Task Offloading and Edge Intelligence in Multi-Access Edge Computing: A Survey

www.surveyx.cn

Abstract

This survey paper provides a comprehensive analysis of task offloading and resource allocation strategies within Multi-Access Edge Computing (MEC) frameworks, emphasizing the optimization of system efficiency and performance metrics. It addresses the challenges of task offloading in mobile edge computing systems, focusing on non-divisible and delay-sensitive tasks under dynamic load conditions. The paper explores advanced optimization techniques, including deep reinforcement learning and game-theoretic approaches, to enhance resource allocation and task scheduling. Key findings highlight the integration of MEC with technologies such as 5G and UAVs, which significantly improve task offloading efficiency and reduce latency. The survey also examines the role of intelligent algorithms in optimizing resource utilization and enhancing edge intelligence, ensuring efficient machine learning task processing and real-time decision-making. Additionally, it discusses the importance of privacy and security considerations in edge computing, addressing the unique challenges posed by decentralized data processing. By offering insights into the latest innovations and challenges in MEC, the survey aims to stimulate further research and development in optimizing edge computing environments, ultimately enhancing system performance and user satisfaction.

1 Introduction

1.1 Significance of Task Offloading and Edge Computing

Task offloading and edge computing are essential in contemporary computational frameworks, effectively overcoming the limitations of traditional cloud models by positioning computational resources closer to data sources. This strategic relocation significantly reduces latency and enhances performance, particularly for latency-sensitive and computation-intensive IoT applications [1]. In mobile edge computing (MEC) systems, task offloading alleviates inter-cell interference and signal attenuation prevalent in traditional cellular networks, improving throughput and minimizing transmission disruptions [2].

The rapid proliferation of Internet of Things (IoT) devices demands efficient resource scheduling, as conventional cloud infrastructures often struggle to meet the real-time processing needs of these devices [3]. Task offloading reduces the computational load on mobile devices, which are typically constrained by hardware and energy limitations, thus optimizing energy consumption and minimizing processing delays [4]. MEC enhances the computing capabilities of wireless devices, providing significant advantages in task offloading for resource-intensive applications, such as video streaming [2].

In hybrid cloud and MEC environments, optimizing offloading and resource allocation is critical for improving energy consumption and latency, highlighting the need for innovative strategies [5]. Traditional static scheduling methods, such as Round-Robin and Priority Scheduling, are inadequate for managing dynamic workloads, necessitating adaptive workload management techniques [6]. The integration of fog computing with MEC further supports resource-constrained IoT devices by

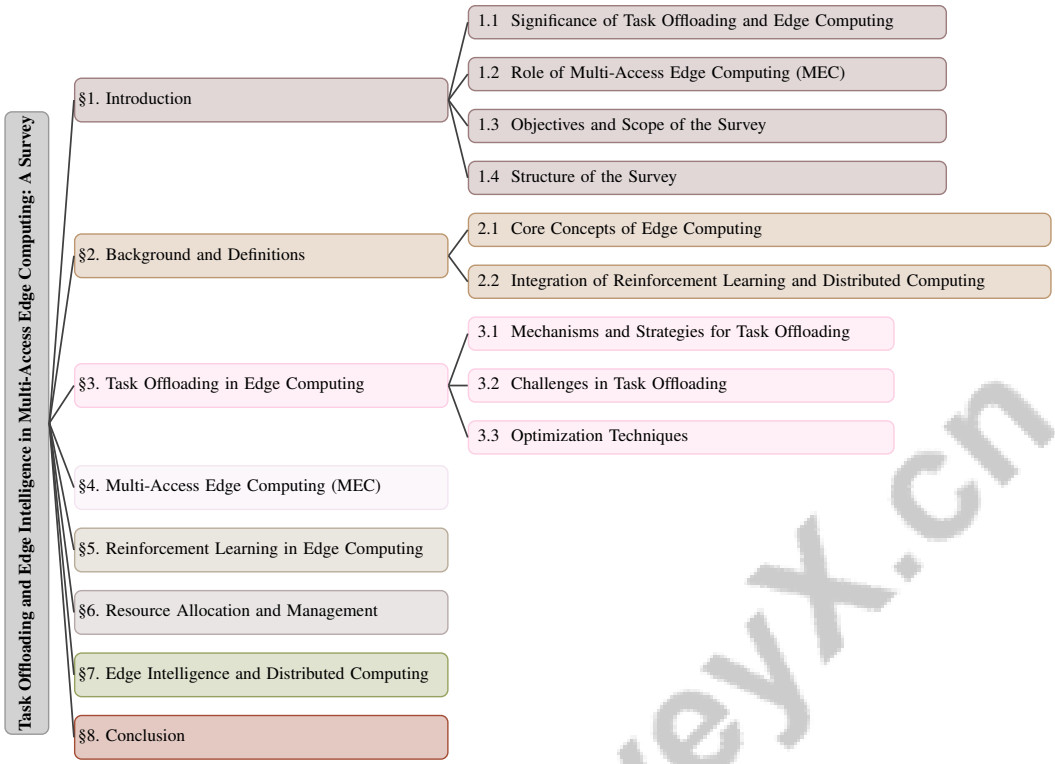


Figure 1: chapter structure

enabling task placement on edge or cloud servers, enhancing both communication and computation performance [7].

Task offloading is vital in MEC systems for optimizing resource allocation and improving user satisfaction by addressing varying delay sensitivities. It is essential for resource-limited edge devices executing complex machine learning models, enhancing performance while reducing costs [8]. MEC provides a promising solution for the challenges posed by delay-sensitive yet computationally intensive IoT applications, given the limited processing power of IoT devices [5].

In addition to performance enhancements, task offloading and edge computing contribute to reducing the carbon footprint of communication systems by optimizing resource usage in edge environments. These technologies facilitate efficient task processing, emphasizing the importance of task offloading in improving distributed computing efficiency. Effective resource management is crucial for meeting Quality-of-Service (QoS) requirements in fog computing environments, where fog nodes typically have limited, heterogeneous, and dynamic computing resources. This decentralized paradigm aims to reduce communication latencies and boost application performance by processing data closer to user devices such as smartphones and sensors [9, 10, 11, 12].

Task offloading and edge computing not only enhance resource allocation and efficiency but also enable the deployment of complex applications in mobile computing environments, addressing latency and resource constraint challenges. Recent advancements in edge computing systems, driven by the integration of computing, storage, and network resources at the network's edge, present significant research opportunities. These developments facilitate the rapid deployment of edge applications and provide users with comprehensive insights into various edge computing systems and tools. By analyzing current initiatives and offering a comparative overview of open-source tools, researchers can assist users in selecting suitable edge computing solutions tailored to specific applications, while addressing critical considerations such as energy efficiency, deep learning optimization, and sustainable development [13, 14].

1.2 Role of Multi-Access Edge Computing (MEC)

Multi-Access Edge Computing (MEC) significantly enhances computational capabilities by deploying small-scale data centers, known as MEC servers, that are sensitive to both radio and computing resources [2]. By positioning these resources closer to end-users, MEC mitigates high latency and low scalability issues associated with traditional cloud computing models. This strategic placement enables real-time processing and efficient management of finite computational resources, allowing compute-intensive workloads to be offloaded from IoT devices to local edge nodes, optimizing resource utilization and minimizing latency [4].

The integration of MEC with advanced communication technologies, such as 5G, amplifies its potential by enhancing task offloading efficiency and reducing latency. The efficiency of MEC systems relies on well-designed computation offloading policies that consider the characteristics of computation tasks and the dynamics of wireless channels. In vehicular edge computing (VEC) scenarios, MEC optimally leverages heterogeneous V2X communication technologies to support real-time vehicular applications [8].

Innovative approaches, such as the multi-task learning-based feedforward neural network (MTFNN) model, improve efficiency and accuracy in decision-making by simultaneously predicting offloading decisions and resource allocations [4]. Additionally, Unmanned Aerial Vehicles (UAVs) play a pivotal role in MEC environments, particularly in areas with limited processing capabilities, enhancing computational capabilities for task offloading [8].

MEC enhances a broad range of applications by effectively managing dynamic fluctuations in wireless channels and varying task arrival rates, which are crucial for maintaining optimal computational performance and ensuring timely processing of computationally intensive tasks. This capability is particularly significant in multi-cell wireless networks, where MEC servers facilitate task offloading and resource allocation, improving task completion times and reducing energy consumption for users. By employing advanced optimization techniques, MEC adapts to the diverse requirements of IoT devices and other applications, ensuring efficient use of limited computing resources at the network's edge [4, 15, 16]. By enabling mobile devices to offload tasks to nearby edge servers, MEC reduces latency and energy consumption, supporting an array of applications from real-time data processing to advanced robotic systems. The integration of MEC with collaborative frameworks, such as those involving UAVs, facilitates the distribution of computational tasks across multiple nodes, enhancing overall system capabilities and supporting emergent applications.

1.3 Objectives and Scope of the Survey

This survey provides an in-depth analysis of task offloading and resource allocation strategies within Multi-Access Edge Computing (MEC) frameworks, emphasizing system efficiency and adherence to stringent performance metrics. A primary objective is to address task offloading challenges in mobile edge computing systems, particularly for non-divisible and delay-sensitive tasks, while considering the dynamic load conditions of edge nodes [17]. Additionally, the survey aims to minimize the overall carbon footprint in edge computing networks through effective task scheduling, offloading, and battery management strategies [18].

The scope encompasses the optimization of offloading decisions and resource allocation in MEC systems, formulated as a mixed-integer nonlinear programming (MINLP) problem known for its NP-hard complexity [19]. This complexity presents significant challenges to traditional methods, which often struggle to achieve optimal solutions efficiently due to high computational demands and slow adaptability to environmental changes [20].

Furthermore, the survey explores innovative approaches to task offloading and resource allocation that meet the stringent delay and jitter requirements of intelligent applications [21]. It also examines the optimization of deep neural network (DNN) partitioning and resource allocation for multiple user equipments (UEs) in resource-constrained edge settings, focusing on reducing the maximum execution latency of DNN tasks [22].

