Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062371

ARTICLE

Quantum-Enhanced Edge Offloading and Resource Scheduling with Privacy-Preserving Machine Learning

Junjie Cao[1,2], Zhiyong Yu[2,*], Xiaotao Xu[1], Baohong Zhu[3] and Jian Yang[2]

[1]College of Information and Communication, National University of Defense Technology, Wuhan, 430035, China
[2]Xi'an Research Institute of High Technology, Xi'an, 710025, China
[3]Tencent Cloud Computing (Xi'an) Co. Ltd., Xi'an, 710075, China

*Corresponding Author: Zhiyong Yu. Email: yutouzy@163.com

Received: 17 December 2024; Accepted: 13 March 2025; Published: 19 May 2025

ABSTRACT: This paper introduces a quantum-enhanced edge computing framework that synergizes quantum-inspired algorithms with advanced machine learning techniques to optimize real-time task offloading in edge computing environments. This innovative approach not only significantly improves the system's real-time responsiveness and resource utilization efficiency but also addresses critical challenges in Internet of Things (IoT) ecosystems—such as high demand variability, resource allocation uncertainties, and data privacy concerns—through practical solutions. Initially, the framework employs an adaptive adjustment mechanism to dynamically manage task and resource states, complemented by online learning models for precise predictive analytics. Secondly, it accelerates the search for optimal solutions using Grover's algorithm while efficiently evaluating complex constraints through multi-controlled Toffoli gates, thereby markedly enhancing the practicality and robustness of the proposed solution. Furthermore, to bolster the system's adaptability and response speed in dynamic environments, an efficient monitoring mechanism and event-driven architecture are incorporated, ensuring timely responses to environmental changes and maintaining synchronization between internal and external systems. Experimental evaluations confirm that the proposed algorithm demonstrates superior performance in complex application scenarios, characterized by faster convergence, enhanced stability, and superior data privacy protection, alongside notable reductions in latency and optimized resource utilization. This research paves the way for transformative advancements in edge computing and IoT technologies, driving smart edge computing towards unprecedented levels of intelligence and automation.

KEYWORDS: Edge offloading; resource scheduling; machine learning; privacy protection

1 Introduction

The rapid advancement of IoT technology has led to the deployment of billions of intelligent devices worldwide, generating massive volumes of data daily. IoT applications now permeate various sectors—from smart homes and industrial automation to intelligent transportation and healthcare—covering nearly all facets of human life. However, this proliferation of IoT devices also brings significant challenges in data processing. Traditional data center processing methods are increasingly unable to meet real-time requirements due to the explosive growth and sheer volume of IoT data. Furthermore, terminal devices often suffer from limited computational power and resource constraints, making it challenging for them to efficiently process locally generated large-scale data. Security concerns, such as data breaches and tampering during transmission, further complicate the landscape [1]. Edge computing has emerged as a promising solution to these issues. By executing computational tasks at the network edge, edge computing

minimizes data transmission distances and times, thereby reducing latency and enhancing response speeds. Additionally, it alleviates pressure on cloud data centers, leading to more flexible and efficient data processing. Despite these advantages, traditional edge computing approaches still struggle with slow response times and inefficient scheduling when handling dynamic workloads and resource uncertainties in complex and diverse IoT environments [2].

In recent years, quantum computing—an emerging field of computation—has attracted considerable attention for its unique computational capabilities and potential in processing large-scale datasets. Leveraging principles of quantum mechanics, quantum computing employs qubits instead of classical binary bits for information storage and processing [3]. Quantum computers can offer substantial speed advantages over classical counterparts in certain tasks, presenting new possibilities for addressing current edge computing challenges.

This study aims to explore a novel paradigm that integrates quantum computing with edge computing, leveraging quantum computing's strengths to compensate for traditional edge computing's limitations in processing complex computational tasks, particularly the demand for real-time data processing in IoT environments [4]. Our objective is to design an edge computing framework that effectively utilizes quantum computing capabilities and evaluate its performance across various IoT application scenarios. Research into quantum-enhanced edge computing holds significant academic value and promises broad application prospects. Main Contributions and Innovations:

(a) Quantum-Enhanced Edge Computing Model: We propose a quantum-enhanced edge computing model that innovatively combines quantum computing with edge computing to optimize real-time data processing performance in IoT environments, especially excelling in low-latency applications.
(b) Quantum-Inspired Task Prediction and Resource Offloading Strategy: A strategy based on quantum-inspired learning is developed for task prediction and resource offloading. By integrating online learning and predictive technologies, this approach anticipates task demands and intelligently allocates computational resources, further reducing response time and energy consumption. The introduction of Grover's algorithm and multi-controlled Toffoli gates enhances scheduling efficiency and robustness.
(c) Improved Optimization Algorithm: An innovative quantum-enhanced optimization algorithm is designed, significantly improving data processing speed and efficiency. Security measures, including encryption techniques and efficient monitoring mechanisms, ensure maximum real-time performance and resource utilization while safeguarding data privacy.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 describes the system architecture and problem formulation; Section 4 introduces the improved algorithms; Section 5 presents experimental results and analysis; and Section 6 concludes the paper.

2 Related Work

With the rapid development of IoT technology, billions of smart devices have been deployed globally, generating vast amounts of real-time data. These data are not only voluminous but also span a wide range of application scenarios such as smart homes, intelligent transportation, and industrial automation. However, traditional cloud computing centers can no longer meet the requirements for real-time processing, leading to increased data processing delays and degraded user experience. To address these issues, edge computing has emerged as a new computing paradigm. By executing computational tasks at the network edge, it reduces data transmission distances and times, thereby lowering latency and improving response speed. Nevertheless, with the increasing complexity and diversity of IoT applications, traditional edge computing still faces limitations, particularly in handling large-scale complex computational tasks where performance may be

constrained. Integrating quantum computing with edge computing has become a hot research topic aimed at leveraging the advantages of quantum computing to compensate for the deficiencies of traditional edge computing in processing complex tasks.

2.1 Distributed Computing Offloading

In recent years, research on distributed computation offloading techniques has explored multiple dimensions focusing on efficiency improvement and dynamic adaptability. At the algorithm optimization level, Shi et al. [5] proposed a task offloading framework based on dual-network deep reinforcement learning, which reduces Q-value bias by separating action selection from evaluation mechanisms, achieving lower latency on the dataset. However, its static task queue assumption and independent resource allocation strategy limit its applicability in dynamic scenarios. For mobility-enhanced scenarios, Dai et al. [6] innovatively introduced a Unmanned Aerial Vehicle (UAV)-assisted offloading architecture, combining Lyapunov optimization with Markov approximation methods to validate latency optimization and long-term energy consumption stability using real traffic datasets. Yet, the single-UAV service model and binary offloading mechanism restrict the flexibility of complex task scheduling. In green computing, notably, Liu et al. [7] investigated energy-efficient computing offloading in a Wireless Powered Mobile Edge Computing (WP-MEC) network with multiple Hybrid Access Points (HAPs). They proposed a Two-stage Multi-Agent deep reinforcement learning-based Distributed computation Offloading (TMADO) framework to optimize energy consumption while meeting constraints on computing delay and data demand. Although effective in reducing energy use, the approach has limitations in real-time performance and data privacy. This underscores the need for more adaptable offloading strategies in dynamic edge environments. Despite breakthroughs in different dimensions, these studies share common limitations: ineffective coupling of dynamic environment modeling with channel state changes, load fluctuations, and long-term energy constraints; lack of heterogeneous resource collaborative scheduling mechanisms, with untapped potential for cross-domain optimization among UAVs, HAPs, and ground nodes; rigid task offloading models lacking differentiated adaptation mechanisms for compute-intensive and latency-sensitive tasks.

2.2 Information Freshness Optimization

In edge computing and IoT scenarios, Age of Information (AoI) serves as a critical metric for measuring data freshness, with optimization research trending towards multidimensional development. Current work primarily focuses on communication architecture innovation, intelligent resource scheduling, and multi-objective collaboration. In novel communication architecture design, Pei et al. [8] proposed a UAV-assisted edge computing system, optimizing communication and computational resource allocation through a three-tier transmission architecture of sensor-UAV-base station, utilizing the Lyapunov method for long-term energy constraint management. This scheme significantly reduces system AoI compared to traditional transmission modes but is limited by assumptions of a single-UAV scenario, requiring improved capabilities in handling multi-user interference. For task segmentation scenarios, Kim et al. [9] introduced an AoI-aware mechanism into the edge computing decision-making framework, optimizing task segmentation strategies by modeling the time-varying characteristics of computational resources, providing new insights for the joint optimization of information freshness and task success rates. In distributed intelligent scheduling, Zhang et al. [10] combined non-orthogonal multiple access technologies with deep reinforcement learning to design a hierarchical agent architecture addressing multi-base station collaborative optimization challenges. The core innovation lies in optimizing throughput, fairness, and dynamic balancing mechanisms through independent agents, effectively overcoming performance bottlenecks of traditional centralized scheduling, offering scalable solutions for AoI control in large-scale distributed scenarios. Notably, such

methods must still address the balance between policy generalization and computational overhead in dynamic environments.

