not only an algorithmic problem but also a systems and observability problem [15].

Furthermore, this stream indicates that a meaningful comparison must account for: (i) what cost components are modeled (compute, storage, egress, spot/reserved dynamics), (ii) whether energy is measured or proxied, and (iii) how QoS/SLO penalties are integrated into the objective dimensions that vary widely across prior work [35].

AI-Enhanced Load Balancing, Reliability, and Security Constraints

AI-enhanced load balancing aims to reduce latency and errors by dynamically distributing traffic across services, nodes, or regions under changing demand and failures. Reliability-aware load balancing approaches propose meta-heuristic optimization (e.g., Grey Wolf Optimization variants) that incorporate reliability signals into the balancing decision to reduce response time and cost in simulated cloud environments, see Figure 7 [22]. Recent works focused on AI-driven load balancing and optimization similarly argue that adaptive policies can outperform static heuristics, though evaluation settings and controller stability constraints differ [33].

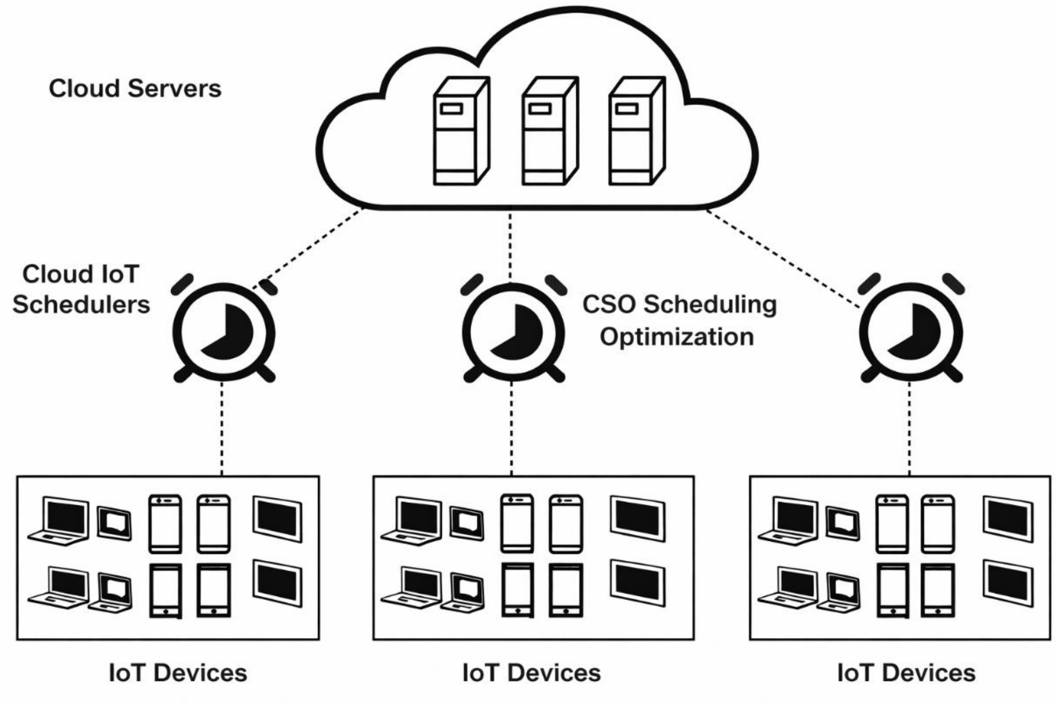

Figure 7. IoT Device Scheduling Framework with Cloud Optimization

Security and privacy constraints introduce additional complexity that interacts directly with routing and scheduling decisions. For example, secure workflow scheduling models report reductions in time/power/energy under security constraints in heterogeneous cloud contexts, emphasizing that cryptographic or secure data-handling requirements can materially shift optimal scheduling policies [45]. Multi-cloud privacy and availability mechanisms (e.g., secret sharing and fault-tolerant designs) further highlight that robustness and confidentiality are not optional add-ons but must be treated as system-level requirements that constrain orchestration strategies [47]. Private cloud security evaluations underscore that detection performance (e.g., SVM/RF) can be strong but requires continuous tuning and operational alignment to remain effective in practice [30]. Taken together, these works show that load balancing in realistic environments must be studied together with reliability and security constraints, and that evaluation should include failure/burst regimes and not only steady-state performance [22].

Cross-Stream Evidence and Open Gaps

Across all four streams, the literature suggests that the most practical gains occur when forecasting, allocation, optimization, and load balancing are integrated rather than treated independently. However, several recurring gaps motivate the need for comparative synthesis:

1. ***Fragmented evaluation contexts*** (simulators vs. traces vs. production-like settings) and inconsistent metric definitions limit cross-study comparability [9].
2. ***Portability and multi-cloud constraints*** (quotas, affinity/anti-affinity, heterogeneity of pricing and performance) are often under-modeled despite being central in real deployments [1-7].
3. ***Reliability/security as first-class signals*** is acknowledged across multiple works, yet rarely standardized into portable metrics for cross-provider decision-making [30-35].
4. ***Reproducibility and configuration transparency*** (hyperparameters, assumptions, workload definitions) remain uneven, hindering practitioner adoption and reliable benchmarking [9-11].

In response, this paper structures the review around the four analytical streams, applies a consistent comparative lens (technique class, objectives, evaluation context, reported metrics, and limitations), and derives a conceptual reference framework that highlights integration points and actionable gaps for future research and operational practice. Overall, the reviewed literature shows that AI-driven approaches can significantly improve the performance of cloud administration, but the evidence remains uneven and dependent on the context of evaluation. Many studies report significant benefits in resource usage, cost or latency; however, these results are often based on simulations or limited scenarios that do not fully reflect the complexity of real multi-cloud environments. Furthermore, the analysis shows that forecasting, scheduling, cost optimization, and load balancing are mostly treated as independent modules, overlooking the chain effects that decisions in a pillar have on overall system performance. These observations reinforce the need for a cross-pillar comparative synthesis and an integrated conceptual framework to link existing results to real implementation constraints and serve as a basis for analysis and discussion in the following sections.

METHODOLOGY

This study follows a literature review with comparative synthesis design to analyze AI-driven approaches for cloud administration. The goal is to map the research landscape, compare state-of-the-art (SOTA) solutions across key dimensions, and derive a conceptual reference framework informed by the evidence reported in prior studies. The paper does not claim an implemented system or empirical benchmarking; instead, it consolidates findings, trade-offs, and open gaps reported in the literature.

The review covers publications from 2016 to 2025, capturing both foundational ML/optimization techniques and recent cloud-native/multi-cloud AI management research. The reference corpus used in the review includes 57 sources (N = 57), predominantly peer-reviewed journal articles and conference papers, complemented by a small number of technical reports, arXiv preprints, and book chapters where relevant to definitions and background.

Relevant literature was retrieved from major scholarly databases and publisher digital libraries typically used for cloud computing and AI research, including:

- IEEE Xplore
- ACM Digital Library
- SpringerLink
- Elsevier ScienceDirect
- Wiley Online Library
- MDPI
- Taylor & Francis / journals indexed in Scopus/Web of Science (when accessible)
- arXiv (for emerging work not yet formally indexed)

Search and screening were guided by combinations of keywords reflecting the paper's four focal domains, such as: AI-driven cloud administration, workload forecasting, autoscaling, reinforcement learning scheduling, meta-heuristic optimization, cost-aware scheduling, energy-efficient resource management, multi-cloud orchestration, service mesh load balancing, reliability-aware scheduling, trust-aware resource allocation, cloud security constraints.

Review Protocol and Screening Process

The review process followed a structured study selection protocol consisting of several stages. Initially, the literature was identified through searches of academic databases using defined keyword strings. In the screening phase, titles and abstracts were reviewed to eliminate papers that were not directly related to cloud management. Next, in the appropriateness assessment phase, full texts were analyzed against the inclusion criteria, with a focus on the use of AI/ML for operational decision-making and the reporting of relevant metrics. The final included studies formed the analytic corpus used in the comparative synthesis. This process follows general practices of systematic literature reviews (SLR), ensuring transparency and reproducibility in the selection and analysis of studies, even though the study does not aim for a formal meta-analysis. To ensure relevance to cloud administration (and avoid unrelated ML-only papers), studies were included if they:

1. address at least one of the four cloud administration functions (forecasting, allocation, cost/energy optimization, load balancing), and
2. specify an AI/ML/optimization technique used for operational decision-making, and
3. report at least one operational outcome/metric (e.g., utilization, makespan, latency, SLO/SLA violations, energy, cost, reliability/security indicators), or provide a detailed comparative/survey synthesis.

Studies were excluded when the primary application domain was not related to cloud operations (e.g., ecology-only, logistics-only) unless used strictly for conceptual background and clearly marked as non-core.

Each reference was conceptually assigned to one or more of four analytical streams (themes), which form the structure of both the Related Work and the comparative synthesis:

- ***Predictive Workload Analysis (Forecasting)***. Time-series forecasting, demand prediction, concept drift handling, feature engineering for workload shape and seasonality, and how forecasts feed into autoscaling/capacity decisions.
- ***Dynamic Resource Allocation and Scheduling***. RL-based scheduling, evolutionary/meta-heuristic optimization, hybrid algorithms, multi-objective placement, VM/container scheduling, and constraint handling (quotas, affinity/anti-affinity, heterogeneity).
- ***Cost- and Energy-Aware Optimization***. Cost-aware scheduling, spot/reserved planning, DVFS and consolidation strategies, joint objectives (cost–energy–QoS), and FinOps-relevant telemetry.
- ***AI-Enhanced Load Balancing, Reliability, and Security Constraints***. Traffic distribution policies, latency/error reduction, stability under failures/bursts, reliability-aware balancing, trust-aware allocation, and integration of security constraints (e.g., secure workflows, threat-aware scheduling).