In response to inefficiencies in scheduling computing tasks within edge AI systems, which can lead to unmet application constraints such as latency and resource underutilization, the survey investigates dynamic and distributed scheduling solutions [23]. Additionally, it assesses the efficient placement

of application tasks modeled as Directed Acyclic Graphs (DAGs) on suitable servers within fog computing environments, aiming to minimize execution time and energy consumption [24].

The survey also highlights task offloading mechanisms in fog computing, focusing on security and efficiency improvements through Deep Reinforcement Learning (DRL) [7]. Moreover, it underscores the importance of DRL-based task orchestrators, such as DeepEdge, in autonomously managing task offloading decisions [6].

By offering a comprehensive examination of these topics, the survey aspires to provide valuable insights and stimulate further research in optimizing edge computing environments. It ensures that the exploration of MEC is thorough and focused on the most pressing challenges and innovations in the field, aligning with the goal of optimizing task partitioning and parallel scheduling to enhance computational resource utilization and reduce processing delays [5].

1.4 Structure of the Survey

This survey is meticulously organized to provide a comprehensive understanding of task offloading and edge intelligence within Multi-Access Edge Computing (MEC) frameworks. The paper begins with an **Introduction** section, emphasizing the significance of task offloading and edge computing in modern computational paradigms, followed by an exploration of the role of MEC and the survey's objectives and scope.

The **Background and Definitions** section establishes foundational knowledge by defining core concepts related to edge computing and detailing the integration of reinforcement learning and distributed computing techniques, setting the stage for more complex discussions.

In the **Task Offloading in Edge Computing** section, the survey delves into mechanisms and strategies for task offloading, addressing the challenges and optimization techniques essential for efficient task distribution and processing. This section also considers the cooperative MEC network architecture, where edge nodes collaborate to minimize energy consumption while adhering to delay constraints [25].

The **Multi-Access Edge Computing (MEC)** section examines the architecture and functionalities of MEC, discussing its integration with existing infrastructures and the diverse applications it supports. This section introduces reconfigurable intelligent surfaces (RIS) to enhance transmission rates and minimize delays [26].

Reinforcement Learning in Edge Computing is analyzed for its role in optimizing resource allocation and task scheduling through deep reinforcement learning approaches, multi-agent reinforcement learning, and game-theoretic methods. The decentralized approach to task offloading using model-free deep reinforcement learning is particularly emphasized for its ability to enable independent decision-making by mobile devices [17].

The **Resource Allocation and Management** section explores strategies for efficient resource utilization in distributed computing environments, discussing the challenges and intelligent algorithms involved in dynamic resource management. It highlights the Successive Convex Approximation (SCA) method for optimizing resources while maintaining latency constraints [27].

In the **Edge Intelligence and Distributed Computing** section, the survey discusses the conceptual framework of edge intelligence and its significance in distributed computing, exploring how intelligent algorithms enhance processing capabilities and improve scalability and efficiency.

Finally, the **Conclusion** section summarizes the key findings and insights from the survey, discussing implications and future research directions, including privacy and security considerations as outlined in the structured approach to offloading decision algorithms and resource allocation [28].

By following this structured approach, the survey ensures a thorough exploration of the challenges and innovations in optimizing edge computing environments, providing valuable insights into task partitioning and parallel scheduling [22].The following sections are organized as shown in Figure 1.

2 Background and Definitions

2.1 Core Concepts of Edge Computing

Edge computing enhances computational frameworks by relocating data processing and storage closer to data sources, thereby optimizing response times and minimizing bandwidth usage [23]. This paradigm is especially beneficial for real-time applications like IoT systems and mobile augmented reality, which require rapid data processing and low-latency communication [29]. Task offloading, a key component of edge computing, involves transferring computation-intensive tasks from resource-constrained devices to more capable edge nodes. This process is essential for optimizing resource utilization and enhancing performance in latency-sensitive applications through effective task queuing and execution delay management [2, 30].

Multi-Access Edge Computing (MEC) extends traditional cloud computing capabilities by deploying localized data centers at the network edge, strategically designed to minimize latency and enhance service delivery [31]. In MEC, task offloading is a multi-objective optimization problem aimed at minimizing latency and energy consumption while maximizing network reliability [32]. The dynamic characteristics of wireless-powered MEC systems introduce additional complexities, necessitating effective energy allocation and resource management strategies [31]. Furthermore, MEC's integration with advanced technologies, such as collaborative UAV operations, significantly improves task offloading efficiency and reduces latency [33].

Optimizing resource allocation and cache placement within MEC is crucial for enhancing energy efficiency and reducing latency [30]. The heterogeneous nature of edge devices demands sophisticated management techniques to balance load, conserve energy, and effectively manage mobility [34]. In fog computing contexts, optimizing task offloading involves balancing efficiency, security, and resource allocation [7].

Challenges in MEC systems arise from high variability in user demand and server load, leading to unpredictable queuing delays [34]. Efficient resource allocation mechanisms are essential to ensure optimal matching between offloading tasks and available computational resources [35]. Innovative strategies, such as service caching placement and computation offloading, enhance efficiency and scalability, addressing the limitations of traditional methods [32].

Edge computing also requires the allocation of limited wireless resources to support simultaneous model training across diverse learning tasks, underscoring the need for efficient task offloading mechanisms [36]. In MEC-enabled vehicular networks, the demand for reliable low-latency computing services is particularly critical under high mobility conditions [37]. Multi-Tier Edge Computing (M-TEC) further optimizes resource allocation and task execution by distributing computational tasks across various nodes from the cloud to edge devices [29].

The core concepts of edge computing, including task offloading, MEC, and intelligent resource management, are integral to addressing contemporary computational challenges. These concepts facilitate the deployment of efficient, scalable, and responsive edge computing solutions that meet diverse application demands, ensuring energy efficiency and compliance with delay requirements in cooperative MEC networks [30]. The strategic distribution of computationally intensive tasks from resource-limited devices to more capable ones enhances system performance, as evidenced by various task offloading benchmarks [3]. This orchestration within dynamic environments, influenced by mobile user behavior, is a pivotal aspect of edge computing [6].

2.2 Integration of Reinforcement Learning and Distributed Computing

Integrating reinforcement learning (RL) and distributed computing within edge environments is crucial for addressing the complexities of resource allocation, task offloading, and adaptive decision-making. Reinforcement learning, particularly deep reinforcement learning (DRL), optimizes decision-making processes by dynamically adapting to network dynamics and environmental changes [37]. This adaptability is essential for managing the heterogeneity and limited computational capabilities of edge devices, as well as the dynamic nature of workloads [2].

A significant challenge in edge computing is the complexity of task dependencies, often represented as Directed Acyclic Graphs (DAGs), complicating the accurate prediction of state transitions necessary for optimal offloading decisions [38]. Utilizing DRL aids in learning optimal strategies for

task scheduling and resource allocation without precise environmental models, thus enhancing the efficiency of edge computing systems [39].

Innovative approaches, such as integrating DRL with blockchain technology, have been proposed to enhance the security and efficiency of task offloading in fog computing environments [7]. This integration ensures secure and reliable task execution, which is vital in distributed networks where data privacy and integrity are critical.

Furthermore, combining graph neural networks with reinforcement learning provides a robust method for enhancing task offloading in dynamic environments, leveraging network topology structural information to improve decision-making [8]. The incorporation of emerging technologies like massive MIMO, IRS, and edge AI within a multi-tier computing framework further augments existing task offloading mechanisms, yielding significant performance and efficiency improvements [21].

The development of standardized simulation environments, such as PeersimGym, facilitates the testing and optimization of task offloading strategies using RL in edge computing systems [3]. These benchmarks offer a customizable platform for evaluating the effectiveness of various RL-based strategies in real-world scenarios.

The synergy between reinforcement learning and distributed computing within edge environments enhances the capability to efficiently process data and reduce latency. By employing sophisticated optimization techniques and advanced algorithms, this integration fosters the creation of adaptive and scalable edge computing solutions tailored to contemporary application requirements. These solutions ensure optimal performance in dynamic and resource-constrained environments and address critical challenges such as job scheduling, resource allocation, and task offloading, which are vital for enhancing system efficiency in both edge and cloud computing contexts. The convergence of edge computing with artificial intelligence further enhances data processing capabilities at the network edge, enabling real-time responses and improving reliability for low-latency applications [40, 41].

3 Task Offloading in Edge Computing

To effectively understand the complexities and innovations associated with task offloading in edge computing, it is essential to first explore the various mechanisms and strategies that underpin this critical process. These strategies not only facilitate the efficient distribution of computational tasks but also address the unique challenges posed by the dynamic nature of edge environments. Table 1 provides a comparative analysis of various optimization methods employed in task offloading within edge computing environments, illustrating their respective approaches to resource allocation and adaptability. Figure 2 illustrates the hierarchical structure of task offloading in edge computing, encompassing mechanisms and strategies, challenges, and optimization techniques. This figure highlights collaborative and decentralized approaches, addresses complex optimization problems and dynamic network conditions, and showcases advanced learning algorithms and heuristic methods for enhancing efficiency and adaptability in edge environments. The following subsection delves into the specific mechanisms and strategies for task offloading, highlighting their significance in optimizing resource utilization and enhancing overall system performance.

3.1 Mechanisms and Strategies for Task Offloading

Task offloading in edge computing is a critical mechanism for optimizing resource utilization and enhancing system performance, especially in environments characterized by limited computational resources and dynamic network conditions. Various strategies have been developed to efficiently distribute and process tasks, addressing the complexities inherent in edge environments. A significant method involves task partitioning, scheduling, and collaborative computing among edge servers, which minimizes service latency and improves reliability [31]. This collaborative approach is essential for managing the diverse and dynamic workloads typical of edge computing scenarios.