2.3 Quantum-Enhanced Edge Computing

Quantum computing has shown significant potential in enhancing edge computing capabilities, particularly in task scheduling, resource allocation, and security. For instance, Wang et al. [11] applied the Quantum Particle Swarm Optimization (QPSO) algorithm to device-edge-cloud collaborative computing, demonstrating superior performance in user satisfaction and resource efficiency. In resource allocation, Bhatia et al. [12] leveraged the Quantum Approximate Optimization Algorithm (QAOA) to optimize power allocation in edge computing environments, achieving significant improvements in latency and reliability. Similarly, Mastroianni et al. [13] utilized Variational Quantum Algorithms (VQAs) to address resource allocation challenges in cloud-edge architectures, demonstrating excellent success rates and execution times. However, these methods face practical challenges due to hardware limitations and quantum noise. In the realm of security and privacy, quantum computing offers robust solutions. Singamaneni et al. [14] proposed a Quantum Hash-based Attribute-Based Encryption (QH-ABE) method, enhancing data integrity and access control in edge computing. Telsang et al. [15] introduced a blockchain-based authentication mechanism using quantum keys, achieving high security and low latency. Despite these advancements, the practical deployment of quantum security protocols remains challenging due to issues such as quantum key distribution reliability.

2.4 Intelligent Task Scheduling

Intelligent task scheduling, as a core technology in edge computing and autonomous driving systems, has made significant progress in areas such as 6G communications, vehicle-road collaboration, and UAV-assisted mobile edge computing (MEC). Current research mainly focuses on dynamic environmental adaptability, multi-objective optimization, and heterogeneous resource management, gradually overcoming the limitations of traditional methods by integrating deep reinforcement learning, federated scheduling, and multi-agent collaborative strategies. In vehicle-road collaboration scenarios, Li et al. [16] proposed a fully decentralized multi-agent proximal policy optimization-based vehicle-infrastructure network, dynamically allocating multi-sensor tasks to edge servers and idle vehicles to achieve joint optimization of task completion time and energy consumption. Its innovation lies in constructing real-time decision models considering vehicle mobility, network load, and task characteristics, expanding resource pools via Vehicle-to-Everything (V2X) communication. Similarly, Xu et al. [17] designed a Deep Q-Network-driven cooperative task placement algorithm, building a cloud-edge collaborative framework that optimizes computational offloading in vehicular networks through state-space modeling and dynamic pricing mechanisms, significantly reducing system latency and energy consumption. For energy-constrained embedded systems, Mohammadi et al. [18] proposed an energy-harvesting-aware federated scheduling algorithm, employing windowed scheduling and dynamic core allocation strategies, combining first-fit and last-fit methods to optimize multi-core real-time task scheduling. Their core contribution involves establishing mathematical models for task execution time and core allocation, theoretically proving the impact of battery capacity thresholds on algorithm optimality, providing efficient energy management solutions for high-utilization parallel tasks. In multi-UAV-assisted MEC system optimization, Abdel-Basset et al. [19] proposed a bi-level optimization framework, enhancing upper-level deployment optimization algorithms through serialized replacement strategies maintaining individual characteristics, combining greedy algorithms to achieve joint optimization of UAV deployment and task scheduling. This scheme exhibits stability advantages at a scale of 900 tasks, introducing local escape operators and gradient search rules to avoid local optima and accelerate convergence. The current

research still has core bottlenecks: the balance problem between the efficiency of multi-intelligence body collaboration and communication overhead; the dynamic conflict management between energy supply and task timeliness; and the unexpected task response delay problem of heterogeneous node clusters. The future breakthrough direction will focus on the design of lightweight algorithms, the fusion application of digital twin and federated learning, and the construction of cross-modal optimization theory for dynamic resource scheduling.

While significant progress has been made in intelligent task scheduling, distributed computing offloading, information freshness optimization AoI, and quantum-enhanced edge computing, several critical challenges remain unresolved, limiting the practical deployment and scalability of existing solutions. Existing methods often struggle to adapt to rapidly changing task demands and resource availability, exhibit high computational complexity in large-scale problems, and fail to balance computational efficiency with real-time responsiveness, particularly in latency-sensitive applications. Additionally, the integration of quantum computing with edge systems faces hardware limitations, high costs, and compatibility issues, while current approaches often lack generality and flexibility to handle diverse edge offloading scenarios. Data privacy and security also remain significant concerns, as existing solutions are often insufficiently robust against advanced threats, including quantum computing attacks. To address these challenges, this study proposes a Quantum-Enhanced Edge Computing Framework (QECF) that integrates quantum-inspired algorithms with machine learning techniques. By leveraging Grover's algorithm for accelerated search and multi-controlled Toffoli gates for efficient constraint evaluation, QECF enhances computational efficiency, adaptability, and robustness. The framework also incorporates advanced encryption and monitoring mechanisms to ensure data privacy and security, making it well-suited for dynamic and resource-constrained edge environments. QECF's ability to handle diverse edge offloading scenarios while ensuring real-time performance, scalability, and security represents a significant advancement in the field, bridging the gap between theoretical advancements and practical deployment in real-world applications such as smart cities, industrial IoT, and healthcare.

3 System Architecture and Problem Description

This paper introduces a novel hybrid architecture model named the QECF. The QECF aims to integrate the strengths of quantum computing with traditional edge computing to achieve a high-performance, secure, and scalable computational environment.

3.1 System Architecture

The architecture is composed of three layers, as illustrated in Fig. 1.

(a) Terminal Layer. This layer encompasses various IoT devices and user equipment that generate data. These devices transmit data to edge nodes via wireless communication technologies. The terminal layer serves as the front line for data collection and initial transmission.

(b) Edge Core Layer. The edge core layer receives data from the end-user layer, performs preprocessing, and handles task allocation through a Coordinating Management Unit (CMU). This layer features two types of processing nodes:

- Quantum Edge Nodes (QENs): Equipped with quantum processors and quantum storage devices, these nodes execute quantum computing and communication tasks. Positioned close to data sources at the network edge, QENs minimize latency and enhance data processing speed.

- Traditional Edge Nodes (TENs): These nodes handle conventional computational tasks that do not require quantum capabilities and manage the coordination of QENs. Both QENs and TENs communicate through Quantum Gateways (QGWs).