Within each stream, studies were analyzed using a consistent comparison lens:

- Technique class (ML forecasting / RL / meta-heuristics / hybrid / multi-objective optimization)
- Target objective(s) (QoS/SLO, utilization, cost, energy, reliability/security)
- Evaluation context (simulator vs. production traces; single-cloud vs. multi-cloud; edge/fog involvement)

- Reported metrics and claims (direction and magnitude when provided)
- Limitations and threats (reproducibility, generalization, stability, overhead, missing benchmarks)

When possible, the comparative synthesis aggregated reported intervals for improvements (eg in cost, utilization or latency) to derive cross-study summary observations, without pretense of statistical meta-analysis. The cross-stream synthesis then identifies interdependencies (e.g., forecasting → allocation decisions; cost objectives → load balancing choices), highlights recurring open gaps (benchmarks, portability of trust metrics, explainability, lifecycle cost models), and supports the derivation of a conceptual reference framework.

Studies were included if they addressed at least one cloud administration function (forecasting, allocation/scheduling, cost–energy optimization, or load balancing), employed an AI/ML/optimization technique for operational decisions, and reported operational outcomes or provided a comparative synthesis. Studies were excluded when the application domain was not clouding operations or when AI techniques were not linked to actionable management decisions. Quality appraisal: Evaluation of the quality of studies was performed based on several key dimensions: (i) clarity of system and load modeling, (ii) realism of the evaluation context (simulation vs. real tracks or prototypes), (iii) presence of comparative baselines, (iv) completeness of reported metrics, and (v) discussion of limitations and threats to validity. This assessment was not used to exclude studies, but to weigh the strength of evidence during comparative synthesis and interpretation of results. Figure 8 depict the PRISMA flow diagram of the study selection process

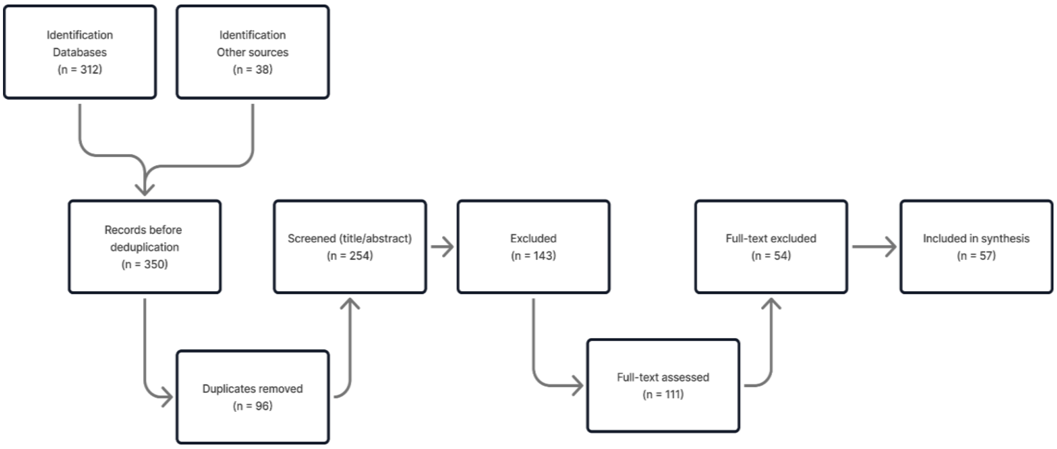

Figure 8. PRISMA Flow Diagram of the Study Selection Process

The review is guided by the following research questions which are as follows:

- RQ1: What evidence does the literature provide that short-horizon AI-based forecasting reduces SLO violations and resource over-provisioning compared with reactive policies?

- RQ2: How do RL and meta-heuristic schedulers compare with rule-based baselines in latency, makespan, utilization, and stability under volatile workloads?
- RQ3: What trade-offs are reported between cost, energy efficiency, and QoS/reliability in state-of-the-art cost-aware optimization methods?
- RQ4: To what extent do AI-enhanced load balancing approaches improve latency/error performance and robustness under failures and multi-cloud constraints?

The review covers literature published between 2016 and 2025, capturing fundamental optimization and machine learning approaches as well as recent advances in cloud-native, multi-cloud, and AI-driven cloud management. This review is limited by several methodological threats common to the existing literature. First, a significant proportion of studies rely on simulations or simplified setups, increasing the risk of bias towards real production environments. Second, the heterogeneity of metrics and assumptions (eg energy models, QoS definitions) limits direct comparability between studies. Third, publication bias may affect the overestimation of positive results, as failures or compromises are often not fully reported. These limitations are taken into account in the interpretation of the results and reinforce the need for standardized benchmarks and more reproducible assessments.

RESULTS AND FINDINGS

This section presents the results of the comparative literature synthesis structured according to the research questions. In the absence of new empirical evaluation, the results interpret the evidence reported in existing studies by identifying recurring patterns, typical improvement intervals, and key trade-offs between performance, cost, energy, and reliability. Where possible, reported benefits are aggregated in a summary fashion to draw cross-study conclusions, while also highlighting limitations and contexts in which these results remain valid.

RQ1 – Forecasting, SLO & Over-provisioning

The literature findings show that short-term load forecasting is a fundamental element for moving from reactive management to proactive cloud control. Through empirical findings on real workloads demonstrate that burstiness, seasonality and non-stationarity directly affect the stability of autoscaling and the risk of SLO violations [2]. Advanced model-based approaches, such as DeepAR and LSTM, report significant reduction in over-provisioning and improved resource efficiency when used in testbed or real-trace environments [41]. However, most works treat prediction as an isolated module [2], without analysing how prediction errors translate into poor allocation decisions and QoS degradation in multi-cloud environments with continuous drift [15]. Specifically, evidence supports that forecasting can reduce over-provisioning and SLO violations, but only when it is directly integrated with allocation and operational control mechanisms. Overall, the evidence from the reviewed studies shows that short-term forecasting models report typical reductions in over-provisioning in the range of about 20–60%, with sustained reductions in SLO violations only in cases where forecasting is directly linked to autoscaling and scheduling mechanisms. When forecasting is evaluated as an isolated module, operational benefits remain unstable under concept-drift and multi-cloud variability.

RQ2 – RL & Meta-heuristics for Allocation/Scheduling

In the field of dynamic allocation and scheduling, reinforcement learning and multi-objective meta-heuristics are reported to outperform static heuristics in utilization [7-12], make span and energy efficiency, especially under volatile loads [19-25]. Hybrid and swarm-based algorithms demonstrate advantages in convergence and Pareto quality in scenarios with competing objectives [29-37]. However, systematic reviews highlight recurring limitations: high optimization overhead, sensitivity to parameters, and lack of evaluation in real multi-cloud configurations [9]. These problems are directly related to RQ1, since intelligent allocators depend on the quality and stability of the prediction signals [1-7]. RL and meta-heuristics offer significant operational improvements, but their performance remains conditional on integration with forecasting and realistic modelling of multi-cloud constraints. Overall, the reviewed studies show that schedulers based on RL and meta-heuristics report typical improvements in utilization and makespan in the order of 15–30% compared to static heuristics, however these benefits depend significantly on the quality of the prediction signals, the parameterization of the algorithms, and the realism of the modelled constraints in multi-cloud environments.

RQ3 – Cost–Energy–QoS Trade-offs

Cost–energy optimization has moved from simple “min-cost” formulations to multi-objective approaches [11], that balance monetary cost, energy, and QoS/SLO penalties [35]. Various studies report significant reductions in energy and OPEX through consolidation, DVFS and cost-aware scheduling [46], but often with increased risk of performance instability under load bursts [57]. A persistent shortcoming is that cost and energy are measured with heterogeneous assumptions, while the interaction of cost-aware decisions with reliability and scheduling is rarely analyzed in an integrated way [15]. This directly links RQ3 to RQ2 and RQ4, as cost decisions directly affect allocation and load balancing policies. The literature confirms the existence of strong trade-offs between cost, energy and QoS, emphasizing the need for coordinated optimization with scheduling and load balancing. In summary, the reviewed studies report cost and energy reductions typically ranging from 20% to over 70%, particularly through consolidation and price-aware planning. However, these benefits are often associated with increased latency variability or the risk of QoS degradation during unexpected loads, indicating that cost optimization without coordination with planning and load balancing can damage operational stability.

RQ4 – AI Load Balancing under Failures & Multi-cloud

AI-enhanced load balancing is reported to improve latency and error-rate compared to static policies, especially under unstable traffic and failure scenarios [22-23]. Reliability-aware and security-aware approaches show that the integration of reliability signals can reduce service degradation and operational costs in cloud environments [45]. However, many of these solutions work as black-box controllers, with limited explainability and little analysis on routing stability in service-mesh and multi-cloud environments [33]. These constraints are directly related to RQ2 and RQ3, since load balancing must respect allocation decisions and cost/energy constraints. The findings show that robustness, but requires stronger integration with scheduling, cost-aware policies and explainability mechanisms. Overall, the evidence from the reviewed studies suggests that AI-driven load balancing approaches report typical improvements in latency and error rate in the range of 10–40% compared to static policies, especially in scenarios with unstable traffic or failures. However, these benefits are conditional on integration with scheduling and cost optimization mechanisms, as well as the presence of explainability mechanisms and security boundaries, as black-box controllers can cause routing instability in service-mesh and multi-cloud environments.