Advanced learning algorithms play a significant role in optimizing task offloading. The proposed Energy-Efficient Joint Offloading and Wireless Resource Allocation Strategy (EEJS) optimizes task offloading decisions and resource allocation to minimize total energy consumption, demonstrating the potential of integrating task offloading with resource management [2]. Similarly, the Joint Task Offloading and Resource Allocation (JTO-RA) method optimally allocates resources and determines

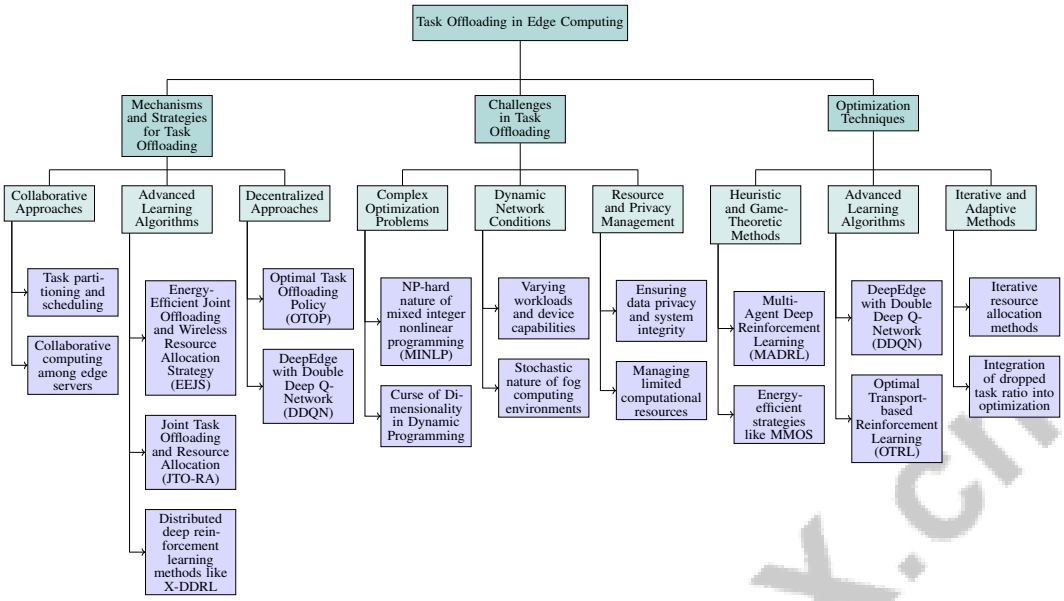


Figure 2: This figure illustrates the hierarchical structure of task offloading in edge computing, encompassing mechanisms and strategies, challenges, and optimization techniques. It highlights collaborative and decentralized approaches, addresses complex optimization problems and dynamic network conditions, and showcases advanced learning algorithms and heuristic methods for enhancing efficiency and adaptability in edge environments.

task offloading strategies for IoT devices, highlighting the importance of adaptive strategies in dynamic environments [35].

Optimized task distribution is further enhanced by distributed deep reinforcement learning methods like X-DDRL, which utilizes IMPortance weighted Actor-Learner Architectures (IMPALA) for better exploration and faster convergence [24]. This innovative approach addresses a previously untackled problem space, ensuring efficient task allocation and resource utilization across edge nodes.

Decentralized approaches also contribute significantly to task offloading strategies. The Optimal Task Offloading Policy (OTOP) is designed to minimize expected time-average costs in task offloading by evaluating a finite subset of states in the Dynamic Programming equation [42]. Similarly, DeepEdge utilizes a Double Deep Q-Network (DDQN) algorithm to model task orchestration as a Markov process for effective decision-making [6].

The integration of reinforcement learning with task offloading is exemplified by the use of Q-learning to continuously optimize scheduling strategies based on real-time feedback from the system's state, thus improving performance and resource management [34]. This method enhances the efficiency of task distribution by dynamically adapting to network conditions and resource availability.

In multi-tier computing environments, identifying computational tasks that need to be offloaded and determining the appropriate node (end-user, fog/edge, or cloud) for task execution is critical for optimizing resource allocation and minimizing latency [29]. The decentralized approach to managing task offloading and resource allocation further enhances overall system performance by dynamically adapting to changing network conditions [23].

These diverse strategies and algorithms underscore the complexity and innovation within task offloading in edge computing. By leveraging advanced learning techniques and optimization frameworks, these approaches enhance the efficiency, scalability, and adaptability of edge computing systems, ultimately improving system performance and user satisfaction [7]. The introduction of customizable simulation platforms like PeersimGym further supports the development and testing of these strategies, allowing for flexible modeling of various network topologies and task parameters [3].

As shown in Figure 3, this figure illustrates the hierarchical categorization of task offloading strategies in edge computing, highlighting collaborative strategies, learning algorithms, and decentralized

approaches as primary categories. Each category encompasses specific methods and strategies that optimize task distribution, resource allocation, and system performance, showcasing the complexity and innovation in edge computing environments. The illustrative examples provided in the figures highlight two key aspects of task offloading. The first figure demonstrates a networked system where tasks are strategically offloaded from a server to an airplane, showcasing a real-world scenario of offloading and task scheduling. This system includes multiple servers, a base station, and a buffer managing tasks with specific deadlines, emphasizing the importance of timely and efficient task management. The second figure offers a comparative analysis of various algorithms in a multi-user environment, illustrating how different strategies perform under varying user loads. This graph provides insights into the effectiveness of algorithms like Relax, Whittle, EDF, LST, and Greedy, each depicted with distinct markers and colors, across a user range from 20 to 80. Together, these examples underscore the critical role of well-designed mechanisms and strategies in optimizing task offloading within edge computing frameworks [42, 43].

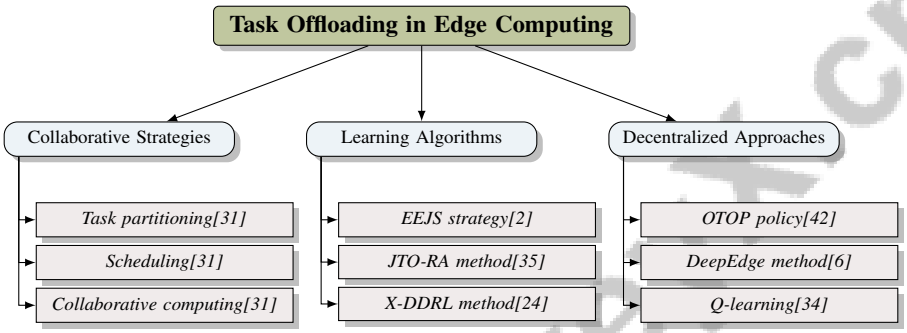


Figure 3: This figure illustrates the hierarchical categorization of task offloading strategies in edge computing, highlighting collaborative strategies, learning algorithms, and decentralized approaches as primary categories. Each category encompasses specific methods and strategies that optimize task distribution, resource allocation, and system performance, showcasing the complexity and innovation in edge computing environments.

3.2 Challenges in Task Offloading

Task offloading in edge computing environments is fraught with numerous challenges, stemming from the inherent complexity and dynamic nature of these systems. A significant challenge is the NP-hard nature of the mixed integer nonlinear programming (MINLP) problem, which complicates the joint optimization of task offloading, packet scheduling, and resource allocation . This complexity is further exacerbated by the need to minimize total system costs while adhering to constraints related to task delays and available resources [4].

The dynamic nature of edge environments, characterized by varying workloads and device capabilities, poses significant obstacles to effective task scheduling. Existing methods often lack adaptability to handle uncertainties such as UAV failures and dynamic network conditions, which are prevalent in heterogeneous edge computing scenarios [8]. This variability is compounded by the stochastic nature of fog computing environments, where current heuristic methods struggle to adapt to constant changes and complexities [3].

Another critical challenge is the inability of traditional rule-based orchestrators to effectively manage the dynamic nature of edge computing environments, leading to suboptimal performance and resource utilization [6]. The 'Curse of Dimensionality' in the Dynamic Programming approach further complicates task offloading, making computations intractable due to the infinite state space [42].

Ensuring data privacy, maintaining system integrity, and managing the limited computational resources of fog nodes are additional challenges that must be addressed to optimize task offloading strategies. The NP-hard complexity of the parallel scheduling problem, particularly when considering specific task partitioning actions, further hampers the effectiveness of conventional optimization methods [5].

Existing benchmarks face challenges in scalability, flexibility, and the ability to accommodate diverse task offloading scenarios. PeersimGym aims to overcome these by providing high configurability and

support for multi-agent reinforcement learning, thereby enhancing the adaptability and scalability of task offloading strategies [3]. Addressing these challenges is crucial for optimizing resource utilization and reducing latency in edge computing environments, necessitating the development of innovative algorithms and approaches to enhance the scalability, performance, and adaptability of task offloading strategies.

3.3 Optimization Techniques

Optimization techniques in task offloading processes are pivotal for enhancing resource utilization and minimizing latency in edge computing environments. These techniques address dynamic network conditions and limited computational resources through a variety of strategies. A notable method involves the use of heuristic optimization combined with game-theoretic principles to achieve near-optimal performance in task offloading. The Multi-Agent Deep Reinforcement Learning (MADRL) approach, as utilized in UCMEC, effectively integrates convex optimization to manage the complexities of task offloading and resource allocation, ensuring efficient task distribution across edge nodes [32].

The MMOS algorithm exemplifies an energy-efficient task offloading and resource allocation strategy for multi-user MEC systems, optimizing resource allocation while minimizing energy consumption [44]. Similarly, the Joint Task Offloading and Resource Allocation (JTO-RA) method transforms non-convex optimization problems into solvable convex ones, enabling efficient resource allocation strategies that adapt to dynamic environments [35].

Advanced learning algorithms significantly enhance task offloading optimization. DeepEdge, employing a Double Deep Q-Network (DDQN) algorithm, autonomously learns and adapts to network conditions, optimizing task orchestration through real-time feedback [6]. The Dynamic Distributed Scheduler, proposed by Hu et al., dynamically adjusts scheduling decisions based on the current system state, contrasting with static or centralized approaches, thereby enhancing adaptability and efficiency [23].

The Optimal Transport-based Reinforcement Learning (OTRL) approach highlights the adaptability of offloading policies based on real-time data and interactions, leveraging the strengths of both optimal transport and reinforcement learning to optimize task offloading decisions [38]. This adaptability is crucial for maintaining service quality in dynamic edge computing environments.

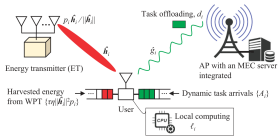
Iterative resource allocation methods, such as those proposed by Tang et al., iteratively adjust resource allocations to minimize the maximum latency experienced by any user equipment (UE) during deep neural network (DNN) execution, thereby enhancing system performance and user satisfaction [22].