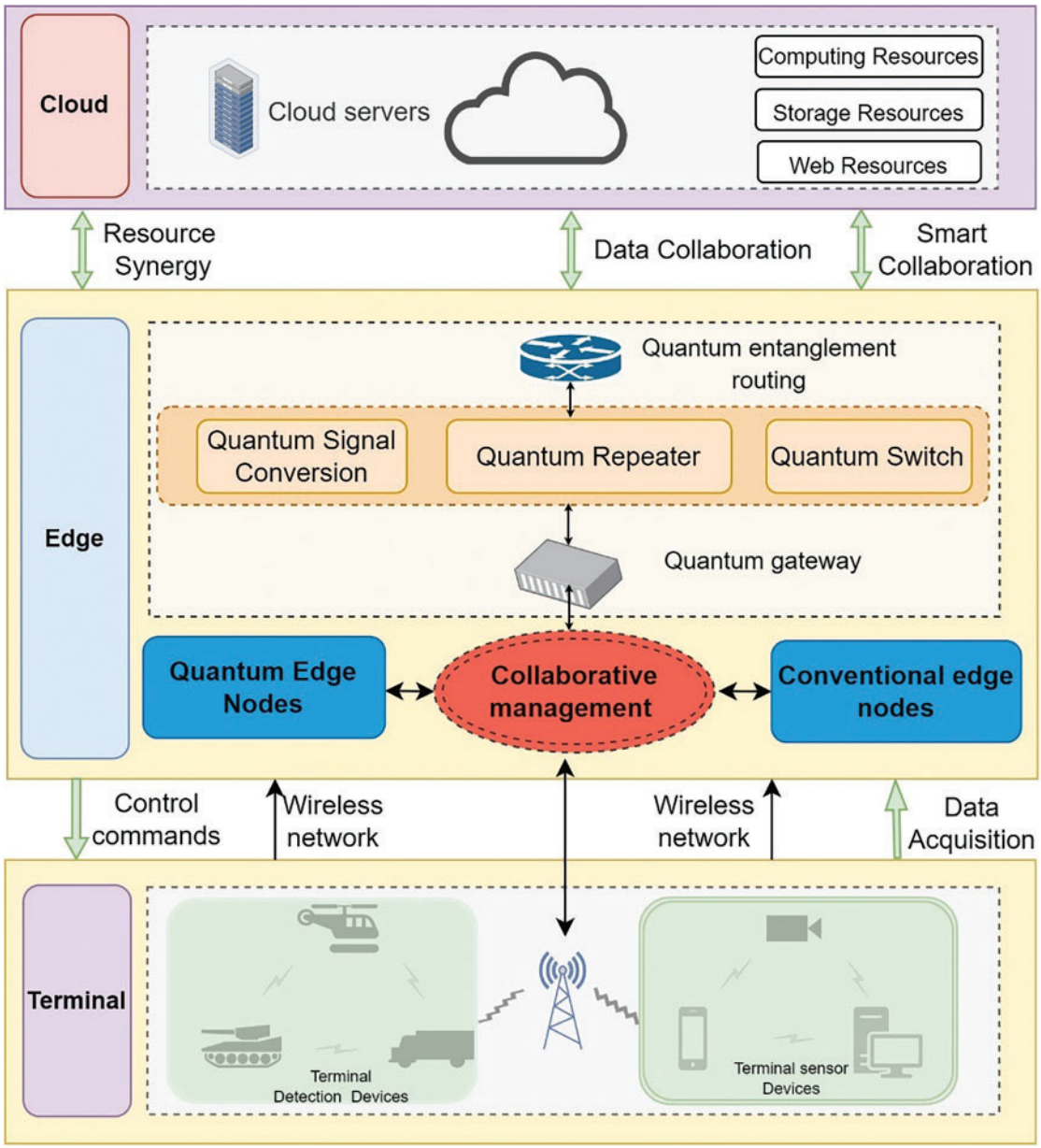

Figure 1: Quantum-enhanced edge computing framework

Additionally, this layer centers around the quantum unit, which returns optimized results to either the edge nodes or cloud servers. The CMU ensures efficient collaboration between quantum and classical edge computing units by dynamically allocating resources based on task requirements.

(c) Cloud Layer. The cloud layer provides extensive computational power, storage resources, and advanced analytics capabilities. It also manages and monitors the entire network, ensuring seamless operation across all layers.

To better understand how the QECF achieves efficient data processing, resource management, and secure transmission, it is essential to delve into its key components:

QENs integrate quantum processors and quantum storage devices, enabling them to perform quantum computing and quantum communication tasks. They are distributed at the network edge, close to data sources, to reduce latency and increase data processing speed. TENs, on the other hand, handle conventional computational tasks that do not require quantum computing capabilities and manage and coordinate the Quantum Edge Nodes. The Coordinating Management Unit ensures efficient collaboration between quantum units and classical edge computing units, dynamically allocating resources based on the specific requirements of each task. QGWs, serving as a bridge between quantum networks and traditional networks, are responsible for the conversion and routing of quantum signals, ensuring the secure transmission of quantum information. Quantum Repeaters are used to extend the quantum communication network, overcoming the attenuation issues of quantum signals during transmission. Quantum Switches perform quantum entanglement routing within the quantum network, similar to optical switches in traditional networks, ensuring the effective transmission and processing of quantum information.

To ensure data privacy and security, it employs advanced encryption techniques, including AES-256 for data encryption and RSA-2048 for key management. AES-256 (256-bit key, 14-round substitution-permutation network) encrypts data blocks in CBC-GCM (Cipher Block chaining-Galois/Counter Mode) mode, providing authenticated encryption with 128-bit integrity tags. RSA-2048 (2048-bit modulus based on the hardness of integer factorization) manages key exchange via OAEP (Optimal Asymmetric Encryption Padding) padding, preventing chosen-ciphertext attacks. Additionally, we have implemented a quantum key distribution protocol to defend against potential quantum computing attacks. The monitoring mechanism continuously tracks the state of tasks and resources, ensuring timely detection and response to any security threats. These measures provide robust protection for sensitive data in edge computing environments.

Traditional edge computing architectures are typically designed for static or semi-static environments, making them ill-suited to adapt to dynamic workloads, resulting in low resource utilization and increased latency. Existing quantum integration solutions often focus on specific tasks such as scheduling or energy optimization, without fully addressing data privacy and security concerns. Additionally, current AoI management methods rely on preset rules, lacking sufficient flexibility in highly dynamic environments. In contrast, the QECF introduces multi-layer innovations, significantly enhancing the system's flexibility, efficiency, and security. By integrating QENs and TENs, QECF can handle a wide range of workloads from conventional to high-demand quantum tasks, ensuring efficient task allocation and execution. Utilizing a CMU, resources are dynamically allocated based on real-time needs, allowing the system to adapt to rapidly changing workloads and network conditions. The introduction of QGWs and Quantum Repeaters ensures secure transmission of quantum signals and overcomes the limitations of traditional networks in handling quantum information. Furthermore, QECF's novel AoI management mechanism, which combines machine learning models with quantum-enhanced algorithms, predicts and minimizes information freshness in dynamic environments, ensuring timely and accurate data processing.

3.2 Workflow

The entire workflow is depicted in Fig. 2. At the terminal user layer, various sensors are strategically placed to collect raw data from the environment and transmit this data to the data acquisition and interaction unit. During the task allocation phase, the CMU intelligently assesses each task's specific requirements

and the current availability of computational resources. Based on this assessment, the CMU determines whether tasks should be processed by classical edge computing units or quantum units. In the data processing phase, traditional edge computing units handle data preprocessing, feature extraction, and occasionally initial model training. Meanwhile, quantum units tackle computationally intensive tasks such as complex optimization problems and advanced pattern recognition. Next, during the result integration phase, the CMU synthesizes the outputs from both quantum and classical edge computing units. This synthesis ensures the consistency and accuracy of all information, providing a unified and reliable dataset. Finally, in the result feedback phase, the processed data or decision instructions are transmitted to the appropriate actuators or destinations, such as cloud servers, for further application or storage. This seamless flow not only leverages the complementary strengths of classical and quantum computing but also underscores their potential for efficient collaboration within IoT environments [20].

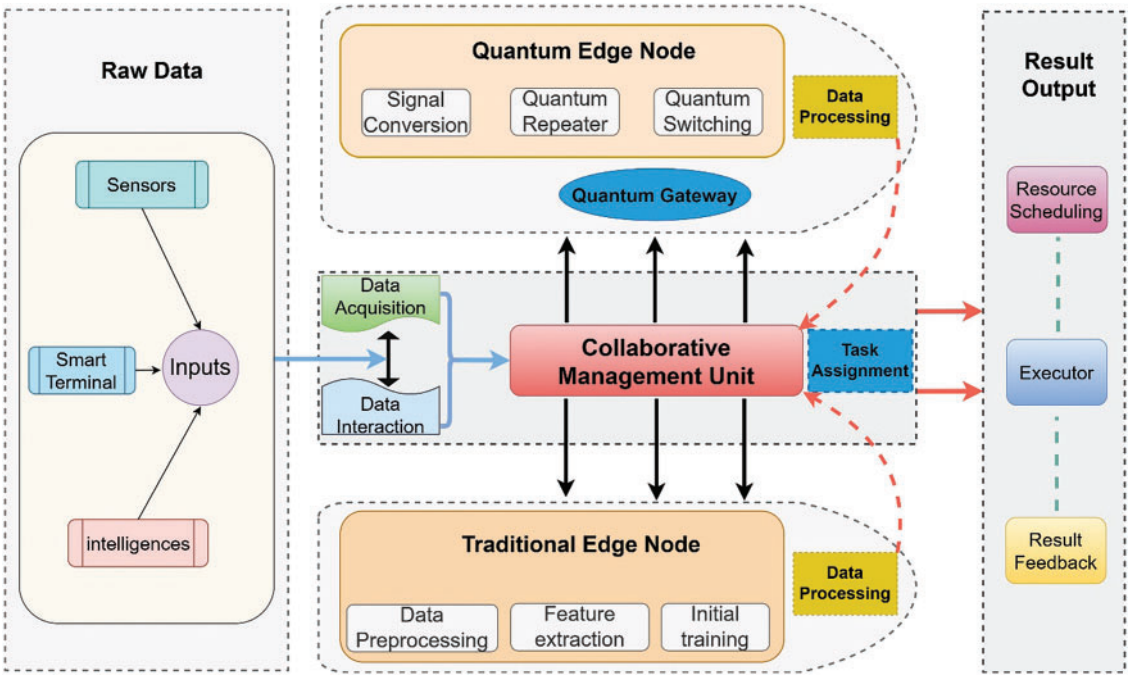

Figure 2: QECF workflow diagram

The QECF integrates the advantages of quantum computing, quantum communication, and traditional edge computing to provide a high-performance, secure, and scalable hybrid computing framework. The innovations in this architecture are highlighted as follows:

(a) Integration of QENs and TENs: This integration considers the synergy between quantum and classical computing to enhance computational efficiency and processing capabilities. By incorporating quantum computing power at the edge layer, QECF can handle complex computational tasks that are beyond the reach of traditional edge computing.

(b) Introduction of QGWs and Quantum Switches: These components are responsible for the conversion and routing of quantum signals, serving as key enablers for integrating quantum communication networks with traditional networks. Through quantum gateways and quantum repeaters, QECF achieves seamless integration between quantum and traditional networks, thereby improving the security and efficiency of data transmission.

