IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 January 2023, accepted 6 March 2023, date of publication 13 March 2023, date of current version 16 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3256522

RESEARCH ARTICLE

Resource Scheduling in Edge Computing: Architecture, Taxonomy, Open Issues and Future Research Directions

MOSTAFA RAEISI-VARZANEH[1], OMAR DAKKAK[1], ADIB HABBAL[1], (Senior Member, IEEE), AND BYUNG-SEO KIM[2], (Senior Member, IEEE)

[1]Department of Computer Engineering, Karabük Üniversitesi, 78050 Karabük, Türkiye
[2]Department of Software and Communications Engineering, Hongik University, Sejong City 32603, South Korea

Corresponding author: Byung-Seo Kim (jsnbs@hongik.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korea Government, Ministry of Science and ICT (MSIT), under Grant 2022R1A2C1003549.

ABSTRACT The implementation of the Internet of Things and 5G communications has pushed centralized cloud computing toward edge computing resulting in a paradigm shift in computing. Edge computing allows edge devices to offload their overflowing computing tasks to edge servers. This procedure may completely exploit the edge server's computational and storage capabilities and efficiently execute computing operations. However, transferring all the overflowing computing tasks to an edge server leads to long processing delays and surprisingly high energy consumption for numerous computing tasks. Aside from this, unused edge devices and powerful cloud centers may lead to resource waste. Thus, hiring a collaborative scheduling approach based on task properties, optimization targets, and system status with edge servers, cloud centers, and edge devices is critical for the successful operation of edge computing. The primary motivation behind this study is to introduce the most recent advancements related to resource scheduling techniques and address the existing limitations. Firstly, this paper presents a novel taxonomy of resource scheduling in edge computing that includes applications, computational platforms, algorithm paradigms, and objectives. Secondly, it briefly summarizes the edge computing architecture for information and task processing. Resource scheduling techniques are then discussed and compared based on four collaboration modes. According to the literature surveyed, we briefly looked at the fairness and load balancing indicators in scheduling. Additionally, the survey conducted provides a comprehensive review of the state-of-the-art edge computing issues and challenges. Finally, this paper highlights deep learning, multi-objective optimization, and using green resources as key techniques for future directions.

INDEX TERMS Edge computing, resource scheduling, task offloading, fairness, load balancing.

I. INTRODUCTION

By increasing the Internet of Things (IoT) applications, more smart devices, such as intelligent sensors and smartphones, can access a network as the IoTs, leading to significant network data [1], [2], [3]. Even though their processing capacity is continually rising, they cannot meet resource-hungry applications' requirements [4], [5]. Cloud computing is capable of handling highly complicated computing tasks and services [6]. Users can rely on a cloud computing center's enormous storage and computing resources to extend their devices' computing and storage capability and quickly accomplish task processing under a cloud computing paradigm [7], [8]. Despite the convenience of cloud computing centers in providing easy access to computational resources, offloading tasks to the remote cloud would mean a significant delay in transmission, jeopardizing users' quality of experience (QoE) [9], [10], [11]; consequently, a better approach is required [12], [13]. With Cloud computing functions increasingly moving to network edges in recent

The associate editor coordinating the review of this manuscript and approving it for publication was Jie Tang.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

TABLE 1. List of acronyms used in this manuscript.

Acronym	Description
AR	Augmented Reality
BAN	Body Area Network
BCD	Block Coordinate Descent
CAV	Connected Autonomous Vehicle
CN	Core Network
DDQN	Double Deep Q-Network
DRL	Deep Reinforcement Learning
EC	Edge Computing
ED	Edge Device
EGA	Evolutionary Genetic Algorithm
ES	Edge Server
FiWi	Fibre Wireless Access Network
GN	Ground Node
IoT	Internet of Things
IIoT	Industrial Internet of Things
MD	Mobile Device
MEC	Mobile Edge Computing
NP-hard	Non-deterministic Polynomial time
PV	Parked Vehicle
QoS	Quality of Service
QoE	Quality of Experience
RSU	Roadside Unit
SLA	Service Level Agreement
SP	Service Provider
UAV	Unmanned Aerial Vehicle
UE	User Equipment
VEC	Vehicle Edge Computing
VIA	Value Iteration Algorithm

years, a new trend in computing has emerged [14]. Edge Computing (EC) is a relatively new concept in the computing world. In conjunction with fog and Mobile Edge Computing (MEC) [15], edge computing enables mobility-based and location-aware services and utilities to be delivered closer to end users, resulting in faster processing and application response time [16], [17], [18]. Furthermore, users can enjoy a higher Quality of Service (QoS) and QoE with closer Service Providers (SP) in edge computing [19]. List of acronyms used in this paper are listed in Table 1.

In the wireless network paradigm, the demand for high-quality wireless services increases exponentially as mobile communication expands, particularly 5G communication. We live in an age where the IoT plays an increasingly important role in our daily lives. A modern communication infrastructure interconnects thousands of IoT nodes, allowing them to collect and exchange data [20]. In addition to smartphones and tablets, new business scenarios are emerging in mobile network services, including autonomous driving, face recognition, and Augmented/Virtual Reality (AR/VR), as well as more practical business scenarios like smart cities and environmental sensing [21]. With the growth of these innovative services, 5G features like time delay, energy efficiency, and reliability are becoming more important. In this environment, meeting the high-performance requirements of users is challenging due to restricted bandwidth, latency, and high-power usage in the centralized architecture of cloud computing. The MEC provides computing and storage resources to the mobile network's edge, allowing it to run high-demand applications on user devices while

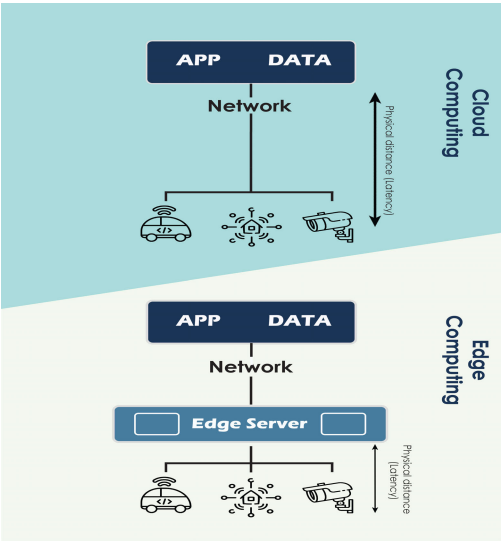


FIGURE 1. Cloud computing Vs Edge computing.

satisfying strict performance goals [22]. Through EC, cloud-based services and functionalities are brought closer to users by integrating cloud computing platforms with their networks to provide powerful processing, storage, and networking [23]. Figure 1 compares the edge and cloud computing functionalities and EC concept is depicted in Figure 2.

To achieve the foreseeable benefits of EC, it is imperative to optimize its resource utilization, which is closely related to solving the following challenges: 1) The task offloading problem, which determines where each task should be offloaded. 2) The task scheduling problem, in which tasks execution order should be decided [24]. In EC, scheduling and computational offloading policies are crucial to determining efficiency and achievable performance [25], such as latency, energy consumption, and QoS. Resource scheduling can be described as a multi-dimensional and multi-objective optimization problem and is known as Non-deterministic Polynomial time (NP-hard) [26], [27], [28]. Consequently, formulating an efficient resource scheduling algorithm poses a great challenge. On the other hand, task offloading problems are always formulated as mixed integer nonlinear programming, which is also NP-hard [29]. Several approaches have been introduced to reach a near-optimal solution (within polynomial time) for these NP-hard problems. Therefore, many survey papers in the literature have investigated the impacts of offloading and scheduling approaches on network quality features and explored the potential future directions in this era. However, most of these studies still need to include important quality metrics like fairness and load balancing, and only a few have provided comprehensive guidance for future research directions. The mentioned issues have motivated this work to fill these gaps.

The main contributions of this paper are:

1) Analyzing various elements of edge computing resource scheduling.

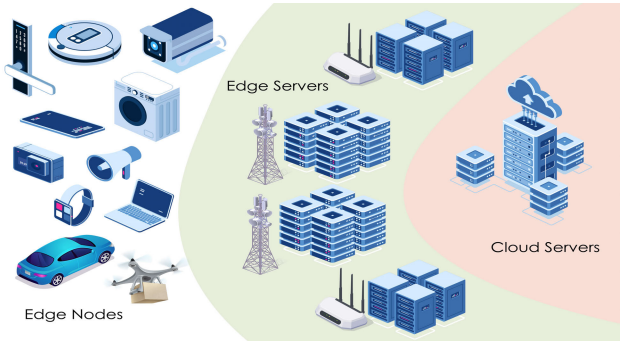


FIGURE 2. Edge computing paradigm.

2) An analysis of hierarchical edge network architecture in information and task processing.
3) An assessment of the most recent developments in resource scheduling and task offloading techniques within edge computing networks by categorizing existing works multi-dimensionally concerning scheduling objectives and collaboration approaches, particularly focusing on fairness and load balancing.
4) Identifying future unresolved or poorly addressed research challenges for improving resource scheduling and computation offloading in edge computing.

This paper proceeds as follows: Section II presents the related works. Section III analyzes different entities of edge computing networks cooperating in resource scheduling. Section IV lists the edge computing computational platforms and discusses the hierarchical multi-layer architecture widely used for task scheduling. Recent research regarding collaborative computing is investigated in Section V. A comprehensive discussion of computational offloading, including analyzing different types and reviewing recent studies demonstrated in Section VI. Fairness and load balancing, as well as methods of scheduling related to these concepts are presented in Section VII. Current issues and future directions are comprehensively examined and discussed in Sections VIII and IX. Finally, the paper concludes itself in Section X.

II. RELATED WORKS

The purpose of this section is to review recent survey papers on task offloading and resource allocation in EC. A more comprehensive review of some literature studies will then follow. Several technical terms in fog computing, edge computing, and MEC have the same functionality and meaning. Nevertheless, these technologies still maintain their generality and integrity.

Djigal et al. [30] examined how machine learning and deep learning methods are used for resource allocation in MEC from three perspectives: task offloading, task scheduling, and joint resource allocation. It investigates each perspective regarding different resource allocation objectives, such as energy consumption, latency, and QoS. It outlines a broad range of challenges and future research directions associated with applying machine learning and deep learning techniques for resource allocation in MEC.

Feng et al. [29] described different offloading modes and discusses various offloading objectives, such as reducing energy consumption, minimizing delay, maximizing revenue, and optimizing system utility. Different algorithm paradigms are used to accomplish these objectives, categorized, and discussed in detail, such as integer programming, heuristic algorithms, and game theory techniques. The paper's final part examined computational offloading in MEC networks from different perspectives and addressed its challenges and future directions.

Various collaborative resource scheduling methods are described in [19] under the architecture of edge computing. The paper discussed resource allocation, task offloading and resource providing as three main scheduling research challenges in edge computing. A centralized and distributed resource scheduling technique is discussed and compared. A literature review and a summary of leading performance indicators are also included.

Islam et al. [31] provided a comprehensive overview of task offloading schemes for MEC discussed by several researchers. A literature review has been conducted on recent research efforts on task offloading for MEC, dividing the models into three categories, computational model, decision-making, and algorithm paradigm. Differently from the above surveys, it attempted to classify and describe the architectural model involved in task offloading. A discussion was also held regarding the design challenges, potential research directions, and practical use cases related to this area.

Shakarami et al. [32] presented a classical taxonomy comprising three categories: reinforcement, supervised, and unsupervised learning mechanisms. The performance metrics, case studies, methods, and evaluation tools of these three classes are then compared. In the final section, the survey concludes with open issues and future research challenges that are uncovered or inadequately addressed.

By focusing on offloading modeling, Lin et al. [33] provided a comprehensive overview of fundamental and recent advances in computation offloading in EC. Some basic offloading models are discussed, including channel, computation, communication, and energy harvesting models. The article moves on to discuss different offloading modeling methods, such as convex optimization, game theory, or machine learning. The paper concludes with a brief discussion of research directions and challenges in offloading modeling at edge computing networks.

An in-depth literature review is conducted in [34] to reveal the state-of-the-art computation offloading at the edge. Several aspects of computation offloading are discussed, such as minimizing energy consumption, ensuring QoS, and enhancing QoE. Additionally, their review includes insight into resource allocation approaches, such as game theory and heuristics-based computation offloading optimizations. Figure 3 illustrates the contents of this comprehensive survey.

There is an in-depth comparison of the survey articles listed above in Table 2. Various aspects of these articles are compared along with their objectives. "Slightly covered,

Partially covered, and Well covered'' terms aid in better comparison, reflecting how much material each subject covers.

While most of the existing surveys in Table 2 examined resource scheduling from many perspectives, almost none focused on fairness and load balancing. This paper aims to investigate collaboration computing and computational offloading in edge computing and summarize the fairness and load balancing indicators, besides other factors, according to the literature surveyed.

