Article:
Patsias, V., Amanatidis, P., Karampatzakis, D. et al. (2023) Task allocation methods and optimization techniques in edge computing: a systematic review of the literature. Future Internet, 15 (8). 254. ISSN: 1999-5903

https://doi.org/10.3390/fi15080254

future internet

MDPI

Review

Task Allocation Methods and Optimization Techniques in Edge Computing: A Systematic Review of the Literature

Vasilios Patsias [1], Petros Amanatidis [1], Dimitris Karampatzakis [1,*], Thomas Lagkas [1], Kalliopi Michalakopoulou [2] and Alexandros Nikitas [2]

[1] Department of Computer Science, International Hellenic University, 65404 Kavala, Greece; vaprtsi@cs.ihu.gr (V.P.); peamana@cs.ihu.gr (P.A.); tlagkas@cs.ihu.gr (T.L.)
[2] Department of Logistics, Marketing, Hospitality and Analytics, Huddersfield Business School, University of Huddersfield, Huddersfield HD1 3DH, UK; k.michalakopoulou@hud.ac.uk (K.M.); a.nikitas@hud.ac.uk (A.N.)
* Correspondence: dkara@cs.ihu.gr

Abstract: Task allocation in edge computing refers to the process of distributing tasks among the various nodes in an edge computing network. The main challenges in task allocation include determining the optimal location for each task based on the requirements such as processing power, storage, and network bandwidth, and adapting to the dynamic nature of the network. Different approaches for task allocation include centralized, decentralized, hybrid, and machine learning algorithms. Each approach has its strengths and weaknesses and the choice of approach will depend on the specific requirements of the application. In more detail, the selection of the most optimal task allocation methods depends on the edge computing architecture and configuration type, like mobile edge computing (MEC), cloud-edge, fog computing, peer-to-peer edge computing, etc. Thus, task allocation in edge computing is a complex, diverse, and challenging problem that requires a balance of trade-offs between multiple conflicting objectives such as energy efficiency, data privacy, security, latency, and quality of service (QoS). Recently, an increased number of research studies have emerged regarding the performance evaluation and optimization of task allocation on edge devices. While several survey articles have described the current state-of-the-art task allocation methods, this work focuses on comparing and contrasting different task allocation methods, optimization algorithms, as well as the network types that are most frequently used in edge computing systems.

Keywords: task offloading; edge computing; task allocation; optimization algorithms

Citation: Patsias, V.; Amanatidis, P.; Karampatzakis, D.; Lagkas, T.; Michalakopoulou, K.; Nikitas, A. Task Allocation Methods and Optimization Techniques in Edge Computing: A Systematic Review of the Literature. *Future Internet* **2023**, *15*, 254. https://doi.org/10.3390/fi15080254

Academic Editor: Guan Gui

Received: 8 July 2023
Revised: 20 July 2023
Accepted: 26 July 2023
Published: 28 July 2023

1. Introduction

Edge computing refers to a type of computing that is decentralized and allows for data processing to occur closer to the data source, rather than relying on centralized data centers [1]. This approach is particularly advantageous for applications that demand a high bandwidth and low latency, such as autonomous vehicles, Internet of Things systems, and augmented reality applications at the edge of the network [2]. In edge computing, the distribution of tasks among the various nodes in a network is referred to as task allocation. Choosing the best place for each task is one of the primary difficulties of task allocation in edge computing. This is because the different tasks demanding varying amounts of computing power, storage, and network bandwidth. A work that needs real-time processing, for instance, could need to be assigned to a node with high computing power, while a task that needs a lot of storage would need to be assigned to a node with a lot of storage space. Additionally, tasks that are sensitive to network latency may need to be allocated to nodes that are close to the source of the data. This necessitates a trade-off between the node's computing power, energy usage, and network delay. In Figure 1, we can see the architecture of and edge–fog–cloud computing system.

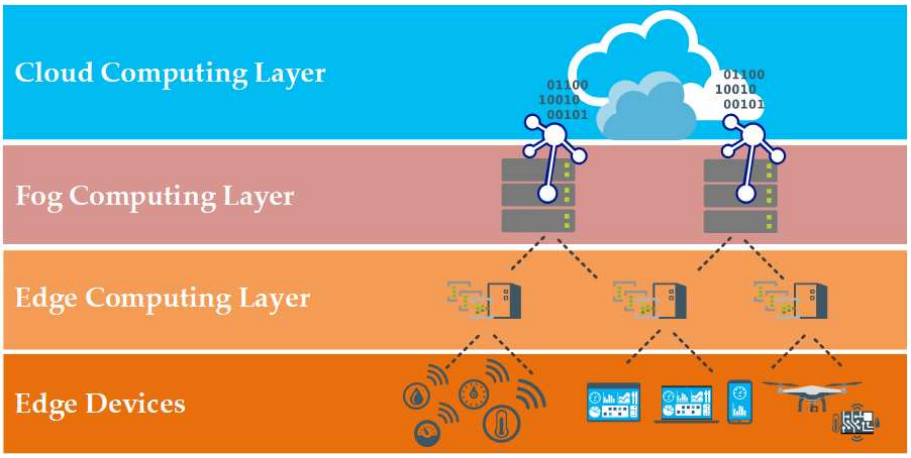

Figure 1. Edge, fog, and cloud computing architecture.

Furthermore, task allocation in edge computing is dealing with the dynamic nature of the network. Edge computing networks are often composed of a large number of devices that can join or leave the network at any time. This means that the number of available resources and the location of those resources can change frequently. Task allocation algorithms must thus be able to adapt to these changes and decide in real time. Furthermore, several competing goals may need to be taken into account, including energy efficiency, data privacy and security, and QoS assurances.

In edge computing, there are several methods for allocating tasks, each with its advantages and disadvantages. Using a central algorithm that makes decisions based on the network's present state is a common strategy. This strategy calls for a central controller to keep an eye on the network and decide how to distribute the computational tasks. This strategy, however, may be susceptible to network outages and result in a decision-making bottleneck. Also, it may not be feasible to have a central controller in large-scale edge computing networks. Additionally, the use of a decentralized algorithm allows each node in the network to make decisions about task allocation independently. This strategy is predicated on the notion that every node has a local view of the network and can take decisions based on that perspective. This approach can be more robust to network failures and can lead to faster decision making. However, it may lead to sub-optimal solutions due to lack of global information [3–9].

The algorithms that adopt both centralized and decentralized techniques are called hybrid. These algorithms are designed to take advantage of the strengths of both centralized and decentralized approaches. For example, a hybrid algorithm might use a centralized algorithm to make high-level decisions about task allocation, while allowing each node to make low-level decisions based on its local view of the network. This approach can produce more effective solutions by balancing the trade-offs between centralized and decentralized methods. Utilizing machine learning methods to enhance work allocation is another current strategy. Algorithms that use machine learning can learn from past data and forecast how the network will behave in the future [10]. As a result, the algorithm may be able to allocate tasks more precisely. The network's energy usage may also be optimized using machine learning methods. For instance, a machine learning algorithm may be taught to predict how much energy various nodes will need and make decisions about task allocation based on that prediction.

This work examines the current methods for task allocation on edge devices. We evaluate the key techniques discussed in the literature to see how effective each of them is for edge computing and which would be most helpful for future research. We examine and evaluate the corpus of contemporary research on workload distribution on edge intelligence devices. Our work provides a thorough assessment of the main methods and algorithms used in edge computing for efficiently executing difficult workloads [11–79].

The main contribution of this paper is that it provides a review of the state-of-the-art task allocation methods. More precisely, this paper only used research published in the last three years and it highlights the most frequently used task allocation methods and optimization algorithms in combination with the network types which are used in edge computing systems according to the application and the task allocation algorithm used. The purpose of this work was to show the task allocation and all the optimization algorithms and methods that have been, very recently, the most frequently used in order to advance the current understanding of choice-making when it comes to adopting the optimal task allocation method, algorithm, and networks according to the use case. All the task allocations methods reviewed in this paper tried to give a solution to the edge servers or the devices located in the edge computing layer by distributing their workloads to the end devices, fog layer devices, and cloud computing layer devices in order to provide optimizations according to the use cases of the applications.

The document is divided into six sections. The systematic literature review (SLR) approach employed in this article is described in Section 2, including the definition of research questions (RQs). Next, in Section 3, we provide the list of findings in the literature. In Section 4, we provide a summary of the collected papers. In Section 5, we provide a critical discussion of our findings and identify future work challenges. We finally conclude this review in Section 6.

2. Methodology and Research Questions

Conducting a thorough assessment of the existing literature is commonly achieved through a systematic literature review (SLR), which is recognized as one of the most widely used approaches [80–83]. This type of study requires a clearly defined process to accurately identify relevant research. A review should follow a specific protocol to effectively collect and assess previous works. The methodology which was followed in this paper incorporates four steps: first, recognize the research questions; second, explain the literature sources and the search string; third, select applicable studies; and fourth, assess the studies gathered and extract the necessary data before combining them.

This review aims to identify the task allocation approaches on edge devices. Thus, the RQs defined are the following:

- **RQ1:** What are the contemporary task allocation techniques in edge computing?
- **RQ2:** What are the most frequently used task allocation optimization algorithms used in edge computing?
- **RQ3:** What are the most used computer networks in edge computing?

The following method was employed to create a search string clearly corresponding to our RQs. The key search terms first and foremost need to be RQs-specific. Then, further search terms are obtained utilizing previously examined documents. The next stage is to search for alternative spellings and synonyms for the main keywords. The final step is to combine the keywords to create the search string.

3. Research Process

To better understand task allocation in edge computing, in the following review, we will examine the most recent papers that refer to the specific topic. To achieve this, we conducted an extensive search in the Scopus search engine [84] for papers relevant to "task allocation on edge computing". Therefore, search terms such as "task allocation" and "edge computing" are used to search in the titles, abstracts, and keywords of publications. We narrowed the search to the years after 2020 to catch out on the latest technologies in the field we are examining. The initial search yielded back 218 documents. Consequently, using the following inclusion/exclusion criteria:

- In cases where the study has a conference version and a journal version of a study, the journal version is retained while the conference version is discarded.

- If there are multiple published versions of a study, only the latest or most recent version is retained.
- If a study is available in multiple sources, only one version is included.

Exclusion criteria were also used to omit research that did not meet the requirements for inclusion. The following are the specific exclusion criteria:

- Papers not available on an open access basis or through our university access systems that are opening the vast majority of the reputable editor houses and journal titles were not included in the final literature corpus.
- Only documents written in the English language were used.
- Only papers related to computer science were included; works that were only peripheral to edge computing were excluded.
- Papers dealing with very specific datasets were excluded. We focused on conceptual papers, case studies, methodological papers, and position papers.

We do acknowledge that potentially missing a relatively minor number of papers that we excluded from our systematic review approach due to the unavailability of access to the sources, these were published, and may have a minimal impact on the completeness of the corpus and thus on our results. We would argue, however, that our Universities' journal access agreements in Greece and the UK are very inclusive and only miss very specialized and small publishers. We anticipate that this had an extremely limited impact on our work.

In this research, we used the following keywords (with logical connectives) and their combinations: edge computing, task allocation, IoT, energy utilization, mobile edge computing, computation offloading, integer programming, resource allocation, resource management, antennas, reinforcement learning, task analysis, fog computing, energy efficiency, green computing, task offloading, UAV, 5G mobile communication systems, cloud computing, data handling, job analysis, MEC, network architecture, cloud-computing, computing environments, deep learning, edge server, fog, learning algorithms, big data, computation-intensive task, computational modeling, computing resource, energy consumption, learning systems, multi-agent systems, optimization, QoS, servers, task allocation algorithm, and total energy consumption.

To answer RQ1, the task allocation methods are categorized according to the following methods:

1. Resource-aware task allocation: Allocating tasks depending on the resources that are currently accessible on edge devices, such as processing power, memory, and battery life.
2. Distributed task allocation: Allocating tasks across a network of edge devices to optimize the performance and reduce latency.
3. Dynamic task allocation: Adapting task allocation in real-time based on changes in device performance or network conditions.
4. Machine learning-based task allocation: Using machine learning algorithms to predict resource utilization and allocate tasks accordingly.
5. Energy-efficient task allocation: Allocating tasks in a way that minimizes energy consumption in edge devices.
6. Quality of service (QoS)-aware task allocation: Allocating tasks based on the required QoS levels for different tasks.
7. Collaborative task allocation: Allocating tasks across a network of edge devices by taking into account the collaboration between devices.
8. Context-aware task allocation: Allocating tasks based on the context of the edge devices, such as location, available resources, and user preferences.

Table 1 and Figure 2 present a comparative analysis of those papers based on the classification and contextualization of their key results. Note that the columns in Table 1 are directly related to the answers addressing our RQs.

Table 1. Summary of task allocation methods in edge computing.