Table 1 depict the comparative evidence matrix with architectural role, decision interfaces and constraints

Table 1. Comparative evidence matrix with architectural role, decision interfaces and constraints

Ref.	Stream	Techn. / Model	Evaluation Context	Key Metrics	Key Findings	Limitations / Gaps	Decision Output (Action/ Artifact)	Architectural Role	Constraints Modeled	Pros / Cons (SOTA)
[41]	Forecasting	DeepAR, LSTM, ARIMA	CloudAIBus testbed, prod traces	CPU over-prov., MAE/MAPE	DeepAR cuts unused CPU from ~98% to ~32%	Weak coupling to allocators	Forecast (demand) + (implicit) capacity signal	Forecasting Engine	Drift/uncertainty partly; operational constraints not explicit	+ realistic traces/testbed; – limited end-to-end control integration
[2]	Forecasting	Workload characterization	Large-scale cloud traces	Burstiness, diurnal patterns	Reveals strong non-stationarity	No control/action layer	Workload properties/insights (no direct action)	Telemetry/Analytics input to Forecasting	No explicit constraints (analysis study)	+ strong empirical grounding; – no decision/control outputs
[21]	Forecasting / Compliance	RF, SVM, MSMO	Distributed systems	Detection accuracy, SLA	ML effective for violation detection	Limited scaling integration	Violation risk / compliance alerts	Policy & Constraint Engine (Guardrails) (partial)	Compliance rules partially; no quotas/placement	+ policy/compliance signal; – weak coupling to orchestration actions
[31]	Allocation / Reliability	Whale + multi-objective	Edge–cloud sim	Energy, reliability	+25–55% reliability, −energy	Simulator-only	Placement/scheduling plan (Pareto solution)	Scheduler/Allocator (+ reliability objective)	Reliability objective; limited real multi-cloud rules	+ multi-objective incl. reliability; – sim-only realism
[7]	Allocation	Hybrid GOA/SOA	Multi-cloud	Makespan, cost, energy	Better Pareto fronts	No real multi-	Workflow schedule	Scheduler/Allocator	Cost/energy modeled; quotas/affini	+ strong multi-objective search;

Ref.	Stream	Techn. / Model	Evaluation Context	Key Metrics	Key Findings	Limitations / Gaps	Decision Output (Action/ Artifact)	Architectural Role	Constraints Modeled	Pros / Cons (SOTA)
			workflows			cloud quotas	+ resource assignment		ty not modeled	− missing real constraints (quotas/affinity)
[37]	Allocation	PSO + ACO	CloudSim	Energy, delay	Hybrid swarms outperform single	Tuning sensitive	Resource allocation plan	Scheduler/Allocator	Basic constraints (capacity); not policy-rich	+ hybrid improves convergence; − sensitive + sim assumptions
[23]	Allocation	Intelligent workflow sched.	IoT–cloud	Throughput	ML improves workflow QoS	Legacy workloads	Workflow schedule / task mapping	Scheduler/Allocator	Limited constraints; platform assumptions older	+ QoS-aware workflow mapping; − legacy workload + limited modern constraints
[25]	Allocation	Multi-objective DSS	Analytical + sim	Utilization, cost	Balanced objectives	Abstracted infra	Decision support recommendations (allocation choices)	Scheduler/Allocator + Optimizer (partial)	Abstract constraints; limited real provider rules	+ structured DSS; − abstraction reduces deployability
[9]	Survey	SLR	Literature	—	Taxonomy of schedulers	No synthesis framework	Taxonomy/insights (no action)	N/A (Review)	N/A	+ broad coverage; − lacks system-level integration framework
[11]	Cost/Energy	Power modeling + DVFS	Cloud DC	Energy, makespan	Accurate power estimation	Needs forecast coupling	Power/energy model + DVFS policy guidance	Cost–Energy–QoS Optimizer (input)	Energy model; not linked to QoS+policies end-to-end	+ stronger energy modeling; − not integrated with forecasting/planning loop
[35]	Cost/Energy	RPSEO	Scientific workflows	Cost, energy	−44% energy, −74% cost	Domain-specific	Cost/energy-optimized schedule	Cost–Energy–QoS Optimizer (+ scheduling)	Cost+energy ; QoS constraints limited	+ strong savings; − domain-specific + transferability unclear
[16]	Cost/Energy	Joint cost–energy model	Cloud sim	Cost, reliability	Trade-off modeling	Limited QoS depth	Trade-off decision rules / plan selection	Cost–Energy–QoS Optimizer	Cost+reliability; QoS modeling weak	+ explicit trade-off modeling; − QoS depth + sim setting
[15]	Cost / Multi-cloud	BI cost framework	Multi-cloud BI	OPEX	Reserved + placement	BI-specific	FinOps policy/planning framework	Policy Engine + Cost Optimizer (conceptual)	Budget/cost; security partly; scheduling	+ governance/FinOps view; − narrow domain

Ref.	Stream	Techn. / Model	Evaluation Context	Key Metrics	Key Findings	Limitations / Gaps	Decision Output (Action/ Artifact)	Architectural Role	Constraints Modeled	Pros / Cons (SOTA)
					saves cost				detail limited	+ limited system design detail
[57]	Cost	Cost-aware Spark sched.	Apache Spark	Latency, cost	Price-aware improves SLA	Platform-specific	Job scheduling + cost-aware placement	Scheduler/Allocator + Cost Optimizer	Cost+deadline/QoS; multi-cloud constraints limited	+ platform-realistic; – portability beyond Spark unclear
[22]	Load balancing	GWO (reliability-aware)	CloudSim	Resp. time, cost	Reliability improves LB	Static thresholds	LB decision (traffic distribution)	Load Balancing Controller	Reliability signal; limited policy/security	+ reliability-aware LB; – sim + simplistic control assumptions
[33]	Load balancing	RL + optimization	Service-level sim	Latency, efficiency	AI LB beats heuristics	Explainability lacking	Routing policy / LB weights	Load Balancing Controller	Some performance constraints; stability/guardrails unclear	+ adaptive LB; – black-box + explainability gap
[56]	Load balancing	Krill Herd	Cloud sim	Makespan	Swarm LB effective	Outdated assumptions	Task distribution / LB policy	Load Balancing Controller	Minimal constraints; old infra assumptions	+ baseline swarm idea; – outdated + low realism
[49]	Reliability	AI automation	Cloud infra	Downtime	Predictive automation	Vendor-centric	Reliability alerts / automation actions	Telemetry + Guardrails (partial)	Reliability focus; limited portability	+ ops-oriented reliability; – vendor-centric/generalization unclear
[45]	Security / Scheduling	SEWS (secure WF)	Heterogeneous cloud	Energy, time	−79% time, −90% energy	Crypto overhead	Secure workflow schedule	Policy/Constraint Engine + Scheduler	Security constraints explicit; cost/QoS trade-offs limited	+ explicit security constraints; – overhead + limited end-to-end QoS analysis
[47]	Security / Multi-cloud	Hybrid privacy	Multi-cloud	Confidentiality	Improved availability	No cost modeling	Privacy/availability mechanism (policy)	Policy & Constraint Engine	Privacy/availability; no cost/energy	+ multi-cloud privacy focus; – missing cost/FinOps linkage
[1]	Trust / Allocation	Trust-aware AI	Multi-cloud	Delay, integrity	Trust improves placement	Trust metric portability	Placement decision with trust weighting	Policy/Constraint Engine + Scheduler	Trust/integrity constraints; portability weak	+ trust-aware placement; – trust metric not standardized across providers
[36]	Edge–Cloud	Edge pre-screen + LSTM	IoT–edge–cloud	Latency, energy	−35% latency, −28% energy	Limited scale	Forecast/ anomaly signal +	Telemetry + Forecasting (split)	Limited constraints; scale limits	+ edge+cloud split improves responsiveness; – limited scale

Ref.	Stream	Techn. / Model	Evaluation Context	Key Metrics	Key Findings	Limitations / Gaps	Decision Output (Action/ Artifact)	Architectural Role	Constraints Modeled	Pros / Cons (SOTA)
							control hint			+ unclear integration depth
[24]	Allocation	ML bin-packing	Analytical	Utilization	ML improves packing	No failure analysis	Placement plan (bin-packing decision)	Scheduler/Allocator	Capacity constraints only; no failures/policies	+ efficient packing; – lacks failures/guardrails realism
[52]	Reliability	Hybrid approach	Cloud sim	Availability	Better reliability	No cost linkage	Reliability mechanism/strategy	Guardrails / Reliability management (conceptual)	Availability; no cost/QoS integration	+ reliability improvement; – not linked to cost/QoS/scheduling

Comparative Synthesis and Conceptual Framework

This section consolidates the cross-stream evidence from RQ1–RQ4 into a system-engineering view of AI-driven cloud administration. Rather than proposing a novel algorithm, the section derives a reference architecture that formalizes how forecasting, allocation/scheduling, cost–energy–QoS optimization, and AI-enhanced load balancing interact as an end-to-end decision pipeline. The literature synthesis in Section 4 indicates that reported improvements are often demonstrated in isolation (e.g., forecasting accuracy or scheduler makespan), while chain effects across the management loop remain under-specified. In production-like settings, however, uncertainty in forecasting propagates into planning decisions, and aggressive consolidation or cost-driven placement can amplify QoS instability during burst regimes and failures.

To address the reviewer-identified gap in technical depth, the proposed conceptual framework is specified here as a reference architecture with explicit system boundaries, data flows, state variables, and decision interfaces. This architecture is grounded in recurring patterns across the reviewed studies, including (i) short-horizon predictive signals [2] to reduce reactive oscillations [7], (ii) constraint-aware allocation and scheduling under heterogeneity [1-6], (iii) multi-objective cost–energy optimization with QoS penalties [35-40] and (iv) adaptive load balancing under volatility, reliability, and security constraints [30-35]. The following subsections introduce a scenario-based context, define the architecture at context and internal levels, and specify the operational control loop (5.3) that closes the pipeline with feedback and drift monitoring.

