Resource Management for Mission-Critical Applications in Edge Computing: Systematic Review on Recent Research and Open Issues

NAYEREH RASOULI, Department of Computing Scinece, Umeå University, Umeå, Sweden
CRISTIAN KLEIN, Department of Computing Science, Umeå University, Umeå, Sweden
ERIK ELMROTH, Department of Computing Sceince, Umeå University, Umeå, Sweden

In the realm of edge computing, the optimization of latency, energy, bandwidth, and local computation is critical, especially for mission-critical applications in sectors like disaster management and healthcare. Such applications, exemplified by deploying UAVs and autonomous robots, demand instantaneous data processing. Given the inherent constraints of edge servers–characterized by their limited capacity–meticulous resource management becomes paramount. This entails judicious resource allocation, astute provisioning, strategic task offloading, and judicious application placement, all pivotal for both fixed and mobile resource service delivery. This survey delves deep into the nuances of deploying mission-critical applications in an edge environment, dissecting their technological prerequisites. Our exploration employs a systematic literature review grounded in a conventional review methodology. We analyze the cornerstone quality of service metrics pivotal for such critical applications in edge contexts, aiming for efficient service delivery. Moreover, we identified some major gaps in current resource management strategies. Our overarching ambition is to pave the way for robust edge computing paradigms tailored for mission-critical applications.

CCS Concepts: • **General and reference → Surveys and overviews;**

Additional Key Words and Phrases: Mobile edge computing, resource management, resource allocation, scheduling, Internet of Things (IoT), mission-critical applications, latency-aware applications

ACM Reference Format:
Nayereh Rasouli, Cristian Klein, and Erik Elmroth. 2025. Resource Management for Mission-Critical Applications in Edge Computing: Systematic Review on Recent Research and Open Issues. *ACM Comput. Surv.* 58, 3, Article 71 (September 2025), 37 pages. https://doi.org/10.1145/3762181

1 Introduction

The proliferation of **Internet of Things (IoT)** applications has significantly altered our digital engagement methods. The advent of IoT has resulted in a novel epoch characterized by the interconnected various devices. This interconnection spans a wide range of domains, including smart

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.
Author's Contact Information: Nayereh Rasouli, Department of Computing Scinece, Umeå University, Umeå, Västerbotten County, Sweden; e-mail: nrasouli@cs.umu.se; Cristian Klein, Department of Computing Science, Umeå University, Umeå, Västerbotten County, Sweden; e-mail: cklein@cs.umu.se; Erik Elmroth, Department of Computing Sceince, Umeå University, Umeå, Västerbotten County, Sweden; e-mail: elmroth@cs.umu.se.

ACM 0360-0300/2025/09-ART71
https://doi.org/10.1145/3762181

homes, connected automobiles, industrial automation, and healthcare monitoring. Consequently, IoT facilitates the smooth flow of data and the acquisition of real-time insights. The functionality of these IoT applications is dependent on a diverse range of intelligent devices, including sophisticated sensors and mobile devices, which establish connections with IoT networks in order to collect, analyze, and transfer data. However, as the scope and complexity of IoT applications grow, so do the resource demands on individual devices. Despite continuous advancements in processing capacity, many IoT devices struggle to handle the intensive computational tasks required by resource-intensive applications. Cloud computing has traditionally offered a viable solution to this challenge, providing extensive storage and computing resources in centralized cloud data centers. Users can offload resource-intensive tasks to these remote cloud servers, enhancing their devices' computing and storage capabilities and enabling efficient task processing. However, offloading tasks to remote cloud servers often results in transmission delays, which can negatively impact user experiences, especially for latency-sensitive applications. In the context of real-time monitoring and control systems, many of which are now powered by IoT devices, such delays are not just about user experience but also have direct implications on the system's safety and operational reliability. Recognizing these challenges, there has been a paradigmatic shift toward edge environments, paving the way for the rise of edge computing. This approach brings computation and data storage closer to the location of end-users and IoT devices, significantly reducing latency and enhancing system responsiveness and safety.

Edge computing's fusion with mission-critical applications has the potential to revolutionize industries. These applications, due to the high stakes involved—be it the loss of lives, critical assets, or societal structures – demand utmost reliability. Examples abound: disaster management where rapid response during the golden relief time can save lives [1], military operations safeguarding national security [2], telemedicine offering crucial medical interventions [3], and industrial systems managing vital infrastructures. The intersection of edge computing with IoT technology elevates these applications, ensuring quicker responses, precise decision-making, and optimal resource management. However, integrating them into edge-cloud architectures, especially in multi-tenant setups, brings unique challenges. The key lies in managing reliability, availability, cost, and energy efficiency harmoniously across varied applications.

This work provides a comprehensive overview of edge resource management, examining recent advances and identifying research needs. This article is important to the wider community because it identifies the main problems and benefits in the field, helping others understand what has been done and what still needs attention. The ambition is to provide a picture of resource management for mission-critical edge computing applications at this time for future researchers. Table 1 includes a compilation of the abbreviations utilized within the article.

This article undertakes a nuanced, semantic review of resource management for mission-critical applications in edge computing, based on references published between 2018 and 2023. While other prior studies have conducted comparable evaluations, our research sets itself apart by

- **A Comprehensive Lens**: We capture the broader landscape, encompassing not just edge computing but also **Mobile/Multi-access Edge Computing** (**MEC**), Edge-Cloud, and Fog environments. This begins with a detailed assessment of review papers to explain the literature landscape and identify trends (Section 2).
- **Deep-dive into Mission-Critical Challenges**: Our focus intensifies on the unique challenges posed by mission-critical applications within the edge computing context, for this purpose, we cover mission-critical edge computing applications in detail, establishing the groundwork for later sections (Section 3). Then, our study technique is explained in Section 4. This includes identifying our research topics, discovery method, and article inclusion/exclusion criteria.

Table 1. Abbreviation Table

	Description		Description		Description
AP	Access Point	GA	Genetic Algorithm	PSO	Particle Swarm Optimization
AR	Augmented Reality	GT	Game Theory	QoS	Quality of Service
CAME	Cloud Assisted Mobile Edge	IoT	Internet of Things	RL	Reinforcement Learning
CCC	Conventional Cloud Computing	IoV	Internet of Vehicles	SDN	Software-Defined Network
CSA	Cuckoo Search Algorithm	MAR	Mobile Augmented Reality	SLA	Service Level Agreements
DAG	Directed Acyclic Graphs	MCC	Mobile Cloud Computing	SLR	Systematic Literature Review
DL	Deep Learning	MDP	Markov Decision Process	UAVs	Unmanned Aerial Vehicles
DNN	Deep Neural Network	MEC	Mobile/Multi-Access Edge Computing	V2X	Vehicle-to-Everything
DQN	Deep Q-Network	MINLP	Mixed-Integer Nonlinear Programming	VAF	Vehicle Adaptive Feedback
DRL	Deep Reinforcement Learning	ML	Machine Learning	VEC	Vehicle Edge Computing
DVFS	Dynamic Voltage and Frequency Scaling	MTG	Multi-Technology Gateway	VMs	Virtual Machines
FL	Federated Learning	NFV	Network Function Virtualization	VNF	Virtualized Network function
FO	Full Offloading	NN	Neural Network	VR	Virtual Reality

— **Emphasis on QoS Resource Management**: We shine a spotlight on the intricate balancing act in **Quality of Service** (**QoS**) resource management, highlighting its significance (Section 5). A taxonomy of all reviewed publications to help readers understand this section is also designed.
— **Future Research Needs**: We identify and highlight gaps in current research, presenting readers with promising opportunities for future studies. Topics discussed include multi-objective solutions, fault tolerance, renewable energy, and others (Section 6). The report concludes with reflective insights and conclusions drawn from our comprehensive domain exploration in Section 7.

2 Related Works

Designing efficient systems and applications requires understanding resource management in edge computing, which is expanding rapidly. This section summarizes current survey articles on edge computing resource management to provide readers with the latest research trends and perspectives. We will review relevant material more thoroughly in Section 5. A summary of comparable works is shown in Table 2. It is important to note that certain technical terms in edge computing, MEC, fog computing, and IoT share similar functionality and meaning. Nevertheless, these technologies maintain their generality and integrity, especially when addressing the requirements and challenges posed by mission-critical applications and tradeoffs between specific QoS parameters.

2.1 Focus on Methods

Zabihi et al. [4] performed a thorough and organized examination of the utilization of **Reinforcement Learning** (**RL**) and **Deep Reinforcement Learning** (**DRL**) techniques in computation offloading for fog Computing, MEC, and **Mobile Cloud Computing** (**MCC**) paradigms. Their review was grounded in categorizing previous studies that had explored RL/DRL in this context. The researchers examined several scholarly articles published between the timeframe of 2013 to 2017. Acheampong et al. [5] reviewed the current state of task offloading and resource allocation in edge computing. Their work included an examination of offloading strategies, offloading algorithms, and the factors that influence offloading decisions. Based on **Full Offloading** (**FO**) and **Partial Offloading** (**PO**), relevant surveys were presented. They also classified articles into two main groups: **Machine Learning** (**ML**) and non-ML algorithms. Zhang et al. [6] focused their efforts on discussing optimization problems that arise from both the system and application perspectives in MEC systems. The researchers focused on the investigation of compute offloading decision-making, resource allocation, and configuration adaptability to efficiently manage the dynamic nature of edge systems. They focused on optimizing video analytic applications in the following ways: ensuring a balance between video analytic performance and cost on mobile devices, determining task placements for diverse video pipelines (sequential, parallel, and hybrid configurations) based on **Directed Acyclic Graph** (**DAG**) representations, and optimizing **Deep Neural Network** (**DNN**) operations, including distributed and collaborative model inference. Djigal et al. [7] carried out an extensive investigation into the advantages of applying ML and **Deep Learning** (**DL**) in MEC, focusing on ML/DL-enabled MEC. They presented three use cases from different perspectives and discussed the potential technologies, algorithms, and recent works related to resource allocation in MEC using ML and DL. They also analyzed the lessons learned from existing ML and DL-based methods and discussed the challenges and future research directions fin utilizing ML and DL for resource allocation in MEC. Shakarami et al. [8] primarily concentrated on examining the resource provisioning framework and mechanism. They extensively explored various mechanisms, including heuristic/meta-heuristic-based, framework-based, model-based, ML-based, and **Game Theory** (**GT**)-based approaches, to facilitate a comprehensive comparison among articles related to resource provisioning. Through their findings, it was observed that general heuristic-based and model-based mechanisms had a notable impact, followed by hybrid methods and RL in ML-based mechanisms. Furthermore, container-based mechanisms in framework-based approaches were identified as significant contributors to the existing literature. Lin et al. [9] discussed several fundamental models, including the channel model, computation and communication model, and energy harvesting model, that had been proposed in the field of offloading modeling. They then explained various offloading modeling methods, which were rooted in non-convex optimization, the **Markov Decision Process** (**MDP**), and others. Shakarami et al. [10] focused on ML-based computation offloading mechanisms in the MEC environment. They presented a taxonomy divided into three categories: RL-based, supervised learning-based, and unsupervised learning-based mechanisms. A comparison of these categories was provided based on performance metrics, case studies, techniques used, and evaluation tools, highlighting their advantages and weaknesses. Duc et al. [11] explored the challenge of ensuring reliable resource provisioning in joint edge-cloud environments. They examined various technologies, mechanisms, and methods aimed at enhancing the reliability of applications that are dispersed over a variety of different network configurations. The primary emphasis was placed on utilizing ML methodologies to tackle the complexity of the issue and efficiently oversee and regulate complex distributed applications. The reliable resource provisioning challenge was classified in the study into three primary techniques: workload characterization and prediction, component placement and system consolidation, and application elasticity.

2.2 Focus on Energy Consumption, Delay, Latency, and Mobility as an Objective

Zhou et al. [12] introduced the fundamental concepts and common application scenarios of MEC while also presenting the formulation of the task offloading problem. They then proceeded to analyze and consolidate the current state of industry research concerning essential technologies and use cases for MEC. They classified literature based on FO and PO by focusing on only two objectives, such as delay and energy. They furthermore provided guidance on areas for further investigation in follow-up research work. Singh et al. [13] conducted a comparative analysis of their study with other surveys published between 2018 and 2021, focusing on various parameters such as architecture, use cases, and mobility. They moreover explored the impact of mobility on the functional components of MEC. The article highlighted open and future research directions concerning mobility-related aspects of MEC. Lyu et al. [14] summarized the existing research related to the reduction of time delay and energy consumption from the perspectives of both single nodes and multiple nodes. They conducted an analysis of the methods utilized in previous studies, highlighting the merits and disadvantages associated with these approaches. Jiang et al. [15] reviewed articles in computation offloading for edge computing. They examined various aspects, including QoS assurance and enhancement of quality of experience. They surveyed resource allocation approaches and optimization techniques, such as GT and heuristics, for improving system performance and reducing overheads in computation offloading decision-making.

2.3 Focus on Applications

Sonkoly et al. [16] identified three key dimensions that influenced the placement problem and potential solutions. These dimensions included the platform capabilities (such as cloud-edge architecture and placement controller policy), the application characteristics (including supported application types, service structure, and security and privacy considerations), and the mathematical aspects (problem formulation, optimization methods, objectives, constraints, and solution complexity). Santi et al. [17] undertook a review of mission-critical applications and resource allocation. They specifically examined two use cases, namely disaster management and military operations, to demonstrate the practical implementation of MEC. The study identified future challenges in the areas of mobility, security, and privacy, the development of sustainable and environmentally friendly MEC solutions, and the establishment of effective test beds for experimentation and validation.

2.4 Focus on Architecture and Model

Raeisi et al. [18] conducted a thorough and extensive comparison between their research and a select few review papers. Their study effectively addressed the topic of computational offloading, which was also explored in the other papers. However, their work delved deeper into areas such as collaboration computing, fairness, and load distribution. The authors provided comprehensive coverage of these aspects, demonstrating an understanding of the subject matter. Hou et al. [19] performed a comprehensive analysis, comparing MEC with other paradigms such as MCC, **Conventional Cloud Computing** (**CCC**), and edge-cloud computing. The authors furthermore extensively discussed the features, challenges, and open issues related to MEC-enabled vehicular architecture. Luo et al. [20] provided an analysis of the architectural structure and presented a three-tier heterogeneous edge computing network consisting of things layer, edge layer, and cloud layer. However, it should be noted that several other surveys and research papers have also examined similar architectural frameworks. They moreover proposed a foundational framework for the allocation of resources in edge computing. This model guided users in determining whether to engage in offloading actions based on the current state of communication and computing resources.

In contrast to the previously stated efforts outlined in Table 2, this article presents its own unique contributions.

(1) In this article, we specifically concentrate on filtering and analyzing research papers, both surveys and related articles, from the last five years. By focusing on recent contributions, we aim at presenting a comprehensive review of the latest advancements and developments in the field, providing relevant and up-to-date insights.
(2) As a brief explanation of the architecture for resource scheduling in edge computing, along with its collaboration methods and even a basic model, was discussed in most of the aforementioned reviews, we have made the decision to omit its discussion from this article.
(3) Most of the mentioned works have thoroughly covered explanations and clarifications related to computational offloading. As a result, this aspect falls beyond the scope of this study.
(4) In the landscape of resource allocation and task offloading in edge computing, researchers' endeavors encompass tailoring resource management to suit individual users' needs and optimizing resource allocation for multiple users concurrently. While the classification of single-user and multi-user scenarios is widely explored in existing surveys, we consider avoiding utilizing this classification approach.
(5) A restricted number of surveys examined resource management regarding "applications", with just one of them specifically targeting mission-critical applications, in which only two

Table 2. Comparison of Surveys on Resource Allocation and Computation Offloading in Edge Computing

Ref.	Year	Paradigm	Research Area	Main Focus
[4]	2023	Mobile Edge Computing Cloud Computing Fog Computing	Computing Offloading	Focus on task offloading with RL/DRL
[5]	2023	Edge Computing	Resource Allocation Computation Offloading	Focus on PO and FO, Two types categorized algorithms
[6]	2023	Mobile Edge Computing	Resource Allocation Computing Offloading	Focus on decentralized task offloading based on Learning-based Optimization, Video Analytic
[12]	2023	Mobile Edge Computing	Computation Offloading	The existing body of literature derived from Delay and Energy
[13]	2023	Multi-access Edge Computing	Resource Allocation Computation Offloading Service Migration,	Caching, Mobility, Architecture, Computing paradigms
[18]	2023	Edge Computing	Resource Allocation Computation Offloading	Architecture, Collaboration computing, Offloading mode, Comparison between surveys
[7]	2022	Multi-access Edge Computing	Resource Allocation Computation Offloading	ML/DL-based RA mechanisms in MEC
[8]	2022	Fog Computing Edge Computing	Resource provisioning	Review of resource provisioning approaches in computation paradigms
[19]	2022	Multi-access Edge Computing Cloud Computing	MEC framework and Cloud Architecture	Review of MEC-Enabled Vehicular Networks
[14]	2021	Mobile Edge Computing	Computation Offloading	Challenges of Latency and Energy consumption
[16]	2021	Edge Computing	Computation Offloading	Offloading modeling methods
[20]	2021	Edge Computing	Resource scheduling	Comparison of techniques based on centralized and distributed modes
[9]	2020	Edge Computing	Computation Offloading	Categorizes previous works on Computation Offloading
[10]	2020	Multi-access Edge Computing	Resource Allocation Computation Offloading	ML-based Computation Offloading Mechanisms
[11]	2019	Edge-Cloud Computing	Resource Allocation Workload Prediction	Reliability of distributed applications ML Approaches
[15]	2019	Edge Computing	Computation offloading	Covering QoS, Quality of Experience, and Decision Making
This Survey	2023	Mobile Edge Computing Edge-Cloud Vehicular Edge Computing	Resource Allocation Computation Offloading Application Placement & Replica	Resource Management for Mission-critical applications and other Time-sensitive applications

types of mission-critical applications were addressed, and also the resource management aspects were not considered.