	Proposed Task Allocation Optimization (RQ2)	Applied Network Type (RQ3)
		Distributed task allocation
[18]	FL	WHAB
[22]	OFB	VANETs
[24]	GT, RL	MEC
[29]	MVAA	MEC
[45]	PSO	IoT
[50]	GT IoT	
[55]	Hybrid	MEC
[59]	CW	Blockchain
[61]	BSUB	MEC
[63]	FL	MEC
[31]	DLA	IoT
[33]	RL	EC
[34]	CFG	C-RAN
[35]	AsP	IoT
[36]	GSO	MEC, Fog
[65]	AsP	NDN-IoT
[66]	AB, GA	Wireless, cellular, satellite
[67]	Heuristic	MEC
		Collaborative task allocation
[12]	PFA	EC
[77]	MINLP	EC
[17]	ACO	IoT
[27]	Heuristic	IoT
		Context-aware task allocation
[52]	DFD	DAG
[69]	GT	IoT, 5G
[78]	LPA	MEC
		Energy-efficient task allocation
[68]	MAPE-K	IoT
[11]	Lyapunov	MEC
[19]	Bi-Level	MEC
[25]	DRL	IoV
[30]	QT	5G
[43]	PSO	IoT
[47]	ILP	MEC
[71]	JTORA	MEC
[73]	hybrid RF-FSO	Industrial IoT
		Dynamic task allocation
[13]	MPSO	VVECNs, VANETs
[79]	MH	IoV
[20]	AA	Fog
[23]	DP	MEC
[41]	DRL	IoT
[44]	ECTA	EC
[46]	AA	Fog
[49]	ILP	
[56]	GA, GEN	IoT
[60]	PSO	EC
[64]	BPSO	

Table 1. *Cont.*

	Proposed Task Allocation Optimization (RQ2)	Applied Network type (RQ3)
		Machine learning-based task allocation
[28]	DQN-D	EC
[54]		IoT
[38]	MARL	IoT
[42]	ACO	IoT, 5G
[51]	FLOM-Opt	
[53]	Q-Learning	IoT, IoV
[57]	MA	IoT
[58]	Heuristic, RL	EC
		Quality of service task allocation
[26]		EC
[32]	MMAS	EC
[48]	QT	EC, IoT
		Resource-aware task allocation
[76]	MAPPO	MEC
[62]	MDP	EC
[14]	MAP	VEC
[75]	JTORA	VEC
[72]	JTORA	MEC
[74]	JTORA	MEC
[15]	DNF	
[16]	MARL	MEC
[21]	JTORA	NOMA-MEC
[37]		EC
[39]	ELB	
[40]	Knapsack	MCS
[70]		EC, 5G

The task allocation methods (RQ1) covered by the papers we examined are shown in Figure 3. Most of the papers use distributed task allocation (26.1%), followed by resource-aware (18.8%) task allocation. Dynamic task allocation (15.9%), energy-efficient (13.2%), and machine learning (11.6%) task allocation followed next. Collaborative (5.8%), quality-of-service (4.3%), and context-aware (4.3%) task allocation are the methods that are mentioned less. It is also very important to note that, in many papers more than one task allocation methods was used and applied.

Regarding the task allocation optimization algorithms (RQ2), as shown in Figure 4, the most frequently used is the swarm optimization in various forms (10.1%), various methods of reinforcement learning (10.1%). JTORA (7.2%), multi-agent methods (5.8%), the game theory (4.3%), the heuristic approach (4.3%), and the auction algorithm (4.3%). We also identified the use of many other task allocation optimization algorithms that were unique; these cases refer to 53.8% of the sample examined. We should note again that many authors in their works implemented more than one optimization algorithm to illustrate the effectiveness of each optimization algorithm.

The communication network types (RQ3) are illustrated in Figure 5. Most papers applied IoT networks (27.6%), followed by mobile edge computing (MEC) networks (26.1%), EC networks (18.8%), vehicular networks (10.1%), and 5G networks (5.8%). The rest are illustrated as one category 'various kinds of networks' (11.6%).

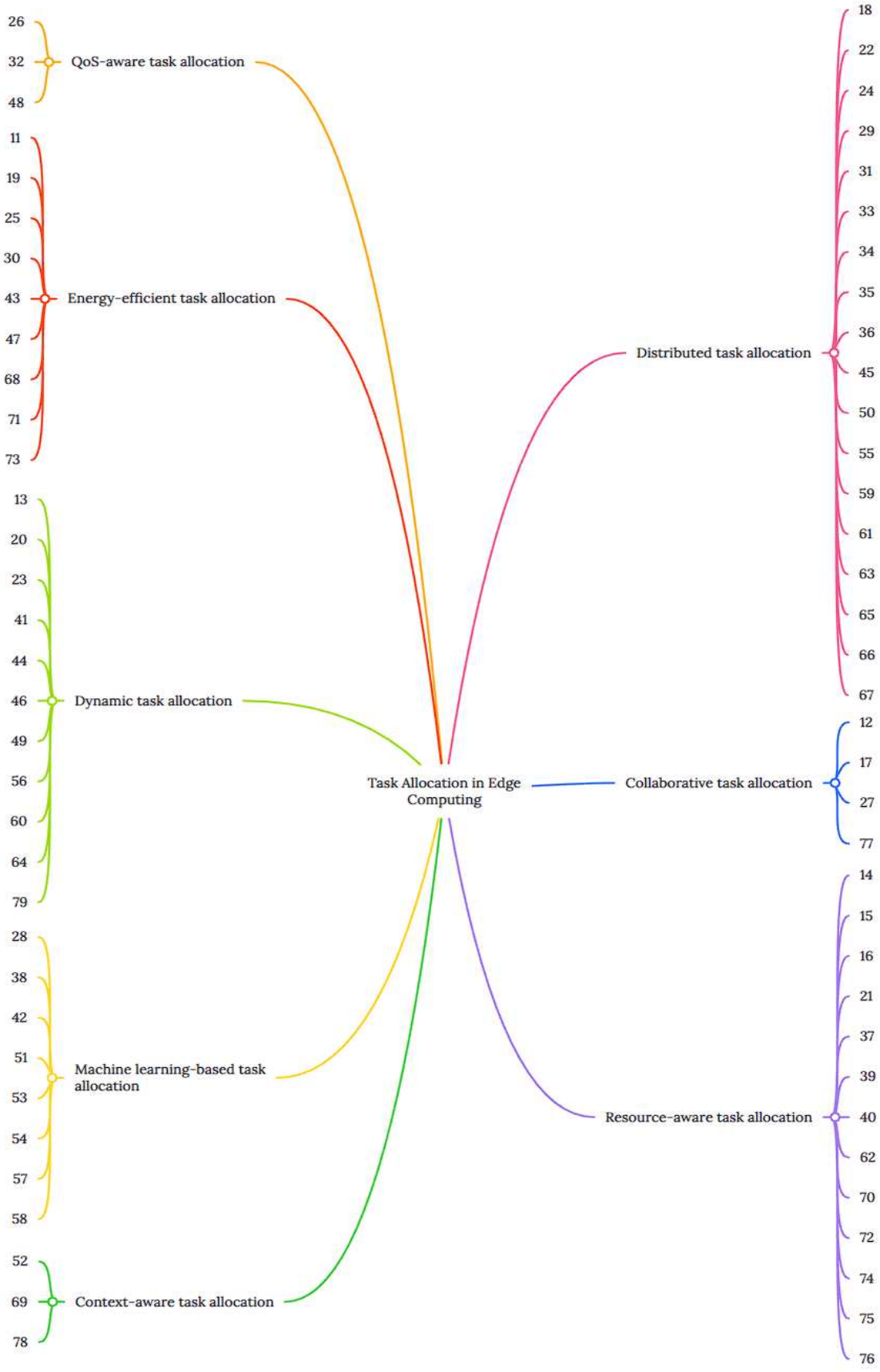

Figure 2. The classification mind map from the analysis of papers in task allocation methods in edge computing.

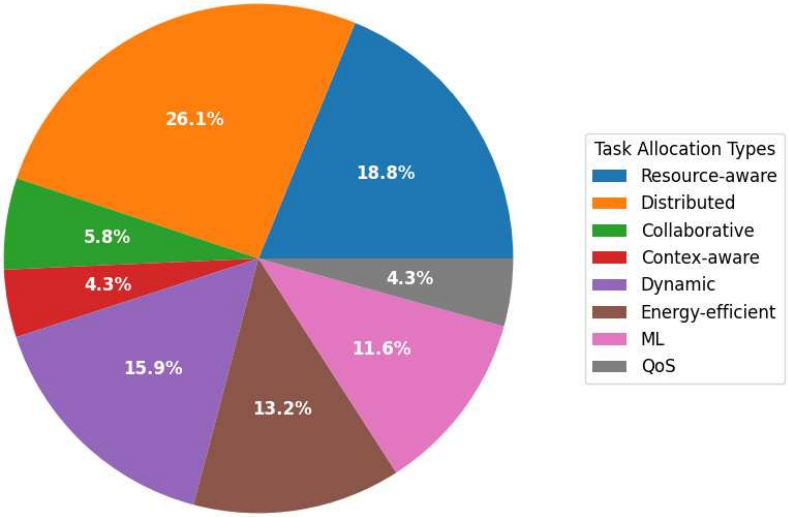

Figure 3. Percentages of the task allocation methods used.

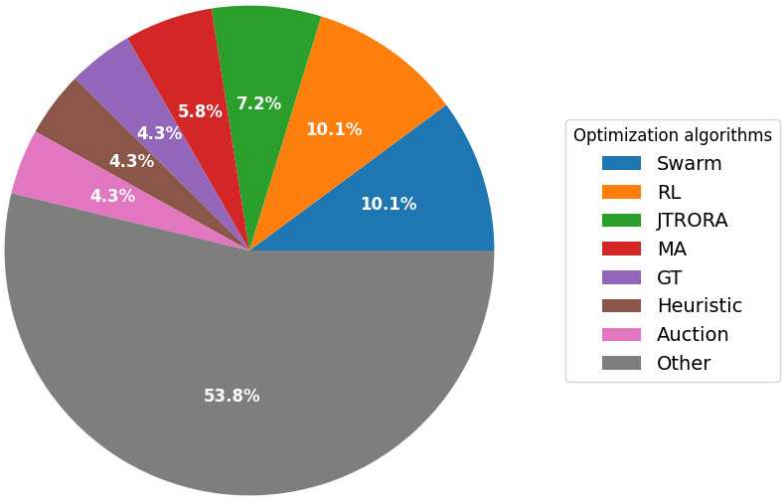

Figure 4. Percentages of allocation optimization methods used.

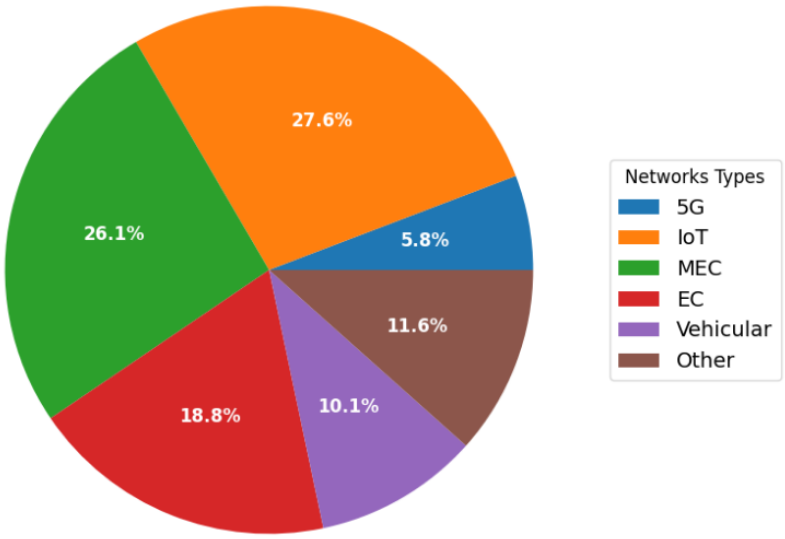

Figure 5. Percentages of communication networks used in task allocation.

4. Literature Review and Data Extraction

This section presents a classification of the task allocation methods used in the recent literature. Each task allocation method in combination with the computer network type and the deep learning compression methods has its own strengths and it is used to address different objectives such as minimized computational power, energy consumption, and network delay.

4.1. Distributed Task Allocation

Applying edge computing in high-altitude balloons (HABs), as per Sihua Wang et al. [18], take into account a HAB network that supports MEC and allows users to periodically request computing tasks with different data volumes. The HABs must dynamically decide on the best job allocation, service sequence, and user association to provide their users with computing services. The objective of this combination of job allocation, service sequence, and user association was to reduce the weighted total of the time and energy used by all users. The authors propose a federated learning (FL) approach that uses support vector machines (SVMs) to predict user association in advance, addressing the optimization challenge in task allocation. By using this approach, the heterogeneous autonomous base stations (HABSs) can collaboratively train an SVM model that can accurately predict the best user association without needing to transmit large amounts of task data or previous user association results. The original, nonconvex issue is split into two sub-optimization problems—the task allocation optimization problem and the service sequence optimization problem—which are then addressed repeatedly based on the expected user association. Specifically, the authors developed a closed-form formula for the ideal service sequence given the job allocation vector. This optimization problem may be converted into a piecewise linear problem that can be addressed using linear programming if the best service sequence is known.

