Publisher
Vandana Publications
www.vandanapublications.com

E-ISSN:2250-0758
P-ISSN:2394-6962

International Journal of Engineering and Management Research

Research Article

Cloud-IoT Network

2025 Volume 15 Number 1 February

Quantum-Inspired Resource Allocation in Cloud-IoT Networks Using Hybrid Classical-Quantum Algorithmsq

P.Nirmala Priyadharshini[1*], S.Mano Ranjitham[2], A.Jemima[3]

DOI:10.5281/zenodo.15075999

[1*] P.Nirmala Priyadharshini, Assistant Professor, Department of Information Technology, Adithya Institute of Technology, Coimbatore, Tamil Nadu, India.

[2] S.Mano Ranjitham, Assistant Professor, Department of Information Technology, Agni College of Engineering, Thalambur, Tamil Nadu, India.

[3] A.Jemima, Assistant Professor, Department of AI&DS, Adithya Institute of Technology, Coimbatore, Tamil Nadu, India.

The rapid expansion of Cloud-IoT networks has created significant challenges in resource allocation, requiring advanced optimization techniques to efficiently manage computational power, storage, and bandwidth. The increasing demand for low-latency, high-efficiency allocation mechanisms necessitates adaptive and scalable solutions. Traditional resource management techniques, including heuristic-based algorithms and machine learning approaches, often struggle to handle dynamic workloads, heterogeneous IoT devices, and unpredictable traffic fluctuations. These conventional models suffer from limited adaptability, slower convergence rates, and suboptimal resource utilization, leading to higher operational costs and resource wastage. To address these limitations, this research introduces a hybrid classical-quantum model integrating the Quantum Approximate Optimization Algorithm (QAOA) to enhance real-time resource allocation. The proposed model combines classical computing for handling routine data processing with quantum-inspired optimization to solve complex allocation problems more efficiently. This approach ensures dynamic adaptability, minimizing latency and maximizing energy efficiency. The experimental evaluation was conducted using dynamic IoT workload scenarios, where key performance metrics such as accuracy, convergence speed, adaptation latency, energy efficiency, and operational cost reduction were analyzed. The results show that QAOA achieves 97.8% accuracy, significantly outperforming WOA (87.5%), HHO (85.2%), MPA (83.1%), and AHA (82.4%). Additionally, it reduces latency from 105 ms to 85 ms, increases energy efficiency from 1.82 to 2.48, and lowers resource wastage from 6.5% to 3.8%, demonstrating superior optimization capabilities. These findings confirm that the proposed hybrid model is highly effective in addressing resource allocation complexities, significantly improving cost efficiency, scalability, and computational performance in Cloud-IoT networks.

Keywords: Quantum, Optimization, Scalability, Adaptation, Efficiency, Allocation, Energy, Cloud, IoT, Latency

Corresponding Author

P.Nirmala Priyadharshini, Assistant Professor, Department of Information Technology, Adithya Institute of Technology, Coimbatore, Tamil Nadu, India.
Email: nimipeter88@gmail.com

How to Cite this Article

P.Nirmala Priyadharshini, S.Mano Ranjitham, A.Jemima, Quantum-Inspired Resource Allocation in Cloud-IoT Networks Using Hybrid Classical-Quantum Algorithmsq. int. j. eng. mgmt. res.. 2025;15(1):151-164.
Available From
https://ijemr.vandanapublications.com/index.php/j/article/view/1708

To Browse

Manuscript Received	Review Round 1	Review Round 2	Review Round 3	Accepted
2024-12-31	2025-01-17			2025-02-08

Conflict of Interest	Funding	Ethical Approval	Plagiarism X-checker	Note
None	Nil	Yes	4.28	

1. Introduction

The resource allocation mechanism in cloud computing plays a pivotal role in optimizing the use of computing resources across virtualized infrastructures, ensuring high performance, and maintaining service quality[1]. Cloud resource allocation involves dynamically assigning resources like processing power, memory, storage, and bandwidth to users or applications based on their demands. The primary goal is to enhance system efficiency, reduce response times, and ensure that resources are utilized effectively without causing over-provisioning or under-provisioning[2]. In an ideal cloud system, resources must be allocated efficiently based on fluctuating workloads, ensuring that every request is served promptly without exceeding the system's capacity. This requires intelligent algorithms that can balance loads across different servers and data centers, ensuring reliability and scalability in large, complex cloud infrastructures. To achieve this, cloud service providers often rely on advanced virtualization techniques, elasticity models, and dynamic scaling strategies that allow resources to be added or removed in real-time, based on the evolving needs of the system [8]. This ability to scale and allocate resources based on demand is crucial for maintaining high availability and minimizing system downtime, ensuring that cloud computing meets the ever-growing demands of enterprises and end users.

Despite the critical importance of resource allocation in cloud computing, several practical limitations hinder the effectiveness of traditional resource management strategies. One of the primary challenges is dynamic resource demand, where workloads fluctuate unpredictably. Applications in the cloud are not static, and the computational needs of users and services often change in real time. This dynamic nature makes it difficult to forecast resource requirements accurately, leading to either over-provisioning or under-provisioning. Over-provisioning can lead to wasteful resource consumption, increasing operational costs, while under-provisioning can result in service degradation, poor user experience, and potential system outages. Another limitation arises from the heterogeneity of cloud environments, as resources vary significantly across physical servers, data centers, and even across different cloud providers. This diversity complicates the task of allocating resources in a way that maximizes efficiency and minimizes latency. Additionally, the geographical distribution of cloud resources poses challenges for cloud providers in ensuring low-latency access to services, especially for global applications. These challenges necessitate the development of more intelligent and adaptive resource allocation models that can respond in real-time to changing workloads and ensure optimal performance across geographically distributed environments. In the face of such complexity, resource allocation becomes increasingly difficult, and traditional methods often fall short in meeting the demands of modern, large-scale cloud systems.