III. RESOURCE SCHEDULING

Numerous Edge Servers (ES) can fulfil service requests from IoT devices in the edge environment. Each service request can be broken down into a series of tasks. In order to achieve Service-Level Agreement (SLA), the resource scheduling problem in EC determines an optimal assignment of various submitted tasks to accomplish on the edge nodes. The fundamental purpose of task scheduling is to assign the proper resources to submitted tasks [35], [36], [37], [38]. In addition to reducing energy consumption for processing and communicating, the task scheduler should satisfy the latency constraint of computation tasks [39], [40]. Although resource scheduling could be a more general term as opposed to task scheduling, they have been utilized interchangeably in the literature. In this paper, both terms determine the order of edge computational resources allocation to EU's tasks and might be seen interchangeably. Resource scheduling generally includes a set of operations and methodologies utilized to properly assign resources to the tasks and satisfy participants' objectives based on resource accessibility [41], [42]. The elements of resource scheduling can be summarized as follows:

- Various resources exist, allowing for significant serviceability and the completion of tasks in the edge network. The three main categories of resources in an edge network are communication, storage, and computation [2], [43], [44].
- A task is known as a set of instructions, data, and control information capable of being executed by computational resources to accomplish some purpose. The task categories may differ depending on the application and objective. Data from high-definition cameras on Connected Autonomous Vehicles (CAV) is used for an efficient and safe driving experience [45], [46], [47]. Fields are equipped with smart agricultural sensors to monitor temperatures and prevent fungus [48]. Body Area Networks (BAN) data is mostly utilized for the disease detection and prevention and healthcare control [49], [50], while materials, people, and places security are facilitated through surveillance camera data [51].
- Various participants – cloud servers, edge, and users (things) – cooperate through diverse collaborative strategies to complete tasks.
- Multiple objectives are pursued by different participants during task processing. In safety-related applications, for example, CAVs strive for low latency, while infotainment applications strive for high throughput [52], [53], Unmanned Aerial Vehicles (UAV) and smart health gadgets are designed to consume less energy and have longer battery life [54], and AR data requires low latency and quick data processing to ensure that accurate information is provided to the users according to the dynamic user's location and orientation [55]. AR and image-aided navigation, intelligent vehicle control, traffic management, and in-vehicle entertainment are just some of the computation-intensive applications in vehicular edge networks that require massive computing and storage resources [56], [57]. In addition to enhancing road safety and situational awareness, increasing comfort, reducing traffic congestion, lowering air pollution, and reducing costs associated with road infrastructure, users expect these networks to lead to improved road infrastructure [58], [59].
- Actions are the mechanism through which participants can attain their objectives. Edge computing has three main actions: computation offloading, resource allocation, and provisioning.
- The methodology involves the strategies, techniques, and algorithms used to carry out better the acts mentioned above to achieve the participants' goals. Two categories of methodologies can be distinguished: centralized and distributed. A control center is required for the centralized methodology to receive global data, in contrast to the distributed method [60], [61].

Although edge computing improves edge network serviceability by bringing powerful processing, storage, and communication capabilities, due to the limited resources available, task scheduling is crucial to maximizing the QoE [62]. Billions of heterogeneous user devices are scattered worldwide [63], and the quantity of information created by those end devices and their associated applications is similarly heterogeneous. Appropriate resource scheduling solutions are required to orchestrate the restricted edge resources to analyze those data better. The edge computing network includes static end-devices (e.g., smart city sensors) as well as dynamic devices (e.g., Unmanned Aerial Vehicles), making resource management even more challenging because of insecure connectivity, unpredictable resources, and dynamic computing environments [64]. Furthermore, various applications may have different QoS requirements. Video frame decoding, for example, must be completed within milliseconds to guarantee multimedia applications' performance [65]; since, reducing latency is their primary objective. Likewise, several Mobile Devices (MDs) and the IoT strive for affordable data service charges. As a result, proper resource scheduling solutions are required to meet these objectives.

Furthermore, edge computing networks are composed of several entities, including edge infrastructure, edge SPs, and

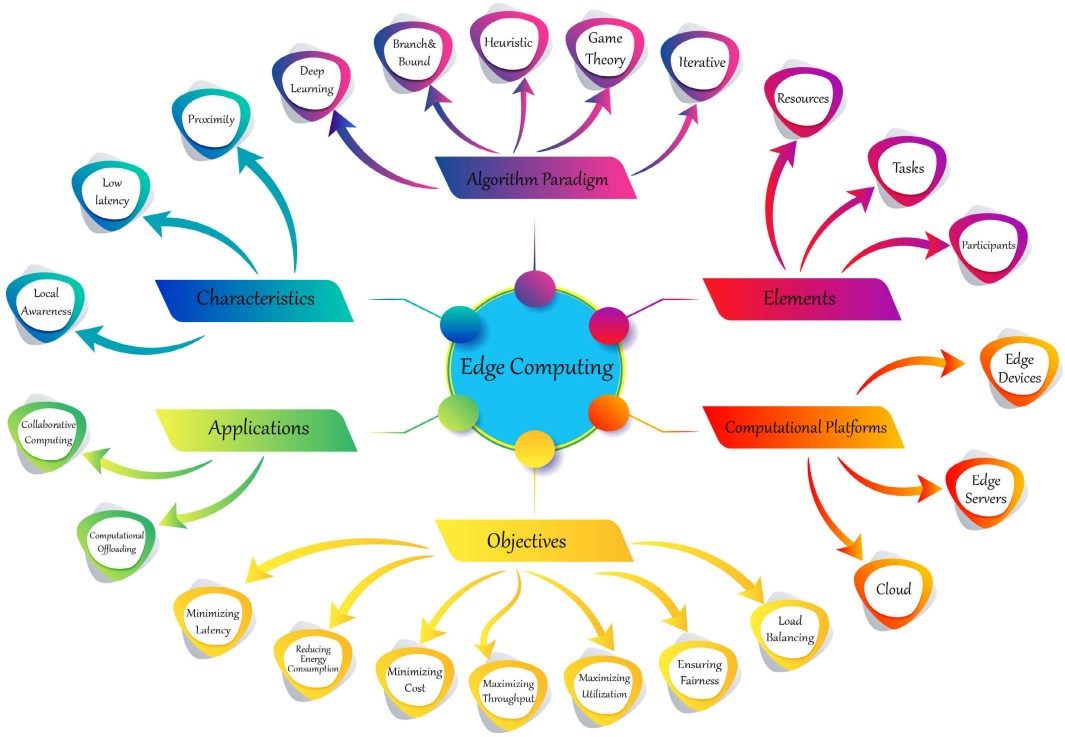


FIGURE 3. Edge computing: a taxonomy.

TABLE 2. This survey vs other surveys: comparison of edge task offloading and scheduling.

Survey	Architecture	RA	CC	CO	OM	Objectives								Algorithm Paradigm					CS	OI
						Latency	Time	Energy	Cost	QoS	Fault	Fairness	Load	ML	GT	Mathematical	LO	Heuristic		
[30]	◔	●	◑	●	◑	◔	◑	◔	◔	◔	○	○	○	●	◔	○	◔	◔	◔	●
[29]	◔	◑	◔	●	◔	◔	◑	◔	◔	◔	○	○	○	◔	◔	◔	◑	◔	◑	◔
[19]	◔	●	●	●	●	◑	◑	◑	◑	◑	○	○	◔	◔	◔	◔	◔	◔	●	●
[31]	◔	◑	◑	●	◑	◔	◑	◑	◔	◔	◔	○	◔	◑	◔	◑	◔	◑	◔	◑
[32]	◔	◔	◔	●	◔	◔	◔	◔	◔	◔	◔	○	◔	●	◔	◔	◔	◔	◑	●
[33]	◔	◑	◑	●	●	◑	◔	◑	◑	◔	○	○	○	◔	◔	◔	◔	◔	◔	●
[34]	◔	◔	◑	●	◔	◔	◑	◑	◔	◑	◔	○	○	○	◔	◔	◔	◔	◑	◑
Our Survey	◔	◔	◔	●	◔	◑	◑	◑	◔	◔	○	◔	◔	◔	◔	◔	◔	◔	◔	●

RA: Resource Allocation **CC**: Collaboration Computing **CO**: Computational Offloading **OM**: Offloading Mode **ML**: Machine Learning **GT**: Game Theory
LO: Lyapunov Optimization **CS**: Case Studies **OI**: Open Issues

○ Not covered ◔ Slightly covered ◑ Partially covered ◔ Well covered ● Comprehensively covered

mobile network carriers, in addition to users. While these SPs and operators have many resources and could provide many services, they are all business companies looking to maximize their revenue by delivering services [66], [67]. Designing a suitable resource scheduling strategy in this context can assist them in maximizing income while minimizing costs during service-providing competition. Moreover, edge resources are dispersed and distributed in the edge network. Utilizing these dispersed resources inappropriately could lead to resource dissipation. Parked Vehicles (PV), for example, have a significant portion of all vehicles on the roads and offer convenient access to computational resources that can be utilized for a wide range of computational workload [68], [69]. They can be joined to create a cost-effective and scalable computing resource center [70] that helps to relieve server workloads in the edge computing paradigm.

IV. INFORMATION PROCESSING ARCHITECTURE IN EDGE COMPUTING

Due to bandwidth constraints and severe stress on the network infrastructure, traditional cloud computing struggles to meet users' high demands for real-time response and minimal energy consumption. Nonetheless, since the edge computing paradigm lacks the same resource capacity as cloud computing, it cannot be used to replace cloud computing. Therefore, cloud computing and EC are mutually reinforcing and complementary. In most cases, edge computing is in conjunction with cloud computing to enhance and support

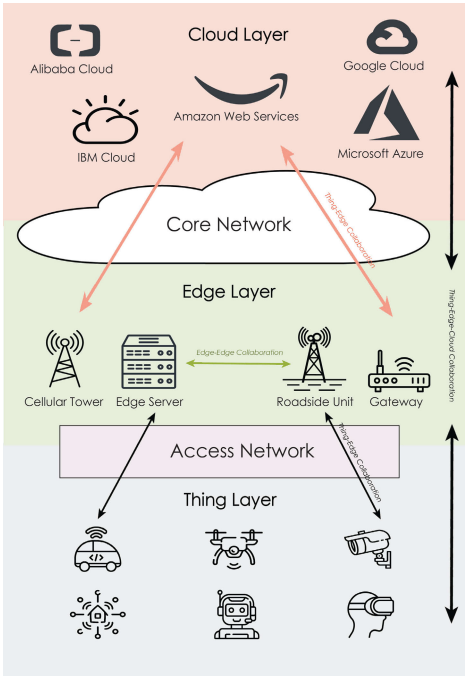


FIGURE 4. Resource scheduling architecture in EC.

the performance of end devices. Thus, resource scheduling is managed among the cloud, users, and the edge in edge computing. The OpenFog consortium defines a logically hierarchical architecture for information processing [71]. As part of edge computing's three-layer architecture, the thing layer, edge layer, and cloud layer all play a role in resource scheduling [72], as depicted in Figure 4. Many existing works [22], [45], [73], [74], [75] use this well-known and accepted architecture. This model depicts the connection between the various edge computing components. This three-tier architecture works well with applications that are both time-sensitive and computation-intensive. Tasks that require a high level of computation are handled in the cloud, while time-sensitive tasks are handled at the edge [33].

The following subsections provide a brief overview of the three-layer architecture.

A. THING LAYER

The thing layer, most often referred to as the user layer, consists of numerous heterogeneous end devices (known as things), such as smartphones, wearables (e.g., Apple Watch and Google Glass), cameras, and vehicles. There are also several ways to refer to end devices in various works, such as MDs or Mobile Users (MUs). Things are always generating and collecting text, audio, video, touch or motion-based data [76] concerning storage, processing, and interface capabilities [32]. Various devices can sense and store information, along with computational and storage capabilities. It may sometimes be possible to perform complex analytics on the end devices; however, battery life will be drained due to middleware and hardware limitations [77]. Therefore, data will be analyzed locally or sent to the edge and cloud, depending on the end devices' requirements [78].

B. EDGE LAYER

As part of the three-layer architecture, the edge layer bridges the thing and cloud layers [79]. The edge layer comprises numerous networking and computer devices, such as ES, gateway and controller. The end devices no longer need to be connected to a central cloud data center for their computing and storage-intensive tasks [80]. By connecting to large-scale and resource-rich cloud computing infrastructures within network edge and backhaul/Core Networks (CN), the need for a fast, interactive response can be addressed, ensuring low latency and fast connections [81]. Typically, the thing layer does not offer as many sophisticated storage or computing capabilities as the edge layer. Most data storage and computation will take place in this near-end environment in edge computing. The resource scheduling procedure occurs at this layer. A scheduler coordinates all edge devices, selecting and allocating resources at the edge layer. Scheduler determines which application modules should be offloaded to the upper layer and which should be placed at the edge layer based on defined scheduling policies. It is the edge layer's responsibility – particularly edge scheduler – to manage edge resources to facilitate quick task response and minimize CN bandwidth utilization, power consumption, and communication overhead.

C. CLOUD LAYER

A cloud layer comprises pre-existing computing entities, such as processing units and storage devices, all connected to a CN, giving them access to the edge layer. The cloud layer has the most sophisticated computational and storage center SPs among all three tiers providing centralized computing, integration, and analysis [82]. While ESs can manage vast requests to reduce delay and power usage, the edge computing paradigm still relies on the cloud's processing power and large storage capacity to perform complicated tasks [83]. In addition, the cloud layer could also dynamically manage resource scheduling algorithms based on network resource distribution.

The following Section explains different collaboration paradigms in edge computing resource scheduling of computing tasks in an EC paradigm.

V. COLLABORATION RESOURCE SCHEDULING

In contrast to cloud computing, some users' applications can be executed on ESs close to end devices, dramatically decreasing data transmitting latency and minimizing workload in edge-cloud collaboration [84]. Edge computing gains an additional advantage by preventing long-distance data transmissions and ensuring user data security more effectively. The traditional task scheduling in edge computing involves offloading all processing tasks from user devices to be processed on ESs. However, it may lead to wasting processing and storage resources in Edge Devices (ED)

and cloud computing centers. Furthermore, a large number of users can lead to a lengthy task queue due to the finite computing resources at edge computing servers [85]. To overcome the following issues, we may use task offloading to integrate the cloud and ESs with EDs to control the computing tasks of EDs effectively. It can assist in decreasing the load on ESs, enhancing resource usage, and decreasing average task completion time on the edge.