(c) Three-Layer Architecture Design: Dividing the architecture into endpoint, edge, and core layers, each with specific functions and responsibilities, this layered approach facilitates the clear organization and management of complex computational resources. It ensures a structured way to manage various tasks and optimize resource allocation across different layers.

(d) Intelligent Resource Scheduling and Quantum Security Protocols: Dynamic resource allocation and quantum security measures are proposed to adapt to changing computational demands and enhance network security. Intelligent algorithms dynamically distribute computational tasks, optimizing resource usage and improving network responsiveness and computational efficiency. Quantum security protocols ensure robust protection against potential threats, maintaining the integrity and confidentiality of data.

3.3 Problem Description

In IoT environment, terminal devices such as sensors and mobile devices generate substantial volumes of real-time data that require prompt processing to meet various application demands. Although traditional edge computing can partially address these needs, it still encounters challenges when handling large-scale datasets, including limited computational resources, restricted network bandwidth, and concerns over data security. To address these issues, we propose a quantum-enhanced edge computing model aimed at leveraging the powerful processing capabilities and low-latency characteristics of quantum computing to optimize data processing performance [21].

Assuming there are N terminal devices in the IoT system, each generating a data volume denoted by d_i, where $i = 1, 2, \ldots, N$. This data is transmitted to M edge nodes, each with a computational capacity C_j, storage capacity s_j, and network bandwidth b_j, where $j = 1, 2, \ldots, M$. Additionally, the quantum computing center has a computational capacity C_q, storage capacity S_q, and network bandwidth B_q.

To optimize task scheduling, we define a task scheduling matrix X, where X_{ij} indicates whether the data from the i-th terminal device is assigned to the j edge node for processing; $X_{ij} = 1$ means assignment, while $X_{ij} = 0$ means no assignment. Similarly, we define a quantum task scheduling matrix Y, where Y_{jk} denotes whether the data from the j edge node is further assigned to the quantum computing center for processing; $Y_{jk} = 1$ indicates assignment, whereas $Y_{jk} = 0$ indicates no assignment.

3.3.1 Total Delay

The total delay includes the data transmission delay D_t and the data processing delay D_p.

The data transmission delay D_t: This represents the time required to transmit data from the terminal devices to the edge nodes and from the edge nodes to the quantum computing center. It is calculated as:

$$D_t = \sum_{i=1}^{N} \sum_{j=1}^{M} X_{ij} \frac{d_i}{b_j} + \sum_{j=1}^{M} \sum_{K=1}^{K} Y_{jk} \frac{\sum_{i=1}^{N} X_{ij} d_i}{B_q} \tag{1}$$

where d_i is the data volume generated by the i-th terminal device, b_j is the network bandwidth of the j-th edge node, and B_q is the network bandwidth of the quantum computing center. X_{ij} and Y_{jk} are binary variables indicating whether the data is assigned to the j-th edge node or the quantum computing center, respectively.

The data processing delay D_p: This represents the time required to process the data at the edge nodes and the quantum computing center. It is calculated as:

$$D_p = \sum_{j=1}^{M} \left(\frac{\sum_{i=1}^{N} X_{ij} d_i}{C_j} + \sum_{K=1}^{K} Y_{jk} \frac{\sum_{i=1}^{N} X_{ij} d_i}{C_q} \right) \tag{2}$$

where C_j is the computational capacity of the j-th edge node, and C_q is the computational capacity of the quantum computing center.

Therefore, the total delay D can be expressed as:

$$D = D_t + D_p \tag{3}$$

3.3.2 Total Energy Consumption

The total energy consumption includes the energy consumption of the edge nodes E_e and the energy consumption of the quantum computing centre E_q.

Energy Consumption of Edge Nodes E_e: This represents the energy consumed by the edge nodes to process the data. It is calculated as:

$$E_e = \sum_{j=1}^{M} \left(\frac{\sum_{i=1}^{N} X_{ij} d_i}{c_j} * e_j \right) \tag{4}$$

where e_j is the unit energy consumption of the j edge node.

Energy Consumption of Quantum Computing Center E_q: This represents the energy consumed by the quantum computing center to process the data. It is calculated as:

$$E_q = N_q \frac{d_q}{C_q} * e_q \tag{5}$$

where e_q is the unit energy consumption of the quantum computing centre.

Therefore, the total energy consumption E can be expressed as:

$$E = E_e + E_q \tag{6}$$

3.3.3 Data Processing Efficiency

The data processing efficiency can be expressed as the amount of data processed per unit time. Assuming that the processing rate of each edge node is r_j and the processing rate of the quantum computing centre is r_q, the data processing efficiency P can be expressed as:

$$P = \sum_{j=1}^{M} \left(\sum_{i=1}^{N} X_{ij} d_i * r_j \right) + \sum_{J=1}^{M} \sum_{K=1}^{K} Y_{jk} \left(\sum_{i=1}^{N} X_{ij} d_i * r_q \right) \tag{7}$$

Our objective is to minimize the total delay D and total energy consumption E, while maximizing the data processing efficiency P. The specific optimization objective can be expressed as:

$$\min \alpha D + \beta E - \gamma P \tag{8}$$

s.t.

$$\sum_{j=1}^{M} X_{ij} = 1, \forall i = 1, 2, \ldots N \tag{9}$$

$$\sum_{i=1}^{N} X_{ij} d_i \le s_j, \forall j = 1, 2, \ldots M \tag{10}$$

$$\sum_{i=1}^{N} X_{ij} d_i \le c_j, \forall j = 1, 2, \ldots M \tag{11}$$

$$\sum_{j=1}^{M} \sum_{K=1}^{K} Y_{jk} \left(\sum_{i=1}^{N} X_{ij} d_i \right) \le S_q \tag{12}$$

$$\sum_{j=1}^{M}\sum_{K=1}^{K} Y_{jk}\left(\sum_{i=1}^{N} X_{ij}d_i\right) \le C_q \tag{13}$$

where α, β, γ are weight coefficients to balance the importance of different objectives. To ensure the feasibility of task scheduling and the stability of the system, we need to satisfy the above constraints ((9)–(13)):

The data of each terminal device can only be assigned to one edge node; The amount of data of each edge node cannot exceed its storage capacity; The amount of data processing of each edge node cannot exceed its computational capacity; The amount of data assigned to the quantum computing centre cannot exceed its storage capacity; The amount of data processing assigned to the quantum computing centre cannot exceed its computational capacity.

3.3.4 Challenges of the Formulated Problem

The formulated problem presents several key challenges:

(a) High Demand Variability: IoT environments are characterized by highly dynamic task demands, which require real-time adaptation of task offloading and resource allocation strategies.
(b) Resource Unpredictability: The availability of computational resources at the edge nodes and the quantum computing center can vary significantly, making it difficult to ensure efficient task scheduling.
(c) Data Privacy Requirements: The need to protect sensitive data during transmission and processing adds an additional layer of complexity to the problem.

These challenges highlight the need for a sophisticated approach that can dynamically adapt to changing conditions while maintaining high levels of security and efficiency. Addressing these issues effectively will require a solution that integrates advanced algorithms capable of handling dynamic task demands, managing unpredictable resource availability, and ensuring robust data privacy. In the following sections, we introduce an enhanced algorithm specifically designed to tackle these challenges.

4 Improved Algorithm

This paper proposes an optimization algorithm that combines quantum computing with dynamic task scheduling—Quantum-Enhanced Adaptive Task Offloading and Scheduling (QATOS)—aimed at addressing the challenges faced in task allocation within edge computing environments. The QATOS algorithm is designed to address the optimization problem formulated in Section 3, which aims to minimize total delay and energy consumption while maximizing data processing efficiency. The algorithm leverages quantum-inspired techniques and machine learning to dynamically adapt to changing task demands and resource availability. Specifically, the Grover search algorithm is used to accelerate the search for optimal task offloading solutions, while the multi-controlled Toffoli gates ensure efficient constraint evaluation. The integration of these techniques allows the QATOS algorithm to effectively solve the multi-objective optimization problem discussed in Section 3.

4.1 Algorithm Description

The QATOS algorithm achieves efficient and flexible task scheduling by dynamically updating task and resource states in real-time, adaptively adjusting quantum optimization parameters, and leveraging online learning and prediction techniques. In edge computing environments, task allocation faces challenges such as high demand volatility and uncertain resource availability, which traditional static scheduling methods struggle to address. To this end, the core of the Quantum Enhanced Edge Computing framework consists of advanced algorithms that optimize task scheduling and resource allocation. These algorithms leverage quantum computing principles to improve efficiency and effectiveness. Specifically, we integrate Grover

search algorithms and multi-control Toffoli gates to achieve superior performance in task allocation and constraint evaluation.