(6) Some researchers in previous works primarily focused on specific objectives, such as energy and latency, while neglecting other equally important aspects. It is crucial to consider that resource management in edge computing involves multiple objectives, each playing a significant role.

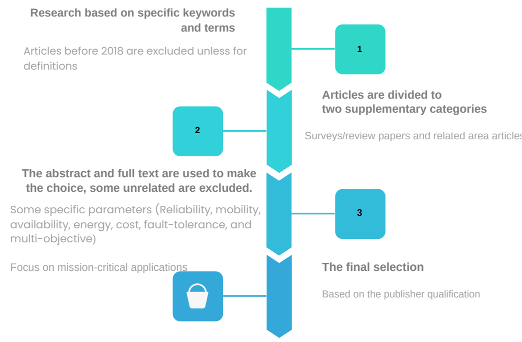

Fig. 1. The process of searching and selecting articles.

(7) Lastly, another aspect the review papers emphasize is the distinction between ML and non-ML methods. As the development methodologies were briefly covered, providing a detailed discussion or categorization of these approaches is beyond the scope of this work.

3 Research Methodology

The primary objectives of this study revolve around identifying, differentiating, analyzing, and evaluating key articles in the field of resource management for specific types of applications like mission-critical applications in edge computing. The focus is investigating the tradeoffs between various QoS parameters in the edge computing environment. We have employed a Systematic Literature Review (SLR) methodology to achieve these goals. The SLR approach allows us to comprehensively study the different elements and characteristics of methods utilized in this domain, with an aim at gaining a comprehensive understanding of the significant challenges and issues encountered within this industry. We conducted a systematic review following the technique outlined by Kitchenham [21], which is in accordance with the criteria provided by the "Centre for Reviews and Dissemination". To guide our investigation, we have formulated a set of research questions, as follows:

(1) How can QoS requirements for mission-critical applications be effectively captured and defined in the context of edge computing and IoT?
(2) What are the existing resource management techniques that consider QoS requirements in edge computing environments?
(3) How can resource allocation and scheduling algorithms be optimized to meet the QoS requirements of mission-critical applications in edge computing?

The article search and selection methodology employed in this study is a comprehensive and systematic process that has been meticulously partitioned into four distinct stages, as visually depicted in Figure 1. In the initial phase, to ensure a thorough search, we employed a carefully curated list of phrases and keywords, as presented in Table 3. Our search for articles was extensive, encompassing an array of electronic databases renowned for scholarly publications, such as Google Scholar, IEEE, ACM, Springer Link, Wiley, Elsevier, and MDPI. For the sake of this study, papers published prior to 2018 were removed from the literature review.

To maintain the rigor of our selection process, we utilized the Snowball Sampling procedure [22], allowing us to systematically expand our search by identifying additional relevant articles from the reference lists of the initially retrieved papers. This iterative approach enabled us to unearth a comprehensive pool of papers related to our study's objectives. From this pool, we ultimately selected 154 papers for detailed analysis. Notably, 17 of these selected papers are survey papers, and

Table 3. Search Terms and Keywords

S#	Search Terms and Keywords
S1	"Resource Management" and "Edge Computing"
S2	"Mission-critical Application" and "Edge Computing"
S3	"Mission-critical Application" and "IoT"
S4	"Low-latency" and "Resource management" in Edge
S5	"Fault tolerance" and "Resource allocation"
S6	"Energy efficient" and "resource allocation" in MEC
S7	"Resource scheduling" in Edge computing
S8	"Availability in Edge Computing"
S9	"Task Offloading Multi-Access Edge Computing" and "Mission-critical"

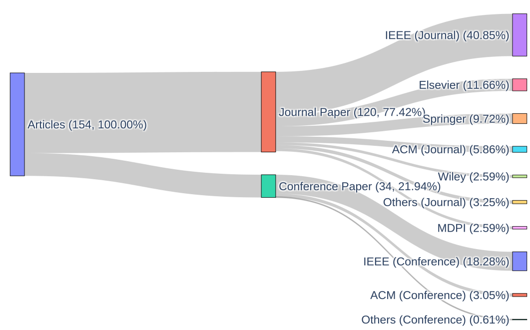

Fig. 2. Artifact track.

we performed a thorough comparison between our study and these survey papers to contextualize and position our work. The main body of our investigation focused on 115 recent articles, which we rigorously examined to contribute significant insights to our research.

Upon completion of the search, we diligently categorized the publications based on their titles, leading to interesting insights into the distribution of the literature. Remarkably, out of a total of 154 articles, a substantial proportion of roughly 77.42% was categorized as journal papers, indicating the notable importance of this format in the realm of resource allocation and job offloading within the domain of edge computing. Furthermore, it was found that around 21.94% of the sources fell under the category of conference papers, while a mere 0.65% of the materials were identified as books.

For the conference papers, IEEE remains predominant, accounting for 18.28% of the total papers. ACM-affiliated papers represent 3.05%, while a lesser fraction, 0.61%, is sourced from other publishers.

For the journal papers, IEEE continues its dominant presence, comprising 40.85% of the total papers. Elsevier follows with a contribution of 11.66%, and Springer at approximately 9.72%. ACM publications account for 5.86% of the dataset. Additionally, both Wiley and MDPI each make up about 2.59%. We also incorporated an additional 3.25% from various esteemed publishers, emphasizing the diverse range of sources in our study to ensure a comprehensive analysis. Figure 2 depicts this distribution.

The thoroughness of our article search and selection methodology encompasses a wide range of databases and employs a rigorous categorization process. This makes us believe that our study's findings are based on a diverse and representative set of publications in the domain of resource management for mission-critical applications in edge computing.

4 Foundational Concepts

As we explore the complexities of edge computing solutions designed for mission-critical situations, it is crucial to base our discourse on fundamental principles. This section provides a fundamental framework, explaining essential concepts and terms that are vital to our research. The readers can expect to gain a thorough comprehension of virtual edge computing and its significant involvement in mission-critical applications. The text will provide in-depth analysis and knowledge of mission-critical applications such as **Augmented Reality** (**AR**), healthcare, and retail. The section exhibits a systematic organization, commencing with overarching notions and progressively delving into more specific applications, thereby providing a coherent and rational progression. By comprehending these fundamental components, readers will possess enhanced capabilities to comprehend the sophisticated dialogues in the ensuing sections, consequently integrating the comprehensive storyline of the article.

4.1 Vehicular Edge Computing

Edge computing promotes data processing near the source over cloud data centers. Automobile-specific edge computing is called **Vehicular Edge Computing** (**VEC**). This technology allows vehicles to analyze data locally, making connected and autonomous vehicle services more efficient. It supports intelligent transport systems, smart cities, and urban computing, reducing latency and meeting service requirements [23]. VEC leverages vehicle and nearby infrastructure resources to optimize data-intensive and real-time applications, minimizing reliance on centralized cloud data centers. This enhances vehicular applications, contributing to safer, more effective transportation systems. Mission-critical applications often involve tasks that require real-time responsiveness and are highly sensitive to delays. In scenarios where these applications are deployed in vehicular environments, such as emergency response systems or autonomous vehicles, the mobile nature of the platforms introduces unique challenges that need to be addressed to ensure reliable and efficient operations. The integration of mission-critical applications with VEC provides a more reliable, responsive, and efficient computing environment for tasks that require high reliability, low latency, and real-time processing in vehicular settings.

4.2 Mission-Critical Applications

Mission-critical applications refer to those applications that typically perform critical functions, requiring real-time or near-real-time processing, analysis, and decision-making at the edge. In Figure 3, several prevalent classifications of mission-critical applications are depicted which are opted through reviewing related articles. The failure of these applications carries significant weight; it extends beyond IT systems and can result in severe societal and human consequences. For instance, if mission-critical applications fail, it can lead to loss of life, halting of essential societal functions, and significant financial loss. This is not a mere assessment based on technical analysis; such applications are considered "mission-critical" in broader regulatory contexts as well. For example, the European Union [24] has regulations that underline the essential nature of certain applications and services, acknowledging their critical role in society and the severe consequences associated with their failure. For each of them, an example is listed in Table 4.

4.2.1 Unmanned Aerial Vehicles. **Unmanned Aerial Vehicles** (**UAV**) are conceptualized as interconnected devices capable of forming groups and being functionally designated to achieve strategic objectives within the framework of the Internet of Drones [27]. It refers to an aircraft

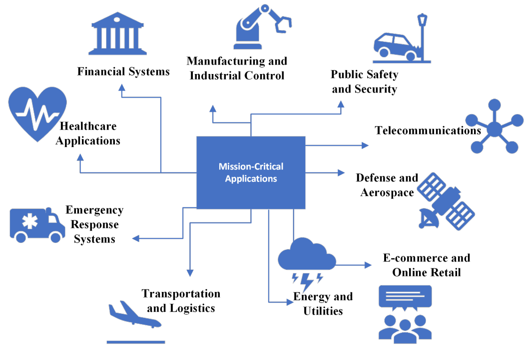

Fig. 3. Classification of mission-critical applications.

Table 4. Application Categories in Mission-Critical Context

Categories	Examples
Financial Systems	Trading platforms, Core banking systems, Risk management systems
Healthcare Applications	Electronic health records, Hospital information systems, Picture archiving and communication systems, Telemedicine platforms
Emergency Response Systems	Computer-aided dispatch systems, Emergency communication systems, Incident management systems, Disaster recovery systems
Transportation and Logistics	Air traffic control systems, Railway signaling systems [25], Fleet management systems, Supply chain management systems
Energy and Utilities	Power grid control systems, Supervisory control and data acquisition systems, Smart grid management systems, Oil and gas pipeline monitoring systems
Defense and Aerospace	Command and control systems [26], Satellite communication systems, Military-grade encryption systems, UAVs control systems
Telecommunications	Telecommunication network management systems, Mobile network switching systems, VoIP (Voice over Internet Protocol) infrastructure, Network security and monitoring systems
Public Safety and Security	Video surveillance and monitoring systems, Access control and alarm systems, Emergency alert systems, Forensic analysis systems
Manufacturing and Industrial Control	Industrial automation systems, Robotics control systems, Quality control systems, Production planning and scheduling systems
E-commerce and Online Retail	Inventory management systems, Payment processing systems

that is operated without a human pilot on board. UAVs are commonly known as drones and are used for various purposes such as aerial surveillance, photography, delivery, research, and military applications. For instance, MEC servers can perform real-time traffic monitoring and analysis at intersections [28].

4.2.2 Augmented Reality. AR overlays computer-generated images onto the real world. [29]. It uses inputs like visual images, video, sound, graphics, or GPS data. Low latency and high data processing are crucial for accurate AR services, and the MEC platform on the cloud provides localized data and meets these requirements. An example scenario involves the placement of a **Virtual Reality** (**VR**) control center on a MEC server to enhance the tracking accuracy of VR applications [30].

4.2.3 The Internet of Vehicles. The **Internet of Vehicles** (**IoVs**) connects vehicles to the Internet, enabling communication between vehicles, infrastructure, and devices [31]. It enhances transportation efficiency and safety through applications like real-time traffic information and vehicle-to-vehicle communication.

4.2.4 Smart Grid. Smart grid management systems are advanced technologies that monitor, control, and optimize electrical grids. They use sensors, communication networks, and data analytics

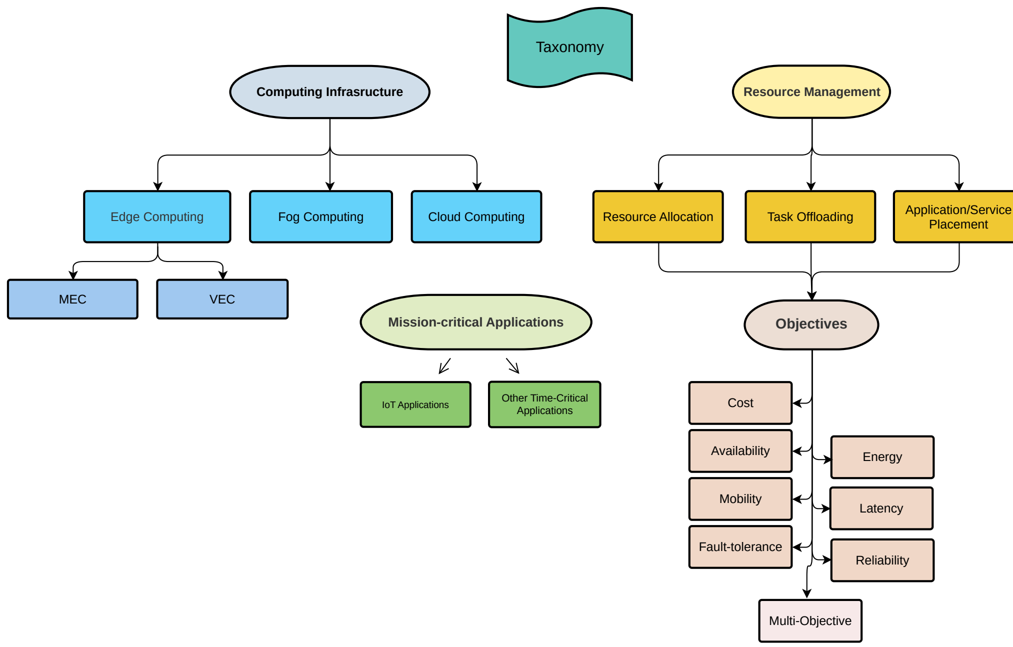

Fig. 4. Our proposed taxonomy.

to improve energy management, reliability, and renewable energy integration. These systems enable remote monitoring, outage detection, load balancing, and demand response [32].

4.2.5 Healthcare Applications. Healthcare applications in mission-critical MEC involve utilizing MEC technology to enhance healthcare services in urgent and critical situations. This includes enabling telemedicine and remote monitoring for real-time patient consultations and data analysis, supporting emergency medical services with rapid access to vital patient information, providing real-time analytic and decision support for healthcare providers, and developing mobile health applications for personalized healthcare management [33]. For instance, humanoid robots would be able to provide caretaking services to elderly patients by utilizing the collected information with MEC technology [34].

4.2.6 E-Commerce and Online Retail. Mission-critical applications play a crucial role in the success and operation of e-commerce and online retail [35]. These applications encompass various systems and processes that are essential for conducting online transactions, managing inventory, processing payments, ensuring secure transactions, and providing a seamless shopping experience for customers. The retail industry is one of the applications that can leverage the deployment of MEC technology. For instance, MEC servers can be utilized to offer WiFi connectivity to users in retail stores [36].

5 Review on the Recent Papers in Resource Management

In this section, we propose a taxonomy to organize the literature from our exploration of the selected references, as shown in Figure 4. The main branch of our taxonomy is the objective, which is either energy consumption, cost, availability, reliability, fault tolerance, mobility, or even a combination of each as a multi-objective. Each sub-section focuses on one objective and briefly describes papers related to that objective. If, within an objective, there are noteworthy papers related to different infrastructures of application types, we highlight those in separate paragraphs. For each objective, we also provide a concise summary of relevant academic publications in tabular format. This approach aims to provide the reader with a quick and comprehensive overview of the literature surveyed. Each table categorizes the reviewed literature based on its objectives and constraints, as well as on which researchers focus based on their research areas technique, such as specific types of applications or general tasks, and so on, and the deployment methods employed. These tables provide a comprehensive overview of the different research areas explored in the field of computing. Each row in the table represents a unique research study, and the columns contain specific information about each study. The "Ref" column provides the corresponding reference or citation for each study. The "Platform" column specifies the platform or environment on which the research was conducted. The "Research Area" column indicates the specific area of interest or focus in the study. The "Focus" column identifies the main focus of each paper in the area of computation offloading, resource allocation, or any other aspects of resource management in the computing era. The "Constraint" column lists any constraints or limitations addressed in the research. The "The other Objective" column highlights any additional targets or considerations in the study. Finally, the "Methodology" column outlines the approach or methodology used in each research study.

5.1 Energy Consumption

The role of energy consumption is pivotal in determining the cost of services within edge computing systems and carries substantial implications for the environment. Furthermore, in heterogeneous edge computing systems, energy efficiency is a critical concern, particularly for a wide range of latency-sensitive applications. These applications demand both rapid response times and effective power management. Therefore, researchers have proposed several solutions in resource management by using artificial intelligence or mathematical algorithms with the aim of enhancing energy efficiency in the context of edge computing.