Previous research showed [22] the occurrence of a multi-device game that offloads processing for several users. Computing efficiency may be significantly increased by using the various available intelligent devices that industrial vehicles can assign calculation assignments to perform the operation in parallel. Moreover, to determine the optimal offloading strategy, they formulated a sequential game of multi-user computation offloading, treating multiple industrial vehicles as multi-devices for handling multiple targets. They took into account that the system cost is primarily made up from the price of rental IDs, energy consumption, and execution time. To improve the chances of establishing wireless connections between industrial vehicles (IVs) and unmanned aerial vehicles (UAVs), a dynamic scheduling technique based on the density of small partitions was introduced. The current vehicle density in the partition determines the UAV's residence time, which was scheduled by the software-defined network controller. They developed an algorithm for minimum incremental task allocation (MITA), a job distribution mechanism for industrial vehicles to distribute some of the computational duties among several IDs. MITA was determined as the best multi-target task allocation strategy and reduces the system cost associated with task execution.

In order to reduce the user response time, balancing the load, and maximizing server resource usage, Zhenjiang Zhang et al. [24] developed a task offloading technique based on a multi-agent approach. Additionally, the authors addressed different problems that centralized scheduling would create and also presented the benefits of centralized training and dispersed operation. The authors used an experimental approach to confirm the viability and efficacy of their distributed task offloading algorithm based on multi-agent and load balancing (DTOMALB) method and the analysis of the simulation results.

This paper [29] addressed the problem in MEC of indivisible task allocation with heterogeneous resources (TAHRC). In a scenario with diverse resources, the authors provided an accurate mathematical method for indivisible task allocation. Furthermore, an effective task allocation that takes into account various types of resources, as well as binary computation offloading, was presented. The authors defined TAHRC as an integer

programming problem. The TAHRC was divided into two subproblems: the resource allocation and offloading choice. Due to the difficulty of computing the optimal allocation, a heuristic method was proposed to address the resource allocation subproblem, along with an approximation algorithm to handle the offloading choice subproblem.

A recent study by Tingyan Long et al. [45] proposed a novel approach for fault-tolerant scheduling in collaboration procedures carried out in edge IoT environments. In this work, a solution that combines a deep Q-learning network (DQN) and a PB mechanism to ensure process execution by choosing the best backup component of the workflow was presented. The job completion rate, server active time, and resource consumption reveal that the suggested technique outperforms conventional methods, according to the authors' comprehensive simulations. The study addresses the differences between the suggested methodology and earlier work while also discussing related fault-tolerance method studies. Overall, the suggested technique offers a fault-tolerant scheduling mechanism for multi-task processing on distributed edge-IoT systems that is more effective and dependable.

A novel architecture for IoT networks that incorporates UAVs functioning as mobile fog nodes, termed hybrid-hierarchical spatial–aerial–terrestrial edge-centric (H2TEC) was presented by Abbas Jamalipour et al. [50]. While ensuring network dependability and outage probability, the suggested task allocation protocol (TOP) reduces energy consumption and maximizes the transmission rate of terminal nodes (TNs). When a UAV's energy falls below a set threshold, the protocol uses energy harvesting techniques to transmit energy to the UAV. The collected power and harvesting time are improved, and the suggested protocol outperforms the baseline procedure, according to the numbers. In order to enhance the functionality of fog-IoT networks, the study evaluates the literature on mobile fog nodes and suggests an energy model and a throughput model. The research also offers a thorough system model, explains how network latency, energy consumption, and UAV flight height are affected by one another, and suggests a problem for optimizing the flight altitude. Overall, the research offers a unique approach to the problem of mobile fog nodes in fog-IoT networks with a constrained energy budget.

Da Li et al. [55] investigated whether the use of massive MIMO and non-orthogonal multiple access (NOMA) technology can improve the realization of quick fault elimination, by minimizing the data transmission time, and boosting the transmission rate. The computation of the acquired data can be offloaded to the MEC server. In order to accomplish the quick correction of errors in the intelligent distribution network and achieve stable operation, the massive MIMO–NOMA technology is thought to be applied to MEC.

Recently, a work was presented by Weize Xu et al. [59] discussing a decentralized crowdsourcing system and a task assignment algorithm used to address the issues of task allocation. The authors offer a blockchain-based solution to address the problems with traditional, centralized crowdsourcing platforms, such as user privacy leaks and extra service charges, for multi-skill mobile crowdsourcing assignments. With the help of blockchain technology, the suggested system enables direct contact between the requester and the worker, and offers infrastructure services that are highly accessible, decentralized, and private. The system design allows mobile crowdsourcing projects with varied skill needs by including worker and task management contracts as well as a fair and transparent task allocation method. Additionally, the system has a solid mechanism for ensuring fairness and is economically viable with substantially less added consumption. Comparing the suggested system with current crowdsourcing models based on blockchain, the experimental findings reveal that the proposed system provides accurate task matching at a lower cost and greater job allocation rates. Overall, the suggested approach offers a better method for handling multi-skill mobile crowdsourcing jobs, improving the process efficiency and fairness.

A hierarchical fog-computing cloud radio access network (CRAN) architecture that combines fog access points (FAPs) with MEC servers to reduce task offloading's wait time was published by Yijin Pan et al. [61]. The system includes multiple levels of MEC-L servers in radio units (RUs), MEC-H servers in distributed units (DUs), and a cloud computing service in central units (CUs), which helps decrease the distance needed for

offloading computational tasks and improving the delay performance for computation-intensive and delay-sensitive applications like virtual reality and automated driving. The authors suggest a mixed-integer nonlinear programming problem aimed at minimizing delay by optimizing the distribution of jobs, the transmission beamforming vectors, the processing speed, and the allocation of transmission bandwidth. They loosened the integer restrictions to convert the issue into a convex problem and solved it using the Lagrange dual approach. The hierarchical C-RAN network based on fog computing that was suggested outperforms current networks in terms of latency performance, according to the simulation findings. The suggested structure enables access to pooled processing power and storage resources, which can assist in managing a sizable number of offloading requests and saving operational expenses.

A recent work [63] discusses an FL method that uses attention to handle the collaborative task allocation (CTA) issue in contexts with edge-assisted smart grids. The CTA challenge entails improving both user equipment task completion delay and energy usage while taking subtask dependencies into consideration. The authors offer an attentive approach to extract information from both the power UE and the power base and characterize the relationships as a directed acyclic graph (DAG). The attention-aided FL method was then suggested in order to determine the best course of action for the CTA issue as an MDP. The suggested method outperforms existing options, according to the simulation findings. The suggested method is anticipated to increase the data accuracy and efficacy in situations involving smart grids, as well as data privacy and security during multi-party data sharing.

Muhammad Mudassar et al. [31] suggested that the on-time execution of a distributed, resource-intensive activity requires a method to perform the decentralized grouping of edge nodes. The authors offered a mathematical framework for distributing resources fairly among tasks in distributed edge execution environments. Nodes in the edge network have their reliability characteristics which were assessed. An effective decentralized fault tolerance mechanism is offered for heterogeneous edge networks that are prone to errors. Furthermore, in this work, an algorithm for backup nodes based on a dependability model that offers redundancy was selected and used. Finally, they thoroughly evaluate the suggested strategies to confirm their effectiveness.

Prior research [33] suggests that a graph-based combinatorial optimization problem can be employed to simulate a distributed deployment classifier. Even though the greedy approach can obtain a reasonable approximation of the answer, real-time systems have trouble keeping up with its slow response time. Moreover, an reinforcement learning (RL)-based classifier deployment technique was provided that uses graph convolutional network to learn network structure details in order to choose the best modification strategy for the given circumstance at each step until all nodes are chosen. The in-depth simulation tests showed off the effectiveness of their technique. First, they demonstrated a decent approximation solution and greatly minimized the computation time by comparing their approach to the greedy one. Second, a demonstration of contrasting distributed throttle techniques using a classifier yielded superior results. Finally, the authors compared their deployment technique with several other randomly dispersed deployment strategies to demonstrate its advantage.

A more systematic analysis was presented by Yijin Pan et al. [34] which studied a hierarchical fog-computing CRAN network with three levels of computational services, namely the MEC server (MEC-L) located in radio units, the MEC server (MEC-H) located in distribution units, and cloud computing located in central units. The authors focused on optimizing the received beamforming vectors, task distribution, processing speed, and transmission bandwidth. More specifically, they aimed to reduce the overall latency of the computing tasks, by lowering the integer constraints. In addition, they defined a mixed-integer nonlinear problem (MINLP) and transformed it into a convex problem before applying the Lagrange dual method to solve it.

Furthermore, this work [35] offered a working prototype of their multi-layered architecture, which combines fog computing and cloud computing layers to perform latency-

sensitive analysis. The major objective of this research was to improve the real-time data analysis system based on IoT devices by using this multi-layer fog computing platform. Their study is based on the OpenFog Consortium policy, which offers good results as well as monitoring and data analysis features. Through case studies, this work demonstrated how their suggested prototype architecture performed better in terms of delay time, network utilization, and energy consumption than the cloud-only environment.

Previous research made by Ke Fu et al. [36] presented a study on multi-user multi-MEC server scenarios of the MEC environment. In this paper, to balance delay and energy consumption, a penalty mechanism was used, aiming to mimic real-world scenarios. The authors characterized the offloading choice in order to minimize the delay under the restrictions of energy consumption. Furthermore, a detailed examination of the glowworm swarm optimization (GSO)'s computing methodology was provided. Finally, the simulation tests were carried out, and the findings demonstrated that their approach has a higher performance of energy savings and delay reduction in the MEC environment compared to the alternative compute offloading technique by up to 25%.

Recently, researchers [65] have also focused on utilizing MEC to effectively address the processing demands of new IoT applications while assuring the security of the sent data. To reduce the overall time delay in computing the executions of intelligent terminals, this research proposes an optimization problem that considers the allocation of computing tasks and the security performance requirements of intelligent devices within the edge computing topology. The authors recommended resource management and genetic algorithms to identify the optimal solutions to the problem. This study also takes security issues into account by assessing the likelihood of a level of security breach by representing the security level of computing data. Furthermore, the authors enhanced IoT applications' efficiency and security when leveraging MEC technologies. In order to confirm the efficiency of the suggested algorithms, the researchers compared them to heuristic algorithms using data testing and simulations. The research offers information on how MEC might improve the productivity and security of IoT applications.

Jianhua Liu et al. [66] proposed a thorough optimization model that was built for the UAV environment. The model aimed to optimize the costs of migration, operation, and QoS, while accounting for the energy constraints and location sensitivity of the nodes. An online method was suggested for this convex optimization task. The proposed method was thoroughly analyzed in terms of competitiveness, and it was shown that it has a parameterized competitive ratio. The results of the simulation indicate that the suggested approach may save 20% of the overall cost compared to previous approaches and has dependable robustness independent of the user and UAV node counts.

A resource allocation plan for a multi-robot cooperation (MRC) system that reduces the overall energy use of SRs while meeting the task implementation delay limit was proposed by Lanxin Qiu et al. [67]. The authors considered a complex, realistic situation in which a master robot is in charge of task offloading and wireless communication resource management while the slave robots are in charge of sensing and data gathering. As a result, the suggested task implementation technique for the MEC-based MRC system in which they divided each task into three components is a fair and efficient resource scheduling plan that may reduce the SR's energy usage. The findings demonstrated that the suggested plan may successfully lower the slave robot's overall energy usage.

4.2. Collaborative Task Allocation

Xiaoheng Deng et al. [12] developed an edge computing system called cloudlet assisted cooperative task assignment (CACTA), which uses a method to allocate dynamically workloads to edge nodes across several time slots. The goal of this approach is to reduce the time required to complete tasks, either by itself or by working together to minimize both the overall task completion time and the cost of the system. The authors formulated this problem by considering the variable computation capabilities and costs of edge nodes, which are subject to uncertainty and stochastic behavior. They then derived an offline

optimal solution used as a benchmark to evaluate the online algorithm's performance. The proposed method (PA-EMA) was created to solve the issue where the system does not have access to long-term historical data. The effectiveness of the multi-round allocation approach was validated by this algorithm. Also, the authors presented the PA-OPT algorithm, which can deliver performance that is close to optimal to deal with the second circumstance of acquiring historical data. The PA-OPT algorithm implements the ARIMA approach to predict the computing capacity and cost of a node by examining historical data and information obtained during the task execution. The CACTA system used actual parameters from a video processing application and an empirical dataset from the Google Cluster to conduct experiments. The presented findings recommend that the suggested online prediction algorithm was able to achieve outcomes that were almost as good as optimal, especially in cases where an edge node had high computational power and cost predictability. However, when the edge node's predictability is low, its performance is similar to that of other algorithms. In conclusion, this work presented an online algorithm based on Q learning to achieve joint optimization and real-time resource load balancing. The algorithm used a traditional RL approach and showed better performance compared to other existing algorithms.