A. THINGS-EDGE COLLABORATION

The things-edge collaboration takes place between the things and the edge layer. ESs can overcome the limitations of end devices computing capacity [86]. It is possible to execute smart device tasks locally or to offload them to ESs. Offloading decisions are based on collaboration methods, as well as smart device QoS and QoE requirements. Ali et al. [87], for example, recommended offloading the optimum set of computation tasks to ESs to reduce MD energy usage. Wang et al. [88] indicated that offloading tasks to the nearby ES site could reduce overall system expenditures and ensure users' QoE. To address these issues and to enhance resource and system utilization, Liu et al. [89] investigated a Vehicle Edge Computing (VEC) paradigm and considered vehicles as vehicular ESs to facilitate other fixed ESs in task processing. Using UAVs as ESs is also a research topic [90], [91], [92]. Yang et al. [93] examined a UAV cooperated MEC paradigm, in which UAVs process MU computation tasks to reduce the system's total power consumption. Unlike prior research in which users transmit the tasks to ESs and subsequently give the results back, Chen et al. [94] studied relay-assisted computation offloading (RACO). A MEC-enhanced relay platform referred to (MERS) is used in the RACO scenario to help users share the outcomes of computational tasks by assigning computational and communication resources.

B. THINGS-EDGE-CLOUD COLLABORATION

Despite its great potential, the things-edge collaboration method neglects the massive computing resources provided by the Cloud servers. In the age of smart devices and applications that consume a great deal of resources, focusing entirely on edge layer resources to meet smart device service requirements will become increasingly challenging. As a result, it is critical to fully utilize Edge and cloud computing to develop a collaborative network. Through the adoption of hybrid Fibre Wireless Access Network (FiWi), Guo et al. [95] presented a generic architecture and proposed a distributed collaborative computation offloading scheme by adopting the game theory. Integrating cloud computing FiWi and edge computing provides great scalability, high mobility and reliability, supporting various wireless access technologies, and low latency. A multi-hop IIoT-edge-cloud collaborative computation offloading paradigm is proposed in [96] for resource-intensive applications to reduce power consumption and task processing time. The concept of ''heterogeneous multi-layer MEC'' is proposed in [97]. Suppose ESs cannot provide an acceptable completion time for an offloaded task. In that case, it could be transmitted to the cloud center in HetMEC to reduce communication and computation time. Dwelling on the same issue, but within a different context, Dinh et al. [98] regarded renting virtual machines from the cloud to extend the edge layer, thereby reducing the processing cost at the edge and the cost of using clouds VMs.

C. EDGE-EDGE COLLABORATION

Collaborative computing between the edge-edge infrastructures is a current research hotspot, which can solve the contradiction between limited resources in a single ES for intelligent computational-intensive applications and long transmission time for communication-intensive tasks to use resourceful cloud servers [99]. In general, the edge-to-edge collaboration paradigm does not occur in isolation. It is often linked to the things-edge or things-edge-cloud collaboration models. There is another option for task processing through edge-to-edge collaboration. This type of partnership has been the subject of numerous research. Huang et al. [68] proposed the concept of Parked Vehicle Edge computing (PVEC) as a new computing paradigm, which allows idle PVs' to be effectively used. VEC servers in a PVEC architecture seek appropriate resources from PVs to serve workloads. Na et al. [100] proposed using edge gateways to facilitate task processing in the edge layer to reduce the workload on ESs. In order to enhance the efficiency of IoT systems, a resource coordination method between EGs and ES is also proposed. Alameddine et al. [24] presented a novel approach to address the task offloading, application resource allocation and the task scheduling problems in a MEC network, where the task assignment of applications and the sequence in which they are executed are both taken into account. To meet User Equipment's (UE) QoE requirement, tasks that cannot be handled by their corresponding ES could be transmitted to another ES. Miao et al. [101] suggested an intelligent offloading technique, in which tasks are distributed across MDs, ESs, and the cloud, to decrease overall task latency. Furthermore, under this technique, the ES can choose to share its overload with other ES via edge-edge collaboration. On the other hand, Thai et al. [102] offers horizontal and vertical collaborations in a cloud-edge computing paradigm to reduce the overall cost. Horizontal collaboration refers to offloading tasks between nodes in the same layer, whereas vertical collaboration corresponds to offloading tasks between nodes in different tiers.

D. EDGE-CLOUD COLLABORATION

Significant delay will be obtained if many processing requests are conducted in the cloud computing center in the proposed three-layer architecture, which violates users' QoE. Rimal et al. [103] showed task delay problems can be mitigated by transmitting some tasks from cloud centers to the edge. Many applications can benefit from edge-cloud collaboration. Mobile client shopping, for example, has grown in popularity, with clients frequently operating the shopping basket. The shopping basket status is changed in

TABLE 3. Resource scheduling: A comparison of different collaboration methods.

References	Collaboration	Research Topic		Features	Methodology
		Task Offloading	Resource Allocation		
[87]	Things-edge	✓	✗	Selecting an optimal number of processing tasks to transmit to ESs to reduce MD energy consumption	Deep learning
[88]	Things-edge	✓	✗	Consider the task offloading problem as a game	Game theory
[89]	Things-edge	✓	✓	Regarding stochastic vehicle traffic, dynamic computing demands, and communication circumstances that change over time	Reinforcement learning
[93]	Things-edge	✗	✓	Enhance user association, power management, resource assignment, and location planning all at the same time	Compressive sensing, search method
[94]	Things-edge	✓	✓	Optimize the transmission power, processors speeds, bandwidth, and offloading ratio together	Iterative algorithm
[95]	Things-edge-cloud	✓	✗	Reduce the energy usage of all MDs while meeting the MDs' computation execution time restrictions	Game theory
[96]	Things-edge-cloud	✓	✗	Optimize energy usage, and processing time	Game theory
[97]	Things-edge-cloud	✗	✓	Consider the cost and capacity of local processing at the edge, as well as the cloud's different rental alternatives	Offline and online algorithms
[98]	Things-edge-cloud	✗	✓	Coordination of communication and computing resources in different layers for task management	Latency minimization algorithm
[68]	Things-edge and edge-edge	✗	✓	Using idle PVs	Stackelberg game, iterative algorithm
[100]	Things-edge and edge-edge	✗	✓	Consider ES and EG computing capacities, as well as EG interference	Lagrangian
[24]	Things-edge and edge-edge	✓	✓	The UEs tasks are distributed among several ESs	Decomposition technique
[101]	Things-edge and edge-edge	✓	✗	AI and local computing have been integrated	Deep learning
[102]	Things-edge and edge-edge	✓	✗	Offloading vertically and horizontally; the workload optimization issue has been addressed	Branch-and-bound method
[107]	Things-edge-cloud and edge-cloud	✗	✓	Latency-aware task scheduling in edge layer	Auction-based contracts
[108]	Things-edge-cloud and edge-cloud	✗	✓	Tasks from UEs are assigned to SPs at the edge layer to be processed in the base station or cloud centre	Decentralized resource allocation

the cloud center for the first time, and then the product view on the MD is updated, resulting in considerable latency. The video transcoding application is another example. MUs need high QoE for video streaming, and online video traffic on MDs is expanding network traffic exponentially [104], [105]. Video transcoding has evolved into an efficient method of transmitting video data. Video transcoding, on the other hand, requires a lot of computing and storage resources; thus, it is usually done on a remote offline media server. Unfortunately, during video streaming, the redirecting latency may increase, and the real-time streaming service may be inaccessible. Yoon et al. [106] recommended running video transcoding on edge nodes. Experiments have shown this approach to be efficient, scalable, and transparent. Furthermore, Xu et al. suggested in [107] that EC services can be provided by micro data centers at the edge layer. A Zenith paradigm was also presented, where resource utilization and task execution time could be handled efficiently by service and infrastructure providers' collaboration. Likewise, Zhang et al. [108] proposed that SPs be deployed in the edge layer to handle MU task processing. In an edge-cloud collaboration, SPs can deliver high-quality services by offloading tasks to the cloud or edge while optimizing the benefits of all SPs. Table 3 summarizes a review of articles concentrating on different collaborative methods for resource scheduling.

VI. COMPUTING TASK ANALYSIS

The computing tasks are then evaluated to guarantee the desired results, such as the shortest task completion time and the least energy usage, by effectively allocating them to the right node. This procedure is known as task offloading. In practice, edge-cloud computing must address task offloading as a critical challenge, as it decides when and where tasks are performed [109]. Many benefits can be gained from it, including prolonged battery life, reduced latency, and better performance [110]. A decision must be made on whether a task can be divided and whether or not subtasks are interdependent based on task attributes [111]. Simple or highly integrated tasks cannot be separated and must be processed locally at corresponding EDs or totally

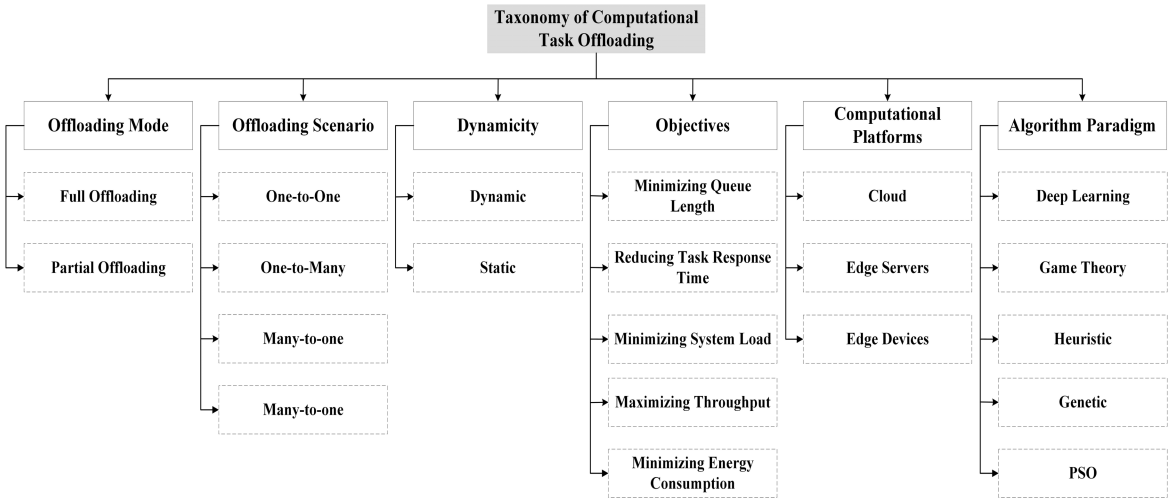


FIGURE 5. Offloading computational tasks at the edge: a taxonomy.

transmitted to ESs. They can be separated into many pieces and offloaded if they can be subdivided based on their code [112], [113]. Locally performed computing-intensive tasks result only in delays caused by local processing. The delay, however, represents the sum of the time it takes for data to be transmitted to the ES, the time it takes for the ES to process the data, and finally, the results transmission time. Therefore, offloading computing-intensive tasks to ESs directly impacts user QoS [29]. Figure 5 depicts the taxonomy of computational task offloading in the edge environment.

In summary, there are three modes for given computing tasks [114]: local execution, partial offloading, and full offloading as illustrated in Figure 6. Based on the current network state, a specific offloading site should be selected to process the tasks. Tasks offloading is meant to decrease application execution times and reduce UE energy consumption [31]. The analysis of different offloading scenarios is summarized in Figure 7.

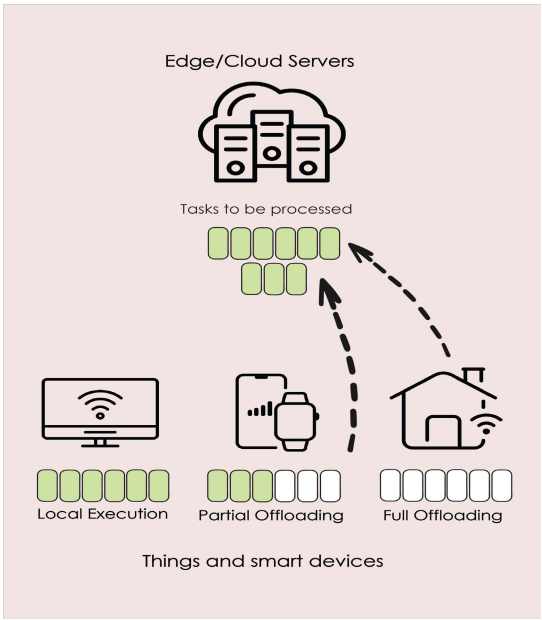


FIGURE 6. Different offloading modes.

A. LOCAL EXECUTION

To determine whether edge computing tasks should be performed locally, EDs' resources and the network and resource conditions of ESs should be considered [115]. The computing task can only be completed locally if sufficient network bandwidth is unavailable to transmit a task successfully. Furthermore, if the computing resources of ESs are unavailable, causing the computing tasks to be delayed, the tasks must be completed locally. If an ED's computational capacity is sufficient to meet service requirements, it executes tasks locally, minimizing the ES workloads and network bandwidth demands [12].

B. FULL OFFLOADING

Full offloading (Also known as binary offloading) is a method that allows relatively simple or highly integrated tasks to be executed either locally at the ED or offloaded completely to the ES. In order to determine whether or not edge computing tasks are totally offloaded to an ES or scheduler, the resources of EDs, the existing network, and the availability of ES resources must be examined [73]. Suppose the currently available network bandwidth ensures successful task transmitting, or the ESs or other ED are idle, and the successfully offloaded computing tasks can be processed immediately. In that case, the results of local execution and full offloading are compared to deciding on task offloading or local execution. If the objective is to reduce the time it takes to complete a task, for example, it is vital to compare the local execution time with the time it takes to offload to an ES/cloud computing center. Tasks should be processed locally if the local execution time is less. Otherwise, they should be processed on edge or cloud servers.