The energy-efficient smart allocator algorithm for **Virtual Machines** (**VMs**) was represented in Reference [37], which utilized DL to efficiently allocate requests to optimal servers. The problem of minimizing energy consumption while considering task completion time [38], [39], [40] has been a recurring focus in numerous articles. With the same objective, while satisfying caching and deadline constraints, in Reference [41] a joint caching, offloading, and time allocation policy was proposed. Otherwise, authors in Reference [42] considered user mobility when making offloading decisions. They introduced a lightweight mobility prediction and offloading framework that utilizes the artificial **Neural Network** (**NN**) to predict user locations and efficiently offload compute-intensive tasks. The problem is challenging and falls into the category of a two-timescale **Mixed-Integer Nonlinear Programming** (**MINLP**) problem. The study [43] presented a mathematical framework for workload allocation, modeling power consumption, and delay functions for fog and clouds. A modified least laxity first algorithm was proposed to minimize the maximum delay threshold. A non-dominated **Particle Swarm Optimization** (**PSO**) approach was further introduced to reduce energy consumption. To continue, the new MEC model [44] took into account task dependencies and data distribution in a comprehensive manner. In this article, to address the optimization problem, a multi-objective evolutionary algorithm framework was employed. The authors in Reference [45] proposed an approach to introduce a heuristic algorithm for workflow applications with dependent tasks. It involves a task offloading step based on the **Genetic Algorithm** (**GA**) and an additional step that utilizes the **Dynamic Voltage and Frequency Scaling** (**DVFS**) technique to minimize energy usage. The suggested architecture in Reference [46] incorporates user-side decision-making algorithms that rely on locally received feedback. These algorithms were specifically designed to minimize control overhead, expedite the decision-making process, and ensure that the QoS requirements of the users were met. In the study [47], the resource allocation problem was formulated as a mixed integer linear program. This formulation was chosen because computing capacities had an impact on both the execution time of instances and energy consumption. Authors [48] addressed achieving energy-effective computation offloading in a wireless power transfer MEC. The objective was to allocate resources efficiently while considering the hard constraints. To overcome the doubly near-far problem, a user cooperative scheme was proposed, where a near user forward the tasks of a far user to the edge cloud using its surplus harvested energy. The scheme aimed to maximize users' energy efficiency, defined as the ratio of user throughput to harvested or consumptive energy while considering green network principles. The authors [49] focused on optimizing the energy consumption of stochastic tasks on DVFS-enabled embedded systems with reliability and deadline constraints. They introduced a metric for modeling the deadline constraint of soft real-time applications and evaluated their technique using a UAV electronic system. To minimize total time consumption and ensure the energy limit of mobile devices, in Reference [50] a MEC-enabled **Federated Learning** (**FL**) model was introduced. This model addressed the tradeoff between the performance of the training model, the total time required, and the power consumption of mobile devices. Minimizing completion time, a bisection-based algorithm to obtain the optimal solution was introduced [38], which had an impact on reducing energy. The problem of service migration was represented and formulated in Reference [51] using MDP, and then the **Deep Q-Network** (**DQN**) algorithm was utilized to find the best possible solution for the model. The comprehensive dataset generated from our mathematical model was used to train DNN, resulting in reduced objectives in this category [52].

In terms of application types, energy-efficient MEC plays a vital role in mission-critical applications, as frequent node deployment or battery replacement is challenging in these scenarios. Some examples of mission-critical applications that benefit from energy-efficient MEC include healthcare applications, environmental monitoring, control applications (such as Tsunami alerts, earthquake alerts, and areas prone to volcanic eruptions), and surveillance applications [53]. These applications demand effective power management to achieve energy efficiency without surpassing critical latency thresholds. Tackling challenges in offloading IoT tasks to the cloud, including suboptimal execution time, resource contention, and the need for dynamic VM optimization in resource scheduling has grabbed attention. For instance, an energy-aware resource allocation and scheduling strategy based on the multi-criteria decision-making method was introduced in [54] in the IoT environment. The authors presented a cooperative approach based on the technique for order of preference by similarity to the ideal solution, which was a multi-criteria decision-making method. The work [55] considered the entire high-complexity and latency-sensitive application and divided it into DAG-based subtasks. The goal was to satisfy reliability and latency requirements through computation offloading in order to reduce power consumption. Authors [56] calculated the expected worst-case energy consumption for time-sensitive applications implemented on the Android platform. In [57], two classes of latency-sensitive applications were considered: those requiring FO and PO. The authors formulated joint optimization problems for each class to minimize overall energy consumption while satisfying latency and computational budgets. An efficient technique by using ant colony was introduced in [58] that aimed to minimize the energy overhead in time-constrained applications modeled by DAG within such systems.

IoT services refer to the collection of applications provided or enabled through IoT devices and systems. Researchers [59] have suggested a priority-based offloading strategy that takes into account the QoS needs of mission-critical services and green resource allocation. Lyapunov optimization minimized queue lengths with energy and task priority considerations. The approach in [60] monitored the QoS requirements of IoT services and the capabilities of available fog nodes. In this article, the whale optimization algorithm was used to determine an efficient service placement plan. The evolutionary-based mechanism considered throughput and energy usage to find a desirable

Table 5. Categorized Based on the Energy as an Objective and Review the Constraints, Focus of the Research Area, and the Deployment Method

Ref.	Platform	Research Area	Focus	Constraint	The other Objective	Methodology
[38]	NOMA-based	Resource Allocation	User data	----	Task completion time	Bisection-based Algorithm
[39]	MEC	Task offloading, Resource Allocation	Task	Delay	UEU	Heuristic Algorithm
[41]	MEC	Software Caching, Task Offloading	Latency-sensitive computation services	Deadline, Cache size	----	Convex Optimization
[43]	Cloud-Fog	Workload Allocation	Task	Delay	----	Non-dominated PSO
[44]	MEC	Task allocation with dependency	Task	Delay	Task Computation time	GA, DAG
[45]	MEC	Task Offloading	Workflow	Mobility	Application execution time	DAG model, Green DVFS
[50]	MEC	Resource Allocation	Task	----	Total time consumption	FL
[58]	Heterogeneous edge computing	Task Offloading	Task	Deadline, Priority	The completion time	Modified DVFS
[64]	VEC	Resource Allocation	Vehicular	----	Communication	RL, GT
[65]	Edge-based Smart Grid Networks	Task offloading, Resource Allocation	Task	----	Task completion time	Non-convex Optimization
[66]	Edge computing based Electric Power IoT	Task Offloading, Resource Allocation on UAV	Task-Container	Business delay	----	Improved decomposition based methods
[67]	Fog Computing	Task Scheduling	IoT Application	Delay	----	RL
[68]	Edge Computing	Resource Allocation	User Workload	Mobility, Delay between edge-clouds	----	The regularization technique

placement plan. Cui et al. in Reference [61] proposed a technique for mapping tasks in parallel applications to achieve energy efficiency while ensuring real-time constraints. Moreover, taking into account the impact of **Mobile AR** (**MAR**) client energy consumption, service delay, and detection accuracy during task offloading, a function was devised to assess the energy efficiency of MAR clients by using a heuristic algorithm [62].

Concerning the VEC environment, Qin et al. [63] proposed an iterative algorithm with a double-loop structure, meaning there were two main iterative processes in the optimization routine. They used Dinkelbach's method and the block coordinate descent technique in the optimization, which are mathematical methods for solving optimization problems. The primary objective was to maximize the energy efficiency of the RIS-assisted UAV-enabled MEC system using the NOMA protocol. They aimed to achieve this by optimizing several parameters: bit allocation, transmit power, phase shift, and UAV trajectory. The authors of Reference [64] examined a difficulty related to the effective allocation of resources in autonomous vehicular communication networks. The primary emphasis was placed on the distribution of joint spectrum and power in vehicular communication networks, which has the potential to optimize energy utilization. Efficient power allocation can result in optimized energy utilization, enabling vehicular networks to employ the minimum required power for communication. Consequently, this practice has the potential to extend the operating duration of equipment or cars and conserve energy resources. In Table 5, we have listed the articles that address "energy" as an objective.

5.2 Latency

Latency plays a critical role in mission-critical applications where situations can be life-threatening, as seen in search and rescue missions. As a result, there has been a substantial body of literature addressing the need to reduce latency, particularly in the context of resource allocation and task offloading. Many articles have been published, focusing on strategies to minimize latency in these scenarios. An online algorithm, assisted by DL, was proposed in Reference [69], which could make fast decisions. In particular, an offline solver for the problem was designed, and a DNN was used to emulate the solver, allowing for online task offloading with the overall system latency minimized. In Reference [70], the authors introduced collaborative resource allocation optimization in MEC and proposed two optimal edge server deployment schemes, enumeration-based optimal and ranking-based near-optimal edge server placement algorithms, based on **Software-Defined Network** (**SDN**) technology. The research contributed [71] to the understanding and improvement of optimal **Virtualized Network Functions** (**VNF**) placement and routing for low-latency IoT services within a multi-tier cloud network. The authors also introduced learning-based heuristics for VNF provisioning, taking into account IoT device mobility and energy conservation. The work [72] focused on optimizing microservice coordination in MEC to reduce service delays for mobile users. Clarity-based services can be improved by dynamically deploying and migrating lightweight, independent microservices across edge clouds. This work addressed the challenge of increased service delays caused by frequent edge-cloud switching due to user mobility. To address reliability, authors in Reference [73] utilized a mathematical programming framework to derive an optimal task allocation, considering multiple operational constraints such as latency, energy in computation and communication, and the reliability demands of the application. In MEC with small cell networks, authors [74] presented task offloading scenarios. To achieve lower time consumption for all tasks and reduce the number of task failures, their proposed collaborative architecture shared computing resources among mobile devices, distributed small base stations MEC servers, and remote cloud in MEC-enabled small cell networks. In Reference [75], researchers focused on optimizing offloading with caching enhancements in the femto-cloud and MEC scenarios, specifically targeting multi-user environments. This scheme significantly reduced execution delays.

In terms of application types, time-sensitive applications generate a massive volume of data at the edge, which is then stored and analyzed in the cloud. However, traditional cloud infrastructure suffers from issues such as response delays, network congestion, and overall inefficiency due to the centralized nature of the system [76]. By bringing computation power and storage capabilities closer to end users at the network edge, edge computing enables the deployment of mission-critical applications on edge devices. This approach facilitates real-time data analysis, reduces latency, and enhances QoS for this type of application. For resource allocation, while considering industrial applications, the greedy border allocation into Kubernetes was introduced [77]. The authors [76] designed a conceptual model for the QoS paradigm in edge computing, specifically for time-sensitive applications. This model had the potential to enhance latency in time-critical applications. The precise placement of fog nodes is crucial, as it directly impacts the latency experienced by the hosts utilizing them. Therefore, careful assessment of fog node placement is necessary to meet the stringent QoS requirements of critical and time-sensitive IoT applications [78]. In this article, the authors formulated the fog node placement problem and proposed optimal and approximated methods to solve it by reducing latency and execution time. On the other hand, in Reference [79], the problem of task assignment in MEC for scenarios involving multiple tasks and users was investigated. The assignment was treated as a constraint satisfaction problem, and a lightweight algorithm was presented that considered task information and user mobility. The study [80] introduced a method aimed at addressing the resource allocation challenge in edge-IoT by employing DRL. Additionally, it provided a compilation of common edge-IoT applications along with their specific attributes. The prioritization of these applications was determined based on the significance of resource allocation for each of them.

Concerning the VEC environment, considering QoS parameters, the mobility of vehicles and the latency involved in offloading tasks to VEC servers could be divided into three components: the time taken to transmit the data to the nearest VEC server, the readiness time of the task on the VEC server, and the execution time on those servers. Focusing on time-critical tasks as one of the aspects of mission-critical applications within IoVs, researchers in Reference [81] reduced latency and minimized the distance between endpoints and service nodes. They prioritized the computing tasks based on their maximum tolerated latency and utilized a clustering algorithm to make informed

Table 6. Categorized Based on the Latency as an Objective and Review the Constraints, Focus of Research Area, and the Deployment Method

Ref.	Platform	Research Area	Focus	Constraint	The other Objective	Methodology
[69]	Heterogeneous MEC	Task Offloading	Time-sensitive Task	Communication, Processing Latency	–––	DL, DNN
[70]	MEC	Resource Allocation	Service	Delay	Deployment of edge servers	Enhanced Clustering
[73]	Cloud/hub/Edge	Task Allocation	Critical Applications	Reliability, Energy Budgets, Availability of resources	–––	Dual Modular Redundancy
[74]	MEC-Enabled Small-Cell Networks	Task Offloading	AR Applications	The processing time	Task Completion Time, Task Failure	Mathematical Modeling, Optimization Technique
[75]	MEC	Task Offloading, Offloading Decision	Application	Caching Enhancement	Single-user, Multi-user	Cooperative GT
[77]	Fog Computing	Container orchestration	Industrial Fog Application	Failover time requirements	–––	Heuristic Algorithm
[80]	Edge-IoT	Resource Allocation	IoT Applications	Priority of applications' sensitivity	Quality of Experience	DRL, DNN
[81]	IoV	Task Offloading, Resource Allocation	Time-critical Tasks	Power, Priority	–––	Priority-based Clustering, Constrained Optimization

offloading decisions. The research [82] focused on addressing the problem of optimally placing **Vehicle-to-Everything** (**V2X**) services in a hybrid cloud/edge environment. To achieve this, an analytical optimization model was developed. The objective of this model was to place V2X services on computing nodes in a way that minimizes the aggregate average delay/latency. The model considered various constraints, including delay, computation resource availability, redundancy, and placement constraints. Articles that consider "latency" as an objective are cataloged in Table 6.

5.3 Reliability

Given the nature of computing platforms, applications deployed in edge-cloud computing environments inevitably confront a multitude of challenges stemming from limited bandwidth, the unreliable and heterogeneous nature of wireless communications, computation offloading, and security considerations [11]. Deploying these devices in critical applications like search and rescue missions or monitoring and controlling critical infrastructure poses a significant challenge to ensuring the overall reliability of the application. A few articles that focused on ensuring reliability have been investigated here. The objective of Reference [83] was to ensure high service reliability while minimizing the reserved edge resources necessary for system redundancy. This was achieved by proposing a heuristic algorithm. In work, Reference [84] an intelligent agent using deep Q-learning at the edge computing node developed a real-time adaptive policy for resource allocation. The goal was to improve average end-to-end reliability for offloaded tasks from multiple users.

In 5G mobile networks, high-rate mission-critical traffic requires support from multiple radio access technologies and dynamic resource allocation. To address this in Reference [85], a softwarized 5G architecture was introduced along with a mathematical framework to model critical session transfers in the network. With regard to resource allocation, [86], a metric to measure the balance condition of physical resources was proposed, and also an algorithm that optimized resource allocation based on this metric to improve resource utilization and acceptance rate.

In terms of application type, certain tasks within mission-critical applications hold the utmost significance, necessitating a certain level of reliability from MEC to ensure their successful execution. This paragraph delves into various articles that concentrate on this specific objective, exploring ways to enhance MEC's reliability for handling these crucial tasks. Mission-critical machine-type communications applications have challenges in terms of insufficient radio resource management and orchestration. One solution [87] involved the optimal selection of **Multi-Technology Gateway** (**MTG**) and resource allocation. This was achieved through a combination of predicting MTG loading using recurrent NN-based long short-term memory and adjusting MTG loading DRL techniques. These challenges lead to a decrease in end-to-end communication reliability, particularly in edge areas. Mission-critical task offloading in non-orthogonal multiple access-assisted MEC networks was considered [88], where local information collected from multiple local devices was involved. This design was constrained by strict delay requirements and limited energy resources. Authors [83] developed a resource model for latency-critical services in distributed edge computing. To minimize reserved placeholders for edge node failures, they proposed a fast heuristic algorithm. Authors [89] focused on improving service reliability in the deployment of application instances on edge servers. They introduced the coverage-reliability for the K-Budgeted edge application deployment problem and proposed optimal and greedy approaches to solve it. In Reference [90] focused on a scenario where multiple edge nodes collaborate to complete the AR task. The task's component dependencies were modeled by a DAG, and an integer PSO-based algorithm was proposed. In Reference [91], the authors presented a method for optimizing the allocation of hardware resources in a multi-tier MEC hierarchy. To enhance the overall computational capacity of a 5G-based MEC system, a parametric Bayesian optimizer was employed for allocating hardware resources. It was shown in Reference [92] that extending the deadline of IoT applications can effectively avoid QoS violations without impacting bandwidth consumption, application completion time, or the execution time of the approach. The work focused on achieving an equilibrium state where the bandwidth consumption of IoT applications was minimized while simultaneously maximizing the reliability level of these applications during task offloading. Conversely, the article [90] addressed collaborative AR task completion among multiple edge nodes. Task dependencies were modeled using a DAG. An integer PSO-based algorithm was designed, but its computational complexity restricts real-time applicability for AR services. Nguyen et al. [93] proceeded to discuss the MEC platform that had its resources spread out over a metropolitan region. Additionally, a scenario that entails the implementation of an application that was exceptionally sensitive to latency like AR applications. They emphasized difficulties associated with allocating resources to support the aforementioned application, to ensure satisfactory performance thresholds.

5.4 Cost

Through this section, valuable insights into MEC resource management for cost optimization are uncovered. GA was combined with the problem of computing offloading of multi-service tasks based on edge computing, which was proven to be NP-hard in Reference [94]. The influence factors of genes on chromosomes were considered, and a probability formula for crossover selection was designed. In work [95], it was pointed out that the **Cloud Assisted Mobile Edge** (**CAME**) computing framework tackled scalability challenges in MEC by outsourcing mobile requests to diverse cloud instances. Proposing a solution to optimize resource provisioning and cloud tenancy within CAME, the proposed optimal resource provisioning algorithms exploited problem convexity. Authors [68] addressed the problem of online edge resource allocation by formulation mixed nonlinear optimization problem. They proposed a mobility-agnostic algorithm, decomposed the problem into subproblems with regularized objective functions, and utilized convex programming methods for their solution. An optimization scheme for computation offloading and resource allocation in a network with multiple wireless access points and MEC was proposed in Reference [96]. The paper's objective was to minimize the cost of the system. Authors in Reference [97] explored MEC in an ultra-dense network, where mobile users could choose from multiple Base Stations for computation offloading. They formulated the problem as MDP to determine an optimal computation offloading policy to minimize long-term costs.