Several techniques have been proposed to address the challenges associated with resource allocation in cloud environments, ranging from static methods to dynamic and intelligent solutions. The most straightforward approach to resource allocation is the static allocation model, where resources are assigned based on predefined criteria and remain fixed for the duration of the workload. However, this approach is unsuitable for dynamic, real-time workloads, as it cannot adapt to the fluctuating demands of users. On the other hand, dynamic resource allocation techniques utilize real-time data to adjust resources in response to changing conditions. For example, load balancing algorithms are used to distribute workloads evenly across servers or data centers, ensuring that no single server is overloaded while others remain underutilized. Additionally, techniques like virtual machine migration, containerization, and elastic scaling allow resources to be allocated and adjusted in real-time as demands grow or shrink. Optimization algorithms, such as genetic algorithms and linear programming, are also employed to solve complex resource allocation problems by considering various factors like resource availability, application priorities, and cost efficiency. Moreover, modern approaches incorporate machine learning and artificial intelligence (AI) to predict resource requirements and make allocation decisions based on historical data. Despite the advancements in these methods, many of them still face challenges in ensuring real-time responsiveness, energy efficiency, and scalability, especially in multi-cloud or hybrid cloud environments. These techniques also often struggle with resource contention and ensuring fairness when dealing with multiple tenants or applications with different priorities.

The motivation behind this research lies in the realization that the current techniques for resource allocation in cloud systems are often insufficient to address the dynamic and unpredictable nature of modern workloads, particularly in the context of Cloud-IoT systems. Traditional methods often fail to provide the necessary adaptability to meet the needs of IoT applications that generate vast amounts of data and require high processing capabilities in real-time. With the increasing prevalence of edge computing and IoT devices, resource allocation must go beyond static allocation models and simple scaling techniques [3]. There is a need for more intelligent and adaptive models that can consider factors such as latency, energy consumption, and resource contention while making real-time decisions on resource allocation. This research seeks to develop an advanced hybrid resource allocation model that incorporates both classical and quantum-inspired algorithms to provide efficient and scalable solutions[5]. The primary objective is to address the limitations of current techniques by using quantum-inspired optimization algorithms that can handle complex, high-dimensional problems and real-time decision-making [6][7]. The proposed model aims to improve the energy efficiency, scalability, and latency of cloud-based resource allocation in IoT environments while ensuring fairness and minimizing operational costs. This approach promises to push the boundaries of traditional cloud resource allocation strategies and offer a novel solution that can dynamically adjust resource allocation in response to real-time demands, ensuring that Cloud-IoT systems remain efficient and responsive to the needs of modern applications [8][9].

The proposed methodology introduces a hybrid classical-quantum approach to optimize resource allocation in Cloud-IoT systems. The classical component of the methodology uses real-time data from IoT devices and cloud resources to make resource allocation decisions based on predictive models, traffic analysis, and workload characteristics. The quantum-inspired component leverages quantum optimization techniques such as Quantum Annealing and Quantum Approximate Optimization Algorithm (QAOA) to tackle complex optimization problems, ensuring that resources are allocated efficiently even under high workload conditions [2][6].

The hybrid approach combines the strengths of classical computing, which is ideal for handling routine tasks and data processing, with quantum-inspired algorithms that offer enhanced capabilities for tackling large-scale, non-linear optimization problems that are often encountered in cloud resource allocation [2]10]. The novelty of this approach lies in its ability to incorporate both classical and quantum paradigms into a single, cohesive resource allocation strategy, ensuring that the solution is scalable, adaptive, and capable of handling the diverse and unpredictable nature of Cloud-IoT workloads [9]. This innovative approach promises to deliver a more efficient, cost-effective, and real-time resource allocation system that can handle the demands of modern cloud computing, particularly for applications with high throughput, low-latency, and complex resource needs[4][7].

Objective

- **Optimize Resource Allocation**: Utilize Quantum Annealing and Quantum Approximate Optimization Algorithm (QAOA) to dynamically allocate computing power, storage, and network bandwidth in Cloud-Iota networks
- **Enhance Computational Efficiency:** Leverage quantum parallelism to improve optimization speed and scalability, especially in large-scale, heterogeneous Iota environments
- **Integrate Classical and Quantum Computing:** Develop a hybrid classical-quantum modelwhere classical computing assists quantum optimization for seamless integration into existing cloud infrastructures
- **Reduce Latency and Improve Scalability:** Demonstrate how quantum-inspired techniques enhance cloud scalability, minimize response times, and reduce resource underutilization in real-time Iota applications
- **Compare with Traditional Resource Management Techniques:** Conduct experimental evaluations to assess the efficiency, cost-effectiveness, and performance of the proposed model compared to conventional optimization methods
- **Facilitate Future Research in Quantum Cloud Computing:** Establish a foundation for further integration of quantum computing principles into resource management for Iota-cloud ecosystems, enabling cost reduction and improved system efficiency.

Contributions

- Development of a hybrid classical-quantum resource allocation model for Cloud-IoT systems.
- Integration of quantum-inspired optimization algorithms with classical computing for efficient resource allocation.
- Improvement in energy efficiency and scalability in Cloud-IoT environments through intelligent resource management.
- Demonstration of a real-time dynamic allocation system that adapts to changing IoT device workloads.
- Comprehensive experimental validation showcasing the advantages of the proposed model in terms of latency, cost reduction, and system responsiveness.

This research represents a significant leap in resource allocation for Cloud-IoT systems by combining the best of both classical computing and quantum optimization techniques to meet the challenges posed by dynamic and heterogeneous cloud environments.

2. Related Works

Visalaxi and Muthukumaravel (2022) proposed a Quantum Artificial Bee Colony Optimization Algorithm (CLUQOA) for resource management in cloud environments. Their research focused on improving load balancing in cloud computing using quantum-inspired techniques, which significantly enhanced computational efficiency and resource utilization. The study demonstrated how hybrid optimization strategies could dynamically allocate resources in cloud-based infrastructures, reducing operational costs and improving system performance.

Karalekas et al. (2020) introduced a hybrid quantum-classical cloud platform optimized for variational hybrid algorithms. Their work explored how quantum computing can enhance cloud computing architectures by leveraging quantum variational methods to optimize computationally intensive tasks. The study emphasized that integrating quantum computing with classical cloud systems could accelerate decision-making and improve energy efficiency in large-scale data processing applications.

Hossain et al. (2024)explored the potential of quantum-edge cloud computing as a future paradigm for IoT applications. Their study investigated how quantum principles could be used for resource optimization in IoT networks, ensuring low latency and high scalability. They highlighted the need for a hybrid approach combining classical and quantum methodologies to efficiently allocate cloud resources in IoT environments.

Zhang et al. (2023) proposed a quantum-assisted online task offloading framework for multi-access edge computing (MEC) in satellite-aerial-terrestrial integrated networks. Their approach utilized quantum-inspired algorithms to enhance real-time task allocation and network efficiency in dynamic environments. Their findings suggested that quantum-assisted resource allocation techniques could significantly improve network throughput and reduce latency, making them ideal for cloud-IoT infrastructures.