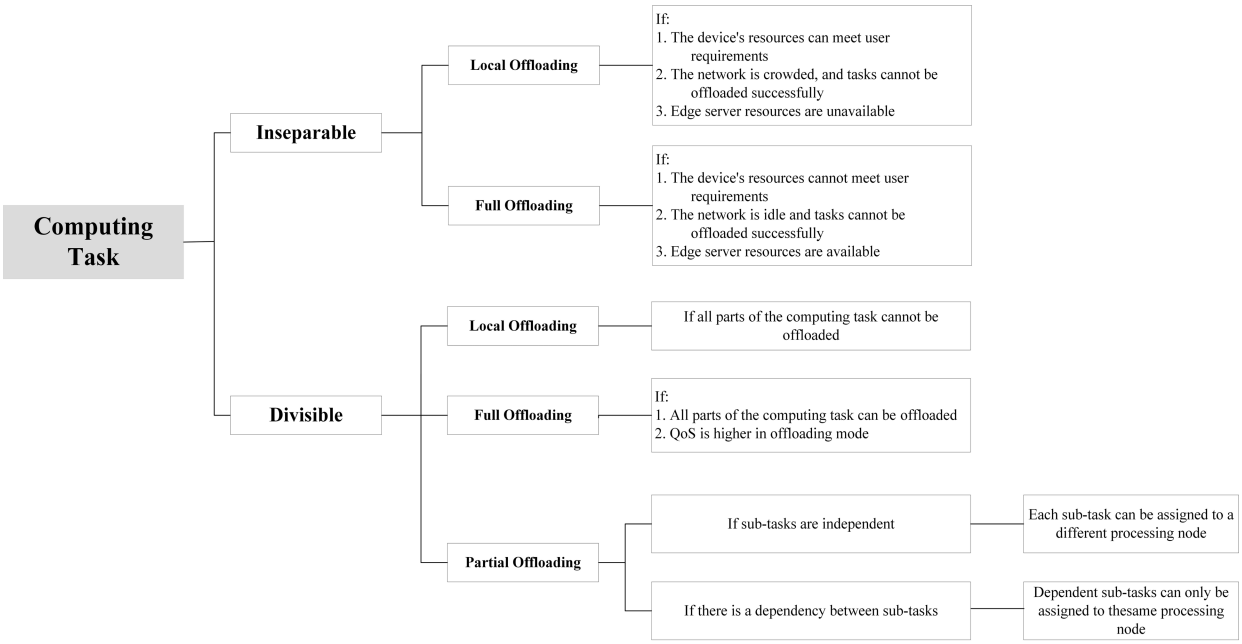


FIGURE 7. Different offloading scenarios.

It has been stressed in numerous studies that each user request should be fully offloaded and handled by a single ES in MEC networks. To reduce average task response time and total system energy consumption while ensuring task offloading performance, Tong et al. [116] proposed an integrated trust evaluation mechanism in Deep Reinforcement Learning (DRL) in combination with a Double Deep Q-network (DDQN) algorithm. Materwala et al. [117] proposed an offloading algorithm based on the Evolutionary Genetic Algorithm (EGA) to optimize the energy consumption of edge and cloud servers simultaneously in vehicular networks by maintaining the application's SLA concerning latency and processing time. Tang et al. [118] offered a method for joint optimization based on the discrete binary particle swarm optimization algorithm (JOBPSO), designed to minimize the system's energy consumption, load and delay. Using deep Q-learning, Tang and Wong [119] proposed a distributed algorithm to address the unknown load dynamics at edge nodes. This approach allows each MD to make its own offloading decisions without being aware of other MDs' information (e.g., task models and offloading decisions). Dropped task ratio and average delays can be reduced with the proposed algorithm. Based on a Markov decision process, Yang et al. [120] developed a strategy to select optimal offloading nodes by applying a Value Iteration Algorithm (VIA) for addressing delay-sensitive task offloading in a MEC system, concerning task uploading time, task execution time, and results downloading time. Energy-efficient task offloading problem in MEC formulated in [121] as a stochastic optimization problem to minimize energy consumption while maintaining average queue length. This paper proposed an energy-efficient dynamic offloading algorithm called EEDOA by transforming the stochastic problem into a deterministic optimization problem. The decision of task offloading can be made dynamically with polynomial-time complexity.

C. PARTIAL OFFLOADING

Leveraging parallelism between the network and end devices makes partial offloading more suitable for time-sensitive applications than full offloading. Additionally, since the communication network's bandwidth is limited, offloading partial applications is more reasonable than offloading the entire application [122]. It is beneficial to offload partial segments of a complex task composed of multiple parallel segments in order to enhance efficiency [123]. It is possible to partition the program into two portions, one being executed on the ED and the other offloaded for ES execution while scheduler tries to assign an optimal processing unit to each sub. Xu et al. [124] proposed an algorithm based on game theory to minimize the energy consumption of each MD and meet the delay constraints in the cloud-edge-terminal collaboration scenario. Partially offloading tasks under dynamic conditions, including energy consumption, transmission rate and task arrival, is investigated in [125]. It introduces a feasible method for characterizing dynamic factors in offloading tasks called the Time-Expanded Graph (TEG). Lastly, practical algorithms based on TEG are devised to maximize IoT terminal utility under surplus energy constraints in both singular and multiple terminal environments. Yu et al. [126] developed a UAV-assisted cooperative offloading system that utilizes Block Coordinate Descent (BCD) to minimize energy consumption by user terminals. Methodologically, the problem is decomposed into three convex subproblems, and the number of local computation tasks, number of computing offloaded tasks

and trajectories of UAV are further optimized. In order to minimize energy consumption and task response time, Wu et al. [127] developed an energy-efficient dynamic task offloading algorithm. By applying the Lyapunov optimization technique, different applications and dynamic scenarios can be controlled regarding computation and communication costs. An online and polynomial-time complexity algorithm has been developed by leveraging the Lyapunov optimization technique to formulate the offloading decision problem as an optimization problem. Using the Lyapunov optimization technique, determine where and whether to offload in such a way that the IoT device's energy consumption is reduced by only sacrificing some delay. Table 4 compares recent studies on full and partial offloading.

VII. FAIRNESS AND LOAD BALANCING

Fairness and load balancing can be mentioned among the scheduling objectives that have received less attention in previous articles, particularly in survey papers. This section briefly reviews these two factors and the related scheduling methods. The concept of fairness has been explored from several social justice and welfare economics perspectives, including those advanced by Nash Jr [128] and Rawls [129]. As per Rawls's theory of justice, resources should be allocated to the least fortunate users to increase satisfaction and utility to the greatest extent possible. The Nash bargaining theory offers an alternative view, which states that resources should be reallocated to those who will benefit more from them than those who lose them.

Schedulers must be fair as one of their essential characteristics. Schedulers were traditionally designed to share resources between processes fairly, which had been interpreted in the context of tasks. Schedulers must treat users fairly rather than just tasks, as recently realized [130]. Through a fair scheduling system, competing users get their fair share of resources over a long period of time. When dealing with connection and disconnection from Roadside Units (RSUs) as the vehicle moves from one RSU to another, Strugeon et al. [131] proposed an agent-based approach to ensure long-term fair allocation of computing resources in the autonomous driving domain. In order to assess whether resource allocation is fair, a fairness ratio is employed. Zhao et al. [132] formulated the UAV's energy consumption minimization problem as a mixed integer nonlinear programming problem, where the UAV's trajectory, resource allocation, task decision, and bits scheduling are jointly considered to ensure fairness among Ground Nodes (GN). In addition to protecting the UAV from being overwhelmed by the data from one or more GNs, this approach guarantees fairness across all operations, such as local computing, task offloading for computing and caching, and avoids the situation where the UAV caches all offloaded bits. Xu et al. [133] developed a job scheduling algorithm based on Berger's distributive justice theory in cloud environment. A fair scheduling model with trust enhancement for the cloud-fog-edge environments was described in [134]. Blockchain technology is used to build a decentralized trust framework to address the service credibility. Lastly, it proposed the concept of service fairness and an evaluation method based on Berger's theory of wealth distribution. A scheduling strategy was proposed by Mukherjee et al. [135] to maximize the number of tasks completed within their corresponding deadlines in a dual queue system in edge/fog environment while keeping both queues strongly stable. The queued tasks were scheduled using the Lyapunov drift-plus-penalty function. The scheduling policy decides how many tasks should be offloaded to underloaded nodes to take advantage of all the computational resources in the network.

The literature on mobile computation offloading has studied fairness, most of which focused on energy consumption and some on the distribution of resources. Multicast-based computation offloading by device-to-device is designed in [136] to ensure fairness based on each MU's delay constraint and battery level. MIMO-based MEC systems examined in [137] by analyzing the min-max power consumption of MDs in offloading computation tasks and optimizing resource allocation. Mao et al. [138] discussed the max-min energy efficiency optimization problem in wireless powered full-duplex MEC systems, which aims to achieve fair energy efficiency among multiple mobiles through the optimization of computation offloading and resource allocation. To ensure fairness in wireless-powered MEC, Ji et al. [139] studied max-min energy efficiency problem, where energy efficiency is defined as the amount of energy harvested per unit of the user throughput. Zhou et al. [140] distributed computing resources proportionally according to the weights of the live jobs on each server to ensure fairness. Furthermore, Kim et al. [141] and Zeng et al. [142] assessed fairness in application throughput and data transmission rates.

Since users consume different resources, the system is prone to load imbalances if resource allocation is unfair [143]. It is beneficial to utilize the load balancing feature at the edge network level in order to reduce latency, energy utilization, and bandwidth consumption [144]. It is possible to interpret fairness at the server level as load balancing. Whenever some edge nodes are overloaded, computing tasks become more challenging, so the load should be passed to less overloaded nodes. Edge nodes require load balancing to minimize resource consumption and response times, thereby increasing resource utilization [145]. In this way, communication overhead is further reduced at the edge, enhancing the whole process [146]. For vehicular applications, Dai et al. [147] proposed integrating load balancing with offloading. It formulates the joint load balancing and offloading problem as a system utility maximization problem with permissible latency constraints. Through a combination of joint optimization of selection decisions, offloading ratios, and computation resources, the joint algorithm for selection decision, computation resource and offloading (JSCO) algorithm was then developed with a low complexity. A study was conducted to minimize processing delays in FiWi-enhanced vehicular edge computing networks

TABLE 4. Comparison of papers focusing on different offloading techniques.

References	Collaboration	Research Topic	Performance Metrics	Methodology	Dynamicity	Servers/Users	Experimental Environment
[116]	UE and ESs	Full Offloading	System energy consumption, average task response time, and task offloading and execution success rate	DDQN	Dynamic	Single User / Multiple Servers	Python 3.6
[117]	Edge and cloud servers	Full Offloading	Energy consumption of the edge–cloud integrated computing system and SLA violations	EGA	Static	Multiple Users / Multiple Servers	N/a
[118]	MD and ES	Full Offloading	Response time, MD energy consumption and system load	Discrete binary PSO algorithm	Static	Single User / Single Server	MATLAB
[119]	MD and ES	Full Offloading	Average delay and Ratio of dropped tasks	DRL	Dynamic	Multiple Users / Multiple Servers	RMSProp optimizer
[120]	MD, edge, and cloud	Full Offloading	Offloading time	VIA	Dynamic	Multiple Users / Multiple Servers	Python 3.6
[121]	IoT devices and ES	Full Offloading	Transmission Energy Consumption and Queue Length	Lyapunov optimization technique	Dynamic	Multiple Users / Single Server	N/a
[124]	MD, edge, and cloud	Partial Offloading	Energy Consumption	Game theory	Static	Multiple Users / Single Server	JAVA
[125]	IoT devices, edge, and cloud server	Partial Offloading	Throughput, Utility and Running Time	Heuristic	Static	Single Or Multiple Users / Multiple Servers	N/a
[126]	UE and UAV	Partial Offloading	UE Energy Consumption	BCD	Static	Multiple Users / Multiple Servers	MATLAB R2019a
[127]	IoT device, MEC and cloud servers	Partial Offloading	Energy Consumption of IoT Devices and Task Response Time	Lyapunov optimization technique	Dynamic	Single User / Single Server	N/a

in [148]. This paper proposes an architecture for task offloading based on SDN and formulates the offloading problem as a constrained optimization problem. It solved this problem, using game theory-based offloading schemes in order to balance the workload of the MEC servers. A Deep Reinforcement Learning Based Resource Allocation (DRLRA) scheme, proposed in [149], enables adaptive allocation of computing and network resources, thereby reducing average service times and balancing resource usage in varying MEC circumstances. A comparison of the proposed DRLRA method to the classical Open Shortest Path First (OSPF) algorithm revealed significantly better performance. Xu et al. [150] proposed a blockchain-based computational offloading technique named BeCome to decrease edge computing devices' offloading time and energy consumption. In this work, Blockchain technology is utilized in tandem with nondominated sorting genetic algorithms for generating optimal resource allocation strategies in EC. Yang et al. [151] constructed a multi-UAV-aided MEC system, which provides computing offloading services for ground IoT nodes without sufficient computing power. The access problem is modelled as a generalized assignment problem and is solved with near-optimal algorithms using differential evolution-based multi-UAV deployment. This allows IoT nodes to balance their load while maintaining coverage constraints and ensuring their QoS.

VIII. OPEN ISSUES

The edge computing paradigm provides lower task communication delays than cloud computing because ESs have more powerful processing and storage resources. Furthermore, the task scheduling issue in edge computing is NP-hard due to the scarcity of edge resources. Providing high-performance approaches are extremely important, but it is impossible to achieve a precise global optimal solution for large problems in general. Despite the huge number of studies on collaborative task offloading and scheduling in EC, the following concerns need be addressed.