Scenario Based System Context

This section presents a typical scenario that embodies recurring patterns identified in the researched literature, seeking to contextualize the reference architecture within a practical operating environment, without asserting the existence of an implemented prototype. The scenario is purposefully aligned with the system boundaries and data flows illustrated in Figures 9–13, encapsulating the principal sources of complexity that drive AI-driven cloud administration: variable workloads, multi-cloud heterogeneity, policy limitations, and environments susceptible to failure. [1-6].

As shown in Figure 9, think about a microservice-based software that runs on Kubernetes clusters from two different cloud providers, Cloud-A and Cloud-B. The application uses a service mesh/load balancer to send requests to stateless microservices that rely on stateful backend databases that are hosted independently in each cloud location. The system can add edge nodes for requests that are sensitive to latency, but the main control logic is all in the cloud.

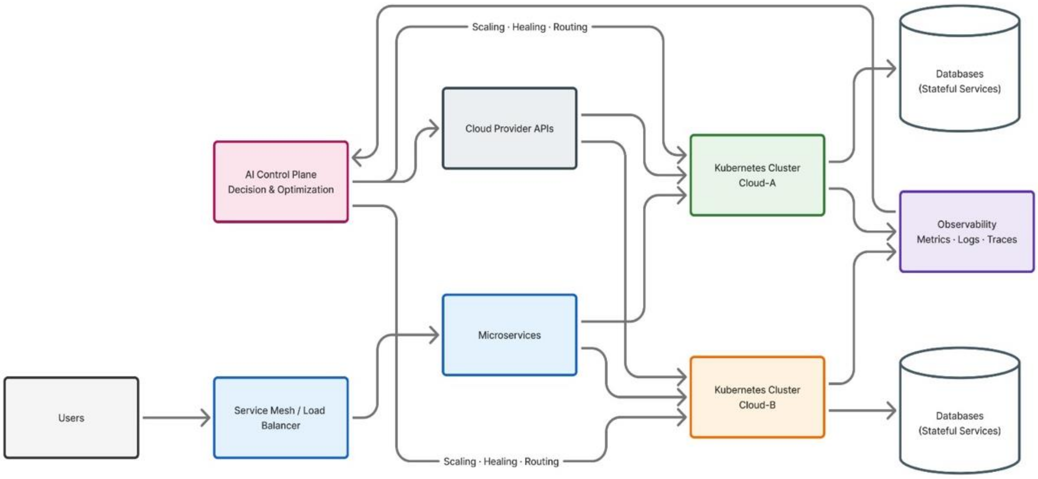
Scaling · Healing · Routing
Databases (Stateful Services)
Cloud Provider APIs
AI Control Plane Decision & Optimization
Kubernetes Cluster Cloud-A
Observability Metrics · Logs · Traces
Microservices
Users
Service Mesh / Load Balancer
Kubernetes Cluster Cloud-B
Databases (Stateful Services)
Scaling · Healing · Routing

Figure 9. AI-driven Multi-Cloud Kubernetes Administration

The workload has strong daily patterns and random spikes in promotions, which cause unexpected changes in the regime that are similar to concept drift. The system is also vulnerable to regional failures and partial outages, including node unavailability or degraded network pathways within a single provider [2]. These traits are similar to what has been found in large-scale workload studies and research on multi-cloud orchestration [15].

The cloud operator defines a set of explicit and competing objectives that must be enforced continuously:

- QoS / SLO objectives: latency targets (e.g., p95 response time), error-rate thresholds, and availability requirements at the service level.
- Resource constraints: provider-specific quotas, affinity and anti-affinity rules for microservice placement, and heterogeneous VM/container capabilities across Cloud-A and Cloud-B.
- FinOps constraints: a monthly OPEX budget, dynamic pricing signals (on-demand, spot, reserved instances), and cross-region cost differences.

- Reliability and security policies: minimum availability targets, failure-domain isolation rules, and secure placement requirements for sensitive microservices or data.

These constraints are not static; pricing, resource availability, and failure conditions evolve over time, requiring adaptive decision-making rather than fixed heuristics.

The core control problem addressed by the AI-driven cloud administration platform is to translate continuously arriving telemetry and external signals into coordinated operational decisions. Inputs include real-time observability data (metrics, logs, traces), workload forecasts with uncertainty estimates, pricing and energy signals, and explicit policy constraints. Based on these inputs, the system must:

1. Forecast short-horizon demand and uncertainty, accounting for burstiness and drift.
2. Generate feasible scaling and placement plans across Cloud-A and Cloud-B that satisfy quotas, affinity rules, and security constraints.
3. Select cost–energy trade-offs that respect budget limits while minimizing the risk of QoS degradation.
4. Adapt traffic routing and failover policies through the service mesh to maintain latency and error-rate stability under bursts or failures.

Execution feedback (resource utilization, observed QoS, failure events) is continuously fed back into the control plane, closing the loop and enabling corrective actions such as conservative fallback policies or model retraining when instability or drift is detected.

Within this scenario, the AI-driven cloud administration platform functions as a decision control plane, interfacing with:

- Cloud provider APIs (for scaling, placement, and resource management),
- Kubernetes clusters in multiple clouds,
- Service mesh and load balancing components, and
- Observability and monitoring systems.

This scenario provides the concrete operational context for the reference architecture detailed in mentioned section. The following diagrams formalize this context first at the system boundary level (DFD Level 0) and then through an internal architectural decomposition (DFD Level 1), making explicit how AI-driven forecasting, scheduling, cost–energy optimization, and load balancing interact within a unified control loop.

Reference Architecture of AI-Driven Cloud Administration

Based on the comparative synthesis, the proposed framework is specified as a reference architecture that makes the cloud administration pipeline explicit at the level of system boundaries, inputs/outputs, internal modules, and decision interfaces. The architecture is defined around three core elements:

i. Inputs that represent telemetry and operational context;
ii. state that represents the current system configuration and model confidence; and

iii. outputs that represent actionable decisions for scaling, placement, and routing. This definition enables the review to compare SOTA approaches not only by algorithm class, but by how they integrate into a closed-loop system.

Architecture Definition (System View).

The main inputs include:

1. observability telemetry (metrics, logs, traces)
2. short-horizon workload forecasts and uncertainty summaries
3. pricing and cost signals (on-demand/spot/reserved price, egress considerations)
4. energy or power models (measured or proxy-based)
5. constraints and policies (quota, affinity/anti-affinity, region rules, security/compliance rules)
6. reliability and failure signals (health checks, error bursts, incident alerts).

State includes: current cluster and service state (resource utilization, replica counts, node availability), current allocation/placement snapshot, recent routing weights, and ML/RL control state such as model confidence and drift indicators.

Outputs include:

1. proactive scaling plans (replicas/VMs, headroom targets)
2. placement and scheduling decisions (node/region/provider selection with constraint satisfaction)
3. cost–energy trade-off actions (consolidation/DVFS suggestions, instance mix selection)
4. load balancing actions (routing weights, failover rules, circuit-breaker thresholds)
5. operator-facing alerts (budget risk, predicted SLO violation risk, policy violations).

This architecture addresses a key limitation observed across the literature: many studies report improvements within a single function (e.g., forecasting error or scheduler makespan) but do not specify how decisions are coordinated end-to-end under realistic constraints. By making the pipeline explicit, the architecture supports a consistent comparison of integration patterns: open-loop vs. closed-loop control, uncertainty-aware vs. point-estimate planning, and policy-guarded vs. unconstrained optimization.

High-Level Context Diagram (DFD Level 0)

Figure 10 presents the high-level context diagram (DFD Level 0) of the AI-driven cloud administration platform. At this abstraction level, the platform is modelled as a decision control plane with a clearly defined system boundary, focusing on external actors and data flows, while deliberately abstracting internal decision logic and algorithms. The platform receives operator intent from human stakeholders (Admin / SRE / FinOps), including service-level objectives (SLOs), budget constraints, and security policies. In parallel, it continuously ingests runtime telemetry metrics, logs, and traces produced by the observability stack and the service mesh/load balancing layer. These inputs represent the operational state of applications, infrastructure resources, and traffic behaviour.

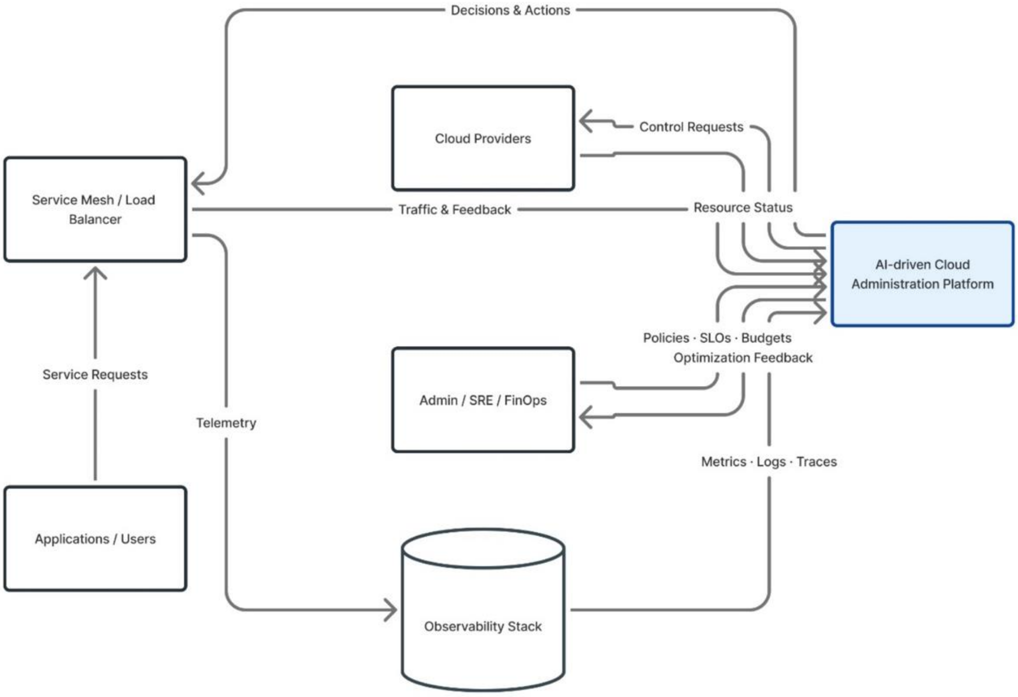

Figure 10. DFD Level 0 – AI-driven Cloud Administration Platform

Based on this information, the AI-driven cloud administration platform issues operational control actions to heterogeneous cloud execution environments through cloud provider and orchestration interfaces. These actions include scaling, placement, healing, and routing decisions, which are enforced by cloud providers and container orchestration platforms. Execution status, resource state, and QoS feedback are continuously returned to the platform, closing the control loop at the system boundary level.