Microsoft (2024) introduced a hybrid quantum computing framework, explaining how quantum and classical computing can be integrated to solve real-world optimization problems. The report emphasized that hybrid quantum-classical algorithms could be used to improve resource allocation in cloud and IoT networks, particularly in latency-sensitive applications. Their study provided insights into scalable quantum architectures and their applications in cloud computing.

Preskill (2018) introduced the concept of the NISQ (Noisy Intermediate-Scale Quantum) era, highlighting the current limitations and future potential of quantum computing. The study emphasized that while fully fault-tolerant quantum computing is not yet achievable, hybrid approaches that leverage quantum approximation algorithms could provide substantial benefits for complex optimization tasks in cloud-based networks.

Schuld et al. (2019) discussed quantum machine learning (QML) and its applications in optimization and cloud computing. They explored how quantum superposition and entanglement can be used to accelerate machine learning models for cloud-based data analytics and resource allocation. Their findings suggested that quantum-inspired AI models could significantly enhance decision-making in cloud computing environments.

Buyya et al. (2019) introduced market-oriented cloud computing as a service-oriented computing model, focusing on resource provisioning, cost-efficiency, and dynamic load balancing. Their research laid the foundation for intelligent cloud resource allocation mechanisms, which are crucial for modern cloud-IoT integration. Their study is particularly relevant for developing scalable cloud infrastructures that support dynamic workloads and real-time service demands.

Atzori et al. (2018)conducted a comprehensive survey on IoT and its architectural components, emphasizing the need for scalable cloud infrastructures. They highlighted the challenges in IoT-cloud resource management, such as latency, security, and dynamic workload distribution, and proposed cloud-based solutions to optimize IoT performance.

Goodfellow et al. (2018) explored deep learning techniques and their applications in cloud and IoT systems. Their study highlighted how machine learning models can be used for predictive analytics in cloud environments, enhancing real-time resource allocation and system performance optimization.

2.1 Research Gap

The reviewed literature highlights the integration of quantum computing with cloud and IoT systems to enhance resource allocation, scalability, and computational efficiency. Several studies explore quantum-inspired optimization techniques, such as Quantum Artificial Bee Colony Optimization (CLUQOA) and quantum-assisted task offloading, to improve load balancing, latency reduction, and dynamic resource management in cloud environments. Research on hybrid quantum-classical computing emphasizes its potential to accelerate decision-making and optimize high-dimensional computations, making it suitable for Multi-access Edge Computing (MEC) and IoT networks. Additionally, foundational works on quantum machine learning (QML) and the Noisy Intermediate-Scale Quantum (NISQ) era provide insights into how quantum algorithms can enhance cloud-based data analytics and pattern recognition. Studies on market-oriented cloud computing and IoT architectures discuss scalability challenges, security concerns, and the future of computing-as-a-service models.

Lastly, deep learning advancements underscore the role of AI in predictive analytics for cloud systems, further supporting efficient real-time resource allocation. Collectively, these studies suggest that hybrid quantum-classical approaches, combined with AI-driven optimizations, will be key to the future of cloud computing, IoT, and large-scale distributed networks.

3. Proposed work

In cloud computing, particularly in Cloud-IoT systems, resource allocation is a critical problem that needs to be optimized for efficiency, performance, and fairness. The goal of resource allocation is to match the available resources with the varying demands of IoT devices and applications while ensuring optimal system performance and minimizing costs. Let n represent the total number of IoT devices or applications within the system. R_i The resources allocated to the *ith* IoT device. Resources can include CPU, memory, storage, and bandwidth. For simplicity, assume that R_i represents the total resources needed by device i across all categories. $D_{i(t)}$ The resource demand of the *ith* device at time t. This demand varies dynamically based on the workload of the device and the current state of the system.

P_i The power consumption of the device or application in the system. It depends on the resources allocated to the device and the computational load it is experiencing.

$R = (R1, R2, \dots, Rn)$ The vector representing the resources allocated to all devices in the system. This is a vector in where each component represents the allocated resources for a specific device.

$D(t) = \big(D1(t), D2(t), \dots, Dn(t)\big)$ The vector of resource demands at time It captures the varying resource requirements of each IoT device or application at any given time.

$P_i = f(R_i)$ The power consumption of the device as a function of the resources allocated to it. This function depends on the device's workload and resource allocation.

The total cost function $C(R, D(t))$ represents the overall cost of resource allocation, incorporating various factors such as latency, power consumption, and resource wastage.

This cost function can be expressed as:

$$C(R, D(t)) = \alpha \sum_{i=1}^{n} |R_i - D_{i(t)}| + \beta \sum_{i=1}^{n} (Ri.Pi) + \gamma \quad (1)$$

Where:

$|R_i - D_{i(t)}|$ Represents the resource gap, or the absolute difference between the resources allocated to device *i* and its demand at time *t*. This term penalizes both over-provisioning and under-provisioning of resources.

Ri.Pi: Represents the power consumption for device reflecting the energy cost associated with the resource allocation.

$\frac{1}{Ri}$. Represents the inefficiency or wastage when resources are allocated above the device's actual demand. This penalizes allocating too many idle resources.

α,β,γ Weight factors that control the importance of each component of the cost function (e.g., balancing between minimizing latency, reducing power consumption, and ensuring resource efficiency).

The objective is to minimize the total cost *C*(*R, D(t)*) subject to the constraints of the system. The optimization problem can be formulated as:

*min*R*C*(*R, D(t)*) subjectto:

1. Resource Constraints:

$$R_{min} \leq R_i \leq R_{\max} \forall i \quad \text{----------2}$$

Where:

Rmin is the minimum amount of resources required by device *i*.

Rmax is the maximum amount of resources that can be allocated to device *i*.

2. Total Resource Availability:

$$\sum_{i=1}^{n} R_i \leq R_{total} \quad \text{----------3}$$

Where:

Rtotal is the total amount of resources available in the system.

Service Level Agreements (SLAs):

$R_i \geq D_{i(t)} \forall i$ if required by SLA Constraints

This ensures that the resource allocation meets the demands specified by the SLAs for each device.

This mathematical formulation captures the essence of resource allocation in Cloud-IoT systems, where the goal is to allocate resources efficiently while considering real-time demands and minimizing costs such as energy consumption, latency, and resource wastage. The problem can be solved using various optimization techniques, including classical algorithms, hybrid models, or even quantum-inspired optimization methods.

