engineering proceedings

Proceeding Paper

Exploration of Multi-Task Scheduling in Multi-Access Edge Computing †

J. Anand and B. Karthikeyan *

School of Electronics Engineering, Vellore Institute of Technology, Vellore 632014, India; anand.j2022@vitstudent.ac.in
* Correspondence: bkarthikeyan@vit.ac.in

† Presented at the 2nd Computing Congress 2023, Chennai, India, 28–29 December 2023.

Abstract: The emergence of multi-access edge computing (MEC) has brought about significant advancements in application design and deployment by providing computing resources at the network's edge. MEC provides computing resources on the fringes of the network, allowing for near-real-time data processing and fast responses to user requests. In this context, scheduling plays a crucial role in offloading decisions in multi-access edge computing. The motivations for scheduling are to improve the quality of the experience, reduce latency, and increase performance. In this paper, we explore the various scheduling techniques available for MEC systems, including static scheduling, dynamic scheduling, heuristics, meta-heuristics and hybrid scheduling. We analyze the advantages and disadvantages of each technique and discuss how they can be used to optimize the performance of MEC applications. We also present a case study of an MEC system and demonstrate how the various scheduling techniques can be used to maximize its performance. Finally, we address both the challenges and prospects of MEC scheduling and suggest directions for future research.

Keywords: cloud computing; multi-access edge computing; quality of service; resource allocation; task offloading; task scheduling techniques

check for updates

Citation: Anand, J.; Karthikeyan, B. Exploration of Multi-Task Scheduling in Multi-Access Edge Computing. *Eng. Proc.* **2024**, *62*, 4. https://doi.org/10.3390/engproc2024062004

Academic Editors: Geetha Ganesan, Xiaochun Cheng and Valentina Emilia Balas

Published: 29 February 2024

1. Introduction

An edge-to-center trend may be seen in the current state of traditional cloud computing. Multi-access edge computing (MEC) has emerged as a pivotal paradigm to meet the escalating demand for low-latency, high-throughput services in contemporary computing landscapes. As computing resources move closer to end-users and devices, the need for efficient task scheduling becomes paramount. The exploration of multi-task scheduling in multi-access edge computing (MEC) is motivated by the increasing demand for efficient resource utilization in modern computing environments. Multi-task scheduling involves allocating resources to multiple tasks with varying resource requirements and priorities. In this literature review, we will discuss several research papers that address multi-task scheduling in MEC. We will discuss various scheduling algorithms, including heuristic, optimization, and machine-learning-based approaches. This paper also addresses the difficulties and potential advantages in this field [1]. It considers the balance between energy consumption and task completion time and is shown to outperform heuristic algorithms [2]. The multi-objective optimization approach considers both task completion time and energy consumption. The approach employs dynamic voltage and frequency scaling to reduce energy consumption and is shown to outperform other scheduling algorithms [3]. A deep reinforcement learning-based approach was introduced for multi-task scheduling in MEC. The approach learns to make optimal scheduling decisions by observing the network environment and is shown to outperform other scheduling algorithms [4]. Due to its plethora of resources, cloud computing is ideally suited to overcome such obstacles and provide a smooth platform [5,6].

The planning of tasks should be assigned either in the cloud or at the edge, and scheduling is crucial to the effectiveness of this edge cloud collaboration. Figure 1, below, depicts the edge cloud's architecture. The three tiers are the device tier, edge tier, and cloud tier. When local resources are inadequate, each connected device in the device layer first computes its work locally before transmitting it to the cloud. The process of allocating resources to the specific user who requests service is known as scheduling. It is still challenging to decide whether to schedule a certain job in the edge cloud at the edge or in the cloud to optimize resource utilization. The following are the study's main contributions:

- Offer a thorough analysis of scheduling tasks in edge-cloud computing.
- Develop a clear taxonomy to categorize and classify the different multitask scheduling strategies in MEC.
- Investigate and contrast the present task scheduling techniques.
- Explore forthcoming research issues and emerging concerns in multitask scheduling for MEC.