This DFD Level 0 view clarifies who interacts with the platform, what information is exchanged, and where the system boundary lies, without exposing internal architectural modules. By separating the external interaction context from internal decision mechanisms, the diagram addresses a key limitation observed in many state-of-the-art approaches, where system boundaries and decision responsibilities are left implicit or conflated with algorithmic details. The internal decomposition of the platform into forecasting, policy reasoning, scheduling, cost–energy optimization, and load-balancing components is presented in mentioned section

Internal Architecture Decomposition (DFD Level 1 – UML Component)

Figure 11 displays the internal architectural breakdown of the AI-driven cloud administration platform (DFD Level 1 / UML component view). The internal decision-making pipeline, the main functional parts, and the clear data and decision contracts between them are all shown in this picture. The DFD Level 0 context diagram, on the other hand, looks at external actors and system boundaries. The breakdown is based on a meta-analysis of the best ways to use AI to manage cloud services. It formalizes the functional roles that are typically addressed in isolation within the literature. The architecture shows cloud management as a closed-loop decision pipeline, where forecasting, policy enforcement, scheduling, optimization, and load balancing are all linked together instead of being looked at separately.

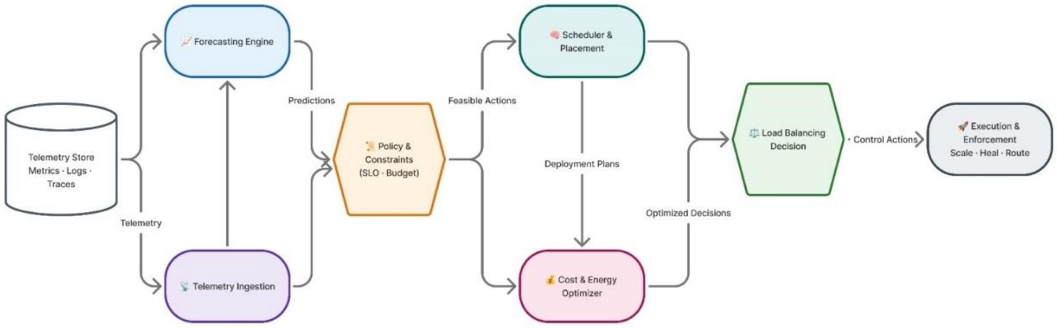

Figure 11. DFD Level 1 – Internal Decision Pipeline (Modern Architecture)- (Author design)

Telemetry Ingestion and Feature Pipeline: The telemetry pipeline gathers data from the observability stack and service mesh and puts it all together. We put raw signals in order by time, filter them, and turn them into organized characteristics that can be used later for inference and decision-making. A feature store keeps track of both new and old features. This lets models be updated both online and offline. This lets you check for drift and make sure that the results can be repeated. Contract:

- Input: health signals, measurements, logs, and traces
- Output: feature vectors that are time-aligned and statistics that have been combined

The forecasting engine uses data from the telemetry pipeline to produce short-term workload predictions at time t+h. It also makes summaries of uncertainty, such as quantiles or prediction intervals [2]. These outputs let you plan for capacity with risk in mind and do something before problems happen [14]. Drift signals based on prediction residuals or changes in feature distribution might lead to retraining, fallback heuristics, or cautious decision modes, as shown in previous work [41-49]. Contract:

- Input: feature vectors and prior demand
- Output: estimates of demand and explanations of uncertainty

Policy and Constraint Engine (Guardrails): The policy and constraint engine combines the operator's goals with the system's rules, like SLO objectives, budget limits, quota limits, affinity/anti-affinity rules, and security and compliance requirements. This section doesn't only filter out candidates; it also checks and limits what they can do to make sure they are safe and possible in multi-cloud settings. [1-7]. Contract:

- Input: candidates' predictions and actions
- Output: feasible action space and signs that constraints are being met

The scheduler looks at signs of low demand and makes plans for where to put things and how to scale them. It picks nodes, regions, or providers that fit both operational needs and predicted demand. The literature proposes numerous alternative methodologies, including reinforcement learning, evolutionary strategies, and hybrid meta-heuristics [1-7]. But the architectural contract maintains the same: the scheduler must give unambiguous deployment plans with expected QoS and resource consequences. [19]. Contract:

- Input: guesses and probable actions
- Output: placement and scaling plans that are sure to operate as long as specific conditions are met

The Cost–Energy–QoS Optimizer looks at the trade-offs between money, energy use, and QoS penalties. It also makes scheduling outputs better when there are more than one plan [11]. It shows operator-tunable trade-off knobs (such aggressiveness vs. robustness) by combining cost models (including on-demand [35], spot, and reserved pricing) [46], energy proxies or power models, and stability penalties. This is in line with multi-objective formulations in SOTA investigations [57]. Contract:

- Input: deployment plans, cost/energy models
- Output: the best plans with trade-offs that can be measured

At runtime, the load balancing controller updates routing weights, failover rules, and resilience policies based on predicted demand and health signals [22]. This component is distinct from black-box routing approaches since it clearly lists the requirements for reliability and security [30-33]. This makes it less likely that there will be routing problems during bursts or failures. [45]. Contract:

- Input: plans that are optimized, health and traffic signals
- Output: policies for routing and reliability

Execution adapters turn vague decisions into actions that can be carried out in the cloud. They do this by using Kubernetes APIs, cloud provider APIs, and service mesh configuration endpoints. The telemetry pipeline gets back the status of the execution and any effects that were detected. This closes the internal control loop and allows for ongoing correction. Contract:

- Input: decisions concerning control
- Output: current status of execution and resources

Figure 12 depict the Internal Decision Pipeline of the A-Driven Cloud Administration Platform (Author design). Why this architecture is more than merely SOTA fragmentation - The literature study indicates that forecasting is frequently seen as an independent accuracy issue, schedulers are evaluated in simulators devoid of real-world constraints, cost-energy optimization is not associated with reliability, and load balancing typically functions as a black-box controller.

The suggested internal structure, on the other hand, clearly combines:

1. Uncertainty and drift propagation from forecasting into planning;

2. Constraint-based feasibility as a first-class decision stage;

3. Multi-objective cost–energy–QoS reasoning before execution;

4. Reliability- and security-aware routing within the same control loop.

The architecture provides a system engineering reference grounded in SOTA evidence, rather than a collection of disparate algorithms, by delineating these interactions into distinct components and agreements.

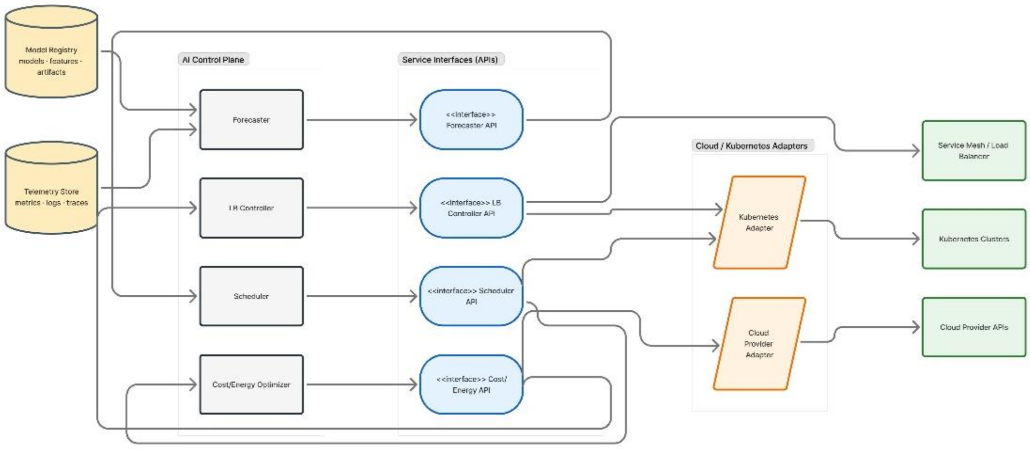
Model Registry
models · features · artifacts
Telemetry Store
metrics · logs · traces
AI Control Plane
Service Interfaces (APIs)
Forecaster
LB Controller
Scheduler
Cost/Energy Optimizer
<<interface>> Forecaster API
<<interface>> LB Controller API
<<interface>> Scheduler API
<<interface>> Cost/ Energy API
Cloud / Kubernetes Adapters
Kubernetes Adapter
Cloud Provider Adapter
Service Mesh / Load Balancer
Kubernetes Clusters
Cloud Provider APIs

Figure 12. Internal Decision Pipeline of the A-Driven Cloud Administration Platform

Figure 13 specifies the end-to-end operational control loop corresponding to the reference architecture. The loop begins with continuous telemetry ingestion and periodic (or event-triggered) forecasting. Forecast outputs, including uncertainty summaries, are passed through policy guardrails to ensure feasibility with respect to quotas, affinity rules, and security constraints. The scheduler/allocator then generates candidate scaling and placement actions, which are evaluated by the cost–energy–QoS optimizer to select an action that balances OPEX, energy, and SLO risk according to operator-defined trade-off parameters. Finally, the load balancing controller updates routing weights and resilience policies to stabilize latency and error rates during bursts and failures.