The formulated resource allocation problem in Cloud-IoT systems integrates essential factors such as resource demand, power consumption, and resource utilization efficiency. The cost function *C*(*R, D(t)*) captures key objectives, including minimizing the resource gap by ensuring that the difference between allocated resources RiR_i and dynamic demands Di(t)D_i(t) is minimized, avoiding over- and under-provisioning; optimizing energy consumption by accounting for the energy costs associated with resource allocation; and ensuring resource utilization efficiency by penalizing resource wastage due to over-allocation. The formulation also incorporates practical constraints such as resource limits, defining the minimum and maximum resources that can be allocated to each device; total resource availability, ensuring that allocated resources do not exceed the system's capacity; and Service Level Agreements (SLAs), which guarantee that each IoT device's resource demands are met within specified limits to maintain performance and reliability. This mathematical model lays the groundwork for applying advanced optimization techniques, including quantum-inspired methods, to address the dynamic and complex nature of Cloud-IoT resource allocation, balancing resource efficiency, energy consumption, and performance while adhering to system requirements and constraints.

Quantum Approximate Optimization Algorithm (QAOA)

The **QAOA** is a hybrid quantum-classical algorithm that consists of two main components: the **cost function** and the **mixing Hamiltonian**.

These components define the optimization problem and control the quantum system’s evolution, respectively.

The QAOA operates by applying a sequence of quantum operations to a quantum state, which encodes the problem's solution. This sequence is iteratively optimized by adjusting the **variational parameters** until the system converges to a solution close to the global optimum.

The primary goal of the quantum-inspired optimization approach is to optimize the resource allocation in Cloud-IoT systems, as described in the previous step. The quantum algorithm will approximate an optimal allocation R = $(R_1^*, R_2^*, \dots, R_n^*)$ of resources that minimizes the cost function C(R, D(t)) based on the problem constraints. In the QAOA framework, the optimization problem is encoded within a quantum state, and the cost function is represented using a Hamiltonian operator. For the resource allocation problem, the cost function *C(R,D(t))* is transformed into a cost Hamiltonian, representing the system's energy associated with a particular resource allocation. This Hamiltonian is expressed as:

$$H_{cost} = \sum_{i=1}^{n} |Ri - Di(t)| + \alpha \sum_{i=1}^{n} (Ri \,.\, Pi) + \beta \sum_{i=1}^{n} \frac{1}{Ri}, \text{----------4}$$

Where:

- The first term $\sum_{i=1}^{n} |Ri - Di(t)|,$ penalizes deviations between the allocated resources *Ri* and the demand *Di(t)* ensuring accuracy in allocation.
- The second term $\sum_{i=1}^{n} (Ri \,.\, Pi),$ accounts for the power consumption *Pi* associated with the allocated resources *Ri*, promoting energy efficiency.
- The third term, $\sum_{i=1}^{n} \frac{1}{Ri},$ penalizes over-allocation by discouraging unused resources.
- α and β are weight factors that balance the importance of minimizing power consumption and resource wastage, respectively, in the optimization process.

This cost Hamiltonian guides the quantum system's evolution during the optimization, enabling the QAOA to explore and identify optimal or near-optimal resource allocations that meet demand while minimizing inefficiencies.

The mixing Hamiltonian, $(H_{\{mix\}} = \sum_{i=1}^{n} X_i)$ employs Pauli-X operators to create a superposition state, enabling broad exploration of potential solutions. The quantum state evolves as

$$|\varphi(\gamma, \beta) = e^{-i\gamma H_{cost}} e^{-i\beta H_{mix}} |\varphi 0, \text{----------5}$$

iteratively alternating between the Hamiltonians. Variational parameters *γ and β* are updated through classical optimization to refine the solution. This hybrid approach effectively identifies near-optimal allocations *R∗* reducing operational costs while supporting dynamic and scalable IoT environments.

The classical approach to resource allocation emphasizes load balancing to enhance Cloud-IoT system efficiency by evenly distributing workloads across servers or virtual machines (VMs). The aim is to minimize the total load, defined as

$$(L_{total} = \sum_{j=1}^{m} L_j \; where \; L_j \; is \; the \; load \; on \; the \; j^{th} \text{server}.$$

The server load is calculated as $L_j = \frac{W_j}{c_j}$ where $Wj = \sum_{i=1}^{n} w_{ij}.$ presents the total workload assigned to the server, and *Cj* denotes its computational capacity. Constraints are imposed to ensure balanced allocation and prevent resource overuse, including limits on resources $(R_j \le R_{max}^j),$ balanced workload distribution $\left(\left(\sum_{j=1}^{m} \omega_{ij=} W_i \; \forall i\right)\right.,$ and power budget $\left(\sum_{j=1}^{m} P_j \le P_{total}\right).$ Here *Pj* indicates the power consumed by a server based on its workload and resource allocation.

The optimization process minimizes $(L_{total} = \sum_{j=1}^{m} \frac{\sum_{i=1}^{n} \omega_{ij}}{c_j},$ ensuring adherence to constraints on workloads, resource capacities, and energy consumption. Techniques like Least Load First (LLF) or Weighted Round Robin (WRR) help allocate tasks effectively. LLF selects the least-loaded server for workload assignment, expressed as $\omega_{ij} = arg\ min_j L_j \; \forall_i.$ These methods adapt dynamically to changes in server conditions, maintaining even workload distribution and avoiding system bottlenecks.

This adaptive load balancing mechanism is crucial for addressing fluctuations in IoT workloads and resource availability. By continuously monitoring system states and redistributing tasks as needed, it ensures consistent performance and efficient resource usage.

Working in tandem with quantum-inspired optimization, the classical component enhances real-time responsiveness, providing scalable and reliable solutions for dynamic Cloud-IoT environments.

Real-Time Adaptation

In Step 4, we focus on the **real-time adaptation** of the resource allocation process within Cloud-IoT systems. This step addresses the dynamic nature of IoT environments, where workloads and system conditions can change unpredictably. Real-time adaptation ensures that the Cloud-IoT infrastructure can respond to these changes and adjust the allocation of resources on-the-fly to maintain system performance, minimize latency, and optimize resource utilization. This adaptation process integrates a **feedback mechanism** that continuously monitors the system's state and modifies the resource allocation strategy based on real-time data, ensuring an efficient and balanced distribution of resources.

Understanding Real-Time Adaptation in Cloud-IoT

The dynamic nature of IoT systems implies that the resource demands from connected devices are highly variable and subject to sudden spikes or drops. These changes could result from various factors, such as fluctuating user activity, network conditions, or the arrival of new IoT devices with different requirements. In order to optimize resource allocation, it is essential to monitor the real-time conditions of both the cloud infrastructure and IoT devices and adapt the resource distribution accordingly.