A. SECURITY

Edge Computing poses significant security risks compared to traditional cloud computing. EDs will face many new threats that cannot be managed with the security solutions of cloud computing [152]. Security in resource scheduling means providing mechanisms such as confidentiality, integrity, availability, access control, and authentication between EDs and servers to protect the scheduling and computation offloading process. Due to its multi-tier architecture, EC is prone to hostile attacks regarding resource scheduling [153]. Information may be stolen or tampered with by attackers accessing ESs. Eavesdropping and traffic injection attacks are also ways malicious attackers can gain control over communication networks [154], [155]. In an attack on an

edge node, the system integrity and robustness of the entire edge system may be compromised, rendering the resource scheduling ineffective. Thus, resource scheduling in edge networks must be subjected to further security research. It is common to find security issues related to cloud computing in this context as well. However, the limited number of EDs and wireless access are specific characteristics of EC that make security mechanisms more challenging. Edge computing does not have sufficient studies on security threats; therefore, it can be considered an open issue. A practical approach must be developed to predict, prevent, protect, and recover the edge network in case of catastrophic events.

B. PRIVACY

Regularly, ESs return confidential information in their computing results. For example, a patient's personal information may be included in the analysis data retrieved from the ES in intelligent healthcare. Therefore, it is crucial to take privacy protection into account. Unauthorized users threaten end-user privacy by leaking information about their location, usage, and data through IoT networks. A large amount of private information is involved in resource scheduling, especially computation offloading. The existing research fosters unconditional trust and easy access to user data, EN interactions, and computing data at the edge [156]. Many layers of protection must be applied at the network edge or the cloud to protect this huge amount of data generated by EDs. The privacy problem, however, cannot be circumvented by existing resource approaches. To maintain data integrity and privacy, scheduling approaches should account for the risk of data replication/sharing attacks, data altering attacks, and data loss at the edge or cloud level in the future.

C. HETEROGENEOUS TASKS AND RESOURCES

There has been a tendency among studies to assume that the target edge computing system has a bounded number of similar computing resources. However, the use of hybrid architectures, such as multiple cores and GPUs, has become more common in high-performance computing systems. In order to support the growing number of IoT applications in the future, SAGIN is predicted to be the future trend by integrating multidimensional networks like space, air, and ground [157]. New resource allocation issues have emerged as a result of these novel architectures. In most previous studies on resource scheduling, computation tasks were also assumed to be homogeneous. Scheduling models can be oversimplified with this assumption [158]. There are, however, different tasks that need to be accomplished in practice. Some tasks are pre-emptive (i.e., they could pre-empt other tasks), while others are not. As a result of this heterogeneity of tasks, scheduling becomes significantly more difficult. To cope with hybrid tasks and resources, modern mechanisms must be investigated.

D. MOBILITY

Due to the mobility of most connected things, including MDs, vehicles, and drones, the IoT suffers from frequent link failures among devices and servers. This problem compromises QoS and security of edge systems [159]. The support of high-mobility devices is a critical issue for future networks [160]. Maintaining the connection with the ES despite leaving the area of origin to receive high-dynamic services is very important [161]. Reaching an effective task scheduling and offloading mechanism is highly challenging when users are frequently mobile. In some cases, offloading decisions made at this moment may not apply in the future, or the node may no longer be within the service range of the user [45]. Current research rarely explores users' mobile characteristics in different application scenarios, and most studies idealize and disregard mobile characteristics.

E. SCALABILITY

The scalability of a system refers to its ability to provide elastic services efficiently without compromising QoS. Network architectures must be scalable to handle the growing demands, requests, and services, such as MDs in edge networks. There has been a rapid increase in the number of connected things, which may impair the QoS and cause network bottlenecks due to an enormous amount of data being generated [84]. Some mobile applications require high data rates' offloading, such as AR, VR, online gaming, and self-driving. Despite the heterogeneity of MDs and the dynamic behaviour of requests from the applications mentioned in the edge computing environment, edge computing systems should be scalable in terms of the number of servers and services required.

F. FAULT TOLERANCE

A high functional failure probability is always associated with EDs due to their distributed nature. Thus, there are many possible causes for device failure, including hardware failure, software failure, and user error. Furthermore, connectivity, mobility, and power source also play an important role. The ability of a system to continue working despite faults is referred to as fault tolerance [162]. While the concept of fault-tolerant computing refers to using systems that can perform correctly in the presence of errors [163]. Any failure of a central controller will result in retransmission in all computing paradigms. The retransmission of a submitted task when it fails causes delay because the task is again offloaded to the same or a different host. A scheduling system must be highly reliable to cope with disasters and bad situations because it is impossible to offload applications without faults.

G. ENERGY MANAGEMENT

Due to the distributed nature of edge computing, energy consumption will be high, increasing costs. Therefore, a new energy protocol for edge computing systems needs to be optimized and developed to address this issue [164]. Another area of future research is the challenges associated with greenhouse gas emissions and carbon emissions, which were not addressed by most of the techniques studied. With the help of green resources, such as light and wind, IoT devices

can significantly reduce carbon emissions and pollution while consuming less power and expecting longer battery life. In the field of energy harvesting and wireless charging-enabled edge computing, many studies are being conducted [123], [165]. It is more complex to schedule resources when extra energy supplements are introduced since energy consumption during task transmission, task processing, and the harvested energy must be considered.

H. LOAD BALANCING AND TASK DISTRIBUTION

The task distribution or load balancing process is perhaps the most challenging step in scheduling. In order to obtain the best scheduling performance, several parameters and metrics must be considered. According to various descriptions, it is an NP-hard problem to determine the optimal way to distribute tasks to reduce application processing time and minimize failure rates [166]. Therefore, it can take considerable computing resources to distribute these tasks intelligently and efficiently. It may be possible to manage edge networks' dynamic and heterogeneous characteristics with hybrid load balancing techniques. The development of load balancing mechanisms in edge networks is obvious in light of the evolution of IoT-based applications. Most existing load balancers do not consider fault tolerance mechanisms in an edge environment. In order to avoid a disruption in overall performance the ESs, it must be possible to detect failed edge nodes and send requests to the nearby ESs. Aside from response time and overall cost, energy consumption is another key factor that potential researchers could address. Due to the dynamic nature of IoT-based applications, the energy requirements of ESs are substantially increasing. This means that load-balancing approaches must be able to cope with energy consumption. A load balance between all devices on the network remains an open issue from the macro perspective. Task processing requests should not be directed only to the most powerful devices on a network or should not be directed only to a few network nodes. Therefore, algorithms should be designed to utilize the computational and network resources to the fullest extent possible. Several devices can be overloaded, and many can be idle simultaneously if they are not managed properly, which can negatively affect the performance of networks and applications.

I. INTEROPERABILITY

Edge computing encounters interoperability issues due to heterogeneous platforms, architectures, and infrastructure [167]. In order to deploy edge computing successfully, interoperability is the major challenge [168]. Load balancing in edge domains is difficult by the nodes and servers' variety and distribution. Interoperability is, therefore, an essential component of success. There are many SPs out there. Therefore, consumers search for their favorites as well as looking for important factors like price and functionality. Through interoperability, consumers can change between IoT/edge products or combine different services and products to create smart environments tailored to their needs [169]. For interoperability to be possible, an intermediate interface and controller must be provided to facilitate communication between the system's entities. Developing such interconnections between EDs and servers requires a flexible architecture and system model. To successfully fulfill offloading, researchers must develop new methods for addressing interoperability issues directly due to the highly dynamic behavior of MEC environments with high data rates on the one hand and heterogeneity on the other.

J. COST AND PRICING

Edge networks dynamically allocate resources for storage, computing, and communication, based on the demands of users. Due to these differences, optimal pricing policies differ from legacy pricing policies. It is common for edge environments to have multiple actors offering services at different prices. In addition to different operating models, management models, and policy approaches, these actors also have different payment methods. When customers care about the price, the pricing policy significantly affects the edge and cloud's profit [170]. The rational person hypothesis [171] states that suppliers want to maximize their profits, while purchasers want to obtain the best service quality at an affordable price. Higher prices generally increase profits, but lower purchase intentions, which are also impacted by commodity quality. The existence of competitors also influences transaction probabilities. Thus, for systems stakeholders, a balanced profit margin is desirable from a commercial perspective which can achieve by frequently updating the data offloading price and edge network profit model in response to network conditions. There is still much work to be done on the issue of balancing users' costs with SPs' revenues.

K. SUBTASKS DEPENDENCIES

As ESs have limited computing resources, larger tasks are usually divided into interconnected subtasks offloaded to remote cloud servers. This allows multiple ESs to serve users collaboratively. Resource scheduling presents challenges in minimizing the time to complete a task due to dependency on its subtasks and the dependency between the input of one subtask and the output of another [172], [173]. Specifically, subtask resource allocation should consider not just the number of allocated resources but also the starting point of the allocated time slot. A user's task may also contain multiple online requests [120]. To guarantee concurrent execution of various tasks, resources should be allocated to subtasks according to the supertask delay constraint.

L. PARTITIONING AND INTEGRATION

Granularity and partitioning of offloaded code are the factors relating to the size of the code that can be offloaded to run remotely to make the application run more efficiently under time or resource-constrained conditions. When designing the life cycle of an application, some aspects of granularity

TABLE 5. List of challenges and potential future direction with their hotness and trends.

References	Category	Challenge	Challenge Hotness	Research Trend
[2, 38, 106]	Cost and pricing	Costumers' costs and SPs revenue	Normal	Well covered
[19, 31, 38, 179, 180]	Energy management	Energy consumption and green resources	Hot	More Studies Needed
[26, 29, 31]	Fault tolerance	Keep working despite faults	Normal	More Studies Needed
[2, 19, 26-28, 30, 31, 144, 181]	Heterogeneous tasks and resources	Tasks and network heterogeneity	Very Hot	Fresh Area
[29]	Interoperability	Nodes and servers' variety and distribution	Idle	Fresh Area
[19, 31]	Joint scheduling of all resources	Investigating communication and computational resource simultaneously	Hot	More Studies Needed
[144]	Load balancing and task distribution	Distribute tasks efficiently	Normal	More Studies Needed
[29-31, 38, 106, 180-184]	Mobility	Dynamic and moving network entities	Normal	More Studies Needed
[28, 29, 31]	Multi-objective optimization	Considering various QoS parameters	Normal	Fresh Area
[19, 29, 31, 185]	Partitioning and integration	Partitioning tasks while offloading and result integration	Hot	Fresh Area
[2, 19, 26, 30, 83, 181-183, 185-187]	Privacy	Keeping information confidentially	Normal	More Studies Needed
[19, 29, 31, 181-183]	Scalability	Handle the growing demands, requests, and services	Very Hot	Fresh Area
[2, 19, 26, 27, 29, 30, 83, 106, 144, 181, 183, 185-187]	Security	Appropriate reaction against security threats	Normal	More Studies Needed
[144, 184]	Simulation platform	Modeling and simulation on software platforms or real testbeds	Hot	More Studies Needed
[27]	Subtasks dependencies	Workflow task scheduling	Idle	Fresh Area

are considered. Computational tasks might be divided into two parts, either offloaded to remote edge nodes or executed locally. According to most existing works, the offloaded section of a task is represented by an offloaded ratio [174]. An optimal ratio of offloaded resources and other optimization variables is determined in resource scheduling. This part of the task is directly offloaded once the optimal offload ratio is obtained [54]. An optimization solution for a certain task may not result in a divisible part that is the same as the optimal offloaded part. For this reason, future research should further explore how task partitioning can be used during computation offloading. The dispersed results of the task must be integrated after it has been partitioned and processed by different nodes. This process may also raise the question: Are the integrated results the same as those obtained through non-partitioning? Therefore, it will be important to investigate how to integrate the results of processing from different nodes.

M. JOINT SCHEDULING OF ALL RESOURCES

Processing nodes should receive task data and cache it in a data queue, waiting to process the offloaded tasks. Real-time task processing relies heavily on the caching and queuing process. However, several existing works ignore the caching