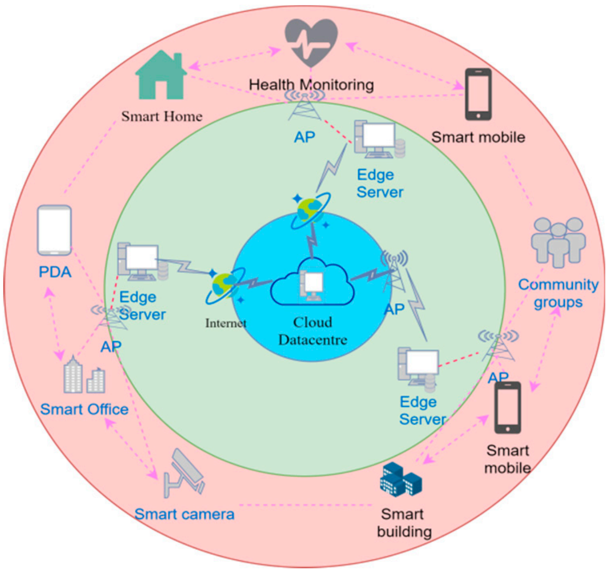

Figure 1. The edge cloud computing scenario.

The remaining sections are arranged as follows: Section 2 discusses the related work of various studies. Section 3 specifies the details of issues and challenges in edge cloud systems. Section 4 outlines the various limitations and QoS criteria that go along with the task scheduling categorization methods used in edge computing. The summary of scheduling algorithms with QoS parameters is provided in Section 5. Section 6 is the analysis, and is the critical study of this paper. Future directions are described in Section 7, which follows. Finally, Section 8 provides a conclusion.

2. Related Work

Task offloading and scheduling techniques are a key strategy for overcoming the edge device processing, storage and power resource limitations in an IoT network based on edge computing. To speed up job processing, conserve energy and shorten reaction times, the edge device can offload some or all of the computing activities to an edge computing server. The crucial aspect of scheduling in edge cloud is still in its embryonic stage and requires extensive research. In this section, we will explore several survey studies relevant to scheduling in edge cloud environments, which can provide valuable insights for our survey. Wang et al. [7] have presented an edge cloud perspective in which task scheduling techniques consider factors such as task types, user mobility, cooperativeness and multi-objective functions, including response time, energy consumption and load balancing.

In Figure 2, the task flow process in edge computing is illustrated.

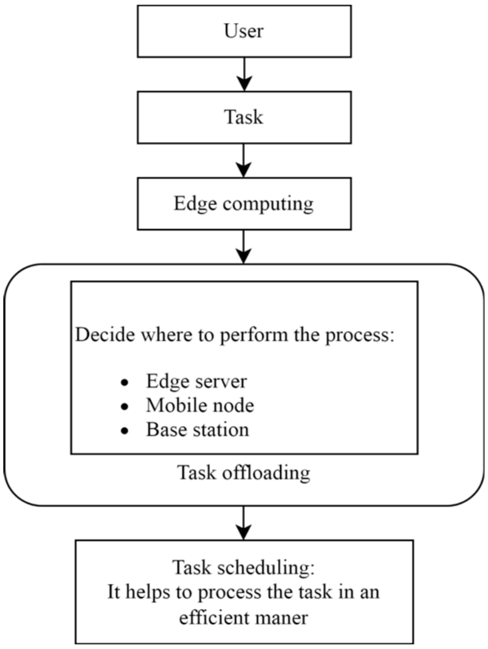

Figure 2. Task flow process in edge computing.

Requests made by users are known as jobs. A job encompasses a set of tasks that can include inputting data into the system, processing it, utilizing specific infrastructure, storing data in designated data centers or delivering results to end users.

3. Multi-Task Scheduling in MEC: Issues and Challenges

Scheduling problems in edge cloud have been gaining significant attention in recent years. This problem deals with allocating resources in edge cloud computing environments, where the resources are scattered across a distributed control plane. One key challenge in edge cloud scheduling is ensuring efficient resource utilization while providing low-latency services. Various approaches have been proposed to address the scheduling problems in edge cloud systems. These include heuristic algorithms, meta-heuristic algorithms and hybrid algorithms. *Heterogeneous and Dynamic Edge Environments*: The MEC environment consists of heterogeneous edge devices with varying computing capabilities and resource availability. Moreover, the dynamic nature of edge networks poses challenges for efficient multi-task scheduling [8]. *Real-Time and Latency Requirements*: Many MEC applications have stringent latency requirements, especially for real-time and latency-sensitive tasks such as augmented reality (AR) and autonomous driving. Scheduling multiple tasks while meeting these real-time constraints is a significant challenge [9]. *Energy Efficiency and Battery Life*: Energy consumption is a critical concern in MEC, as edge devices are often resource-constrained and powered by batteries. Balancing task execution time and energy consumption poses a significant challenge when aiming to achieve energy-efficient multi-task scheduling [10]. *Mobility and Task Migration*: Edge devices in MEC are often mobile, and tasks may need to be migrated between different edge servers as devices move. Mobility-aware predictive schemes based on genetic algorithms can significantly reduce the offloading failure rate of high-mobility users [11].