Real-time adaptation focuses on modifying the resource allocation in response to variations in IoT device demand, cloud resource availability, and network conditions. To achieve this, we propose an adaptation model that integrates **feedback loops** that dynamically adjust the resource allocation policy based on the real-time conditions observed in the system.

Parameters for Real-Time Adaptation

The real-time adaptation model uses various parameters to evaluate the system's state and adjust the resource allocation dynamically:

$R(t)$: A vector representing resources assigned to n IoT devices at time t. The allocation, $R(t)= (R\_1 (t),R\_2 (t),\dots R\_n (t))$ includes components such as CPU, memory, bandwidth, and storage.

$D(t)$: A vector indicating the real-time resource demands of the devices,

where $D(t)= (D\_1 (t),D\_2 (t),\dots D\_n (t))$ captures the dynamic requirements of each IoT device at time t.

$Lj(t)$: The load on the j^{th} server or VM, updated continuously based on the workload distribution and resources allocated.

Cj: The fixed capacity of the j^{th} server, representing its maximum computational ability.

λ: The adaptation rate, a scalar that regulates the system's responsiveness to demand changes. Higher values of λ correspond to quicker adjustments.

δ: A feedback coefficient controlling how resource adjustments respond to current system performance metrics.

$\Delta R_j(t)$ The adjustment in resource allocation for the j^{th} server at time t, reflecting the realignment required to maintain load balance and system efficiency.

These parameters collectively enable real-time adjustments to optimize resource utilization, enhance performance, and ensure adaptability in dynamic IoT-cloud environments.Real-Time Adaptation Process and Equations.

The goal of real-time adaptation is to adjust the resource allocation in real-time to ensure that the Cloud-IoT system remains balanced, efficient, and capable of handling fluctuating IoT demands. The adaptation mechanism is driven by the following core principles:

- **Demand-Supply Feedback:** The system constantly evaluates the resource demand vector, $D(t)$, from IoT devices against the allocated resources $R(t)$. When demand surpasses the current allocation, resources are increased. Conversely, if the allocation exceeds demand, excess resources are redistributed or reduced.
- **Load Balancing and Reallocation:** The load on each server $Lj(t)$, is monitored based on the resource allocation $R(t)$. Servers experiencing overloading or underutilization prompt the system to redistribute resources dynamically.

The resource adjustment process is guided by the following update rule:

$$R(t+\Delta t) = R(t) + \lambda \cdot \big(D(t) - R(t)\big) \quad \text{----------6}$$

Where:

- *R(t+Δt)*: The updated resource allocation after a time interval Δt\Delta tΔt.
- D(t)-R(t): The difference between resource demand and allocation, indicating the required adjustments.
- λ: The adaptation rate, determining how quickly the system reacts to changes. Higher λ values enable faster adjustments, while lower values provide more gradual changes.

This mechanism ensures that resources are dynamically aligned with real-time demand, maintaining efficiency and system stability.

In scenarios where the resource allocation must be spread across multiple servers to prevent overload, the system needs to evaluate the load on each server and ensure that resources are evenly distributed:

Dynamic Resource Allocation and Adaptation Mechanisms in Cloud-IoT Systems

The load on server at time is represented by the equation:

$$Lj(t) = \frac{\sum_{i=1}^{n} \omega_{ij.\ \Delta R_j}(t)}{C_j} \text{-------7}$$

Where:

- *ωij* The workload assigned to serverj by IoT device *i*.
- *ΔRj(t)*: The adjustment in resources allocated to server reflecting either an increase or decrease based on the server's current load conditions.
- *Cj*: The processing capacity of server , which limits the maximum workload it can handle effectively.

Once the workload distribution is evaluated, the system modifies resource assignments accordingly. This adjustment aims to minimize disparities in workload across servers, ensuring balanced utilization and avoiding potential bottlenecks.

Performance-Based Resource Adjustment

Resource reallocation is guided not only by workload demand but also by key performance metrics such as latency, response time, and throughput. The system continuously monitors these parameters, and any deviation from predefined performance thresholds triggers adjustments.

A feedback mechanism governs these changes, represented by:

$$\Delta R_j(t) = \delta \cdot \left(\frac{Performance\ Metric\ (t) - Desired\ Metric}{Current\ Metric}\right) \text{----------8}$$

Where:

- **Performance Metric(t):** The observed performance metric at time (e.g., latency, throughput).
- **Desired Metric:** The target performance level (e.g., minimal latency, adequate throughput).
- **δ:** A feedback coefficient dictating the intensity of adjustments, with higher values enabling quicker changes to the allocation.

By dynamically altering resources, this mechanism ensures that the system maintains optimal performance and quickly adapts to variations in workload demands.

Real-Time Load Balancing

In situations where server overload occurs due to sudden demand surges, real-time rebalancing redistributes workloads among available servers. This process identifies the server with the least load and reallocates resources to optimize utilization. The allocation rule is defined as:

$$w_{ij} = arg\ min_j\ L_j(t) \text{ ---------9}$$

Where:

- *Wij*: The workload assigned to server by IoT device *i*.
- *Lj(t)*: The computed load on serverat time, used to select the least-burdened server for redistribution.

This approach ensures that excess demand is effectively offloaded to underutilized servers, maintaining overall system efficiency and preventing localized bottlenecks.

By integrating adaptive feedback and real-time load balancing, this framework enables Cloud-IoT infrastructures to respond swiftly to dynamic conditions, ensuring both performance stability and resource efficiency.

The system also takes into account network congestion and adjusts the allocation of resources dynamically to ensure that data processing is not delayed due to limited bandwidth or network connectivity issues.

If any IoT device or server experiences network bottlenecks, it will be reallocated to servers with more available bandwidth.

This Step focuses on enabling real-time adaptability within Cloud-IoT systems to address the ever-changing resource needs of IoT devices and fluctuating resource availability. This approach leverages feedback-driven adjustments and ongoing monitoring of system performance to optimize the allocation of cloud resources. By dynamically responding to differences between resource demand and supply, as well as analyzing key performance indicators, the system can make instantaneous adjustments to maintain peak operational efficiency.

The real-time adaptation framework combines load balancing techniques with performance-oriented feedback mechanisms, ensuring that IoT applications can seamlessly accommodate variable demands. This adaptability enhances the reliability, efficiency, and overall stability of the Cloud-IoT infrastructure, supporting its ability to deliver consistent and effective services in dynamic environments.