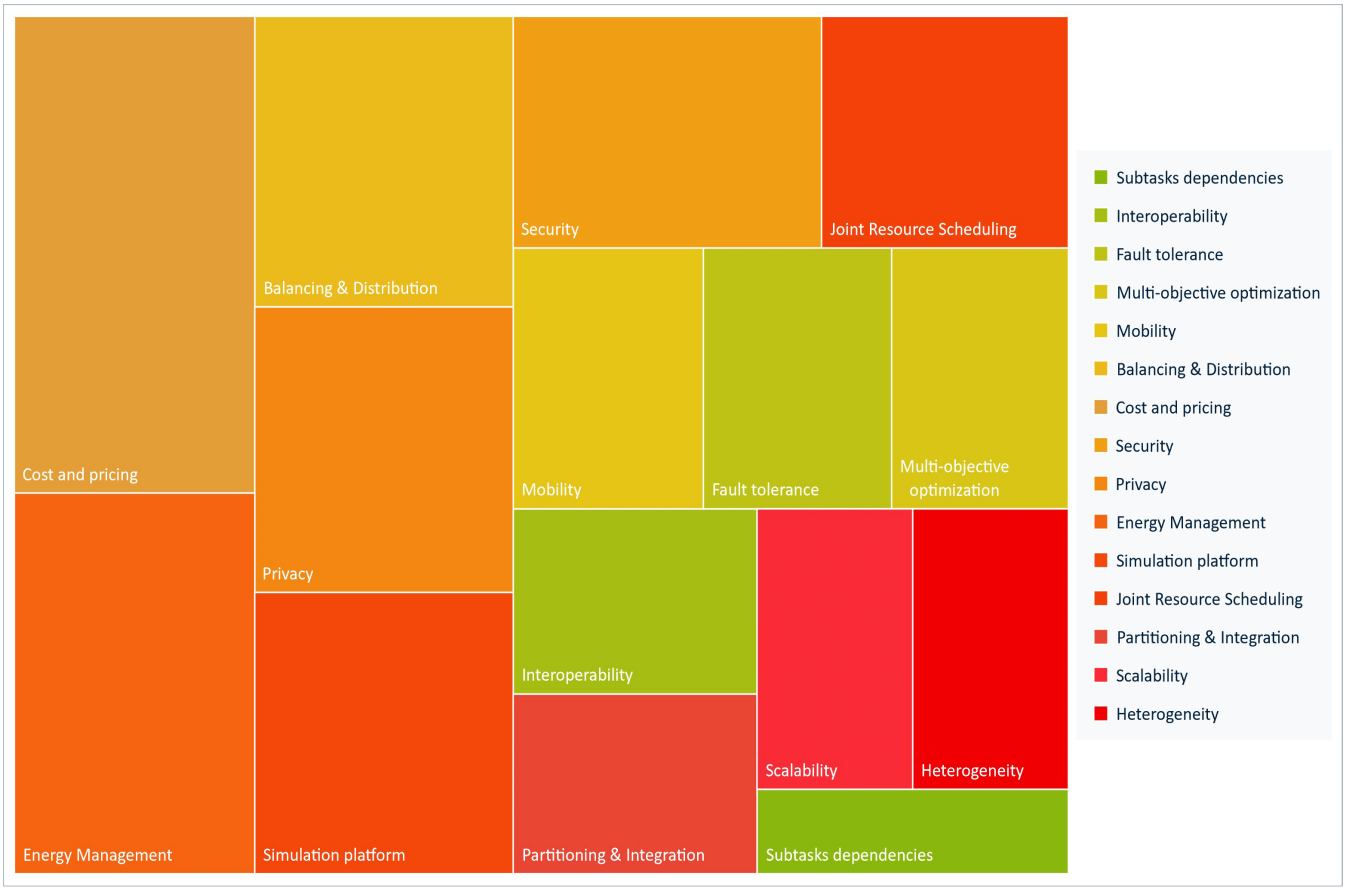


FIGURE 8. Challenges hotness and research trends.

and queuing processes in calculating the total task processing time, considering only the sum of the local processing, transmission, and offloading times. Moreover, a few attempts have combined communication and computing resource allocations. Caching and queuing should be considered for future research on the joint scheduling of edge computing resources.

N. MULTI-OBJECTIVE OPTIMIZATION

An edge network does not have any mechanism for defining QoS parameters that will influence resource scheduling. For example, some algorithms disregard such variables as scalability, reliability, and security in favor of energy, cost, or response time. Future multi-objective optimization methods should incorporate QoS parameters into scheduling decision-making and establish a trade-off between them.

O. SIMULATION PLATFORM

Simulation involves modeling a real scenario using mathematical formulas and implementing it with programming languages. Using simulations gives us a better understanding of the system and enables us to conduct experiments at a lower cost. Many resources are required for experiments and testing on real-world edge computing infrastructure, which is not feasible for all researchers. The researchers, instead, are more likely to use simulation platforms to implement their new ideas in EC. The next generation of simulation platforms must reinforce end users' mobility and the migration of tasks and provide an energy consumption model for servers and end users. Simulation tools such as iFogSim [175], EdgeCloudSim [176], and MyiFogSim [177] are generally used in current research to evaluate the performance of scheduling algorithms. A general simulation platform such as MATLAB is also used for this evaluation. Only a few studies have evaluated their algorithms in real edge systems. Despite edge computing being investigated, most researchers do not yet have access to a real testbed, and most assessments were conducted using simulators. Since the results of the discussed algorithms in a real testbed may differ from those in a simulation, implementing them in a real environment can be challenging. Therefore, additional effort will be needed to develop testbeds or prototypes for evaluating scheduling algorithms in real systems.

IX. RECOMMENDATIONS FOR FUTURE RESEARCH DIRECTIONS

Before concluding this survey, this section provides a useful overview of current research and future directions. Some topics are expected to get more attention in the future in resource scheduling domain in edge networks. Some of

the previously mentioned research directions are introduced and strongly recommended for resource scheduling in edge computing.

- Security and privacy are two significant directions being pursued that could enhance performance and the end user experience. Resource scheduling in EC may cause user information leakage and is prone to various potential hostile attacks. Therefore, a substantial amount of research is needed in this area to address privacy and security concerns. Distributed ledger as a new burn technology can ensure tamper-proof and traceable resource scheduling records, which makes blockchain an appropriate candidate to support resource scheduling in untrustworthy environments for future research.
- In most previous works, scheduling methods have been investigated in homogeneous environments. However, this assumption is simplistic according to the various end devices and different computing, communication, and storage resources within the edge network. Therefore, researchers must explore modern mechanisms to handle hybrid tasks and resources in a heterogenous environment.
- The mobility of most connected things, including MDs, vehicles, and drones, causes frequent link failures between devices and servers, compromising the scheduling QoS. Sometimes, the scheduling decisions made at this moment may not apply in the future, or the node is no longer in the user's service range. Achieving an efficient task scheduling and loading mechanism is challenging when users are often on the move, which can be a hot topic for future research.
- As a newly emerging computing paradigm, edge computing can make 5G and IoT applications more robust and reliable. However, the nature of geographically distributed edge resources is often characterised by unreliable communications and constantly changing environments, resulting in a diversity of potential vulnerabilities and instability. Consequently, adopting a fault-tolerant scheduling methodology in collaborative, distributed, and dynamic environments prone to failures and faults takes work. Task execution can be affected adversely by such faults. To counter these devastating consequences and ensure the accuracy and consistency of collaborative scheduling, fault-tolerant strategies are urgently needed.
- Edge computing capabilities will need to be expanded and scaled in response to the ever-growing number of devices and endpoints that generate huge amounts of data that require management and oversight. These numbers can fluctuate, requiring efficient and comfortable resource system management to track and process them as end users need.

A total of twenty survey papers have been reviewed to understand the challenges mentioned, future directions, and the impressions they have on the current state of research. According to the information extracted from these survey papers and the number of publications on each topic retrieved from Google Scholar, the challenges hotness, and research trends are illustrated in Table 5 and Figure 8 above. Table 5 demonstrates that the most recent survey papers on edge resource scheduling have introduced security, privacy, and resource heterogeneity as critical future directions. Furthermore, edge computing researchers are becoming increasingly interested in heterogeneous resource scheduling. Additionally, there has been little research on scalability, which makes it a super-hot topic.

Similarly, the Treemap chart shown in Figure 8 compares the fifteen hottest future directions in task scheduling proposed by contemporary edge computing studies. Bigger rectangles represent challenges with a larger share of edge computing publications in the last five years. The challenges hotness is also illustrated using a color spectrum from light green to dark red, representing very hot and idle challenges.

X. CONCLUSION

Since the early days of emerging edge computing, when computing and storage nodes are positioned close to mobile devices and sensors, industry investment in this paradigm has grown dramatically. The crucial role that resource scheduling is massively involved in evolving the edge, has attracted research trends toward this niche area. The resource scheduling process aims to ensure the quality of services by assigning the proper resources to submitted tasks.

This paper provides a comprehensive survey related to remarkable studies in resource scheduling conducted in the past few years. First, the paper has analyzed and described collaboration methods and computation task analysis for resource scheduling in edge computing. Then, the structure and features of all four collaborative computation scenarios are explored in detail. For executing computing tasks, there are three well-known approaches: local execution, partial offloading, and full offloading, which have been described clearly. Additionally, the most recent studies have been examined, compared, and covered thoroughly. Moreover, this survey has demonstrated fairness and load balancing and provided a decent insight into principles. Finally, the paper has introduced several open issues in the resource scheduling field. The significance of each was investigated by scanning the literature to pave the way for possible research directions.

REFERENCES

[1] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.
[2] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.
[3] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.
[4] T. Q. Thinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.
[5] Y. Song, S. S. Yau, R. Yu, X. Zhang, and G. Xue, "An approach to QoS-based task distribution in edge computing networks for IoT applications," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 32–39.

[6] M. N. Hindia, A. W. Reza, O. Dakkak, S. A. Nor, and K. A. Noordin, "Cloud computing applications and platforms: A survey," in *Proc. 3rd Int. Conf. Comput. Eng. Math. Sci. (ICCEMS)*, Dec. 2014.

[7] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[8] O. Dakkak, S. A. Nor, M. S. Sajat, Y. Fazea, and S. Arif, "From grids to clouds: Recap on challenges and solutions," *AIP Conf. Proc.*, vol. 2016, no. 1, 2018, Art. no. 020040.

[9] X. Xu, Y. Li, T. Huang, Y. Xue, K. Peng, L. Qi, and W. Dou, "An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks," *J. Netw. Comput. Appl.*, vol. 133, pp. 75–85, May 2019.

[10] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.

[11] I. Martinez, A. S. Hafid, and A. Jarray, "Design, resource management, and evaluation of fog computing systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2494–2516, Feb. 2021.

[12] S. Chen, Q. Li, M. Zhou, and A. Abusorrah, "Recent advances in collaborative scheduling of computing tasks in an edge computing paradigm," *Sensors*, vol. 21, no. 3, p. 779, Jan. 2021, doi: 10.3390/s21030779.

[13] P. Hosseinioun, M. Kheirabadi, S. R. Kamel Tabbakh, and R. Ghaemi, "ATask scheduling approaches in fog computing: A survey," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, Mar. 2022, Art. no. e3792.

[14] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[15] C. Guleria, K. Das, and A. Sahu, "A survey on mobile edge computing: Efficient energy management system," in *Proc. Innov. Energy Manag. Renew. Resour.*, 2021, pp. 1–4.

[16] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Gener. Comput. Syst.*, vol. 97, pp. 219–235, Aug. 2019.

[17] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–8.

[18] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, and E. Benkhelifa, "SDMEC: Software defined system for mobile edge computing," in *Proc. IEEE Int. Conf. Cloud Eng. Workshop (IC2EW)*, Apr. 2016, pp. 88–93.

[19] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2131–2165, 4th Quart., 2021.

[20] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[21] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[22] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[23] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, May 2016.

[24] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 668–682, Mar. 2019.

[25] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[26] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*, vol. 149, no. 2. New York, NY, USA: Springer, 2006.

[27] M. Raeisi-Varzaneh and H. Sabaghian-Bidgoli, "A Petri-net-based communication-aware modeling for performance evaluation of NoC application mapping," *J. Supercomput.*, vol. 76, no. 11, pp. 9246–9269, Nov. 2020.

[28] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[29] C. Feng, P. Han, X. Zhang, B. Yang, Y. Liu, and L. Guo, "Computation offloading in mobile edge computing networks: A survey," *J. Netw. Comput. Appl.*, vol. 202, Jun. 2022, Art. no. 103366.

[30] H. Djigal, J. Xu, L. Liu, and Y. Zhang, "Machine and deep learning for resource allocation in multi-access edge computing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 4, pp. 2449–2494, 4th Quart., 2022.

[31] A. Islam, A. Debnath, M. Ghose, and S. Chakraborty, "A survey on task offloading in multi-access edge computing," *J. Syst. Archit.*, vol. 118, Sep. 2021, Art. no. 102225.

[32] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107496.

[33] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102781.

[34] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," *IEEE Access*, vol. 7, pp. 131543–131558, 2019.

[35] P. G. V. Naranjo, Z. Pooranian, M. Shojafar, M. Conti, and R. Buyya, "FOCAN: A fog-supported smart city network architecture for management of applications in the Internet of Everything environments," *J. Parallel Distrib. Comput.*, vol. 132, pp. 274–283, Oct. 2019.

[36] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop on Mobile Big Data*. Hangzhou, China: Association for Computing Machinery, 2015, pp. 37–42.

[37] O. Dakkak, S. Arif, and S. A. Nor, "Resource allocation mechanisms in computational grid: A survey," *Asian Res. Publishing Netw.*, vol. 10, no. 15, Aug. 2015.

[38] O. Dakkak, S. A. Nor, and A. S. C. M. Arif, "Proposed algorithm for scheduling in computational grid using backfilling and optimization techniques," *J. Telecommun., Electron. Comput. Eng.*, vol. 8, no. 10, pp. 133–138, 2016.

[39] T. Zhu, T. Shi, Z. Cai, X. Zhou, and J. Li, "Task scheduling in deadline-aware mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4854–4866, Jun. 2019.

[40] O. Dakkak, Y. Fazea, S. A. Nor, and S. Arif, "Towards accommodating deadline driven jobs on high performance computing platforms in grid computing environment," *J. Comput. Sci.*, vol. 54, Sep. 2021, Art. no. 101439.

[41] M. Zamzam, T. Elshabrawy, and M. Ashour, "Resource management using machine learning in mobile edge computing: A survey," in *Proc. 9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2019, pp. 112–117.

[42] H. Tan, Z. Han, X.-Y. Li, and F. C. M. Lau, "Online job dispatching and scheduling in edge-clouds," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, May 2017, pp. 1–9.

[43] P. Wang, C. Yao, Z. Zheng, G. Sun, and L. Song, "Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2872–2884, Apr. 2019.

[44] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, pp. 289–330, Sep. 2019.

[45] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading toward edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1584–1607, Aug. 2019.

[46] J. Li, G. Luo, N. Cheng, Q. Yuan, Z. Wu, S. Gao, and Z. Liu, "An end-to-end load balancer based on deep learning for vehicular network traffic control," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 953–966, Feb. 2019.

[47] Q. Wu, X. Xu, Q. Zhao, and F. Dai, "Tasks offloading for connected autonomous vehicles in edge computing," *Mobile Netw. Appl.*, vol. 27, no. 6, pp. 2295–2304, Dec. 2022.

[48] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big IoT data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[49] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless body area networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1658–1686, 3rd Quart., 2014.