These challenges and issues highlight the complexities involved in multi-task scheduling in mobile edge computing. Researchers are actively addressing these challenges to develop efficient scheduling algorithms, optimization techniques, and frameworks that can enhance task allocation and resource management in MEC environments.

4. Analysis of Scheduling Methods in Edge Cloud: Task Scheduling Techniques in Edge Computing

In the context of edge computing, task scheduling involves the allocation of computational tasks to the accessible edge devices within a network. The objective is to maximize resource utilization, minimize task execution time and improve the overall performance of the system. Figure 3 discusses the taxonomy of task-scheduling techniques used in edge computing. *Data-Proximity-Aware Task Scheduling*: Data-proximity-aware scheduling algorithms in edge computing optimize task placement based on physical location and network topology, ensuring efficient processing near data sources to minimize latency and reduce bandwidth usage [12]. *Heuristic algorithm*: This aims to allocate tasks to the optimal sBS (small Base Station) in a mobile edge computing environment. The algorithm considers task details, user mobility and network constraints as a constraint satisfaction problem. The algorithm proceeds by sending a message from users to the central MEN controller, which allocates each task to an sBS where the delay is the shortest. During the allocation procedure, the system also takes user mobility prediction into account [13]. *Genetic algorithm*: Addressing the challenges faced by traditional cloud computing in providing storage and task computing services in the power grid. The genetic algorithm has a beneficial effect on energy consumption and load balancing and reduces latency. It uses genetic operations like crossover and mutation to find near-optimal solutions for resource allocation and task scheduling [14].

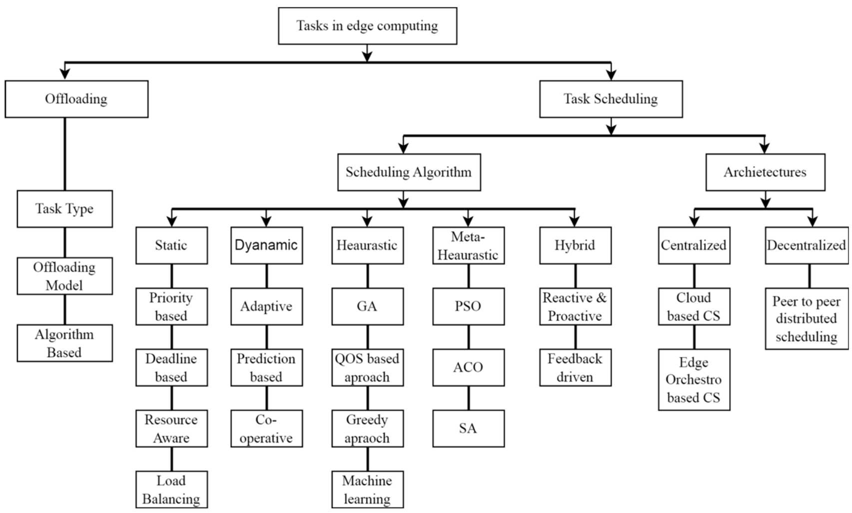

Figure 3. Task scheduling techniques in edge computing.

Greedy Algorithm: The greedy algorithm is used to select the best action at each time slot based on the current state of the system without considering the impact of this decision on future states. While this approach may not always lead to the globally optimal solution, it can be an effective way to make decisions in real-time systems where the state of the system is constantly changing [15]. *Centralizing and Decentralized Algorithm*: In centralized algorithms, a central entity is responsible for making decisions and coordinating the actions of all nodes in the network. The purpose of decentralizing the algorithm is to reduce the computational and communication overhead associated with centralized algorithms with the help of distributed decision-making and coordination tasks among the nodes in the network [16]. *Simulated Annealing*: Its ability to escape local optima and explore the solution space makes it particularly useful in complex and dynamic edge computing environments where traditional optimization approaches may struggle to find optimal solutions [17].

5. Summary of Scheduling Algorithm with QoS Parameters

In Table 1, we mention various scheduling algorithms and QoS parameters and their constraints.

Table 1. Reviews and remarks.