Several research studies have put forward proposals to optimize joint offloading and compute resource allocation within the edge infrastructure, like [98] and [99]. The work [100] considered various network states such as channel conditions, computation task queue, and remaining computation resources of end devices in this area. It formulated the decision-making problem at end devices as a MDP and employed a RL approach to address it.

Researchers are working on addressing security and privacy concerns in MEC, which is another significant area of focus. The authors [101] aimed to reduce the cost of energy by proposing a distributed algorithm that optimizes various parameters, such as clock frequency configuration, transmission power allocation, channel rate schedule, and offloading strategy selection. They considered a satisfactory user experience while achieving high energy efficiency in MEC. In order to meet the deadline constraint and minimize data access costs, the problem of scheduling data replication had been formulated in [102] as an integer programming problem. To address this problem, a faster meta-heuristic algorithm was developed and applied. Authors [103] analyzed the relationship between minimal purchasing cost, computation delay, and available computation resources to maximize long-term profits. They designed schemes for resource reservation, expansion, and real-time purchasing to optimize resource allocation dynamically.

In terms of application type, research papers have been analyzed to identify effective resource management techniques for cost reduction in MEC and their influence on mission-critical tasks. The study [104] pointed to minimizing energy costs for mission-critical IoT in a MEC system. The problem was decomposed into two sub-problems: task offloading decision-making and resource optimization. In Reference [105], service placement occurred at a larger scale, specifically at the level of frames, to prevent system instability. Request scheduling for data-intensive applications was performed at a smaller scale, known as slots, to support real-time services. A budget constraint was furthermore imposed to control the operation costs associated with service replication and migration. Lou et al. in Reference [106], proposed deciding which tasks, along with their subsequent tasks (excluding the final task), should be executed in the cloud to meet the overall project deadline for latency-sensitive applications such as LIGO, which is used in physics for detecting gravitational waves and involves CPU-intensive tasks that consume large amounts of memory and CyberShake which is used to characterize earthquake hazards by generating synthetic seismograms. To address scheduling decisions in Reference [107], the IoT application was allocated to a specific SLA category, such as gold, silver, or bronze. These categories indicated the relative significance or criticality of each application. Additionally, each category was assigned a penalty weight that measures the financial consequences of failing to meet the performance requirements associated with an application in that particular category.

Concerning the VEC environment, in Reference [108], latency as a constraint with regard to mobility was examined to reduce the PO cost. Considering the same constraints, authors [65] proposed a model in which a vehicle passed through the coverage area of an **Access Point** (**AP**) at fast-moving speed, and the offloading tasks could be processed by the MEC servers deployed at AP independently to drop the system cost. To ensure the delay tolerance of all vehicular user equipment, the proposed problem in Reference [109] aimed to minimize the system cost. This problem was formulated as a MINLP. To aim for the identical goal, the work in Reference [110], the vehicle-DQN algorithm, was proposed. The vehicle-DQN algorithm incorporated a **Vehicle Adaptive Feedback** (**VAF**) algorithm, which determined the priority assignment of processing tasks for service vehicles. Subsequently, the vehicle-DQN algorithm utilized the results from VAF to develop a task-offloading strategy. Table 7 encompasses the articles where "cost" is explicitly addressed as an objective.

5.5 Mobility

Although mobility-aware scheduling has been extensively investigated in MEC, primarily as a constraint that must be managed (e.g., Refs. [111], [112]), there are few studies that have concentrated

Table 7. Categorized Based on the Cost as an Objective and Review the Constraints, Focus of the Research Area, and the Deployment Method

Reference	Platform	Research Area	Focus	Constraint	The other Objective	Methodology
[91]	5G Network	Resource Allocation	Task	Budget, Latency	Higher Computational Capacity	Sequential Model-Based Optimization
[95]	Cloud Assisted MEC	Resource Provisioning	Edge nodes	Delay tolerant rate, Delay-sensitive applications ensuring QoS	—–	Convex Optimization
[96]	MEC	Task Offloading, Resource Allocation	Task	Task to server via multiple wireless APs	Transmission power allocation	GT
[97]	MEC	Task Offloading, Offloading Decision	Computation Task	—–	Computation task execution delay the Handover delay trade-off	DRL
[98]	MEC	Service Placement, Computation Scheduling	Service	Multiple Types of Resources	Feasibility	Optimization Theory
[99]	MEC	Task Offloading, Resource Allocation	AR Applications	Energy, User Mobility	System performance under different task	Convex within GT
[102]	Edge Computing	Replication Management	Latency-sensitive Data Workload	Deadline	Violation of number of data transmission	Metaheuristic Algorithm
[103]	Edge-Cloud	Resource Management	Computation Task	Available computation resources, Delay	—–	ML
[104]	Mission-critical IoT in MEC	Task offloading, Offloading Decision, Resource Allocation	Short Packet Transmission	—–	—–	MINLP
[105]	Edge-Cloud	Service Placement, Request Scheduling	Data-intensive Applications	Communication Budget	System Stability	GA
[106]	MEC	Task Scheduling	Latency-sensitive Applications	Tight Deadline	—–	Heuristic Algorithm
[108]	VEC	Task Scheduling	Task	Mobility, Latency	—–	Heuristic Algorithm
[109]	NOMA-Enabled VMEC	Offloading Decision-making, Resource Allocation	Task	Latency requirement	Vehicular User Equipment	Mathematical Optimization

on mobility enhancement approaches, which are designed to actively enhance user or service mobility as part of resource management. Mouradian et al. in Reference [113] explored the cost-efficient deployment of robots for a search and rescue use case during large-scale disaster management. The majority of the examined physical robots are mobile, unlike traditional IaaS resources such as data centers. This mobility characteristic of robots significantly raises the likelihood of QoS degradation. Therefore, the mobility aspect of robots is of great importance. With regard to user mobility [114], the task offloading problem was decomposed into two subproblems: a convex computation allocation subproblem and an INLP offloading decision subproblem. These algorithms focus on optimizing the system utility under the constraints of shortages of resources, and task latency. The primary goal of authors in [115] was to optimize the availability of resources utilized in response to MEC requests. They incorporated mobility prediction to minimize unnecessary re-allocations due to user mobility. To predict user flow across various MECs, a Gaussian method is utilized, aiming to anticipate resource utilization. Additionally, service time and network consumption were taken into account to minimize service allocation. Vasilakos et al. [116] presented a theoretical framework that utilizes the capabilities of ML at the network's edge. This framework involved deploying intelligent models near mobile users, allowing them to take proactive or reactive measures such as predicting mobility for ultra-low-latency 5G streaming services. Based on the mobility of users, the work [117] proposed an approach based on thresholds, which created new registries when application provisioning times became excessively high while deprovisioning existing registries located far from users to prevent resource wastage. A selection of the relevant articles is presented in Table 8.

5.6 Availability

In terms of the availability of physical resources, edge computing resource allocation involves optimizing computing resources, storage capacity, and network bandwidth in a resource-constrained

Table 8. Categorized Based on the Objectives and Review the Constraints, Focus of the Research Area, and the Deployment Method

Obj	Ref.	Platform	Research Area	Focus	Constraint	The other Objective	Methodology
Reliability	[83]	Edge Computing	Resource Provisioning	Latency-critical Service	Delay	——-	Mathematical Model, Heuristic Algorithm
	[84]	Edge Computing	Resource Allocation	Task	Latency	——-	Deep Q-learning
	[85]	5G Networks	Softwarized 5G Architecture	Mission-critical Data	Packet delay	——-	Mathematical Model
	[88]	NOMA-assisted MEC	Task Offloading	Mission-critical Tasks	Available resource, Delay, Limited energy budget	——-	Modified Block Coordinate Descent
	[89]	MEC	Application Deployment	Application Instances	Application Deployment Budget	Both Small and Large-Scale Scenarios	Heuristic Algorithm
	[90]	MEC	Code-Partitioning Offloading	AR service	Latency	Service Failure Probability	Metaheuristic Optimization
	[92]	MEC	Task Offloading	IoT Applications	——-	Bandwidth Consumption	Heuristic Algorithm
Mobility	[79]	MEC	Task assignment	Task	User Mobility	Task execution	Heuristic Algorithm
	[114]	MEC	Task Offloading, Resource Allocation	User Task	User Mobility, Latency	System Utility	Heuristic Algorithm
Availability	[119]	Edge-Cloud	Replica Placement, Load Balancing	Data Node, Data Blocks	Data Availability	Improves Storage, Service Performance	Mathematical Model Based
	[121]	Fog Computing	App-Aware Placement	Microservices	Bandwidth Consumption	Throughput	MINLP, Heuristic Algorithm
	[122]	Edge Computing	Replica Placement	Data Blocks	——-	Improves Storage, Service Performance	Metaheuristic Optimization
	[123]	Vehicular Fog Computing	Resource Allocation	MEC Service	The Handover Frequency	——-	Mathematical Method
Fault Tolerance	[124]	MEC	Service Scheduling	Service	Mobility, Threshold	Energy	RL
	[127]	Edge-Cloud	Task Allocation	IoT Application	Latency, Reliability Requirements, Availability of Resources	Deadline Missing Ratio	Mathematical Model
	[128]	Fog Computing	Task Allocation	Real-Time Task	Communication delay, Task Reliability	Successfully Executed Task	RL

edge environment. Efficient resource management ensures the edge infrastructure can handle workloads, provide reliable services, and meet data availability requirements. The adaptive replica placement strategy and associated algorithms considered user access characteristics, the number of replicas, replicas' location, and resource-related factors (such as node performance and replica consistency) to optimize data availability while effectively managing the aforementioned physical resources in that environment. Considering both availability and reliability [118], the replica synchronization strategy improved data readability by reducing write latency and enhancing write throughput, ensuring replica consistency. The replica recovery strategy enhanced data reliability by reducing write time and improving read throughput. Adding the cost factor, authors in Reference [119] proposed a resource management strategy to optimize workloads and costs in the edge cloud. They also presented a replica management strategy to enhance the user experience, reduce storage overheads, and ensure data consistency. In Reference [120], the authors developed analytical models and solutions to evaluate the availability of IoT systems with Byzantine fault-tolerance for mission-critical applications. They aimed to understand the relationship between the number of servers in the system, the proportion of honest users (non-malicious nodes), and the overall availability of the network.

In terms of application type, Faticanti et al. [121] focused on the efficient orchestration of applications in a combination of a cloud module and a fog module, fog computing. Multiple constraints arose from the computation and communication demands of fog applications, available infrastructure resources, and the location of target IoT objects. They developed a greedy algorithm to solve the app-aware placement formulation. As the Internet of Everything expands, IoT applications in areas like smart manufacturing and transportation are growing. These applications have needs like quick data transmission, adequate bandwidth, and security. To address these demands and provide users with swift connections, instantaneous services, smart applications, and secure data management, authors in Reference [122] offered services that place copies of data at this network edge. They considered four practical use cases and discussed how their approach was implemented in smart manufacturing, transportation, cities, communities, and AR.

Concerning the VEC environment, Souza et al. [123] employed RL to optimize resource availability by minimizing handover frequency in vehicular scenarios through intelligent resource placement. Table 8 displays a compilation of the pertinent articles.

5.7 Fault Tolerance

Smart devices used as edge nodes are resource-constrained and face a higher failure rate than traditional distributed systems. Therefore, ensuring reliability and application availability through fault-tolerant systems becomes crucial. To reach this goal, the authors [124] represented fault tolerance in a diverse multi-user MEC environment. To enhance service offloading efficiency, a semi-online and RL-based model was introduced to determine users' offloading behavior. A threshold value was incorporated to prevent offloading failure due to resource overload. An adaptive check-pointing, like the previous article, the offloading algorithm was developed to ensure reliable offloading as well.

In terms of application type, to ensure QoS, the infrastructure responsible for QoS management must possess a thorough understanding of the application's characteristics and the SLA requirements in order to effectively enforce QoS [125]. To enable the execution of latency-oriented IoT applications within an edge computing environment, workflows offered a standardized approach for managing multiple applications across distributed systems [126]. In Reference [127], a fault tolerance technique for critical IoT applications that aimed to increase task reliability in distributed edge computing was introduced. The technique involved locally assessing the reliability of an edge node and providing fault tolerance to enhance overall application availability. The method involved simulating a smart battlefield scenario and using decentralized edge computing and fault tolerance techniques to handle fault tolerance in a critical and changing setting like a battlefield. To enhance the reliability of executing real-time tasks in fog-based IoT systems, in [128] a RL approach was proposed to assign primary and backup tasks to appropriate nodes in this system. The authors mentioned that the existing methods often focused on delay and functionality but neglected the reliability of real-time tasks in dynamic environments. The authors [129] highlighted the importance of a rapid response from the healthcare system, considering it a safety-critical system. The fault-tolerant data transmission feature ensured that the transmission of data between the storage nodes and the processing nodes were reliable in the fog environment.

Given our discussion related to objectives including reliability, mobility, availability, and fault-tolerance within the context of resource management, we note a strong interconnection among these objectives. In order to illustrate the platforms where these objectives have garnered significant attention and have become primary targets for researchers, as well as the constraints they are bound by, a selection of the relevant articles is presented in Table 8.

5.8 Multi objectives

5.8.1 Simultaneous Optimization of Energy Consumption, Cost Minimization, and Latency Reduction, with or without reliability. With the perspective of fog computing, researchers in [130] decoded the interplay between latency, reliability, cost, and energy in the resultant resource provisioning. This analysis provides valuable insights into how these factors impact the process of resource allocation and helps understand their effects on resource provisioning outcomes. In an identical computing era, [131] introduced a fault-tolerant scheduling algorithm that addressed the issue of communication reliability in IoT. The proposed algorithm incorporated energy-efficient check-pointing and load-balancing techniques, resulting in improved performance for the delay, energy consumption, execution cost, network usage, and the total number of executed modules.

In Reference [132], the tradeoff between conflicting objectives was considered. Reducing the task processing time leads to higher costs for VMs. Similarly, maximizing task associativity requires increasing the number of VMs, which also increases the service cost. To address this, a task scheduling and load-balancing technique based on an integrated accelerated PSO algorithm was introduced. Taking into account energy consumption, time delay, and computing cost, [133] considered joint optimization of resource allocation and offloading decisions. The main goal of the research was to optimize the performance of a MEC-based industrial IoT network. The problem was divided into two subproblems: the first subproblem was solved using the Lagrangian multiplier method, and the other by GT. In [134], researchers developed a computation offloading scheme in two steps: centralized user clustering and distributed computation offloading. They used the K-means clustering algorithm, considering user priorities as features. Clusters with high and low priorities were assigned for edge computing and local computing, respectively. Using DQN, an optimal distributed computation offloading policy was learned for users in other clusters. This was done to keep long-term system costs as low as possible. The article [135] introduced a resource provisioning model for microservices in smart healthcare systems, emphasizing device mobility and addressing service migration mobility. It effectively lowers operational costs and mitigates its impact on energy consumption. In [136], authors suggested employing iterative methods to enhance the assignment of servers to tasks, taking into explicit consideration agent mobility. Their algorithm aimed to optimize the allocation process, considering factors like communication latency. In this article, the application use case involved a fire emergency management service implemented within a smart factory setting.

Concerning the VEC environment, the work in [137] introduced the cooperative offloading and task scheduling algorithm, which iteratively scheduled tasks and generated near-optimal solutions within a polynomial time complexity. The algorithm optimized task execution by considering the tradeoff between latency, cost, and resource performance, enabling efficient utilization of available resources. In this article, different types of applications were used for generating experience results, such as E-transport and healthcare.

5.8.2 Only two conflicting objectives together. Several research works have suggested joint optimization of computational costs and another objective in edge infrastructure. In the research [138], the authors used GA to address the service placement problem for critical IoT applications, aiming to optimize cost and availability by placing multiple replicas in different network parts. The second subproblem was addressed using GT principles. Researchers in [139] calculated cost-optimal groupings, assigned compute resources, and placed application components while meeting end-to-end latency constraints. They not only deployed tasks onto the edge and cloud resources accordingly but also optimized the placement of externalized data to minimize data access latency. The tradeoff between system cost and reliability for deadline-sensitive IoT services was investigated in [140], where the authors presented the weighted best fit decreasing algorithm, driven by the best fit decreasing algorithm used to solve the bin packing problem.

To address the high sensitivity of IoT applications to latency, especially in critical sectors like medical and healthcare industries, [141] introduced a non-dominated sorting GA tailored for data-intensive IoT applications in fog infrastructure. The algorithm aimed to enhance data availability, reduce data access costs, and account for latency. A privacy-preserving and cost-efficient task offloading scheme was represented [142]. The scheme, based on the generic Lyapunov optimization framework, aided in preserving user privacy while ensuring the best possible user experience. The article [143] focused on cost and energy consumption in a multi-tier MEC system, achieving this by optimizing task offloading, transmission power and radio resource allocation, and server computation. The problem is formulated as a mixed-integer non-convex program, taking into account devices' latency requirements. An online dynamic resource allocation scheme based on

GT was presented in [144], which adapted to changing workloads, ensuring application latency requirements. The study utilized two types of applications: a near real-time scenario involving the well-known electroencephalography tractor beam game and a delay-tolerant scenario with a video surveillance/object-tracking application.