The Grover search algorithm is a quantum algorithm designed for searching unsorted databases, capable of finding target items in $O(\sqrt{N})$ time complexity, compared to the $O(N)$ time complexity of classical algorithms. This can significantly enhance the efficiency of data retrieval. The Grover search algorithm is applied in quantum edge nodes (QENs) to accelerate the search process for optimal task allocation. The algorithm provides a quadratic speedup compared to classical search methods by efficiently searching for potential solutions. When a task is received from an end device, the QENs preprocess the data and apply Grover's algorithm to identify the best way to assign the task to available resources. By iteratively amplifying the magnitude of the correct solution, the Grover algorithm quickly converges to the optimal or near-optimal solution, significantly reducing the time complexity of the optimization process.

Assuming a database with N items where the target item is denoted as ω, the number of iterations R required by the Grover search algorithm can be expressed as:

$$R = \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor \tag{14}$$

After each iteration, the probability P of finding the target item can be expressed as:

$$P = \sin^2\left((2R+1)\theta\right) \tag{15}$$

where $\theta = \sin^{-1}\frac{1}{\sqrt{N}}$.

While this provides a significant speedup compared to classical algorithms, it may still strain computational resources in resource-constrained edge computing environments. To address this challenge, we have optimized the Grover search algorithm by reducing the number of iterations and implementing efficient quantum gate operations.

Multi-control Toffoli gates are used to evaluate constraints imposed by the system. These gates allow precise control of multiple quantum bits, supporting the complex operations required for resource allocation and task scheduling. After the Grover algorithm identifies potential task allocations, the multi-control Toffoli gates verify that these allocations satisfy all system constraints. If a constraint is violated, these gates adjust their state accordingly, ensuring that only feasible solutions are considered. This dual approach not only ensures that the system is able to find optimal solutions quickly, but also guarantees that the system remains robust under various constraints. By integrating these quantum elements into our framework, we achieve faster convergence of optimal solutions while ensuring that all constraints are accurately evaluated and enforced. This enhances the overall performance and reliability of our quantum-enhanced edge computing framework, especially in low-latency applications.

The core of the QATOS algorithm lies in leveraging the advantages of quantum computing for dynamic task scheduling problems, enhancing system performance through the following three main aspects:

(a) Quantum Acceleration: Utilizing the Grover search algorithm to accelerate the discovery of task allocation solutions that satisfy all constraints. Theoretically, for a problem with N possible solutions, Grover's algorithm can find the target solution within approximately $\sqrt{N}$ iterations.

(b) Dynamic Adaptability: Implementing a monitoring system that continuously tracks changes in task and resource states. Upon detecting any changes, the system immediately triggers an update process to ensure that the algorithm responds promptly to the latest conditions.

(c) Intelligent Prediction: Employing machine learning models to forecast task demands and resource availability over future periods. This allows for preemptive planning of task allocation strategies, reducing the pressure on real-time decision-making.

Additionally, the QATOS algorithm features adaptive parameter adjustment capabilities. Based on performance feedback, it automatically tunes key parameters such as the maximum number of iterations for Grover's search and other critical settings. This ensures optimal solution processes and maintains high-performance levels under varying conditions.

4.2 Algorithm Steps

(a) Initialization Phase. First, initialize all necessary parameters, including but not limited to the task list, node lists (quantum nodes, classical nodes, and cloud nodes), threshold settings, and other foundational configurations. Concurrently, establish a monitoring system to capture changes in task and resource states, ensuring timely responses to these changes.

(b) Dynamic Updates and Predictions. In the main loop, continuously monitor for new tasks arriving or changes in existing resource states. Upon detecting changes, update based on the latest information and use pre-trained machine learning models to predict future task demands and resource availability.

(c) Adaptive Adjustment. Based on the prediction results, adaptively adjust parameters such as the maximum number of iterations for Grover's search and other relevant settings.

(d) Execution of Grover Search. Construct an Oracle specific to the task allocation problem; this Oracle can identify states that satisfy all constraints within a quantum circuit. Then, by iteratively applying the Oracle and diffusion operators, gradually amplify the probability amplitudes of correct solutions while reducing those of incorrect ones. After completing the predetermined number of iterations, measure the quantum state to obtain the most likely solution.

(e) Iterative Optimization. After each execution, record relevant performance data, such as computation time and the quality of the optimal solution found, for subsequent analysis and learning. Additionally, periodically retrain or update the prediction models based on accumulated data to ensure their accuracy and reliability.

4.3 Algorithm Implementation

The following is the pseudo-code framework for the QATOS algorithm (Algorithm 1).

Algorithm 1: Quantum-enhanced adaptive task offloading and scheduling

Input: Initial tasks, quantum nodes, classical nodes, cloud nodes, threshold, initial ML model
Output: Optimal task allocation solution

```
Initialize tasks list.
Initialize node lists for quantum nodes, classical nodes, and cloud nodes.
Set threshold value for evaluating solutions.
Load or train the initial machine learning model for predictions.
Establish monitoring mechanisms to detect changes in tasks and resource states.
Set up event listeners or polling intervals to check for updates periodically.
Implement logging functionality to track changes over time.
while True do
    Check for new tasks arriving or changes in resource availability using efficient data structures like queues or sets.
    if changes detected then
```

(Continued)

Algorithm 1 (continued)

```
        Fetch updated information about tasks and resources from external sources.
            Sync with any external APIs or databases that provide real-time updates.
            Ensure consistency between internal state and external systems through validation checks.
            Prepare input features based on historical data and current system state.
            Invoke the prediction method of ML model using prepared features.
            Predicted State = Return predicted state as a structured object containing expected task demands and resource availabilities.
            Calculate new maximum iterations for Grover search:
                Max Iterations = floor(pi/4 * sqrt(N)) + Predicted State.growth_rate * alpha
            Where N is the size of the search space, Predicted State.growth_rate is the expected growth rate of tasks, and alpha is a tuning factor.
            Optionally adjust other parameters such as oracle complexity or diffusion operator settings.
            Create a quantum circuit with N qubits plus one auxiliary qubit for marking solutions.
            Encode constraints into quantum logic operations using multi-controlled Toffoli gates.
            For each constraint (cost, delay, load limit), implement corresponding quantum circuits to evaluate these conditions.
            Mark valid solutions by flipping the auxiliary qubit when all constraints are satisfied.
            Oracle = Constructed Oracle.
            Initialize a quantum circuit with N qubits and apply Hadamard gates to create superposition.
            For i from 1 to Max Iterations do:
                Apply Oracle to mark potential solutions within the superposition.
                Apply diffusion operator to amplify marked states relative to average amplitude.
            Perform measurement on the final quantum state to obtain the optimal solution.
            Optimal Solution = Measured solution as a binary string representing task assignments.
            Decode the binary string of Optimal Solution into specific task-node assignments.
            Communicate assignment decisions to relevant nodes via API calls or messaging protocols.
            Monitor execution progress and handle any issues during deployment.
            Collect metrics including execution time, number of iterations performed, quality of found solutions.
            Store metrics in a persistent storage system for long-term analysis.
            Analyze collected data to identify trends and areas for improvement.
        Evaluate criteria such as elapsed time since last update, amount of new data accumulated, performance degradation.
        if retraining is necessary then
            Gather recent observations and label them appropriately.
            Split data into training and validation sets.
            Train a new ML model or refine existing one using selected algorithms.
            Validate model performance on unseen data before deployment.
        End if
End while
```

By integrating the powerful search capabilities of quantum computing with the predictive functions of machine learning, The QATOS algorithm achieves high responsiveness and optimized decision-making in complex, dynamic environments. The algorithm introduces continuous monitoring and real-time update detection mechanisms (steps 5–7), ensuring that the system can instantly perceive changes in task demands and resource states. Based on the latest data, it performs model predictions to make task allocation strategies more aligned with actual operational conditions. When sufficient new data accumulates, the system triggers retraining of the models (steps 39–43), ensuring their timeliness and accuracy while preventing overfitting to old data. This endows the QATOS algorithm with strong adaptive capabilities. In terms of prediction, the QATOS algorithm uses machine learning models to forecast future task demands and resource availability (steps 15–16). This not only provides a basis for adjusting parameters in Grover's search, such as the maximum number of iterations (step 18), but also helps in more accurately planning task allocations. By combining predicted growth rates and tuning factors like alpha, the QATOS algorithm can pre-estimate future loads and adjust its search strategy accordingly, accelerating the convergence to optimal solutions. Notably, by constructing specific Oracles to mark potential solutions (steps 25–29), the introduction of quantum computing significantly enhances search efficiency. Compared to traditional methods, it can find better solutions on larger datasets. Most importantly, the QATOS algorithm innovatively integrates quantum computing with machine learning—not merely applying them in parallel, but using predictive models to guide key parameter settings during the quantum search process, forming a collaborative closed-loop system. This integration not only boosts the system's intelligence level but also broadens the technical pathways for solving complex problems, showcasing the immense potential of combining quantum computing with AI (Artificial Intelligence) technologies. Moreover, the QATOS algorithm supports distributed processing modes, reducing the need for centralized storage of sensitive data and lowering the risk of data breaches. Data privacy and security protection can be further enhanced through the adoption of privacy-preserving machine learning techniques and quantum encryption communication. In summary, the core competitiveness of the QATOS algorithm lies in its highly dynamic adaptability, efficient task allocation strategies, and the innovative combination of quantum computing and machine learning technologies, making significant contributions to ensuring data privacy and security [22].