Furthermore, a paper presented by Guangshun Li et al. [17] proposed a method to collect node state information in their paper. The method involves categorizing a node's state by combining intrinsic and real-time attribute values. An intermediate node is then used to send back the produced outcome information. After that, the authors of the study proposed a task allocation model to achieve dynamic system balance. The model involves allocating new tasks to the comparatively lightest nodes and the task arrival node, while temporarily not assigning tasks to the other nodes. The initial state of the nodes was classified using the naive Bayes method by the authors. Additionally, the normalization of the initial data prevented stressing the importance of higher-value indicators in the thorough analysis when the levels of the indicators differ significantly. In order to accomplish dynamic balance, the nodes with relatively light categorization states and the nodes of task arrival as the target nodes were chosen to allocate new tasks. Finally, to analyze the load balancing issue among edge nodes, a mathematical framework was constructed. The load balancing was achieved by assigning tasks and estimating the completion time based on the transmission rate between edge nodes, computation speed, and the current task calculation time.

A cooperative edge computing task allocation system was presented by Qianjun Wang [27] that utilized the maximized resource efficiency and minimized service lag. The proposed model creates a cooperative computing model for two edge nodes. The model was converted into an integer nonlinear programming problem, and the best solution is found using the two-edge-node cooperative-task allocation based on the improved particle swarm optimization (TCA-IPSO) technique. The simulation results showed that the TCA-IPSO algorithm significantly outperformed the benchmark and QBTD methods, with a reduction in the average task completion latency by 53.8% and 36.0%, respectively. In an IoT scenario, this study showed that cooperative edge computing is a potential method for distributing tasks and utilizing resources as efficiently as possible.

4.3. Context-Aware Task Allocation

Previous research [52] highlighted the need for high-performance computing in smart environments. In this work, the authors proposed an effective task allocation algorithm for global heterogeneous systems that are DFD (Data-Flow-Driven). In order to achieve local computation and low latency, the authors suggested fog computing and edge computing topologies. To address the work distribution issue for large-scale scientific computing systems leveraging remote data sources, this paper provides two optimization techniques based on mixed-integer linear programming (ILP). Compared to the optimum matching model, the staged optimization model's solutions were shown to be nearly ideal in all situations. In conclusion, the authors acknowledged the need for more study in the areas of

availability monitoring, predicting application behavior, and finding nearly ideal solutions in a very short amount of time.

A collaborative edge computing system presented by Yanzan Sun et al. [69] is a solution tailored to increase application loading, workload distribution, and resource allocation while preserving application service provider profitability. The advent of mobile edge computation, which offers computation, storage, network, and communication resources at the edge for various applications, has made it feasible to connect telephony with Internet services. Physical memory limitations prevent the edge server from running many application services at once, and uneven traffic distribution over the mobile network makes it challenging to make the most out of the edge network's resources. The suggested approach utilized a profit-aware technique for minimizing system latency in long-term stochastic optimization issues with assignment problem (ASP) profit constraints. It also ensured ASP profitability and used genetic algorithms to achieve a near-optimal approach for each time slot. The simulation results showed that the algorithm reduced the system latency in the long term while maintaining application service provider profits.

4.4. Energy-Efficient Task Allocation

Feng Wang et al. [11] covered a complete solution for wireless-powered MEC systems—which combine MEC with wireless power transfer (WPT). The suggested system was a single-user MEC system that employs energy beamforming for WPT toward the user and has a single multi-antenna energy transmitter. The user uses the captured energy to run computations locally and offloads some of them to a distant MEC server through an access point that is integrated with the server. This study proposes a method to optimize the allocation of transmission energy for WPT and the allocation of local computing and offloading tasks over a certain period while considering the energy and task causality constraints caused by channel fluctuations and dynamic task arrivals. This study provides an offline optimization that, under the assumption of perfect information, uses convex optimization techniques to produce an optimum result that is well structured. When channel and task state information are only causally known, online designs are also created for the combined allocation of energy and tasks. The suggested solutions outperform traditional myopic systems and benchmark schemes in terms of energy usage according to numerical data.

An architecture for a secure MEC system was presented by Jun-Bo Wang et al. [19] where a multi-antenna access point (AP) contains an MEC server and offers to compute offloading services for several mobile devices. Each MD used a protocol for partial offloading to complete the computations in blocks. The authors provided secured offloading limits that state that each MD's offloading rate cannot exceed its attainable secrecy rate to the AP in order to prevent information loss during offloading to the eavesdropper. Following that, an optimization problem was created to reduce the system's total energy usage, with respect to the restrictions on secure offloading and computation latency. The distribution of computing jobs, offloading power, local CPU frequency, and timeslots were all cooperatively optimized. The initial optimization issue was nonconvex because of the safe offloading requirements. To find the iterative convex approximation of the issue, the authors used the difference of the convex approach. The answer was then derived in a semi-closed form using the Lagrange dual technique. The authors also provided a low-complexity algorithm to further minimize the computational cost. Finally, quantifiable results are provided about how well the suggested strategies performed. It was demonstrated that the suggested strategies greatly lower the MEC systems' energy usage. In particular, the existence of the eavesdropper had no effect on task offloading when the distance between the MDs and the eavesdropper reaches a specific distance, and the energy consumption of the computing jobs remains consistent. As result, a large separation between the mobile devices and the listener could lower the power needed for safe offloading.

Recently, a framework for offloading the workloads to optimize energy consumption in the Internet of Vehicles (IoV) was presented by Michailidis et al. [25]. This work suggested a network architecture supported by UAVs for offloading computationally demanding

activities to roadside units (RSU) that incorporate MEC functions. A dual-reconfigurable intelligent surface (RIS) configuration that serves two neighboring network nodes was introduced. Each RIS unit functions as a data offloading aerial relay as well as an aerial-roadside unit (ARSU). To reduce the energy use of the cars and ARSU while taking into account transmit power limits, timeslot scheduling, and job allocation, an optimization technique was provided. In order to confirm the effectiveness of the improved dual-RIS-assisted wireless transmission model, the authors carried out extensive numerical computations. According to the article, RIS units can enhance end-to-end communication during data offloading. This study also covered issues of highly reliable communication networks, NOMA configurations, and earlier MEC-enabled IoV contributions.

A cooperative task offloading solution using the base station in 5G heterogeneous networks was presented in [30]. This study considers both the waiting energy and delay consumption of the offloaded jobs and uses queuing theory to address them. The issue of choosing whether to offload a task or not is transformed into a task assignment probability problem. The MDs-centered minimization of delay and energy consumption was specified as the optimization objective. In addition, the study proposes a method that assigns probabilities to job assignments, resulting in combined optimization of mobile device consumption, while considering varying demands for delay and energy consumption. The proposed method uses a quasi-Newton interior point (TA-QNIP) algorithm to minimize consumption. Additionally, queuing theory is used to account for the waiting energy and delay consumption of offloaded jobs, and the offloading choice issue is converted to a task assignment probability problem. Additionally, the suggested algorithm's complexity was explored, and the convergence performance was confirmed.

Previous studies like [43] suggested a creative approach to offload computations in IoT. The goal of this study was to reduce both the energy consumption and time delays commonly associated with standard edge computing offloading strategies. The authors put forth an enhanced method based on particle swarm optimization (PSO) that involves creating a model and optimization objective function focused on communication and personalized load models for multi-user computation tasks. By using a PSO algorithm to determine the optimal solutions for task offloading strategies, they succeed in decreasing the energy utilization during task allocation. According to their simulations, the suggested methodology works better than two other algorithms in terms of average time delay, energy usage across a variety of mobile devices, and data transmission speeds. The outcomes reveal their strategy's capability to diminish the task execution's lag while using less power; hence, it deals effectively with traditional edge computing problems involving high-energy usage or excessive latency duration. Furthermore, they also examine convergence performance while evaluating how crossover plus mutation rates contribute to their suggested technique.

A novel UAV-assisted MEC system was presented by Shuqi Huang et al. [47]. The authors proposed an architecture for providing computing services to users with low latency requirements by maximizing the UAV's flight height and workload allocation ratio. They designed and developed a functional solution for the optimal UAV flying altitude and for the optimal job allocation ratio. The authors demonstrated the accuracy of the research findings, and it was discovered that the user's horizontal distance from the UAV affects the UAV's optimal flying altitude. Additionally, the impact of the transmit power, bandwidth, and CPU frequency on the optimal job allocation ratio was also examined.

Previous research in [68] implemented a strategy for dealing with the issue of workload handling. More specifically, in this work, the workloads were assigned throughout the fog in an entirely autonomous manner by putting into use a monitor-analyze–plan–execute over a knowledge base (MAPE-K) control loop was defined in implementation problems. Using a control loop to break the problem down into a number of phases to find the best placements is provided by a central coordinating mechanism. The coordination mechanism was used for registering the on-device agents which carried out monitoring and program execution. In situations where coordination mechanisms were dispersed, device-specific agents were responsible for performing the phases of analysis and planning.

4.5. Dynamic Task Allocation

Literature [13] suggested the idea of a "resource pool" made up of service vehicles that move together. They suggest a cooperative task scheduling strategy to reduce the task execution time, based on these idle resources that may be allocated in resource pools. They define the min–max issue with the allowable latency constraint for the job execution time optimization model. Additionally, they formulated problems that take node mobility into account. To find the optimum work allocation strategy and reduce the task execution time for the min–max issue, the authors adapted the max–min fairness algorithm and PSO algorithm, respectively, depending on whether to summon all service cars. The simulation results showed the success of the suggested strategies.

Another effort made by Ammar Awad Mutlag et al. [20] applied edge computing in the healthcare system. The authors presented a critical healthcare task management (CHTM) paradigm for monitoring electrocardiograms (ECGs), in a fog-cloud computing topology. In more detail, this paper showed how fog computing might improve service delivery and network congestion, as well as the limits of cloud computing in terms of real-time healthcare applications. Fog computing does not have a distributed design, and its nodes are diverse and exclusive in how they share resources and space. In order to manage crucial activities, the suggested CHTM model intends to offer scheduling, resource sharing, interoperability, and dynamic task allocation. The model suggested a multi-agent system to completely control the network from the edge to the cloud, as well as a resource scheduling model for fog nodes. The outcomes of the simulation demonstrated how the suggested approach lowers the network utilization, reaction time, network latency, energy consumption, and instance cost. Finally, the authors discussed potential future study options as well as the comparisons of the suggested model with relevant work. Overall, the CHTM model has met its performance requirements and offers an effective resource scheduling plan for crucial healthcare activities between the fog node layer, edge layer, and cloud.

Tristan Braud et al. [23] provided a model for a multi-server task allocation method to optimize allocation over various connections. A solution for mobile augmented reality (MAR) and the ways in which mobile devices are limited by the processing and latency requirements of MAR apps was provided. The authors illustrated the durability of the system in situations of network instability using simulations and real-world trials. The authors also go over 802.11ax and the forthcoming 5G technologies. A task dependency graph, optimization methods for related tasks, and a scheduling algorithm for work distribution over several wireless links to D2D, cloud, and edge servers were presented. The authors' contributions included conducting multiple simulations and implementing a real-world multipath, multi-server mapping application.

A recently published paper by Yan Chen et al. [41] addressed the issue of dynamic task allocation and service migration (DTASM) in edge-cloud IoT systems in order to improve the task allocation strategy and reduce the load transferred to the cloud server while fulfilling migration, latency, and computing capacity restrictions. The work suggested a deep reinforcement learning (DRL)-based solution. A cloud server was used to administer the whole system, together with a number of RAN nodes and IoT users running various sorts of apps. Extensive simulations were used to test the suggested technique, and the results demonstrate that it outperforms other benchmarks' task allocation strategies. Investigating the DTASM problem in a heterogeneous edge-cloud IoT system that needs dynamic task allocation and offering a DRL-based solution to the problem of a large discrete action space are two contributions made by this study. Overall, the study examined the service latency restriction and the smooth service migration requirement, making it appropriate for edge-cloud systems with several IoT users.

Moreover, Ping-Chun Huang et al. [44] addressed the problems of how edge computing can become a viable tool for reducing the communication lag. In order to balance the workload across edge servers and reduce the transmission distance, the authors suggested a load-balancing model that takes the computational load on edge servers as well

as transmission distance into account. Edge server placement (ESP) in conjunction with the proposed algorithm performed better than conventional heuristics. Furthermore, the proposed method was more successful in integrating work allocation and server placement, according to their simulation findings.

Bartosz Kopras et al. [46] analyzed task distribution in latency-constrained fog computing networks, where fog nodes (FNs) are closer to the network edge and have lesser computational capabilities than the cloud resources. An optimization issue to reduce task-related energy for transmission and processing with delay limitations was presented in this study. The optimization takes into account both the workload distribution across the nodes and the CPU frequency at each active node. Two useful algorithms termed energy-efficient resource allocation (EEFFRA) and low-complexity energy-efficient resource allocation (LC-EEFFRA) were developed after the issue was converted with successive convex approximation and decomposed using the primal and dual decomposition techniques. The utilization of EEFFRA/LC-EEFFRA greatly decreased the count of computational requests that did not meet the expected delay requirements. Additionally, the authors made use of dynamic voltage and frequency scaling (DVFS) to reduce energy usage while still meeting delay requirements. Modeling the energy consumption and delays associated with transmission and processing in the fog and cloud layers, as well as putting forth and resolving a challenging optimization problem, were the paper's key contributions.