However, the topic of reducing energy usage while considering latency has attracted consistent attention in numerous articles. The authors in [145] introduced a technique named Mobile-Kube, which utilizes the RL algorithm to decrease the latency of Kubernetes applications on MEC devices. Ramzanpoor et al. [146] proposed a mechanism that enhanced the reliability of IoT applications using the **Cuckoo Search Algorithm** (**CSA**). They formulated the component deployment as a multi-objective optimization problem, aiming to reduce both energy consumption and latency in fog structures. Similarly, a joint-clustering algorithm was introduced in [147], utilizing probabilities to cluster fog nodes based on specific application requirements. This algorithm enables the selection of appropriate fog nodes and allows a node to be part of multiple clusters simultaneously, unlike traditional clustering methods. The task assignment and power allocation were jointly formulated in [148] to minimize the total task execution latency, taking into consideration user mobility, task properties, and energy constraints of the user device. Initially, a GA-based evolutionary scheme was proposed to solve the formulated MINLP problem. The diverse requirements for service delay and energy consumption in different application scenarios are evident. For instance, in industrial IoT applications, minimal delay is critical to avoid disasters, while autonomous driving prioritizes low energy consumption alongside critical latency needs. Online gaming on mobile devices, like AR, demands low service latency and high energy efficiency, necessitating a balance between these factors to enhance the user's quality of experience. Hu et al. [149] conducted a comprehensive investigation of the fundamental tradeoff between energy efficiency and service delay in a multi-server multi-user IoT network. They considered various factors, such as user mobility, random channel states, and stochastic task arrivals to explore this tradeoff. In another study by Matrouk et al. [150] tasks were categorized into four types (real-time, non-real-time, delay-sensitive, and resource-intensive) and scheduled based on constraints like priority and data size. The Di-process modular NN was used to calculate the weight value of each task, which was then assigned to the optimal fog node using the capacity-based Hungarian assignment algorithm. This approach effectively optimized task allocation in the network. Authors [66] addressed the UAV inspection business collaboration problem in the electric power IoT using edge computing. A two-stage mechanism is proposed to optimize energy consumption and delay, achieving a balance between the two objectives.

Considering fault tolerance, the article [126] introduced a fault-tolerance methodology based on check-pointing and replication. In another work focusing on VNF, a placement scheme specifically designed for MEC within a **Network Function Virtualization** (**NFV**) environment was proposed in [151]. The problem of VNF placement was tailored to ultra-reliable low-latency communications with two conflicting objectives: minimizing access latency and maximizing service availability. By addressing these various aspects and tradeoffs, researchers aim at developing efficient resource management strategies tailored to different application requirements in the context of MEC, IoT, and fog computing.

Concerning the VEC environment, being multi-objective, in [152], an optimization problem to minimize the weighted sum of service delays for all IoT devices and power consumption for UAVs were presented. This was achieved by jointly optimizing the position of the UAV, the allocation of communication and computing resources, and decisions related to task splitting. Considering the limited resources of mobile devices [153], mobility of devices and migration of services were considered in the offloading technique, resulting in an improvement in the energy and latency of industrial IoT applications. Authors [154] introduced a new mobile edge mechanism known as V-edge, which involves deploying edge computing capabilities on vehicles. Their focus was on

Table 9. Optimization Considerations in MEC: Costs, Energy, Reliability, Latency, and Availability

Objective	Parameters
Energy	– The energy consumed by mobile devices during task execution – The energy consumed during data transmission between mobile devices and edge servers or other network entities – Radio Access Network (RAN) Energy Consumption – The overall energy consumption of the entire MEC system, considering all components involved – The beneficial offloading energy
Cost	– The energy consumed by mobile devices, edge servers, or network communication – The computational cost involves efficient utilization of available resources. – The communication cost involves optimizing data transmission and network resource usage. – The resource costs – The trade-off between the desired quality of service and the resources required to achieve it
Reliability	– It involves ensuring the availability and robustness of edge servers to handle incoming tasks and deliver reliable services – Fault Tolerance: It refers to the ability of the system to continue operating reliably in the presence of failures or faults – The resource Reliability
Latency	– The execution latency – The communication Latency – Offloading decision latency – The time taken to receive a response or result after task offloading and execution (Response latency)
Availability	– The device availability – The edge server availability – The service availability named data storage

maximizing the completion of tasks with sensitive deadlines using the V-edge. They optimized both mobility and cost. In work [155] proposed an adaptive resource allocation approach to address queuing delays in mission-critical applications. By substituting computing configurations and efficiently controlling data drop rate, MINL stochastic optimization minimized computing quality loss while meeting task deadlines and balancing queuing delay in multi-cell, multi-user edge computing networks. The algorithm adaptively adjusts task placement, radio resource, computation configuration, and data drop, enhancing the long-term user experience for vehicles without prior knowledge of future network states.

Table 9 provides a complete review of edge computing optimization issues. It displays many aspects such as costs, energy consumption, delay, dependability, and availability. Highlighting these criteria provides a comprehensive understanding of aspects affecting MEC system efficiency, performance, and effectiveness. This information can guide decision-making processes and inform the design and development of edge computing and MEC solutions that prioritize these optimization objectives.

In Table 10, we examine papers on resource management that consider multiple objectives. Many studies now treat various QoS factors as limitations rather than aims. Numerous surveys have examined the combination of energy and delay, thus we have focused on other variables in this table. The "Others" column in the "Objectives" may include delay, availability, performance, and makespan.

6 Open Issues

This section identifies and analyzes key open issues in resource management for edge computing, with particular emphasis on mission-critical applications that demand high reliability, low latency, and real-time responsiveness. Despite growing interest, many aspects of resource management

Table 10. Categorized Based on more than One Objective and Review the Constraints, Focus of the Research Area, and the Deployment Method

ref	Platform	Research Area	Type	Constraint	Objectives					Methodology
					Energy	Latency	Cost	Reliability	Others	
[70]	MEC	Resource Allocation	Application service	Avg of response delay, Service type			✓		✓	Clustering enhanced Heuristic
[71]	Multi-Tier MEC	VNF Placement	IoT application	Implementation cost	✓				✓	Approximation Algorithm
[72]	MEC	Microservices Coordination	Microservice	Number of Vehicles, the Shortest-Path			✓		✓	RL
[87]	Software-Defined MEC	Offloading Decision, Resource Allocation	Task	MTG loading prediction				✓	✓	DRL, NN
[116]	5G Network	Conceptual Edge architecture	Service	Wireless and MEC Mobility Application level Downtime		✓			✓	ML
[124]	MEC	Service Scheduling	Service	Mobility, Threshold	✓			✓		RL
[126]	Edge Computing	Replica Placement	Latency-oriented IoT Application	Deadline, Application Latency				✓	✓	Uncoordinated Check-pointing Model
[130]	Fog-Enabled IoT Network	Provisioning Resources	Fog Instances	Latency Reliability	✓	✓	✓			DVFS Technique, Analytical Model
[131]	Fog Computing	Load Balancing	IoT Application	—–-	✓		✓		✓	Model based Optimization
[132]	MEC	Resource Allocation	Task-VMs	Divide Task			✓		✓	Metaheuristic Optimization
[133]	MEC	Task Offloading, Resource Allocation	User Task Migration	Mobility	✓		✓		✓	GT
[134]	IoT Networks	Resource Allocation	Task User Clustring	———	✓	✓	✓			ML
[135]	Smart Healthcare Systems	Resource Provisioning	Microservices	Delay, Mobility of Nodes	✓	✓	✓			Metaheuristic Alogrithm
[136]	Edge-Cloud	Resource Allocation	Robotic Workflow	Mobility	✓		✓		✓	GA
[137]	Vehicular Fog-Cloud	Task Scheduling	Microservices	Mobility-Aware,Deadline		✓	✓			GA
[138]	Edge Computing	Service Placement, Load Distribution	Latency-sensitive Applications	Deadline, Resource, Replica of an Application			✓		✓	GA
[141]	Fog Computing	Replica Placement	Data-intensive IoT Applications	Latency			✓		✓	Metaheuristic Algorithm
[142]	MEC	Task Offloading	Task	Privacy			✓		✓	Lyapunov Optimization
[144]	Cloud-Fog	Multi parties placement	Task-VMs	Execution time of applications, Deadline	✓		✓			DRL, NN
[146]	Fog Computing	Optimizing deployment of application components	Fog Nodes	Reduction of Bandwidth Wastage, Resource	✓	✓				A nature-inspired Optimization Algorithm
[147]	Fog-Cloud	Application Placement	Applications	Application Requirements	✓	✓				Clustering Enhanced
[148]	MEC	Task Scheduling, Resource Allocation	Task	Mobility-Aware	✓	✓				GA
[149]	MEC-Enabled Mobile IoT Networks	Task Offloading	Service	Maximum CPU-cycle Frequency	✓				✓	Control Algorithms
[151]	MEC-NFV	VNF placement	VMs	Delay		✓			✓	GA

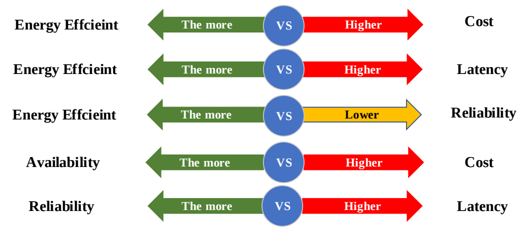

Fig. 5. The conflicting objectives.

remain underexplored, especially when considering the unique constraints of edge infrastructure, such as limited computational capacity, energy availability, unstable connectivity, and geographic distribution. We highlight specific research gaps where existing solutions fall short of meeting the requirements of edge-native, mission-critical deployments. These challenges not only pose technical difficulties but also represent opportunities for impactful innovation across various dimensions of edge resource management.

This section seeks to enhance the narrative of the study by outlining these concerns and suggesting key areas where future research might make substantial contributions. In doing so, it establishes a foundation for further exploration beyond the existing state of knowledge.

6.1 Multi-Objective Solutions

The contradiction between energy, latency, availability, cost, and reliability in edge computing arises from the fact that optimizing one objective often comes at the expense of another, as shown in Figure 5.

These objectives are interconnected performance measures; thus, improving one may hurt another. Managing these competing goals in edge computing task offloading and resource allocation is difficult. From the literature review, it is evident that many studies have concentrated on "energy consumption and latency". Meanwhile, the challenges posed by reliability restrictions, which demand cost-effective solutions, have also been a major focus. Several multi-objective articles do not meet the varying needs of different applications and user requirements, especially in addressing inherent conflicts. The strong impact of application types on these factors demands additional attention in joint task offloading and resource allocation methodologies.

Redundancy and fault-tolerant methods in multi-objective mission-critical systems are important. Mission-critical systems cannot afford a single point of failure that could be disastrous. Therefore, multi-objective solutions should be fault-tolerant for high availability and dependability. Redundancy, where vital functions are repeated or distributed over numerous edge nodes, reduces the chance of hardware or software failure.

Researchers and practitioners must find unique techniques that combine various objectives with redundancy and fault-tolerant tactics to provide robustness and resilience in mission-critical edge computing settings. Thus, mission-critical applications can be better protected from failures and disturbances, ensuring high system availability and flawless performance even in the most demanding scenarios.

6.2 Fault Tolerance

Edge computing devices may fail, lose power, or lose connectivity, which presents new issues compared to cloud systems. Most edge computing task offloading and resource provisioning studies have focused on reliability as a constraint rather than an objective. Although some works are focusing in MEC on fault tolerance, the majority of works on fault-tolerance techniques have focused on cloud systems. Data replication, load balancing, redundancy, and failover solutions have been vital to edge computing resource management, but edge devices and their dynamic settings require a more personalized approach. The ability to endure faults and system interruptions is crucial to edge computing mission-critical operations, requiring both proactive methods to prevent or mitigate issues before they occur and reactive methods to recover quickly and maintain service continuity after failures. This makes fault tolerance essential for mission-critical applications. Edge computing, where devices are resource-constrained, is overlooked in the cloud systems' focus.

Fault tolerance mechanisms tailored to edge devices and mobile units are needed to ensure dependable operation, fault recovery, and service continuity in edge environments. Considering mobile network architectural changes is crucial. Microcells, small cells in mobile networks, are becoming more widespread [156]. With several towers per region and lower output power, microcells reduce interference and boost subscriber bandwidth. When one microcell fails, an adjacent microcell can dynamically increase its power to cover its area, improving edge computing fault tolerance. Thus, future research should focus on edge device fault-tolerant methods and mobile network infrastructures for reliability and fault tolerance.

6.3 Green Edge Computing

Mission-critical applications in edge computing often come with their own unique set of challenges beyond just energy consumption. Considering the nature of these applications and the possibility of resource-constrained mobile units, it becomes crucial to address multiple aspects to ensure the effectiveness of "Green Edge Computing" for such scenarios.

In the context of mission-critical applications, the issue of energy efficiency should be balanced with the performance and reliability requirements of the applications. While reducing energy consumption remains vital, it is equally important to optimize resource utilization to meet the real-time demands of critical operations. This entails exploring techniques that prioritize critical tasks, efficiently manage computing resources, and minimize unnecessary data transmissions to preserve battery life in mobile units. Moreover, in mobile edge computing scenarios, energy conservation should extend beyond the devices themselves to include the development of intelligent algorithms that minimize data processing and communication overhead. Task offloading and workload partitioning strategies should be tailored to the specific needs of mission-critical applications, considering the available battery capacity and network conditions.

6.4 Availability

High availability ensures that edge services and resources are continuously accessible to users, even in the presence of failures or varying demands. Edge devices typically have limited computing power, storage capacity, and battery life. Managing these scarce resources efficiently is therefore a significant challenge. Resource allocation algorithms must consider the available resources, especially for mission-critical applications on the edge device, and distribute them optimally among various tasks and applications. This objective emphasizes the need to maintain service levels under changing conditions. Based on our investigation of many recent articles, this objective is in most cases, considered indirectly.

6.5 Heterogeneity

Edge computing environments include smartphones, IoT devices, and edge servers. This diverse landscape makes resource management difficult because devices have different processing power, memory, and storage capacities. Resource management systems must account for heterogeneity to maximize resource use. A decade ago, Schwarzkopf et al. [157] stressed the significance of heterogeneity in cloud computing. Edge computing still has this gap, which is surprising. Looking ahead, society's ravenous desire for seamless connectivity is clear. **Satellite-Air-Ground Integrated Networks (SAGINs)** are gaining popularity as wireless network technologies progress beyond B5G/6G [158]. Space, aerial, and ground networks are harmoniously integrated with these revolutionary networks. SAGINs' ability to increase network coverage and throughput is their main draw. They provide uninterrupted connectivity in the most demanding environments without infrastructure by combining satellites, high-altitude platform stations, and UAVs. SAGINs can improve network coverage and speed by using non-terrestrial communication components. Satellites, high-altitude platform stations, and UAVs ensure device connectivity even in infrastructure-less environments.

6.6 Security and Privacy

Edge devices play a pivotal role in critical sectors such as healthcare, where they collect and process sensitive data. For instance, a mission-critical application in healthcare might handle patient health records. Such unauthorized access to this data can lead to grave privacy breaches. Given the paramount importance of security and privacy [3], it is no surprise that these concerns are capturing the attention of researchers. The demand for more research is particularly pronounced in the development of resource management systems. These systems need to incorporate access control mechanisms, encryption protocols, and data isolation techniques to safeguard data at the edge, all while being mindful of privacy regulations and compliance requirements. Furthermore, in the context of MEC, addressing security challenges is crucial. This encompasses concerns related to edge server security, data security, and network security. Notably, task offloading techniques have frequently faced hurdles in these areas.

6.7 Mobility

As discussed in Section 5.5, mobility is often considered a constraint in most cloud or edge infrastructures due to its impact on latency, connectivity, and service continuity, particularly in mission-critical applications such as emergency response, autonomous vehicles, and industrial automation. However, mobility should not merely be tolerated; it is a capability that must be actively supported and enhanced.

7 Conclusion

This survey sets itself apart from previous research in its scope and objectives. In this article, we presented a detailed review of the mission-critical application and also the time-sensitive application, which could be considered a specific type of mission-critical application in the edge-cloud, mobile, VEC, and even fog with IoT areas. A new age of real-world applications associated with IoT and vehicular systems has different needs compared with typical applications at the edge. We performed an SLR of these applications in a variety of concepts, such as computational offloading, resource allocation, service or application placement, and even replicas. We examined the merits and drawbacks of several systematic and peer-reviewed studies related to resource management or computational offloading in edge computing. We summarized the current research based on some different performance indicators in resource management in different environments, with a focus on the method and the type of application. Finally, we discussed various unresolved challenges and potential research directions. Through this study, our aim is to aid researchers in designing edge computing resource management schemes tailored to highly dynamic applications, such as mission-critical and time-sensitive applications, and effectively harnessing the full potential of edge computing.