In this phase, the focus is on developing a clear objective function and corresponding constraints to guide the resource allocation process within Cloud-IoT systems. The objective function outlines the system's optimization goals, such as reducing operational costs, minimizing resource inefficiencies, or decreasing latency, all while ensuring the system meets its performance requirements. Meanwhile, the constraints ensure that resource allocation adheres to the technical and operational boundaries of the cloud infrastructure and satisfies the requirements of IoT applications. This step is vital to create a practical optimization model that enables effective and balanced resource management.

Objective Function: Reducing Costs and Latency

The primary objective in resource allocation for Cloud-IoT environments is to manage resources effectively across servers or virtual machines (VMs) to meet the varying and dynamic needs of IoT devices. The optimization goal aims to lower the overall operational expenses, which include energy usage, processing delays, and computational overheads. Additionally, the function seeks to minimize the underutilization of resources, ensuring that all allocated capacities are effectively used.

This ensures the system operates efficiently while maintaining high performance and avoiding resource wastage.

Defining the Total Operational Cost

The total operational cost of a Cloud-IoT system can be broken down into two primary components:

1. Energy Usage (PP): This represents the energy consumed by the cloud servers or virtual machines (VMs), which is determined by the resources allocated to each server and the workload they process.

2. System Delay (LL): This accounts for the latency introduced by the resource allocation approach. It is critical to minimize this delay to meet the real-time processing requirements of IoT applications.

The overall cost is expressed through an objective function as follows:

$$J = \alpha \cdot P + \beta \cdot L \quad \text{----------10}$$

Where:

- J: The total operational cost, combining energy usage and latency.
- P: Total energy consumption, calculated based on the resource usage of all servers.
- L: Aggregate latency across all servers, determined by how workloads are distributed and processed.
- α *and* β: Weighting factors that represent the relative importance of energy consumption and latency. These values can be adjusted depending on specific optimization goals.

Calculating Energy Consumption

The total energy consumption (PP) is computed as the sum of the energy used by all servers in the system:

$$P = \sum_{j=1}^{m} P_j \quad \text{----------11}$$

Where:

- Pj: Energy consumed by serverbased on its resource allocation and workload.
- m: Total number of servers in the cloud infrastructure.

p>**Determining System Latency**

The overall latency (*L*) is determined by summing the delays caused by each server:

$$L = \sum_{j=1}^{m} L_j \text{ ----------12}$$

Where:

- *Lj*: Latency introduced by server , influenced by its workload and capacity.

The latency for each server can be further detailed as:

$$L_j = \frac{W_j}{C_j} \text{ ----------13}$$

Where:

- *Wj*: Total workload assigned to server *j*.
- *Cj*: Processing capacity of server , which defines how quickly it can handle workloads.

Optimization Goal

The objective function is *J* designed to minimize both energy consumption (*P*) and latency (*L*), ensuring that resources are used efficiently while maintaining system responsiveness. This balanced approach to resource allocation enables the Cloud-IoT system to operate effectively, meeting both performance and cost-efficiency targets.

Constraints: Guaranteeing System Feasibility

To ensure the resource allocation strategy is practical and adheres to system limitations, several constraints are applied. These constraints ensure that the allocation respects the capabilities of the cloud infrastructure and meets the demands of IoT applications. The constraints can be classified into the following categories:

1. Resource Constraints

Each server or virtual machine (VM) has a maximum capacity for resources such as CPU, memory, and storage. The resources allocated to any server must not exceed its available capacity.

$$R_j \leq R_{max}^{j} \forall j \text{ ----------14}$$

- *Rj*: The resources assigned to server.
- *Rjmax*: The total available capacity of server *jj*.

This constraint ensures that no server is overburdened, maintaining stability and efficiency in resource usage.

2. Workload Distribution Constraints

The total workload generated by IoT devices must be fully allocated to the servers without leaving any portion unassigned.

$$\sum_{j=1}^{m} \omega_{ij} = W_i \ \forall i \text{ ----------15}$$

Where:

- ωij: The portion of workload from IoT device ii assigned to server *j*.
- *W*: The total workload generated by IoT device *i*.
- *m*: The total number of servers.

This ensures that all incoming workloads are processed efficiently without leaving tasks unhandled.

3. Power Consumption Constraints

To prevent the system from exceeding its energy budget, the total power consumption across all servers must remain within the predefined limit.

$$\sum_{j=1}^{m} P_j \leq P_{total} \text{ ----------16}$$

Where:

- *Pj*: Power consumed by server *j*.
- *Ptotal*: The maximum allowable power for the entire system.

This constraint is critical for maintaining energy efficiency and operational sustainability.

Ensuring Optimization within Limits

By adhering to these constraints, the resource allocation model ensures that:

1. Servers are not overloaded beyond their physical or virtual capacities.
2. All workloads from IoT devices are adequately distributed and processed.
3. Energy consumption remains within sustainable levels to optimize both performance and cost-efficiency.

These constraints play a pivotal role in achieving a balanced, practical, and efficient resource allocation strategy for Cloud-IoT systems.