A critical closed-loop element is feedback and stability management: after actions are executed, observed QoS and utilization outcomes are measured; if SLO violations increase or drift indicators rise, the platform may reduce aggressiveness (e.g., conservative scaling headroom, safer routing weights) [9], activate fallback heuristics, or trigger model retraining. This formalization aligns with the synthesis findings that benefits are more robust when AI-driven methods operate as part of an integrated, observable [33], policy-aware control loop rather than isolated optimizers [35]. The control loop therefore serves as a practical "design map" for how evidence from forecasting, scheduling, cost/energy optimization, and load balancing should be integrated and evaluated end-to-end.

Although this review does not propose an empirical benchmark or implementation, the control loop in Figure 13 defines a minimal system-level evaluation blueprint implied by the surveyed literature, the reference architecture implies a minimal evaluation blueprint for future work: (i) end-to-end metrics across forecasting, scheduling, cost and load balancing; (ii) stress scenarios including burst demand and failures; and (iii) comparison against reactive baselines under identical constraints. This blueprint provides a consistent lens for evaluating future AI-driven cloud controllers beyond isolated component metrics.

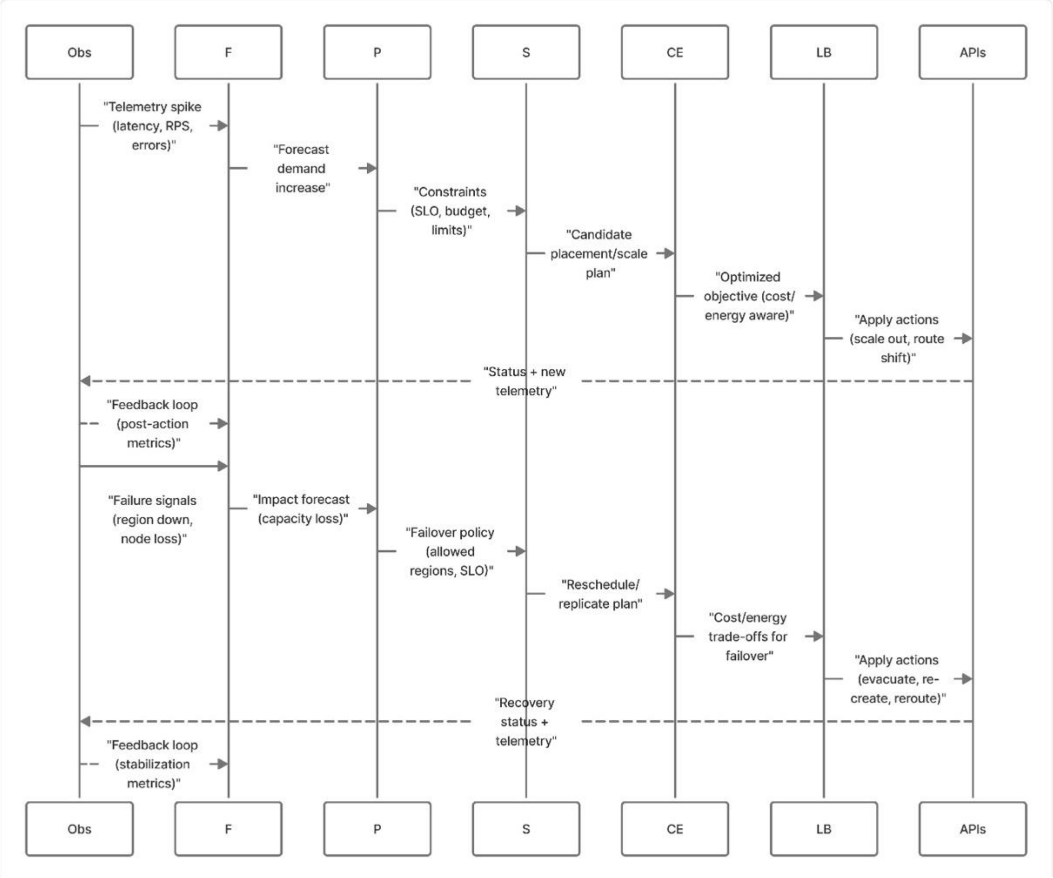
Obs
F
P
S
CE
LB
APIs
"Telemetry spike (latency, RPS, errors)"
"Forecast demand increase"
"Constraints (SLO, budget, limits)"
"Candidate placement/scale plan"
"Optimized objective (cost/ energy aware)"
"Apply actions (scale out, route shift)"
"Status + new telemetry"
"Feedback loop (post-action metrics)"
"Failure signals (region down, node loss)"
"Impact forecast (capacity loss)"
"Failover policy (allowed regions, SLO)"
"Reschedule/ replicate plan"
"Cost/energy trade-offs for failover"
"Apply actions (evacuate, re-create, reroute)"
"Recovery status + telemetry"
"Feedback loop (stabilization metrics)"
Obs
F
P
S
CE
LB
APIs

Figure 13. UML Sequence Diagram – Decision Loop (Traffic Surge + Region Failure). *Obs-Observability stack; F-Forecasting engine; P-Policy & constraint engine; S- Scheduler/Allocator; CE-Cost–Energy–QoS optimizer; LB-Load balancing controller.*

Evaluation Framework and Metrics for Ai-Driven Cloud Administration

This review does not introduce a new algorithm or an empirical benchmark; nonetheless, the comparative synthesis and the reference architecture outlined in mentioned sections indicate a unified assessment framework for AI-driven cloud administration. A prevalent deficiency in the examined literature is that evaluations often concentrate on isolated components, such as forecasting accuracy or scheduler performance, neglecting the comprehensive behaviour of the entire control loop in realistic operational scenarios. This section addresses the gap by presenting an evaluative perspective grounded in the best available facts and aligned with the end-to-end control loop mentioned in Figure 13.

The proposed evaluation method is clearly system-level and end-to-end, which indicates how AI-driven cloud management works together. Evaluation should not look at each approach on its own. Instead, it should look at how forecasting, scheduling, cost–energy optimization, and load balancing function together when there are common constraints, uncertainty, and feedback. This is consistent with the findings of RQ1–RQ4, which indicated that improvements at the component level do not invariably result in consistent operational enhancements when integration effects are overlooked. The literature identifies multiple persistent evaluation dimensions. Forecasting should be assessed not only through error metrics but also by its impact on over-provisioning, SLO breaches, and stability in the context of concept drift. You should think about how well resources are used, how well they function with different providers, and how well they fit with quotas and affinity rules when you plan and allocate them. Cost- and energy-aware optimization must be assessed simultaneously with QoS results, including distinct trade-offs between cost reduction, energy conservation, and fluctuations in delay or error rates. When assessing load balancing, it should not just look at steady-state latency, but also at how well it can handle surge traffic and failure, as well as how quickly it can recover. System-level control behaviour is also very significant, along with metrics for each portion. The study underscores the importance of evaluating control-loop stability, including action oscillations, convergence time after workload or failure events, and the effectiveness of fallback mechanisms when predictions worsen or constraints increase. These aspects are sometimes overlooked in SOTA evaluations, despite their importance for real-world application. From previous section is seen that the review should happen when things are stressful, as when demand is strong, an area or node fails, prices fluctuate, resources are restricted, or workloads vary. To figure out what the real gain is, distinct from adjusting for specific scenarios, it's vital to compare AI-driven control to reactive or rule-based baselines under the same conditions. This review does not provide a benchmark; instead, the aforementioned framework outlines a minimum evaluation template as indicated by the reference design. Future study should encompass end-to-end measurements, precisely define multi-cloud limitations, and evaluate integrated behaviour rather than analysing isolated components. If evaluations aren't as strict, the benefits that are claimed might only apply to certain scenarios and be hard to use in other situations. as would make the current state of the art even more fragmented.

Implications and Future Research

From a practical perspective, the findings of this review indicate that integrated [9], end-to-end AI-driven cloud administration systems are inherently more stable and operationally robust than isolated optimization of individual components [35]. The reference architecture and control loop defined in the previous section highlight that forecasting, scheduling, cost–energy optimization, and load balancing must be treated as interdependent functions governed by shared constraints, uncertainty, and feedback. For future research, several priorities emerge. First, there is a clear need for reproducible benchmarks based on real or production-like traces that reflect multi-cloud heterogeneity and policy constraints [1], as implied by the evaluation framework outlined in Section 5.4. Second, the lack of standardized reliability and security metrics remains a major obstacle for cross-provider comparison and portability of AI-driven controllers [33]. Third, future studies should emphasize end-to-end evaluation of the full administration pipeline, explicitly analysing how forecast errors, cost trade-offs, and routing decisions propagate across the control loop. Finally, improved explainability, safety guardrails, and trust mechanisms for AI-based controllers are essential to enable adoption in critical and regulated cloud environments.