The integration of the dropped task ratio into the optimization process, as explored by Moshiri et al., provides a comprehensive evaluation of system performance, addressing the trade-offs between resource allocation efficiency and system reliability [45].

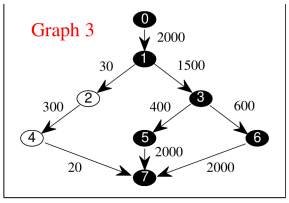
By employing advanced optimization techniques, edge computing environments can significantly enhance their performance, scalability, and resource efficiency, addressing critical challenges such as job scheduling, resource allocation, and task offloading, which are essential for meeting the low-latency and high-reliability demands of real-time applications. These techniques leverage computational intelligence to solve complex optimization problems that traditional methods struggle with, ultimately enabling more effective management of decentralized resources at the network edge. [40, 13, 46, 11]. These methods address the complexities of dynamic network conditions and resource constraints, ensuring efficient task processing and optimal resource utilization while maintaining high service quality and user satisfaction.

As shown in Figure 4, The example of "Task Offloading in Edge Computing; Optimization Techniques" is illustrated through three distinct scenarios, each highlighting different aspects of optimization in edge computing environments. The first scenario, "5G-based Multi-User Offloading for Real-Time Applications," presents a network setup where multiple devices, such as smartphones and servers, are interconnected via a 5G network. This setup is designed to handle real-time applications by categorizing devices based on urgency levels, thereby optimizing the offloading process to ensure efficient resource allocation and timely task execution. The second scenario, "Energy Harvesting and Offloading in a Wireless Network," focuses on the integration of energy harvesting mechanisms within a wireless network. Here, energy is harvested from a wireless power

(a) 5G-based Multi-User Real-Time Offloading for Applications[47]



(b) Energy Harvesting and Offloading in a Wireless Network[48]



(c) Graph 3[49]

Figure 4: Examples of Optimization Techniques

transfer system to power user devices, with the energy transfer dynamics captured by specific mathematical expressions. This approach highlights the potential for sustainable energy management in edge computing. Lastly, "Graph 3" provides a visual representation of a directed graph with weighted nodes, demonstrating the application of graph theory in optimizing task offloading. Each node and its associated weight reflect the computational load and resource requirements, offering insights into how tasks can be strategically distributed across the network to enhance performance. Together, these examples underscore the diverse optimization techniques employed in edge computing to improve efficiency, energy utilization, and real-time processing capabilities. [**?**]moshiri2025jointoptimizationcompletionratio,wang2020optimalenergyallocationtask,dilorenzo2016jointoptimizationradioresour

Feature	**EEJS**	**JTO-RA**	**X-DDRL**
Optimization Method	Energy-efficient	Joint Optimization	Deep Reinforcement
Adaptability	Dynamic Environments	Adaptive Strategies	Faster Convergence
Resource Allocation	Wireless Resources	Iot Devices	Distributed Nodes

Table 1: Comparison of Optimization Methods for Task Offloading in Edge Computing, detailing the specific optimization method, adaptability, and resource allocation focus of each approach. The table highlights the strengths of the Energy-Efficient Joint Offloading and Wireless Resource Allocation Strategy (EEJS), Joint Task Offloading and Resource Allocation (JTO-RA), and Distributed Deep Reinforcement Learning (X-DDRL) in addressing the challenges of dynamic environments and efficient resource management.

4 Multi-Access Edge Computing (MEC)

Multi-Access Edge Computing (MEC) represents a paradigm shift in digital ecosystems, enhancing computational efficiency and service delivery by situating computing, storage, and intelligence resources near end users. This proximity reduces latency and improves the quality of experience (QoE) for applications, addressing mobile device limitations such as battery life and processing capacity. MEC facilitates real-time processing, computation offloading, and decentralized resource allocation to meet stringent performance requirements [50, 51, 52]. Understanding MEC architecture and its integration within existing network infrastructures is crucial for achieving contemporary applications' low-latency and high-efficiency goals.

4.1 MEC Architecture and Integration

The MEC architecture enhances computational capabilities by strategically positioning resources near end users, optimizing resource allocation and task offloading through a hierarchical model of end-user devices, edge nodes, and centralized cloud resources [2]. This integration facilitates efficient data processing and task execution close to data sources, reducing delays and improving response times [1]. Key to MEC architecture is dynamic resource allocation, exemplified by the Energy-Efficient Joint Offloading and Wireless Resource Allocation Strategy (EEJS), which uses a bi-level optimization framework for power and subcarrier allocation alongside task offloading [2]. Intelligent Reflecting Surface (IRS) technology further enhances wireless communication within MEC frameworks [8].

AI-generated, for reference only.

Collaborative frameworks, such as distributed deep reinforcement learning, significantly improve service reliability and latency reduction. The X-DDRL framework utilizes a pre-scheduling phase for task prioritization and a distributed actor-learner structure to learn optimal policies, facilitating efficient task offloading and resource utilization [24]. Decentralized approaches, including the Optimal Task Offloading Policy (OTOP), enhance MEC efficiency by making task offloading decisions based on current system states and available resources [42]. The OTPPS framework integrates deep reinforcement learning with graph theory to optimize task partitioning and scheduling [5].

Methods like the MECNC approach use Lyapunov optimization for dynamic resource allocation in multi-hop MEC networks [1], supporting local processing and enabling edge devices to offload tasks effectively, alleviating the computational burden on centralized cloud resources [2]. MEC's adaptability is further exemplified by integrating advanced techniques such as deep learning, enhancing caching strategies and resource allocation, leading to improved energy efficiency and system performance [4]. Utilizing personal devices for computation in Multi-Tier Edge Computing (M-TEC) fosters peer-to-peer offloading, reducing reliance on commercial edge servers [29].

Overall, MEC architecture represents a sophisticated framework characterized by advanced algorithms, decentralized processing techniques, and multi-tiered structures. This interplay is essential for optimizing resources at the network edge, enabling efficient computation offloading, and ensuring real-time processing capabilities for resource-intensive applications. By leveraging local computing and storage resources, MEC enhances user experiences while addressing mobile devices' limitations, particularly in emerging applications like virtual reality and real-time data analytics [50, 52, 53, 54, 51].

As illustrated in Figure 5, the hierarchical structure of MEC architecture and integration is pivotal in enhancing modern digital infrastructures, particularly in smart city environments. The images visually explore MEC architecture and its integration into various technological landscapes. The first image highlights edge computing's role in urban settings, emphasizing the synergy between mobile edge computing nodes and IoT devices within city infrastructures. This image effectively delineates how edge computing facilitates seamless connectivity and processing power at the network's edge, crucial for real-time applications in smart cities. The second image presents a complex network diagram, symbolizing the intricate web of processes and connections inherent in MEC systems, underscoring the dynamic flow of information across multiple nodes. Lastly, the "Offloading Advantages" diagram succinctly maps out the benefits of computation offloading, optimizing resource usage by reducing transmission latency, conserving mobile device battery life, minimizing redundancy, and lowering transmission costs. Together, these examples illustrate MEC's transformative potential in creating more responsive, efficient, and sustainable digital ecosystems, as they encompass the core aspects of resource optimization, collaborative frameworks, and advanced techniques such as IRS technology, Lyapunov optimization, and M-TEC integration [52, 54, 55].

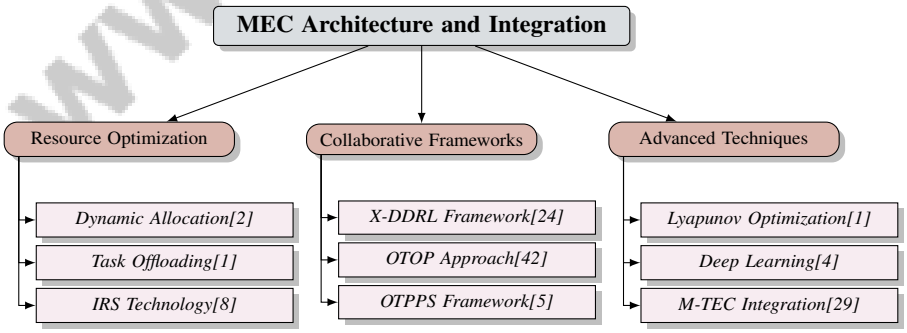


Figure 5: This figure illustrates the hierarchical structure of MEC architecture and integration, focusing on resource optimization, collaborative frameworks, and advanced techniques. Resource optimization involves dynamic allocation, task offloading, and IRS technology. Collaborative frameworks include the X-DDRL framework, OTOP approach, and OTPPS framework. Advanced techniques cover Lyapunov optimization, deep learning, and M-TEC integration.

4.2 Applications and Services Supported by MEC

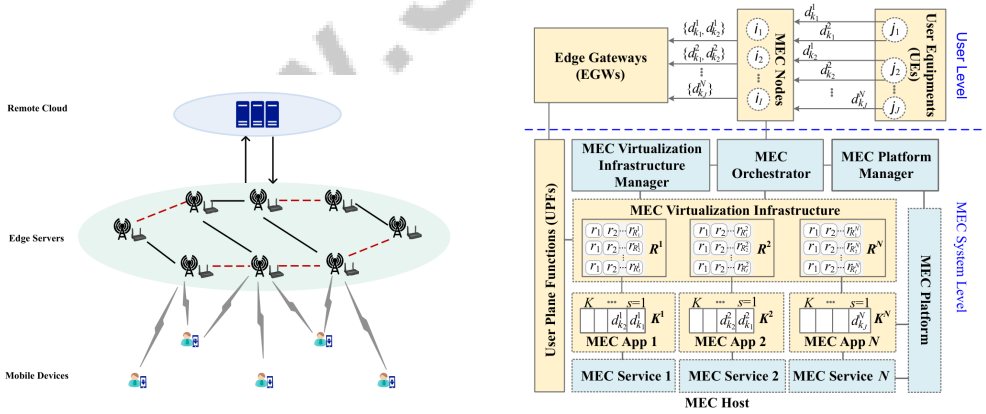
Multi-Access Edge Computing (MEC) supports a wide array of applications and services by leveraging its proximity to end users, enhancing computational efficiency and reducing latency. The integration of MEC with existing recognition frameworks, such as AFEM, exemplifies its applicability across diverse domains, facilitating real-time data processing and decision-making [56]. This capability is particularly beneficial for applications requiring rapid response times and minimal latency, such as augmented reality (AR), virtual reality (VR), and autonomous driving systems.