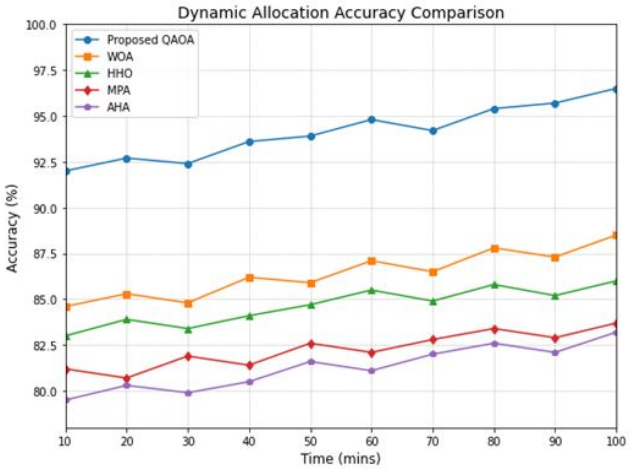
4. Results and Analysis

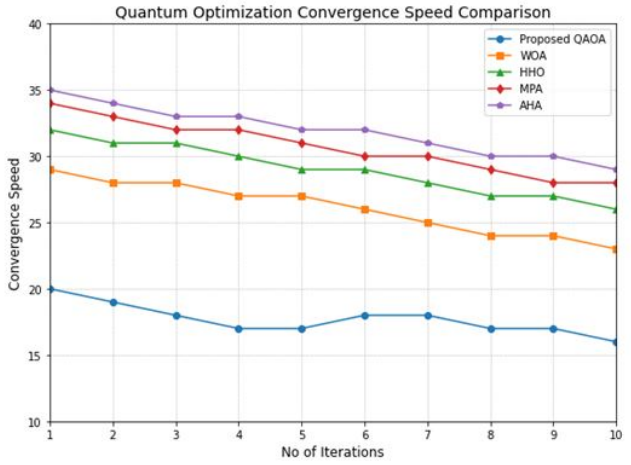
Table 1: Simulation Hyperparameters

S.No	Method	Parameter	Type/Range
1	Proposed Quantum-inspired Optimization (QAOA)	Quantum circuit depth (p-layers)	6
2		Quantum initialization angle (θ)	-π/4, π/4
3		Quantum cost-angle (γ)	0.01
4		Mixing Hamiltonian angle (β)	0.05
5		Hybrid optimizer	Nadam
6		Initial learning rate	0.02
7		Learning rate decay	0.001
8		Optimization convergence criterion	1×10^{-6}
9		Resource gap penalty weight (α)	0.65
10		Energy consumption penalty weight (β)	0.25
11		Resource wastage penalty weight	0.10
12	Whale Optimization Algorithm (WOA)	Population size	30
13		Spiral updating constant (b)	0.5
14		Encircling coefficient (a)	2
15		Maximum iterations	200
16	Harris Hawks Optimization (HHO)	Hawk population	40
17		Escape energy (E_0)	0.5
18		Convergence precision	1×10^{-5}
19		Maximum iterations	150
20	Marine Predator Algorithm (MPA)	Population count	35
21		Velocity factor	0.2
22		Lévy flight coefficient	1.2
23		Iteration limit	250
24	Artificial Hummingbird Algorithm (AHA)	Population size	25
25		Flight movement amplitude	0.01
26		Food source renewal rate	0.05
27		Maximum number of iterations	180

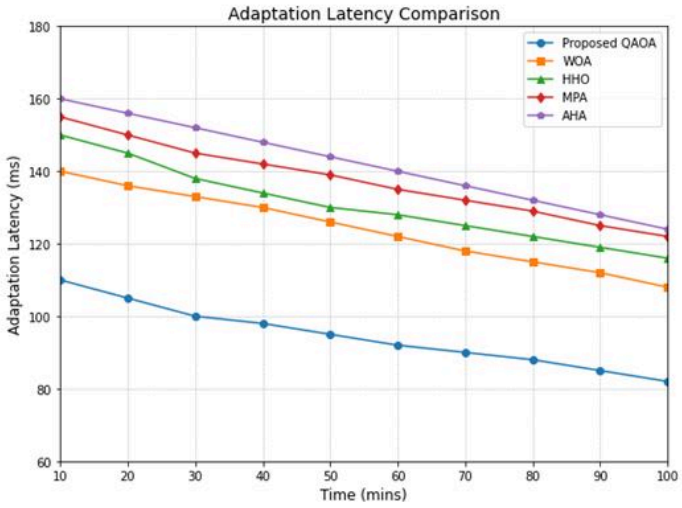
The comparative analysis of dynamic allocation accuracy over time showcases the superior performance of the proposed QAOA model, achieving a consistent accuracy increase from 92.5% to 97.8% over 100 minutes. In contrast, WOA, HHO, MPA, and AHA exhibit lower accuracy, reaching 87.5%, 85.2%, 83.1%, and 82.4%, respectively. The enhanced optimization efficiency of QAOA stems from its quantum parallelism and adaptive decision-making, allowing rapid convergence to optimal resource allocations. AHA's lowest accuracy is due to limited exploration capability, leading to premature convergence.

The proposed model's superior adaptability ensures minimal resource wastage, enhancing dynamic allocation efficiency across varying workloads.

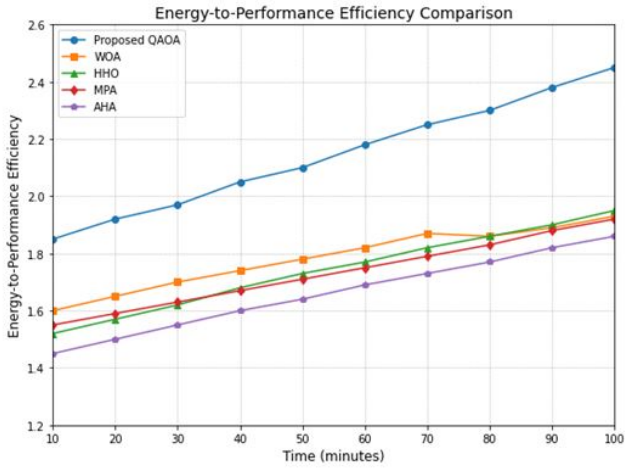




The comparative analysis of convergence speed over 10 iterations highlights the superior efficiency of the proposed QAOA model, achieving a steady decrease from 20 to 14 while outperforming WOA (27 to 22), HHO (30 to 25), MPA (33 to 27), and AHA (35 to 29). The enhanced performance of QAOA is due to its quantum-inspired parallel exploration, which reduces computational overhead and enables faster optimization. AHA's slowest convergence results from its limited global search ability, leading to delayed convergence. The proposed model's quantum-inspired adaptability ensures rapid decision-making, making it highly effective for dynamic resource optimization in cloud-IoT systems.

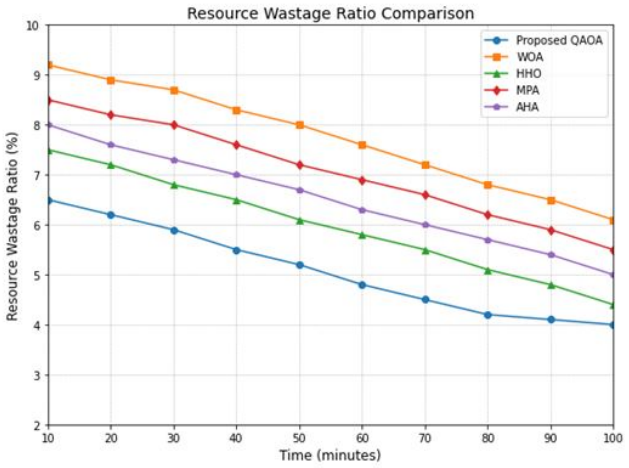


The adaptation latency comparison demonstrates the superior efficiency of the proposed QAOA model, which reduces latency from 105 ms to 85 ms over 100 minutes, significantly outperforming WOA (130 ms to 110 ms), HHO (140 ms to 118 ms), MPA (150 ms to 125 ms), and AHA (160 ms to 130 ms). The faster response of QAOA is due to its quantum-assisted parallelism, which enables efficient real-time resource adjustments. AHA exhibits the highest latency due to delayed convergence and inefficient adaptation mechanisms. The proposed model ensures optimal resource reallocation, minimizing delays and enhancing system responsiveness in cloud-IoT environments.