[50] N. A. Askar, A. Habbal, A. H. Mohammed, M. S. Sajat, Z. Yusupov, and D. Kodirov, "Architecture, protocols, and applications of the Internet of Medical Things (IoMT)," *J. Commun.*, vol. 17, no. 11, pp. 900–918, 2022.

[51] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 127, pp. 160–168, May 2019.

[52] X. Xu, Y. Xue, L. Qi, Y. Yuan, X. Zhang, T. Umer, and S. Wan, "An edge computing-enabled computation offloading method with privacy preservation for Internet of Connected Vehicles," *Future Gener. Comput. Syst.*, vol. 96, pp. 89–100, Jul. 2019.

[53] A. Filali, A. Abouaomar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-access edge computing: A survey," *IEEE Access*, vol. 8, pp. 197017–197046, 2020.

[54] F. Zhou, Y. Wu, H. Sun, and Z. Chu, "UAV-enabled mobile edge computing: Offloading optimization and trajectory design," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[55] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[56] Y. Liu, S. Wang, Q. Zhao, S. Du, A. Zhou, X. Ma, and F. Yang, "Dependency-aware task scheduling in vehicular edge computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4961–4971, Jun. 2020.

[57] A. Waheed, M. A. Shah, S. M. Mohsin, A. Khan, C. Maple, S. Aslam, and S. Shamshirband, "A comprehensive review of computing paradigms, enabling computation offloading and task execution in vehicular networks," *IEEE Access*, vol. 10, pp. 3580–3600, 2022.

[58] M. Noor-A-Rahim, Z. Liu, H. Lee, G. G. M. N. Ali, D. Pesch, and P. Xiao, "A survey on resource allocation in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 701–721, Feb. 2022.

[59] A. B. De Souza, P. A. L. Rego, T. Carneiro, J. D. C. Rodrigues, P. P. R. Filho, J. N. De Souza, V. Chamola, V. H. C. De Albuquerque, and B. Sikdar, "Computation offloading for vehicular environments: A survey," *IEEE Access*, vol. 8, pp. 198214–198243, 2020.

[60] H. Guo, J. Liu, and J. Zhang, "Computation offloading for multi-access mobile edge computing in ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 14–19, Aug. 2018.

[61] E. Meskar and B. Liang, "Fair multi-resource allocation with external resource for mobile edge computing," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 184–189.

[62] S. Sheng, P. Chen, Z. Chen, L. Wu, and Y. Yao, "Deep reinforcement learning-based task scheduling in IoT edge computing," *Sensors*, vol. 21, no. 5, p. 1666, Feb. 2021, doi: 10.3390/s21051666.

[63] L. Wan, L. Sun, X. Kong, Y. Yuan, K. Sun, and F. Xia, "Task-driven resource assignment in mobile edge computing exploiting evolutionary computation," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 94–101, Dec. 2019.

[64] A. J. Ferrer, J. M. Marquès, and J. Jorba, "Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–36, Nov. 2019.

[65] Q. Li, S. Wang, A. Zhou, X. Ma, F. Yang, and A. X. Liu, "QoS driven task offloading with statistical guarantee in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 278–290, Jan. 2022.

[66] A. Samanta and Z. Chang, "Adaptive service offloading for revenue maximization in mobile edge computing with delay-constraint," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3864–3872, Apr. 2019.

[67] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: A market equilibrium approach," *IEEE Trans. Cloud Comput.*, vol. 9, no. 1, pp. 302–317, Jan./Mar. 2021.

[68] X. Huang, R. Yu, J. Liu, and L. Shu, "Parked vehicle edge computing: Exploiting opportunistic resources for distributed mobile applications," *IEEE Access*, vol. 6, pp. 66649–66663, 2018.

[69] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[70] S. Abdelhamid, H. Hassanein, and G. Takahara, "Vehicle as a resource (VaaR)," *IEEE Netw.*, vol. 29, no. 1, pp. 12–17, Jan. 2015.

[71] (2016). *OpenFog Architecture Overview*. [Online]. Available: https://site.ieee.org/denver-com/files/2017/06/OpenFog-Architecture-Overview-WP-2-2016.pdf

[72] M. Ashouri, P. Davidsson, and R. Spalazzese, "Cloud, edge, or both? Towards decision support for designing IoT applications," in *Proc. 5th Int. Conf. Internet Things: Syst., Manage. Secur.*, Oct. 2018, pp. 155–162.

[73] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[74] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[75] C. Hong and B. Varghese, "Resource management in fog/edge computing: A survey on architectures, infrastructure, and algorithms," *ACM Comput. Surv.*, vol. 52, no. 5, p. 97, 2019.

[76] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "Challenges and opportunities in edge computing," in *Proc. IEEE Int. Conf. Smart Cloud (SmartCloud)*, Nov. 2016, pp. 20–26.

[77] A. Houmansadr, S. A. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," in *Proc. IEEE/IFIP 41st Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2011, pp. 31–32.

[78] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Apr. 2017.

[79] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 998–1010, Apr. 2018.

[80] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, "The future of mobile cloud computing: Integrating cloudlets and mobile edge computing," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–5.

[81] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[82] Z. Lin, J. Liu, J. Xiao, and S. Zi, "A survey: Resource allocation technology based on edge computing in IIoT," in *Proc. Int. Conf. Commun., Comput., Cybersec., Informat. (CCCI)*, Nov. 2020, pp. 1–5.

[83] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[84] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018.

[85] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[86] T. Li, X. He, S. Jiang, and J. Liu, "A survey of privacy-preserving offloading methods in mobile-edge computing," *J. Netw. Comput. Appl.*, vol. 203, Jul. 2022, Art. no. 103395.

[87] Z. Ali, L. Jiao, T. Baker, G. Abbas, Z. H. Abbas, and S. Khaf, "A deep learning approach for energy efficient computational offloading in mobile edge computing," *IEEE Access*, vol. 7, pp. 149623–149633, 2019.

[88] C. Wang, C. Dong, J. Qin, X. Yang, and W. Wen, "Energy-efficient offloading policy for resource allocation in distributed mobile edge computing," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 366–372.

[89] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11158–11168, Nov. 2019.

[90] X. Hu, K. K. Wong, K. Yang, and Z. Zheng, "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4738–4752, Oct. 2019.

[91] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, Feb. 2017.

[92] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[93] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, Sep. 2019.

[94] X. Chen, Y. Cai, Q. Shi, M. Zhao, B. Champagne, and L. Hanzo, "Efficient resource allocation for relay-assisted computation offloading in mobile-edge computing," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2452–2468, Mar. 2020.

[95] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber-wireless networks," *IEEE Trans. Wireless Commun.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[96] Z. Hong, W. Chen, H. Huang, S. Guo, and Z. Zheng, "Multi-hop cooperative computation offloading for industrial IoT-edge-cloud computing environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 12, pp. 2759–2774, Dec. 2019.

[97] P. Wang, Z. Zheng, B. Di, and L. Song, "HetMEC: Latency-optimal task assignment and resource allocation for heterogeneous multi-layer mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4942–4956, Oct. 2019.

[98] T. Q. Dinh, B. Liang, T. Q. S. Quek, and H. Shin, "Online resource procurement and allocation in a hybrid edge-cloud computing system," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2137–2149, Mar. 2020.

[99] C. Jiang and J. Wan, "A thing-edge-cloud collaborative computing decision-making method for personalized customization production," *IEEE Access*, vol. 9, pp. 10962–10973, 2021.

[100] W. Na, S. Jang, Y. Lee, L. Park, N.-N. Dao, and S. Cho, "Frequency resource allocation and interference management in mobile edge computing for an Internet of Things system," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4910–4920, Jun. 2019.

[101] Y. Miao, G. Wu, M. Li, A. Ghoneim, M. Al-Rakhami, and M. S. Hossain, "Intelligent task prediction and computation offloading based on mobile-edge cloud computing," *Future Gener. Comput. Syst.*, vol. 102, pp. 925–931, Jan. 2020.

[102] M.-T. Thai, Y.-D. Lin, Y.-C. Lai, and H.-T. Chien, "Workload and capacity optimization for cloud-edge computing systems with vertical and horizontal offloading," *IEEE Trans. Netw. Serv. Manage.*, vol. 17, no. 1, pp. 227–238, Mar. 2020.

[103] B. P. Rimal, D. P. Van, and M. Maier, "Mobile-edge computing vs. centralized cloud computing in fiber-wireless access networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2016, pp. 991–996.

[104] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco visual networking index (VNI) complete forecast update, 2017–2022," in *Proc. Americas/EMEAR Cisco Knowl. Netw. (CKN)*, 2018, pp. 1–30.

[105] J. Erman, A. Gerber, K. K. Ramadrishnan, S. Sen, and O. Spatscheck, "Over the top video: The gorilla in cellular networks," in *Proc. ACM SIGCOMM Conf. Internet Meas. Conf.* Berlin, Germany: Association for Computing Machinery, Nov. 2011, pp. 127–136.

[106] J. Yoon, P. Liu, and S. Banerjee, "Low-cost video transcoding at the wireless edge," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 129–141.

[107] J. Xu, B. Palanisamy, H. Ludwig, and Q. Wang, "Zenith: Utility-aware resource allocation for edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 47–54.

[108] C. Zhang, H. Du, Q. Ye, C. Liu, and H. Yuan, "DMRA: A decentralized resource allocation scheme for multi-SP mobile edge computing," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 390–398.

[109] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "A survey and taxonomy on task offloading for edge-cloud computing," *IEEE Access*, vol. 8, pp. 186080–186101, 2020.

[110] T. Zheng, J. Wan, J. Zhang, C. Jiang, and G. Jia, "A survey of computation offloading in edge computing," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Oct. 2020, pp. 1–6.

[111] Y.-H. Kao, B. Krishnamachari, M.-R. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3056–3069, Nov. 2017.

[112] E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*. San Francisco, CA, USA: Association for Computing Machinery, Jun. 2010, pp. 49–62.

[113] O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[114] J. Zhang and X. Zhao, "An overview of user-oriented computation offloading in mobile edge computing," in *Proc. IEEE World Congr. Services (SERVICES)*, Oct. 2020, pp. 75–76.

[115] A. A. Al-Habob, O. A. Dobre, A. G. Armada, and S. Muhaidat, "Task scheduling for mobile edge computing using genetic algorithm and conflict graphs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8805–8819, Aug. 2020.

[116] Z. Tong, F. Ye, J. Mei, B. Liu, and K. Li, "A novel task offloading algorithm based on an integrated trust mechanism in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 169, pp. 185–198, Nov. 2022.

[117] H. Materwala, L. Ismail, R. M. Shubair, and R. Buyya, "Energy-SLA-aware genetic algorithm for edge–cloud integrated computation offloading in vehicular networks," *Future Gener. Comput. Syst.*, vol. 135, pp. 205–222, Oct. 2022.

[118] X. Tang, Z. Wen, J. Chen, Y. Li, and W. Li, "Joint optimization task offloading strategy for mobile edge computing," in *Proc. IEEE 2nd Int. Conf. Inf. Technol., Big Data Artif. Intell. (ICIBA)*, Dec. 2021, pp. 515–518.

[119] M. Tang and V. W. S. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1985–1997, Jun. 2022.

[120] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via Markov decision process in mobile-edge computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2483–2493, Feb. 2021.

[121] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. Shen, "Energy efficient dynamic offloading in mobile edge computing for Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 9, no. 3, pp. 1050–1060, Jul. 2021.

[122] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[123] S. Bi and Y. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[124] F. Xu, Y. Xie, Y. Sun, Z. Qin, G. Li, and Z. Zhang, "Two-stage computing offloading algorithm in cloud-edge collaborative scenarios based on game theory," *Comput. Electr. Eng.*, vol. 97, Jan. 2022, Art. no. 107624.

[125] J. Wang, J. Zhang, L. Liu, X. Zheng, H. Wang, and Z. Gao, "Utility maximization for splittable task offloading in IoT edge network," *Comput. Netw.*, vol. 214, Sep. 2022, Art. no. 109164.

[126] X.-Y. Yu, W.-J. Niu, Y. Zhu, and H.-B. Zhu, "UAV-assisted cooperative offloading energy efficiency system for mobile edge computing," *Digit. Commun. Netw.*, Mar. 2022, doi: 10.1016/j.dcan.2022.03.005.

[127] H. Wu, K. Wolter, P. Jiao, Y. Deng, Y. Zhao, and M. Xu, "EEDTO: An energy-efficient dynamic task offloading algorithm for blockchain-enabled IoT-edge-cloud orchestrated computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2163–2176, Feb. 2021.

[128] N. F. Nash Jr., "The bargaining problem," *Econometrica, J. Econ. Soc.*, vol. 18, no. 2, pp. 155–162, 1950.

[129] J. Rawls, *A Theory of Justice*. Cambridge, MA, USA: Harvard Univ. Press, 1999.

[130] J. Kay and P. Lauder, "A fair share scheduler," *Commun. ACM*, vol. 31, no. 1, pp. 44–55, Jan. 1988.

[131] E. G.-L. Strugeon, H. Ouarnoughi, and S. Niar, "A multi-agent approach for vehicle-to-fog fair computation offloading," in *Proc. IEEE/ACS 17th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2020, pp. 1–8.

[132] M. Zhao, W. Li, L. Bao, J. Luo, Z. He, and D. Liu, "Fairness-aware task scheduling and resource allocation in UAV-enabled mobile edge computing networks," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 4, pp. 2174–2187, Dec. 2021.

[133] B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," *Adv. Eng. Softw.*, vol. 42, no. 7, pp. 419–425, 2011.