A recent paper by Mingjin Zhang et al. [49] proposed an entirely novel edge-native task scheduling system (ENTS) that co-schedules networking and computing resources to improve the performance of edge-native applications. Although Kubernetes is now the de facto standard for container orchestration, it does not enable edge-native apps or take into consideration their unique throughput and latency needs. This paper includes the case study of a video analytics application to show how task distribution and data flow scheduling enable ENTS to maximize work throughput. To tackle the mixed-integer nonlinear problem, this paper offers two online methods. The research utilized metrics like task throughput and the average waiting time to compare the performance of ENTS to different baseline techniques on a real-world testbed. The findings demonstrated the necessity for specialized scheduling systems for strengthening edge-native applications by demonstrating considerable gains in work throughput and latency. This study also offered suggestions for potential future advancements, including the creation of more sophisticated algorithms for group job scheduling and the incorporation of software-defined networking into the network controller.

Furthermore, Shida Lu et al. [56] suggested a network architecture of cloud edge fusion for the MEC system to enable thorough query and computing for real-time applications. By utilizing MEC's short distance from the user equipment (UE), difficult computational tasks can be executed by UE and useless data can be removed before flowing to the cloud. A suggested distributed task scheduling technique was based on the network architecture to deal with complicated computational tasks; several MEC servers cooperate, work in parallel, and eventually accelerate the task execution response times. To demonstrate that this approach performs better, the authors ran many simulation trials.

A task allocation optimization strategy for distributed edge networks to reduce energy usage while achieving a high quality-of-service was published by Philippe Buschmann et al. [60]. ILP, PSO, and DRL are the three optimization techniques that were compared in this paper. PSO outperforms ILP in smaller problems, but DRL is better suited to larger problems and has the lowest upper bound for the optimality gap of the three techniques. According to the study's findings, the performance of the three techniques varies depending on the size of the job allocation problem in the edge networks. The PSO is more suited for smaller issue sizes, whereas DRL is better suited for bigger ones. According to the findings, the extended PSO algorithm is unreliable and useless for task allocations of greater than 20 and 60, respectively. The study highlighted the potential future research areas, such as investigating the trade-offs between speed and optimality in ILP problems that prioritize previously assigned tasks, and exploring the field of heterogeneous networks.

Furthermore, Junling Yu [64] suggests a binary particle swarm optimization (BPSO)-based MEC application in the classroom assessment system for English education. It addresses the NP-hard problem of decoupling the channel resource allocation problem, proposes a multi-user and multi-MEC scenario based on mobile edge computing to maximize total revenue to finish the work, and formulates an objective function to reduce the task execution cost. This study uses big data allocation research from mobile edge computing resources for the application of the English teaching classroom assessment, accelerating the growth of classroom instruction. It offers recommendations on how to enhance and put into practice the English education assurance system.

4.6. Machine-Learning-Based Task Allocation

A method for online scheduling optimization for requests based on DAGs in edge computing networks was proposed by Yaqiang Zhang et al. [28]. The process was characterized as MDP, where the best task allocation strategy was learned at each decision step via temporal-difference learning. Temporal restrictions between concurrent requests were satisfied while the system's long-term latency and energy use were reduced. In comparison to state-of-the-art approaches, the suggested mechanism exhibits promising results in lowering long-term delay and energy usage. The offloading of complicated structured tasks and pertinent strategies for task scheduling in edge networks are also reviewed and discussed in this paper. The suggested approach attempts to ensure the quality of experience (QoE), particularly when requests are latency-sensitive and include a lot of network traffic.

A work published by Zeina Houmani et al. [54] addressed the issues with managing deep learning applications over an edge-cloud architecture where data are evaluated and transported between resources at the network's edge. Trade-offs between accuracy and latency are important for DL applications since they need outcomes close to real-time with a precision that the user has set as acceptable. When trade-off improvements are possible, the research recommended a data-driven scheduling technique and a data management strategy that lowers the resolution of incoming data. This architecture is for time-critical DL operations. The pipeline was deployed on distributed resources using the suggested system, which also has the ability to monitor each job separately to maintain the system performance. To effectively manage the full DL pipeline in practical deployments while resolving trade-offs between quality of service (QoS) indicators, the paper emphasizes the need for resource and data management solutions. It also highlights related developments in edge computing, microservices, and edge-enhanced data analytic systems. The architecture of the system has three levels: data management, infrastructure, and workflow management. The developer's requirements were met by the system's timeliness and accuracy thanks to the data management level's selection of the data quality distribution for data sources. The study defines the system's objective and analytical models to evaluate the accuracy and end-to-end latency of K data sources. The study looked at object detection in a multi-user setting on Grid'5000 and found that it increased the average system makespan by up to 54.4% compared to a cloud-only setup. To improve this, future research should develop a resource allocation tool for deep learning workflows that considers load, tasks, and resource requirements to meet performance constraints.

Another work [38] studied the topic of load balancing as well as task allocation in computational infrastructures for smart cities that include edge and cloud computing. The authors provided an RL method for task distribution that takes into account the differences between nodes and tasks, as well as their interdependencies and time delays. The authors showed a simulator to evaluate their method with existing allocation methods and introduce an abstract description of the computing infrastructure for smart cities. The suggested method handles dynamic changes well and evenly distributes the workload across computing components. The contribution of this study was to find ways to optimize resource usage for smart city applications.

A new tool to guide the task controller in making decisions (i.e., the allocation process) was provided by Madalena Soula et al. [42]. Such decisions are dynamically influenced by the effectiveness of the edge computing nodes and the updated datasets. The authors assumed that TCs assign various jobs to various edge computing nodes simultaneously as they process the incoming tasks in batches. A clustering-based allocation method and a bio-inspired model (i.e., a modified version of the well-known PSO technique) were both proposed as alternative allocation techniques. The goal of both models was to identify the best allocations at a given moment in time. They relied on procedures that did not require any training. In order to show the cost for each allocation based on the demands of the jobs and the condition of the edge computing nodes, they also considered the cost of allocation as stated in their past work [85]. The cost of allocation is also used as an indication for calculating the ranks of edge computing nodes, resulting in a "rewarding" mechanism for resolving the issue at hand.

A method for handling large-scale tasks in edge computing was presented by [51] using the federated learning-based optimization methodology (FLOM). To accomplish accurate task categorization, global load balancing, and lower task processing costs, FLOM combines FL and deep feature learning approaches. The suggested method incorporates an FL architecture that permits data exchange and model parameter updates across several edge computing and cloud centers without disclosing sensitive information to outsiders. To learn the deep features of task requests and hosts in the substrate network, the authors provide a deep network model. The experimental findings show that the FLOM technique outperforms other approaches and efficiently handles large-scale task categorization and allocation. In edge computing environments, the research emphasizes the necessity for effective task allocation methods that can identify independently jobs while achieving system load balancing. The suggested method has several potential applications in industries including intelligent manufacturing, intelligent IoT, and smart cities.

Previous research [53] tried to apply edge computing in smart cities in order to implement a task distribution among the various nodes. The authors researched a variety of allocation techniques, from stochastic allocation to intuitive methods to integer programming-based mathematical optimization. Furthermore, a demonstration of how efficient job distribution in HEµCs offered appreciable gains in total performance. The authors also showed that it is completely possible to execute integer programming at the periphery with little overhead (less than 2% of the total makespan time in their tests in the worst case scenario) for the size of the scheduling they performed. Their research contribution was that they defined and provided a motivation for the idea of heterogeneous edge micro-clusters (HEµCs) and also that they showed the examples of common HEµC compute platform tasks. This research demonstrated the effectiveness of using mathematical optimization as a method for batch job distribution in HEµCs.

Ishihara et al. [57] proposed a task allocation method where all agents involved in executing tasks can express their preference for which tasks they want to work on, and the manager agent then allocates tasks based on these preferences and system performance requirements, without employing any extra knowledge of the agents' capabilities or state. The authors also took into account the manager's particularity, which means that each manager has a different set of customer requests based on the place and the time. Therefore, by balancing their workloads, agents must be able to dynamically select which manager to support with the specified tasks. The author suggests that, when making decisions about service choice, justice and societal considerations should be given priority over the concept of centralized control. This is because centralized control is not suitable for existing applications, where services are provided by multiple independent organizations.

Lastly, Liu and Liu [58] proposed an approximate technique for task allocation that uses the least amount of overall energy while taking into account multi-task parameters in MEC. Taking into account the binary computation offloading mode and restricted frequency subchannels, the authors simulated the multi-task allocation problem, resulting in an integer programming problem that is severely NP-hard. It is the first of its type, and

trials demonstrated that the proposed PTAS algorithm for multi-task allocation in MEC can locate nearly optimum solutions while striking a decent compromise between speed and quality. The study recognized the need to develop an effective solution to address the multi-task allocation problem.

4.7. QoS Task Allocation

A recent article [26], provided a summary of the techniques, advantages, and limitations of the G-PATA approach. This methodology is unique in that it simultaneously optimizes conflicting objectives, namely the reduction in energy consumption at the rational edge and the minimization of processing time at the server, while also fulfilling specific privacy requirements. The authors discussed the challenge of task allocation in edge computing systems that take into account privacy concerns for delay-sensitive applications related to social sensing. They also formally outlined the problem's goals after presenting the task model, underlying assumptions, and privacy model they utilized.

Furthermore, Zhou Li et al. [32] presented edge computing into mobile crowd sensing (MCS) and suggested a three-tier architecture, in which they presented ideas for task distribution and user data input. The edge servers transmitted the job requirements to users during task allocation so that exact personal information is no longer required. Edge servers enable participants to contribute high-quality altered data without compromising their privacy.

The authors outline the challenge of sensing cost reduction while maintaining the privacy and provide the data sensing mechanism with user privacy preserved (DS-UPP) method as a solution. In DS-UPP, they created a compressive sensing-based method to reduce the quantity of required sensing data and an LDP-based algorithm to safeguard the privacy of the participants. They also examined the DS-UPP that satisfies e-differential privacy. Taking into account the limitations posed by the privacy budget and task recovery error, the authors established both the minimum and maximum number of participants required to mathematically complete the task, along with the expected amount of data that each participant should contribute. Additionally, through the use of a simulator, they conducted in-depth simulations to assess the effectiveness of the DS-UPP. After applying the PrivKV algorithm for comparison, according to the experimental findings, DS-UPP may, on average, cut the cost of sensing by over 90% while still maintaining privacy and data quality standards.

A method to lower the overall latency on an MEC platform by providing a work allocation mechanism was proposed by Katayama et al. [48]. Three different server types were available on this platform: a dedicated cloud server, a MEC server, and a shared MEC server. The authors initially calculated the processing time and transmission delay for different types of servers to determine the time taken for a task to be submitted and a response to be received. They used queuing theory to determine the transmission latency for the shared MEC server, with the bottleneck node being modeled as an M/M/1 queueing model. To minimize the overall latency for all tasks, they formulated an optimization problem for task allocation. Tasks may be properly distributed across the MEC servers and cloud servers by resolving this optimization challenge. Even with the use of a meta-heuristic approach like the genetic algorithm, the computation time is still exceedingly long. As a result, the authors also suggested a heuristic technique to find the roughly ideal answer more quickly. A core algorithm and three supporting algorithms make up this heuristic algorithm's four components. Tasks are split into two groups in this method, and task distribution is carried out for each group. The examination of the performance of their suggested heuristic algorithm in comparison to the results provided by the genetic algorithm and other techniques was provided. The task allocation methods are relatively simple and can be easily implemented on an MEC platform. However, there is a risk of falling into a local minimum when using the suggested heuristic approach. To address this issue, the authors utilized the random search approach (ARSET) and heuristic random optimization (HRO) to avoid getting trapped in local minimums.

4.8. Resource-Aware Task Allocation

The authors in [14] employed their proposed model to simulate a practical vehicular environment and investigate the problem in a large-scale network. A small part of the source task was processed at the vehicle, and the remaining task was offloaded at other vehicles and then at a vehicular edge computing server level. This lowers the cost of the entire system while also allowing for the utilization of ample vehicle resources and a reduction in the strain on the overworked VEC server. The MAP algorithm was used in the VEC scenario which enabled each vehicle to choose its adjacent cars depending on the finest resources available at the lowest cost. Additionally, they estimated the transmission rates for V2I and V2V communication while taking into account realistic assumptions. The determination of whether a job should be computed locally, on a nearby vehicle, or at the vehicle edge computer (VEC) was based on the percentage of the job and was conditional on two factors: the maximum allowable delay and the vehicle's stay time. Finally, by contrasting it with other methods, they could assess how various variables and vehicular contexts affect their MAP task offloading approach.

In another paper published by Chen et al. [15], the authors suggest a cooperative learning process that makes use of simulated and real-time captured data to decrease the quantity of data required to produce a trustworthy data-driven model. The authors employed the DCTA technique to distribute tasks in a data-driven system. Through both a trace-driven simulation and a brand-new, in-depth real-world AIOps case study that connects theory and practice with a novel architecture, a key component was designed within an AIOps system. The authors also assessed numerous distinct work allocation methodologies. At the end, the DCTA method significantly reduces the processing time by 3.24 times and saves 48.4% of the energy consumption compared to state-of-the-art approaches when addressing task importance for multi-task learning (MTL) using task allocation in time-constrained and integrated management (TATIM).