4.4 Algorithm Scalability

The QATOS algorithm is designed to scale efficiently in large-scale edge computing environments. By leveraging distributed quantum and classical edge nodes, the QATOS algorithm can handle thousands of concurrent tasks with minimal latency and energy consumption. Additionally, the adaptive resource allocation mechanism ensures that the QATOS algorithm can dynamically adjust to varying workloads, making it suitable for real-world applications with high computational demands.

The key contributions of the QATOS algorithm include:

(a) Quantum-Enhanced Optimization: By leveraging Grover's search algorithm, the QATOS algorithm significantly accelerates the search for optimal task offloading solutions.
(b) Dynamic Adaptability: The algorithm's ability to dynamically adjust to changing task demands and resource availability ensures high performance in dynamic environments.
(c) Data Privacy Protection: The integration of quantum encryption techniques ensures robust data privacy, addressing a critical challenge in IoT environments.

5 Experimental Evaluation

To validate the effectiveness of the QATOS algorithm, we designed a series of experiments aimed at evaluating its performance against established methods.

5.1 Algorithm Comparison

(a) NSGA-II (Non-dominated Sorting Genetic Algorithm II): NSGA-II is a widely used evolutionary algorithm for multi-objective optimization problems. It explores the solution space by simulating natural selection and genetic mechanisms, capable of finding a set of Pareto-optimal solutions.

(b) Ant Colony Optimization (ACO): ACO mimics the food-foraging behavior of ants, using pheromone update mechanisms to find optimal paths. It is particularly suitable for solving combinatorial optimization problems and has demonstrated good adaptability and robustness in dynamic edge computing environments.

(c) Mixed Integer Linear Programming (MILP): MILP is a classic optimization method extensively applied to resource allocation and scheduling problems. It provides a stable and reliable benchmark for performance reference. In resource allocation and task scheduling, MILP offers powerful exact solutions.

5.2 Experimental Results and Analyses

Fig. 3 illustrates the latency performance under different numbers of concurrent tasks. The experimental results show that the QATOS algorithm exhibits a significant low latency advantage in processing concurrent tasks, especially when the number of concurrent tasks increases, its latency growth is relatively flat, which is significantly better than that of NSGA-II, ACO, and MILP. Specifically, at low concurrent tasks (0–200 tasks), the latency performances of all the methods are relatively close to each other, but the QATOS algorithm still exhibits the lowest latency. As the number of concurrent tasks increases (200–600 tasks), the latency growth of the QATOS algorithm is relatively flat, showing good scalability and stability. In contrast, NSGA-II has the second lowest latency performance than the QATOS algorithm, but better than ACO and MILP, whose latency also increases with the number of concurrent tasks, but the growth trend is relatively smooth; ACO's latency performance is in between that of NSGA-II and MILP, and its latency grows significantly with the number of concurrent tasks; and the latency of MILP has the worst latency performance, and its latency grows significantly with the number of concurrent tasks. MILP has the worst latency performance, with the most significant latency growth as the number of concurrent tasks increases, especially at high concurrent tasks (600–1000 tasks). These results show that the QATOS algorithm is able to handle large-scale concurrent tasks more effectively in edge computing, ensuring low latency and high efficiency, thus confirming its superior performance and potential in practical applications.

Fig. 4 illustrates the total cost for different number of concurrent tasks. The QATOS algorithm has the smallest cost range with the median located at a lower position (around 16) and the upper and lower boundaries (i.e., interquartile ranges) of the box-and-line plot are also narrower, which indicates that its cost performance is very stable and low. The low number of outliers further validates its efficiency and economy in handling the task. NSGA-II has a slightly wider cost range than the QATOS algorithm, with the median located at around 28 and the upper and lower boundaries of the box-and-line plot are also wider, suggesting that its cost fluctuates more. Although it outperforms ACO and MILP, its cost performance is not stable enough compared to the QATOS algorithm. The cost range of ACO is significantly larger than that of the QATOS algorithm and NSGA-II, with the median located at around 36, and the upper and lower boundaries of the box-and-line plot are very wide, suggesting that its cost fluctuates more and that there are more outliers. This suggests that ACO's cost performance is not stable enough in processing tasks and in some cases the costs are higher. MILP has the largest range of costs with the median located at around 38 and the upper and lower boundaries of the box-and-line plot are very wide and there are a high number of outliers. This suggests that MILP has the most erratic cost performance for processing tasks and is very costly in some cases. In summary, the QATOS algorithm shows significant low-cost advantage in processing tasks with the

smallest range and least fluctuation in cost, indicating that it is able to process tasks more efficiently in edge computing, ensuring low latency and high efficiency with excellent cost control. In contrast, NSGA-II has the next best performance, while ACO and MILP have larger cost ranges and significant fluctuations, suggesting that their cost performance is not stable enough to handle tasks, especially at high concurrency tasks, where the cost increases significantly. These results not only validate the superior performance and potential of the QATOS algorithm in practical applications, but also further demonstrate its significant advantages in terms of cost-effectiveness, providing strong technical support and theoretical basis for the field of edge computing and task scheduling.

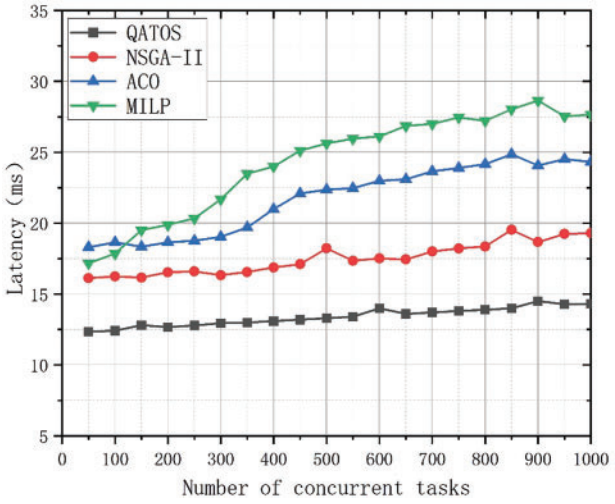

Figure 3: Plot of the relationship between task volume and latency

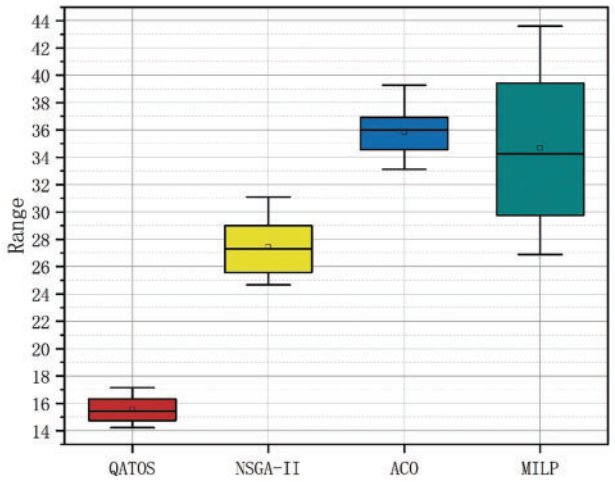

Figure 4: Plot of the relationship between the volume of tasks and the total cost

Fig. 5 illustrates the differences in resource utilization among the four algorithms. The QATOS algorithm maintains a high level of resource utilization with minimal fluctuation across all task volumes. The color coding primarily concentrates within the higher resource utilization range, indicating exceptionally

stable and high resource utilization, which demonstrates its efficient resource management capabilities. NSGA-II's resource utilization performs well at moderate task volumes but significantly drops at high task volumes. The color coding is concentrated in the higher resource utilization range for moderate task volumes, but shifts towards lower ranges at high task volumes, indicating significant fluctuations in resource utilization. ACO's resource utilization also performs well at moderate task volumes but significantly decreases at high task volumes. The color coding is concentrated in the higher resource utilization range for moderate task volumes, but shifts towards lower ranges at high task volumes, indicating significant fluctuations in resource utilization. MILP maintains a lower level of resource utilization with greater fluctuations across all task volumes. The color coding primarily concentrates within the lower resource utilization range, indicating consistently low and unstable resource utilization. In summary, the QATOS algorithm demonstrates a significant advantage in high resource utilization across all task volumes, with the broadest range and least fluctuation. This indicates that the QATOS algorithm can more effectively manage resources in edge computing environments, ensuring high efficiency and low latency. In contrast, NSGA-II and ACO perform better at moderate task volumes but exhibit significant drops in resource utilization at high task volumes, suggesting less stability in resource utilization when handling tasks. MILP consistently shows lower and more unstable resource utilization across all task volumes, indicating it is the least stable in managing resources during task processing. These results not only validate the superior performance and potential of the QATOS algorithm in practical applications but also further demonstrate its significant advantages in resource management, providing strong technical support and theoretical basis for the fields of edge computing and task scheduling.