S No.	Author	Observations	Advantages	Evaluation	Shortcomings
1.	Piyush Gupta et al. [18] (2023)	Smart healthcare monitoring system utilizing edge computing and IoT, including a CNN-based prediction model with a 99.23% accuracy rate.	Reduced latency and improved efficiency.	Accuracy and error rate and lower processing time.	Limited resource capacity and network reliability.
2.	Jun Liu et al. [19] (2022)	The focus (PTAS-MAMEC) is to tackle within MEC, aiming to achieve minimum total energy consumption.	Good balance of speed and quality.	Optimality loss and time complexity.	Limited scalability consideration.
3.	Yu Zhang, Bing Tang et al. [20] (2022)	For scheduling tasks, considering deadlines and time sensitivity, a dynamic time-sensitive scheduling algorithm (DSOTS) was implemented.	Reduced latency and meet user deadlines efficiently.	Average processing time and cost.	Fault-tolerant techniques
4.	Boyin Zhang et al. [21] (2022)	Two task scheduling strategies (QDMP) were devised for the collaborative inference of edge-cloud systems, focusing on the partial migration of tasks.	Minimizes the overall delay.	Network delay and queuing delay.	A limited number of edge devices.
5.	A. S. Abohamama et al. [22] (2022)	A novel semi-dynamic real-time task scheduling technique is introduced specifically designed for high applications.	Reduces the cost of execution and task failure rate.	Total execution time and failure rate.	Dynamic scheduling problem
6.	Mohammed Maray et al. [13] (2022)	Heuristic and fully distributed offloading in static and dynamic contexts, as well as optimization technique.	Latency reduction and energy consumption.	Energy consumption and execution delays	Not addressed for large number of tasks.
7.	K. Kumaran E. Sasikala [23] (2022)	To address the NP-hard offloading problem, a dynamic weighed quantum arithmetic optimization algorithm (DWQAOA) was proposed.	To reduce network costs.	Task latency	Failed to incorporate real-time constraints.
8.	Songyue Han et al. [24] (2022)	An optimal computing offloading optimization algorithm for optimized mobile edge computing is centered on offloading.	Improves computation and prevents bottlenecks.	Total energy consumption; response latency and stability.	In this work, they have not focused on dynamic and complex environments.
9.	Li et al. [25] (2021)	Artificial fish swarm algorithm proposed for the edge device or the mobile node itself was calculated using MAFSA using delay and energy as its constraint values.	Minimum latency	The time delay was analyzed in terms of network size; Mu value; Beta value.	Mu value was fixed as 0.6 to achieve minimal time delay.
10.	Deng et al. [26] (2022)	Markov decision model designed to minimize the latency.	Maximizing a throughput.	Task latency vs. arrival.	Not suitable for real-time.
11.	Dong, Peiran et al. [16] (2020)	This study aims to research an edge-computing-based healthcare system in IOMTs to lower systems.	System-wide cost is minimized.	Energy consumption, delay and throughput.	Local edge nodes do not collaborate with adjacent MECs.
12.	Z. Sun et al. [15] (2020)	Task placement, resource allocation.	Full offloading.	Network delay, energy saving.	Nearest base stations not considered.
13.	Olokodana et al. [27] (2020)	The standard kriging approach presented a real-time seizure detection model in an edge computing context.	Enhanced latency by adopting the MEC layer.	Average detection latency.	IOMT devices were used only for data acquisition.
14.	T. Yang et al. [28] (2020)	A multi-user, multi-server MEC system environment in which task offloading and scheduling are optimized for better performance.	Enhanced computation and latency minimization.	Energy consumption and latency.	Large numbers of MEC servers needed.
15.	Ali et al. [29] (2020)	The authors developed a deep learning model for epileptic seizure detection using MEC.	Latency optimizations.	Load balancing level (85.71%).	MEC is only used for preprocessing.
16.	H.A. Alameddine et al. [30] (2019)	Task placement, resource allocation.	Numerical.	Task admission and execution time.	As no. of UEs increases, the admission rate decreases.
17.	H. Guo et al. [31] (2018)	Resource Allocation.	Enhanced network capacity.	Energy consumption and latency.	Can only achieve a Near-optimal solution.

6. Task Scheduling: An In-Depth Analysis

The findings emphasize the importance of effective scheduling in maximizing the performance and responsiveness of MEC systems, ultimately paving the way for further advancements in this dynamic field. A proficient analysis elucidates the importance of load balancing with energy awareness in optimizing the allocation of resources and improving system performance within mobile edge computing (MEC) environments. Additionally, the paper addresses the challenges and prospects of MEC scheduling and suggests directions for future research.