An optimization problem that aims to minimize the total number of users and transmission delays using a multistack RL method was presented by Wang et al. [16]. The results of this study showed that each base station (BS) opted to allocate a greater number of downlink subcarriers and transmit power in the downlink direction to a user whose task needed to be processed by the MEC server, to reduce the maximum delay among all users. The preference of each BS was to give more uplink subcarriers and a greater uplink transmit power to a user who needs to locally complete a job. The authors presented a novel strategy for load balancing in edge computing by allocating tasks through intermediary nodes.

Xue and An [21] analyzed a network situation with several mobile edge server nodes (MSNs) and a number of edge devices (EDs), each of which had an MEC server to offer wireless and computation resources. Their study concentrated on ED-specific tasks, which could be broken into any number of smaller portions for both local and remote computing needs. Data communication during task loading was achieved using NOMA technology to increase the resource usage and MEC performance. The authors took into account the limitations of wireless and communication resources, and they formulated a combined optimization problem of task loading and resource allocation to increase the system's capacity for job processing. To address the defined MINLP problem with the characteristics of the objective function, the authors divided the original problem into two subproblems: the resource allocation (RA) problem and the task allocation (TA) problem. Then, they further divided the RA problem by allocating them as either computation and communication resources. Initially, the authors assumed that the power distribution of subchannels was equal when allocating communication resources. The researchers approached the subchannel allocation problem in the context of a two-sided matching process between mobile terminals (MTs) and subchannels. They proposed a sub-optimal algorithm with low complexity to allocate subchannels, and then treated the transmission power allocation as a convex optimization problem, using the Lagrange multiplier approach. A task allocation method based on resource allocation was used to overcome the task allocation issue.

Computer simulation results demonstrated that the suggested task offloading and resource allocation technique enhanced the MEC system's overall performance.

Recently, Canete et al. [37] proposed a set of four modules that work together to support developers in adjusting the deployment process based on the specific requirements of an application and the capabilities of the infrastructure. These modules are designed to function independently while also complementing one another. The work first defined the FMs feature models (FMs) of the infrastructure and application. An FM depicted, based on characteristics, the relationship between the common and variable parts of a product family (either an application family or a system family). FMs were represented as a collection of features that were arranged hierarchically, with parent–child connections between the features and a set of constraints (known as cross-tree constraints) that show the links between the features. The infrastructure consisted of a range of different devices, each with various hardware and software attributes, such as the device type, computational capacity, peripherals, network capabilities, operating system, third-party libraries, and so on. These attributes were related to the types of services or tasks that could be performed on the execution platform, and they pertained to the hardware and software requirements of the applications. The use of feature modeling (FM) helped create the more accurate models of the infrastructure, which is often overlooked in the software product line (SPL) models. This approach allowed the mapping of application features from the application of FM to the software and hardware features of the infrastructure, even when different nodes had different capabilities. To simplify the configuration of the infrastructure's feature model (FM) and manage the evolution of hardware and software characteristics separately, the authors divided the hardware and software features into two separate FMs.

A load-balancing method for MEC servers was published by Chen et al. [39] which operates in ultra-dense networks based on load estimated. The suggested method accounts for the dynamic distribution of user equipment and ad hoc application activities, and it provides a novel idea of task unit load transfer overhead for load balancing that is both low-complexity and high-efficiency. In addition, the study investigated load estimations based on overlapping coverage and user load prediction, and it suggested a three-tiered computing network design made up of devices, edge nodes, and cloud servers that improves load balancing while taking energy use and latency into account. Simulated experiments were performed to verify the efficacy of the remedies that were suggested.

Li et al. [40] integrated the edge computing with MCS and put up a three-tier architecture. The authors worked on creative task allocation and user data submission ideas. The edge servers transmit the job requirements to users during task allocation so that exact personal information is no longer required. Edge servers assist participants in submitting high-quality altered data that protect their privacy. They define the challenge of privacy-preserving sensing cost reduction and provide a solution using DS-UPP. In DS-UPP, a compressive sensing-based method to reduce the quantity of required sensing data and an LDP-based algorithm to safeguard the privacy of the participants were created. Also, the authors examined the DS-UPP's theoretical properties. It has been established that DS-UPP complies with the "differential privacy with the restrictions that privacy budget is" a recovery error. To solve this problem, the authors provided the mathematical lower bound and upper bound of the number of participants needed for the task achievement as well as the typical quantity of data that should be supplied by a participant. Lastly, they applied the algorithm PrivKV for comparison. According to their experimental findings, DS-UPP gained lower costs for sensing by roughly 90% while maintaining privacy and data quality standards.

Moreover, Xuefeng et al. [62] addressed the drawbacks of the aforementioned current technologies. The paper created a platform for intelligent operation inspection that relies on a multi-agent system and is used for the live line measurement of substation equipment. The multi-agent system technology was used in the creation of the operational framework. The paper proposed integrating edge computing into the data processing stage of a multi-agent system, which enables the intelligent operation inspection platform to overcome

problems associated with high latency, network instability, and limited resources. This integration also leads to increased flexibility, reliability, and resource utilization. This design enhances the systematization and intelligence level of substation operation inspection by creating an intelligent operation inspection system for substation live inspection. This covers various aspects such as decision making, wireless communication, data processing, data sensing, and analysis.

Lastly, Cumino and Sargento [70] discussed the deployment of UAVs in 5G and beyond MEC scenarios, along with how they may be used as mobile base stations and communication relays. This paper described the difficulties in implementing flying edge computing. The authors suggested a multi-tier architecture using UAVs with various mobility models, and assesses the effectiveness of UAVs as edge nodes. This study also described the benefits of a multi-tier viewpoint and talked about the topology and design of a smart city network. The article concludes by summarizing related research on UAV networks, job offloading, and edge computing models that might increase the available computational resources, and it highlights current research gaps and future directions.

5. Discussion

Edge computing is a rapidly emerging and dynamically diversifying field that allows for data processing to occur closer to the source of data, rather than solely relying on centralized data centers. This review offers a thorough analysis of edge computing task allocation techniques. This draws attention to the difficulties in choosing the best place for tasks depending on resources such as the processing power, storage, and network bandwidth as well as in adjusting to the network's dynamic nature. The research explores several task allocation strategies, including centralized, decentralized, hybrid, and machine learning algorithms, and assesses their efficiency in completing challenging tasks. After applying the inclusion and exclusion criteria, a total of 69 papers were included in our SLR. Based on the network types, deep learning compression techniques, and job allocation optimization algorithms utilized in edge computing, the research gives a comparative analysis of task allocation approaches. Overall, the paper highlights the difficulties and complexities of allocation in edge computing and underlines the emerging need to achieve a compromise between a number of competing goals, including energy economy, data privacy, security, latency, and QoS.

The optimization techniques are the main source of challenge regarding task distribution. Dynamic scaling, resource-aware task distribution, resistive PSO, and meta-heuristic optimization approaches are some of the typical optimization strategies. The selection of an optimization approach must take into account the particular needs of the application and strike a balance between a number of competing goals, including energy efficiency, data privacy, security, latency, and QoS. This review assesses the efficacy of each method that is suggested for further study, emphasizing the difficulty of job distribution in edge computing and the need to compromise between conflicting objectives. These techniques are applied in various networks such as IoT networks, MEC networks, fog networks, vehicular networks, 5G networks, and distributed networks.

Deep learning compression methods play a decisive role in choosing the appropriate technique for task allocation in edge computing. General deep learning compression, conventional neural networks, knowledge distillation, auto-encoders, and pruning are among the deep learning compression techniques that merit additional study. These techniques were applied in a number of suggested task optimization algorithms in many network types, including satellite, wireless, cellular, MEC, and the IoT [86–90]. The development of more effective and efficient task allocation mechanisms for edge computing may be influenced by further investigation into these compression techniques.

The main contribution of our work is providing a solid answer to each of the research questions we made in Section 2. Regarding RQ1, the most contemporary task allocation techniques in edge computing are resource-aware task allocation, distributed, dynamic, machine learning-based, energy-efficient, quality-of-service-aware, collabora-

tive, and Contex-aware allocation techniques which are also shown in Figure 3. As far as answering RQ2, the most used optimization techniques are the swarm, reinforcement learning, JTORA, multi-agent, game theory, heuristic approach, and last auction-based algorithms, which are also shown in Figure 4. When it comes to answering RQ3, the most used computer networks in edge computing are the MEC, the EC, vehicular, and lastly, the 5G. After answering the RQs, it is clear that every task allocation technique, optimization algorithm, and network has its own strengths and must be carefully chosen according to the application. For example, applications such as edge-based video analytics use the resource-aware task allocation method; the distributed method is used in edge computing for IoT systems; and dynamic traffic management systems are using the dynamic task allocation method; predictive maintenance systems are based on machine learning task allocation methods; the energy-aware methods are usually used in IoT systems; and finally, the QoS-aware allocation method is used in real-time video streaming applications. The optimization algorithms and network types are depending on the application and the topology that needs addressing.

Future research [91–95] in the area of task allocation in edge computing must deal with the difficulties of adjusting to the dynamic nature of each network and establishing a balance of trade-offs between several competing objectives such as energy efficiency, data privacy, security, latency, and QoS. In addition, it is necessary to create new techniques and algorithms that can manage the growing complexity of edge computing networks. Further study is required on hybrid strategies that mix centralized and decentralized methods to maximize work distribution while taking into account the special features of edge computing settings. Also, further research is needed in the field of task allocation optimization using machine learning methods. Lastly, research must continue to concentrate on enhancing the task allocation's efficacy and efficiency so that edge computing systems can manage workloads that are getting more complicated.

6. Conclusions

The challenge of task allocation in edge computing is complex and demanding. It calls for a thorough analysis of the needs of each application. Multiple approaches for distributing tasks on edge intelligence devices, including centralized, decentralized, hybrid, and machine learning algorithms, were explored and assessed in this systematic review of task allocation methods in edge computing. Finding the optimum location for each task based on processing power, storage, and network bandwidth needs, as well as being able to adjust to the network's dynamic nature, are the key issues in task allocation. This review investigated and assessed studies for many competing objectives, including data privacy, security, energy efficiency, QoS, IoT, task analysis, fog computing, and computation-intensive tasks, among other topics. We thoroughly covered all the 69 articles that were identified by our Scopus-based keyword combination search. We also note that the number of publications on the topic has seen an upward trend over the years, indicating the importance of task allocation in edge computing.

A comprehensive understanding of task allocation approaches in edge computing was attained via the analysis of these papers. More specifically, we classified and contextualized the different types of task allocation approaches that are used, also including the network types and the deep learning compression methods for optimal task offloading. The suggested approaches include dynamic scaling for effective resource allocation in fog computing as well as resource-aware task allocation based on the priority queue and available computing resources. Additionally, simulations and field testing were used in each study to assess the efficacy of the recommended approaches; something that we reported on.

Overall, the analysis comes to the conclusion that task allocation in edge computing is a difficult but necessary problem that the requires careful consideration of the application's specific needs. Research studies on alternative job distribution strategies, competing goals, and methods such as simulation and field testing might shed light on the most effective

workload distribution on edge intelligence devices. Our review provides robust evidence of the growing significance of task allocation in edge computing for IoT devices, industrial applications (under the umbrella of Industry 5.0), driverless cars, UAVs, and augmented reality, among other applications.