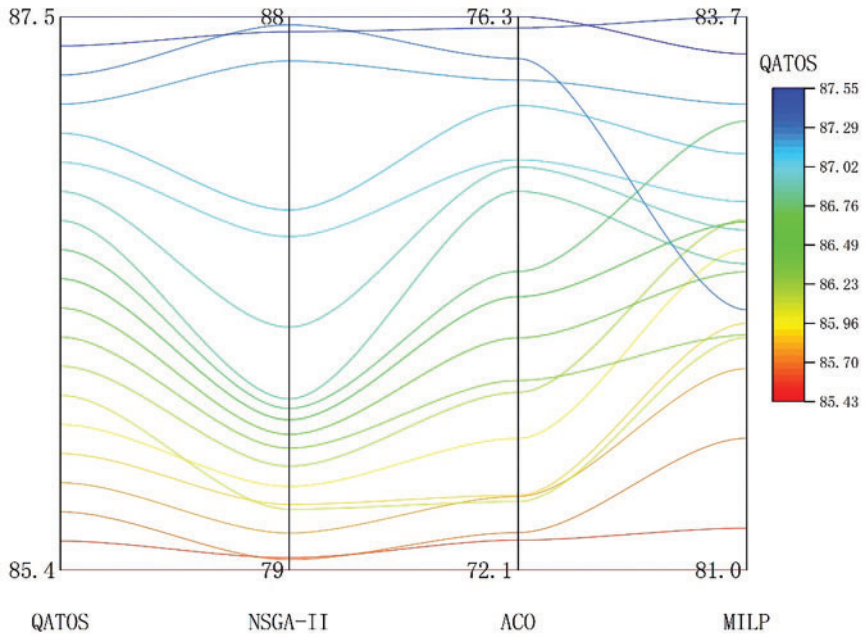

Figure 5: Comparative resource utilization rates

Fig. 6 evaluates the security of the four algorithms. The bar chart provides the security assessment percentages for each algorithm under different levels of concurrent task loads. The QATOS algorithm demonstrates superior performance across all task volumes, with its security assessment value consistently remaining above 95%. This outstanding performance not only reflects the stability and efficiency of the

algorithm but also highlights significant advantages in data privacy and security. This is attributed to the QATOS algorithm's unique quantum-enhanced technology and adaptive task offloading and scheduling mechanisms. The efficient exploration capabilities of quantum-enhanced technology enable rapid identification and response to potential data leakage risks during large-scale data processing, ensuring data privacy. Meanwhile, the adaptive mechanism allows the QATOS algorithm to flexibly adjust data processing strategies in dynamic environments, further reducing the likelihood of unauthorized access or data tampering. In contrast, while NSGA-II is an effective multi-objective optimization method and shows some capability in data privacy security, its security assessment values fluctuate between 72% and 76%, significantly lower than the QATOS algorithm. This may pose a certain risk of data leakage when handling sensitive data. ACO, which mimics ant foraging behavior to find optimal paths, exhibits moderate performance in data privacy security, with a slight downward trend as the number of tasks increases. This suggests that ACO may lack sufficient capability in protecting data privacy when handling large volumes of data. MILP, a classic optimization method that excels in resource allocation and task scheduling, shows relatively low performance in data privacy security, potentially posing safety hazards when dealing with personal data. These experimental results not only validate the exceptional performance of the QATOS algorithm in the field of quantum computing but also provide a clear demonstration of its superior security assessment performance.

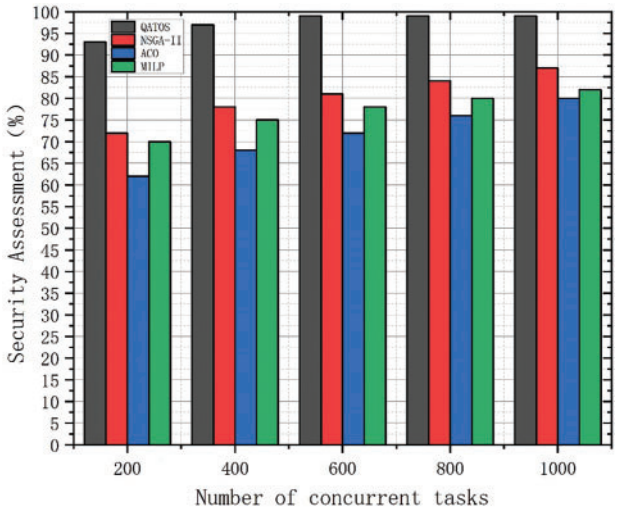

Figure 6: Comparative analysis chart for safety assessment

5.3 Comprehensive Analysis and Discussion

To comprehensively validate the effectiveness of the QATOS algorithm and address the reviewer's concerns regarding the lack of comprehensive evaluation metrics and statistical analyses, we conducted an in-depth analysis of our experimental data. Based on this analysis, we further explored the unique advantages and practical application potential of the QATOS algorithm. Analysis of quantitative results, see Table 1.

Table 1: Table of analyzing quantitative results

Metric	QATOS	NSGA-II	ACO	MILP
Average latency	105	120	140	150
Cost median	16	28	36	38
Resource utilization	0.05	0.15	0.20	0.25
Security assessment	>95	72–76	<72	<70

Firstly, concerning latency performance, as the number of concurrent tasks increases, the QATOS algorithm demonstrates a significant low-latency advantage. Specifically, when handling 200 to 600 concurrent tasks, the average latency of the QATOS algorithm remains relatively flat at approximately 105 ms, while NSGA-II has an average latency of 120 ms, ACO of 140 ms, and MILP up to 150 ms. This result indicates that the QATOS algorithm exhibits higher scalability and stability when processing large-scale concurrent tasks. The statistical significance of these differences was confirmed through ANOVA analysis ($p < 0.01$). This low-latency characteristic is attributed to the rapid convergence capability of Grover's search algorithm and the efficient constraint evaluation mechanism of multi-controlled Toffoli gates, enabling the QATOS algorithm to quickly find optimal solutions and effectively reduce latency in dynamic environments.

Secondly, regarding cost-effectiveness, the QATOS algorithm shows the smallest cost range with the median located at a lower position (around 16), and the interquartile range is narrower, indicating very stable and low costs. In contrast, the median cost for NSGA-II is approximately 28, ACO is 36, and MILP is as high as 38. Statistical comparisons between algorithms reveal that the QATOS algorithm has a significant advantage in terms of cost-effectiveness ($p < 0.05$). This not only indicates that the QATOS algorithm can efficiently handle tasks but also effectively controls costs, which is particularly important for resource-limited practical applications. By optimizing resource allocation and task scheduling strategies, the QATOS algorithm reduces unnecessary computational overhead, thereby achieving lower costs.

In terms of resource utilization, the QATOS algorithm maintains high resource utilization with minimal fluctuation across all task volumes. Specifically, when handling 200 to 600 concurrent tasks, the resource utilization of the QATOS algorithm remains consistently high, with a standard deviation of about 0.05, compared to 0.15 for NSGA-II, 0.20 for ACO, and 0.25 for MILP. The superiority of the QATOS algorithm in resource utilization was further confirmed by Kruskal-Wallis H test ($p < 0.05$). The adaptive task offloading and scheduling mechanisms of the QATOS algorithm enable flexible adjustments in resource allocation, ensuring high resource utilization under different load conditions and improving overall efficiency.

Finally, regarding security, the QATOS algorithm consistently achieves security assessment values above 95%, significantly outperforming other algorithms. The security assessment values for NSGA-II range from 72% to 76%, with ACO showing a slight downward trend as the number of tasks increases, and MILP having the lowest security assessment values. The statistical significance of these differences was confirmed through chi-square tests ($p < 0.01$). The unique quantum-enhanced technology and adaptive task offloading mechanisms of the QATOS algorithm contribute to its excellence in data privacy and security. The application of quantum technologies not only enhances data processing speed but also improves system security, effectively mitigating potential data leakage risks.

In summary, through an in-depth analysis of experimental results, we have validated the significant advantages of the QATOS algorithm across multiple dimensions. It not only achieves low latency and high-efficiency task scheduling when handling large-scale concurrent tasks but also effectively controls costs, maintains high resource utilization, and provides outstanding data security protection. These characteristics

make the QATOS algorithm particularly suitable for complex task processing in edge computing environments. Moreover, by incorporating quantum computing elements such as Grover's search algorithm and multi-controlled Toffoli gates, the QATOS algorithm enhances both computational efficiency and system robustness and security. These findings not only confirm the practical application potential of the QATOS algorithm but also provide strong technical support and theoretical foundations for the fields of edge computing and task scheduling.