7. Future Directions

In the realm of multi-access edge computing (MEC), several promising future research directions are emerging. Dynamic task offloading and migration algorithms are a critical field for research because they can adjust in real time to changing network circumstances and resource availability. Advanced machine learning techniques can be harnessed to enhance scheduling decisions by predicting resource demands and user preferences. With the proliferation of diverse edge devices, future research aims to develop scheduling algorithms that can efficiently harness resources across a wide spectrum of devices, including IoT sensors with proper security.

8. Conclusions

The emergence of multi-access edge computing (MEC) has revolutionized the way applications are designed and deployed. However, efficient scheduling techniques are essential to effectively manage distributed computing resources in MEC systems. This paper has explored various scheduling techniques, including static, dynamic and hybrid scheduling, and has analyzed their advantages and disadvantages. By conducting a case study, the paper exemplifies the utilization of these scheduling techniques to enhance the performance of multi-access edge computing (MEC) applications. Additionally, the paper has highlighted the challenges and opportunities in MEC scheduling and has suggested future research directions.

Author Contributions: Conceptualization, J.A. and B.K.; methodology, J.A. and B.K.; investigation, J.A. and B.K.; writing—original draft preparation, J.A. and B.K.; writing—review and editing, J.A. and B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Authors declare no conflicts of interest.

References

1. Islam, A.; Debnath, A.; Ghose, M.; Chakraborty, S. A survey on task offloading in multi-access edge computing. *J. Syst. Archit.* **2021**, *118*, 102225. [CrossRef]
2. Lin, B.; Lin, X.; Zhang, S.; Wang, H.; Bi, S. Computation task scheduling and offloading optimization for collaborative mobile edge computing. In Proceedings of the 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), Hong Kong, China, 2–4 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 728–734.
3. Ajmal, M.S.; Iqbal, Z.; Khan, F.Z.; Bilal, M.; Mehmood, R.M. Cost-based energy efficient scheduling technique for dynamic voltage and frequency scaling system in cloud computing. *Sustain. Energy Technol. Assess.* **2021**, *45*, 101210.
4. Sheng, S.; Chen, P.; Chen, Z.; Wu, L.; Yao, Y. Deep reinforcement learning-based task scheduling in IoT edge computing. *Sensors* **2021**, *21*, 1666. [CrossRef] [PubMed]
5. Boukerche, A.; Guan, S.; Grande, R.E.D. Sustainable offloading in mobile cloud computing: Algorithmic design and implementation. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–37. [CrossRef]
6. Qi, Q.; Wang, J.; Ma, Z.; Sun, H.; Cao, Y.; Zhang, L.; Liao, J. Knowledge-driven service offloading decision for vehicular edge computing: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4192–4203. [CrossRef]