Author Contributions: Conceptualization, V.P., P.A., T.L. and D.K.; methodology, V.P., P.A., A.N. and D.K.; data curation, V.P. and K.M.; writing—original draft preparation, V.P., P.A. and D.K.; writing—review and editing, T.L., K.M. and A.N.; visualization, V.P., P.A. and K.M.; supervision, D.K.; project administration, D.K.; funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work received funding from the European Union's Horizon Europe Research and Innovation Programme under Grant Agreement No. 101070181. This paper only reflects the authors' views, and the Commission is not responsible for any use that may be made of the information it contains.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AA	Autonomous-Agents
AB	Auction-Based
ACO	Ant Colony Optimization
AP	Access Point
ARIMA	Autoregressive Integrated Moving Average
ARSU	Aerial-Roadside Units
AsP	Assignment Problem
AVA	Application Variability Adaptor
BPSO	Binary Particle Swarm Optimization
BSUB	Block Successive Upper Bound
CFG	Coalition Formation Game
CRAN	Cloud Radio Access Network
CTA	Collaborative Task Allocation
CW	Crowdsourcing Workflows
DAG	Directed Acyclic Graph
DFD	Data-Flow-Driven
DLA	Distributed Learning Automata
DNF	Deep Network Flow
DP	Dynamic Programming
DQN	Deep Q-Learning Network
DQN-D	Deep Q-Learning Network with Double Q-Learning
DRL	Deep Reinforcement Learning
DS-UPP	Data Sensing Mechanism with User Privacy Preserved
DTASM	Dynamic Task Allocation and Service Migration
DTOMALB	Distributed Task Offloading Algorithm Based on Multi-Agent and Load Balancing
DVFS	Dynamic Voltage and Frequency Scaling
EC	Edge Computing
ECLAM	Energy and Latency Minimizer
ECTA	Edge Computing Task Allocation
EDAF	Edge-Deployment Alternatives Finder
EEFFRA	Energy-Efficient Resource Allocation
ELB	Enhanced Load-Balancing
ET	Energy Transmitter
FAP	Fog Access Points
FL	Federated Learning

FLOM	Federated Learning-Based Optimization Methodology
FM	Feature Model
GA	Greedy Algorithm
GEN	Genetic
GSO	Glowworm Swarm Optimization
GT	Game Theory
HAB	High-Altitude Balloon
ILP	Integer Linear Programming
IoT	Internet of Things
IoV	Internet of Vehicles
JTORA	Joint Task Offloading and Resource Allocation
LC-EEFFRA	Low-Complexity Energy-Efficient Resource Allocation
LPA	Location Privacy-Aware
MA	Multi-Agent
MAH	Multi-Agent Approach Networks with Multiple Agents Heuristic
MAP	Mobility-Aware Partial
MAPE-K	Monitor Analyze Plan Execute Knowledge
MAPPO	Multi-Agent Proximal Policy Optimization
MARL	Multi-Agent Reinforcement Learning
MCS	Mobile Crowd Sensing
MDP	Markov Decision Process
MEC	Mobile Edge Computing
MH	Multi-Hop
MIMO	Multiple Input–Multiple Output
MINLP	Mixed-Integer Nonlinear Problem
MITA	Minimum Incremental Task Allocation
MMAS	Max–Min Ant System
MPSO	Modified Particle Swarm Optimization
MRC	Multi-Robot Cooperation
MVAA	Modified Version of the Auction Algorithm
NDF	New Devices Finder
NDN-IoT	Named Data Networking—Internet of Things
NOMA	Non-Orthogonal Multiple Access
OFB	Objective Function Based
PB	Primary-Backup
PFA	Proportional Fairness Algorithm
PSO	Particle Swarm Optimization
QBTD	Quality of Service-based Task Distribution
QoE	Quality of Experience
QoS	Quality of Service
QT	Queueing Theory
RIS	Reconfigurable Intelligent Surface
RL	Reinforcement Learning
RQ	Research Question
RSU	RoadSide Units
SLR	Systematic Literature Review
TAHRC	Task Allocation with Heterogeneous Resources
TCA-IPSO	Two-Edge-Node Cooperative-Task Allocation based on the Improved Particle Swarm Optimization
UAV	Unmanned Aerial Vehicle
VANETs	Vehicular Ad Hoc Networks
VEC	Vehicular Edge Computing
WHAB	Wireless High-Altitude Balloon Networks
WPT	Wireless Power Transfer

References

1. Amanatidis, P.; Karampatzakis, D.; Iosifidis, G.; Lagkas, T.; Nikitas, A. Cooperative Task Execution for Object Detection in Edge Computing: An Internet of Things Application. *Appl. Sci.* **2023**, *13*, 4982. [CrossRef]
2. Nikitas, A.; Michalakopoulou, K.; Njoya, E.T.; Karampatzakis, D. Artificial Intelligence, Transport and the Smart City: Definitions and Dimensions of a New Mobility Era. *Sustainability* **2020**, *12*, 2789. [CrossRef]
3. Thinh, T.Q.; Tang, J.; La, Q.D.; Quek, T.Q.S. Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling. *IEEE Trans. Commun.* **2017**, *65*, 3571–3584. [CrossRef]
4. Cao, K.; Liu, Y.; Meng, G.; Sun, Q. An Overview on Edge Computing Research. *IEEE Access* **2020**, *8*, 85714–85728. [CrossRef]
5. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
6. Shi, W.; Dustdar, S. The Promise of Edge Computing. *Computer* **2016**, *49*, 78–81. [CrossRef]
7. Yang, X.; Rahmani, N. Task scheduling mechanisms in fog computing: Review, trends, and perspectives. *Kybernetes* **2021**, *50*, 22–38. [CrossRef]
8. Asim, M.; Wang, Y.; Wang, K.; Huang, P.Q. A Review on Computational Intelligence Techniques in Cloud and Edge Computing. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *4*, 742–763. [CrossRef]
9. Luo, Q.; Hu, S.; Li, C.; Li, G.; Shi, W. Resource Scheduling in Edge Computing: A Survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2131–2165. [CrossRef]
10. Tsagkis, P.; Bakogiannis, E.; Nikitas, A. Analysing urban growth using machine learning and open data: An artificial neural network modelled case study of five Greek cities. *Sustain. Cities Soc.* **2023**, *89*, 104337. [CrossRef]
11. Wang, F.; Xu, J.; Cui, S. Optimal Energy Allocation and Task Offloading Policy for Wireless Powered Mobile Edge Computing Systems. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2443–2459. [CrossRef]
12. Deng, X.; Li, J.; Liu, E.; Zhang, H. Task allocation algorithm and optimization model on edge collaboration. *J. Syst. Archit.* **2020**, *110*, 101778. [CrossRef]
13. Chen, C.; Chen, L.; Liu, L.; He, S.; Yuan, X.; Lan, D.; Chen, Z. Delay-optimized V2V-based computation offloading in urban vehicular edge computing and networks. *IEEE Access* **2020**, *8*, 18863–18873. [CrossRef]
14. Raza, S.; Liu, W.; Ahmed, M.; Anwar, M.; Mirza, M.; Sun, Q.; Wang, S. An efficient task offloading scheme in vehicular edge computing. *J. Cloud Comput.* **2020**, *9*, 28. [CrossRef]
15. Chen, Q.; Zheng, Z.; Hu, C.; Wang, D.; Liu, F. On-Edge Multi-Task Transfer Learning: Model and Practice with Data-Driven Task Allocation. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 1357–1371. [CrossRef]
16. Wang, S.; Chen, M.; Liu, X.; Yin, C.; Cui, S.; Vincent Poor, H. A Machine Learning Approach for Task and Resource Allocation in Mobile-Edge Computing-Based Networks. *IEEE Internet Things J.* **2021**, *8*, 1358–1372. [CrossRef]
17. Li, G.; Yao, Y.; Wu, J.; Liu, X.; Sheng, X.; Lin, Q. A new load balancing strategy by task allocation in edge computing based on intermediary nodes. *Eurasip J. Wirel. Commun. Netw.* **2020**, *2020*, 3. [CrossRef]
18. Wang, S.; Chen, M.; Yin, C.; Saad, W.; Hong, C.; Cui, S.; Poor, H. Federated Learning for Task and Resource Allocation in Wireless High-Altitude Balloon Networks. *IEEE Internet Things J.* **2021**, *69*, 8843–8854. [CrossRef]
19. Wang, J.B.; Yang, H.; Cheng, M.; Wang, J.Y.; Lin, M.; Wang, J. Joint Optimization of Offloading and Resources Allocation in Secure Mobile Edge Computing Systems. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8843–8854. [CrossRef]
20. Mutlag, A.; Ghani, M.; Mohammed, M.; Lakhan, A.; Mohd, O.; Abdulkareem, K.; Garcia-Zapirain, B. Multi-agent systems in fog–cloud computing for critical healthcare task management model (CHTM) used for ECG monitoring. *Sensors* **2021**, *21*, 6923. [CrossRef]
21. Xue, J.; An, Y. Joint Task Offloading and Resource Allocation for Multi-Task Multi-Server NOMA-MEC Networks. *IEEE Access* **2021**, *9*, 16152–16163. [CrossRef]
22. Zhao, L.; Yang, K.; Tan, Z.; Song, H.; Al-Dubai, A.; Zomaya, A.; Li, X. Vehicular Computation Offloading for Industrial Mobile Edge Computing. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7871–7881. [CrossRef]
23. Braud, T.; Zhou, P.; Kangasharju, J.; Hui, P. Multipath Computation Offloading for Mobile Augmented Reality. In Proceedings of the 18th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2020, Austin, TX, USA, 23–27 March 2020.
24. Zhang, Z.; Li, C.; Peng, S.; Pei, X. A new task offloading algorithm in edge computing. *Eurasip J. Wirel. Commun. Netw.* **2021**, *2021*, 1–21. [CrossRef]
25. Michailidis, E.; Miridakis, N.; Michalas, A.; Skondras, E.; Vergados, D. Energy optimization in dual-ris uav-aided mec-enabled internet of vehicles. *Sensors* **2021**, *21*, 4392. [CrossRef]
26. Zhang, D.; Ma, Y.; Sharon Hu, X.; Wang, D. Toward Privacy-Aware Task Allocation in Social Sensing-Based Edge Computing Systems. *IEEE Internet Things J.* **2020**, *7*, 11384–11400. [CrossRef]
27. Wang, Q.; Shao, S.; Guo, S.; Qiu, X.; Wang, Z. Task allocation mechanism of power internet of things based on cooperative edge computing. *IEEE Access* **2020**, *8*, 158488–158501. [CrossRef]
28. Zhang, Y.; Zhou, Z.; Shi, Z.; Meng, L.; Zhang, Z. Online Scheduling Optimization for DAG-Based Requests through Reinforcement Learning in Collaboration Edge Networks. *IEEE Access* **2020**, *8*, 72985–72996. [CrossRef]
29. Liu, X.; Liu, J.; Wu, H. Energy-Efficient Task Allocation of Heterogeneous Resources in Mobile Edge Computing. *IEEE Access* **2021**, *9*, 119700–119711. [CrossRef]

30. Xue, J.; Wang, Z.; Zhang, Y.; Wang, L. Task Allocation Optimization Scheme Based on Queuing Theory for Mobile Edge Computing in 5G Heterogeneous Networks. *Mob. Inf. Syst.* **2020**, *2020*, 1501403. [CrossRef]
31. Mudassar, M.; Zhai, Y.; Liao, L.; Shen, J. A Decentralized Latency-Aware Task Allocation and Group Formation Approach with Fault Tolerance for IoT Applications. *IEEE Access* **2020**, *8*, 49212–49223. [CrossRef]
32. Ding, X.; Lv, R.; Pang, X.; Hu, J.; Wang, Z.; Yang, X.; Li, X. Privacy-preserving task allocation for edge computing-based mobile crowdsensing. *Comput. Electr. Eng.* **2022**, *97*, 107528. [CrossRef]
33. Zhang, H.; Hao, J.; Li, X. A method for deploying distributed denial of service attack defense strategies on edge servers using reinforcement learning. *IEEE Access* **2020**, *9*, 78482–78491. [CrossRef]
34. Pan, Y.; Jiang, H.; Zhu, H.; Wang, J. Latency Minimization for Task Offloading in Hierarchical Fog-Computing C-RAN Networks. In Proceedings of the IEEE International Conference on Communications, Dublin, Ireland, 7–11 June 2020.
35. Muneeb, M.; Ko, K.M.; Park, Y.H. A fog computing architecture with multi-layer for computing-intensive iot applications. *Appl. Sci.* **2021**, *11*, 11585. [CrossRef]
36. Fu, K.; Ye, J. Computation Offloading Based on Improved Glowworm Swarm Optimization Algorithm in Mobile Edge Computing. *J. Phys. Conf. Ser.* **2021**, *1757*, 012195. [CrossRef]
37. Cañete, A.; Amor, M.; Fuentes, L. Supporting IoT applications deployment on edge-based infrastructures using multi-layer feature models. *J. Syst. Softw.* **2022**, *183*, 111086. [CrossRef]
38. Alorbani, A.; Bauer, M. Load Balancing and Resource Allocation in Smart Cities using Reinforcement Learning. In Proceedings of the 2021 IEEE International Smart Cities Conference, ISC2 2021, Virtual Conference, 7–10 September 2021.
39. Chen, W.; Zhu, Y.; Liu, J.; Chen, Y. Enhancing mobile edge computing with efficient load balancing using load estimation in ultra-dense network. *Sensors* **2021**, *21*, 3135. [CrossRef]
40. Li, Z.; Song, Z.; Chen, X. Privacy-Preserving Cost Minimization in Mobile Crowd Sensing Supported by Edge Computing. *IEEE Access* **2020**, *8*, 121920–121928. [CrossRef]
41. Chen, Y.; Sun, Y.; Wang, C.; Taleb, T. Dynamic Task Allocation and Service Migration in Edge-Cloud IoT System Based on Deep Reinforcement Learning. *IEEE Internet Things J.* **2022**, *9*, 16742–16757. [CrossRef]
42. Soula, M.; Karanika, A.; Kolomvatsos, K.; Anagnostopoulos, C.; Stamoulis, G. Intelligent tasks allocation at the edge based on machine learning and bio-inspired algorithms. *Evol. Syst.* **2022**, *13*, 221–242. [CrossRef]
43. Li, A.; Li, L.; Yi, S. Computation Offloading Strategy for IoT Using Improved Particle Swarm Algorithm in Edge Computing. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9319136. [CrossRef]
44. Huang, P.C.; Chin, T.L.; Chuang, T.Y. Server Placement and Task Allocation for Load Balancing in Edge-Computing Networks. *IEEE Access* **2021**, *9*, 138200–138208. [CrossRef]
45. Long, T.; Ma, Y.; Wu, L.; Xia, Y.; Jiang, N.; Li, J.; Fu, X.; You, X.; Zhang, B. A novel fault-tolerant scheduling approach for collaborative workflows in an edge-IoT environment. *Digit. Commun. Netw.* **2022**, *8*, 911–922. [CrossRef]
46. Kopras, B.; Bossy, B.; Idzikowski, F.; Kryszkiewicz, P.; Bogucka, H. Task Allocation for Energy Optimization in Fog Computing Networks With Latency Constraints. *IEEE Trans. Commun.* **2022**, *70*, 8229–8243. [CrossRef]
47. Huang, S.; Zhang, J.; Wu, Y. Altitude Optimization and Task Allocation of UAV-Assisted MEC Communication System. *Sensors* **2022**, *22*, 8061. [CrossRef]
48. Katayama, Y.; Tachibana, T. Optimal Task Allocation Algorithm Based on Queueing Theory for Future Internet Application in Mobile Edge Computing Platform. *Sensors* **2022**, *22*, 4825. [CrossRef]
49. Zhang, M.; Cao, J.; Yang, L.; Zhang, L.; Sahni, Y.; Jiang, S. ENTS: An Edge-native Task Scheduling System for Collaborative Edge Computing. In Proceedings of the Proceedings—2022 IEEE/ACM 7th Symposium on Edge Computing, SEC 2022, Seattle, WA, USA, 5–8 December 2022.
50. Jamalipour, A.; Abkenar, F. Efficient Task Allocation Protocol for a Hybrid-Hierarchical Spatial-Aerial-Terrestrial Edge-Centric IoT Architecture. *IEICE Trans. Commun.* **2022**, *2022*, 116–130. [CrossRef]
51. Wu, C.; Li, Y. FLOM: Toward Efficient Task Processing in Big Data with Federated Learning. *Secur. Commun. Netw.* **2023**,*2023*, 9821793. [CrossRef]
52. Garcia, J.; Aguiló, F.; Asensio, A.; Simó, E.; Zaragozá, M.; Masip-Bruin, X. Data-flow driven optimal tasks distribution for global heterogeneous systems. *Future Gener. Comput. Syst.* **2021**, *2021*, 792–805. [CrossRef]
53. Alhaizaey, Y.; Singer, J.; Michala, A. Optimizing task allocation for edge micro-clusters in smart cities. In Proceedings of the Proceedings—2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2021, Pisa, Italy, 7–11 June 2021.
54. Houmani, Z.; Balouek-Thomert, D.; Caron, E.; Parashar, M. Enabling microservices management for Deep Learning applications across the Edge-Cloud Continuum. In Proceedings of the Proceedings—Symposium on Computer Architecture and High Performance Computing, Virtual Event, 26–29 October 2021.
55. Li, D.; Qin, N.; Li, B.; Jing, X.; Du, C.; Wan, C. Resource allocation method based on massive MIMO NOMA MEC on distribution communication network. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Surakarta, Indonesia, 24–25 August 2021.
56. Lu, S.; Gu, R.; Jin, H.; Wang, L.; Li, X.; Li, J. QoS-Aware Task Scheduling in Cloud-Edge Environment. *IEEE Access* **2021**, *9*, 56496–56505. [CrossRef]