The architectural design of MEC systems underscores their adaptability and performance in real-world scenarios, effectively managing computational tasks and optimizing resource allocation [14]. This adaptability is crucial for supporting the dynamic requirements of Internet of Things (IoT) applications, which demand efficient data processing and swift communication to enhance user experiences [57]. By integrating edge computing with IoT, MEC significantly improves response times, ensuring seamless interaction between devices and users.

In artificial intelligence (AI), MEC enhances resource allocation efficiency through advanced techniques such as reinforcement learning and federated learning [53]. These AI-driven approaches optimize the distribution of computational resources across distributed nodes, ensuring efficient task processing. The cooperative use of nodes in MEC environments further enhances system efficiency, as nodes assist in computation and transmission [58].

MEC also addresses the critical need for efficient GPU resource management, essential for supporting compute-intensive applications such as deep learning and high-performance computing [10]. Current studies highlight the ongoing need for innovative solutions to effectively manage GPU resources, ensuring that MEC systems can support a wide range of applications without compromising performance.

MEC's capability to support diverse applications and services highlights its sophisticated architecture and advanced resource management features, facilitating real-time processing, computation offloading, and improved user experience by bringing computational and storage resources closer to end users. This architecture addresses mobile devices' limitations and enhances performance in latency-sensitive scenarios [50, 51, 52]. By integrating cutting-edge technologies and optimizing resource allocation, MEC enhances the performance and efficiency of modern computational environments, making it an indispensable component of future digital ecosystems.



(a) Cloud Computing Network Architecture[52] (b) MEC System Architecture[59]

Figure 6: Examples of Applications and Services Supported by MEC

As depicted in Figure 6, in the rapidly evolving landscape of network computing, Multi-Access Edge Computing (MEC) emerges as a transformative technology, offering enhanced capabilities for applications and services by bringing computational resources closer to end users. The concept of MEC is vividly illustrated through two distinct architectural models: the traditional Cloud Computing Network Architecture and the innovative MEC System Architecture. The former highlights a network where edge servers link mobile devices to the cloud through wireless networks, emphasizing high-speed connectivity and reduced latency, crucial for optimizing performance in real-time applications.

In contrast, the MEC System Architecture delineates a more localized framework, comprising Edge Gateways, MEC Nodes, and User Equipments. Here, the Edge Gateways facilitate connectivity to MEC Nodes, pivotal in executing services and managing infrastructure, ultimately enhancing application delivery by minimizing latency and improving efficiency. This dual architectural depiction underscores MEC's pivotal role in supporting a diverse array of applications and services, revolutionizing data processing and delivery in contemporary network environments [52, 59].

5 Reinforcement Learning in Edge Computing

Advancements in optimizing resource allocation and task management within edge computing environments have increasingly relied on Deep Reinforcement Learning (DRL) techniques. These methodologies enhance decision-making capabilities and address the complexities of resource-constrained settings. The subsequent subsection delves into various DRL models, showcasing their efficacy in elevating performance and efficiency in edge computing systems.

5.1 Deep Reinforcement Learning Approaches

Deep Reinforcement Learning (DRL) models are pivotal in refining resource allocation and task scheduling in edge computing. By employing sophisticated algorithms, these models tackle the challenges of dynamic and resource-limited systems, thereby refining decision-making processes. DRL's integration with task partitioning and scheduling reduces processing delays and ensures fairness among users [5]. Methods such as the Multi-Task Feedforward Neural Network (MTFNN) utilize historical data to predict and adapt to evolving network conditions, optimizing resource management [4].

DRL also enhances task offloading efficiency and security, particularly through its integration with blockchain technology, ensuring secure task execution in distributed networks [7]. Moreover, graph neural networks within reinforcement learning frameworks improve task processing decisions in uncertain environments by leveraging structural network data [8].

Benchmarking of DRL models, like the Double Deep Q Network (DDQN) and Advantage Actor-Critic (A2C), reveals their superior performance in task offloading optimization compared to traditional models [3]. Additionally, the application of Lyapunov control theory within distributed algorithms complements DRL by dynamically adjusting resource allocation in response to network changes [1]. This adaptability is vital for maintaining service quality in dynamic edge environments.

Future research should aim at developing adaptive algorithms that dynamically respond to variations in task arrival and processing capabilities, further enhancing system efficiency [42]. DRL models provide a robust framework for navigating the complexities of edge computing, including resource management and real-time data processing, optimizing performance while ensuring data integrity in dynamic environments [41, 13, 60, 61, 62]. By employing advanced learning algorithms, DRL approaches offer efficient, scalable, and adaptive solutions that meet the diverse demands of modern applications, ultimately enhancing the performance of edge computing systems.

5.2 Multi-Agent Reinforcement Learning (MARL)

Multi-Agent Reinforcement Learning (MARL) significantly advances edge intelligence and decision-making in edge computing environments. By employing multiple agents that learn and adapt through interactions with their environment and each other, MARL optimizes task offloading and resource allocation strategies, addressing the dynamic complexities of edge networks. The Multi-Agent Deep Deterministic Policy Gradient (MADDPG) method exemplifies this approach, facilitating optimal task management in vehicular edge computing scenarios [63].

As illustrated in Figure 7, MARL's application in edge computing focuses on task offloading optimization, decentralized decision-making, and adaptability and robustness. This figure highlights key methods and challenges addressed by MARL, emphasizing its critical role in enhancing edge intelligence and decision-making processes.

MARL's decentralized framework enables scalable decision-making processes, crucial for managing diverse conditions in edge computing. Agents operate independently and collaboratively within a blockchain-based MEC framework, optimizing local decisions related to task offloading, channel

selection, and resource management. This cooperative approach enhances individual agent performance and overall system utility, as demonstrated by a novel multi-agent deep reinforcement learning algorithm addressing latency issues through a Proof-of-Reputation consensus mechanism [64, 52]. Such decentralized decision-making is particularly beneficial in high-mobility environments like autonomous vehicles and IoT devices.

MARL integration with edge computing not only improves task offloading efficiency but also enhances adaptability and robustness. By continuously learning from dynamic environments, MARL equips edge devices to respond swiftly to network fluctuations and resource availability changes. This adaptability is crucial for optimizing performance in edge computing scenarios, especially in applications requiring high Quality of Service (QoS), such as augmented reality and online gaming [41, 65, 62, 52].

MARL fosters coordination among multiple agents, significantly improving resource management and task distribution. This is evident in mobile edge computing, where agents optimize computation and communication resources, increasing efficiency and reducing latency. MARL frameworks also facilitate the development of emergent communication protocols, enhancing decision-making in complex environments like hybrid cloud and edge computing systems [33, 66, 64, 67]. This cooperative behavior is vital for achieving global optimization objectives, such as minimizing latency and energy consumption while maximizing throughput and service quality.

The application of MARL in edge computing significantly enhances edge intelligence capabilities, providing a robust framework for dynamic decision-making. By leveraging the collective learning capabilities of multiple agents, MARL enhances the scalability and efficiency of edge computing systems. This is crucial for meeting the demands of modern applications, such as dynamic multimedia streaming and industrial IoT tasks, while managing challenges like tail latency and resource constraints. Through advanced techniques, including emergent communication protocols and AI-driven scheduling methods, MARL ensures that edge computing systems maintain high Quality of Service (QoS) standards [41, 65, 62, 67].

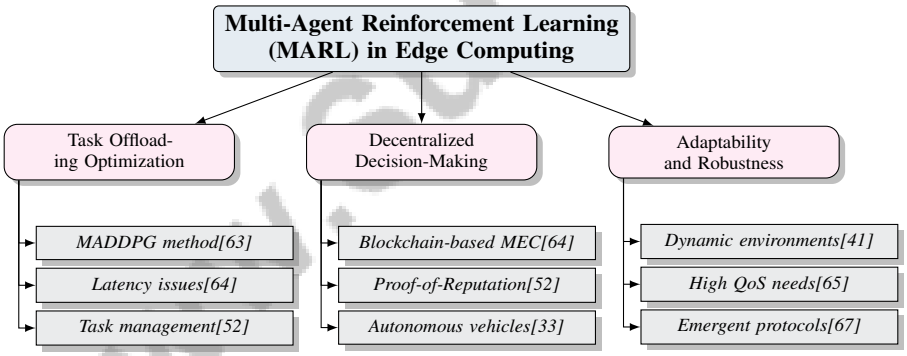


Figure 7: This figure illustrates the application of Multi-Agent Reinforcement Learning (MARL) in edge computing, focusing on task offloading optimization, decentralized decision-making, and adaptability and robustness. It highlights key methods and challenges addressed by MARL, emphasizing its role in enhancing edge intelligence and decision-making processes.

5.3 Game-Theoretic and Hybrid Approaches

Game-theoretic and hybrid approaches in reinforcement learning offer innovative solutions for optimizing task offloading and resource allocation in edge computing. These approaches utilize game theory to model complex interactions among decision-makers, such as edge servers and terminal entities in MEC environments. This framework facilitates the analysis of strategic behaviors and optimizes resource allocation among competing agents, addressing challenges like task allocation and conflicting interests [68, 69].

The integration of game-theoretic concepts with reinforcement learning, as seen in the Evolutionary Multi-Objective Reinforcement Learning (EMORL) framework, highlights the potential of hybrid approaches. EMORL effectively addresses the Task Completion Time Optimization (TCTO) problem by considering multiple objectives, such as latency and energy consumption, while adapting to

dynamic conditions [70]. This adaptability is crucial for maintaining optimal performance in edge computing systems, where network conditions change rapidly.