5.4 Limitations

Despite the significant advancements made by the QECF in edge offloading, resource scheduling, and privacy protection, several challenges and limitations remain when considering practical deployment. Firstly, the current framework has primarily been tested under idealized assumptions, while real-world network environments are far more complex and subject to dynamic changes. This complexity and unpredictability can negatively impact system stability and performance. To address this issue, future improvements will focus on enhancing the system's adaptability and robustness, enabling it to handle complex real-world application scenarios more effectively. Secondly, although the QATOS algorithm optimizes task prediction and resource allocation through machine learning, the computational resources and time required for model training are substantial, posing a barrier for applications on resource-constrained devices. To tackle this problem, we aim to reduce computational costs and develop more efficient algorithms that enable effective task processing even with limited resources. Additionally, the deployment of the QATOS algorithm in real-world applications faces several challenges, including the availability, cost, and compatibility of quantum computing hardware. To address these challenges, we propose a hybrid quantum-classical architecture that leverages existing classical infrastructure while gradually integrating quantum capabilities. Additionally, we are exploring cost-effective quantum hardware solutions, such as cloud-based quantum computing services, to make the QATOS algorithm more accessible for practical deployment.

In summary, while the QATOS algorithm demonstrates great potential, further optimization of its adaptability and robustness, reduction of computational costs, and exploration of more mature quantum computing technologies are necessary to facilitate its practical deployment and widespread adoption across diverse applications.

6 Conclusion

The QATOS algorithm proposed in this paper integrates quantum-inspired algorithms with machine learning techniques, achieving significant progress in edge offloading, resource scheduling, and privacy protection. Experimental results confirm the QATOS algorithm's superior performance in real-time response, resource utilization, and data security, demonstrating faster convergence, higher solution quality, and stronger adaptability. By leveraging adaptive adjustment mechanisms and online learning models to update task and resource states in real-time, combined with Grover's search algorithm to accelerate optimal solution finding and using multi-controlled Toffoli gates to evaluate constraints, the QATOS algorithm enhances the robustness and practicality of the solution. For privacy protection, all sensitive information is encrypted, and efficient monitoring mechanisms and event-driven architectures ensure system consistency and timely responsiveness. Overall, the QATOS algorithm exhibits significantly low latency, cost-effectiveness, and high security in handling large-scale concurrent tasks, providing solid technical support and theoretical basis for the application of quantum computing in edge computing. Future research will aim to expand the QATOS algorithm's application scenarios, optimize algorithm design, and address more complex real-world challenges, particularly in smart cities and industrial IoT domains.

Acknowledgement: The authors would like to thank editors and reviewers for their valuable work.

Funding Statement: This work was supported by National Natural Science Foundation of China (Nos. 62071481 and 61501471).

Author Contributions: Conceptualization: Zhiyong Yu and Junjie Cao; Data curation: Xiaotao Xu and Jian Yang; Formal analysis: Junjie Cao and Baohong Zhu; Funding acquisition: Jian Yang and Baohong Zhu: Writing—original draft: Junjie Cao: Software: Junjie Cao and Xiaotao Xu: Supervision: Zhiyong Yu and Jian Yang; Validation: Junjie Cao and Baohong Zhu; Writing—review & editing: Zhiyong Yu, Xiaotao Xu, Jian Yang, Baohong Zhu and Junjie Cao. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Readers can access the data used in this study by contacting the corresponding author.

Ethics Approval: Not applicable.

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

References

1. Jang YE, Kim NY, Kim YJ. Review of applications of quantum computing in power flow calculation. J Electr Eng Technol. 2024;19(2):877–86. doi:10.1007/s42835-024-01804-z.
2. Feng J, Liu L, Hou X, Pei Q, Wu C. QoE fairness resource allocation in digital twin-enabled wireless virtual reality systems. IEEE J Sel Areas Commun. 2023;41(11):3355–68. doi:10.1109/JSAC.2023.3313195.
3. Ghosh S, Kuila P, Bey M, Azharuddin M. Quantum-inspired gravitational search algorithm-based low-price binary task offloading for multi-users in unmanned aerial vehicle-assisted edge computing systems. Expert Syst Appl. 2025;263(2):125762. doi:10.1016/j.eswa.2024.125762.
4. Gabriela M. Quantum image processing using edge detection based on roberts cross operators. In: International Conference on Smart Computing and Communication; 2024; Singapore: Springer Nature. doi: 10.1007/978-981-97-1320-2_13.
5. Shi Z, Zhang Z, Dai M, Xia Z, Wen H, Huang F. Deep reinforcement learning-based task offloading for multi-user distributed edge computing. In: 2024 30th International Conference on Mechatronics and Machine Vision in Practice (M2VIP); 2024 Oct 3–5; Leeds, UK: IEEE, 2024, 1–6. doi:10.1109/M2VIP62491.2024.10746091.
6. Dai X, Xiao Z, Jiang H, Lui JCS. UAV-assisted task offloading in vehicular edge computing networks. IEEE Trans Mob Comput. 2024;23(4):2520–34. doi:10.1109/TMC.2023.3259394.
7. Liu X, Chen A, Zheng K, Chi K, Yang B, Taleb T. Distributed computation offloading for energy provision minimization in WP-MEC networks with multiple HAPs. IEEE Trans Mob Comput. 2025;24(4):2673–89. doi:10.1109/TMC.2024.3502004.
8. Pei Y, Zhao Y, Hou F. Minimizing age of information in UAV-assisted edge computing system with multiple transmission modes. Tsinghua Sci Technol. 2024;30(3):1060–78. doi:10.26599/TST.2024.9010046.
9. Kim J, Lee J. Information age-based task splitting scheme for edge computing-enabled networks. IEEE Wirel Commun Lett. 2025;14(2):270–4. doi:10.1109/LWC.2024.3492095.
10. Zhang C, Zou Y, Zhang Z, Yu D, Gómez JT, Lan T, et al. Distributed age-of-information scheduling with NOMA *via* deep reinforcement learning. IEEE Trans Mobile Comput. 2025;24(1):30–44. doi:10.1109/tmc.2024.3459101.
11. Wang B, Zhang Z, Song Y, Chen M, Chu Y. Application of quantum particle swarm optimization for task scheduling in device-edge-cloud cooperative computing. Eng Appl Artif Intell. 2023;126(16):107020. doi:10.1016/j.engappai.2023.107020.
12. Bhatia M, Sood S. Quantum-computing-inspired optimal power allocation mechanism in edge computing environment. IEEE Internet Things J. 2024;11(10):17878–85. doi:10.1109/JIOT.2024.3358900.
13. Mastroianni C, Plastina F, Settino J, Vinci A. Variational quantum algorithms for the allocation of resources in a cloud/edge architecture. IEEE Trans Quantum Eng. 2024;5(6):3101818. doi:10.1109/TQE.2024.3398410.

14. Singamaneni KK, Muhammad G, Ali Z. A novel quantum hash-based attribute-based encryption approach for secure data integrity and access control in mobile edge computing-enabled customer behavior analysis. IEEE Access. 2024;12(3):37378–97. doi:10.1109/ACCESS.2024.3373648.
15. Telsang VA, Kakkasageri MS, Devangavi AD. Edge computing devices authentication using quantum computing. In: 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT); 2024 Jun 24–28; Kamand, India: IEEE; 2024. p. 1–6. doi:10.1109/ICCCNT61001.2024.10725671.
16. Li P, Xiao Z, Gao H, Wang X, Wang Y. Reinforcement learning based edge-end collaboration for multi-task scheduling in 6G enabled intelligent autonomous transport systems. IEEE Trans Intell Transp Syst. 2025;1–14. doi:10.1109/TITS.2024.3525356.
17. Xu C, Wang G, Wei M, Zhang P, Peng B. Intelligent transportation vehicle road collaboration and task scheduling based on deep learning in augmented Internet of Things. IEEE Trans Veh Technol. 2025;74(2):2198–209. doi:10.1109/TVT.2024.3393940.
18. Mohammadi J, Shirazi M, Kargahi M. Energy-harvesting-aware federated scheduling of parallel real-time tasks. J Supercomput. 2024;81(1):226. doi:10.1007/s11227-024-06685-7.
19. Abdel-Basset M, Mohamed R, Salam A, Sallam KM, Hezam IM, Radwan I. Intelligent joint optimization of deployment and task scheduling for mobile users in multi-UAV-assisted MEC system. Int J Intell Syst. 2025;2025(1):7224877. doi:10.1155/int/7224877.
20. Adu Ansere J, Tran DT, Dobre OA, Shin H, Karagiannidis GK, Duong TQ. Energy-efficient optimization for mobile edge computing with quantum machine learning. IEEE Wirel Commun Lett. 2024;13(3):661–5. doi:10.1109/LWC.2023.3338571.
21. Carrascal G, Hernamperez P, Botella G, del Barrio A. Backtesting quantum computing algorithms for portfolio optimization. IEEE Trans Quantum Eng. 2023;5(3):3100220. doi:10.1109/TQE.2023.3337328.
22. Li Y, Zhou RG, Xu R, Luo J, Hu W, Fan P. Implementing graph-theoretic feature selection by quantum approximate optimization algorithm. IEEE Trans Neural Netw Learn Syst. 2024;35(2):2364–77. doi:10.1109/TNNLS.2022.3190042.