57. Ishihara, Y.; Sugawara, T. Multi-agent task allocation based on the learning of managers and local preference selections. *Procedia Comput. Sci.* **2020**, *2020*, 675–684 [CrossRef]
58. Liu, J.; Liu, X. Energy-efficient allocation for multiple tasks in mobile edge computing. *J. Cloud Comput.* **2022**, *11*, 71. [CrossRef]
59. Xu, W.; Duan, H.; Chen, X.; Huang, J.; Liu, D.; Chen, Y. Blockchain-based multi-skill mobile crowdsourcing services. *Eurasip J. Wirel. Commun. Netw.* **2022**, *2022*, 55. [CrossRef]
60. Buschmann, P.; Shorim, M.; Helm, M.; Bröring, A.; Carle, G. Task Allocation in Industrial Edge Networks with Particle Swarm Optimization and Deep Reinforcement Learning. In Proceedings of the ACM International Conference Proceeding Series, Delft, Netherlands, 7–10 November 2022.
61. Liu, Y.; Cai, Y.; Liu, A.; Zhao, M.; Hanzo, L. Latency Minimization for mmWave D2D Mobile Edge Computing Systems: Joint Task Allocation and Hybrid Beamforming Design. *IEEE Trans. Veh. Technol.* **2022**, *71*, 12206–12221. [CrossRef]
62. Xuefeng, N.; Yao, G. Design of intelligent operation inspection platform based on the multi-agent system for live line measurement of substation. In Proceedings of the Journal of Physics: Conference Series, Foshan, China, 11–13 August 2022.
63. Wang, C.; Jia, B.; Yu, H.; Chen, L.; Cheng, K.; Wang, X. Attention-aided Federated Learning for Dependency-Aware Collaborative Task Allocation in Edge-Assisted Smart Grid Scenarios. In Proceedings of the 2022 IEEE/CIC International Conference on Communications in China, ICCC, Foshan, China, 11–13 August 2022.
64. Yu, J. Mobile Edge Computing Application in English Teaching Classroom Evaluation System Based on BPSO Algorithm. *Mob. Inf. Syst.* **2022**, *2022*, 3744523. [CrossRef]
65. Sun, J. Certificateless Batch Authentication Scheme and Intrusion Detection Model Based on the Mobile Edge Computing Technology NDN-IoT Environment. *J. Funct. Spaces* **2022**, *2022*, 5926792. [CrossRef]
66. Liu, J.; Wu, Z.; Liu, J.; Tu, X. Distributed Location-Aware Task Offloading in Multi-UAVs Enabled Edge Computing. *IEEE Access* **2022**, *10*, 72416–72428. [CrossRef]
67. Qiu, L.; Zhang, Y.; Wang, Y.; Shen, Y.; Yuan, J.; Yin, R. Resource Optimization in MEC-Assisted Multirobot Cooperation Systems. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 377225. [CrossRef]
68. Eyckerman, R.; Reiter, P.; Mercelis, S.; Latre, S.; Marquez-Barja, J.; Hellinckx, P. A Generalized Approach For Practical Task Allocation Using A MAPE-K Control Loop. In Proceedings of the International Conference on ICT Convergence, Jeju Island, Republic of Korea, 20–22 October 2021.
69. Sun, Y.; Xie, X.; Wu, F.; Zhang, S.; Xu, S.; Wu, Y. Application Loading and Computing Allocation for Collaborative Edge Computing. *IEEE Access* **2021**, *9*, 158481–158495. [CrossRef]
70. Cumino, P.; Sargento, S. Flying Mobile Edge Computing towards 5G and beyond: An Overview on current use cases and challenges. In Proceedings of the 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2020, Porto, Portugal, 20–22 July 2020.
71. Pliatsios, D.; Sarigiannidis, P.; Lagkas, T.D.; Argyriou, V.; Boulogeorgos, A.A.A.; Baziana, P. Joint Wireless Resource and Computation Offloading Optimization for Energy Efficient Internet of Vehicles. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 1468–1480. [CrossRef]
72. Elgendy, I.A.; Meshoul, S.; Hammad, M. Joint Task Offloading, Resource Allocation, and Load-Balancing Optimization in Multi-UAV-Aided MEC Systems. *Appl. Sci.* **2023**, *13*, 2625. [CrossRef]
73. Pliatsios, D.; Lagkas, T.; Argyriou, V.; Sarigiannidis, A.; Margounakis, D.; Saoulidis, T.; Sarigiannidis, P. A Hybrid RF-FSO Offloading Scheme for Autonomous Industrial Internet of Things. In Proceedings of the IEEE INFOCOM 2022—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York, NY, USA, 2–5 May 2022; pp. 1–6. [CrossRef]
74. Chai, F.; Zhang, Q.; Yao, H.; Xin, X.; Gao, R.; Guizani, M. Joint Multi-task Offloading and Resource Allocation for Mobile Edge Computing Systems in Satellite IoT. *IEEE Trans. Veh. Technol.* **2023**,*72*, 7783–7795. [CrossRef]
75. Fan, W.; Su, Y.; Liu, J.; Li, S.; Huang, W.; Wu, F.; Liu, Y. Joint Task Offloading and Resource Allocation for Vehicular Edge Computing Based on V2I and V2V Modes. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 4277–4292. [CrossRef]
76. Kang, H.; Chang, X.; Mišić, J.; Mišić, V.B.; Fan, J.; Liu, Y. Cooperative UAV Resource Allocation and Task Offloading in Hierarchical Aerial Computing Systems: A MAPPO Based Approach. *IEEE Internet Things J.* **2023**, *10*, 10497–10509. [CrossRef]
77. Wang, S.; Gong, Y. Joint Power Control and Task Offloading in Collaborative Edge-Cloud Computing Networks. *IEEE Internet Things J.* 2023, *early access*. [CrossRef]
78. Wang, Z.; Sun, Y.; Liu, D.; Hu, J.; Pang, X.; Hu, Y.; Ren, K. Location Privacy-Aware Task Offloading in Mobile Edge Computing. *IEEE Trans. Mob. Comput.* 2023, *early access*. [CrossRef]
79. Liu, L.; Zhao, M.; Yu, M.; Jan, M.A.; Lan, D.; Taherkordi, A. Mobility-Aware Multi-Hop Task Offloading for Autonomous Driving in Vehicular Edge Computing and Networks. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 2169–2182. [CrossRef]
80. Wee, B.V.; Banister, D. How to Write a Literature Review Paper? *Transp. Rev.* **2016**, *36*, 278–288. [CrossRef]
81. Snyder, H. Literature review as a research methodology: An overview and guidelines. *J. Bus. Res.* **2019**, *104*, 333–339. [CrossRef]
82. Nikitas, A.; Thomopoulos, N.; Milakis, D. The Environmental and Resource Dimensions of Automated Transport: A Nexus for Enabling Vehicle Automation to Support Sustainable Urban Mobility. *Annu. Rev. Environ. Resour.* **2021**, *46*, 167–192. [CrossRef]
83. Kitchenham, B.; Pearl Brereton, O.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering—A systematic literature review. *Inf. Softw. Technol.* **2009**, *51*, 7–15. Special Section—Most Cited Articles in 2002 and Regular Research Papers. [CrossRef]
84. Scopus. Available online: https://www.scopus.com/ (accessed on 25 July 2023).

85. Karanika, A.; Soula, M.; Anagnostopoulos, C.; Kolomvatsos, K.; Stamoulis, G. Optimized analytics query allocation at the edge of the network. In Proceedings of the Internet and Distributed Computing Systems: 12th International Conference, IDCS 2019, Naples, Italy, 10–12 October 2019; Proceedings 12; Springer: Berlin/Heidelberg, Germany, 2019; pp. 181–190.
86. Roy, S.; Panda, P.; Srinivasan, G.; Raghunathan, A. Pruning Filters while Training for Efficiently Optimizing Deep Learning Networks. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–7. [CrossRef]
87. Allen-Zhu, Z.; Li, Y. Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning. *arXiv* **2023**, arXiv:2012.09816.
88. Wu, H.; Judd, P.; Zhang, X.; Isaev, M.; Micikevicius, P. Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation. *arXiv* **2020**, arXiv:2004.09602.
89. Abd-Alzhra, A.S.; Al-Tamimi, M.S.H. Image Compression Using Deep Learning: Methods and Techniques. *Iraqi J. Sci.* **2022**, pp. 1299–1312. [CrossRef]
90. Mishra, D.; Singh, S.K.; Singh, R.K. Deep Architectures for Image Compression: A Critical Review. *Signal Process.* **2022**, *191*, 108346. .: 10.1016/j.sigpro.2021.108346. [CrossRef]
91. Akhlaqi, M.Y.; Mohd Hanapi, Z.B. Task offloading paradigm in mobile edge computing-current issues, adopted approaches, and future directions. *J. Netw. Comput. Appl.* **2023**, *212*, 103568. [CrossRef]
92. Raeisi-Varzaneh, M.; Dakkak, O.; Habbal, A.; Kim, B.S. Resource Scheduling in Edge Computing: Architecture, Taxonomy, Open Issues and Future Research Directions. *IEEE Access* **2023**, *11*, 25329–25350. [CrossRef]
93. Chaâri, R.; Cheikhrouhou, O.; Koubâa, A.; Youssef, H.; Gia, T.N. Dynamic computation offloading for ground and flying robots: Taxonomy, state of art, and future directions. *Comput. Sci. Rev.* **2022**, *45*, 100488. [CrossRef]
94. Hamdi, A.M.A.; Hussain, F.K.; Hussain, O.K. Task offloading in vehicular fog computing: State-of-the-art and open issues. *Future Gener. Comput. Syst.* **2022**, *133*, 201–212. [CrossRef]
95. Huang, X.; Zhang, B.; Li, C. Incentive Mechanisms for Mobile Edge Computing: Present and Future Directions. *IEEE Netw.* **2022**, *36*, 199–205. [CrossRef]