7. Wang, J.; Pan, J.; Esposito, F.; Calyam, P.; Yang, Z.; Mohapatra, P. Edge cloud offloading algorithms: Issues, methods, and perspectives. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–23. [CrossRef]
8. Li, N.; Yan, J.; Zhang, Z.; Martinez, J.F.; Yuan, X. Game theory based joint task offloading and resource allocation algorithm for mobile edge computing. In Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN), Tokyo, Japan, 17–19 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 791–796.
9. Liu, S.; Yu, Y.; Guo, L.; Yeoh, P.L.; Vucetic, B.; Li, Y. Adaptive delay-energy balanced partial offloading strategy in Mobile Edge Computing networks. *Digit. Commun. Netw.* **2022**, *9*, 1310–1318. [CrossRef]
10. Fu, Y.; Yang, X.; Yang, P.; Wong, A.K.; Shi, Z.; Wang, H.; Quek, T.Q. Energy-efficient offloading and resource allocation for mobile edge computing enabled mission-critical internet-of-things systems. *EURASIP J. Wirel. Commun. Netw.* **2021**, *2021*, 26. [CrossRef]
11. Li, H.; Sun, Y.; Zhang, Y.; Jin, B.; Wang, Z.; Wu, W.; Fang, C. Mobility-aware Predictive Computation Offloading and Task Scheduling for Mobile Edge Computing Networks. In Proceedings of the 2021 7th International Conference on Computer and Communications (ICCC), Chengdu, China, 10–13 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1349–1354.
12. Sahni, Y.; Cao, J.; Yang, L. Data-aware task allocation for achieving low latency in collaborative edge computing. *IEEE Internet Things J.* **2018**, *6*, 3512–3524. [CrossRef]
13. Maray, M.; Shuja, J. Computation offloading in mobile cloud computing and mobile edge computing: Survey, taxonomy, and open issues. *Mob. Inf. Syst.* **2022**, *2022*, 1121822. [CrossRef]
14. Nan, Z.; Wenjing, L.; Zhu, L.; Zhi, L.; Yumin, L.; Nahar, N. A New Task Scheduling Scheme Based on Genetic Algorithm for Edge Computing. *Comput. Mater. Contin.* **2022**, *71*, 843–854.
15. Sun, Z.; Nakhai, M.R. An online learning algorithm for distributed task offloading in multi-access edge computing. *IEEE Trans. Signal Process.* **2020**, *68*, 3090–3102. [CrossRef]
16. Dong, P.; Ning, Z.; Obaidat, M.S.; Jiang, X.; Guo, Y.; Hu, X.; Hu, B.; Sadoun, B. Edge computing-based healthcare systems: Enabling decentralized health monitoring in Internet of medical Things. *IEEE Netw.* **2020**, *34*, 254–261. [CrossRef]
17. Bi, J.; Yuan, H.; Duanmu, S.; Zhou, M.; Abusorrah, A. Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization. *IEEE Internet Things J.* **2020**, *8*, 3774–3785. [CrossRef]
18. Gupta, P.; Chouhan, A.V.; Wajeed, M.A.; Tiwari, S.; Bist, A.S.; Puri, S.C. Prediction of health monitoring with deep learning using edge computing. *Meas. Sens.* **2023**, *25*, 100604. [CrossRef]
19. Liu, J.; Liu, X. Energy-efficient allocation for multiple tasks in mobile edge computing. *J. Cloud Comput.* **2022**, *11*, 71. [CrossRef]
20. Zhang, Y.; Tang, B.; Luo, J.; Zhang, J. Deadline-aware dynamic task scheduling in edge–cloud collaborative computing. *Electronics* **2022**, *11*, 2464. [CrossRef]
21. Zhang, B.; Li, Y.; Zhang, S.; Zhang, Y.; Zhu, B. An adaptive task migration scheduling approach for edge-cloud collaborative inference. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 8804530. [CrossRef]
22. Abohamama, A.; El-Ghamry, A.; Hamouda, E. Real-time task scheduling algorithm for IoT-based applications in the cloud–fog environment. *J. Netw. Syst. Manag.* **2022**, *30*, 54. [CrossRef]
23. Kumaran, K.; Sasikala, E. Learning based latency minimization techniques in mobile edge computing (MEC) systems: A Comprehensive Survey. In Proceedings of the 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, 30–31 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
24. Han, S.; Ma, D.; Kang, C.; Huang, W.; Lin, C.; Tian, C. Optimization of Mobile Edge Computing Offloading Model for Distributed Wireless Sensor Devices. *J. Sens.* **2022**, *2022*, 9047737. [CrossRef]
25. Li, T.; Yang, F.; Zhang, D.; Zhai, L. Computation scheduling of multi-access edge networks based on the artificial fish swarm algorithm. *IEEE Access* **2021**, *9*, 74674–74683. [CrossRef]
26. Deng, Y.; Chen, Z.; Chen, X.; Fang, Y. Throughput maximization for multiedge multiuser edge computing systems. *IEEE Internet Things J.* **2021**, *9*, 68–79. [CrossRef]
27. Olokodana, I.L.; Mohanty, S.P.; Kougianos, E. Ordinary-kriging based real-time seizure detection in an edge computing paradigm. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 4–6 January 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
28. Yang, T.; Chai, R.; Zhang, L. Latency optimization-based joint task offloading and scheduling for multi-user MEC system. In Proceedings of the 2020 29th Wireless and Optical Communications Conference (WOCC), Newark, NJ, USA, 1–2 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
29. Talaat, F.M.; Saraya, M.S.; Saleh, A.I.; Ali, H.A.; Ali, S.H. A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 4951–4966. [CrossRef]
30. Alameddine, H.A.; Sharafeddine, S.; Sebbah, S.; Ayoubi, S.; Assi, C. Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 668–682. [CrossRef]
31. Guo, H.; Liu, J.; Zhang, J. Computation offloading for multi-access mobile edge computing in ultra-dense networks. *IEEE Commun. Mag.* **2018**, *56*, 14–19. [CrossRef]