[134] W. Li, S. Cao, K. Hu, J. Cao, and R. Buyya, "Blockchain-enhanced fair task scheduling for cloud-fog-edge coordination environments: Model and algorithm," *Secur. Commun. Netw.*, vol. 2021, Apr. 2021, Art. no. 5563312.

[135] M. Mukherjee, M. Guo, J. Lloret, R. Iqbal, and Q. Zhang, "Deadline-aware fair scheduling for offloaded tasks in fog computing with inter-fog dependency," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 307–311, Feb. 2020.

[136] S. Yu, R. Langar, and X. Wang, "A D2D-multicast based computation offloading framework for interactive applications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[137] N. T. Ti, L. B. Le, and Q. Le-Trung, "Computation offloading in MIMO based mobile edge computing systems under perfect and imperfect CSI estimation," *IEEE Trans. Serv. Comput.*, vol. 14, no. 6, pp. 2011–2025, Nov. 2021.

[138] S. Mao, S. Leng, K. Yang, X. Huang, and Q. Zhao, "Fair energy-efficient scheduling in wireless powered full-duplex mobile-edge computing systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[139] L. Ji and S. Guo, "Energy-efficient cooperative resource allocation in wireless powered mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4744–4754, Jun. 2019.

[140] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation efficiency in a wireless-powered mobile edge computing network with NOMA," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[141] Y. Kim, H.-W. Lee, and S. Chong, "Mobile computation offloading for application throughput fairness and energy efficiency," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 3–19, Jan. 2019.

[142] M. Zeng, R. Du, V. Fodor, and C. Fischione, "Computation rate maximization for wireless powered mobile edge computing with NOMA," in *Proc. IEEE 20th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2019, pp. 1–9.

[143] K. Karthiban and J. S. Raj, "An efficient green computing fair resource allocation in cloud computing using modified deep reinforcement learning algorithm," *Soft Comput.*, vol. 24, no. 19, pp. 14933–14942, Oct. 2020.

[144] J. Singh, J. Warraich, and P. Singh, "A survey on load balancing techniques in fog computing," in *Proc. Int. Conf. Comput. Sci. (ICCS)*, Dec. 2021, pp. 47–52.

[145] A. Chandak and N. K. Ray, "A review of load balancing in fog computing," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2019, pp. 460–465.

[146] H. Pydi and G. N. Iyer, "Analytical review and study on load balancing in edge computing platform," in *Proc. 4th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Mar. 2020, pp. 180–187.

[147] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4377–4387, Jun. 2019.

[148] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2092–2104, Feb. 2020.

[149] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1529–1541, Jul. 2021.

[150] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4187–4195, Jun. 2020.

[151] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-UAV-enabled load-balance mobile-edge computing for IoT networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6898–6908, Aug. 2020.

[152] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, 2014, pp. 1–8.

[153] X. He, R. Jin, and H. Dai, "Peace: Privacy-preserving and cost-efficient task offloading for mobile-edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1814–1824, Mar. 2020.

[154] I. A. Elgendy, W.-Z. Zhang, Y. Zeng, H. He, Y.-C. Tian, and Y. Yang, "Efficient and secure multi-user multi-task computation offloading for mobile-edge computing in mobile IoT networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 17, no. 4, pp. 2410–2422, Dec. 2020.

[155] S. M. Karim, A. Habbal, S. A. Chaudhry, and A. Irshad, "Architecture, protocols, and security in IoV: Taxonomy, analysis, challenges, and solutions," *Secur. Commun. Netw.*, vol. 2022, pp. 1–19, Oct. 2022.

[156] T. L. Duc, R. G. Leiva, P. Casari, and P.-O. Östberg, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–39, Sep. 2020.

[157] T. Hong, W. Zhao, R. Liu, and M. Kadoch, "Space-air-ground IoT network and related key technologies," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 96–104, Apr. 2020.

[158] M. SalemAlzboon, "Peer to peer resource discovery mechanisms in grid computing: A critical review," in *Proc. 4th Int. Conf. Internet Appl., Protocols Services (NETAPPS)*, 2015, pp. 48–54.

[159] S. Secci, P. Raad, and P. Gallard, "Linking virtual machine mobility to user mobility," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 927–940, Dec. 2016.

[160] R. Wang, Y. Cao, A. Noor, T. A. Alamoudi, and R. Nour, "Agent-enabled task offloading in UAV-aided mobile edge computing," *Comput. Commun.*, vol. 149, pp. 324–331, Jan. 2020.

[161] M. Alabadi, A. Habbal, and X. Wei, "Industrial Internet of Things: Requirements, architecture, challenges, and future research directions," *IEEE Access*, vol. 10, pp. 66374–66400, 2022.

[162] E. Dubrova, *Fault-Tolerant Design*. New York, NY, USA: Springer, 2013.

[163] M. L. Shooman, *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design*. Hoboken, NJ, USA: Wiley, 2003.

[164] N. Piovesan, A. F. Gambin, M. Miozzo, M. Rossi, and P. Dini, "Energy sustainable paradigms and methods for future mobile networks: A survey," *Comput. Commun.*, vol. 119, pp. 101–117, Apr. 2018.

[165] W. Chen, D. Wang, and K. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," *IEEE Trans. Serv. Comput.*, vol. 12, no. 5, pp. 726–738, Sep. 2019.

[166] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: Autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10660–10675, Dec. 2017.

[167] F. Salleh, S. Hassan, K. Malaysia, A. Habbal, E. Mkpojiogu, and U. Utara, "Internet of Things applications for smart campus," in *Proc. 6th Int. Conf. Internet Appl. Protocols Services (NETAPPS)*, 2020, pp. 9–16.

[168] H. K. Apat, B. Sahoo, P. Maiti, and P. Patel, "Review on QoS aware resource management in fog computing environment," in *Proc. IEEE Int. Symp. Sustain. Energy, Signal Process. Cyber Secur. (iSSSC)*, Dec. 2020, pp. 1–6.

[169] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-fog interoperability in IoT-enabled healthcare solutions," in *Proc. 19th Int. Conf. Distrib. Comput. Netw.*, Jan. 2018, pp. 1–10.

[170] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "Pricing policy and computational resource provisioning for delay-aware mobile edge computing," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Jul. 2016, pp. 1–6.

[171] G. Katona, "Rational behavior and economic behavior," *Psychol. Rev.*, vol. 60, no. 5, p. 307, 1953.

[172] L. Chen, J. Wu, J. Zhang, H.-N. Dai, X. Long, and M. Yao, "Dependency-aware computation offloading for mobile edge computing with edge-cloud cooperation," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2451–2468, Oct. 2022.

[173] J. Yan, S. Bi, Y. J. Zhang, and M. Tao, "Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 235–250, Jan. 2020.

[174] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, Apr. 2020.

[175] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Softw., Pract. Exp.*, vol. 47, no. 9, pp. 1275–1296, 2017.

[176] C. Sonmez, A. Ozgovde, and C. Ersoy, "EdgeCloudSim: An environment for performance evaluation of edge computing systems," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 11, Nov. 2018, Art. no. e3493.

[177] M. M. Lopes, W. A. Higashino, M. A. M. Capretz, and L. F. Bittencourt, "MyiFogSim: A simulator for virtual machine migration in fog computing," in *Proc. Companion 10th Int. Conf. Utility Cloud Comput.*, Dec. 2017, pp. 47–52.

[178] N. Lyu, "Brief review on computing resource allocation algorithms in mobile edge computing," in *Proc. 2nd Int. Conf. Comput. Data Sci. (CDS)*, Jan. 2021, pp. 108–111.

[179] S. Talal, W. S. M. Yousef, and B. Al-Fuhaidi, "Computation offloading algorithms in vehicular edge computing environment: A survey," in *Proc. Int. Conf. Intell. Technol., Syst. Service Internet Everything (ITSS-IoE)*, Nov. 2021, pp. 1–6.

[180] H. T. Malazi, S. R. Chaudhry, A. Kazmi, A. Palade, C. Cabrera, G. White, and S. Clarke, "Dynamic service placement in multi-access edge computing: A systematic literature review," *IEEE Access*, vol. 10, pp. 32639–32688, 2022.

[181] X. He, M. Meng, S. Ding, and H. Li, "A survey of task migration strategies in mobile edge computing," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2021, pp. 400–405.

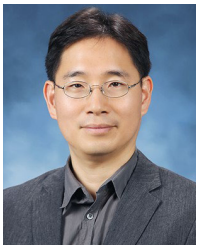
[182] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Collaborative vehicular edge computing networks: Architecture design and research challenges," *IEEE Access*, vol. 7, pp. 178942–178952, 2019.

[183] S. Dubey and J. Meena, "Computation offloading techniques in mobile edge computing environment: A review," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Jul. 2020, pp. 1217–1223.
[184] H. Jin, M. A. Gregory, and S. Li, "A review of intelligent computation offloading in multiaccess edge computing," *IEEE Access*, vol. 10, pp. 71481–71495, 2022.
[185] B. P. Nayak, L. Hota, A. Kumar, A. K. Turuk, and P. H. J. Chong, "Autonomous vehicles: Resource allocation, security, and data privacy," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 117–131, Mar. 2022.
[186] A. Sharma and C. Diwakar, "Future aspects on MEC (mobile edge computing): Offloading mechanism," in *Proc. 6th Int. Conf. Signal Process., Comput. Control (ISPCC)*, Oct. 2021, pp. 34–39.

MOSTAFA RAEISI-VARZANEH received the B.Sc. degree in computer science from the University of Isfahan, Isfahan, Iran, in 2014, and the M.S. degree in software engineering from the University of Kashan, Iran, in 2018. He is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Karabük Üniversitesi, Türkiye. For the last five years, he has worked on resource allocation and workflow task scheduling on networks on chips, cloud, and edge computing. His recent research interests include dynamic circumstances disrupting task scheduling and resource management strategies.

OMAR DAKKAK received the B.E. degree in telecommunication engineering from Ittihad University, Syria, and the M.Sc. and Ph.D. degrees in computer science from Universiti Utara Malaysia (UUM). During his Ph.D. degree, he worked on scheduling problems in grid computing, analyzing the performance of the scheduling policies based on real workloads for better quality of service (QoS) criteria and building scheduling mechanisms considering high-performance computing (HPC) applications through a simulation approach using real workloads. He is currently an Assistant Professor with the Faculty of Engineering, Department of Computer Engineering, Karabük Üniversitesi, Türkiye. His research interests include scheduling algorithms, performance evaluation, optimization in scheduling, and analyzing datasets on HPC platforms. He conducted several studies in other research areas, such as cloud computing, LTE, and MANET.

ADIB HABBAL (Senior Member, IEEE) received the Ph.D. degree in computer science (specializing in networked computing) from Universiti Utara Malaysia (UUM), Malaysia. He is currently a Professor (Associate) of computer engineering and the Founding Head of the Innovative Networked Systems (INETs) Research Group, Karabük Üniversitesi, Türkiye. Before joining Karabük Üniversitesi, in 2019, he was a Senior Lecturer with UUM for ten years and the Head of the InterNetWorks Research Platform for three years. His research projects have been funded by several organizations, including IEEE R10, the IEEE Malaysia Section, the Internet Society, the Chinese Academy of Sciences, the Malaysian Ministry of Higher Education, UUM, and others. He has authored/coauthored more than 100 refereed technical publications in journals and conference proceedings in the areas of future internet and wireless networks. His professional experience includes being a Speaker at a number of renowned research conferences and technical meetings, such as ACM SIGCOMM, APAN, APRICOT, IEEE, and internet2, an editor of top-tier and refereed journals, a technical program committee member of international conferences on computing networks, and an examiner of postgraduate scholars in his research areas. His research interests include future internet protocols and architecture, next-generation mobile networks, WEB3, blockchain technology, and digital trust. He served as an IEEE UUM Student Branch Founding Counselor and an Executive Council Member for the Internet Society Malaysia Chapter. He has received a number of international recognitions for his outstanding educational and research activities, including the UUM Excellent Service Award, in 2010, the UUM Best Research Award, in 2014, the UUM-SOC Prolific Writer Award, in 2016, and many others. He was a recipient of the Internet Society Fellowship to the Internet Engineering Task Force (IETF), the IEEE Malaysia Section Best Volunteer Award, and the Asia–Pacific Advanced Network (APAN) Fellowship to APAN35.



BYUNG-SEO KIM (Senior Member, IEEE) received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2001 and 2004, respectively. His Ph.D. study was supervised by Dr. Yuguang Fang. From 1997 to 1999, he worked as a Computer Integrated Manufacturing (CIM) Engineer with the Advanced Technology Research and Development (ATR&D), Motorola Korea Ltd., Paju, South Korea. From January 2005 to August 2007, he worked with Motorola Inc., Schaumburg, IL, USA, as a Senior Software Engineer of networks and enterprises. From 2012 to 2014, he was the Chairperson of the Department of Software and Communications Engineering, Hongik University, South Korea, where he is currently a Professor. His work has appeared in around 167 publications and 22 patents. In Motorola Inc., his research focused on designing protocol and network architecture of wireless broadband mission-critical communications. His research interests include the design and development of efficient wireless/wired networks, including link adaptable/cross-layer-based protocols, multi-protocol structures, wireless CCNs/NDNs, mobile edge computing, physical layer design for broadband PLC, and resource allocation algorithms for wireless networks. He was a member of the Sejongcity Construction Review Committee and the Ansan-City Design Advisory Board. He served as the General Chair for the 3rd IWWCN 2017 and a TPC Member for the IEEE VTC 2014-Spring, the EAI FUTURE 2016, and the ICGHIC in the 2016 and 2019 conferences. He served as the Guest Editor for Special Issues of the *International Journal of Distributed Sensor Networks* (SAGE), IEEE Access, and the *Journal of the Institute of Electronics and Information Engineers*. He is an Associate Editor of IEEE Access.