Hybrid approaches combining reinforcement learning with game-theoretic models offer advantages in scalability and flexibility. By integrating evolutionary algorithms with reinforcement learning, these approaches adaptively modify task offloading strategies based on real-time feedback. This dynamic adjustment enhances decision-making robustness and improves task offloading efficiency in complex systems, such as UAV-assisted mobile edge computing, where objectives like minimizing task delay and energy consumption must be balanced. These methods leverage multi-objective optimization capabilities, generating multiple optimal policies in a single run, significantly outperforming traditional algorithms [70, 34, 71].

Game-theoretic and hybrid approaches not only enhance resource allocation and task scheduling but also improve overall system resilience. By fostering cooperation and competition among agents, these approaches enable edge devices to optimize performance, ensuring efficient resource utilization and improved service quality. Cooperative behavior among edge resources is vital for achieving global optimization goals, such as reducing latency and energy consumption while enhancing throughput and user satisfaction. This is particularly important in MEC, where resources are positioned closer to end users to enhance QoE through real-time processing. Collaborative approaches, employing frameworks like Stackelberg game theory and neural network optimization, optimize resource allocation and task offloading, meeting stringent QoS requirements while ensuring efficient resource utilization [72, 73, 52].

Game-theoretic and hybrid approaches in reinforcement learning provide a robust framework for addressing task offloading and resource management challenges in edge computing. By integrating game theory and reinforcement learning, these approaches significantly enhance the scalability, adaptability, and efficiency of edge computing systems. They optimize task offloading and resource allocation strategies among competitive mobile users and edge cloud nodes, ensuring effective responses to modern application demands while adjusting to varying network conditions. Concepts like Nash equilibrium and potential games support the development of efficient algorithms that reduce latency and energy consumption while maintaining system stability [74, 68].

6 Resource Allocation and Management

The intricacies of resource allocation and management are pivotal to enhancing performance within edge computing environments. The dynamic nature of workloads, coupled with the competition for limited resources and the need for adaptive solutions, necessitates a comprehensive understanding of the challenges encountered in this domain. The subsequent subsections delve into these challenges, examining the factors that complicate task management and resource distribution across heterogeneous edge computing systems.

6.1 Challenges in Resource Allocation

Resource allocation in edge computing is hindered by the dynamic and heterogeneous characteristics of these systems, impacting task management and resource distribution [6]. A significant challenge lies in managing limited resources within multi-user and multi-MEC server environments, where intense competition for computational and communication resources occurs [2]. The complexity is further heightened by fluctuating workloads and varying device capabilities, necessitating adaptive solutions responsive to changing system states [5].

Efficient task offloading under strict deadlines and dynamic network conditions is another critical challenge in heterogeneous edge environments [8]. This is compounded by the reliance on accurate channel state information and static network topologies, which often do not reflect the rapidly changing realities of edge environments [1]. Additionally, incorporating security considerations into task offloading strategies adds complexity, requiring a balance between delay, energy consumption, and security [7].

The use of synthetic datasets for evaluating task offloading strategies limits the understanding of real-world complexities and unpredictability in edge computing scenarios [3]. This highlights the need for more realistic models and benchmarks that better simulate the diverse conditions encountered in practical deployments.

Addressing these challenges requires the development of advanced algorithms and frameworks capable of dynamically adapting to changing network conditions and efficiently allocating resources. Innovative methods, such as the OTPPS approach, which adjusts resource allocation in response to system state changes, demonstrate potential in overcoming these obstacles and enhancing the efficiency and scalability of edge computing systems [5]. By leveraging adaptive strategies, edge computing environments can achieve improved load balancing, higher task completion rates, and enhanced energy efficiency, aligning with modern application demands and user expectations.

6.2 Intelligent Algorithms for Dynamic Resource Management

Intelligent algorithms are crucial for optimizing dynamic resource management in edge computing environments, effectively addressing challenges posed by fluctuating network conditions and resource constraints. These algorithms employ advanced techniques to assess network states and adjust resource management strategies, enhancing system performance and user satisfaction. For example, a deep learning-based approach for cache placement dynamically adapts to user preferences and channel conditions, ensuring optimal resource utilization [30].

The Energy-Efficient Joint Offloading and Wireless Resource Allocation Strategy (EEJS) highlights the role of intelligent algorithms in optimizing task offloading and resource allocation, enhancing both energy efficiency and system performance [2]. Similarly, the UCMEC framework employs decentralized joint optimization schemes for real-time resource management, showcasing the adaptability and efficiency of intelligent algorithms [32].

Incorporating reinforcement learning, the proposed Q-learning approach offers significant adaptability and real-time optimization capabilities, effectively managing dynamic workloads without relying on static task characteristics [34]. This adaptability is essential for maintaining optimal performance in edge environments characterized by dynamic and unpredictable conditions.

DeepEdge exemplifies the application of intelligent algorithms by utilizing real-time data to optimize task offloading decisions, showcasing the potential of machine learning techniques in enhancing resource management strategies [6]. The integration of intelligent algorithms not only improves task offloading efficiency but also enhances the adaptability and robustness of edge computing systems.

Moreover, the approach proposed in [4] demonstrates improved prediction accuracy and reduced computational complexity, facilitating faster and more efficient offloading decisions. These advantages are critical for optimizing resource allocation in environments with limited computational resources and dynamic network conditions.

Intelligent algorithms, particularly those leveraging advanced optimization techniques such as reinforcement learning, are essential for dynamic resource management in complex computing environments. These algorithms continuously adapt to changing workloads and system states, significantly improving efficiency in resource allocation and task scheduling. For example, Q-learning-based approaches have shown superior performance in optimizing task completion times and resource utilization compared to traditional static methods. As cloud and edge computing systems grapple with challenges such as resource limitations and unpredictable workloads, the integration of computational intelligence techniques is increasingly vital. This evolution not only enhances system performance and reduces operational costs but also promotes sustainable energy consumption, positioning intelligent algorithms as a promising solution for next-generation computing frameworks, including IoT and mobile edge intelligence systems [34, 60, 62, 40, 75]. These algorithms enhance the scalability, adaptability, and efficiency of edge computing systems, ensuring they meet the demands of modern applications and dynamic network conditions.

6.3 Strategies for Efficient Resource Utilization

Efficient resource utilization in edge computing is essential for optimizing performance and reducing operational costs, particularly in environments with dynamic workloads and constrained resources. Various innovative strategies have been developed to enhance resource utilization in fog and edge computing, employing advanced methodologies such as game theory for optimizing competitive resource allocation, cooperative learning to foster collaboration among distributed systems, and deep learning-based optimization techniques that adapt to dynamic workloads and environmental changes. These approaches leverage artificial intelligence (AI) and machine learning (ML) to address

challenges such as resource limitations, workload variability, and the need for real-time decision-making, ultimately improving efficiency and performance in complex computing environments [41, 60, 74, 62].

The Bilateral Game Approach, explored in the DTOA framework, exemplifies the use of game-theoretic methods to enhance resource allocation efficiency, increasing profitability for Edge Servers (ESs) while boosting payoffs for Task Entities (TEs) [69]. Similarly, the Non-Cooperative Game Approach in DPDA highlights the advantages of faster processing times and reduced costs, effectively managing varying dataset sizes to ensure efficient resource utilization [74].

Cooperative learning among edge devices, as implemented in the TOBM framework, is crucial for reducing network congestion and improving resource allocation. This method leverages the collaborative capabilities of edge devices to optimize task offloading, thereby enhancing overall system efficiency [64]. The integration of cooperative strategies in resource management underscores the importance of collective optimization in dynamic edge environments.

The IOPO framework, utilizing deep learning for rapid decision-making, showcases the effectiveness of adaptive strategies in optimizing energy consumption and resource allocation. By enabling quick adaptation to changing conditions, IOPO significantly improves resource utilization efficiency [76], which is essential for maintaining optimal performance in fluctuating network conditions.

Addressing the challenges of lightweight data processing and efficient resource allocation in fog and edge computing remains an ongoing area of research, as highlighted by Hong et al. Despite significant progress, further advancements are needed to enhance the scalability and efficiency of resource management strategies [12].

The congestion-aware distributed task offloading method dynamically adjusts offloading decisions based on network topology and task information, improving resource utilization even in the presence of network congestion, thereby emphasizing the importance of dynamic adaptation in resource management [77].

The strategies outlined in various studies underscore the critical role of advanced techniques and collaborative frameworks in optimizing resource utilization within edge computing environments. By leveraging decentralized computing paradigms like Opportunistic Edge Computing (OEC) and Multi-access Edge Computing (MEC), these approaches aim to enhance scalability and responsiveness while addressing the unique challenges posed by emerging applications and sustainable development [13, 46, 52]. Integrating game-theoretic methods, cooperative learning, and adaptive optimization enhances the scalability, efficiency, and adaptability of edge computing systems, ensuring they meet the demands of modern applications and dynamic network conditions.

7 Edge Intelligence and Distributed Computing

7.1 Conceptual Framework of Edge Intelligence

Edge intelligence represents a transformative paradigm in distributed computing, strategically deploying computational resources at the network's edge to enhance processing capabilities and decision-making. By integrating advanced machine learning techniques and decentralized management strategies, this framework optimizes resource utilization and boosts system efficiency. The mean-field game approach exemplifies a robust mechanism for decentralized task management, enhancing edge intelligence in Multi-Access Edge Computing (MEC) environments [78]. Utilizing under-utilized computational resources on edge devices enables efficient processing of machine learning tasks and real-time decision-making, as demonstrated by frameworks like DeepEdge, which autonomously adapts to varying environments [39, 6].

The integration of digital twinning within edge intelligence further optimizes network management and efficiency by simulating real-world processes to provide insights that enhance decision-making and resource allocation [79]. Adaptive frameworks like AFLA, which quickly adapt to new data, embody edge intelligence by enhancing processing capabilities vital for maintaining performance in dynamic environments [80]. Through advanced learning algorithms and decentralized management strategies, edge intelligence frameworks ensure efficient resource utilization and improve the overall performance of distributed computing systems.