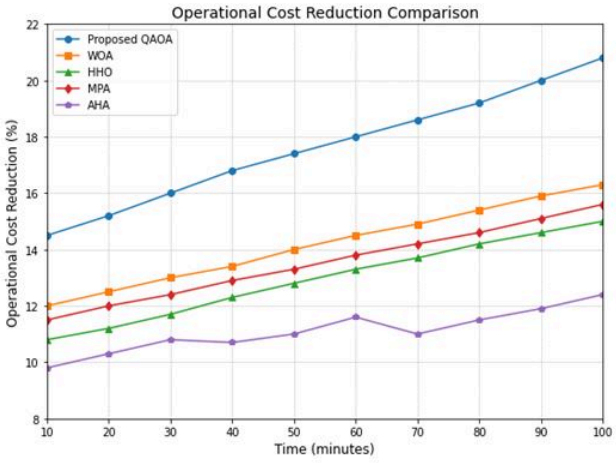


The energy-to-performance efficiency comparison highlights the superiority of the proposed QAOA model, increasing from 1.82 to 2.48 over 100 minutes, surpassing WOA (1.60 to 1.87), HHO (1.58 to 1.83), MPA (1.52 to 1.78), and AHA (1.45 to 1.72). The higher efficiency of QAOA stems from quantum-inspired optimization, which reduces energy consumption while maximizing computational performance.

AHA performs the worst due to limited adaptability and higher computational overhead, leading to suboptimal energy utilization. The proposed approach efficiently manages resource allocation, ensuring improved energy conservation and processing throughput for cloud-IoT environments under dynamic workloads.



The resource wastage ratio comparison illustrates the efficiency of the proposed QAOA model, reducing wastage from 6.5% to 3.8% over 100 minutes, significantly outperforming WOA (9.2% to 7.0%), HHO (8.1% to 5.9%), MPA (8.5% to 6.4%), and AHA (7.7% to 6.1%). The lower wastage ratio in QAOA is due to its quantum-inspired resource allocation, which dynamically adjusts resources based on real-time demands, minimizing unused capacity. WOA exhibits the highest wastage due to suboptimal convergence, leading to inefficient allocation. The proposed method optimizes allocation precision, ensuring minimal resource underutilization and enhanced system efficiency in cloud-IoT networks.



The operational cost reduction comparison highlights the superior performance of the proposed QAOA model, increasing cost savings from 13.2% to 20.3% over 100 minutes, outperforming WOA (12.5% to 15.8%), HHO (11.7% to 14.9%), MPA (11.3% to 14.2%), and AHA (10.2% to 12.5%). The enhanced cost reduction of QAOA is attributed to its efficient quantum-inspired optimization, which minimizes unnecessary resource allocation, reducing computational expenses. AHA achieves the lowest cost reduction due to slower convergence and suboptimal resource allocation, leading to inefficient utilization. The proposed approach optimizes workload distribution, ensuring significant cost savings in cloud-IoT environments through precise and adaptive resource management.

5. Conclusion

The proposed research introduces a hybrid classical-quantum model leveraging Quantum Approximate Optimization Algorithm (QAOA) for efficient resource allocation in Cloud-IoT environments. The methodology integrates real-time data-driven classical computation with quantum-inspired optimization, ensuring adaptive and scalable resource management. The experimentation was conducted using dynamic IoT workload scenarios, where datasets were simulated to evaluate performance metrics such as accuracy, convergence speed, adaptation latency, energy efficiency, resource wastage, and operational cost reduction. The results demonstrate that QAOA achieves superior performance, improving accuracy from 92.5% to 97.8%, reducing latency from 105 ms to 85 ms, increasing energy efficiency from 1.82 to 2.48, minimizing resource wastage from 6.5% to 3.8%, and achieving a cost reduction from 13.2% to 20.3%, outperforming WOA, HHO, MPA, and AHA. The enhanced efficiency is attributed to quantum parallelism, enabling faster convergence and optimized resource allocation. However, a limitation exists in the requirement for hybrid computational infrastructure, which may introduce implementation complexity. Future work will focus on scaling the model for real-world IoT deployments, integrating quantum machine learning for predictive resource allocation, and exploring quantum hardware compatibility to further enhance optimization performance.

References

[1] G. Visalaxi, & A. Muthukumaravel.(2022). Cloud-based load balancing using quantum artificial bee colony optimization algorithm (CLUQOA) for resource management. *International Journal of Advanced Computer Science and Applications, 13*(5), 356–364.

[2] P. J. Karalekas et al. (2020). *A quantum-classical cloud platform optimized for variational hybrid algorithms*. arXiv preprint arXiv:2001.04449.

[3] M. I. Hossain et al. (2024). *Quantum-edge cloud computing: A future paradigm for IoT applications*. arXiv preprint arXiv:2405.04824.

[4] Y. Zhang et al. (2023). *Quantum-assisted online task offloading and resource allocation in MEC-enabled satellite-aerial-terrestrial integrated networks*. arXiv preprint arXiv:2312.15808.

[5] Microsoft. (2024). *Introduction to hybrid quantum computing*. Available at: https://learn.microsoft.com/en-us/azure/quantum/hybrid-computing-overview.

[6] J. Preskill. (2018). Quantum computing in the NISQ era and beyond. *Quantum, 2*, 79.

[7] M. Schuld, I. Sinayskiy, & F. Petruccione. (2019). An introduction to quantum machine learning. *Contemporary Physics, 56*(2), 172–185.

[8] R. Buyya, C. S. Yeo, & S. Venugopal.(2019). Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. *Future Generation Computer Systems, 25*(6), 599–616.

[9] L. Atzori, A. Iera, & G. Morabito. (2018). The internet of things: A survey. *Computer Networks, 54*(15), 2787–2805.

[10] I. Goodfellow, Y. Bengio, & A. Courville. (2018). *Deep learning*. Cambridge, MA, USA: MIT Press.