References

[1] Milan Erdelj and Enrico Natalizio. 2016. UAV-assisted disaster management: Applications and open issues. In *2016 International Conference on Computing, Networking and Communications (ICNC)*. 1–5. DOI: http://dx.doi.org/10.1109/ICCNC.2016.7440563

[2] Ankur Utsav, Amit Abhishek, P. Suraj, and Ritesh Kr. Badhai. 2021. An IoT based UAV network for military applications. In *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. 122–125. DOI: http://dx.doi.org/10.1109/WiSPNET51692.2021.9419470

[3] Tzu-Wei Lin. 2022. A privacy-preserved id-based secure communication scheme in 5G-IoT telemedicine systems. *Sensors* 22, 18 (Jan. 2022), 6838. DOI: http://dx.doi.org/10.3390/s22186838

[4] Zeinab Zabihi, Amir Masoud Eftekhari Moghadam, and Mohammad Hossein Rezvani. 2023. Reinforcement learning methods for computing offloading: A systematic review. *Comput. Surveys* (June 2023), 3603703. DOI: http://dx.doi.org/10.1145/3603703

[5] Abednego Acheampong, Yiwen Zhang, Xiaolong Xu, and Daniel Appiah Kumah. 2023. A review of the current task offloading algorithms, strategies and approach in edge computing systems. *Computer Modeling in Engineering & Sciences* 134, 1 (2023), 35–88. DOI: http://dx.doi.org/10.32604/cmes.2022.021394

[6] Xiaojie Zhang and Saptarshi Debroy. 2023. Resource management in mobile edge computing: A comprehensive survey. *Comput. Surveys* 55, 13s (Dec. 2023), 1–37. DOI: http://dx.doi.org/10.1145/3589639

[7] Hamza Djigal, Jia Xu, Linfeng Liu, and Yan Zhang. 2022. Machine and deep learning for resource allocation in multi-access edge computing: A survey. *IEEE Communications Surveys & Tutorials* 24, 4 (2022), 2449–2494. DOI: http://dx.doi.org/10.1109/COMST.2022.3199544

[8] Ali Shakarami, Hamid Shakarami, Mostafa Ghobaei-Arani, Elaheh Nikougoftar, and Mohammad Faraji-Mehmandar. 2022. Resource provisioning in edge/fog computing: A comprehensive and systematic review. *Journal of Systems Architecture* 122 (Jan. 2022), 102362. DOI: http://dx.doi.org/10.1016/j.sysarc.2021.102362

[9] Hai Lin, Sherali Zeadally, Zhihong Chen, Houda Labiod, and Lusheng Wang. 2020. A survey on computation offloading modeling for edge computing. *Journal of Network and Computer Applications* 169 (Nov. 2020), 102781. DOI: http://dx.doi.org/10.1016/j.jnca.2020.102781

[10] Ali Shakarami, Mostafa Ghobaei-Arani, and Ali Shahidinejad. 2020. A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective. *Computer Networks* 182 (Dec. 2020), 107496. DOI: http://dx.doi.org/10.1016/j.comnet.2020.107496

[11] Thang Le Duc, Rafael García Leiva, Paolo Casari, and Per-Olov Östberg. 2019. Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. *Comput. Surveys* 52, 5 (Sept. 2019), 94:1–94:39. DOI: http://dx.doi.org/10.1145/3341145

[12] Shuchen Zhou, Waqas Jadoon, and Iftikhar Khan. 2023. Computing offloading strategy in mobile edge computing environment: A comparison between adopted frameworks, challenges, and future directions. *Electronics* 12 (May 2023), 2452. DOI: http://dx.doi.org/10.3390/electronics12112452

[13] Ramesh Singh, Radhika Sukapuram, and Suchetana Chakraborty. 2023. A survey of mobility-aware multi-access edge computing: Challenges, use cases and future directions. *Ad Hoc Networks* 140 (March 2023), 103044. DOI: http://dx.doi.org/10.1016/j.adhoc.2022.103044

[14] Ning Lyu. 2021. Brief review on computing resource allocation algorithms in mobile edge computing. In *2021 2nd International Conference on Computing and Data Science (CDS)*. 108–111. DOI: http://dx.doi.org/10.1109/CDS52072.2021.00025

[15] Congfeng Jiang, Xiaolan Cheng, Honghao Gao, Xin Zhou, and Jian Wan. 2019. Toward computation offloading in edge computing: A survey. *IEEE Access* 7 (2019), 131543–131558. DOI: http://dx.doi.org/10.1109/ACCESS.2019.2938660

[16] Balázs Sonkoly, János Czentye, Márk Szalay, Balázs Németh, and László Toka. 2021. Survey on placement methods in the edge and beyond. *IEEE Communications Surveys & Tutorials* 23, 4 (2021), 2590–2629. DOI: http://dx.doi.org/10.1109/COMST.2021.3101460

[17] Nina Santi and Nathalie Mitton. 2021. A resource management survey for mission critical and time critical applications in multi access edge computing. *ITU Journal on Future and Evolving Technologies* 2, 2 (Nov. 2021). Retrieved from https://hal.science/hal-03420193

[18] Mostafa Raeisi-Varzaneh, Omar Dakkak, Adib Habbal, and Byung-Seo Kim. 2023. Resource scheduling in edge computing: Architecture, taxonomy, open issues and future research directions. *IEEE Access* 11 (2023), 25329–25350. DOI: http://dx.doi.org/10.1109/ACCESS.2023.3256522

[19] Ling Hou, Mark A. Gregory, and Shuo Li. 2022. A survey of multi-access edge computing and vehicular networking. *IEEE Access* 10 (2022), 123436–123451. DOI: http://dx.doi.org/10.1109/ACCESS.2022.3224032

[20] Quyuan Luo, Shihong Hu, Changle Li, Guanghui Li, and Weisong Shi. 2021. Resource Scheduling in Edge Computing: A Survey. *IEEE Communications Surveys & Tutorials* 23, 4 (2021), 2131–2165. DOI: http://dx.doi.org/10.1109/COMST.2021.3106401

[21] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering—A systematic literature review. *Information and Software Technology* 51, 1 (Jan. 2009), 7–15. DOI: http://dx.doi.org/10.1016/j.infsof.2008.09.009

[22] Yu-Chen Lee, Fernanda Leite, and Katherine Lieberknecht. 2023. Prioritizing selection criteria of distributed circular water systems: A fuzzy based multi-criteria decision-making approach. *Journal of Cleaner Production* 417 (Sept. 2023), 138073. DOI: http://dx.doi.org/10.1016/j.jclepro.2023.138073

[23] Rodolfo Meneguette, Robson De Grande, Jo Ueyama, Geraldo P. Rocha Filho, and Edmundo Madeira. 2023. Vehicular Edge Computing: Architecture, Resource Management, Security, and Challenges. *Comput. Surveys* 55, 1 (Jan. 2023), 1–46. DOI: http://dx.doi.org/10.1145/3485129

[24] The European Parliament and The Council of the European Union. 2022. Directive of the European Parliament and of the Council on the resilience of critical entities and repealing Council Directive 2008/114/EC. (2022). Retrieved 30 November 2022 from https://data.consilium.europa.eu/doc/document/PE-51-2022-INIT/en/pdf

[25] Doru Calin. 2023. A short overview of mission critical communications technologies for the rail industry. *IEEE Wireless Communications* 30, 3 (June 2023), 6–8. DOI: http://dx.doi.org/10.1109/MWC.2023.10183806

[26] Moumita Mishra, Sayan Kumar Roy, Anwesha Mukherjee, Debashis De, Soumya K. Ghosh, and Rajkumar Buyya. 2020. An energy-aware multi-sensor geo-fog paradigm for mission critical applications. *Journal of Ambient Intelligence and Humanized Computing* 11, 8 (Aug. 2020), 3155–3173. DOI: http://dx.doi.org/10.1007/s12652-019-01481-1

[27] Arash Heidari, Nima Jafari Navimipour, Mehmet Unal, and Guodao Zhang. 2023. Machine learning applications in internet-of-drones: Systematic review, recent deployments, and open issues. *Comput. Surveys* 55, 12 (March 2023), 247:1–247:45. DOI: http://dx.doi.org/10.1145/3571728

[28] Walid Balid, Hasan Tafish, and Hazem H. Refai. 2018. Intelligent vehicle counting and classification sensor for real-time traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems* 19, 6 (June 2018), 1784–1794. DOI: http://dx.doi.org/10.1109/TITS.2017.2741507

[29] Yue Wang, Tao Yu, and Kei Sakaguchi. 2021. Context-based MEC platform for augmented-reality services in 5G networks. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. 1–5. DOI: http://dx.doi.org/10.1109/VTC2021-Fall52928.2021.9625304

[30] Mingzhe Chen, Walid Saad, and Changchuan Yin. 2018. Virtual reality over wireless networks: Quality-of-service model and learning-based resource management. *IEEE Transactions on Communications* 66, 11 (Nov. 2018), 5621–5635. DOI: http://dx.doi.org/10.1109/TCOMM.2018.2850303

[31] Sujie Shao, Lili Su, Qinghang Zhang, Shuang Wu, Shaoyong Guo, and Feng Qi. 2023. Multi task dynamic edge–end computing collaboration for urban Internet of Vehicles. *Computer Networks* 227 (May 2023), 109690. DOI: http://dx.doi.org/10.1016/j.comnet.2023.109690

[32] Clark W. Gellings. 2020. *The Smart Grid: Enabling Energy Efficiency and Demand Response* (1 ed.). River Publishers. DOI: http://dx.doi.org/10.1201/9781003151524

[33] Mostafa Haghi Kashani, Mona Madanipour, Mohammad Nikravan, Parvaneh Asghari, and Ebrahim Mahdipour. 2021. A systematic review of IoT in healthcare: Applications, techniques, and trends. *Journal of Network and Computer Applications* 192 (Oct. 2021), 103164. DOI: http://dx.doi.org/10.1016/j.jnca.2021.103164

[34] Richard Sather Iii, Mahsa Soufineyestani, Nabiha Imtiaz, and Arshia Khan. 2021. Assistive robots designed for eldery care and caregivers. *International Journal of Robotics and Control* 3, 1 (May 2021), 1. DOI: http://dx.doi.org/10.5430/ijrc.v3n1p1

[35] Mohammad Hammoudeh, Ibrahim Ghafir, Ahcène Bounceur, and Thomas Rawlinson. 2019. Continuous monitoring in mission-critical applications using the internet of things and blockchain. In *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*. ACM, Paris France, 1–5. DOI: http://dx.doi.org/10.1145/3341325.3342018

[36] Bin Cheng, Gürkan Solmaz, Flavio Cirillo, Ernö Kovacs, Kazuyuki Terasawa, and Atsushi Kitazawa. 2018. FogFlow: Easy programming of IoT services over cloud and edges for smart cities. *IEEE Internet of Things Journal* 5, 2 (April 2018), 696–707. DOI: http://dx.doi.org/10.1109/JIOT.2017.2747214

[37] Zaiwar Ali, Sadia Khaf, Ziaul Haq Abbas, Ghulam Abbas, Fazal Muhammad, and Sunghwan Kim. 2020. A deep learning approach for mobility-aware and energy-efficient resource allocation in MEC. *IEEE Access* 8 (2020), 179530–179546. DOI: http://dx.doi.org/10.1109/ACCESS.2020.3028240

[38] Zhaohui Yang, Cunhua Pan, Jiancao Hou, and Mohammad Shikh-Bahaei. 2019. Efficient resource allocation for mobile-edge computing networks with NOMA: Completion time and energy minimization. *IEEE Transactions on Communications* 67, 11 (Nov. 2019), 7771–7784. DOI: http://dx.doi.org/10.1109/TCOMM.2019.2935717

[39] Jiangjiang Zhang, Bei Gong, Muhammad Waqas, Shanshan Tu, and Zhu Han. 2023. A hybrid many-objective optimization algorithm for task offloading and resource allocation in multi-server mobile edge computing networks. *IEEE Transactions on Services Computing* (2023), 1–14. DOI: http://dx.doi.org/10.1109/TSC.2023.3268990

[40] Abdullah Lakhan, Mazin Abed Mohammed, Ahmed N. Rashid, Seifedine Kadry, and Karrar Hameed Abdulkareem. 2022. Deadline aware and energy-efficient scheduling algorithm for fine-grained tasks in mobile edge computing. *International Journal of Web and Grid Services* 18, 2 (Jan. 2022), 168–193. DOI: http://dx.doi.org/10.1504/IJWGS.2022.121935

[41] Wanli Wen, Ying Cui, Tony Q. S. Quek, Fu-Chun Zheng, and Shi Jin. 2020. Joint optimal software caching, computation offloading and communications resource allocation for mobile edge computing. *IEEE Transactions on Vehicular Technology* 69, 7 (July 2020), 7879–7894. DOI: http://dx.doi.org/10.1109/TVT.2020.2993359

[42] Sardar Khaliq uz Zaman, Ali Imran Jehangiri, Tahir Maqsood, Nuhman ul Haq, Arif Iqbal Umar, Junaid Shuja, Zulfiqar Ahmad, Imed Ben Dhaou, and Mohammed F. Alsharekh. 2023. LiMPO: Lightweight mobility prediction and offloading framework using machine learning for mobile edge computing. *Cluster Computing* 26, 1 (Feb. 2023), 99–117. DOI: http://dx.doi.org/10.1007/s10586-021-03518-7

[43] Faten A. Saif, Rohaya Latip, Zurina Mohd Hanapi, Mohamed A. Alrshah, and Shafinah Kamarudin. 2023. Workload allocation toward energy consumption-delay trade-off in cloud-fog computing using multi-objective NPSO algorithm. *IEEE Access* 11 (2023), 45393–45404. DOI: http://dx.doi.org/10.1109/ACCESS.2023.3266822

[44] Sanaa Almasri, Moath Jarrah, and Basheer Al-Duwairi. 2022. Multi-objective optimization of task assignment in distributed mobile edge computing. *Journal of Reliable Intelligent Environments* 8, 1 (March 2022), 21–33. DOI: http://dx.doi.org/10.1007/s40860-021-00162-1

[45] Zhiwei Qin, Juan Li, Wei Liu, and Xiao Yu. 2022. Mobility-aware and energy-efficient task offloading strategy for mobile edge workflows. *Wuhan University Journal of Natural Sciences* 27, 6 (Dec. 2022), 476–488. DOI: http://dx.doi.org/10.1051/wujns/2022276476

[46] Mohammad Jalilvand Aghdam Bonab and Ramin Shaghaghi Kandovan. 2022. QoS-aware resource allocation in mobile edge computing networks: Using intelligent offloading and caching strategy. *Peer-to-Peer Networking and Applications* 15, 3 (May 2022), 1328–1344. DOI: http://dx.doi.org/10.1007/s12083-021-01271-7

[47] Tayebeh Bahreini, Hossein Badri, and Daniel Grosu. 2019. Energy-aware capacity provisioning and resource allocation in edge computing systems. In *Edge Computing – EDGE 2019 (Lecture Notes in Computer Science)*, Tao Zhang, Jinpeng Wei, and Liang-Jie Zhang (Eds.). Springer International Publishing, Cham, 31–45. DOI: http://dx.doi.org/10.1007/978-3-030-23374-7_3

[48] Luyue Ji and Songtao Guo. 2019. Energy-efficient cooperative resource allocation in wireless powered mobile edge computing. *IEEE Internet of Things Journal* 6, 3 (June 2019), 4744–4754. DOI: http://dx.doi.org/10.1109/JIOT.2018.2880812

[49] Wei Jiang, Xiong Pan, Ke Jiang, Liang Wen, and Qi Dong. 2019. Energy-aware design of stochastic applications with statistical deadline and reliability guarantees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 38, 8 (Aug. 2019), 1413–1426. DOI: http://dx.doi.org/10.1109/TCAD.2018.2846652

[50] Chit Wutyee Zaw, Shashi Raj Pandey, Kitae Kim, and Choong Seon Hong. 2021. Energy-aware resource management for federated learning in multi-access edge computing systems. *IEEE Access* 9 (2021), 34938–34950. DOI: http://dx.doi.org/10.1109/ACCESS.2021.3055523

[51] Chunlin Li, Yong Zhang, Xiang Gao, and Youlong Luo. 2022. Energy-latency tradeoffs for edge caching and dynamic service migration based on DQN in mobile edge computing. *J. Parallel and Distrib. Comput.* 166 (Aug. 2022), 15–31. DOI: http://dx.doi.org/10.1016/j.jpdc.2022.03.001

[52] Ziaul Haq Abbas, Zaiwar Ali, Ghulam Abbas, Lei Jiao, Muhammad Bilal, Doug-Young Suh, and Md Jalil Piran. 2021. Computational offloading in mobile edge with comprehensive and energy efficient cost function: A deep learning approach. *Sensors* 21, 10 (Jan. 2021), 3523. DOI: http://dx.doi.org/10.3390/s21103523

[53] Gayatri Sakya and Vidushi Sharma. 2019. ADMC-MAC: Energy efficient adaptive MAC protocol for mission critical applications in WSN. *Sustainable Computing: Informatics and Systems* 23 (Sept. 2019), 21–28. DOI: http://dx.doi.org/10.1016/j.suscom.2019.05.001

[54] Eyhab Al-Masri, Alireza Souri, Habiba Mohamed, Wenjun Yang, James Olmsted, and Olivera Kotevska. 2023. Energy-efficient cooperative resource allocation and task scheduling for Internet of Things environments. *Internet of Things* 23 (Oct. 2023), 100832. DOI: http://dx.doi.org/10.1016/j.iot.2023.100832