7.2 Intelligent Algorithms for Enhanced Processing

Intelligent algorithms are critical for optimizing processing capabilities at the edge, employing advanced computational techniques to enhance system efficiency and tackle real-time data processing challenges in Edge AI environments. These algorithms facilitate efficient data analysis and management near data sources, integrating Edge Computing with Artificial Intelligence (AI) to meet demands for low latency, high reliability, and effective resource allocation [41, 81, 62, 40]. By incorporating machine learning and deep learning techniques, these algorithms dynamically adapt resource allocation strategies, ensuring optimal performance amidst fluctuating workloads and limited computational resources [37].

The Multi-Agent Deep Deterministic Policy Gradient (MADDPG) method exemplifies the optimization of task management in complex edge environments through multiple agents learning via environmental interactions, thereby improving resource allocation and task offloading [63]. Game-theoretic and hybrid approaches further illustrate the potential of intelligent algorithms to enhance edge processing capabilities by modeling interactions between decision-makers, providing a structured framework for optimizing resource distribution and task processing [70].

Moreover, the integration of digital twinning and adaptive learning frameworks within edge computing environments enhances processing capabilities by enabling real-time decision-making and proactive resource management [79]. Intelligent algorithms are crucial in enhancing processing capabilities at the edge, enabling efficient resource management, task scheduling, and improved decision-making processes, thereby addressing challenges posed by the vast data produced by IoT devices and mobile technologies. As the field evolves, it presents significant opportunities for advancing resource management and addressing scalability and security challenges in edge environments [41, 62].

7.3 Scalability and Efficiency in Distributed Computing

Scalability and efficiency are essential characteristics of distributed computing systems, particularly in edge computing, where real-time processing and low-latency communication are critical. Distributed computing frameworks enhance scalability by leveraging the collective computational power of multiple nodes, enabling efficient processing of large-scale data and complex tasks [29]. This decentralized approach distributes workloads across various nodes, optimizing resource utilization and ensuring efficient execution of computational tasks, even under dynamic network conditions [6].

The integration of edge computing with distributed systems significantly bolsters scalability by allowing tasks to be processed closer to the data source, minimizing data transmission to centralized servers and reducing latency [3]. This proximity is particularly advantageous for applications requiring rapid response times, such as IoT systems and real-time data analytics [5].

Efficiency in distributed computing is augmented by intelligent algorithms that dynamically manage resource allocation and task scheduling. Deep reinforcement learning-based algorithms enable distributed systems to adapt to changing network conditions and optimize resource utilization in real time [37]. By continuously learning from the environment and adjusting strategies, these intelligent algorithms maintain high efficiency levels, even amidst fluctuating workloads and limited computational resources [8].

Moreover, the scalability of distributed computing is supported by the seamless integration of additional nodes into the network, allowing flexible expansion of computational resources as demand increases [29]. This scalability is essential for managing the growing complexity and volume of data generated by modern applications, ensuring that distributed systems can effectively handle increased workloads without compromising performance [2].

The advantages of distributed computing in enhancing scalability and efficiency are evident in its ability to optimize resource utilization, reduce latency, and improve system performance. By leveraging decentralized systems and integrating advanced algorithms for dynamic resource management, edge computing environments achieve greater scalability and efficiency, meeting the demands of modern applications and ensuring high-quality service delivery [6].

8 Conclusion

8.1 Future Directions and Research Opportunities

Advancements in edge computing are poised to address critical challenges, enhancing system robustness, scalability, and adaptability. Future research will focus on refining algorithms to operate efficiently in dynamic environments, integrating machine learning for superior resource management, and optimizing communication resource allocation frameworks. Emphasis will be placed on dynamic features such as node mobility and task diversity, alongside the implementation of federated reinforcement learning to enhance task offloading strategies. The convergence of multi-task learning with architectures like GNN-RL and the adoption of multi-agent reinforcement learning frameworks will be pivotal in refining decision-making processes within complex networks. Additionally, efforts to adapt algorithms to fluctuating network conditions will ensure robust performance in real-world scenarios. Expanding methodologies to encompass more complex scenarios will be crucial for optimizing resource allocation and task offloading, thereby enhancing the capabilities of edge computing systems to meet the evolving demands of modern applications.

8.2 Privacy and Security Considerations

Privacy and security remain paramount in edge computing due to its decentralized nature, which introduces unique challenges and vulnerabilities. The proximity of edge devices to end-users necessitates stringent privacy protection mechanisms to prevent unauthorized access and data breaches, ensuring user trust and compliance with regulatory frameworks. In multi-tier computing systems, the layered architecture increases security challenges, necessitating comprehensive strategies that include secure communication protocols, authentication, and intrusion detection systems. Advanced encryption techniques and secure data transmission protocols are essential for safeguarding sensitive information during processing and transmission. Implementing decentralized security frameworks can enhance resilience against attacks, distributing security functions across multiple nodes to mitigate the impact of breaches. By prioritizing privacy and security, edge computing environments can effectively manage risks, ensuring the protection of sensitive data and the integrity of computational processes.

References

[1] Yang Cai, Jaime Llorca, Antonia M. Tulino, and Andreas F. Molisch. Mobile edge computing network control: Tradeoff between delay and cost, 2022.

[2] Kang Cheng, Yinglei Teng, Weiqi Sun, An Liu, and Xianbin Wang. Energy-efficient joint offloading and wireless resource allocation strategy in multi-mec server systems, 2018.

[3] Frederico Metelo, Stevo Racković, Pedro Ákos Costa, and Cláudia Soares. Peersimgym: An environment for solving the task offloading problem with reinforcement learning, 2024.

[4] Bo Yang, Xuelin Cao, Joshua Bassey, Xiangfang Li, Timothy Kroecker, and Lijun Qian. Computation offloading in multi-access edge computing networks: A multi-task learning approach, 2020.

[5] Yang Li, Xinlei Ge, Bo Lei, Xing Zhang, and Wenbo Wang. Joint task partitioning and parallel scheduling in device-assisted mobile edge networks, 2023.

[6] Baris Yamansavascilar, Ahmet Cihat Baktir, Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. Deepedge: A deep reinforcement learning based task orchestrator for edge computing, 2022.

[7] Amir Pakmehr. Task offloading in fog computing with deep reinforcement learning: Future research directions based on security and efficiency enhancements, 2024.

[8] Turgay Pamuklu, Aisha Syed, W. Sean Kennedy, and Melike Erol-Kantarci. Heterogeneous gnn-rl based task offloading for uav-aided smart agriculture, 2023.

[9] Siqi Zhang, Na Yi, and Yi Ma. A survey of computation offloading with task types, 2024.

[10] Cheol-Ho Hong and Blesson Varghese. Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms. *ACM Computing Surveys (CSUR)*, 52(5):1–37, 2019.

[11] Wei Du, Tao Lei, Qiang He, Wei Liu, Qiwang Lei, Hailiang Zhao, and Wei Wang. Service capacity enhanced task offloading and resource allocation in multi-server edge computing environment, 2019.

[12] Cheol-Ho Hong and Blesson Varghese. Resource management in fog/edge computing: A survey, 2018.

[13] Andrea Hamm, Alexander Willner, and Ina Schieferdecker. Edge computing: A comprehensive survey of current initiatives and a roadmap for a sustainable edge computing development, 2019.

[14] Fang Liu, Guoming Tang, Youhuizi Li, Zhiping Cai, Xingzhou Zhang, and Tongqing Zhou. A survey on edge computing systems and tools, 2019.

[15] Hyame Assem Alameddine, Sanaa Sharafeddine, Samir Sebbah, Sara Ayoubi, and Chadi Assi. Dynamic task offloading and scheduling for low-latency iot services in multi-access edge computing. *IEEE Journal on Selected Areas in Communications*, 37(3):668–682, 2019.

[16] Tuyen X. Tran and Dario Pompili. Joint task offloading and resource allocation for multi-server mobile-edge computing networks, 2017.

[17] Ming Tang and Vincent W. S. Wong. Deep reinforcement learning for task offloading in mobile edge computing systems, 2020.

[18] Zhanwei Yu, Yi Zhao, Tao Deng, Lei You, and Di Yuan. Less carbon footprint in edge computing by joint task offloading and energy sharing, 2023.

[19] Ruihuai Liang, Bo Yang, Zhiwen Yu, Xuelin Cao, Derrick Wing Kwan Ng, and Chau Yuen. A multi-head ensemble multi-task learning approach for dynamical computation offloading, 2023.

[20] Guanjin Qu and Huaming Wu. Dmro:a deep meta reinforcement learning-based task offloading framework for edge-cloud computing, 2020.

[21] Kunlun Wang, Jiong Jin, Yang Yang, Tao Zhang, Arumugam Nallanathan, Chintha Tellambura, and Bijan Jabbari. Task offloading with multi-tier computing resources in next generation wireless networks, 2022.

[22] Xin Tang, Xu Chen, Liekang Zeng, Shuai Yu, and Lin Chen. Joint multi-user dnn partitioning and computational resource allocation for collaborative edge intelligence, 2020.

[23] Fei Hu, Kunal Mehta, Shivakant Mishra, and Mohammad AlMutawa. A dynamic distributed scheduler for computing on the edge, 2023.

[24] Mohammad Goudarzi, Marimuthu Palaniswami, and Rajkumar Buyya. A distributed deep reinforcement learning technique for application placement in edge and fog computing environments, 2021.

[25] Thai T. Vu, Nguyen Van Huynh, Dinh Thai Hoang, Diep N. Nguyen, and Eryk Dutkiewicz. Offloading energy efficiency with delay constraint for cooperative mobile edge computing networks, 2018.

[26] Mithun Mukherjee, Vikas Kumar, Suman Kumar, Jaime Lloret, Qi Zhang, and Mian Guo. Reconfigurable intelligent surface-assisted edge computing to minimize delay in task offloading, 2021.

[27] Ali Al-Shuwaili and Osvaldo Simeone. Energy-efficient resource allocation for mobile edge computing-based augmented reality applications, 2017.

[28] Ruan Yanjiao. Encryption mechanism and resource allocation optimization based on edge computing environment, 2022.

[29] Xiang Li, Mustafa Abdallah, Yuan-Yao Lou, Mung Chiang, Kwang Taik Kim, and Saurabh Bagchi. Dynamic dag-application scheduling for multi-tier edge computing in heterogeneous networks, 2024.