SUMMARY AND CONCLUSIONS

A thorough and comprehensive analysis of literature published between 2016 and 2025 (N = 57) enabled this review to investigate recent developments in AI-driven cloud management. Through this study, four interrelated administrative functions, such as AI-enhanced load balancing under reliability and security constraints, predictive workload forecasting, dynamic resource allocation and scheduling, and energy cost-QoS optimization, are examined. The main motivation comes from the increasing complexity of modern cloud infrastructures, which are characterized by heterogeneity, multi-cloud deployments and cloud-edge continuities. Literature results show that through reduced provisioning, improved utilization, cost and energy savings, and greater robustness under fluctuating workloads, AI-driven techniques regularly outperform reactive and rule-based policies in stand-alone evaluations, according to benchmarking research. However, there is also a continuing divide in the current state of the art, as shown by the synthesis. There is also a lack of examination of the entire cloud management pipeline when forecasting, planning, cost optimization and load balancing are considered separately, usually in simulator-based systems with different assumptions. In this context, our results have managed to create a theoretical basis for the administration of AI-enabled cloud computing as a closed decision system. The main idea is not to propose a new algorithm or empirical benchmark; instead, it is to formulate clear architectural components, data flows, decision contracts, and feedback mechanisms by deriving system-level design principles from current research.

Our model emphasizes the interaction and mutual determination of operational stability from uncertainty in prediction, constraint-aware planning, cost-energy trade-offs, and reliability-aware load balancing. The paper continues to provide an evaluation plan that the reference design suggests, with an emphasis on comprehensive evaluations under realistic conditions, stress cases, and response dynamics. This perspective goes beyond component-level metrics and provides a unified framework for evaluating future AI-driven cloud controllers in multi-cloud and high-probability-of-failure scenarios. Through this review it is mainly concluded that integrated, transparent, system-level design and evaluation are more important than isolated algorithmic innovation for the operational effectiveness of AI-driven cloud management, thus providing more sustainable research agendas and bringing together different pieces of knowledge about architecture and evaluation.

AUTHOR CONTRIBUTIONS

Conceptualization: L.L.N., and A.D.; Methodology: L.L.N.; Validation: A.I., and B.A.; Investigation: L.L.N; Resources: A.D.; Data Curation, L.L.N; Writing –Original Draft Preparation: L.L.N.; Writing –Review & Editing: A.D.; Visualization: A.I.; Supervision: B.A.; Project Administration: L.L.N.

CONFLICT OF INTERESTS

The authors declare no conflicts of interest.

REFERENCES

1. Prashanth, M.V., Praveen, K.R., Sharma, R., Jain, R., Vaishnav, J., Goyal, M.K. Understanding Multi Cloud Platform: Innovative AI-Assisted Trust-Aware Resource Allocation Technique. *Int. J. Syst. Assur. Eng. Manage.* **2025**, 1–10.
2. Parayil, A., Zhang, J., Qin, X.; Goiri, Í., Huang, L., Zhu, T., Bansal, C. Towards Workload-Aware Cloud Efficiency: A Large-Scale Empirical Study of Cloud Workload Characteristics. In *Proceedings of the 16th ACM/SPEC International Conference on Performance Engineering*; **2025**, pp. 136-146.
3. Ouchaou, L., Nacer, H., Labba, C. Towards a distributed SaaS management system in a multi cloud environment. *Cluster Comput.* **2022**, *25*, 4051–4071.
4. Sarioguz, O. The impact of agentic artificial intelligence on warehouse and delivery operations in modern logistics. *Int. J. Sci. Res. Arch.* **2025**, *15*(3), 1549-1561
5. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V. Support vector regression machines. In *Advances in Neural Information Processing Systems* **1996**, *9*, 155-161.
6. Vanam, L.A.X.M.I., Goel, D.P.U.N.I.T. *Self Service BI: A Roadmap for Empowered Analytics*. Yashita Prakashan Private Limited, **2025**.
7. Mohammadzadeh, A., Masdari, M. Scientific Workflow Scheduling in Multi Cloud Computing Using a Hybrid Multi Objective Optimization Algorithm. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 3509–3529.
8. Sharma, K. Role of AI and machine learning in sustainable innovation. In *Sustainability, Innovation, and Consumer Preference*; IGI Global Scientific Publishing, **2025**, pp 1–28.
9. Awad, W.K., Ariffin, K.A.Z., Nazri, M.Z.A., Yassen, E.T. Resource Allocation Strategies and Task Scheduling Algorithms for Cloud Computing: A Systematic Literature Review. *J. Intell. Syst.* **2025**, *34*, 20240441.
10. Krishnan, R., Durairaj, S. Reliability and performance of resource efficiency in dynamic optimization scheduling using multi agent microservice cloud fog on IoT applications. *Computing* **2024**, *106*, 3837–3878.

11. Tarafdar, A., Sarkar, S., Das, R.K., Khatua, S. Power Modeling for Energy Efficient Resource Management in a Cloud Data Center. *J. Grid Comput.* **2023**, *21*, 10.
12. Kaplan, F., Babalik, A. Performance Analysis of Cloud Computing Task Scheduling Using Metaheuristic Algorithms in Ddos and Normal Environments. *Electronics* **2025**, *14*, 1988.
13. Acharya, B., Panda, S., Das, S., Majhi, S.K., Gerogiannis, V.C., Kanavos, A. Optimizing Task Scheduling in Cloud Environments: A Hybrid Golden Search Whale Optimization Algorithm Approach. *Neural Comput. Appl.* **2025**, *37*, 10851–10873.
14. Karamthulla, M.J., Malaiyappan, J.N.A., Tillu, R. Optimizing Resource Allocation in Cloud Infrastructure Through AI Automation: A Comparative Study. *J. Knowl. Learn. Sci. Technol.* **2023**, *2*, 315–326.
15. Angamuthu, M. Optimizing multi cloud business intelligence: A Framework for Balancing Cost, Performance, and Security. *J. Comput. Sci. Technol. Stud.* **2025**, *7*, 427–437.
16. Hasan, R. A.; Hameed, T. M. Optimizing Cloud Computing: Balancing Cost, Reliability, and Energy Efficiency. *Babylonian J. Artif. Intell.* **2025**, 64–71.
17. Daruvuri, R. Optimized data packet routing in vehicular networks using edge intelligence and GNNs. In *2025 International Conference on Engineering, Technology & Management (ICETM)*; 2025, pp. 1-6.
18. Anand, A., Agarwal, P., Saini, D.K., Gupta, P. Neural Network Based Task Scheduling in Cloud Using Harmony Search Algorithm. In *Sustainable Smart Cities*. Springer, **2022**, pp. 191–204.
19. Mohanraj, T., Santhosh, R. Multi swarm optimization model for multi cloud scheduling for enhanced quality of services. *Soft Comput.* **2022**, *26*, 12985–12995.
20. Ramadi, D., Saputra, H.M. Modelling population dynamics for biodiversity conservation in tropical ecosystems. *Sci. Get J.* **2025**, *2*, 34–43.
21. Thatikonda, K.C. Machine Learning Applications in Distributed System Compliance: From Detection to Prevention. *Inf. Technol. Manage.* **2025**, *16*, 1255–1265.
22. Sefati, S., Mousavinasab, M., Farkhady, R.Z. Load Balancing in Cloud Computing Environment Using the Grey Wolf Optimization Algorithm Based on Reliability: Performance Evaluation. *J. Supercomput.* **2022**, *78*, 18–42.
23. Abualigah, L., Diabat, A., Elaziz, M. A., Intelligent workflow scheduling for big data applications in IoT cloud computing environments. *Cluster Comput.* **2021**, *24*, 2957–2976.
24. Wang, Y., Yang, X. Intelligent Resource Allocation Optimization for Cloud Computing via Machine Learning. *arXiv* **2025**, arXiv:2504.03682.
25. Qi, B., Manoranjitham, M., Zhang, G., Alwabel, A. S. A., Zayani, H.M., Ferrara, M.. Intelligent Multi Objective Decision Support System for Efficient Resource Allocation in Cloud Computing. *Ann. Oper. Res.* **2025**, 1–29.
26. Sharma, S., Kumar, N., Dash, Y., Dubey, A., Devi, K. Intelligent multi cloud orchestration for AI workloads: Enhancing performance and reliability. In *2024 7th International Conference on Contemporary Computing and Informatics (IC3I)*. **2024**, 1421-1426.
27. Ahmadani, A.A.K., Suakanto, S., Fakhrurroja, H., Hardiyanti, M. Improving Creditworthiness Prediction Using Preprocessing Stages and Feature Selection. In *2023 3rd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*. **2023**, pp. 195-200.
28. Alsahaf, A., Petkov, N., Shenoy, V., Azzopardi, G. A Framework for Feature Selection Through Boosting. *Expert Systems with Applications*. **2011**, 187, 115895