[55] Haolin Liu, Le Cao, Tingrui Pei, Qingyong Deng, and Jiang Zhu. 2020. A fast algorithm for energy-saving offloading with reliability and latency requirements in multi-access edge computing. *IEEE Access* 8 (2020), 151–161. DOI: http://dx.doi.org/10.1109/ACCESS.2019.2961453

[56] Tianxi Ji, Changqing Luo, Lixing Yu, Qianlong Wang, Siheng Chen, Arun Thapa, and Pan Li. 2022. Energy-efficient computation offloading in mobile edge computing systems with uncertainties. *IEEE Transactions on Wireless Communications* 21, 8 (Aug. 2022), 5717–5729. DOI: http://dx.doi.org/10.1109/TWC.2022.3142685

[57] Xihan Chen, Yunlong Cai, Liyan Li, Minjian Zhao, Benoit Champagne, and Lajos Hanzo. 2020. Energy-efficient resource allocation for latency-sensitive mobile edge computing. *IEEE Transactions on Vehicular Technology* 69, 2 (Feb. 2020), 2246–2262. DOI: http://dx.doi.org/10.1109/TVT.2019.2962542

[58] Jing Liu, Pei Yang, and Cen Chen. 2023. Intelligent energy-efficient scheduling with ant colony techniques for heterogeneous edge computing. *J. Parallel and Distrib. Comput.* 172 (Feb. 2023), 84–96. DOI: http://dx.doi.org/10.1016/j.jpdc.2022.10.003

[59] Lei Feng, Yu Zhou, Tang Liu, Xiaoyu Que, Peng Yu, Tao Hong, and Xuesong Qiu. 2021. Energy-efficient offloading for mission-critical IoT services using EVT-embedded intelligent learning. *IEEE Transactions on Green Communications and Networking* 5, 3 (Sept. 2021), 1179–1190. DOI: http://dx.doi.org/10.1109/TGCN.2021.3072140

[60] Mostafa Ghobaei-Arani and Ali Shahidinejad. 2022. A cost-efficient IoT service placement approach using whale optimization algorithm in fog computing environment. *Expert Systems with Applications* 200 (Aug. 2022), 117012. DOI: http://dx.doi.org/10.1016/j.eswa.2022.117012

[61] Minyu Cui, Angeliki Kritikakou, Lei Mo, and Emmanuel Casseau. 2023. Near-optimal energy-efficient partial-duplication task mapping of real-time parallel applications. *Journal of Systems Architecture* 134 (Jan. 2023), 102790. DOI: http://dx.doi.org/10.1016/j.sysarc.2022.102790

[62] Yumei Li, Xiumin Zhu, Shudian Song, Shuyue Ma, Feng Yang, and Linbo Zhai. 2023. Task offloading and parameters optimization of MAR in multi-access edge computing. *Expert Systems with Applications* 215 (April 2023), 119379. DOI: http://dx.doi.org/10.1016/j.eswa.2022.119379

[63] Xintong Qin, Zhengyu Song, Tianwei Hou, Wenjuan Yu, Jun Wang, and Xin Sun. 2023. Joint optimization of resource allocation, phase shift and UAV trajectory for energy-efficient RIS-assisted UAV-enabled MEC systems. *IEEE Transactions on Green Communications and Networking* (2023).

[64] Hengxi Zhang, Chengyue Lu, Huaze Tang, Xiaoli Wei, Le Liang, Ling Cheng, Wenbo Ding, and Zhu Han. 2023. Mean-field-aided multiagent reinforcement learning for resource allocation in vehicular networks. *IEEE Internet of Things Journal* 10, 3 (Feb. 2023), 2667–2679. DOI: http://dx.doi.org/10.1109/JIOT.2022.3214525

[65] Chao Yang, Xin Chen, Yi Liu, Weifeng Zhong, and Shengli Xie. 2019. Efficient task offloading and resource allocation for edge computing-based smart grid networks. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 1–6. DOI: http://dx.doi.org/10.1109/ICC.2019.8761535

[66] Shao Sujie, L. I. Yi, G. U. O. Shaoyong, Wang Chenhui, Chen Xingyu, and Q. I. U. Xuesong. 2023. Delay and energy consumption oriented UAV inspection business collaboration computing mechanism in edge computing based electric power IoT. *Chinese Journal of Electronics* 32, 1 (Jan. 2023), 13–25. DOI: http://dx.doi.org/10.23919/cje.2021.00.312

[67] Mekala Ratna Raju and Sai Krishna Mothku. 2023. Delay and energy aware task scheduling mechanism for fog-enabled IoT applications: A reinforcement learning approach. *Computer Networks* 224 (April 2023), 109603. DOI: http://dx.doi.org/10.1016/j.comnet.2023.109603

[68] Lin Wang, Lei Jiao, Jun Li, Julien Gedeon, and Max Muhlhauser. 2019. MOERA: Mobility-agnostic online resource allocation for edge computing. *IEEE Transactions on Mobile Computing* 18, 8 (Aug. 2019), 1843–1856. DOI: http://dx.doi.org/10.1109/TMC.2018.2867520

[69] Yu Liu, Yingling Mao, Zhenhua Liu, and Yuanyuan Yang. 2023. Deep learning-assisted online task offloading for latency minimization in heterogeneous mobile edge. *IEEE Transactions on Mobile Computing* (2023), 1–14. DOI: http://dx.doi.org/10.1109/TMC.2023.3285882

[70] Zhihan Lv and Liang Qiao. 2020. Optimization of collaborative resource allocation for mobile edge computing. *Computer Communications* 161 (Sept. 2020), 19–27. DOI: http://dx.doi.org/10.1016/j.comcom.2020.07.022

[71] Zichuan Xu, Zhiheng Zhang, Weifa Liang, Qiufen Xia, Omer Rana, and Guowei Wu. 2020. QoS-aware VNF placement and service chaining for IoT applications in multi-tier mobile edge networks. *ACM Transactions on Sensor Networks* 16, 3 (Aug. 2020), 1–27. DOI: http://dx.doi.org/10.1145/3387705

[72] Shangguang Wang, Yan Guo, Ning Zhang, Peng Yang, Ao Zhou, and Xuemin Shen. 2021. Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach. *IEEE Transactions on Mobile Computing* 20, 3 (March 2021), 939–951. DOI: http://dx.doi.org/10.1109/TMC.2019.2957804

[73] Andreas Kouloumpris, Maria K. Michael, and Theocharis Theocharides. 2019. Reliability-aware task allocation latency optimization in edge computing. In *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. 200–203. DOI: http://dx.doi.org/10.1109/IOLTS.2019.8854422

[74] Md Delowar Hossain, Luan N. T. Huynh, Tangina Sultana, Tri D.T. Nguyen, Jae Ho Park, Choong Seon Hong, and Eui-Nam Huh. 2020. Collaborative task offloading for overloaded mobile edge computing in small-cell networks. In *2020 International Conference on Information Networking (ICOIN)*. 717–722. DOI: http://dx.doi.org/10.1109/ICOIN48656.2020.9016452

[75] Shuai Yu, Rami Langar, Xiaoming Fu, Li Wang, and Zhu Han. 2018. Computation offloading with data caching enhancement for mobile edge computing. *IEEE Transactions on Vehicular Technology* 67, 11 (Nov. 2018), 11098–11112. DOI: http://dx.doi.org/10.1109/TVT.2018.2869144

[76] Arpit Jain and Dharm Singh Jat. 2020. An edge computing paradigm for time-sensitive applications. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. 798–803. DOI: http://dx.doi.org/10.1109/WorldS450073.2020.9210325

[77] Raphael Eidenbenz, Yvonne-Anne Pignolet, and Alain Ryser. 2020. Latency-aware industrial fog application orchestration with kubernetes. In *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*. 164–171. DOI: http://dx.doi.org/10.1109/FMEC49853.2020.9144934

[78] Juan Luis Herrera, Jaime Galán-Jiménez, Luca Foschini, Paolo Bellavista, Javier Berrocal, and Juan M. Murillo. 2022. QoS-aware fog node placement for intensive IoT applications in SDN-fog scenarios. *IEEE Internet of Things Journal* 9, 15 (Aug. 2022), 13725–13739. DOI: http://dx.doi.org/10.1109/JIOT.2022.3143948

[79] Zi Wang, Zhiwei Zhao, Geyong Min, Xinyuan Huang, Qiang Ni, and Rong Wang. 2018. User mobility aware task assignment for mobile edge computing. *Future Generation Computer Systems* 85 (Aug. 2018), 1–8. DOI: http://dx.doi.org/10.1016/j.future.2018.02.014

[80] Ismail Alqerm and Jianli Pan. 2021. DeepEdge: A new QoE-based resource allocation framework using deep reinforcement learning for future heterogeneous edge-IoT applications. *IEEE Transactions on Network and Service Management* 18, 4 (Dec. 2021), 3942–3954. DOI: http://dx.doi.org/10.1109/TNSM.2021.3123959

[81] Jingxian Liu, Yitian Wang, Wei Zhang, and Kangxu Tian. 2023. A novel offloading and resource allocation scheme for time-critical tasks in heterogeneous internet of vehicles. In *2023 2nd International Conference for Innovation in Technology (INOCON)*. 1–7. DOI: http://dx.doi.org/10.1109/INOCON57975.2023.10101035

[82] Abdallah Moubayed, Abdallah Shami, Parisa Heidari, Adel Larabi, and Richard Brunner. 2021. Edge-enabled V2X service placement for intelligent transportation systems. *IEEE Transactions on Mobile Computing* 20, 4 (April 2021), 1380–1392. DOI: http://dx.doi.org/10.1109/TMC.2020.2965929

[83] László Toka, Dávid Haja, Attila Kőrösi, and Balázs Sonkoly. 2019. Resource provisioning for highly reliable and ultra-responsive edge applications. In *2019 IEEE 8th International Conference on Cloud Networking (CloudNet)*. 1–6. DOI: http://dx.doi.org/10.1109/CloudNet47604.2019.9064131

[84] Tianyu Yang, Yulin Hu, M. Cenk Gursoy, Anke Schmeink, and Rudolf Mathar. 2018. Deep reinforcement learning based resource allocation in low latency edge computing networks. In *2018 15th international symposium on wireless communication systems (ISWCS)*. IEEE, 1–5.

[85] Vitaly Petrov, Maria A. Lema, Margarita Gapeyenko, Konstantinos Antonakoglou, Dmitri Moltchanov, Fragkiskos Sardis, Andrey Samuylov, Sergey Andreev, Yevgeni Koucheryavy, and Mischa Dohler. 2018. Achieving end-to-end reliability of mission-critical traffic in softwarized 5G networks. *IEEE Journal on Selected Areas in Communications* 36, 3 (March 2018), 485–501. DOI: http://dx.doi.org/10.1109/JSAC.2018.2815419

[86] Zhiqi Chen, Sheng Zhang, Can Wang, Zhuzhong Qian, Mingjun Xiao, Jie Wu, and Imad Jawhar. 2018. A novel algorithm for NFV chain placement in edge computing environments. In *2018 IEEE Global Communications Conference (GLOBECOM)*. 1–6. DOI: http://dx.doi.org/10.1109/GLOCOM.2018.8647371

[87] Sa Math, Prohim Tam, Dae-Young Kim, and Seokhoon Kim. 2022. Intelligent offloading decision and resource allocations schemes based on RNN/DQN for reliability assurance in software-defined massive machine-type communications. *Security and Communication Networks* 2022 (April 2022), e4289216. DOI: http://dx.doi.org/10.1155/2022/4289216

[88] Zhilin Liu, Yao Zhu, Yulin Hu, Peng Sun, and Anke Schmeink. 2022. Reliability-oriented design framework in NOMA-assisted mobile edge computing. *IEEE Access* 10 (2022), 103598–103609. DOI: http://dx.doi.org/10.1109/ACCESS.2022.3210510

[89] Lu Zhao, Bo Li, Wenan Tan, Guangming Cui, Qiang He, Xiaolong Xu, Lida Xu, and Yun Yang. 2022. Joint coverage-reliability for budgeted edge application deployment in mobile edge computing environment. *IEEE Transactions on Parallel and Distributed Systems* 33, 12 (Dec. 2022), 3760–3771. DOI: http://dx.doi.org/10.1109/TPDS.2022.3166163

[90] Jianhui Liu and Qi Zhang. 2019. Code-Partitioning Offloading Schemes in Mobile Edge Computing for Augmented Reality. *IEEE Access* 7 (2019), 11222–11236. DOI: http://dx.doi.org/10.1109/ACCESS.2019.2891113

[91] Eugen Šlapak, Juraj Gazda, Weiqiang Guo, Taras Maksymyuk, and Mischa Dohler. 2021. Cost-effective resource allocation for multitier mobile edge computing in 5G mobile networks. *IEEE Access* 9 (2021), 28658–28672. DOI: http://dx.doi.org/10.1109/ACCESS.2021.3059029

[92] Jialei Liu, Ao Zhou, Chunhong Liu, Tongguang Zhang, Lianyong Qi, Shangguang Wang, and Rajkumar Buyya. 2022. Reliability-enhanced task offloading in mobile edge computing environments. *IEEE Internet of Things Journal* 9, 13 (July 2022), 10382–10396. DOI: http://dx.doi.org/10.1109/JIOT.2021.3115807

[93] Chanh Nguyen, Cristian Klein, and Erik Elmroth. 2020. Elasticity control for latency-intolerant mobile edge applications. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 70–83.

[94] Shudian Song, Shuyue Ma, Jingmei Zhao, Feng Yang, and Linbo Zhai. 2022. Cost-efficient multi-service task offloading scheduling for mobile edge computing. *Applied Intelligence* 52, 4 (March 2022), 4028–4040. DOI: http://dx.doi.org/10.1007/s10489-021-02549-2

[95] Xiao Ma, Shangguang Wang, Shan Zhang, Peng Yang, Chuang Lin, and Xuemin Shen. 2021. Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing. *IEEE Transactions on Cloud Computing* 9, 3 (July 2021), 968–980. DOI: http://dx.doi.org/10.1109/TCC.2019.2903240

[96] Qiuping Li, Junhui Zhao, and Yi Gong. 2019. Computation offloading and resource allocation for mobile edge computing with multiple access points. *IET Communications* 13, 17 (2019), 2668–2677.

[97] Xianfu Chen, Honggang Zhang, Celimuge Wu, Shiwen Mao, Yusheng Ji, and Mehdi Bennis. 2018. Performance optimization in mobile-edge computing via deep reinforcement learning. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. 1–6. DOI: http://dx.doi.org/10.1109/VTCFall.2018.8690980

[98] Ran Bi, Ting Peng, Jiankang Ren, Xiaolin Fang, and Guozhen Tan. 2022. Joint service placement and computation scheduling in edge clouds. In *2022 IEEE International Conference on Web Services (ICWS)*. IEEE, 47–56.

[99] Hongbo Jiang, Xingxia Dai, Zhu Xiao, and Arun Iyengar. 2023. Joint task offloading and resource allocation for energy-constrained mobile edge computing. *IEEE Transactions on Mobile Computing* 22, 7 (July 2023), 4000–4015. DOI: http://dx.doi.org/10.1109/TMC.2022.3150432

[100] Xiaolan Liu, Zhijin Qin, and Yue Gao. 2019. Resource allocation for edge computing in IoT networks via reinforcement learning. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 1–6. DOI: http://dx.doi.org/10.1109/ICC.2019.8761385

[101] Quyuan Wang, Songtao Guo, Jiadi Liu, and Yuanyuan Yang. 2019. Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing. *Sustainable Computing: Informatics and Systems* 21 (March 2019), 154–164. DOI: http://dx.doi.org/10.1016/j.suscom.2019.01.007

[102] Yanling Shao, Chunlin Li, Zhao Fu, Leyue Jia, and Youlong Luo. 2019. Cost-effective replication management and scheduling in edge computing. *Journal of Network and Computer Applications* 129 (March 2019), 46–61. DOI: http://dx.doi.org/10.1016/j.jnca.2019.01.001

[103] Wei Wang, Yongmin Zhang, Rui Huang, Ju Ren, Feng Lyu, and Yaoxue Zhang. 2023. Efficient resource management and expansion scheme for collaborative edge-cloud computing. *IEEE Transactions on Mobile Computing* (2023), 1–15. DOI: http://dx.doi.org/10.1109/TMC.2023.3267497

[104] Yaru Fu, Xiaolong Yang, Peng Yang, Angus K. Y. Wong, Zheng Shi, Hong Wang, and Tony Q. S. Quek. 2021. Energy-efficient offloading and resource allocation for mobile edge computing enabled mission-critical internet-of-things systems. *EURASIP Journal on Wireless Communications and Networking* 2021, 1 (Feb. 2021), 26. DOI: http://dx.doi.org/10.1186/s13638-021-01905-7

[105] Vajiheh Farhadi, Fidan Mehmeti, Ting He, Tom La Porta, Hana Khamfroush, Shiqiang Wang, and Kevin S. Chan. 2019. Service placement and request scheduling for data-intensive applications in edge clouds. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 1279–1287. DOI: http://dx.doi.org/10.1109/INFOCOM.2019.8737368

[106] Jiong Lou, Zhiqing Tang, Songli Zhang, Weijia Jia, Wei Zhao, and Jie Li. 2022. Cost-effective scheduling for dependent tasks with tight deadline constraints in mobile edge computing. *IEEE Transactions on Mobile Computing* (2022), 1–17. DOI: http://dx.doi.org/10.1109/TMC.2022.3188770

[107] Amardeep Mehta, Ewnetu Bayuh Lakew, Johan Tordsson, and Erik Elmroth. 2018. Utility-based allocation of industrial IoT applications in mobile edge clouds. In *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 1–10.