[30] Jiechen Chen, Hong Xing, Xiaohui Lin, Arumugam Nallanathan, and Suzhi Bi. Joint resource allocation and cache placement for location-aware multi-user mobile edge computing, 2022.

[31] Feng Wang, Jie Xu, and Shuguang Cui. Optimal energy allocation and task offloading policy for wireless powered mobile edge computing systems. *IEEE Transactions on Wireless Communications*, 19(4):2443–2459, 2020.

[32] Langtian Qin, Hancheng Lu, Yuang Chen, Baolin Chong, and Feng Wu. Towards decentralized task offloading and resource allocation in user-centric mobile edge computing, 2023.

[33] Chong Huang, Gaojie Chen, Pei Xiao, Yue Xiao, Zhu Han, and Jonathon A. Chambers. Joint offloading and resource allocation for hybrid cloud and edge computing in sagins: A decision assisted hybrid action space deep reinforcement learning approach, 2024.

[34] Pochun Li, Yuyang Xiao, Jinghua Yan, Xuan Li, and Xiaoye Wang. Reinforcement learning for adaptive resource scheduling in complex system environments, 2024.

[35] Xuming An, Rongfei Fan, Han Hu, Ning Zhang, Saman Atapattu, and Theodoros A. Tsiftsis. Joint task offloading and resource allocation for iot edge computing with sequential task dependency, 2021.

[36] Liangkai Zhou, Yuncong Hong, Shuai Wang, Ruihua Han, Dachuan Li, Rui Wang, and Qi Hao. Learning centric wireless resource allocation for edge computing: Algorithm and experiment, 2020.

[37] Mushu Li, Jie Gao, Lian Zhao, and Xuemin Shen. Deep reinforcement learning for collaborative edge computing in vehicular networks, 2020.

[38] Zhuo Li, Xu Zhou, Taixin Li, and Yang Liu. An optimal-transport-based reinforcement learning approach for computation offloading, 2021.

[39] Christian Makaya, Amalendu Iyer, Jonathan Salfity, Madhu Athreya, and M Anthony Lewis. Cost-effective machine learning inference offload for edge computing, 2020.

[40] Muhammad Asim, Yong Wang, Kezhi Wang, and Pei-Qiu Huang. A review on computational intelligence techniques in cloud and edge computing, 2020.

[41] Shuiguang Deng, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, and Albert Y Zomaya. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8):7457–7469, 2020.

[42] Khai Doan, Wesley Araujo, Evangelos Kranakis, Ioannis Lambadaris, and Yannis Viniotis. Optimal task offloading policy in edge computing systems with firm deadlines, 2023.

[43] Yizhen Xu, Peng Cheng, Zhuo Chen, Ming Ding, Branka Vucetic, and Yonghui Li. Task offloading for large-scale asynchronous mobile edge computing: An index policy approach, 2020.

[44] Bizheng Liang, Rongfei Fan, and Han Hu. Energy-efficient task offloading and resource allocation for multiple access mobile edge computing, 2021.

[45] Parisa Fard Moshiri, Murat Simsek, and Burak Kantarci. On the interplay between network metrics and performance of mobile edge offloading, 2024.

[46] Richard Olaniyan, Olamilekan Fadahunsi, Muthucumaru Maheswaran, and Mohamed Faten Zhani. Opportunistic edge computing: Concepts, opportunities and research challenges, 2018.

[47] Parisa Fard Moshiri, Murat Simsek, and Burak Kantarci. Joint optimization of completion ratio and latency of offloaded tasks with multiple priority levels in 5g edge, 2025.

[48] Feng Wang, Jie Xu, and Shuguang Cui. Optimal energy allocation and task offloading policy for wireless powered mobile edge computing systems, 2020.

[49] Paolo Di Lorenzo, Sergio Barbarossa, and Stefania Sardellitti. Joint optimization of radio resources and code partitioning in mobile edge computing, 2016.

[50] Mahla Rahati-Quchani, Saeid Abrishami, and Mehdi Feizi. An efficient mechanism for computation offloading in mobile-edge computing, 2020.

[51] Pavel Mach and Zdenek Becvar. Mobile edge computing: A survey on architecture and computation offloading, 2017.

[52] Yiqin Deng, Xianhao Chen, Guangyu Zhu, Yuguang Fang, Zhigang Chen, and Xiaoheng Deng. Actions at the edge: Jointly optimizing the resources in multi-access edge computing, 2022.

[53] Ahmed A. Al-habob and Octavia A. Dobre. Mobile edge computing and artificial intelligence: A mutually-beneficial relationship, 2020.

[54] Gabriel F. C. de Queiroz, José F. de Rezende, and Valmir C. Barbosa. A flexible algorithm to offload dag applications for edge computing, 2023.

[55] Asrar Ahmed Baktayan and Ibrahim Ahmed Al-Baltah. A survey on intelligent computation offloading and pricing strategy in uav-enabled mec network: Challenges and research directions, 2022.

[56] Xinyu Huang, Lijun He, Xing Chen, Liejun Wang, and Fan Li. Revenue and energy efficiency-driven delay constrained computing task offloading and resource allocation in a vehicular edge computing network: A deep reinforcement learning approach, 2020.

[57] Fang Liu, Guoming Tang, Youhuizi Li, Zhiping Cai, Xingzhou Zhang, and Tongqing Zhou. A survey on edge computing systems and tools. *Proceedings of the IEEE*, 107(8):1537–1562, 2019.

[58] Xiangg Li, Rongfei Fan, and Han Hu. Energy-efficient task offloading for relay aided mobile edge computing under sequential task dependency, 2021.

[59] Ummy Habiba, Setareh Maghsudi, and Ekram Hossain. A repeated auction model for load-aware dynamic resource allocation in multi-access edge computing, 2024.

[60] Ai-based fog and edge computing: A systematic review taxonomy and future directions.

[61] Jianyu Wang, Jianli Pan, Flavio Esposito, Prasad Calyam, Zhicheng Yang, and Prasant Mohapatra. Edge cloud offloading algorithms: Issues, methods, and perspectives, 2018.

[62] Sukhpal Singh Gill, Muhammed Golec, Jianmin Hu, Minxian Xu, Junhui Du, Huaming Wu, Guneet Kaur Walia, Subramaniam Subramanian Murugesan, Babar Ali, Mohit Kumar, Kejiang Ye, Prabal Verma, Surendra Kumar, Felix Cuadrado, and Steve Uhlig. Edge ai: A taxonomy, systematic review and future directions, 2024.

[63] Xinyu Huang, Lijun He, and Wanyue Zhang. Vehicle speed aware computing task offloading and resource allocation based on multi-agent reinforcement learning in a vehicular edge computing network, 2020.

[64] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, and H. Vincent Poor. Cooperative task offloading and block mining in blockchain-based edge computing with multi-agent deep reinforcement learning, 2021.

[65] Cheng Zhang, Yinuo Deng, Hailiang Zhao, Tianlv Chen, and Shuiguang Deng. Tail-learning: Adaptive learning method for mitigating tail latency in autonomous edge systems, 2023.

[66] Tesfay Zemuy Gebrekidan, Sebastian Stein, and Timothy J. Norman. Combinatorial client-master multiagent deep reinforcement learning for task offloading in mobile edge computing, 2024.

[67] Salwa Mostafa, Mateus P. Mota, Alvaro Valcarce, and Mehdi Bennis. Emergent communication protocol learning for task offloading in industrial internet of things, 2024.

[68] Jianen Yan, Ning Li, Zhaoxin Zhang, Alex X. Liu, Jose Fernan Martinez, and Xin Yuan. Game theory based joint task offloading and resources allocation algorithm for mobile edge computing, 2019.

[69] Zheng Xiao, Dan He, Yu Chen, Anthony Theodore Chronopoulos, Schahram Dustdar, and Jiayi Du. A bilateral game approach for task outsourcing in multi-access edge computing, 2020.

[70] Fuhong Song, Huanlai Xing, Xinhan Wang, Shouxi Luo, Penglin Dai, Zhiwen Xiao, and Bowen Zhao. Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in uav-assisted mobile edge computing, 2022.

[71] Jin Wang, Jia Hu, Geyong Min, Albert Y. Zomaya, and Nektarios Georgalas. Fast adaptive task offloading in edge computing based on meta reinforcement learning, 2020.

[72] Ting Xiaoyang, Minfeng Zhang, Shu gonglee, and Saimin Chen Zhang. Joint resource optimization, computation offloading and resource slicing for multi-edge traffic-cognitive networks, 2024.

[73] Xin Long, Jigang Wu, and Long Chen. Energy-efficient offloading in mobile edge computing with edge-cloud collaboration, 2018.

[74] Bo Yang, Zhiyong Li, and Wenbin Liu. Non-cooperative game approach for task offloading in edge clouds, 2018.

[75] Zehong Lin, Suzhi Bi, and Ying-Jun Angela Zhang. Optimizing ai service placement and resource allocation in mobile edge intelligence systems, 2021.

[76] Jianqiu Wu, Zhongyi Yu, Jianxiong Guo, Zhiqing Tang, Tian Wang, and Weijia Jia. A fast task offloading optimization framework for irs-assisted multi-access edge computing system, 2023.

[77] Zhongyuan Zhao, Jake Perazzone, Gunjan Verma, and Santiago Segarra. Congestion-aware distributed task offloading in wireless multi-hop networks using graph neural networks, 2024.

[78] Shubham Aggarwal, Muhammad Aneeq uz Zaman, Melih Bastopcu, Sennur Ulukus, and Tamer Başar. Fully decentralized task offloading in multi-access edge computing systems, 2024.

[79] Nikos G. Evgenidis, Nikos A. Mitsiou, Vasiliki I. Koutsioumpa, Sotiris A. Tegos, Panagiotis D. Diamantoulakis, and George K. Karagiannidis. Multiple access in the era of distributed computing and edge intelligence, 2024.

[80] Iman Rahmati, Hamed Shah-Mansouri, and Ali Movaghar. Qeco: A qoe-oriented computation offloading algorithm based on deep reinforcement learning for mobile edge computing, 2024.

[81] Andrea Fresa and Jaya Prakash Champati. Offloading algorithms for maximizing inference accuracy on edge device under a time constraint, 2021.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.