29. Prity, F. S., Uddin, K. A., Nath, N., Exploring swarm intelligence optimization techniques for task scheduling in cloud computing: Algorithms, performance analysis, and future prospects. *Iran J. Comput. Sci.* **2024**, *7*, 337–358.
30. Mahmood, S., Hasan, R., Yahaya, N.A., Hussain, S., Hussain, M. Evaluation of the omni secure firewall system in a private cloud environment. *Knowledge* **2024**, *4*, 141–170.
31. Khaleel, M.I. 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*, 10733–10760.
32. Kumar, B.S.A., Sah, B. Enhancing cloud security and resource management: A comprehensive review. In *International Conference on Internet of Everything and Quantum Information Processing*. **2023**, pp. 1-6.
33. Singhal, A., Goel, P. K., Garg, D., Sharma, C. Enhancing Cloud Performance with AI Driven Load Balancing and Optimization Algorithms. In *2024 4th International Conference on Advancement in Electronics & Communication Engineering (AECE)*. **2024**, pp. 1254-1259.
34. Pandi, M., Kumar, A. S., et al. Efficient Workload Allocation in IOT Fog Cloud Architectures for Energy Efficiency and Low Latency Using DEABC algorithm. *KSII Trans. Internet Inf. Syst.* **2025**, *19*. 368-397.
35. Sangani, S., Patil, R., Goudar, R.H. Efficient algorithm for error optimization and resource prediction to mitigate cost and energy consumption in a cloud environment. *Int. J. Inf. Technol.* **2024**, *16*, 2187–2197.
36. Sathupadi, S., Achar, S., Bhaskaran, S. V., Faruqui, N., Al Wadud, M. A., Uddin, J. Edge cloud synergy for AI enhanced sensor network data: A real time predictive maintenance framework. *Sensors* **2024**, *24*, 7918.
37. Prabakar, D., Iskandarova, N., Iskandarova, N., Kalla, D., Kulimova, K., Parmar, D. Dynamic resource allocation in cloud computing environments using hybrid swarm intelligence algorithms. In *2025 International Conference on Networks and Cryptology (NETCRYPT)*. **2025**, pp. 882-886.
38. Avancha, S., Aggarwal, A., Goel, P. Data driven decision making in IT service enhancement. *Journal of Quantum Science and Technology*. **2024**. *1*(3), 10–24.
39. Ma, Y., Jin, J., Huang, Q., Dan, F. Data preprocessing of agricultural IoT based on time series analysis. In *Intelligent Computing Theories and Application*; **2018**, pp 219-230.
40. Kumar, N., Dash, Y., Abraham, A., Choudhary, R. K., Pandey, S. Cloud enhanced machine learning: Exploring the synergy for intelligent automation and optimization. In *Proceedings of the 15th International Conference on Soft Computing and Pattern Recognition*. **2023**, pp 492-499.
41. Velu, S., Gill, S. S., Murugesan, S. S., Wu, H., Li, X. CloudAIBus: A testbed for AI based cloud computing environments. *Cluster Comput.* **2024**, *27*, 11953–11981.
42. Kosicki, M., Tsiliakos, M., ElAshry, K., Tsigkari, M. Big data and cloud computing for the built environment. In *Industry 4.0 for the Built Environment*. Springer, **2021**, pp 131–155.
43. Ma, X., Gao, H., Xu, H., Bian, M. An IoT Based Task Scheduling Optimization Scheme Considering Deadline and Cost Aware Scientific Workflow for Cloud Computing. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 249.

44. Swain, S. R., Parashar, A., Singh, A. K., Lee, C. N., An intelligent virtual machine allocation optimization model for energy efficient and reliable cloud environment. *J. Supercomput.* **2025**, *81*, 237.
45. Pasha, F., Natarajan, J., An intelligent secure and efficient workflow scheduling (SEWS) model for heterogeneous cloud computing environment. *Knowl. Inf. Syst.* **2025**, 67, 6193–6239.
46. Javadpour, A., Nafei, A., Ja'fari, F., Pinto, P., Zhang, W., Sangaiah, A. K. An intelligent energy efficient approach for managing IoE tasks in cloud platforms. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 3963–3979.
47. Pachala, S., Rupa, C., Sumalatha, L. An improved security and privacy management system for data in multi cloud environments using a hybrid approach. *Evol. Intell.* **2021**, *14*, 1117–1133.
48. Ramidi, R. AI driven automation for period closing in cloud ERP systems. *Int. J. Sci. Technol.* **2025**, *16*, 1-17.
49. Bhamidipati, M., AI Driven Automation and Reliability Engineering: Optimizing Cloud Infrastructure for Zero Downtime and Scalable Performance. *J. Comput. Sci. Technol. Stud.* **2025**, *7*, 1006–1015.
50. Azeroual, O. AI4Knowledge: Shaping the future of research data systems. In *2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*. **2025**, pp. 1-7.
51. Alapati, N.K., Dhanasekaran, S. Addressing data quality and consistency issues in cloud based big data environments. In *2025 International Conference on Networks and Cryptology (NETCRYPT)*. **2025**.
52. Shahid, M. A., Alam, M. M., Su'ud, M. M. Achieving Reliability in Cloud Computing by a Novel Hybrid Approach. *Sensors* **2023**, *23*, 1965.
53. Ait El Mouden, R., Asimi, A., A smart mathematical approach to resource management in cloud based on multi objective optimization and deep learning. In *The International Conference on Artificial Intelligence and Smart Environment*. **2023**, pp 166-172.
54. Gao, X., He, P., Zhou, Y., Qin, X. A smart healthcare system for remote areas based on the edge cloud continuum. *Electronics* **2024**, *13*, 4152.
55. Biau, G., Scornet, E. A random forest guided tour. *Test* **2016**, *25*, 197–227.
56. Hasan, R.A., Mohammed, M.N. A Krill Herd Behavior Inspired Load Balancing of Tasks in Cloud Computing. *Stud. Inform. Control* **2017**, *26*, 413–424.
57. Xu, Q., Wang, C., Yu, J., Fei, H., Ren, X. A Cost Aware and Latency Benefit Evaluation Based Task Scheduling Optimization Strategy in Apache Spark. *Concurrency Comput. Pract. Exp.* **2025**, *37*, e70244.

APPENDIX - FULL SEARCH STRINGS & REPLICABILITY PACK

RQ1 – Predictive Workload Analysis / Forecasting

IEEE Xplore (Advanced)

(("workload forecasting" OR "demand forecasting" OR autoscaling OR "capacity planning") AND ("cloud computing" OR "multi-cloud" OR "edge-cloud" OR "fog computing") AND ("machine learning" OR "deep learning" OR LSTM OR Transformer OR DeepAR OR "time series")) AND (Publication Year:2016-2025) AND (Content Type: Journals OR Conferences)

Scopus (TITLE-ABS-KEY)

TITLE-ABS-KEY(("workload forecasting" OR "demand forecasting" OR autoscaling OR "capacity planning") AND ("cloud computing" OR "multi-cloud" OR "edge-cloud") AND ("machine learning" OR "deep learning" OR LSTM OR Transformer OR DeepAR OR "time series")) AND PUBYEAR > 2015 AND PUBYEAR < 2026 AND (LIMIT-TO(LANGUAGE, "English"))

ACM Digital Library

Abstract: ("workload forecasting" OR "demand forecasting" OR autoscaling) AND ("cloud computing" OR "multi-cloud" OR "edge-cloud") AND ("machine learning" OR LSTM OR Transformer OR DeepAR) AND Publication Date:[2016 TO 2025]

RQ2 – Dynamic Resource Allocation & Scheduling (RL & Meta-heuristics)

IEEE Xplore (Advanced)

(("resource allocation" OR scheduling OR placement) AND ("cloud computing" OR "multi-cloud") AND ("reinforcement learning" OR RL OR DQN OR PPO OR "metaheuristic" OR "evolutionary algorithm" OR GA OR PSO OR ACO OR "multi-objective optimization")) AND (Publication Year:2016-2025) AND (Content Type: Journals OR Conferences)

Scopus (TITLE-ABS-KEY)

TITLE-ABS-KEY(("resource allocation" OR scheduling OR placement) AND ("cloud computing" OR "multi-cloud") AND ("reinforcement learning" OR RL OR DQN OR PPO OR metaheuristic OR "evolutionary algorithm" OR "multi-objective optimization")) AND PUBYEAR > 2015 AND PUBYEAR < 2026 AND (LIMIT-TO(LANGUAGE, "English"))

ACM Digital Library

Abstract: ("resource allocation" OR scheduling) AND ("cloud computing" OR "multi-cloud") AND ("reinforcement learning" OR RL OR metaheuristic OR "multi-objective optimization") AND Publication Date:[2016 TO 2025]

RQ3 – Cost- and Energy-Aware Optimization

IEEE Xplore (Advanced)

(("cost-aware" OR "energy-aware" OR "energy efficient" OR OPEX OR DVFS OR consolidation) AND ("cloud computing" OR "multi-cloud") AND (optimization OR scheduling OR allocation)) AND (Publication Year:2016-2025) AND (Content Type: Journals OR Conferences)

Scopus (TITLE-ABS-KEY)

TITLE-ABS-KEY(("cost-aware" OR "energy-aware" OR "energy efficiency" OR OPEX OR DVFS OR consolidation) AND ("cloud computing" OR "multi-cloud") AND (optimization OR scheduling)) AND PUBYEAR > 2015 AND PUBYEAR < 2026 AND (LIMIT-TO(LANGUAGE, "English"))

ACM Digital Library

Abstract: ("cost-aware" OR "energy-aware" OR DVFS OR consolidation) AND ("cloud computing" OR "multi-cloud") AND optimization AND Publication Date:[2016 TO 2025]

RQ4 – AI-Enhanced Load Balancing, Reliability & Security

IEEE Xplore (Advanced)

(("load balancing" OR "traffic engineering" OR routing) AND ("cloud computing" OR "multi-cloud" OR "service mesh") AND ("machine learning" OR "reinforcement learning" OR optimization OR "reliability-aware" OR "security-aware" OR "trust-aware")) AND (Publication Year:2016-2025) AND (Content Type: Journals OR Conferences)

Scopus (TITLE-ABS-KEY)

TITLE-ABS-KEY(("load balancing" OR "traffic engineering" OR routing) AND ("cloud computing" OR "multi-cloud" OR "service mesh") AND ("machine learning" OR "reinforcement learning" OR optimization OR "reliability-aware" OR "security-aware" OR "trust-aware")) AND PUBYEAR > 2015 AND PUBYEAR < 2026 AND (LIMIT-TO(LANGUAGE, "English"))

ACM Digital Library

Abstract: ("load balancing" OR "traffic engineering") AND ("cloud computing" OR "multi-cloud" OR "service mesh") AND ("machine learning" OR "reinforcement learning" OR "reliability-aware" OR "security-aware") AND Publication Date:[2016 TO 2025]