[108] Salman Raza, Wei Liu, Manzoor Ahmed, Muhammad Rizwan Anwar, Muhammad Ayzed Mirza, Qibo Sun, and Shangguang Wang. 2020. An efficient task offloading scheme in vehicular edge computing. *Journal of Cloud Computing* 9, 1 (June 2020), 28. DOI: http://dx.doi.org/10.1186/s13677-020-00175-w

[109] Jianbo Du, Yan Sun, Ning Zhang, Zehui Xiong, Aijing Sun, and Zhiguo Ding. 2023. Cost-effective task offloading in NOMA-enabled vehicular mobile edge computing. *IEEE Systems Journal* 17, 1 (March 2023), 928–939. DOI: http://dx.doi.org/10.1109/JSYST.2022.3167901

[110] Lingling Wang, Xiumin Zhu, Nianxin Lit, Yumei Liv, Shuyue Ma, and Linbo Zhai. 2022. Dynamic vehicle aware task offloading based on reinforcement learning in a vehicular edge computing network. In *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*. 263–270. DOI: http://dx.doi.org/10.1109/MSN57253.2022.00053

[111] Shanchen Pang, Nuanlai Wang, Min Wang, Sibo Qiao, Xue Zhai, and Neal N. Xiong. 2021. A smart network resource management system for high mobility edge computing in 5G internet of vehicles. *IEEE Transactions on Network Science and Engineering* 8, 4 (2021), 3179–3191.

[112] Yifan Zhang, Xiaoqi Qin, and Xianxin Song. 2020. Mobility-aware cooperative task offloading and resource allocation in vehicular edge computing. In *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 1–6.

[113] Carla Mouradian, Sami Yangui, and Roch H. Glitho. 2018. Robots as-a-service in cloud computing: Search and rescue in large-scale disasters case study. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. 1–7. DOI: http://dx.doi.org/10.1109/CCNC.2018.8319200

[114] Wenhan Zhan, Chunbo Luo, Geyong Min, Chao Wang, Qingxin Zhu, and Hancong Duan. 2020. Mobility-aware multi-user offloading optimization for mobile edge computing. *IEEE Transactions on Vehicular Technology* 69, 3 (March 2020), 3341–3356. DOI: http://dx.doi.org/10.1109/TVT.2020.2966500

[115] Rickson S. Pereira, Douglas D Lieira, Marco AC da Silva, Adinovam H. M. Pimenta, Joahannes BD da Costa, Denis Rosário, Leandro Villas, and Rodolfo I. Meneguette. 2020. RELIABLE: Resource allocation mechanism for 5G network using mobile edge computing. *Sensors* 20, 19 (2020), 5449.

[116] Xenofon Vasilakos, Walter Featherstone, Navdeep Uniyal, Anderson Bravalheri, Abubakar Siddique Muqaddas, Navid Solhjoo, Daniel Warren, Shadi Moazzeni, Reza Nejabati, and Dimitra Simeonidou. 2020. Towards zero downtime edge application mobility for ultra-low latency 5G streaming. In *2020 IEEE Cloud Summit*. 25–32. DOI: http://dx.doi.org/10.1109/IEEECloudSummit48914.2020.00010

[117] Daniel Chaves Temp, Paulo Silas Severo De Souza, Arthur Francisco Lorenzon, Marcelo Caggiani Luizelli, and Fábio Diniz Rossi. 2023. Mobility-aware registry migration for containerized applications on edge computing infrastructures. *Journal of Network and Computer Applications* 217 (Aug. 2023), 103676. DOI: http://dx.doi.org/10.1016/j.jnca.2023.103676

[118] Chunlin Li, Mingyang Song, Min Zhang, and Youlong Luo. 2020. Effective replica management for improving reliability and availability in edge-cloud computing environment. *J. Parallel and Distrib. Comput.* 143 (Sept. 2020), 107–128. DOI: http://dx.doi.org/10.1016/j.jpdc.2020.04.012

[119] Chunlin Li, Jingpan Bai, Yi Chen, and Youlong Luo. 2020. Resource and replica management strategy for optimizing financial cost and user experience in edge cloud computing system. *Information Sciences* 516 (April 2020), 33–55. DOI: http://dx.doi.org/10.1016/j.ins.2019.12.049

[120] Marco Marcozzi, Orhan Gemikonakli, Eser Gemikonakli, Enver Ever, and Leonardo Mostarda. 2023. Availability Evaluation of IoT Systems with Byzantine Fault-Tolerance for Mission-critical Applications. (June 2023). arXiv:2305.09262. Retrieved from https://arxiv.org/abs/2305.09262

[121] Francescomaria Faticanti, Francesco De Pellegrini, Domenico Siracusa, Daniele Santoro, and Silvio Cretti. 2019. Cutting throughput with the edge: App-aware placement in fog computing. In *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. 196–203. DOI: http://dx.doi.org/10.1109/CSCloud/EdgeCom.2019.00026

[122] Chunlin Li, YaPing Wang, Hengliang Tang, Yujiao Zhang, Yan Xin, and Youlong Luo. 2019. Flexible replica placement for enhancing the availability in edge computing environment. *Computer Communications* 146 (Oct. 2019), 1–14. DOI: http://dx.doi.org/10.1016/j.comcom.2019.07.013

[123] Vitor Barbosa Souza, Moises Henrique Pereira, Levi H. S. Lelis, and Xavi Masip-Bruin. 2020. Enhancing resource availability in vehicular fog computing through smart inter-domain handover. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. 1–6. DOI: http://dx.doi.org/10.1109/GLOBECOM42002.2020.9322238

[124] Tingyan Long, Yong Ma, Yunni Xia, Xuan Xiao, Qinglan Peng, and Jiale Zhao. 2022. A mobility-aware and fault-tolerant service offloading method in mobile edge computing. In *2022 IEEE International Conference on Web Services (ICWS)*. 67–72. DOI: http://dx.doi.org/10.1109/ICWS55610.2022.00024

[125] Saravanan Velrajan and V. Ceronmani Sharmila. 2021. QoS Management in Multi-access Edge Compute. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. 109–115. DOI: http://dx.doi.org/10.1109/ICCMC51019.2021.9418012

[126] Muhammad Mudassar, Yanlong Zhai, and Liao Lejian. 2022. Adaptive fault-tolerant strategy for latency-aware IoT application executing in edge computing environment. *IEEE Internet of Things Journal* 9, 15 (Aug. 2022), 13250–13262. DOI: http://dx.doi.org/10.1109/JIOT.2022.3144026

[127] Muhammad Mudassar, Yanlong Zhai, Lejian Liao, and Jun Shen. 2020. A decentralized latency-aware task allocation and group formation approach with fault tolerance for IoT applications. *IEEE Access* 8 (2020), 49212–49223. DOI: http://dx.doi.org/10.1109/ACCESS.2020.2979939

[128] Roozbeh Siyadatzadeh, Fatemeh Mehrafrooz, Mohsen Ansari, Bardia Safaei, Muhammad Shafique, Jörg Henkel, and Alireza Ejlali. 2023. ReLIEF: A reinforcement-learning-based real-time task assignment strategy in emerging fault-tolerant fog computing. *IEEE Internet of Things Journal* 10, 12 (June 2023), 10752–10763. DOI: http://dx.doi.org/10.1109/JIOT.2023.3240007

[129] Kun Wang, Yun Shao, Lei Xie, Jie Wu, and Song Guo. 2020. Adaptive and fault-tolerant data processing in healthcare IoT based on fog computing. *IEEE Transactions on Network Science and Engineering* 7, 1 (Jan. 2020), 263–273. DOI: http://dx.doi.org/10.1109/TNSE.2018.2859307

[130] Kaniyambakkam Elayavilli Srinivasa Desikan, Vijeth J. Kotagi, and C. Siva Ram Murthy. 2023. Decoding the interplay between latency, reliability, cost, and energy while provisioning resources in fog-computing-enabled IoT networks. *IEEE Internet of Things Journal* 10, 3 (Feb. 2023), 2404–2416. DOI: http://dx.doi.org/10.1109/JIOT.2022.3211872

[131] Ahmad Sharif, Mohsen Nickray, and Ali Shahidinejad. 2020. Fault-tolerant with load balancing scheduling in a fog-based IoT application. *IET Communications* 14, 16 (2020), 2646–2657. DOI: http://dx.doi.org/10.1049/iet-com.2020.0080

[132] Taha Alfakih, Mohammad Mehedi Hassan, and Muna Al-Razgan. 2021. Multi-objective accelerated particle swarm optimization with dynamic programing technique for resource allocation in mobile edge computing. *IEEE Access* 9 (2021), 167503–167520. DOI: http://dx.doi.org/10.1109/ACCESS.2021.3134941

[133] Changsen Xia, Zilong Jin, Jian Su, and Baozhu Li. 2023. Mobility-aware offloading and resource allocation strategies in MEC network based on game theory. *Wireless Communications and Mobile Computing* 2023 (Feb. 2023), 1–12. DOI: http://dx.doi.org/10.1155/2023/5216943

[134] Xiaolan Liu, Jiadong Yu, Jian Wang, and Yue Gao. 2020. Resource allocation with edge computing in iot networks via machine learning. *IEEE Internet of Things Journal* 7, 4 (April 2020), 3415–3426. DOI: http://dx.doi.org/10.1109/JIOT.2020.2970110

[135] Shinu M. Rajagopal, M. Supriya, and Rajkumar Buyya. 2023. Resource provisioning using meta-heuristic methods for IoT microservices with mobility management. *IEEE Access* 11 (2023), 60915–60938. DOI: http://dx.doi.org/10.1109/ACCESS.2023.3281348

[136] Mahbuba Afrin, Jiong Jin, Ashfaqur Rahman, Yu-Chu Tian, and Ambarish Kulkarni. 2019. Multi-objective resource allocation for edge cloud based robotic workflow in smart factory. *Future Generation Computer Systems* 97 (Aug. 2019), 119–130. DOI: http://dx.doi.org/10.1016/j.future.2019.02.062

[137] Abdullah Lakhan, Muhammad Suleman Memon, Qurat-ul-ain Mastoi, Mohamed Elhoseny, Mazin Abed Mohammed, Mumtaz Qabulio, and Mohamed Abdel-Basset. 2022. Cost-efficient mobility offloading and task scheduling for microservices IoVT applications in container-based fog cloud network. *Cluster Computing* 25, 3 (June 2022), 2061–2083. DOI: http://dx.doi.org/10.1007/s10586-021-03333-0

[138] Adyson M. Maia, Yacine Ghamri-Doudane, Dario Vieira, and Miguel F. de Castro. 2019. A multi-objective service placement and load distribution in edge computing. In *2019 IEEE Global Communications Conference (GLOBECOM)*. 1–7. DOI: http://dx.doi.org/10.1109/GLOBECOM38437.2019.9014303

[139] István Pelle, Márk Szalay, János Czentye, Balázs Sonkoly, and László Toka. 2022. Cost and Latency Optimized Edge Computing Platform. *Electronics* 11, 4 (Jan. 2022), 561. DOI: http://dx.doi.org/10.3390/electronics11040561

[140] Jingjing Yao and Nirwan Ansari. 2018. Reliability-aware fog resource provisioning for deadline-driven IoT services. In *2018 IEEE Global Communications Conference (GLOBECOM)*. 1–6. DOI: http://dx.doi.org/10.1109/GLOCOM.2018.8647378

[141] Jaber Taghizadeh, Mostafa Ghobaei-Arani, and Ali Shahidinejad. 2022. A metaheuristic-based data replica placement approach for data-intensive IoT applications in the fog computing environment. *Software: Practice and Experience* 52, 2 (2022), 482–505. DOI: http://dx.doi.org/10.1002/spe.3032

[142] Xiaofan He, Richeng Jin, and Huaiyu Dai. 2020. Peace: Privacy-preserving and cost-efficient task offloading for mobile-edge computing. *IEEE Transactions on Wireless Communications* 19, 3 (March 2020), 1814–1824. DOI: http://dx.doi.org/10.1109/TWC.2019.2958091

[143] Elie El Haber, Tri Minh Nguyen, and Chadi Assi. 2019. Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds. *IEEE Transactions on Communications* 67, 5 (May 2019), 3407–3421. DOI: http://dx.doi.org/10.1109/TCOMM.2019.2895040

[144] Muhammad Zakarya, Lee Gillam, Hashim Ali, Izaz Ur Rahman, Khaled Salah, Rahim Khan, Omer Rana, and Rajkumar Buyya. 2022. epcAware: A game-based, energy, performance and cost-efficient resource management technique for multi-access edge computing. *IEEE Transactions on Services Computing* 15, 3 (May 2022), 1634–1648. DOI: http://dx.doi.org/10.1109/TSC.2020.3005347

[145] Saeid Ghafouri, Alireza Karami, Danial Bidekani Bakhtiarvan, Aliakbar Saleh Bigdeli, Sukhpal Singh Gill, and Joseph Doyle. 2022. Mobile-kube: Mobility-aware and energy-efficient service orchestration on kubernetes edge servers. In *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*. 82–91. DOI: http://dx.doi.org/10.1109/UCC56403.2022.00019

[146] Yaser Ramzanpoor, Mirsaeid Hosseini Shirvani, and Mehdi Golsorkhtabaramiri. 2022. Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure. *Complex & Intelligent Systems* 8, 1 (Feb. 2022), 361–392. DOI: http://dx.doi.org/10.1007/s40747-021-00368-z

[147] M.Sri Raghavendra, Priyanka Chawla, and Y. Narasimhulu. 2021. A probability based joint-clustering algorithm for application placement in fog-to-cloud computing. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. 1–5. DOI: http://dx.doi.org/10.1109/ICRITO51393.2021.9596534

[148] Umber Saleem, Yu Liu, Sobia Jangsher, Yong Li, and Tao Jiang. 2021. Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing. *IEEE Transactions on Wireless Communications* 20, 1 (Jan. 2021), 360–374. DOI: http://dx.doi.org/10.1109/TWC.2020.3024538

[149] Han Hu, Weiwei Song, Qun Wang, Rose Qingyang Hu, and Hongbo Zhu. 2022. Energy efficiency and delay tradeoff in an MEC-enabled mobile IoT network. *IEEE Internet of Things Journal* 9, 17 (Sept. 2022), 15942–15956. DOI: http://dx.doi.org/10.1109/JIOT.2022.3153847

[150] Khaled M. Matrouk and Amer D. Matrouk. 2023. Mobility aware-task scheduling and virtual fog for offloading in IoT-fog-cloud environment. *Wireless Personal Communications* 130, 2 (May 2023), 801–836. DOI: http://dx.doi.org/10.1007/s11277-023-10310-w

[151] Louiza Yala, Pantelis A. Frangoudis, and Adlen Ksentini. 2018. Latency and availability driven VNF placement in a MEC-NFV environment. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Abu Dhabi, United Arab Emirates, 1–7. DOI: http://dx.doi.org/10.1109/GLOCOM.2018.8647858

[152] Zhe Yu, Yanmin Gong, Shimin Gong, and Yuanxiong Guo. 2020. Joint task offloading and resource allocation in UAV-enabled mobile edge computing. *IEEE Internet of Things Journal* 7, 4 (April 2020), 3147–3159. DOI: http://dx.doi.org/10.1109/JIOT.2020.2965898

[153] Wei Qin, Haiming Chen, Lei Wang, Yinshui Xia, Alfredo Nascita, and Antonio Pescape. 2023. Mcotm: Mobility-aware computation offloading and task migration for edge computing in industrial Iot. (April 2023). DOI: http://dx.doi.org/10.2139/ssrn.4409668

[154] Yu Liu, Yong Li, Yong Niu, and Depeng Jin. 2020. Joint optimization of path planning and resource allocation in mobile edge computing. *IEEE Transactions on Mobile Computing* 19, 9 (Sept. 2020), 2129–2144. DOI: http://dx.doi.org/10.1109/TMC.2019.2922316

[155] Xiaoke Sun, Junhui Zhao, Xiaoting Ma, and Qiuping Li. 2019. Enhancing the user experience in vehicular edge computing networks: An adaptive resource allocation approach. *IEEE Access* 7 (2019), 161074–161087. DOI: http://dx.doi.org/10.1109/ACCESS.2019.2950898

[156] Yuan Liao and Vasilis Friderikos. 2023. Optimal deployment and operation of robotic aerial 6 G small cells with grasping end effectors. *IEEE Transactions on Vehicular Technology* (2023), 1–13. DOI: http://dx.doi.org/10.1109/TVT.2023.3269906

[157] Malte Schwarzkopf, Derek G. Murray, and Steven Hand. 2012. The seven deadly sins of cloud computing research. In *4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 12)*.

[158] Zhishu Shen, Jiong Jin, Cheng Tan, Atsushi Tagami, Shangguang Wang, Qing Li, Qiushi Zheng, and Jingling Yuan. 2023. A survey of next-generation computing technologies in space-air-ground integrated networks. *Comput. Surveys* (June 2023), 3606018. DOI: http://dx.doi.org/10.1145/3606018

Received 29 September 2023; revised 17 July 2025; accepted 29 July 2025